LIBRARY MANAGEMENT SYSTEM

Test plan template:

Library management system

Prepared by:

Indra Kumar Korupolu

Date: 28/06/2022

TABLE OF CONTENTS

1. Introduction
2. Assumptions
3. Analysis of fundamental test process
4. Test activities and tooling

- Description of test activities within test process

- Evaluation and reasoning of possible tool support including

- Implementation of prototypes for proposed tools

5. Limitations, possible interfacing and integration

6. Cost/ benefit criteria

7. Approvals

1.INTRODUCTION

The primary goal is to evaluate and to propose a set of tools which can support most of the test activities of the project. The paper ends with cost/benefit considerations and possibilities for integration.

The system under test (Sut) is a library management system which provides a web-interface to access the library.

2. ASSUMPTIONS

\* The SUT technology stack

\* The SUT development environment

\* System architecture

\* Customer

\* Software testing experience

\* Testing tools

3. ANALYSIS OF FUNDAMENTAL TEST PROCESS

3.1. Testing process versus software development process

Our test approach for this project shall follow the fundamental test process of the international software testing qualifications board. The process is a generic approach and will be discussed in the following sections in more detail.

Important to note is the fact that the testing process is heavily dependent on the software development process.

3.2. STANDARDS

Before we elaborate on the generic test process itself, it is worth to discuss with the customer the use of standards. This can be achieved to multiple levels.

1. Company standards
2. Best practices
3. Standards for particular industrial sectors
4. Software testing standards

3.3. THE FUNDAMENTAL TEST PROCESS

The testing effort ideally decreases over time due to a sleep learning curve. The preparation and installation for a initial release is much higher compared to the subsequent releases.

3.4. TEST PLANNING AND CONTROL

At the same time a control process escorts all sequence of the plan to allow target-performance comparison. This will lead to ongoing adaptations and improvements.

3.5. TEST ANALYSIS AND DESIGN

\* review the test basis

\* identify test conditions

\* Design the tests

\* evaluate testability of the requirements and systems

\* design the test environment setup and identify required infrastructure and tools

3.6. TEST IMPLEMENTATION AND EXECUTION

\* develop and prioritize test cases by using techniques and create test data for those tests

\* automate some tests using test harness and automated test scripts

\* implement and verify the environment

\* compare actual results with expected results and report discrepancies as incidents

4.TEST ACTIVITIES AND TOOLING

Before we focus on the tools proposed in the specific test levels

- component testing

- integration testing

- system testing

- acceptance testing

**MS Project**

Issued for overall resource planning and work breakdown structure. This means we derive the primary planning information from a MS project plan which is maintained by the overall project manager.

**OpenOffice**

Shall be the tool to create the documentation which cannot be derived

automatically from any testing tool.

This includes test policy, test strategy documents, test design.

4.1. Requirements workspace: the module which supports

- documentation of determined scope and risks including objectives of testing

- requirements can be linked to test cases and test steps can be linked to requirements

- requirements workflow, versioning, logging, printing

- requirements tree (bulk editing, export)

4.2. Test cases workspace: the module which supports

- development and prioritization of test cases

- writing of instructions for carrying out the tests (test procedures)

- creation of test suites from the test cases for efficient test execution (test scenarios)

- test case library

4.3. Campaign workspace: the module which supports

- monitoring and documentation of progress, test coverage and exit criteria

- Execution of test suites and individual test cases following the test procedures

- Re-execute the tests that previously failed in order to confirm a fix

- log the outcome of the test execution and record the identities and versions of the SUT

- compare actual results with expected results and report discrepancies as incidents.

4.4. Management workspace: the module which supports

- measurement and analysis of the results of reviews and testing

- reporting libraries

- custom graphs, dashboards, with wizards

- standard dashboards

* 1. TESTING TYPES

1. Component testing
2. Integrating testing
3. System testing
4. Acceptance testing

5. LIMITATIONS ,POSSIBLE INTERFACING INTEGRATION

Direct class path example

Go to file->project structure

Now select modules and then dependencies tab

Click the + icon and select library

Click new jars or directories and select the jar file

And finally click add selected

6.COST/BENEFIT CRITERIA

This chapter provides cost and licensing information for proposed test tooling. Here we focus on only cash out positions related to the licensing of the software products. The costs needed für consultancy, training, implementation or customization effort is not considered. It is assumed that this know how for the project comes with the testing team.

We do not analyse the benefits of the proposed tooling. We assume there will be further development of the SUT which will result in many succeeding versions. Hence the introduced tooling and the possibility for automated regression testing can easily show a positive business case.

7.APPROVALS

NAME (in capital letters) Signature Date

THE END

Background pattern

Description automatically generated with low confidence

Background pattern

Description automatically generated with medium confidence