Example Test Plan

DOCUMENT INFORMATION	
Project Name	
Project Number	
Project Account Code	
Author	
Issue Date	

DOCUMENT HISTORY			
Version	**Date**	**Summary of change**	**Author**
1.0	12/08/2014	Release version	Pete Castle
2.0	01/10/2015	Wholesale Content Changes	Pete Castle

APPROVALS			
Name	**Title**	**Signature**	**Escalation**

DOCUMENT CIRCULATION			
Version	**Date**	**Person/Group**	**Author**

CONTENTS

1. HOW TO USE THIS DOCUMENT

This document is an example of a Software Test Plan. It uses example sections from various plans and is not a single live document. The intention is to show what type of information should be incorporated into a Test Plan, and is not representative of a single project.

To ensure a well-tested product, testers need a thorough understanding of the context of the project, the project goals, the system functionality, the system risks and vulnerabilities, the test equipment, and the procedures and tools to be used. The Software Test Plan aims to capture some of this information.

It is important to capture as much information about the testing requirements as early as possible within the Software Development Life Cycle (SDLC). Early involvement of testers means less of a learning curve to master the system, more time to determine how to test, time to prepare testing and the test environment, less cost to make changes and adjustments. The test team should actively participate in developing the overall project work plan, contributing to work product breakdown, scheduling, and providing the testing estimates

2. INTRODUCTION

This section describes the context in which the testing is to take place – should be high level – no more than one side of A4. An outline of the dates of the overall project and the testing (start/finish) should also be included here.

For Example:

Portico currently experiences several heavy reporting periods through Business Objects per annum. 2007 proved a difficult year with problems surrounding the performance of the reports, leading to emergency remedial action being undertaken.

The next heavy load period is expected in May 2008, for the Assessment and Awards reporting period. In particular, reports will be required in time for the Examination Board deadlines, at the beginning of June 2008. The Business Objects pre-upgrade review advised that a dual-server, fault-resilient clustered solution be deployed to improve report performance. This configuration is due to go-live on 15th April 2008, prior to the heavy load period.

A capacity test is planned for Wednesday 16th April, to test the robustness of the technical reporting infrastructure.

3. OBJECTIVES

This section describes the objectives of the testing – which can vary from project to project. A testing project might be to verify the quality of software developed by a 3rd party or to concentrate on particularly area of usability to reduce the number of support queries that are being logged.

One mandatory objective for the testing will be to ensure is compliance with the Disability Discrimination Act and The Equalities Act 2012 (see later)

It should also include an outline of the type of testing – references to the different types of testing can be found at http://csqa.info/software_testing_glossary

For example:

An objective for a round of Integration Testing might be "To test the functionality exercised by one part of the system does not cause conflicts or issues elsewhere in the system."

An objective for a round of System Testing might be "To ensure that the products functionality performs as of the User requirements and Technical design Documents"

An objective for a round of User Acceptance Testing might be "To verify that Product X is suitable for release to customers live environments for 09:00 99/99/99."

4. SCOPE

4.1 Test items

This section of the plan will outline those ideas which are in scope for this testing project. It will also include some mandatory elements that must be included in all test plans – non-functional and accessibility requirements.

For example:

This table shows the areas of a proposed system of reporting which require system testing.

Non-functional Requirements
Report performance will be tested against the performance criteria as set out in Functional Requirements XXXX.doc
The upgrade process for quarterly software updates from the supplier will be tested
Accessibility
Accessibility testing will be conducted to ensure the system is compliant with the Disability Discrimination Act and The Equalities Act 2012.
A&A Reports
Compressed_format_report
Classification_with_signatures
Progression_summary_report
Summary_report
Progression_summary_report
Other Reports
Main_Transcript
Change_of _Marks

4.2 Features to be tested

This section details the features within the scope which are to be tested.

For Example:

Due to the time and resource pressures on this release, only those areas identified as high risk and those that have had work undertaken on them will be covered as part of this Integration Test Phase. Reports will be content, layout and performance tested.

4.3 Features not to be tested

It is important to explicitly detail those features and functionality that will not be tested. On far too many projects assumptions are made about what is being tested and by whom – resulting in features going untested and customers finding the issues.

For Example:

The reports will not be tested for levels of concurrent users.

5. APPPROACH

A description of how to test the product should be included here. This will vary from project to project – and is something that should be included in all test planning.

For Example:

Will standard test scripts be written ready for transformation into a regression pack? Will a RAD approach be used, allowing developers and testers to work together, forsaking standard scripts and building ad-hoc test documentation?

This section should also contain the tools to be used (commercial and in-house)

For Example:

- *JAWS – Screen Reader (accessibility)*
- *TestDirector – Test Management*
- *AllClear – Change Management*
- *Beyond Compare Ver.2 – File Comparison Tool*
- *XMLspy – XML Viewer*

5.1 Test Scripts/Documentation

This section describes the scripts and documentation to be produced

- A test Script is defined as ‘a collection of one or more test cases for the software under test’.

- A Test Case is ‘A set of inputs, execution preconditions, and expected outcomes developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.’

BCS SIGIST Glossary of Testing Terms 2000

For Example:

ISD standard test scripts (using Ver. 1.02 template and guidance) are to be produced, and are accessible at <Drive:/folder/sub folder/project.

ISD Standard UCL accessibility script templates will be used (using Ver. 1.02template and guidance)

5.2 Schedule

This section outlines the major dates associated with this plan – it is not a project plan but gives readers the testing milestones.

For Example:

Milestone Task - Phased Plan	***Effort (Days)***	***Start Date***	***End Date***
Test Plan Sign off	*1*	*01/04/08*	*01/04/08*
System Test Scripts	*10*	*03/04/08*	*17/04/08*
System Testing	*7*	*18/04/08*	*28/04/08*
Entry Into UAT	*1*	*29/04/08*	*29/04/08*

5.3 Deliverables

All work products will be subject to a review - either peer review or formal inspection. It is expected that peer review will be sufficient for the majority of testing work products.

For Example:

Those of specific interest to testing are:

Test Plan - this document

Test Estimates – How much testing effort is required for testing the different areas of functionality?

Issues Log incorporating Test Metrics (information distributed during the testing phase to inform stakeholders of progress and issues)

Impacts Matrix - particularly relevant with an integration test

Test Schedule including Test Resources (this may be part of the overall project plan and can therefore be omitted)

Test Scripts – Manual, Automated, SQL, high level scenarios, Accessibility

Test Defects – All Chang

Test Completion Report – Produced after every testing phase

These documents can be found at the http://www.pierre_thru_the_window/

5.4 Defects & Metrics

5.4.1 Defects

This section highlights how defects found during testing are to be classified and managed

For Example:

*Any **defects** identified during testing will be logged in EISD Management Servers Mantis under the project Portico Release Sept 08. Classification of defects by severity will be as follows:*

Priority	***Description***
0 – Showstopper (Fix immediately)	*Showstopper error. Halts release, halts testing*
1 – Urgent (Fix this Release)	*Fatal error/crash, GPF, data corruption, infinite loops, stopping test*
2 – High (Fix for next release)	*Fatal errors in less critical areas, wrong answers to calculations, performance, resource problems, functionality not responding in specified manner, etc*
3 - Medium (To be fixed)	*Non-critical errors, difficult to use, screen/print corruption etc*
4 – Low (Being considered)	*Ideas to improve*
5 – As Agreed	
6 – As and When	

With challenging test schedules, change control must be exercised on all defects. However, it is expected that some defects found will not be fixed. Any defects that are found will be evaluated as to the risks associated to the release, before they are fixed.

Once into UAT defects raised will be managed by the ISD Change Control Board, which will meet as described in the PID for this project. The ISD Change Control board will comprise both Programme and Project Management with input from Test and Development.

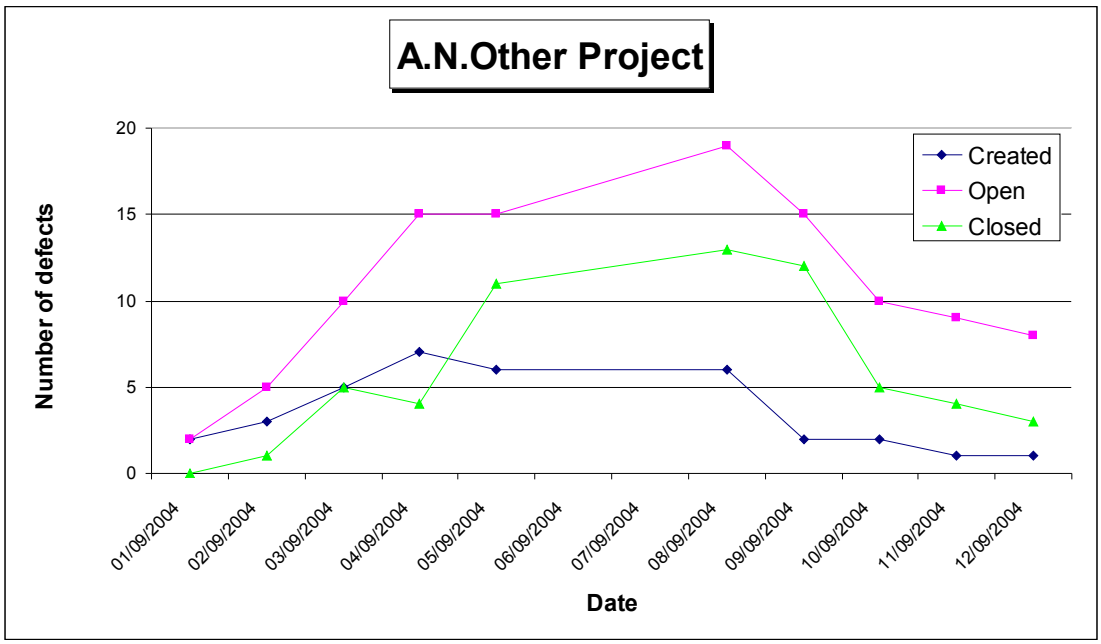
5.4.2 Test Metrics

Testing metrics are an import aid to knowing progress on a test project. The overall number of defects, the rate at which defects are being found, and their severity – are all useful information in assessing the quality of the product under test.

For Example:

Project Test metrics will be produced on a weekly basis during testing.

Total Defects
Outstanding Defects
Defect Severity Analysis
Defect Discovery Rate – (see below for example)



These should be published with the project highlight reports.

5.5 Test Suspension and Entry / Exit Criteria

5.5.1 Suspension Criteria

It is highly unlikely that testing be suspended due to timescales imposed for release, however if an unacceptable amount of faults are found, certain tests may need to be suspended upon consultation with Project/Programme Management.

5.5.2 Entry/Exit Criteria

This section details 'Exit from/ and Entry to' criteria for the differing Phases of Testing

For Example:

Entry into UAT Testing

- *System Testing Test Plan has been 'Signed Off' by all requested parties*
- *Successful Build of the Platform produced and installed in the UAT environment.*
- *UAT Test Scenarios have been created for the Release*
- *UAT Test Data has been sourced and prepared as required*
- *UAT Resources have been scheduled have been identified*

5.5.3 Exit from UAT into Live Criteria

The following exit criteria have been defined for exit from the Integration Phase (it is recognized that these criteria represent targets to aspire to, and may not be met within a particular release):

- No outstanding BTRs (Barriers to Release). Any high priority defects that are still in place are subject to examination by the Change Control Board.
- All Test Metrics to be available
- All outstanding Defects to be documented and the information made available
- User Acceptance Testing has been 'Signed Off' by Prime Customers.

5.6 Resources & Responsibilities

5.6.1 Roles and Responsibilities

This section details the expected workload to be undertaken by the test team and project members required to support the test function. Any additions or modifications will be discussed with the Project Manager.

For Example:

Staff Name:	***Role:***	***Deliverable or Work Package:***	***Responsibility:***
Teddy Tester	*Lead System Tester*	*Test Plan* *Test Schedule* *Test Estimates* *Test Scripts*	*Management of work products and the overall test lead and management of the test activities in addition to the production and management of integration test deliverables. It is also the responsibility of the integration test lead to facilitate communication and cooperation across project teams, and to liaise with project management*
Harry Hardware	*Hardware Support*	*Configuration of Test Environment*	*Ensure the test environment is installed and configured prior to System Test start date of 18/04/08*
Doris Developer	*Software Developer*	*Support System Test as necessary*	*Allocation of 20% to support test team*
Alison Analyst	*Business Analyst*	*Support Test as necessary*	*Allocation of 30% to support test team*

5.6.2 Resources and Environments

This section describes the hardware, software and permissions that are required to undertake the planned testing.

For Example:

1. *The test lab must contain a minimum of 2 networks (Windows 2003 and XP) and 2 standalone installations (Windows 2000, XP, Vista). Test data appropriate for the testing of multiple modules will need to be installed onto these machines.*

2. *A combination of IE6 and IE7, Office 2003 and 2007 should be installed.*

3. *The Test lab domain will have restricted windows users set up – both domain users and power users in addition to the normal administrator. In addition, a user with a mandatory profile will be required, and testing will incorporate this type of user.*

5.6.3 Test Datasets

This section details data sets to be sued and any additional data requirements – especially for new functionality.

For Example:

The testing will take place on a data set in the academic year 07/08. This will comprise a copy of live that has been manipulated with a data masking tool.

6. RISKS & CONTINGENCIES

This section details any risks and contingences for the testing (as against the overall project)

For Example:

Risk Or Issue	***Risk Level***	***Likely Effects And Mitigation***
Insufficient time for testing to be complete	*High*	*Test coverage may not meet exit criteria. Focus testing on critical and high risk areas as defined by Cross Dependencies matrix and risk analysis.*
UAT Testers not available	*High*	*Liaison with Project teams and departments to ensure products are ready for testing as scheduled.*

7. SIGN OFF / APPROVALS

All test plans are subject to sign-off – people within the programme should know how their products are going to be tested, and agree with the planned action. If not they should raise their issues and have the plan changed, if not their objections must be recorded.

For Example:

The following approve the contents and intentions of the XXXXXXXXXXXXX EISD Management Services Test Plan.

Name:	***Appointment:***	***Approval Signature:***	***Signatory Remarks:***
	Project Executive		
	Product Manager		
	Release Manager		
	Project Manager		
	Test Manager		
	Business Analyst		

8. APPENDIX

Anything not quite fitting anywhere else can be added into this section. For example a list of escalation fixes and descriptions which are being delivered into a new version of the software but which were not part of the original project.