Chương trình học toán lớp 10

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <4/5/2010> | <1.0> | Viết Test Plan | Trần Thị Mộng Kiều |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Intended Audience 5

1.4 Document Terminology and Acronyms 5

1.5 References 5

1.6 Document Structure 5

2. Evaluation Mission and Test Motivation 5

2.1 Background 5

2.2 Evaluation Mission 5

2.3 Test Motivators 6

3. Target Test Items 6

4. Outline of Planned Tests 6

4.1 Outline of Test Inclusions 6

4.2 Outline of other candidates for potential inclusion 6

4.3 Outline of Test Exclusions 6

5. Test Approach 7

5.1 Initial Test-Idea Catalogs and other reference sources 7

5.2 Testing Techniques and Types 7

5.2.1 Data and Database Integrity Testing 7

5.2.2 Function Testing 8

5.2.3 Business Cycle Testing 10

5.2.4 User Interface Testing 11

5.2.5 Performance Profiling 11

5.2.6 Load Testing 13

5.2.7 Stress Testing 14

5.2.8 Volume Testing 16

5.2.9 Security and Access Control Testing 17

5.2.10 Failover and Recovery Testing 18

5.2.11 Configuration Testing 20

5.2.12 Installation Testing 21

6. Entry and Exit Criteria 22

6.1 Test Plan 22

6.1.1 Test Plan Entry Criteria 22

6.1.2 Test Plan Exit Criteria 22

6.1.3 Suspension and resumption criteria 22

6.2 Test Cycles 22

6.2.1 Test Cycle Entry Criteria 22

6.2.2 Test Cycle Exit Criteria 22

6.2.3 Test Cycle abnormal termination 22

7. Deliverables 22

7.1 Test Evaluation Summaries 22

7.2 Reporting on Test Coverage 22

7.3 Perceived Quality Reports 22

7.4 Incident Logs and Change Requests 23

7.5 Smoke Test Suite and supporting Test Scripts 23

7.6 Additional work products 23

7.6.1 Detailed Test Results 23

7.6.2 Additional automated functional Test Scripts 23

7.6.3 Test Guidelines 23

7.6.4 Traceability Matrices 23

8. Testing Workflow 23

9. Environmental Needs 24

9.1 Base System Hardware 24

9.2 Base Software Elements in the Test Environment 24

9.3 Productivity and Support Tools 25

9.4 Test Environment Configurations 25

10. Responsibilities, Staffing and Training Needs 25

10.1 People and Roles 25

10.2 Staffing and Training Needs 27

11. Iteration Milestones 28

12. Risks, Dependencies, Assumptions and Constraints 28

13. Management Process and Procedures 29

13.1 Measuring and Assessing the Extent of Testing 30

13.2 Assessing the deliverables of this Test Plan 30

13.3 Problem Reporting, Escalation and Issue Resolution 30

13.4 Managing Test Cycles 30

13.5 Traceability Strategies 30

13.6 Approval and Signoff 30

# Introduction

## Purpose

Tài liệu này được dùng để :

* Test chương trình học toán lớp 10 được thực hiện bởi nhóm 11,12,13,14,15 trong môn xây dựng phần mềm
* Xác định được những phần sẽ được kiểm tra và đưa ra các test case để thực hiện test.
* Đưa ra những công cụ và những phương pháp thực hiện test dự án.
* Thực hiện việc tính công của các thành viên trong nhóm test.
* Đưa ra những kết quả đạt được sau khi test.
* Và để giao cho khách hàng , để khách hàng biết được chất lượng của chương trình.

## Scope

* Kiểm tra thiết kế và toàn bộ hệ thống sau khi tích hợp có thỏa mãn yêu cầu cài đặt hay không. Chú trọng vào các hành vi lỗi trên toàn bộ hệ thống
* Kiểm tra cả các hành vi chức năng của phần mềm lẫn các yêu vầu về chất lượng như độ tin cậy , tính tiện lợi khi sử dụng , hiệu năng và bảo mật. Mức độ kiểm tra này đặt biệt thích hợp cho việc phát hiện lỗi giao tiếp với phần mềm hoặc phần cứng , chẵng hạn như lỗi tắc nghẽn(deadlock) hoặc chiếm dụng vùng nhớ. Nhằm đảm bảo hệ thống đủ khả năng làm việc trong môi trường thực.

## Intended Audience

* Tài liệu này được áp dụng cho nhóm thiết kế test và nhóm thực hiện việc test chương trình học tiếng việt lớp 10, để thực hiện việc test chương trình.Và sẽ được chuyển giao cho khách hàng , cùng lúc với quá trình chuyển giao dự án,

## Document Terminology and Acronyms

* Unit là một thành phần PM nhỏ nhất mà ta có thể kiểm tra được. Theo định nghĩa này, các hàm (Function), thủ tục (Procedure), lớp (Class), hoặc các phương thức (Method) đều có thể được xem là Unit

## References

[This subsection provides a list of the documents referenced elsewhere within the **Test Plan**. Identify each document by title, version (or report number if applicable), date, and publishing organization or original author. Avoid listing documents that are influential but not directly referenced. Specify the sources from which the “official versions” of the references can be obtained, such as intranet UNC names or document reference codes. This information may be provided by reference to an appendix or to another document.]

## 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.]

# 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( ước lượng nhiệm vụ)

* 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
* Kiểm tra business Cycle
* Kiêm tra giao diện
* Kiểm tra khả năng vận hành.

## 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

### Data and Database Integrity Testing

[The databases and the database processes should be tested as an independent subsystem. This testing should test the subsystems without the target-of-test’s User Interface as the interface to the data. Additional research into the DataBase Management System (DBMS) needs to be performed to identify the tools and techniques that may exist to support the testing identified in the following table.]

|  |  |
| --- | --- |
| 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: | • [Invoke each database access method and process, seeding each with valid and invalid data or requests for data.  • Inspect the database to ensure the data has been populated as intended and all database events have occurred properly, or review the returned data to ensure that the correct data was retrieved for the correct reasons.] |
| 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 * base configuration imager and restorer * backup and recovery tools * installation-monitoring tools (registry, hard disk, CPU, memory, and so forth) * database SQL utilities and tools * Data-generation tools] |
| Success Criteria: | [The technique supports the testing of all key database access methods and processes.] |
| Special Considerations: | * [Testing may require a DBMS development environment or drivers to enter or modify data directly in the databases. * Processes should be invoked manually. * Small or minimally sized databases (limited number of records) should be used to increase the visibility of any non-acceptable events.] |

### Function Testing

[Function testing of the target-of-test should focus on any requirements for test that can be traced directly to use cases or business functions and business rules. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box techniques; that is verifying the application and its internal processes by interacting with the application via the Graphical User Interface (GUI) and analyzing the output or results. The following table identifies an outline of the testing recommended for each application.]

|  |  |
| --- | --- |
| Technique Objective: | [Exercise target-of-test functionality, including navigation, data entry, processing, and retrieval to observe and log target behavior.] |
| Technique: | [Execute each use-case scenario’s individual use-case flows or functions and features, using valid and invalid data, to verify that:  • the expected results occur when valid data is used  • the appropriate error or warning messages are displayed when invalid data is used  • each business rule is properly applied] |
| 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 * base configuration imager and restorer * backup and recovery tools * installation-monitoring tools (registry, hard disk, CPU, memory, and so forth) * Data-generation tools] |
| Success Criteria: | [The technique supports the testing of:  • all key use-case scenarios  • all key features] |
| Special Considerations: | [Identify or describe those items or issues (internal or external) that impact the implementation and execution of function test.] |

### Business Cycle Testing

[Business Cycle Testing should emulate the activities performed on the over time. A period should be identified, such as one year, and transactions and activities that would occur during a year’s period should be executed. This includes all daily, weekly, and monthly cycles, and events that are date-sensitive, such as ticklers.]

|  |  |
| --- | --- |
| Technique Objective: | [Exercise target-of-test and background processes according to required business models and schedules to observe and log target behavior.] |
| Technique: | [Testing will simulate several business cycles by performing the following:   * The tests used for target-of-test’s function testing will be modified or enhanced to increase the number of times each function is executed to simulate several different users over a specified period. * All time or date-sensitive functions will be executed using valid and invalid dates or time periods. * All functions that occur on a periodic schedule will be executed or launched at the appropriate time. * Testing will include using valid and invalid data to verify the following:   + The expected results occur when valid data is used.   + The appropriate error or warning messages are displayed when invalid data is used.   + Each business rule is properly applied.] |
| 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 * base configuration imager and restorer * backup and recovery tools * Data-generation tools] |
| Success Criteria: | [The technique supports the testing of all critical business cycles.] |
| Special Considerations: | • [System dates and events may require special support activities.  • A business model is required to identify appropriate test requirements and procedures.] |

### User Interface Testing

[User Interface (UI) testing verifies a user’s interaction with the software. The goal of UI testing is to ensure that the UI provides the user with the appropriate access and navigation through the functions of the target-of-test. In addition, UI testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.]

|  |  |
| --- | --- |
| Technique Objective: | Đối tượng cửa sổ và kí hiệu có thể thực hiện như menus, size….   1. TEST Mỹ Thuật 2. Màu toàn bộ màn hình có chính xác hay không? 3. Màu của các field có chính xac hay không? 4. Kiểm tra kích thướt font chữ có giống nhau hay không 5. Font chữ của các text trên màn hình có chính xác hay không? 6. Các button có được sắp thẳng hàng hay không? 7. Hình ảnh minh họa có chính xác hay không? 8. Thiết kế màn hình có giống như trong thiết kế giao diện hay không? 9. Các chữ trong từng field có đúng chính tả hay không? 10. Tất cả các ký tự và con số anpha trong field có được nằm về phía trái một cách hợp lý hay không? 11. Tất cả các thong báo lỗi trên màn hình có viết đúng chính tả hay không? 12. Chữ in thường , và chữ in hoa có phù hợp hay không , khi người dùng nhập vào? 13. Đảm bảo các text box có dễ nhìn hay không? 14. Tất cả các nút button phải giống nhau về hình dạng , kích thướt ,kiểu chữ, cỡ chữ. 15. Nếu trong màn hình có nhiều button thì phải để theo thứ tự như save , don’t save , cancel. 16. Test Tính Hợp lệ: 17. Khi xảy ra lỗi hay thất bai trong giá trị thì có thông báo lỗi hay không? 18. Nếu người dùng nhập vào giá trị không hợp lệ trong quá trình đăng nhập thì có thong báo lỗi hay không? 19. Test sự di chuyển: 20. Màn hình mới có thể được vào trực tiếp bằng việc chọn trong treeview hay không? 21. Có thể di chuyển lên xuống dễ dàng trong màn hình đang xem hay không? 22. Kiểm tra tính tiện lợi: 23. Các chương trong cây treeview có được sắp theo thứ tự hay không? 24. Các bài học và bài tâp và bài giảng có nằm theo thứ tự hay không? 25. Có thể dùng các phím nóng , hay phím tab để duy chuyển giữa các nhánh trong cây treeview hay không? 26. Test toàn vẹn dữ liệu: 27. Trong khi đang soạn thảo dữ liệu thi chương trình bị đóng đột ngột và phần bài đang làm dang dở chưa kịp save, vậy chương trình có chế độ tự lưu hay không? 28. Khi thực hiện việc thay đổi kích thướt màn hình làm việc , vậy thì dữ liệu có bị biến dạng hay không? 29. Test thông thường: 30. Chương trình có nút “Help ” hay không? 31. Trong chương trình có sử dụng phím nóng hay không? 32. Đảm bảo khi bấm vào button thì chúng thực hiện theo đúng như nhãn của chúng. 33. Việc chọn lựa button được thực hiện dựa vào phím nóng thay vì dùng chuột |
| Technique | Do chương trình được viết theo kiểu ứng dụng web , nên chỉ khác nhau trong quá trình chọn lựa trên cây tree view, và khác nhau ở cửa sổ hiện dữ liệu.   1. Vai trò sinh viên: 2. Chọn bài học\_ xem bài giảng  * Test Mỹ Thuật: * Màu của màn hình có giống với màu trong thiết kế hay không? * Bố cục của màn hình có giống như trong thiết kế hay không? * Đối với text trong màn hình , có đúng kích thướt , font , kiểu chữ? * Hình ảnh được hiển thị trong bài học có đúng hay không? * Có lỗi chính tả xảy ra trong text hay không? * Có ký tự nào bị lỗi font hay không? * Kiểm tra sự di chuyển: * Việc di chuyển lên xuống của màn hình có dễ dàng hay không? * Từ màn hình này sang màn hình khác có dễ dàng hay không? * Kiểm tra tính tiện lợi:   + Các mục trong chương của bài học có được sắp xếp theo đúng trình tự đã được miêu tả hay không?   + Việc di chuyển lên xuống của màn hình có thể sử dụng phím nóng hay không? * Kiểm tra sự toàn vẹn của dữ liệu: * Khi màn hình bị thay đổi kích thướt thì text có lỗi font chữ hay không? * Khi màn hình bị thay đổi kích thướt thì các hình ảnh minh họa có bị biến dạng hay mất sự cân đối hay không?      1. Chọn Bài Tập 2. Luyện trắc nghiệm:  * Test Mỹ Thuật: * Màu của màn hình có giống với màu trong thiết kế hay không? * Bố cục của màn hình có giống như trong thiết kế hay không? * Đối với text trong màn hình , có đúng kích thướt , font , kiểu chữ? * Hình ảnh được hiển thị trong bài học có đúng hay không? * Có lỗi chính tả xảy ra trong text hay không? * Có ký tự nào bị lỗi font hay không? * Các câu hỏi có được bố trí hợp lý và dễ nhìn hay không? * Kiểm tra sự di chuyển: * Việc di chuyển lên xuống của màn hình có dễ dàng hay không? * VIệc bố trí các câu trả lời có tiện cho người sử dụng hay không? * Kiểm tra tính tiện lợi: * Việc di chuyển lên xuống của màn hình có thể sử dụng phím nóng hay không? * Việc di chuyển giữa các câu hỏi có dễ dàng hay không? * Việc check vào các checkbox của đáp án có được thực hiện dễ dàng không? * Các câu hỏi có được đánh số rõ rang hay không? * Kiểm tra sự toàn vẹn của dữ liệu: * Khi màn hình bị thay đổi kích thướt thì text có lỗi font chữ hay không? * Khi màn hình bị thay đổi kích thướt thì các hình ảnh minh họa có bị biến dạng hay mất sự cân đối hay không? * Khi đang đánh trắc nghiệm , tự dưng chương trình bị tắt đột ngột , khi mở lên thì có còn đang ở hiện trạng đang làm việc hay không?  1. Gỉai đề thi- làm bài tập tự luận:  * Test Mỹ Thuật: * Màu của màn hình có giống với màu trong thiết kế hay không? * Bố cục của màn hình có giống như trong thiết kế hay không? * Đối với text trong màn hình , có đúng kích thướt , font , kiểu chữ? * Hình ảnh được hiển thị trong bài học có đúng hay không? * Có lỗi chính tả xảy ra trong text hay không? * Có ký tự nào bị lỗi font hay không? * Kiểm tra sự di chuyển: * Việc di chuyển lên xuống của màn hình có dễ dàng hay không? * Kiểm tra tính tiện lợi: * Các câu hỏi trong đề có được trình bày dễ hiểu hay không? * Việc di chuyển lên xuống của màn hình có thể sử dụng phím nóng hay không? * Khi thực hiện giải bài tập thì việc đánh chữ có dễ dàng hay không? * Có bị lỗi font chữ trong quá trình làm bài hay không? * Việc vẽ hình có được thưc hiện dễ dàng như vẽ trong paint hay không? * Việc ghi các kí tự toán học trong quá trình làm bài được diển ra dể dàng hay không? * Kiểm tra sự toàn vẹn của dữ liệu: * Khi màn hình bị thay đổi kích thướt thì text có lỗi font chữ hay không? * Khi màn hình bị thay đổi kích thướt thì các hình ảnh minh họa có bị biến dạng hay mất sự cân đối hay không?  1. Kiểm tra đáp án:  * Test Mỹ Thuật: * Màu của màn hình có giống với màu trong thiết kế hay không? * Bố cục của màn hình có giống như trong thiết kế hay không? * Đối với text trong màn hình , có đúng kích thướt , font , kiểu chữ? * Hình ảnh được hiển thị trong đáp án có đúng hay không? * Có lỗi chính tả xảy ra trong text hay không? * Có ký tự nào bị lỗi font hay không? * Kiểm tra sự di chuyển: * Việc di chuyển lên xuống của màn hình có dễ dàng hay không? * Từ màn hình này sang màn hình khác có dễ dàng hay không? * Kiểm tra tính tiện lợi: * Các bài giải của từng câu hỏi có được sắp xếp theo đúng trình tự đã được miêu tả hay không? * Việc di chuyển lên xuống của màn hình có thể sử dụng phím nóng hay không? * Việc di chuyển giữa màn hình làm bài lý thuyết và trả lời lý thuyết có được dễ dàng ko? * Hai màn hình có được bố cục theo kiểu mỗi màn hình sẽ chiếm nữa trang của màn hình dể tiện cho việc so sánh hay không? * Kiểm tra sự toàn vẹn của dữ liệu: * Khi màn hình bị thay đổi kích thướt thì text có lỗi font chữ hay không? * Khi màn hình bị thay đổi kích thướt thì các hình ảnh minh họa có bị biến dạng hay mất sự cân đối hay không?  1. Xem đáp án trắc nghiệm  * Test Mỹ Thuật: * Màu của màn hình có giống với màu trong thiết kế hay không? * Bố cục của màn hình có giống như trong thiết kế hay không? * Đối với text trong màn hình , có đúng kích thướt , font , kiểu chữ? * Hình ảnh được hiển thị trong bài học có đúng hay không? * Có lỗi chính tả xảy ra trong text hay không? * Có ký tự nào bị lỗi font hay không? * Kiểm tra tính tiện lợi: * Các câu trả lời có được để đúng trình tự câu hỏi hay không? * Kiểm tra sự toàn vẹn của dữ liệu: * Khi màn hình bị thay đổi kích thướt thì text có lỗi font chữ hay không?  1. Giai trí: 2. Xem phim- nghe nhạc:  * Việc chọn lựa bài hát hoặc đoạn phim trongdanh sach có dễ hay không? * Danh sách các bài hát hoặc phim có được bố trí theo bảng anpha hay không? * Trong quá trình bài hát hoặc đoạn phim được trình bày thì có xảy ra lỗi hay không? * Khi xảy ra lỗi thì có thong báo lỗi , và tìm cách giải quyết vấn đề hay không? * Chất lượng hình ảnh của bài hát và phim có tốt hay không? * Khi đang xem phim hay nghe nhạc, thì chương trình có phải bị dừng lại chờ dữ liệu truyền về hay không? * Khi xem phim và bài hát , có xảy ra tình trạng giật màn hình hay không?  1. Chơi game:  * Màn hinh khi chơi game có bị giật hay không? * Khi đang chơi game có phải ngồi đợi hơn 2s để dữ liệu được truyền về hay không? * Chất lượng hình ảnh của các game đó có tốt hay không?s * Danh sách các game có được sắp xếp theo bảng anpha hay không?  1. Vai trò giáo viên: 2. Soạn bài giảng  * Test Mỹ Thuật: * Màu của màn hình có giống với màu trong thiết kế hay không? * Đối với text trong màn hình , có đúng kích thướt , font , kiểu chữ? * Hình ảnh được hiển thị trong bài học có đúng hay không? * Có lỗi chính tả xảy ra trong text khi chúng ta viết hay không? * Có ký tự nào bị lỗi font hay không? * Kiểm tra sự di chuyển: * Việc di chuyển lên xuống của màn hình có dễ dàng hay không? * Kiểm tra tính tiện lợi: * Người sử dụng có thể chọn đúng dạng mong muốn trong quá trình soạn thảo hay không , như : chèn hình , chèn listbox, check box, việc tạo ra , xóa , sửa có dễ hay không? * Việc di chuyển lên xuống của màn hình có thể sử dụng phím nóng hay không? * Kiểm tra sự toàn vẹn của dữ liệu: * Khi màn hình bị thay đổi kích thướt thì text có lỗi font chữ hay không? * Khi màn hình bị thay đổi kích thướt thì các hình ảnh minh họa có bị biến dạng hay mất sự cân đối hay không? * Sau khi soạn xong , lưu lại , thì chương trình có lưu lại được tất cả những già chúng ta làm không ? và khi biểu diễn lên màn hình thì có biểu diễn đúng hay không?  1. Soạn bài tập trắc nghiệm  * Test Mỹ Thuật: * Màu của màn hình có giống với màu trong thiết kế hay không? * Đối với text trong màn hình , có đúng kích thướt , font , kiểu chữ? * Có ký tự nào bị lỗi font hay không? * Kiểm tra sự di chuyển: * Việc di chuyển lên xuống của màn hình có dễ dàng hay không? * Kiểm tra tính tiện lợi: * Các mục trong chương của bài hoc có được sắp xếp theo đúng trình tự đã được miêu tả hay không? * Việc di chuyển lên xuống của màn hình có thể sử dụng phím nóng hay không? * Việc soạn thảo tài liệu có dễ dàng không? * Việc them , xóa sửa một câu hỏi có dễ dàng hay không? * Việc bố trí câu trả lời có được thực hiện như mong muốn hay không? * Chúng ta có thể canh trái , canh phải cho các đoạn text hay không? * Kiểm tra sự toàn vẹn của dữ liệu: * Khi màn hình bị thay đổi kích thướt thì text có lỗi font chữ hay không?  1. Soạn bài tập tự luận  * Test Mỹ Thuật: * Màu của màn hình có giống với màu trong thiết kế hay không? * Bố cục của màn hình có giống như trong thiết kế hay không? * Đối với text trong màn hình , có đúng kích thướt , font , kiểu chữ? * Có lỗi chính tả xảy ra trong text hay không? * Có ký tự nào bị lỗi font hay không? * Kiểm tra sự di chuyển: * Việc di chuyển lên xuống của màn hình có dễ dàng hay không? * Kiểm tra tính tiện lợi: * Việc di chuyển lên xuống của màn hình có thể sử dụng phím nóng hay không? * Việc soạn thảo đề có được hỗ trợ : các kí hiện toán học hay không? * Việc thêm xóa sửa, một số text, buton, checkbox có thực hiện dễ dàng hay không? * Kiểm tra sự toàn vẹn của dữ liệu: * Khi màn hình bị thay đổi kích thướt thì text có lỗi font chữ hay không? * Khi màn hình bị thay đổi kích thướt thì các hình ảnh minh họa có bị biến dạng hay mất sự cân đối hay không? |
| Oracles | Ở đây do việc test được thực hiện bằng tay nên việc kiểm tra bằng oracle là không thể làm được. Nó chủ yếu dựa vào cảm nhận của một số tester và người dung thử, từ đó đưa ra nhận xét. |
| Required Tools: | Không cần công cụ, vì khu vực này sẽ thực hiện test bằng tay. |
| Success Criteria: | Tiểu chuẩn để xác định việc test thành công , là chương trình đáp ứng được hết những phần test case ở trên |
| Special Considerations: |  |

### Performance Profiling

[Performance profiling is a performance test in which response times, transaction rates, and other time-sensitive requirements are measured and evaluated. The goal of Performance Profiling is to verify performance requirements have been achieved. Performance profiling is implemented and executed to profile and tune a target-of-test's performance behaviors as a function of conditions such as workload or hardware configurations.

**Note**: Transactions in the following table refer to “logical business transactions”. These transactions are defined as specific use cases that an actor of the system is expected to perform using the target-of-test, such as add or modify a given contract.]

|  |  |
| --- | --- |
| 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: | • [Use Test Procedures developed for Function or Business Cycle Testing.  • Modify data files to increase the number of transactions or the scripts to increase the number of iterations that occur in each transaction.  • Scripts should be run on one machine (best case to benchmark single user, single transaction) and should be repeated with multiple clients (virtual or actual, 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 * an application performance profiling tool, such as Rational Quantify * installation-monitoring tools (registry, hard disk, CPU, memory, and so on * resource-constraining tools; for example, Canned Heat] |
| Success Criteria: | The technique supports testing:  • Single Transaction or single user: Successful emulation of the transaction scripts without any failures due to test implementation problems.]  • Multiple transactions or multiple users: Successful emulation of the workload without any failures due to test implementation problems.] |
| Special Considerations: | [Comprehensive performance testing includes having a background workload on the server.  There are several methods that can be used to perform this, including:  • “Drive transactions” directly to the server, usually in the form of Structured Query Language (SQL) calls.  • Create “virtual” user load to simulate many clients, usually several hundred. Remote Terminal Emulation tools are used to accomplish this load. This technique can also be used to load the network with “traffic”.  • Use multiple physical clients, each running test scripts, to place a load on the system.  Performance testing should be performed on a dedicated machine or at a dedicated time. This permits full control and accurate measurement.  The databases used for Performance Testing should be either actual size or scaled equally.] |

### 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

[Installation testing has two purposes. The first is to ensure that the software can be installed under different conditions⎯such as a new installation, an upgrade, and a complete or custom installation⎯under normal and abnormal conditions. Abnormal conditions include insufficient disk space, lack of privilege to create directories, and so on. The second purpose is to verify that, once installed, the software operates correctly. This usually means running a number of the tests that were developed for Function Testing.]

|  |  |
| --- | --- |
| Technique Objective: | [Exercise the installation of the target-of-test onto each required hardware configuration under the following conditions to observe and log installation behavior and configuration state changes:   * new installation: a new machine, never installed previously with * update: a machine previously installed , same version * update: a machine previously installed , older version] |
| Technique: | * [Develop automated or manual scripts to validate the condition of the target machine.   + new: never installed   + same or older version already installed * Launch or perform installation. * Using a predetermined subset of Function Test scripts, run the transactions.] |
| 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)] |
| Success Criteria: | [The technique supports the testing of the installation of the developed product in one or more installation configurations.] |
| Special Considerations: | [What transactions should be selected to comprise a confidence test that application has been successfully installed and no major software components are missing?] |

# Entry and Exit Criteria(vao và ra tiêu chuẩn)

## 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( hệ thống treo và bắt đầu lại tiêu chuẩn)

* 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( vòng tròn test bất thường kết thúc)

* 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, Window Xp,Window Vista, Window 7 |  | Operating System |
| Internet Explorer |  | Internet Browser |
| FireFox |  | InternetBrowser |
| Netscape Navigator |  | Internet Browser |
| Microsoft SQL 2005 |  | Visual Studio 2005 |
| 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.]