# JagTrack
# Test Guidelines

## Version 1.0

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 27/04/12 | 1.0 | Initial version | RUIJIE YUAN |
| | | | |
| | | | |
| | | | |

# Table of Contents

1. Introduction 4

Testing is intended to show that a program does what it is intended to do and to discover program defects before it is put into use. 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Goals of Testing 4

3. Testing Standards 4

4. Key Measures 5

5. Test Completion Criteria 5

6. Defect Management Guideline 5

7. Change Management Criteria 5

# Test Guidelines

## 1. Introduction

Testing is intended to show that a program does what it is intended to do and to discover program defects before it is put into use.

Testing is part of a broader process of software verification and validation.

### 1.1 Purpose

Reflect previous experience of the kind of errors that programmers often make when developing components/

### 1.2 Scope

The scope of this specification is a set of requirements for Test Materials that, if satisfied, will enhance the usability and usefulness of the test materials. It covers the analysis and coverage of specifications, the prioritization and management of test cases, test frameworks and result reporting.

### 1.3 Definitions, Acronyms, and Abbreviations

- Requirements specification
- Design specification
- User guide
- Operations guide
- Installation guide

### 1.4 References

Software engineering
9 edition
LAN Sommerville

### 1.5 Overview

The testing guidelines include goals of testing, testing standards, key measures, defect management and completion criteria.

## 2. Goals of Testing

1. To demonstrate to the developer and the customer that the software meets its requirements. For custom software, this means that there should be at least one test for every requirement in the requirements document, for generic software products, it means that there should be tests for all of the system features, plus combinations of these features that will be incorporated in the product release.

2. To discover situations in which the behavior of the software is incorrect, undesirable, or does not conform to its specification. These are a consequence of software defects. Defect testing is concerned with rooting out undesirable system behavior such as system crashes, unwanted interactions with other systems, incorrect computations, and data corruption.

## 3. Testing Standards

1. Define the expected output or result.
2. Inspect the results of each test completely.
3. Include test cases for invalid or unexpected conditions.
4. Test the program to see if it does what it is not supposed to do as well as what it is supposed to do.
5. Avoid disposable test cases unless the program itself is disposable.

6. Do not plan tests assuming that no errors will be found.

7. The probability of locating more errors in any one module is directly proportional to the number of errors already found in that module.

8. If there exists critical sections in the program, the testing sequence should be designed such that they should be tested as early as possible.

## 4. Key Measures

1. White-box testing: White box testing based on detailed knowledge of the internal design and code. Tests are performed for specific code statements and coding styles. It focuses on the independent logical units of the software to give surety that all logical paths and code statements have been tested and are working properly.

2. Black-box testing: In black box testing you don't need to know the internal design in detail or have a good knowledge about the code for this test. It's mainly based on functionality and specifications, requirements. it focuses on the functional unit to assure that the given input will produce results which is actually accepted by software. Both mentioned above strategies should be used, as per the levels of testing. There are five levels of testing, which is carried out in chronological order and each has a specific functional purpose.

## 5. Test Completion Criteria

The initial run of tests on any given module will be verified by one of the test team personnel. After these tests are verified as correct, they will be archived and used as an oracle for automatic verification for additional or regression testing. As an example, when testing the Control Flow Graph module, an output is deemed correct if the module outputs the correct set of nodes and edges for a particular input C program fragment.

- All the test cases are tested at least once.
- All the critical defects are addressed and verified.
- All the unsolved defects are analyzed and marked with necessary comments / status.
- In the last iteration of testing, there are no critical defects reported.

## 6. Defect Management Guideline

1. Test software with sequences that have only a single value. Programmers naturally think of sequences as made up of several values and sometimes they embed this assumption in their programs. Consequently, if presented with a single value sequence, a program may not work properly.

2. Use different sequences of different sizes in different tests. This decreases the chances that a program with defects will accidentally produce a correct output because of some accidental characteristics of the input.

3. Derive tests so that the first, middle, and last elements of the sequence are accessed. This approach is reveals problem at partition boundaries.

## 7. Change Management Criteria

*No Change Management Criteria is currently available for the project.*