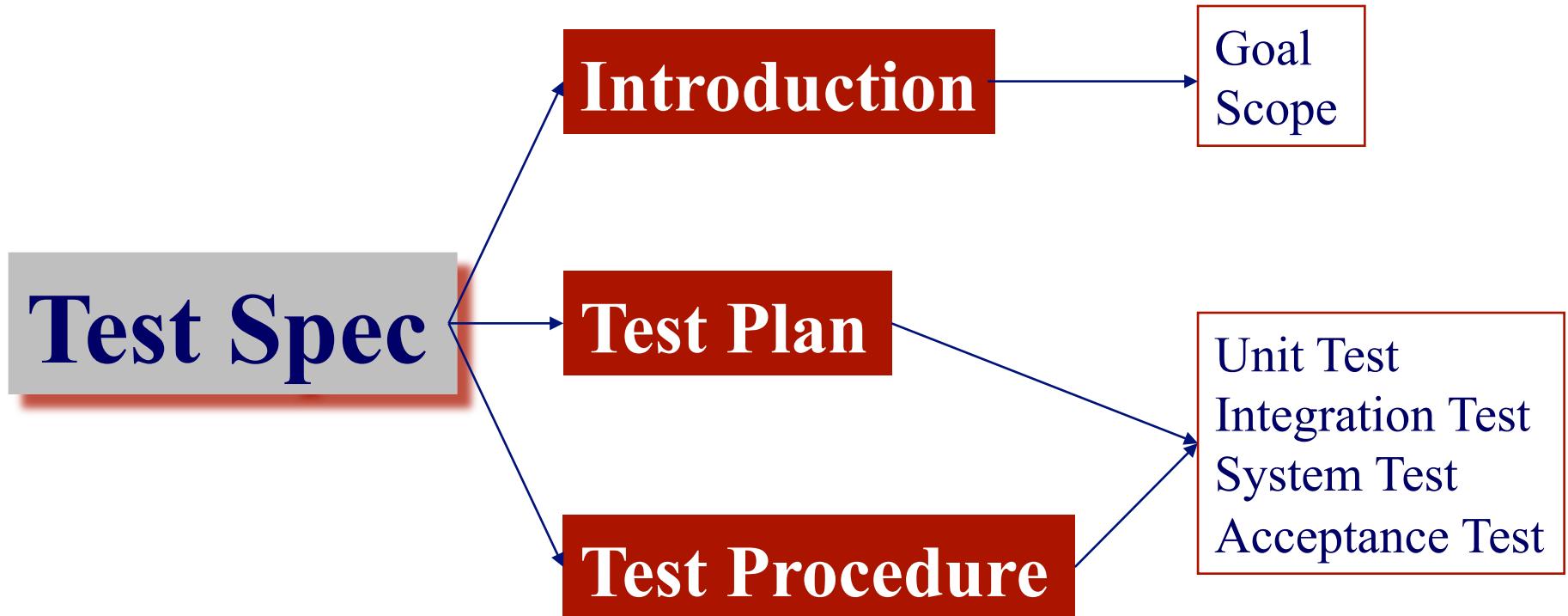


# **Software Test Plan**

# What Must be Included?



# Test Plan – Major Items

## 1.0 Introduction

This section provides an overview of the entire test plan document. This document describes both the test plan and test procedures.

### 1.1 Goals and objectives

Overall goals and objectives of the test process are described.

### 1.2 Statement of scope

A description of the scope of software testing. Functionality/features/behavior to be tested is described. In addition any functionality/features/behavior that is not to be tested is also noted.

# Test Plan – Major Items

## 1.3 Major Constraints

Any business, product line, or technical constraints that will impact the manner in which the software is to be tested are noted here.

## 2.0 Test Plan

This section describes the overall testing strategy and project management issues that are required to properly execute effective tests.

### 2.1 Software to be tested

The software to be tested are identified by name. Exclusions are noted explicitly.

### 2.2 Testing Strategy

The overall strategy for software testing is described.

# Test Plan – Major Items

## 2.2.1 Unit Testing

The strategy for unit testing is described. This includes an indication of the components that will undergo unit tests or the criteria to be used to select components for unit test. Test cases are NOT included here.

## 2.2.2 Integration Testing

The integration testing strategy is specified. This section includes a discussion of the order of integration by software function. Test cases are not included here.

## 2.2.3 System Testing

The system testing strategy is specified.

## 2.2.4 Acceptance Testing

The validation testing strategy is specified. This section includes a discussion of the order of validation by software function. Test cases are NOT included here.

# Test Plan – Major Items

## 3.0 Test Procedure

This section describes the detailed test procedure including test tactics and test cases for the software.

### 3.1 Software to be tested

The software to be tested are identified by name. Exclusions are noted explicitly.

### 3.2 Testing Procedure

The overall procedure for testing is described.

#### 3.2.1 Unit Test cases

The procedure for unit testing is described for each software component (that will be unit tested). This section is repeated for all components (i).

##### 3.2.1.1 Stubs/Drivers for component i

# Test Plan – Major Items

3.2.2 Integration Testing (The integration testing procedure is specified)

3.2.2.1 Testing Procedure for integration

3.2.2.2 Stubs/Drivers required

3.2.2.3 Test cases and their purpose

3.2.2.4 Expected results

3.2.3 System Testing (The system testing procedure is specified)

3.2.3.1 Testing Procedure

3.2.3.2 Test cases and their purpose

3.2.3.3 Expected results

3.2.4 Acceptance Testing (The acceptance testing procedure is specified)

3.2.4.1 Testing Procedure

3.2.4.2 Test cases and their purpose

3.2.4.3 Expected results

# **Case Study**

## **Waste Management Inspection Tracking System (WMITS)**

### **1.1 Goals and Objectives**

The main purpose of WMITS is to help automate the entire process that the Department of Environmental Quality (DEQ) Waste Management Division (WMD) staff members perform throughout an inspection.

# The goals of WMITS are:

- To minimize the time span of any inspection
- To minimize the amount of paper work required
- To provide a searchable database of all past inspections
- To provide an automated channel for the public to request information (under Freedom of Information Act)

# System Context

Eventually, multiple users will be using the product simultaneously. Therefore, concurrent connection will be an issue for implementation. In addition, this is a pilot product that hopefully, if successful, can be used in other locations as well. This leads to issues about future support for a large user base.

# **1.0 Introduction**

This section gives a general overview of the Test Specification for the Waste Management Inspection Tracking System (WMITS).

# 1.1 Goals and Objectives

Put it in a simple way, a good product will work perfectly, doing the right thing at the right time. To do that, the software has to go through a series of tests before its final release. Error free software is extremely difficult to achieve. After all, nothing is perfect. Especially for software developed in a short time frame. But high quality can be achieved with a detailed test specification. **All (or at least most) of the test cases will be listed**, the development team will follow it step by step, item by item, to test all the necessary objects, data flows, limits, boundaries, and constraints of the software.

# 1.1 Goals and Objects

Company XYZ would like to have a test specification to counter any difficulties that may impact the development and the future performance of the software. The team's goal is to assist the project team in developing a strategy to deal with any errors. For this, the team will take a look at the most common errors to some very uncommon errors as well.

# 1.3 Major Constraints

In this section we will talk about the business, technical or resource related constraints that may keep us from performing all tests necessary.

1. The team has limitation on time to test the product at the client's facility. We have access to the facility only during the regular office hours. We also have to set us schedule around the available time of the inspector that is to help us, so time schedule will be a major constraint when we talk about testing at the site.

## 1.3 Major Constraints

2. The team also has got funding for only one hand-held device. This means that we cannot test the software using additional devices from some other brand or that is of lesser price and lower hardware.
3. The team does not know any hacker that can help us test the security problems. So we have to rely on our own knowledge and have to trust the software for security.

## 1.3 Major Constraints

4. The team also does not have large enough group to have many people use the applications at the same time to perform real stress related testing. So we have will not be able to test the product for the larger user base.

Critique: Each of these constraints represents a significant product quality risk. The team should consider risk mitigation strategies.

## 2.0 Testing Plan

We want the product to be bug free. We also want to make sure that there are no defects in the product. So we will be spending large amount of the total software development time on the testing. Below is the description of the testing procedure and strategy. We will also be presenting the timing and schedule of the tests to be carried out.

### 2.1 Software to be tested

## 2.1.1 Interfaces

### Login Window

We will make use of several different names to log in to the system, so will be testing login window. We will also test OK and Cancel buttons on this window by performing test above.

### DEQ – Microsoft Visual Basic [Design] Window

This is the main window that we will use to access the database using Visual Basic. We will have several different drop-down menus in this window. File, Facility, Inspection, Approve, Reports, Maintenance and Help are the drop-down menus that will be available in this window. We will use all the menus and different options available in each of them.

## 2.2 Testing Strategy

In the following section we will describe the testing strategy. We will user four different methods to test our product.

## 2.2.1 Unit Testing

In the unit test cases we will be testing individual modules of the software. We will carry out white box testing where each module or component of the software is tested individually. We will test the components by passing data through them and monitor the data to find the errors.

We will be looking at the entry and exit conditions of the data. We will make sure that all the components work without any problems. Test will primarily be carried out by the programmer who designed and implemented the module. Lead tester will then carry out tests on the modules to finalize the testing results.

## 2.2.2 Integration Testing

In this method of testing we will deploy the software on the test server and run it. We will be looking for any problems with ....

## 2.2.3 Acceptance Testing

In this method of testing, we will be working with the customer to find out if the software developed is valid for the clients. We want to make sure that the clients are getting what they asked for. We will look at the software requirement document in case of conflict or misunderstanding with client regarding software components.

We will perform the black box testing where the software implementation is completed and we test all modules together. We will use several input data sets or test data and corresponding expected output. We will test the system with these data sets and tabulate results produced by the software. We will compare the output from the software with the expected outputs thereby validating the software.

## 2.2.3 Acceptance Testing

In case there are problems with the software we will create a deficiency list with details of the issues encountered. We will test all components and subcomponents of the software as part of validation testing.

We have and will try our best so that we don't have to create a deficiency list. This is necessary because if the errors are found at this stage of the software development we cannot fix them by deliverance due date. In this case we have to negotiate with the customer for an extension on the project.

## 2.2.4 System Testing

Under system testing, we will test for several different conditions....

### Recovery testing

Here we are concerned with the ability of the software to retrieve lost data. We want to make sure that the software is fault tolerant and does not lose data in case of system shutdown or if the system ceases.

## 2.2.4 System Testing

### Security Testing

In this method of test, we want to make sure that the security checks are working and no one is able to tamper with data. This is crucial since our software is designed to track ...

### Stress Testing

In this test method, we want to monitor the stress caused to the software as a result of simultaneous user requests. We want to make sure that the system does not crash under extreme use conditions.

### Performance Testing

Performance bounds are set during the design phase of software development. These bounds will help us in determining the effectiveness of the software....

## 2.6 Test Schedule

Following is the tentative schedule for the testing the WMITS system

- Creation of Project Test Plan: xx/yy/2013 – xx/yy/2013
- Unit & Integration Testing: xx/yy/2013 – xx/yy/2013
- System Testing: xx/yy/2013 – xx/yy/2013
- Acceptance Testing: xx/yy/2013 – xx/yy/2013
- Generate Test Report: xx/yy/2013 – xx/yy/2013

# Testing Resources and Staffing

We need a large number of resources available to us in order to test the entire software thoroughly. We will use help from several different people.

## Resources:

We will take help of the DEQ staff of the Waste Management Division to help us test the product. We are allowed to use DEQ staff member(s) to test the product as part of acceptance testing. We will have the DEQ staff record any errors found in the software and will correct them before the delivery of the software.

# Bug Resource Reports

We will use Bug Resource Report where we will identify the bugs found during testing and try to identify reasons for their occurrence. This will help...

# Staffing

We have decided to use simple method for staffing. Each programmer will test the components or functions developed by him/her and hand it over to the lead tester. Lead tester will test each component and will make a note of the result in test result table. Once the product is completely developed all members of the software project team will test the software as a combined effort. DEQ staff will also assist in testing the software.

# Test Record Keeping and Log

We will use a table created in excel to log all the test cases, describe them, and record the results of the tests. Below is the example of such table. The table will also be our test work product.

	A	B	C
1	<b>Test No.</b>	<b>Description of Test</b>	<b>Results</b>
2	1	Time: Accuracy of the time	Time function works. Accurate time has been displayed.
3	2	Date: Accuracy of the date	Date function works: Accurate date, day and month has been displayed.
4	3	If the Version/Control # is correctly inserted in the form?	The Version/Control # is correctly inserted into the form.
5	4	If the help button "?" for the above works?	The help button brings up correct help data on the topic above.
6	5	If the document name is correctly inserted in the form?	The document name is correctly inserted in the form.
7	6	If the help button "?" for the above works?	The help button brings up correct help data on the topic above.
8	7	If the page number is correctly inserted in the form?	The page number is correctly inserted in the form.
9	8	If the help button "?" for the above works?	The help button brings up correct help data on the topic above.
10	9	If the first name is correctly inserted in the form?	The first name is correctly inserted in the form.
11	10	If the help button "?" for the above works?	The help button brings up correct help data on the topic above.