



PROJECT **PERFECT**
Pty Ltd

Testing User Guide

Prepared By: Neville Turbit
Version 1.0
1 Feb 09

Table of Contents

Document History.....	2
Overview	3
Definitions - Types of testing.....	4
Testing Activities.....	6
Test Strategy	7
Test Plan	9
Test Scripts	10
Test Management	11
Appendix A - Example - Defect Logging Form.....	13
Appendix B - Example - Defect Log.....	16

Document History

Version	Author	Date	Changes
0.1	Neville Turbit	1 Feb 09	First draft

Overview

Purpose of this Document The purpose of this document is to outline how testing should be planned and undertaken.

Testing definition Testing covers the validation of quality for the external deliverables of a project.

Scope of this document This document covers the various types of testing and how to establish and control testing in a project.

Scope exclusions This document does not cover the establishment of a quality plan. Setting up a project quality plan that might include testing is covered in the Quality Management User Guide.
Also excluded from this User Guide are the procedures around:

- Version Control
- Configuration Management

Definitions - Types of testing

Unit Testing Unit testing involves testing each component of a system for desired behaviour. A component is the smallest functional part of a program. The developers usually do unit testing. Unit testing is done against the developers view of expected functionality.

System Testing System testing involves testing that a system or module (for larger applications) built as a single development effort, works as required. System testing is usually done against the developers view of expected functionality.

Prototype Testing Prototype testing involves users reviewing a particular stage of the development in order to provide feedback to developers. Prototype testing provides elaboration and refinement of the business requirements.

User Acceptance Testing (UAT) UAT is testing by the ultimate users of the system to confirm the system:

- Operates in line with the specifications developed prior to development
- Fulfills the business needs

These may be contradictory. The system may fulfill the requirements of the specification however it may now become obvious the system will not fulfill business needs.

Integration Testing Integration testing is required for large systems where there are multiple development streams, and often in Major Enhancement or Maintenance situations where the system developed must interface with existing parts of the application and other systems.

Integration Testing involves testing to ensure different portions (normally modules) of the application work together and integrate with other applications if required.

Performance Testing Performance testing involves testing the application for performance, usually against specified criteria. The Developers and Business Users usually conduct it together.

The normal factors to test for are transaction rates, data volume and individual transaction size. Operation under various configurations of connected users may also be relevant.

Continued on next page

Definitions - Types of testing, Continued

Regression Testing

Regression testing is testing of a part of the system by the users after changes have been made. If a change has been made as a result of UAT, some of the testing may need to be repeated in order to ensure changes have not had an unexpected impact in another area.

Testing Activities

Overview Testing activities fall into three categories:

- Test Strategy
- Test Scenarios
- Test Scripts

Test Strategy The "Test Strategy" is focused on "how" the testing is to be undertaken. It details the approach to testing within the project.

Test Scenarios The "Test Scenarios" identifies what is to be tested. It lists the situations or scenarios.

Example:

- Enter a client name that is already in the system
 - Enter a client name that is not in the system
 - Enter a client name that exceeds the max length for surname
 - Etc.
-

Test Script The "Test Scripts" lists for each "Test Scenario":

- The actual information to be entered
- The function to be performed
- The expected result

Example:

Scenario: Duplicate customer entry

Step	Action
1	Enter a client: J. Smith
2	Enter address as 1 Smith St, Smithtown, 2222
3	Save the client details
4	Repeat step 1 and 2
5	Save the client

Expected result:

Reject with an error message "J. Smith already exists for this address."

Test Plan • The "Test Plan" is a document containing both the "Test Scenarios" and the "Test Scripts".

Test Strategy

Overview	A test strategy document should be produced to outline how the testing is to be undertaken. A template is available and an example is included as an appendix. Listed below are topics that should be covered.
Purpose	A high level statement of why the testing is taking place, the type of testing, and when it is to occur.
Types	Identify the type of testing to be undertaken.
Scheduled for	When is it to occur?
Resources	The people involved in the various types of testing should be nominated. If names are not available, skills should be identified.
Location	The physical environment in which the testing is being undertaken should be identified. This is of particular importance when it comes to UAT. Co-location is invaluable for both development and testing.
IT Environment	The IT environment is important to identify. It may include a dedicated test environment, copies of production environments or a subset of a production environment. The IT environment section may be extensive when it comes to UAT and prototype testing.
Equipment	If any additional equipment is required for testing, it should be identified. This may include PCs and printers for the UAT team or additional software.
Data Requirements	You will probably need some data to be loaded into the testing environment. It can range from a few manually created records to a full copy of the current production environment.
Backup & Restore	The frequency of backups and refreshes will be particularly important when it comes to UAT. It may be that it is done nightly, or on request of the Testing Manager.
Procedures - Problem identification	The procedure for identification of problems needs to be established. It should cover how a problem is identified, who is notified, and how it is to be tracked.

Continued on next page

Test Strategy, Continued

**Procedures -
Defect
rectification**

The procedure for rectifying defects needs to be established. It should cover how a problem is received, setting a priority, allocation, monitoring, testing before return, and returning.

**Process –
Retesting**

The procedure for re-testing should be documented from receipt of the rectification to either acceptance of the fix, or return for more work.

**Procedure -
Sign-off**

The procedure for sign-off should be identified. This includes authority levels and required level of compliance to sign-off the testing.

**Acceptance
Criteria**

How will we know when we have successfully completed testing? What are the criteria for the testing to be closed off and declared a success. It may be that all known defects are complete, or that no more than 4 “Medium” priority bugs are still outstanding.

**Testing
Software**

There are a number of tools on the market to assist testing. They range from tools to automate the data input and measurement of output, to tools that track defect rectification, to tools that link requirements to test scenarios. We provide an Excel based defect tracking log. If any are being used, they should be mentioned.

**Software - Test
management**

Test management software is the software used to identify what is being tested, the current state of the testing and any identified problems, and the ultimate sign-off.

**Software -
Testing**

Testing software is the software used to automate the test process. It essentially involves creating, storing and running custom test scripts within the application under test.

**Software -
Performance
Testing**

If performance testing is a consideration, it is appropriate to outline the software to be used for testing performance.

Test Plan

Overview The test plan should cover test scenarios and test scripts for each scenario.

Scope What is included and what is excluded from testing. There is a reference to Business Requirements, you can cross-reference the two.

Example:

Business Requirements	Test Scenarios
1.4.6 Client Details	2.4.1 Correct client details 2.4.2 Duplicate client details 2.4.3 Incorrect client details

The test plan should outline what functionality is to be tested. For example, it might cover:

- Order entry
 - Order adjustment
 - Order cancellation
 - Invoice generation
 - Etc.
-

Scenario Within each area of functionality, the various scenarios should be identified. The functionality provides a logical hierarchy. For example, within "Order entry", we might cover:

- All details correct
 - Incorrect customer number
 - Delivery date prior to current date
 - Delivery date a year in advance
 - No delivery address
 - Delivery address not listed against customer
 - Etc.
-

Checklist available There is a checklist of what might typically be a testing scenario. The link is in the "Carry out Testing" activity.

Order of testing Identify the order in which testing will be carried out. Usually this will follow some process flow. For example:

- Set up a customer
 - Process changes to customer details
 - Enter an order
 - Modify/delete order details
 - Generate an invoice
 - Modify delete an invoice
 - Produce statements
 - Etc.
-

Test Scripts

Overview “Test Scripts” are the “Test Scenarios” to be tested, the actual data that will be input and the expected results. If multiple tests of the same piece of functionality are to be carried out, there may be multiple scripts for each function.
The test scripts are the signoff mechanism for the testing.

Where to use The “Test Scripts” are primarily used for UAT. They may be used for “System Testing” or “Integration Testing” depending on the size and complexity of the project.

Example The following is an example of the information in a “Test Script”.

Information	Description	Example
Function	The function to be tested	Enter Name
Scenario	The situation to be tested within this functionality	Duplicate Name
Input	The actual data to be input	First Name: John Surname: Smith Address: 1 Smith St Suburb: Smithtown Postcode: 2222
Transaction	Process to be undertaken	Press "Enter" then try to re-enter the same data
Expected Result	What should happen	First entry should accept. Second entry should reject with message "John Smith already exists at this address"

Test Management

Overview Test management is the control of testing, defect management and re-testing.

Preparation for testing In preparation for testing, the following need to be in place.

- Test Plan
 - Test Scripts
 - Testing environment
 - Testers
 - Data loaded
 - Programs ready to test
-

Identification & logging When a suspected defect is identified it should be logged and passed to the development environment for rectification. Logging should include:

- Description of the defect
 - Situation that caused the defect
 - Desired result
 - Error messages
 - Any printouts or information that might help
 - Comments
 - Priority or urgency to have the defect rectified
-

Monitoring Whatever system is used to log the defects, there is a need to monitor the status of the defect.

Re-testing When a defect has been fixed and returned, it will need to be re-tested. Some thought will need to be given as to what else needs to be re-tested as there may be unforeseen impacts in another area. A discussion should be held with the development area to understand likely impacts.

Sign-off If the “Test Scripts” have been well constructed, successful completion of each “Test Script” should be a step towards final completion. When all the “Test Scripts” have been completed technically, the application is ready to sign off. Sign-off will include the signing off by the business based on prearranged acceptance criteria.

Continued on next page

Test Management, Continued

Random testing Whilst the development of “Test Scenarios” is a methodical way to cover all functionality, allow time to "play" with the system. By entering random data and processing transactions that might seem to be ridiculous, problems caused by events that may happen in a real life situation can be discovered. Time should be allowed to try and break the system.

Identify exceptions It is likely that by the deadline for sign-off, there may still be one or two minor defects remaining. To be pragmatic, the sign-off can occur with the proviso that the defects are either corrected before, or soon after the application goes into production. Obviously this will require re-testing.

Appendix A - Example - Defect Logging Form

Project ABC

Testing**Instructions**

Enter the type of testing e.g. UAT, Integration, System

Example:

- Unit Testing
 - System Testing
 - Performance Testing
 - Prototype Testing
 - **UAT**
 - Integration Testing
 - Regression Testing
-

Defect Number**Instructions:**

Set up a numbering system to track defects. The use of a defect log will help in this area.

Example:

No. 5

Urgency

- Critical
 - **High**
 - Medium
 - Low
-

Defect Name**Instructions:**

There should be a short name to identify the defect. If there are multiple similar problems with the same area, you can add a number e.g. Postcode Rejection 1, Postcode Rejection 2,

Example:

Postcode Rejection

Defect Description**Instructions:**

Describe the defect including the context for the defect.

Example:

Whilst entering a new client, an error message occurred when the postcode was entered. The message said the postcode was invalid although we had entered a valid code - 3000

Continued on next page

Appendix A - Example - Defect Logging Form, Continued

Desired Result

Instructions:

What did you expect to happen.

Example:

As 3000 is a valid code, the error message should not occur.

Error

Instructions:

Describe the error. This may be an error number appearing, or a failure to proceed, or the system may just accept the error.

Example:

Error 1234 - Invalid postcode

Printouts

Instructions:

Not only include any screen prints. Also include test scripts if they exist or any other relevant documentation such as requirements.

Example:

Attached is a screen print of the defect

Comments

Instructions:

Use this section to provide any further information that may be relevant. For example, the developer might think that all postcodes should be in one State. Explain how the system should operate and provide any explanations that might help.

Example:

Whilst all clients should be in NSW, it is reasonable to have a mailing address in another state or even overseas. In this case a postcode starting with other than a 2 is allowable. Blank should also be allowable.

Rectification

Instructions:

Explain what was done to rectify the fault

Example:

Changed the input field for name to make it a mandatory field.

Retesting

Instructions:

Explain the work carried out to retest. In particular how widely the testing was carried out.

Example:

We retested the client setup screen to ensure the name field was mandatory. Also tested the client change screen and found that the field was not mandatory on this screen.

Continued on next page

Appendix A - Example - Defect Logging Form, Continued

**Impact on
Business
Requirements**

Instructions:

Very often testing will uncover flawed or missing requirements. The impact on the original requirements should be noted, and requirements updated to maintain consistency. This will assist the supportability of the application.

Example:

The requirements did not mention this as a mandatory field. We have updated the requirements to reflect the changes.

Appendix B - Example - Defect Log

Defect Log

No	Name	Description	Date Raised	Raised By	Test Script	Referred to	Urgency			Date Returned	Rectification	Status		Attachments (Hyperlinks)
							L	M	H			C	Testing	
Totals														
							1	2	1	0		2	1	1
1	Postcode Error	Must enter a postcode. Some O/S addresses have no code.	1-Feb-08	JJ	D1.1	PP			X		3-Feb-08			X
2	Duplicate Name	Rejected the same name although it was a different person.....	1-Feb-08	JJ	D1.4	PP		X			4-Feb-08	X		
3	Name field too small	Only permitted 15 chars. See client 12345	1-Feb-08	JJ	D1.6	KK		X			5-Feb-08	X		
4	Typo "Address 1" on Name screen	Screen has "Address 1"	2-Feb-08	HH	T3.5	KK	X				5-Feb-08		X	

Defect Number **Instructions:**
Set up a numbering system to track defects. The use of a defect log (template available) will help in this area.

Example:

No. 5

Defect Description **Instructions:**
Describe the defect including the context for the defect.

Example:

Whilst entering a new client, an error message occurred when the postcode was entered. The message said the postcode was invalid although we had entered a valid code - 3000

Date Raised / Raised By **Instructions:**
Who found the defect and when

Test Script **Instructions:**
Which Test Scenario did this refer to?

Example:

1.1.1

Referred To **Instructions:**
Name of the person who is responsible for the defect rectification

Urgency **Instructions:**
How important is this to be fixed. For example if it stops any further testing it is critical. If it can wait until after go live, it is Low.

Example:

Low

Date Returned **Instructions:**
The date the defect was rectified and returned for testing.

Status Where is the defect currently? It is either being rectified, retested or is accepted. The status will determine the colour of the description cell.

Attachments Hyperlink to any documents that may be relevant. Could be screen prints or emails or requirements.
