**TÀI LIỆU MÔ TẢ KẾ HOẠCH TEST (TEST PLAN)**

**TEST PLAN - MiniProject**

Phiên bản 0.2 08/05/11

Tài liệu này mô tả chi tiết kế hoạch test cho dự án **MiniProject**

PHIÊN BẢN CẬP NHẬT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Ngày** | **Version** | **Nội dung sửa đổi chính** | **Người sửa** |
| 01 | 10/11/2010 | 0.1 | Bảng thảo tài liệu Test plan | Nguyễn Hữu Quốc |
| 02 | 08/05/2011 | 0.2 | Cập nhật test plan cho MiniProjet | Huy Nguyen |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

MỤC LỤC

1 GIỚI THIỆU 4

1.1 MỤC ĐÍCH CỦA TÀI LIỆU 4

1.2 PHẠM VI DỰ ÁN 4

1.3 CÁC CÔNG ĐOẠN TEST (TEST PHASES) 4

1.3.1 Unit Testing 4

1.3.2 Integration testing 5

1.3.3 System testing 5

1.3.3.1 Functional testing 5

1.3.3.2 Quality attribute testing 5

1.3.4 Acceptance testing 5

1.4 CÁC KẾT QUẢ TEST 6

1.4.1 Unit testing 6

1.4.2 Integration testing 6

1.4.3 System testing 7

1.4.4 Acceptance testing 7

2 KẾ HOẠCH TEST (SCHEDULE) 8

3 NHÂN SỰ THAM GIA TEST 10

3.1 THÔNG TIN VỀ NHÓM 10

3.2 KẾ HOẠCH HUẤN LUYỆN 10

3.3 VAI TRÒ & TRÁCH NHIỆM 10

4 MÔI TRƯỜNG TEST (ENVIRONMENT) 11

4.1 YÊU CẦU VỀ PHẦN CỨNG (HARDWARE) 11

4.2 YÊU CẦU VỀ PHẦN MỀM (SOFTWARE) 11

4.3 CÔNG CỤ TEST (TEST TOOLS) 11

4.4 CƠ SỞ DỮ LIỆU TEST (TEST DATABASES) 12

4.5 CÁC TÀI LIỆU TEST 12

5 CÁC TIÊU CHÍ BẮT ĐẦU VÀ KẾT THÚC 13

5.1 CÁC TIÊU CHÍ BẮT ĐẦU (ENTRANCE CRITERIA) 13

5.1.1 Integration testing 13

5.1.2 System testing 13

5.1.3 User acceptance testing 13

5.2 CÁC TIÊU CHÍ KẾT THÚC (EXIT CRITERIA) 14

5.2.1 Integration testing 14

5.2.2 System testing 14

5.2.3 User acceptance test 14

6 PHỤ LỤC: MỘT SỐ ĐỊNH NGHĨA 15

DANH SÁCH BẢNG

Bảng 1 Kế hoạch thực hiện 8

Bảng 2 Kế hoạch huấn luyện 10

Bảng 3 Yêu cầu về Hardware 11

Bảng 4 Yêu cầu về Software 11

Bảng 5 Các tài liệu test 12

Bảng 6 Bảng mô tả testcase priority 15

Bảng 7 Bảng mô tả bug 15

# GIỚI THIỆU

## MỤC ĐÍCH CỦA TÀI LIỆU

Tài liệu này mô tả chi tiết các kế hoạch và phương thức thực hiện testing cho dự án MINIPROJECT. Đồng thời các yêu cầu về nhân sự, công cụ, tài liệu cần thiết cho testing cũng được định nghĩa ở đây.

## PHẠM VI DỰ ÁN

Tài liệu chỉ mô tả các strategy & plan khi thực hiện các testing phase trong dự án MINIPROJECT. Các mô tả chi tiết về testcases ở từng phase sẽ được thực hiện ở các tài liệu Test specification tương ứng.

Tài liệu này được sử dụng bởi PM, Tech leader, Test leader tham gia vào dự án MINIPROJECT.

## CÁC CÔNG ĐOẠN TEST (TEST PHASES)

### Unit Testing

* Developer sẽ thực hiện việc phát triển các Unit test scripts và kiểm tra tốt các scripts này trước khi bắt tay vào phát triển các functions chức năng (Unit code).
* Áp dụng triệt để phương thức Test driven development theo thứ tự như sau:
  + Phát triển testcases & test data.
  + Phát triển test script.
  + Thực thi test script vừa phát triển.
  + Phát triển Unit code để chứng minh test script đúng.
  + Kiểm tra test script với Unit code vừa phát triển.
  + Sửa Unit code nếu script chạy không đúng.
  + Khi script chạy đúng tiến hành hiệu chỉnh (refactoring) Unit code.
  + Tiến hành phát triển test script cho Unit code kế tiếp.
* Khi phát triển Unit testcases & test data cần áp dụng các White box testing techniques như:
  + Basic path testing.
  + Branch testing.
  + Loop testing.

### Integration testing

* Các yêu cầu tích hợp các nhóm chức năng (components) được mô tả trong Software Design Specification (SDS) phải được thể hiện đúng trong Integration test specification.
* Các luồng thông tin trao đổi giữa các nhóm chức năng phải được thể hiện đầy đủ trong Integration test specification.
* Khi phát triển integration testcases & test data cần áp dụng các integration test strategies như:
* Big Bang
* Top Down
* Bottom Up
* Sandwich Integration
* MM-Path Based Integration

### System testing

#### Functional testing

* Các usecase mô tả tính năng (functionalities) trong SRS phải được thể hiện đầy đủ trong System test specification
* Các thuộc tính giao diện (GUI) của từng screen trong GUI Specification phải được thể hiện đầy đủ trong System test specification.
* Khi phát triển Functional testcases & test data cần áp dụng các Black box testing techniques như:
* Boundary Testing
* Equivalence Class Testing
* Decision Table
* Error Guess Testing

#### Quality attribute testing

* Các Quality attribute mô tả trong SRS phải được thể hiện đầy đủ trong System test specification
* Khi phát triển quality attibute testcases & test data cần áp dụng các Black box testing techniques như:
* Stress testing
* Performance testing
* Usability testing

### Acceptance testing

* Một bộ testcases được trích từ System testcases và được thực hiện tại môi trường làm việc của user.
* Trong kiểu test này, phần mềm sẽ được thực hiện kiểm tra từ người dùng để tìm ra nếu phần mềm phù hợp với sự mong đợi của người dùng và thực hiện đúng như mong đợi.
* Trong giai đoạn test này, tester có thể cũng thực hiện hoặc khách hàng có các tester của riêng họ để thực hiện.

## CÁC KẾT QUẢ TEST

Tất cả các kết quả test phải được lưu lại trong Configuration management system.

### Unit testing

* Unit code phải đảm bảo sạch và rõ ràng, comment đầy đủ.
* Unit testcases và Unit test reports phải đầy đủ.
* Các test scripts phải đảm bảo các yêu cầu sau:
  + Có thể được thực thi tự động và sử dụng lại nhiều lần.
  + Dễ phát triển.
  + Được phát triển một lần và sẽ được dùng trong tương lai.
  + Có thể được thực thi bởi mọi đối tượng (dễ thực thi).
  + Được thực thi bởi 1 thao tác đơn giản (one push button).
  + Phải chạy nhanh khi được thực thi.

### Integration testing

* Hệ thống đã được Integration testing và đáp ứng đầy đủ các yêu cầu về tích hợp trong SDS.
* Phải đảm bảo các unit test được test đầy đủ và toàn diện
* Integration testcases và test reports phải rõ ràng và đầy đủ. Các tài liệu này sẽ được cập nhật liên tục trong suốt chu kỳ phát triển dự án.
* Bug log report và danh sách các bug còn tồn động.
* Trạng thái của bug và bug lifecycle phải được báo cáo thường xuyên

### System testing

* Các Quality attributestrong SRS phải được test đầy đủ.
* Các chức năng (functionalities) được mô tả trong các usecase ở SRS phải được test đầy đủ.
* System testcases và test reports phải rõ ràng và đầy đủ. Các tài liệu này sẽ được cập nhật liên tục trong suốt chu kỳ phát triển dự án.
* Bug log report và danh sách các bug còn tồn động.
* Trạng thái của bug và bug lifecycle được báo cáo thường xuyên

### Acceptance testing

* Các Acceptance testcases phải được test đầy đủ tại môi trường của user.

# KẾ HOẠCH TEST (SCHEDULE)

Kế hoạch test cho MINIPROJECT sẽ được lập và thực hiện cho Integration testing và System testing. Kế hoạch huấn luyện và thực hiện Unit testing sẽ được mô tả cụ thể trong Master plan.

Bảng 1 Kế hoạch thực hiện

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Mô tả** | **Ngày bắt đầu** | **Ngày kết thúc** | **Nhân sự** |
| **Test documentation** | | | | |
| 1 | Unit Test Specification | 01/06/2011 | 11/07/2011 | Team Member |
| 2 | System Test Specification | 20/06/2011 | 14/07/2011 | Team Member |
| 3 | Test Plan | 04/05/2011 | 14/05/2011 | Tri Nguyen  Hai Nguyen |
|  |  |  |  |  |
| **Unit testing (In-house testing)** | | | | |
|  | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
| **System testing (In-house testing)** | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **UAT (User acceptance testing)** | | | | |
|  | |  |  | |
|  |  |  |  |  |
|  |  |  |  |  |

# NHÂN SỰ THAM GIA TEST

## THÔNG TIN VỀ NHÓM

Nhóm test sẽ gồm 6 người ở giai đoạn thực hiện các test documents. Khi vào giai đoạn Integration & System testing sẽ bổ sung thêm nhân sự là các giảng viên:

* Hieu To
* Nam Vu

## KẾ HOẠCH HUẤN LUYỆN

Bảng 2 Kế hoạch huấn luyện

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Mô tả** | **Ngày bắt đầu** | **Ngày kết thúc** | **Nhân sự** | **Người hướng dẫn** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## VAI TRÒ & TRÁCH NHIỆM

* Test leader: Chịu trách nhiệm hướng dẫn, lên kế hoạch và theo dõi tiến độ thực hiện công việc của nhóm test. Trực tiếp báo cáo cho PM.
* Tester: Thực hiện việc test hệ thống theo các tài liệu testing và trao đổi với các developer về việc sửa bugs. Báo cáo trực tiếp cho Test leader.
* Document writer: Chịu trách nhiệm phát triển các tài liệu test (Test plan, testcases, test report).

# MÔI TRƯỜNG TEST (ENVIRONMENT)

## YÊU CẦU VỀ PHẦN CỨNG (HARDWARE)

Bảng 3 Yêu cầu về Hardware

|  |  |  |
| --- | --- | --- |
| **STT** | **Mô tả** | **Cấu hình tối thiểu** |
| 1 | Application server | - CPU:  Core 2 Duo 2.4 (máy server chuyên dụng)  - RAM 2.0GB  - HardDisk: 10GB trống |
| 2 | Work station | Hỗ trợ javascript |

## YÊU CẦU VỀ PHẦN MỀM (SOFTWARE)

Bảng 4 Yêu cầu về Software

|  |  |  |
| --- | --- | --- |
| **STT** | **Mô tả** | **Software** |
| 1 | Web Server / Application server | .Net framework 4.0  Visual Studio 2010  SQL server 2008  Oledb for FoxPro |

## CÔNG CỤ TEST (TEST TOOLS)

Dự kiến sẽ sử dụng Visual studio 2010 để thực hiện unit test, system test scripts.

Sử dụng TFS (Team foundation server) 2010 để quản lý kết quả test và fix bugs.

## CƠ SỞ DỮ LIỆU TEST (TEST DATABASES)

## Tất cả các kết quả test phải được lưu lại trong Configuration management system.

## CÁC TÀI LIỆU TEST

Bảng 5 Các tài liệu test

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên tài liệu** | **Mô tả** |
| 1 | MINIPROJECT Test Plan | Tài liệu mô tả Test strategy & plan khi thực hiện testing dự án MINIPROJECT |
| 2 | MINIPROJECT System Test Specification | Tài liệu mô tả các testcases cho System test phase khi thực hiện testing dự án MINIPROJECT. |
| 3 | MINIPROJECT UnitTest Specification | Tài liệu mô tả các UT scenarios cho từng Unit trong MINIPROJECT. |

# CÁC TIÊU CHÍ BẮT ĐẦU VÀ KẾT THÚC

## CÁC TIÊU CHÍ BẮT ĐẦU (ENTRANCE CRITERIA)

Các tiêu chí này cần được định nghĩa trước khi việc test bắt đầu và được thực hiện bởi Project manager, Technical leader, Test leader.

### Unit testing

* UT specification phải được thực hiện đầy đủ.
* Các UnitTest scripts phải được xây dựng dựa vào UnitTest specification.
* Các production code phải được phát triển theo phương pháp TDD.
* Các production code phải được review.

### Integration testing

* Integration test plan phải được duyệt.
* Integrationtest cases phải sẵn sàng.
* Các unit test phải được pass hoàn toàn
* Nhân sự phải sẵn sàng.
* Môi trường và công cụ test phải sẵn sàng.
* Các tài liệu test documents phải được review

### System testing

* System test plan phải được duyệt.
* System test cases phải sẵn sàng.
* Nhân sự phải sẵn sàng.
* Môi trường và công cụ test phải sẵn sàng.
* Các tài liệu test documents phải được review

### User acceptance testing

* User acceptance test plan và test case phải được duyệt và sẵn sàng.
* Phải có danh sách những issues và bugs chưa xử lý.
* Nhân sự theo kế hoạch phải sẵn sàng.
* Môi trường và công cụ test phải sẵn sàng.
* Các tài liệu test documents phải được review

## CÁC TIÊU CHÍ KẾT THÚC (EXIT CRITERIA)

### Unit testing

* Tất cả các UnitTestcase phải được thực hiện.
* Tất cả các bugs phát hiện phải được sửa và các retest phải được hoàn tất.
* Không có lỗi vớiSeverity1, 2
* Danh sách các issues và bugs còn lại.

### Integration testing

* Tất cả các Integrationtestcases phải được thực hiện.
* Tất cả các bugs phát hiện phải được sửa và các retest phải được hoàn tất.
* Không có lỗi vớiSeverity1, 2
* Danh sách các issues và bugs còn lại.

### System testing

* Tất cả các System test cases phải được thực hiện.
* Không có lỗi với Severity 1, 2
* Danh sách các issues và bugs còn lại.

### User acceptance test

* Tất cả các UAT test cases phải được thực hiện.
* Biên bản nghiệm thu từ khách hàng.
* Báo cáo về những issues.

# PHỤ LỤC: MỘT SỐ ĐỊNH NGHĨA

Bảng 6 Bảng mô tả testcase priority

|  |  |
| --- | --- |
| **Priority** | **Mô tả** |
| 1 | Testcase rất quan trọng và không thể fail được. Phải fix bug ngay nếu testcase fail |
| 2 | Testcase quan trọng và không được fail. Phải fix bug sớm nếu testcase fail |
| 3 | Testcase không quan trọng và có thể được thực hiện hoặc không tùy vào kế hoạch test. |
| 4 | Testcase không quan trọng và không cần thiết phải test nhưng khi thực hiện sẽ giúp nâng cao chất lượng chương trình. |

Trong quá trình test, các bugs sẽ được quản lý bởi các công cụ như Excel, Mantis. Các bug sẽ được phân loại và gán cho developer để fix.

Bảng 7 Bảng mô tả bug

|  |  |
| --- | --- |
| **Bug severity** | **Mô tả** |
| 1 | Bug rất nghiêm trọng phải fix ngay vì làm system treo / thoát hoặc mất data. Bug do testcase Priority 1 fail. |
| 2 | Bug quan trọng phải fix sớm vì có thể gây nên những lổi nghiêm trọng như làm sai tính năng và xử lý của system hoặc làm system treo trong một số trường hợp. Bug do testcase Priority 2 fail. |
| 3 | Bug nhỏ cần fix ASAP vì có thể gây ra các lổi về cách hành xử của system và gây sự khó khăn cho user. Bug do testcase Priority 3 fail |
| 4 | Tính năng nâng cao nên làm và có thể do testcase Priority 1 fail |
| **Bug status** | **Mô tả** |
| New | Bug mới |
| Feedback | Cần thêm thông tin về bug |
| Acknowledged | Bug đã được chú ý nhưng chưa confirmed hay assigned |
| Confirmed | Đã xác nhận và có thể tái tạo lại bug |
| Assigned | Đã gán cho developer |
| Resolved | Bug đã được fix và đang chờ xác nhận |
| Closed | Đóng bug |