GROUP-7

Project Test Plan

**Members : Janipan Sivaguru**

**1. Introduction**

* 1. Test Plan Objectives

Developing a program for a local delivery company with three trucks operating three distinct routes requires a very practical approach, which is what this project entails. Square grids, each measuring 25 by 25, have been added to the city map. Every truck has the same capacity, holding up to 1000 kg of cargo and 36 cubic meters of boxes.

The following tasks are requested of us by the program: When a shipment arrives, locate a truck large enough to accommodate it and one that will deliver the package as near to its destination as feasible. To deliver the package, the trucks are allowed to stray a little from their designated routes and drive through any area of the grid that is white.

After finishing the coding portion of the program, we are meant to proceed to the testing phase, where we will utilize various testing and debugging techniques to ensure that the program is error-free. A number of tests, including integration, acceptance, white-box, and black-box tests, will be conducted. The goal of the entire project is to equip us with practical knowledge on how to test software in a formal team setting.

**2. Scope**

Testing each and every code piece is part of the project's entire scope. We haven't tested the code that was previously provided to us. I'll be incorporating additional code into the program, which also requires testing. After the coding is finished, we will run through a battery of tests that incorporate most, if not all, of the testing techniques we have already studied.

**3. Test Strategy**

The testing phase will be the most challenging phase of the project as the code will go through many types of tests. Here is a list of different type of test which will be performed:

(1) System Test

(2) Performance Test

(3) Security Test

(4) Automated Test

(5) Stress and Volume Test

(6) Recovery Test

(7) Documentation Test

(8) Beta Test

(9) User Acceptance Test

**4. Environment Requirements**

The precise hardware, software, and network configurations required to carry out software testing efficiently are referred to as environment requirements. These specifications are required to guarantee that the testing environment faithfully replicates the end-user environment where the program will be installed.

The following environments must be met for this project:

Operating system: Various operating systems, including Windows, MacOS, and Linux, must be used to test the software.

Network: Various network environments, including LAN, WAN, and mobile networks, must be used to test the software.

Software testers can precisely detect and fix any problems that may occur before the program is made available to end users by making sure the testing environment satisfies all prerequisites.

Depending on the project's scope, it might not be able to develop and test in every one of the environments, but for the majority of software testing projects, all of the aforementioned requirements are applicable.

**5. Execution Strategy**

The plan for carrying out software tests is called the execution strategy. It contains the parameters for the beginning and ending times of tests. If a test successfully completes 95% of the test cases or if no serious or critical flaws are discovered during testing, it may be deemed complete.

If there are no bugs, or if there are 99% or less, the test is deemed finished for the purposes of this project.

A full set of test cases and a stable environment in which to conduct the testing are prerequisites for testing. Reaching the pass rates or fixing every major flaw before the testing is finished are the exit criteria.

The execution plan assists in guaranteeing that the software is consistently and thoroughly tested, and that any bugs are found and fixed prior to the software being made available to end users.

The found bugs' error severity levels can be arranged in ascending order as follows: cosmetic, low, medium, high, and critical. Below is a brief definition for each term: **critical** which cause the system to crash or produce anomalous results,

* + 1. **high** which causes lack of program functionality and might have a work around,
    2. **medium** which is a bug which D crates degrades the quality of a system but often has a work around to give the desired functionality.
    3. **Low** which might be an unclear error message or some other minor error that has minimum impact on functionality
    4. **Cosmetic**s is something that makes the user interface less than optimal but still perfectly functional.
  1. **Test Reporting**

Test reporting refers to the process of documenting and communicating the results of software testing. This includes what reports should be produced, how often they should be produced, who should receive them, and what information they should contain.

In this project, the testing part will be divided among all group members, and everyone will perform each test in their environment. All the tests’ reports will be submitted to the group leaders who will further check the test reports thoroughly and document each and every thing professionally. The report will include all the tests that are passed as well as all the test which failed with a detailed description of what made them to pass or fail.

A group discussion if responses from group members are received, we will be done with the tests which failed so that we can figure out answers of the following questions :

What is the exact issue?

Where is the error?

How can it be fixed?

What did we learn from it?

What needs to be corrected?

This group discussion will help all the members to learn and work effectively while putting the best team efforts to deliver a high quality program to the end user.

**6. Test Schedule**

|  |  |
| --- | --- |
| **WEEK-1** | **Repositories and Jira Account Setup by one team member [Completed]** |
| **WEEK-2** | **Create complete Project Plan and add new Data Structures to the program. [Completed]** |
| **WEEK-3** | **Design required functions and its specifications. Start designing the Black box tests specifications as well.[Completed]** |
| **WEEK-4** | **Complete the implementations of the functions and perform Black Box tests. [** |
| **WEEK-5** | **Create, Implement and Execute Integration Tests** |
| **WEEK-6** | **Complete all test cases** |

**7. Control Procedures**

A crucial component of software testing that contributes to its effectiveness and efficiency are control procedures. Software testing frequently uses a variety of control procedures, such as the following:

Reviews: Every week, the group will get together to discuss and revise any modifications made to the test plan.

Bug Review Meetings: To address the identified bugs, we will hold meetings. These meetings are intended to pinpoint the defect's primary cause and choose the best course of action for fixing it.

Change Request: Following deliberation in the group meeting, it will be determined whether any changes to the project are required.

Defect Reporting: In this step, any flaws found during testing will be tracked down and documented. This entails gathering data regarding the flaw, such as its impact and severity, and monitoring its progress until it is fixed.

Our team will make sure that the software satisfies the required quality standards and that errors are found and fixed promptly by putting these control procedures into practice.

**8. Functions To Be Tested**

We've already been given a lot of the functions, and our group will create a lot more. They must all be put to the test in every manner imaginable. Every function will be tested in accordance with the tests listed in the testing strategy above.

**9. Resources and Responsibilities**

The personnel, tools, and equipment needed to carry out testing tasks are referred to as resources. Tasks, roles, and activities that individuals or teams are in charge of during the testing process are referred to as responsibilities.

The following are some of the tools and duties that our team will be using to test this project:

**Resources:**

(1) Testing tools such as test management software, automated testing tools, and defect tracking software.

(2) Test data such as sample data.

**Responsibilities:**

(1) Test planning, including creating test plans, test cases, and test scenarios.

(2) Test execution, including running test cases and recording test results.

**Test management, including managing the testing process, coordinating with development teams, and ensuring that testing is completed on schedule :**

(1) Test automation, including creating and maintaining automated test scripts and frameworks.

(2) Test analysis, including analysing test results to identify trends, patterns, and areas for improvement.

(3) Test environment management, including ensuring that the test environment is set up correctly and maintained throughout the testing process.

(4) Test data management, including managing and creating test data and ensuring that it is accurate and up-to-date.

**10. Deliverables**

We need to deliver the following at the end of the project delivery date:

(1) Bug free Source Code

(2) Documentation (group contract form, test plan etc)

(3) Test Cases, Obtained result

**11. Suspension / Exit Criteria**

Exit criteria are the conditions that must be met before the testing phase can be completed and the software can be released. These criteria include:

>>> The completion of all planned test cases and the achievement of predetermined test objectives.

>>> The resolution of all critical defects and the confirmation that all other identified defects have been addressed satisfactorily.

**12. Resumption Criteria**

Resumption criteria in software testing refer to the set of conditions or requirements that must be met to resume testing after it has been suspended. These criteria are established to ensure that testing can be resumed effectively and that the testing objectives can still be achieved.

Resumption criteria may vary depending on the reason for the suspension, but they generally include:

(1) Resolution of the issue that caused the suspension: Any issues that caused the suspension of testing, such as hardware or software problems, must be resolved before testing can resume.

(2) Restoration of the testing environment: The testing environment must be restored to its original state or reconfigured as necessary before testing can resume.

(3) Re-evaluation of the test plan: The test plan must be reviewed to ensure that it is still valid, and that any changes or adjustments needed to reflect the suspension have been made.

Resumption criteria are important to ensure that testing can be resumed effectively and that the results obtained after the suspension are reliable.

Top of Form

**13. Dependencies**

Dependencies in software testing refer to the factors or elements that are necessary for testing activities to be completed successfully. These dependencies may be related to people, processes, tools, or infrastructure, and they can have a significant impact on the testing process.

Dependencies in this project are:

(1) Availability of test data: The testing process may require specific test data, such as user information or specific data sets, which need to be available before testing can begin.

**14. Risks**

There might be several risks that must be considered while testing this project:

(1) Inadequate test coverage: If the test coverage is insufficient, it may not be possible to identify all defects, and the software may not meet the quality requirements.

(2) Inaccurate requirements: If the requirements are inaccurate, incomplete, or ambiguous, it may be difficult to design and execute effective tests.

(3) Unforeseen changes: If there are changes in the project scope, requirements, or schedule, it may be necessary to re-plan or adjust the testing process.

(4) Defect leakage: If defects are not identified and addressed effectively, they may be carried forward into subsequent phases, leading to increased costs and reduced quality.

(5) Technical challenges: If the software being tested is complex, or if there are challenges with the testing tools, infrastructure, or environment, it may be difficult to complete testing activities effectively.

Identifying and managing risks is an important part of the software testing process. Risk identification should be done early in the project planning phase, and a risk management plan should be developed to address potential risks. Top of Form

Bottom of Form

**15. Tools**

**These are the tools used for this project:**

**(1) Visual Studio**

**(2) GitHub**

**(3) Jira Software**

**(4) Tortoise Git**

**16. Documentation**

**Here is a complete list of documentations that are to be created and edited while working with this project :**

**(1) Group Contract Form**

**(2) Project Test Plan**

**(3) Test strategy template**

**(4) Traceability Matrix Template**

**(5) Test Report Analysis**

**17. Approvals**

**Awaiting approval per my professor.**