



Developing a Test Strategy

Neville Turbit

Overview

This is the first of two documents on testing. The first is published in February 2006 and the second will be published in March. The purpose of these documents is to provide an outline of how to take the concept of User Acceptance Testing, and turn it into a tested product ready to go live. There are a number of steps to be undertaken along the way, and this white paper will provide a guide as to how and why it should happen in a certain sequence.

As with most undertakings, there is no absolute right way. We do not promote this as the only way to do the work. It is promoted however as the work you should do to improve your chances of getting it right.

Unfortunately, many organisations want to make commercial gain from a relatively straight forward process. In order to do this they create a world of jargon that you need to attend one of their training programs to understand. They create tools that support only their way of doing business. They lock you into a certain path that needs consulting support. By following the process outlined here, training, consulting and tools become options rather than mandatory.

Testing Steps

Looking at UAT from a high level, there are a few basic steps that need to be undertaken:

Step	Description
Test Strategy	Decide how we are going to approach the testing in terms of people, tools, procedures and support
Test Scenarios	What are the situations we want to test
Test Scripts	What are the actual inputs we will use? What are the expected results?

Test Strategy

Why do a Test Strategy? The Test Strategy is the plan for how you are going to approach testing. It is like a project charter that tells the world how you are going to approach the project. You may have it all in your head, and if you are the only person doing the work it might be OK. If however you do not have it all in your head, or if others will be involved, you need to map out the ground rules.

Here are some of the things that need to be covered in a Test Strategy. You could use this as a template for your own strategy.

Project Name

Overview

Testing stage

Instructions:

Identify the type of testing to be undertaken.

Example:

User Acceptance Testing

Scheduled for

Example:

01.04.06 to 15.04.06

Location

Example:

Testing will be carried out in the Test Center on Level X

Participants

Instructions:

Identify who will be involved in the testing. If resources have not been nominated, outline the skills required.

Example:

Testing Manager - J. Smith

2 Testers - To be nominated. The skills required are:

- Broad understanding of all the processes carried out by the accounts receivable area.
 - Should be familiar with manual processes currently undertaken for reversing payments.
 - Preferably spent time dealing with inquiries from customers over the phone
 - Etc.
-

Testing Environment

IT Environment

Instructions:

Outline the details of the environment to be used for testing.

Example:

Two test areas will be set up for testing. The first is to be used by the users to carry out user acceptance testing of the data input and transaction processing. To access the first area, select "Environment 500" on the security screen.

The second area will be used to test reports. This will be accessed as "Environment 501". Security access will need to be arranged through the System Administrator.

The following conditions apply to the environment:

- No more than 5 users need to use the environment concurrently
 - Users should have the ability to run month end processing
 - Transaction logs need to be maintained for all processing
 - Etc.
-

Equipment Environment

Instructions:

Identify the equipment required for testing.

Example:

We will require:

- 2 PCs with Windows XP
 - 2 PCs with Windows 2000
 - All with access to the LAN
 - A black and white laser printer
 - A colour printer
 - Etc.
-

Data

Instructions

Identify the data to be used for testing.

Example:

"Environment 500" will contain:

- 1000 customers selected at random from out production database (every 2000th customer alphabetically).
 - Transaction date ranges between 1990 and 2010 can be used.
 - The database has no records for products x and y
-

Backup

Instructions

Identify how often data backups should be undertaken and who has responsibility for backups. Also cover how long backups should be retained.

Example:

Backups should occur:

- Nightly
- During the day at the request of the Test Manager

Responsibility for backups is listed below:

- Test Manager responsible for requesting backups at unscheduled times
- Test Manager responsible for delaying or canceling nightly backups
- Operations responsible for completing backups

Nightly backups should be a full backup. Unscheduled backups should be incremental backups.

Restore

Instructions:

Identify how and when a restore should take place.

Example:

The Test Manager will advise Operations if and when a restore is required to either do a complete refresh of the data or restore to a previous backup.

Procedures

Problem Identification

Instructions:

Identify the procedure to be used when a tester finds a suspected defect. There should be a nominated resource to receive all defects. In some cases there may be more than one resource e.g. one person for applications problems and one for operational problems.

Example:

Step	Action
1.	Identify suspected problem
2.	Refer to Test Manager
3.	Does the Test Manager validate the defect <ul style="list-style-type: none"> • If yes, go to Step 4 • If no, quit
4.	Fill out a "Defect Logging Form"

5.	Enter in the "Defect Log"
6.	Forward to nominated resource for rectification

**Defect
rectification**

Instructions:

Identify how the defect will be managed once received. This procedure would normally be under the control of the person or people rectifying the defect.

Example:

Step	Action
1.	Receive a "Defect Logging Form"
2.	Schedule the rectification based on the priority
3.	Allocate the defect for investigation and rectification
4.	If defect can be reproduced/validated <ul style="list-style-type: none"> • Rectify and test the defect • Go to Step 6
5.	If defect cannot be reproduced/validated <ul style="list-style-type: none"> • Discuss with Test Manager • Go to Step 6
6.	Return to Test Manager.
7.	Update "Defect Log"

Re-testing

Instructions:

Identify the procedure for re-testing rectified defects.

Example:

Step	Action
1.	Receive rectified defect
2.	Discuss possible implications of rectification with the person who carried out the rectification
3.	Determine the level of re-testing required
4.	Draw up a test plan Or Identify what part of the original test plan will need to be repeated
5.	Carry out testing

6.	<p>If testing is successful</p> <ul style="list-style-type: none"> • Update the "Defect Log" by signing off the fix • Quit
7.	<p>If testing is unsuccessful</p> <ul style="list-style-type: none"> • Update the "Defect Logging Form" • Return to the nominated person • Update the "Defect Log" to show it is once again awaiting rectification

Sign-off testing activities

Instructions:

Identify the procedure that will be used to sign-off each activity in the testing. This includes both the testing outlined in the "Test Plan" and testing of defects that have been rectified.

Example:

Step	Action
1.	Carry out each testing activity in the "Test Plan" using the "Test Scripts".
2.	If testing successful, go to Step 4
3.	If unsuccessful, see procedure for "Problem Identification".
4.	Sign and date the "Test Script"

Sign-off Testing

Instructions:

Identify how the total testing will be signed off. This includes all activities in the "Test Plan" and any rectification of defects.

Example:

Testing will be considered successful when the following conditions are met:

- All testing identified in the "Test Plan" has been completed
- No defects classified as priority "Critical" exist
- No defects classified as priority "High" exist
- Less than 3 defects classified as priority "Medium" exist
- Less than 6 defects classified as priority "Low" exist
- Outstanding defects classified, as "Medium" will be returned from rectification within 3 working days.

Release Management

Release Mgmt considerations

Instructions:

One of the major risks in testing is not having a proper Release Management Procedure. The same program is modified by two people at the same time, and only one modification gets into production. It is important to put in place a proper release management process.

Example:

“Prior to any testing being undertaken, a complete and documented Release Management facility must be in place.

The Release Management facility will take a similar form to the following:

1. There must be specified areas designated as “Release areas” and identified by Release Dates available for storage of accepted software.
 2. All software (modified and new) will be implemented into the production or test environment only after being accepted by the business users, and on a scheduled release date.
 3. The published implementation schedule (dates) will be generated and controlled by the business users in consultation with the Version Control Committee.
 4. No software will be implemented into the production environment outside the published implementation schedule
 5. A complete list of software that is to be implemented on a scheduled release date will be released and published to all business users three days prior to implementation.
 6. A release will be designated “frozen” and no other software will be allowed to be moved into this frozen release within one day of the released date. Any further new or modified software will be scheduled for the next release date.”
-

Software

Test management software

Instructions:

Identify any software that will be used to manage the testing. This includes the organisation of the activities to be tested, and the management of defects. Date ranges between 1990 and 2010 can be used..

Example:

We will use "Test Manager" to manage the testing. Both the test team and applications development will have access and be able to update the status of defects.

Testing Software

Instructions:

If any automated testing software such as WinRunner is to be used, detail the software. Also identify if the software will need to be purchased.

Performance Testing Software

Instructions:

If any performance testing software is to be used, detail the software. Also identify if the software will need to be purchased. In some cases, performance testing may be a separate activity to UAT.

Summary

The template above covers most of the normal issues that need to be in a strategy. For each project there will be specific issues that need to be included. The document will probably not be completed by just one person. It will require input from a range of people who will have different areas of expertise. For example, release procedures will probably be the responsibility of someone in a system administration role.

The more people you talk to, the more problems you will avoid. Part of the reason for developing a strategy is to prompt people to think about the impact on their area. Developing a test strategy is a critical part of test planning.

Neville Turbit has had over 15 years experience as an IT consultant and almost an equal time working in Business. He is the principal of Project Perfect. Project Perfect is a project management software consulting and training organisation based in Sydney Australia. Their focus is to provide creative yet pragmatic solutions to Project Management issues.

Project Perfect sell “Project Administrator” software, which is a tool to assist organisations better manage project risks, issues, budgets, scope, documentation planning and scheduling. They also created a technique for gathering requirements called “Method H”™, and sell software to support the technique. For more information on Project tools or Project Management visit www.projectperfect.com.au