QA Test Strategy

**Springer Nature**

Author: Shravani Palivela

Date: 01-03-2020

Version: 1.0

# Introduction

This document defines the testing approach, what needs to be accomplished and how to achieve it. This document should be discussed with the whole team so that the team is consistent with the approach and responsibilities.

## Purpose

The purpose of this document is to understand the exact process/strategy that needs to be followed while testing the application.

## Objective

The objectives of this document are Testing scope, Test coverage, Testing approach, Test Stategies to accomplish all levels of testing .

## Underlying Testing Principles

The following testing principles will be adopted:

1. Early testing
2. Exhaustive Testing is Not Possible
3. Testing is Context-Dependent
4. Defect Clustering
5. Pesticide Paradox
6. Testing Shows the Presence of Defects
7. Absence of Error

# Test Strategy

## Testing Methodology

Testing Methodology used is agile methodology and continuous testing. This methodology involves a process of testing early, testing often, test everywhere, and automate. This strategy of evaluating quality at every step is an integral part of Continuous Delivery Process

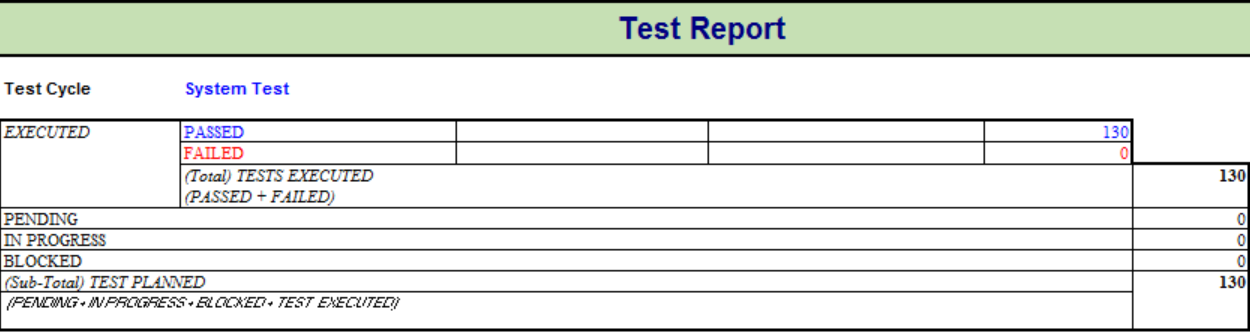
## Develop Test Plans

Test plan is derived from the test strategy document and contains list of tasks and milestones in a baseline plan to track the progress of the project This helps us determine the total QA effort required for the sprint.

## Test Design & Preparation

For Test Design,**Test Rail or QC** can be used. Attached is the template that can be used for designing test cases.

## Test Summary Report



# Testing Types

Since our objective is continuous delivery , the levels of testing that should be targeted are **unit testing,Integration testing ,System Testing and Acceptance testing.**

When the requirements are tagged to a sprint for release by the Product owner, **Static testing** on the requirements/**Jiras** need to be executed by the team to identify the missing user stories, acceptance criteria and other edge case scenarios. When a bug is identified in any level of testing , all the levels of testing should be performed again to release the build to production.

## Unit / Component Testing

Each requirement is developed on a feature branch and are not directly committed to the develop or master branch.When the development is complete on the feature branch , automated unit tests should be executed to determine early defects. Unit tests should be executed till all the tests pass. **QUALITY GATES(like code coverage,vulnerabilities, code smells and technical debt)** can be configured to determine if build should pass or fail. **CODE REVIEWS** should be performed by Peers to identify code level issues.

## Integration Testing

Incremental integration testing should be the focus at this stage. Execute **AUTOMATED SMOKE TESTS** and **UNIT TESTS** which cover the major functionality of the application every time an a feature branch is integrated to the develop branch. Smoke tests can be **FUNCTIONAL TESTS or API TESTS** to find the issues with integration of modules.**LOAD TESTS** can also be executed at API LEVEL to validate the performance and response times.All these tests should be executed till the final build is ready and all the release components are integrated.

**3.3 System Testing**

**AUTOMATED REGRESSION TESTS** should be scheduled after the smoke tests to validate the system functionality**. AUTOMATED END TO END FUNCTIONAL TESTS/UI TESTS/DB TESTS** should be execute to validate the business flows. **AUTOMATED** **LOAD TESTS,STRESS TESTS, ENDURANCE TESTS** should be executed and **LOAD TIMES, CPU USAGE, MEMORY USAGE** should be captured to validate performance at system level. **AUTOMATED SECURITY TESTS** should be executed at this stage to uncover vulnerabilities, threats, risks in the application. **ACCESSIBILITY TESTING(AUTOMATED TESTS)** should be performed if the application has features implemented . **EXPLORATORY TESTING(MANUAL TESTS)** should be done at this stage.All the tests should be executed till the build is production ready.

**3.4 Acceptance Testing**

All the tests at this stage are **FUNCTIONAL TESTS(MANUAL TESTS)** executed by Product Owner, Users,Customers and stakeholders with the help of testers. Validations are performed on production environments after deployment. Mostly positive tests are executed to make sure the functionality meets the business requirements.

# 

# 4.Automated Testing Tools

## 4.1 Test Management

### 4.1.1 Test Management Tool

**QC or Test rail** can be used as a test management tool to track all the testing activities from test planning to test closure.

### 4.1.2 Defect Management Tool

**JIRA or QC or Test Rail** can be used as defect management tool to track the bugs till they are fixed and closed.

## 4.2 Test Automation Tools

## 4.2.1 Test Automation Tool

**UNIT TESTS** - Junits,JAVA/JEST,Mocha,JavaScript

**ACCEPTANCE TESTS** - Cucumber, Selenium,JavaScript/JAVA,

**E2E Testing UI Automation Tool** - Cypress, JavaScript and Mocha

**API Testing** - RestAssured/SOAPUI, Postman

### 4.2.2 Performance Testing Tool

L**OAD TESTS** - Jmeter,Blazemeter

**MONITORING TESTS** - Dynatrace,Splunk

**4.3.3 Continuous Testing Tool**

**DEPENDENCY MANAGEMENT -** Gradle, Maven

**BUILD MANAGEMENT-** Jenkins,CircleCI

**CLOUD TESTING PLATFORM -** Saucelabs

# 

# 5.Test Environment Strategy

As there is no staging or preprod environment , we can deploy our web application using Paas solutions like AWS Elastic Beanstalk, Heroku ,Google App engine etc. We should use solutions like **AWS Elastic Beanstalk which** supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby.