Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/05/2010 | 1.0 | Soạn testplan | Vũ Thị Ngọc Nhi |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Intended Audience 3

1.4 Document Terminology and Acronyms 3

1.5 References 3

1.6 Document Structure 3

2. Evaluation Mission and Test Motivation 3

2.1 Background 3

2.2 Evaluation Mission 3

2.3 Test Motivators 3

3. Target Test Items 3

4. Outline of Planned Tests 3

4.1 Outline of Test Inclusions 3

4.2 Outline of other candidates for potential inclusion 3

4.3 Outline of Test Exclusions 3

5. Test Approach 3

5.1 Initial Test-Idea Catalogs and other reference sources 3

5.2 Testing Techniques and Types 3

5.2.1 Data and Database Integrity Testing 3

5.2.2 Function Testing 3

5.2.3 Business Cycle Testing 3

5.2.4 User Interface Testing 3

5.2.5 Performance Profiling 3

5.2.6 Load Testing 3

5.2.7 Stress Testing 3

5.2.8 Volume Testing 3

5.2.9 Security and Access Control Testing 3

5.2.10 Failover and Recovery Testing 3

5.2.11 Configuration Testing 3

5.2.12 Installation Testing 3

6. Entry and Exit Criteria 3

6.1 Test Plan 3

6.1.1 Test Plan Entry Criteria 3

6.1.2 Test Plan Exit Criteria 3

6.1.3 Suspension and resumption criteria 3

6.2 Test Cycles 3

6.2.1 Test Cycle Entry Criteria 3

6.2.2 Test Cycle Exit Criteria 3

6.2.3 Test Cycle abnormal termination 3

7. Deliverables 3

7.1 Test Evaluation Summaries 3

7.2 Reporting on Test Coverage 3

7.3 Perceived Quality Reports 3

7.4 Incident Logs and Change Requests 3

7.5 Smoke Test Suite and supporting Test Scripts 3

7.6 Additional work products 3

7.6.1 Detailed Test Results 3

7.6.2 Additional automated functional Test Scripts 3

7.6.3 Test Guidelines 3

7.6.4 Traceability Matrices 3

8. Testing Workflow 3

9. Environmental Needs 3

9.1 Base System Hardware 3

9.2 Base Software Elements in the Test Environment 3

9.3 Productivity and Support Tools 3

9.4 Test Environment Configurations 3

10. Responsibilities, Staffing and Training Needs 3

10.1 People and Roles 3

10.2 Staffing and Training Needs 3

11. Iteration Milestones 3

12. Risks, Dependencies, Assumptions and Constraints 3

13. Management Process and Procedures 3

13.1 Measuring and Assessing the Extent of Testing 3

13.2 Assessing the deliverables of this Test Plan 3

13.3 Problem Reporting, Escalation and Issue Resolution 3

13.4 Managing Test Cycles 3

13.5 Traceability Strategies 3

13.6 Approval and Signoff 3

# Introduction

## Purpose

Tài liệu này được viết nhằm mục đích đưa ra một kế hoạch và phạm vi kiểm tra ứng dụng web và thẩm định lại hệ thống so với yêu cầu cảu khách hàng đảm bảo hệ thống giao cho khách hàng sẽ đúng với yêu cầu khách hàng và ít lỗi nhất có thể.

Mục tiêu cần phải đạt được là áp dụng tất cả các kỹ thuật của kiểm chứng phần mềm để test trang web Advising System nhằm đảm bảo các yêu cầu tối thiểu về các chức năng của ứng dụng do người dùng yêu cầu và các yêu cầu cần thiết đối với một ứng dụng Web.

## Scope

* Kiểm tra các chức năng. (Function Testing).
* Kiểm tra giao diện. (GUI Testing).
* Kiểm tra tính tiện dụng (Usability Testing).
* Kiểm tra hiệu suất ( Performance Testing).
* Kiểm tra tính tương thích (Compatibility Testing).
* Kiểm tra tính bảo mật (Security Testing)/

## Intended Audience

Tài liệu này được áp dụng cho nhóm thiết kế test và thực hiện kiểm tra ứng dụng web Advising System. Tài liệu này sẽ được chuyển giao cho khách hàng nếu khách hàng yêu cầu.

## Document Terminology and Acronyms

Testcase: các trường hợp test.

Usecase: các trường hợp sử dụng của người dùng.

## References

Danh sách cách tài liệu tham khảo.

## Document Structure

[This subsection outlines what the rest of the **Test Plan** contains and gives an introduction to how the rest of the document is organized. This section may be eliminated if a Table of Contents is used.]

Cấu trúc tài liệu

# Evaluation Mission and Test Motivation

## Background

* Với tài liệu này có thể giúp chúng ta theo dõi sự tiến triên của test nhằm đo đạc , theo dõi và dễ dàng phát hiện được vấn đề khi có lỗi xảy ra.
* Việc thực test nhằm đảm bảo chương trình luôn thực thi tốt trong mọi trường hợp và mọi tình huống có thể xảy ra, nhằm làm tăng sự hài lòng của khác hàng đối với chương trình của chúng ta.

## Evaluation Mission

* Đưa ra các testcase để thực hiện việc test chương trình nhằm phát hiện ra những lỗi hàm ẩn trong chương trình. Trong quá trình thiết kế test case cần tránh những lỗi do chủ quan. Đối với một số kiểu test đặt biệt thì chung ta phải đưa ra những chuẩn rõ ràng để test.
* Trong quá trình test chúng ta nên test theo kiểu hồi quy để đảm bảo phiên bản phần mêm mới thực hiện tốt các chức năng như phiên bản phần mềm cũ và sự thay đổi không gây ra những lỗi mới trên những chức năng vốn có và đã thực hiện tốt.

## Test Motivators

* Mong muốn chương trình có chất lượng cao nhất thỏa mãn yêu cầu của khách hàng.
* Do có những hạn chế về kỹ thuật và môi trường cài đặt , và những rủi ro trong quá trình cài đặt cũng như sử dụng, có thể xãy ra những lỗi ngoài ý muốn.
* Có thể xảy ra những rủi ro với tài nguyên , sức chứa cơ sở dữ liệu, thời gian truy cập cơ sở dữ liệu, cũng như khả năng chiu đựng của chương trình.
* Mong muốn biết được những khuyết điểm của chương trình , để có biện pháp khắc phục, nhằm tạo ra chương trình có chất lượng tốt nhất.
* Việc phát hiện ra lỗi càng sớm , thì chúng ta càng ít tốn thời gian và chi phi để khắc phục những hậu quả do lỗi đó gây ra.

# Target Test Items

Viêc thực hện test này sẽ được dùng để test các phần sau:

* Kiểm tra toàn bộ dữ liệu và cơ sở dữ liệu
* Kiểm tra chức năng.
* Kiểm tra business Cycle.
* Kiểm tra giao diện.
* Kiểm tra khả năng vận hành
* Kiểm tra khả năng chịu tải(load test và stress test,volume test)
* Kiểm tra khả năng bảo mật
* Kiểm tra khả năng phục hồi
* Kiểm tra cấu hình.

# Outline of Planned Tests

## Outline of Test Inclusions

Trong giai đoạn 2 này, thì việc test được tập trung vào viết test case và testcho các các phần sau:

* Kiểm tra toàn bộ dữ liệu và cơ sở dữ liệu
* Kiểm tra chức năng
* Kiêm tra giao diện
* Kiểm tra hiệu suất.

## Outline of Other Candidates for Potential Inclusion

Và có những phần test sẽ được làm trong giai đoạn tới:

* Kiểm tra khả năng chịu đựng
* Kiểm tra tính bảo mật
* Kiểm tra khả năng phục hồi
* Kiểm tra cấu hình.

## Outline of Test Exclusions

Phần test không được thực hiện tới đó là:

* Installation Testing.
* Kiểm tra mức đơn vị(Nunit Test): thường do người lập trình thực hiện, được thực hiện trong giai đoạn viết code và xuyên suốt chu kì phát triển phần mềm.
* Kiểm tra tích hợp(Interation Test): kết hợp các thành phần của một ứng dụng và kiểm tra như một ứng dụng cụ thể. Nó đã được người lập trình test trong quá trình viết code
* Kiểm tra chấp nhận sản phẩm(acceptance Test): chúng ta chỉ test trên chương trình được xây dựng lên , mà không test dựa vào quá trình sử dụng của khach hàng và phát hiện lỗi.

# Test Approach

[The Test Approach presents the recommended strategy for designing and implementing the required tests. Sections 3, Target Test Items, and 4, Outline of Planned Tests, identified **what** items will be tested and **what** types of tests would be performed. This section describes **how** the tests will be realized.

One aspect to consider for the test approach is the techniques to be used. This should include an outline of how each technique can be implemented, both from a manual and/or an automated perspective, and the criterion for knowing that the technique is useful and successful. For each technique, provide a description of the technique and define why it is an important part of the test approach by briefly outlining how it helps achieve the Evaluation Mission or addresses the Test Motivators.

Another aspect to discuss in this section is the Fault or Failure models that are applicable and ways to approach evaluating them.

As you define each aspect of the approach, you should update Section 10, Responsibilities, Staffing, and Training Needs, to document the test environment configuration and other resources that will be needed to implement each aspect.]

## Initial Test-Idea Catalogs and Other Reference Sources

[Provide a listing of existing resources that will be referenced to stimulate the identification and selection of specific tests to be conducted. An example Test-Ideas Catalog is provided in the examples section of RUP.]

## Testing Techniques and Types

### Database Testing

|  |  |
| --- | --- |
| Technique Objective: | [Exercise database access methods and processes independent of the UI so you can observe and log incorrect functioning target behavior or data corruption.] |
| Technique: | * Phân tích giá trị giới hạn: là kỹ thuật chọn lựa dữ liệu nằm ở các điểm tới hạn của miền dữ liệu. Các giá trị tới hạn có thể là: giá trị lớn nhất, nhỏ nhất, cận trong, cận ngoài, các giá trị lỗi. Nguyên tắc: nếu hệ thống có thể hoạt động tốt với những giá trị tới hạn thì hệ thống cũng sẽ hoạt động tốt với các giá trị bên trong của các điểm tới hạn đó. Ứng với mỗi điểm tới hạn sẽ cho ta một testcase. * Xác định các mốc giới hạn dựa vào kỹ thuật phân vùng tương đương: chia miền giá trị thành các vùng hợp lệ, vùng không hợp lệ và vùng đặc biệt. * Với 2 vùng liền kế nhau ta sẽ chọn ra các giá trị biên. Mỗi giá trị biên sẽ cho ta một testcase và một giá trị test data. * Ngoài ra, khi xét các điểm tới hạn chúng ta cũng cần quan tâm đến các điểm giới hạn tự nhiên của kiểu dữ liệu dùng trong chương trình. Đó là các điểm không được dùng làm mốc phân định vùng tương đương nhưng dễ gây ra lỗi cho chương trình. * Giữa các miền có thể có điểm giới hạn ẩn. Khi đó chúng ta phải dự đoán các trường hợp có thể nguy cơ xảy ra lỗi để viết Testcase và tạo TestData. * Các bước thực hiện:   . Xác định số lượng bảng có trong dự án.  . Phân định bảng nào cần tạo dữ liệu bắt buộc, bảng nào chỉ cần tạo dữ liệu thông thường, bảng nào cần tạo cả hai.  . Xây dựng dữ liệu bắt buộc và dữ liệu thông thường: phân vùng dữ liệu, xác định các giá trị của giới hạn để tạo trường dữ liệu. |
| Oracles: | * Không có. |
| Required Tools: | * Không có. |
| Success Criteria: | * Việc test hoàn thành chỉ khi tất cả các testcase đều đã được thực hiện và có kết quả. * Các lỗi tìm thấy đều được ghi nhận vào một giải pháp cụ thể. * Các lỗi mở ra (nếu có) đều phải được ghi nhận và giải quyết. |
| Special Considerations: |  |

### Function Testing

|  |  |
| --- | --- |
| Technique Objective: | * Kiểm tra các chức năng của người dung với vai trò khách, sinh viên và quản trị. * Kiểm tra tất cả các liên kết trong trang web. * Kết nối cơ sở dữ liệu. * Các biểu mẫu được sử dụng trong trang web để đưa thông tin lên trang web hoặc lấy thông tin về. |
| Technique: | * Với mỗi chu trình sự kiện của mỗi usecase, sẽ xác định một tập các giao dịch đại diện cho mỗi hành động của tác nhân khi thực hiện usecase. * Tối thiểu phải có 2 testcase cho mỗi giao dịch: một testcase cho kết quả hợp lệ, và một testcase cho kết quả không hợp lệ. * Với kiểm tra sự liên kết các link, phải đưa ra bảng tất cả các trạng thái để xác định trạng thái liên kết giữa các link trong trang web, cũng như ngoài trang web. |
| Oracles: | * Không có chuẩn. |
| Required Tools: | * Hiện tại thì chưa có công cụ. |
| Success Criteria: | * Tiêu chuẩn để xác định việc test hoàn thành là tất cả các testcase đã được thực hiện. * Tất cả các lỗi được xác định phải được ghi nhận, mô tả, và có một giải pháp thỏa thuận. * Không có lỗi mới nào phát hiện. |
| Special Considerations: |  |

### User Interface Testing

|  |  |
| --- | --- |
| Technique Objective: | * Kiểm tra tính đúng đắn. * Tính thẩm mĩ. * Tính tiện dụng. |
| Technique: | * Với mỗi màn hình phải đưa ra các check list cho từng control để việc kiểm tra được dễ dàng. * Với mỗi tính cần kiểm tra: đưa ra danh sách các điều kiện kiểm tra. |
| Oracles: | * Không có chuẩn. |
| Required Tools: | * Không có công cụ. |
| Success Criteria: | * Việc test hoàn thành chỉ khi tất cả các testcase đều đã được thực hiện và có kết quả. * Các lỗi tìm thấy đều được ghi nhận vào một giải pháp cụ thể. * Các lỗi mở ra (nếu có) đều phải được ghi nhận và giải quyết. |
| Special Considerations: |  |

### Performance Profiling

|  |  |
| --- | --- |
| Technique Objective: | [Exercise behaviors for designated functional transactions or business functions under the following conditions to observe and log target behavior and application performance data:  • normal anticipated workload  • anticipated worst-case workload] |
| Technique: | * Với mỗi Testcase , xác định một tập các giao dịch. * Đưa ra kịch bản test (testscript) dung để thực hiện mỗi giao dịch: đo thời gian phản hồi, * Đánh giá kết quả test thông qua biểu đồ trạng thái để xác định kết quả có như mong đợi hay không. |
| Oracles: | * Không có. |
| Required Tools: | * Chưa xác định. |
| Success Criteria: | * Việc thực hiện test hoàn thành khi các kịch bản test được thực hiện và đánh giá qua biểu đồ. |
| Special Considerations: | * Việc test phải lập lịch vào những giờ không còn các giao dịch khác trên mạng. |

### Load Testing

[Load testing is a performance test that subjects the target-of-test to varying workloads to measure and evaluate the performance behaviors and abilities of the target-of-test to continue to function properly under these different workloads. The goal of load testing is to determine and ensure that the system functions properly beyond the expected maximum workload. Additionally, load testing evaluates the performance characteristics, such as response times, transaction rates, and other time-sensitive issues).]

[**Note**: Transactions in the following table refer to “logical business transactions”. These transactions are defined as specific functions that an end user of the system is expected to perform using the application, such as add or modify a given contract.]

|  |  |
| --- | --- |
| Technique Objective: | [Exercise designated transactions or business cases under varying workload conditions to observe and log target behavior and system performance data.] |
| Technique: | • [Use Transaction Test Scripts developed for Function or Business Cycle Testing as a basis, but remember to remove unnecessary interactions and delays.  • Modify data files to increase the number of transactions or the tests to increase the number of times each transaction occurs.  • Workloads should include (for example, Daily, Weekly, Monthly and so forth) Peak loads.  • Workloads should represent both Average as well as Peak loads.  • Workloads should represent both Instantaneous and Sustained Peaks.  • The Workloads should be executed under different Test Environment Configurations.] |
| Oracles: | [Outline one or more strategies that can be used by the technique to accurately observe the outcomes of the test. The oracle combines elements of both the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, oracles will be self-verifying, allowing automated tests to make an initial assessment of test pass or failure, however, be careful to mitigate the risks inherent in automated results determination.] |
| Required Tools: | [The technique requires the following tools:   * Test Script Automation Tool * Transaction Load Scheduling and control tool * installation-monitoring tools (registry, hard disk, CPU, memory, and so on) * resource-constraining tools (for example, Canned Heat) * Data-generation tools] |
| Success Criteria: | [The technique supports the testing of Workload Emulation, which is the successful emulation of the workload without any failures due to test implementation problems.] |
| Special Considerations: | • [Load testing should be performed on a dedicated machine or at a dedicated time. This permits full control and accurate measurement.  • The databases used for load testing should be either actual size or scaled equally.] |

### Stress Testing

[Stress testing is a type of performance test implemented and executed to understand how a system fails due to conditions at the boundary, or outside of, the expected tolerances. This typically involves low resources or competition for resources. Low resource conditions reveal how the target-of-test fails that is not apparent under normal conditions. Other defects might result from competition for shared resources, like database locks or network bandwidth, although some of these tests are usually addressed under functional and load testing.]

[**Note**: References to transactions in the following table refer to logical business transactions.]

|  |  |
| --- | --- |
| Technique Objective: | [Exercise the target-of-test functions under the following stress conditions to observe and log target behavior that identifies and documents the conditions under which the system **fails** to continue functioning properly  • little or no memory available on the server (RAM and persistent storage space)  • maximum actual or physically capable number of clients connected or simulated  • multiple users performing the same transactions against the same data or accounts  • “overload” transaction volume or mix (see Performance Profiling above)] |
| Technique: | • [Use tests developed for Performance Profiling or Load Testing.  • To test limited resources, tests should be run on a single machine, and RAM and persistent storage space on the server should be reduced or limited.  • For remaining stress tests, multiple clients should be used, either running the same tests or complementary tests to produce the worst-case transaction volume or mix. |
| Oracles: | [Outline one or more strategies that can be used by the technique to accurately observe the outcomes of the test. The oracle combines elements of both the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, oracles will be self-verifying, allowing automated tests to make an initial assessment of test pass or failure, however, be careful to mitigate the risks inherent in automated results determination.] |
| Required Tools: | [The technique requires the following tools:   * Test Script Automation Tool * Transaction Load Scheduling and control tool * installation-monitoring tools (registry, hard disk, CPU, memory, and so on) * resource-constraining tools (for example, Canned Heat) * Data-generation tools] |
| Success Criteria: | The technique supports the testing of Stress Emulation. The system can be emulated successfully in one or more conditions defined as stress conditions and an observation of the resulting system state during and after the condition has been emulated can be captured.] |
| Special Considerations: | • [Stressing the network may require network tools to load the network with messages or packets.  • The persistent storage used for the system should temporarily be reduced to restrict the available space for the database to grow.  • Synchronize the simultaneous clients accessing of the same records or data accounts.] |

### Volume Testing

[Volume testing subjects the target-of-test to large amounts of data to determine if limits are reached that cause the software to fail. Volume testing also identifies the continuous maximum load or volume the target-of-test can handle for a given period. For example, if the target-of-test is processing a set of database records to generate a report, a Volume Test would use a large test database, and would check that the software behaved normally and produced the correct report.]

|  |  |
| --- | --- |
| Technique Objective: | [Exercise the target-of-test under the following high volume scenarios to observe and log target behavior:  • Maximum (actual or physically-capable) number of clients connected, or simulated, all performing the same, worst case (performance) business function for an extended period.  • Maximum database size has been reached (actual or scaled) and multiple queries or report transactions are executed simultaneously.] |
| Technique: | • [Use tests developed for Performance Profiling or Load Testing.  • Multiple clients should be used, either running the same tests or complementary tests to produce the worst-case transaction volume or mix (see Stress Testing) for an extended period.  • Maximum database size is created (actual, scaled, or filled with representative data) and multiple clients are used to run queries and report transactions simultaneously for extended periods.] |
| Oracles: | [Outline one or more strategies that can be used by the technique to accurately observe the outcomes of the test. The oracle combines elements of both the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, oracles will be self-verifying, allowing automated tests to make an initial assessment of test pass or failure, however, be careful to mitigate the risks inherent in automated results determination.] |
| Required Tools: | [The technique requires the following tools:   * Test Script Automation Tool * Transaction Load Scheduling and control tool * installation-monitoring tools (registry, hard disk, CPU, memory, and so on) * resource-constraining tools (for example, Canned Heat) * Data-generation tools] |
| Success Criteria: | [The technique supports the testing of Volume Emulation. Large quantities of users, data, transactions, or other aspects of the system use under volume can be successfully emulated and an observation of the system state changes over the duration of the volume test can be captured.] |
| Special Considerations: | [What period of time would be considered an acceptable time for high volume conditions, as noted above?] |

### Security and Access Control Testing

[Security and Access Control Testing focuses on two key areas of security:

• Application-level security, including access to the Data or Business Functions

• System-level Security, including logging into or remotely accessing to the system.

Based on the security you want, application-level security ensures that actors are restricted to specific functions or use cases, or they are limited in the data that is available to them. For example, everyone may be permitted to enter data and create new accounts, but only managers can delete them. If there is security at the data level, testing ensures that “user type one” can see all customer information, including financial data, however, “user two” only sees the demographic data for the same client.

System-level security ensures that only those users granted access to the system are capable of accessing the applications and only through the appropriate gateways.]

|  |  |
| --- | --- |
| Technique Objective: | [Exercise the target-of-test under the following conditions to observe and log target behavior:   * Application-level Security: an actor can access only those functions or data for which their user type is provided permissions. * System-level Security: only those actors with access to the system and applications are permitted to access them. |
| Technique: | * [Application-level Security: Identify and list each user type and the functions or data each type has permissions for.]   + Create tests for each user type and verify each permission by creating transactions specific to each user type.   + Modify user type and re-run tests for same users. In each case, verify those additional functions or data are correctly available or denied. * System-level Access: [See Special Considerations below] |
| Oracles: | [Outline one or more strategies that can be used by the technique to accurately observe the outcomes of the test. The oracle combines elements of both the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, oracles will be self-verifying, allowing automated tests to make an initial assessment of test pass or failure, however, be careful to mitigate the risks inherent in automated results determination.] |
| Required Tools: | [The technique requires the following tools:   * Test Script Automation Tool * “Hacker” security breach and probing tools * OS Security Admin Tools] |
| Success Criteria: | [The technique supports the testing of for each known actor type the appropriate functions or data affected by security settings can be tested.] |
| Special Considerations: | [Access to the system must be reviewed or discussed with the appropriate network or systems administrator. This testing may not be required as it may be a function of network or systems administration.] |

### Failover and Recovery Testing

[Failover and recovery testing ensures that the target-of-test can successfully failover and recover from a variety of hardware, software or network malfunctions with undue loss of data or data integrity.

For those systems that must be kept running failover testing ensures that, when a failover condition occurs, the alternate or backup systems properly “take over” for the failed system without any loss of data or transactions.

Recovery testing is an antagonistic test process in which the application or system is exposed to extreme conditions, or simulated conditions, to cause a failure, such as device Input/Output (I/O) failures, or invalid database pointers and keys. Recovery processes are invoked, and the application or system is monitored and inspected to verify proper application, or system, and data recovery has been achieved.]

|  |  |
| --- | --- |
| Technique Objective: | [Simulate the failure conditions and exercise the recovery processes (manual and automated) to restore the database, applications, and system to a desired, known, state. The following types of conditions are included in the testing to observe and log target behavior after recovery:  • power interruption to the client  • power interruption to the server  • communication interruption via network servers  • interruption, communication, or power loss to DASD (Dynamic Access Storage Devices) and DASD controllers  • incomplete cycles (data filter processes interrupted, data synchronization processes interrupted)  • invalid database pointers or keys  • invalid or corrupted data elements in database] |
| Technique: | [The tests already created for Function and Business Cycle testing can be used as a basis for creating a series of transactions to support failover and recovery testing, primarily to define the tests to be run to test that recovery was successful.  • Power interruption to the client: power the PC down.  • Power interruption to the server: simulate or initiate power down procedures for the server.  • Interruption via network servers: simulate or initiate communication loss with the network (physically disconnect communication wires or power down network servers or routers).  • Interruption, communication, or power loss to DASD and DASD controllers: simulate or physically eliminate communication with one or more DASDs or controllers.  Once the above conditions or simulated conditions are achieved, additional transactions should be executed and, upon reaching this second test point state, recovery procedures should be invoked.  Testing for incomplete cycles uses the same technique as described above except that the database processes themselves should be aborted or prematurely terminated.  Testing for the following conditions requires that a known database state be achieved.  Several database fields, pointers, and keys should be corrupted manually and directly within the database (via database tools). Additional transactions should be executed using the tests from Application Function and Business Cycle Testing and full cycles executed.] |
| Oracles: | [Outline one or more strategies that can be used by the technique to accurately observe the outcomes of the test. The oracle combines elements of both the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, oracles will be self-verifying, allowing automated tests to make an initial assessment of test pass or failure, however, be careful to mitigate the risks inherent in automated results determination.] |
| Required Tools: | [The technique requires the following tools:   * base configuration imager and restorer * installation monitoring tools (registry, hard disk, CPU, memory, and so on) * backup and recovery tools] |
| Success Criteria: | The technique supports the testing of:   * One or more simulated disasters involving one or more combinations of the application, database, and system. * One or more simulated recoveries involving one or more combinations of the application, database, and system to a known desired state.] |
| Special Considerations: | * [Recovery testing is highly intrusive. Procedures to disconnect cabling (simulating power or communication loss) may not be desirable or feasible. Alternative methods, such as diagnostic software tools may be required. * Resources from the Systems (or Computer Operations), Database, and Networking groups are required. * These tests should be run after hours or on an isolated machine.] |

### Configuration Testing

[Configuration testing verifies the operation of the target-of-test on different software and hardware configurations. In most production environments, the particular hardware specifications for the client workstations, network connections, and database servers vary. Client workstations may have different software loaded⎯for example, applications, drivers, and so on⎯and, at any one time, many different combinations may be active using different resources.]

|  |  |
| --- | --- |
| Technique Objective: | [Exercise the target-of-test on the required hardware and software configurations to observe and log target behavior under different configurations and identify changes in configuration state.] |
| Technique: | * [Use Function Test scripts. * Open and close various non-target-of-test related software, such as Microsoft Excel and Word applications, either as part of the test or prior to the start of the test. * Execute selected transactions to simulate actors interacting with the target-of-test and the non-target-of-test software. * Repeat the above process, minimizing the available conventional memory on the client workstation.] |
| Oracles: | [Outline one or more strategies that can be used by the technique to accurately observe the outcomes of the test. The oracle combines elements of both the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, oracles will be self-verifying, allowing automated tests to make an initial assessment of test pass or failure, however, be careful to mitigate the risks inherent in automated results determination.] |
| Required Tools: | [The technique requires the following tools:   * base configuration imager and restore * installation monitoring tools (registry, hard disk, CPU, memory, and so on)] |
| Success Criteria: | [The technique supports the testing of one or more combinations of the target test items running in expected, supported deployment environments.] |
| Special Considerations: | * [What non-target-of-test software is needed, is available, and what is accessible on the desktop? * What applications are typically used? * What data are the applications running; for example, a large spreadsheet opened in Excel or a 100-page document in Word? * The entire system’s netware, network servers, databases, and so on, also needs to be documented as part of this test.] |

### Installation Testing

Hệ thống web nên không cần phải cài đặt.

# Entry and Exit Criteria

## Test Plan

### Test Plan Entry Criteria

* Được thực thi khi đã thu thập được một phần của yêu cầu và đã bắt đầu vào thiết kế hệ thống , giao diện và các thiết kế liên quan.

### Test Plan Exit Criteria

* Được kết thúc khi chuyển giao phần mềm cho khách hàng sử dụng.

### Suspension and Resumption Criteria

* Xảy ra khi trong quá trình test , phát hiện có quá nhiều test case sai, hoặc test case không phù hợp với những mong muốn test của tester. Hoặc các test case không phủ hết tất cả các trường hợp xảy ra của phần mềm, tạo nên những lỗi tiềm ẩn.

## Test Cycles

### Test Cycle Entry Criteria

* Việc thực hiện test đươc thực hiện theo một chu kỳ lặp, khi qua một pharse mới thì Test plan lại được chỉnh sửa thêm dựa vào những thay đổi mới của phần yêu cầu và thiết kế vừa được cập nhật và thực thi việc test chương trình.

### Test Cycle Exit Criteria

* Việc kết thúc được xác định khi tất cả các phần cần được test với các test case đã được test xong, và chuẩn bị sang một pharse mới.

### Test Cycle Abnormal Termination

* Khi trong test case có nhiều lỗi, không đãm bảo được việc test chương trình sẽ cho ra kết quả chính xác, thì việc test sẽ được dừng lai , và các test case sẽ được thay đổi.

# Deliverables

[In this section, list the various artifacts that will be created by the test effort that are useful deliverables to the various stakeholders of the test effort. Don’t list all work products; only list those that give direct, tangible benefit to a stakeholder and those by which you want the success of the test effort to be measured.]

## Test Evaluation Summaries

[Provide a brief outline of both the form and content of the test evaluation summaries, and indicate how frequently they will be produced.]

## Reporting on Test Coverage

[Provide a brief outline of both the form and content of the reports used to measure the extent of testing, and indicate how frequently they will be produced. Give an indication as to the method and tools used to record, measure, and report on the extent of testing.]

## Perceived Quality Reports

[Provide a brief outline of both the form and content of the reports used to measure the perceived quality of the product, and indicate how frequently they will be produced. Give an indication about to the method and tools used to record, measure, and report on the perceived product quality. You might include some analysis of Incidents and Change Request over Test Coverage.]

## Incident Logs and Change Requests

[Provide a brief outline of both the method and tools used to record, track, and manage test incidents, associated change requests, and their status.]

## Smoke Test Suite and Supporting Test Scripts

[Provide a brief outline of the test assets that will be delivered to allow ongoing regression testing of subsequent product builds to help detect regressions in the product quality.]

## Additional Work Products

[In this section, identify the work products that are optional deliverables or those that should not be used to measure or assess the successful execution of the **Test Plan**.]

### Detailed Test Results

[This denotes either a collection of Microsoft Excel spreadsheets listing the results determined for each test case, or the repository of both test logs and determined results maintained by a specialized test product.]

### Additional Automated Functional Test Scripts

[These will be either a collection of the source code files for automated test scripts, or the repository of both source code and compiled executables for test scripts maintained by the test automation product.]

### Test Guidelines

[Test Guidelines cover a broad set of categories, including Test-Idea catalogs, Good Practice Guidance, Test patterns, Fault and Failure Models, Automation Design Standards, and so forth.]

### Traceability Matrices

[Using a tool such as Rational RequisistePro or MS Excel, provide one or more matrices of traceability relationships between traced items.]

# Testing Workflow

[Provide an outline of the workflow to be followed by the Test team in the development and execution of this **Test Plan**.]

The specific testing workflow that you will use should be documented separately in the project's Development Case. It should explain how the project has customized the base RUP test workflow (typically on a phase-by-phase basis). In most cases, we recommend you place a reference in this section of the **Test Plan** to the relevant section of the Development Case. It might be both useful and sufficient to simply include a diagram or image depicting your test workflow.

More specific details of the individual testing tasks are defined in a number of different ways, depending on project culture; for example:

* defined as a list of tasks in this section of the **Test Plan**, or in an accompanying appendix
* defined in a central project schedule (often in a scheduling tool such as Microsoft Project)
* documented in individual, "dynamic" to-do lists for each team member, which are usually too detailed to be placed in the **Test Plan**
* documented on a centrally located whiteboard and updated dynamically
* not formally documented at all

Based on your project culture, you should either list your specific testing tasks here or provide some descriptive text explaining the process your team uses to handle detailed task planning and provide a reference to where the details are stored, if appropriate.

For Master Test Plans, we recommend avoiding detailed task planning, which is often an unproductive effort if done as a front-loaded activity at the beginning of the project. A Master Test Plan might usefully describe the phases and the number of iterations, and give an indication of what types of testing are generally planned for each Phase or Iteration.

**Note**: Where process and detailed planning information is recorded centrally and separately from this Test Plan, you will have to manage the issues that will arise from having duplicate copies of the same information. To avoid team members referencing out-of-date information, we suggest that in this situation you place the minimum amount of process and planning information within the Test Plan to make ongoing maintenance easier and simply reference the "Master" source material.]

# Environmental Needs

[This section presents the non-human resources required for the **Test Plan**.]

## Base System Hardware

The following table sets forth the system resources for the test effort presented in this *Test Plan*.

[The specific elements of the test system may not be fully understood in early iterations, so expect this section to be completed over time. We recommend that the system simulates the production environment, scaling down the concurrent access and database size, and so forth, if and where appropriate.]

[**Note**: Add or delete items as appropriate.]

| **System Resources** | | |
| --- | --- | --- |
| **Resource** | **Quantity** | **Name and Type** |
| Database Server |  |  |
| —Network or Subnet |  | TBD |
| —Server Name |  | TBD |
| —Database Name |  | TBD |
| Client Test PCs |  |  |
| —Include special configuration requirements |  | TBD |
| Test Repository |  |  |
| —Network or Subnet |  | TBD |
| —Server Name |  | TBD |
| Test Development PCs |  | TBD |

## Base Software Elements in the Test Environment

The following base software elements are required in the test environment for this *Test Plan*.

[Note: Add or delete items as appropriate.]

| **Software Element Name** | **Version** | **Type and Other Notes** |
| --- | --- | --- |
| NT Workstation |  | Operating System |
| Windows 2000 |  | Operating System |
| Internet Explorer |  | Internet Browser |
| Netscape Navigator |  | Internet Browser |
| MS Outlook |  | eMail Client software |
| Network Associates McAfee Virus Checker |  | Virus Detection and Recovery Software |

## Productivity and Support Tools

The following tools will be employed to support the test process for this *Test Plan*.

[Note: Add or delete items as appropriate.]

| **Tool Category or Type** | **Tool Brand Name** | **Vendor or In-house** | **Version** |
| --- | --- | --- | --- |
| Test Management |  |  |  |
| Defect Tracking |  |  |  |
| ASQ Tool for functional testing |  |  |  |
| ASQ Tool for performance testing |  |  |  |
| Test Coverage Monitor or Profiler |  |  |  |
| Project Management |  |  |  |
| DBMS tools |  |  |  |

## Test Environment Configurations

The following Test Environment Configurations needs to be provided and supported for this project.

| **Configuration Name** | **Description** | **Implemented in Physical Configuration** |
| --- | --- | --- |
| Average user configuration |  |  |
| Minimal configuration supported |  |  |
| Visually and mobility challenged |  |  |
| International Double Byte OS |  |  |
| Network installation (not client) |  |  |

# Responsibilities, Staffing, and Training Needs

[This section presents the required resources to address the test effort outlined in the **Test Plan**—the main responsibilities, and the knowledge or skill sets required of those resources.]

## People and Roles

This table shows the staffing assumptions for the test effort.

[**Note**: Add or delete items as appropriate.]

| **Human Resources** | | |
| --- | --- | --- |
| **Role** | **Minimum Resources Recommended**  **(number of full-time roles allocated)** | **Specific Responsibilities or Comments** |
| Test Manager |  | Provides management oversight.  Responsibilities include:   * planning and logistics * agree mission * identify motivators * acquire appropriate resources * present management reporting * advocate the interests of test * evaluate effectiveness of test effort |
| Test Analyst |  | Identifies and defines the specific tests to be conducted.  Responsibilities include:   * identify test ideas * define test details * determine test results * document change requests * evaluate product quality |
| Test Designer |  | Defines the technical approach to the implementation of the test effort.  Responsibilities include:   * define test approach * define test automation architecture * verify test techniques * define testability elements * structure test implementation |
| Tester |  | Implements and executes the tests.  Responsibilities include:   * implement tests and test suites * execute test suites * log results * analyze and recover from test failures * document incidents |
| Test System Administrator |  | Ensures test environment and assets are managed and maintained.  Responsibilities include:   * administer test management system * install and support access to, and recovery of, test environment configurations and test labs |
| Database Administrator, Database Manager |  | Ensures test data (database) environment and assets are managed and maintained.  Responsibilities include:   * support the administration of test data and test beds (database). |
| Designer |  | Identifies and defines the operations, attributes, and associations of the test classes.  Responsibilities include:   * defines the test classes required to support testability requirements as defined by the test team |
| Implementer |  | Implements and unit tests the test classes and test packages.  Responsibilities include:   * creates the test components required to support testability requirements as defined by the designer |

## Staffing and Training Needs

This section outlines how to approach staffing and training the test roles for the project.

[The way to approach staffing and training will vary from project to project. If this section is part of a Master Test Plan, you should indicate at what points in the project lifecycle different skills and numbers of staff are needed. If this is an Iteration Test Plan, you should focus mainly on where and what training might occur during the Iteration.

Give thought to your training needs, and plan to schedule this based on a Just-In-Time (JIT) approach—there is often a temptation to attend training too far in advance of its usage when the test team has apparent slack. Doing this introduces the risk of the training being forgotten by the time it's needed.

Look for opportunities to combine the purchase of productivity tools with training on those tools, and arrange with the vendor to delay delivery of the training until just before you need it. If you have enough headcount, consider having training delivered in a customized manner for you, possibly at your own site.

The test team often requires the support and skills of other team members not directly part of the test team. Make sure you arrange in your plan for appropriate availability of System Administrators, Database Administrators, and Developers who are required to enable the test effort.]

# Iteration Milestones

[Identify the key schedule milestones that set the context for the Testing effort. Avoid repeating too much detail that is documented elsewhere in plans that address the entire project.]

| **Milestone** | **Planned Start Date** | **Actual Start Date** | **Planned End Date** | **Actual End Date** |
| --- | --- | --- | --- | --- |
| Iteration Plan agreed |  |  |  |  |
| Iteration starts |  |  |  |  |
| Requirements baselined |  |  |  |  |
| Architecture baselined |  |  |  |  |
| User Interface baselined |  |  |  |  |
| First Build delivered to test |  |  |  |  |
| First Build accepted into test |  |  |  |  |
| First Build test cycle finishes |  |  |  |  |
| [Build Two will not be tested] |  |  |  |  |
| Third Build delivered to test |  |  |  |  |
| Third Build accepted into test |  |  |  |  |
| Third Build test cycle finishes |  |  |  |  |
| Fourth Build delivered to test |  |  |  |  |
| Fourth Build accepted into test |  |  |  |  |
| Iteration Assessment review |  |  |  |  |
| Iteration ends |  |  |  |  |

# Risks, Dependencies, Assumptions, and Constraints

[List any risks that may affect the successful execution of this **Test Plan**, and identify mitigation and contingency strategies for each risk. Also indicate a relative ranking for both the likelihood of occurrence and the impact if the risk is realized.]

| **Risk** | **Mitigation Strategy** | **Contingency (Risk is realized)** |
| --- | --- | --- |
| Prerequisite entry criteria is not met. | <Tester> will define the prerequisites that must be met before Load Testing can start.  <Customer> will endeavor to meet prerequisites indicated by <Tester>. | * Meet outstanding prerequisites * Consider Load Test Failure |
| Test data proves to be inadequate. | <Customer> will ensure a full set of suitable and protected test data is available.  <Tester> will indicate what is required and will verify the suitability of test data. | * Redefine test data * Review Test Plan and modify * components (that is, scripts) * Consider Load Test Failure |
| Database requires refresh. | <System Admin> will endeavor to ensure the Database is regularly refreshed as required by <Tester>. | * Restore data and restart * Clear Database |

[List any dependencies identified during the development of this **Test Plan** that may affect its successful execution if those dependencies are not honored. Typically these dependencies relate to activities on the critical path that are prerequisites or post-requisites to one or more preceding (or subsequent) activities You should consider responsibilities you are relying on other teams or staff members external to the test effort completing, timing and dependencies of other planned tasks, the reliance on certain work products being produced.]

| **Dependency between** | **Potential Impact of Dependency** | **Owners** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

[List any assumptions made during the development of this **Test Plan** that may affect its successful execution if those assumptions are proven incorrect. Assumptions might relate to work you assume other teams are doing, expectations that certain aspects of the product or environment are stable, and so forth].

| **Assumption to be proven** | **Impact of Assumption being incorrect** | **Owners** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

[List any constraints placed on the test effort that have had a negative effect on the way in which this **Test Plan** has been approached.]

| **Constraint on** | **Impact Constraint has on test effort** | **Owners** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# Management Process and Procedures

[Outline what processes and procedures are to be used when issues arise with the **Test Plan** and its enactment.]

## Measuring and Assessing the Extent of Testing

[Outline the measurement and assessment process to be used to track the extent of testing.]

## Assessing the Deliverables of this Test Plan

[Outline the assessment process for reviewing and accepting the deliverables of this **Test Plan**]

## Problem Reporting, Escalation, and Issue Resolution

[Define how process problems will be reported and escalated, and the process to be followed to achieve resolution.]

## Managing Test Cycles

[Outline the management control process for a test cycle.]

## Traceability Strategies

[Consider appropriate traceability strategies for:

* Coverage of Testing against Specifications — enables measurement the extent of testing
* Motivations for Testing — enables assessment of relevance of tests to help determine whether to maintain or retire tests
* Software Design Elements — enables tracking of subsequent design changes that would necessitate rerunning tests or retiring them
* Resulting Change Requests — enables the tests that discovered the need for the change to be identified and re-run to verify the change request has been completed successfully]

## Approval and Signoff

[Outline the approval process and list the job titles (and names of current incumbents) that initially must approve the plan, and sign off on the plans satisfactory execution.]