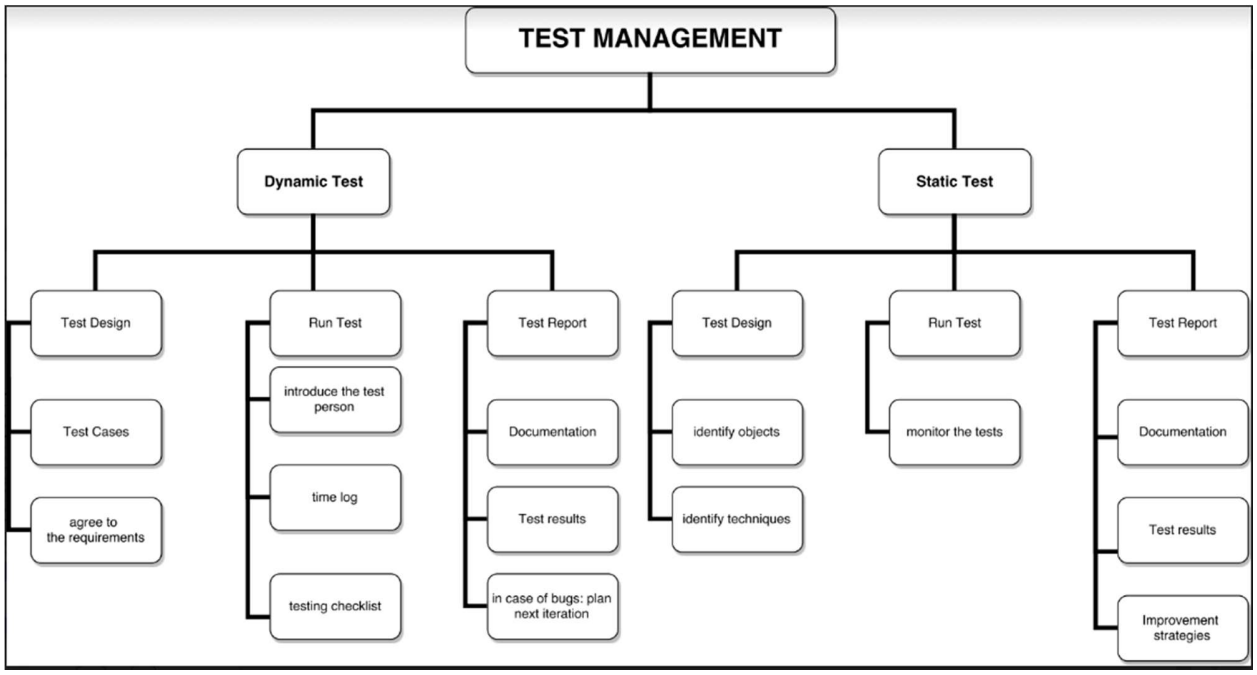
Test Plan



Introduction

Software Testing is important for knowing if the program works the way it is supposed to work and to find the faults such as defects, errors, flaws and bugs. This project is about developing a server for a library client. This paper describes different standards for system testing of the specified client application.

Objectives

This Test Plan document supports the following objectives:

Identify existing project information and the features that should be tested.
List the recommended test requirements.
Identify the required resources and provide an estimate of the test efforts.
List the deliverable elements of the test activities.

Purpose

The main purpose is to describe the various testing strategies and tools used for the whole System testing of this project.

Scope

The resources and objects will be tested mainly for correct functionality.

To be tested:

- implemented resources
- all Methods
- objects Book and Catalog
- booksDAO class

Testing Process

- Creating Test cases for every Use case we have and continually adding Test cases
When adding new features and create new Use cases.
- Creating Test Data (objects Book and Catalog) so that the test cases can be tested individually.
- Executing Test Cases when the server is reloaded. In case of bugs, an appropriate error massage will be displayed in the console.

Test Strategies

Unit test

To be tested: both Objects Book and Catalog and the class booksDAO.

API tests

All resources that are found in the API specification will be tested, to see whether the Communication between client and server is working as expected.

Tools

JUnit library + dropwizard-testing module for Java.

Test Schedule

Planning:

The Test Plan document has to be finished.

Design Phase:

Test cases need to be written.

Code Complete function:

Unit and API testing needs to be finished when functionality is implemented.

Feature Complete:

When all features have been implemented, last faults should be found and solved and re-test.