Client Logo

<Project Name>

Information Management Test Plan

Version 0.1

<Date>

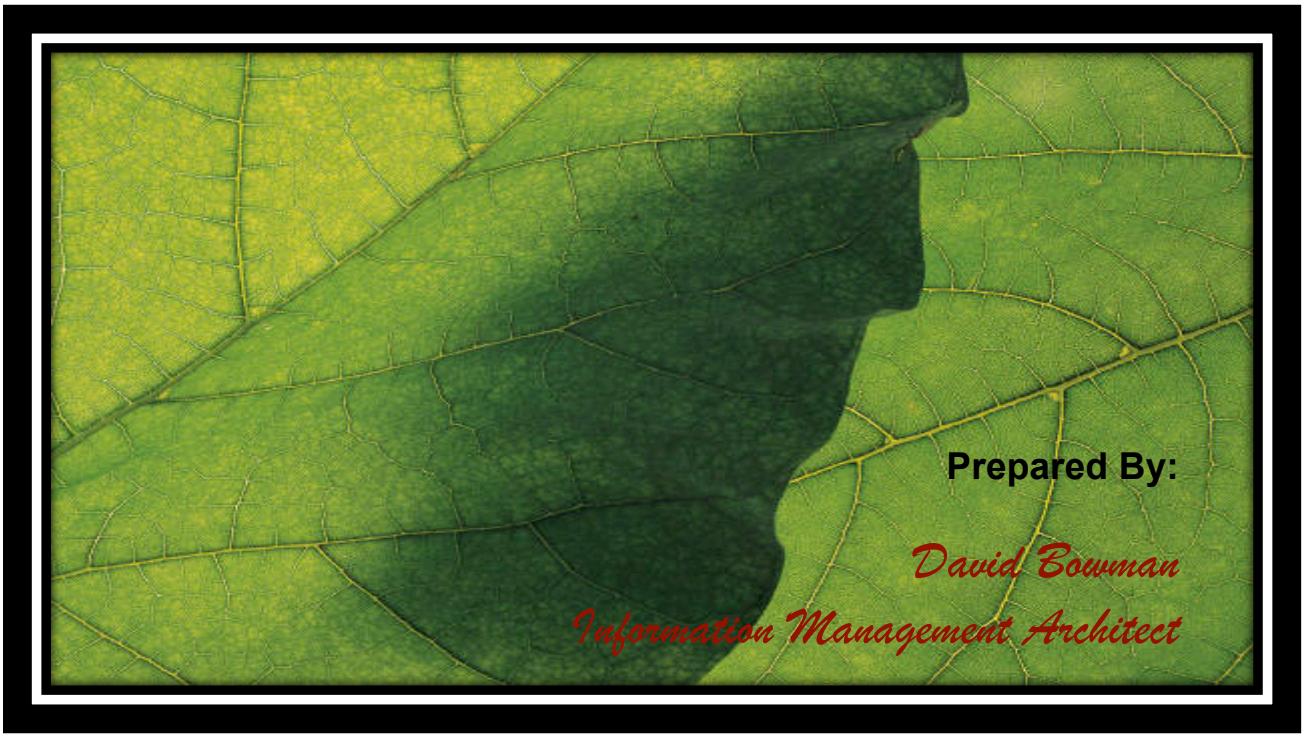

Table of Contents

Revision History

Version	Date	Description	Author

1. Introduction

Purpose

This plan documents the understanding, approach and commitment to test the **<Project Name>**

1.1. Reference Documents

Document Name	Version	Version Date	Location

1.2. Glossary

<Provide a refer to the project glossary of terms>

2. Testing Scope

2.1. Test Objectives

<The following sample can be modified as necessary>

2.1.1. Unit Test Objective

Unit testing certifies that each individual module is built to design specifications, in the developer's personal schema, and is ready for system testing

2.1.2. System Test Objective

System testing certifies that a related group of modules is built to design specifications and, is ready for integration testing.

2.1.3. Integration Test Objective

Integration testing certifies that a "build" is built to design specifications and is ready for QA testing

2.1.4. Regression Test Objective

Regression testing certifies that a related group of modules does not adversely impact code that is currently in production.

2.1.5. QA Test Objective

QA testing certifies that a release meets business requirements and is ready for user acceptance testing (UAT)

2.1.6. UAT Test Objective

UAT certifies that a release meets user expectations and is ready for release to production.

2.2. Scope

<List the scope of testing included in this plan>

2.2.1. In Scope

<List all in-scope items>

2.2.2. Out of Scope

<List all out-scope items>

2.3. Test Deliverables

<Document the test deliverables. The following sample can be modified as necessary>

Phase	Deliverable	Owner	Contributors
Analysis	Test Plan		
Design	System Test Case/Steps		
	Integration Test Cases/Steps		
	QA Test Cases/Scripts		
	UAT Test Cases/Script		
	Test Data Acquisition Plan		
Build	Prepared System & Integration Test Environments		
	Unit Test Cases		
	Unit Test Results		
	System Test Scripts		
	Approved System Test Data		
	System Test Results		
	Integration Test Scripts		
	Approved Integration Test Data		
	Integration Test Results		
Test	Prepared QA Test Environment		
	QA Test Scripts		
	Approved QA Test Data		
	QA Test Results		
	Approved UAT Test Data		
	UAT Test Results		

2.4. Assumptions

<Identify any test assumptions>

2.5. Test Constraints

<Identify any test constraints>

3. Resource Plan

<Document the roles and responsibilities for everyone involved with testing. Be sure to include the test team, the technical team, the user team and any production support staff that may be required to help run test processes>

3.1. Roles & Responsibilities

Role	Responsibilities	Assigned To

3.2. Environment

<List the physical environments needed to perform testing, including hardware, software, and database needs. Identify all requirements for each environment. The following provides sample items for consideration>

System Resources	
Resource	**Name/Type**
Database Server	*<Server or region name>*
Include special configuration requirements	*<Software that is needed on the server to conduct testing>*
Network/Subnet	
Server Name	
Database Name	
Non-Production requirements (and why they are being requested)	*<Some examples could be 'log level set to xx - lower than production'>*
Web Server	
Communication Server	
Middleware Server	
Client Test PC's	
Include special configuration requirements	
Non-Production requirements (and why they are being requested)	

System Resources	
Resource	**Name/Type**
Test Artifact Repository	***<Where test documentation will be kept – whether in a Tool (like Test Director) or in a file system>***
Network/Subnet	
Server Name	
Test Development PC's	
User Access	***<Any special User ID profiles (for access to the network or server) that will be needed for proper access>***

3.3. Test Tools

This section identifies the tools needed to perform the tests that will be used by the project team and the version of the tool.

<Add or delete table items, as necessary>

Tool Type	Tool	Vendor/In-house	Version
Test Management			
Defect Tracking			
Automated Tool			
Performance testing			
Test Coverage Monitor or Profiler			
Project Management			
DBMS tools			
Configuration Management			

3.4. Training Considerations

<This section should address any action required to eliminate the skills gap through training. The project should only be responsible for providing and funding training on skills specific to the project and outside of the general body of knowledge for the role specified>

3.5. Test Team Organization

<Describe the organizational context, both technical and managerial, within which the planned testing is to be implemented. An organizational chart may be inserted to identify the following:

- ***All organization units that participate in or are responsible for any project activity:***
- ***The functional roles of these organizational units within the project structure.***
- ***Relationships between organizational units.***

Organizational units may consist of:

- ***Vendor and customer.***
- ***Prime contractor and subcontractors.***
- ***Different groups within one organization>***

4. Test Plan

<The following sample should be created for each component of the test plan e.g.

- Unit Test;
- System Test;
- Integration Test;
- Regression test;
- QA test;
- UAT Test>

4.1. Unit Test

4.1.1. Scope

<Document scope of this test>

4.1.2. Unit Test Entrance Criteria

<List the entrance criteria that should be met before this test commences>

4.1.3. Test Cases

<Specific who will create test cases, how they will be created and where they will be stored>

4.1.4. Test Data

<Describe what test data is required and who will create it>

4.1.5. Test Scripts

>Identify who will create test scripts>

4.1.6. Test Execution

<Describe how testing will be completed>

4.1.7. Defect Management

<Describe the defect management approach>

4.1.8. Test Results

<Describe where test results will be stored—Usually in the test tool>

4.1.9. Unit Test Exit Criteria

<List the exit criteria that should be met before this test is considered complete>

5. Test Data Strategy

5.1. Unit Test Data

Unit Test data shall be created by the individual developer who shall create sufficient data to handle positive/negative tests;

5.2. System Test Data

<Define test data strategy>

5.3. Integration Test Data

<Define test data strategy>

5.4. QA Test Data

<Define test data strategy>

5.5. UAT Test Data

<Define test data strategy>

6. Testing Communications Plan

6.1. Test Status Reporting

<Describe the test status reporting process>

6.2. Test Defect Reporting

<Define how defects will be reported. Identify the tool that will be used—usually the testing tool. Describe the severity levels (the following is suggested)>

Severity	Definition
1-Critical .	Defects that prevent further progress until resolved. In testing, the entire application component or function will not work. There is no work-around available, and testing cannot continue. This kind of defects will definitely impact testing schedule and project end-dates if not fixed immediately. Affected component cannot go into production without this defect resolved.
2 – Serious	Defects that occur in a major function and are grossly wrong. In testing, the entire application component or function will not work. There is a bypass available, and some testing can continue. In acceptance testing, defects that invalidate high or medium priority test cases but the end-date impact is manageable. Component can go into production, but requires manual user workaround.
3 – Moderate	Defects that occur in a function but progress can continue. In testing, the function tested will not perform as expected but testing can continue. In acceptance testing, defects that invalidate low priority test cases but the end-date impact is manageable.
4 – Low	Defects are cosmetic, such as errors in format or spelling. ○ A defect in the work product under inspection which, if not fixed, would **not** cause a malfunction, would **not** prevent the attainment of a required result, and would **not** result in a Problem Report. ○ In testing, some of the function tested will not work as expected but testing can continue. ○ In acceptance testing, defects that do not invalidate test cases and do not affect end-date. ○ Severity 4 includes nice-to-have features and future enhancements
5-Very Low to None	

7. Test Schedule

7.1. Test Milestones

<The master project plan should list major test milestones>

7.2. Schedule

<Refer to the location of the test schedule>

Appendix A: Test Entrance Criteria

<Document the test entrance criteria>

Appendix B: Test Exit Criteria

<Document the test exit criteria>