

Test Plan

1. Introduction

a. Test Plan Objectives

- i. The goal of this test plan is to make sure that the delivery correctly assigns packages to trucks. It does this by considering factors like the truck's capacity, how close it is to the destination, and the available route.

2. Scope

- a. Calculate the destination for the closest route
- b. Assign package to truck considering its weight and size
- c. Implement a system to notify or record cases where packages cannot be delivered on the same day.

3. Test Strategy

3.1. System Test: This is the last part of our software testing for the final milestone. This makes sure we meet the user's want and reduce the problems after release. In our project, we need to ensure the package will be delivered to the building I want to send to.

3.2. Performance Test: Checking how well software systems work in different situations, focusing on how fast and efficient they are.

3.3. Automated Test: Testing the program with the Native unit test tool in Visual Studio and test the program by designed test cases.

3.4. Stress and Volume Test: Keep inputting data to test the situation in which all the vans are fully loaded then the program is expected to display a message then stop.

3.5. Documentation Test: This test focuses on ensuring that the project documentation accurately reflects the project's goals and requirements.

3.6. User Acceptance Test: Test the application and ensure the information is valid, also working as expected to meet the user expectation.

3.7. Exploratory test

a. Testers come up with real-life situations and pretend to be users, trying out different things to see how the system handles various situations, functions, and tricky situations.

b. Testers concentrate on the most important areas that are more likely to have problems or significant impact, prioritizing their testing efforts accordingly. For example, package weight and size

Overview :

- a. You could describe the test design process and give an overview of how it will be conducted. You could provide a broad overview of
 - i. how to understand requirements,
 - ii. build a traceability matrix,

- iii. prepare test cases,
- iv. and have them reviewed by another member of the quality assurance team.

4. Environment Requirements

- a. Hardware: Both Windows and macOS requires computers for the test environment
- b. Software: Testing will be conducted on both Windows and macOS using the latest versions of their respective operating systems. Both of them required installing any relevant development tools, libraries, or compiler.
- c. Test harness: Utilize pre-existing testing tools, ensuring installation, proper configuration, and readiness for conducting tests on test machines.

5. Execution Strategy

- a. Entry criteria: The software code guarantees that everything runs smoothly.
Exit criteria: Tests are finished when all test scripts pass without any severe or critical issues remaining or when 90% of the tests pass it considers this program pass.
- b. severity levels:
 - i. **Critical:** Make the system crash or give unexpected results.
 - ii. **High:** Defects that lead to a loss of program functions, but there could be a workaround.
 - iii. **Medium:** Defects that lower the system's quality but have a way to achieve the desired function.
 - iv. **Low:** Minor errors with minimal impact, like unclear error messages
 - v. **Cosmetics:** Issues that are more about appearance than functionality.
- c. **Test Reporting**
 - i. Reports: We'll make reports to keep track of how testing is going.
 - ii. Frequency: We'll make reports every day to show how many tests we did, which ones worked, and which ones didn't. The reports will also say what parts we tested and what parts didn't do so well.
 - iii. Recipients: We'll send the reports to the project manager, the people making the software, and the quality assurance team.
 - iv. Communication: Testers will tell the project managers about any problems they find, and then the project managers will tell the developers to fix them. We'll talk a lot through meetings, emails, and systems to keep everyone on the same page.

6. Test Schedule

- a. Testing: It takes two to three weeks to test the code, to ensure the code works as expected.
- b. Complete time: It may expect complete all the test in the final week of the project.

7. Control Procedures

7.1 Review: Regularly scheduled reviews will be carried out to monitor testing progress and ensure the testing output is achieved with project requirements and objectives. Also, review can know the process of team, like the assignment for every member, ensuring the process is complete as expected. During the process, it may have some problem caused, so we can make some adjustments to solve the problem.

7.2 Bug Review Meetings: Scheduled meetings will discuss issues together. Testing and development teams can communicate about the bugs, and explain the reason why code is wrong, and think out the solutions.

7.3 Change Request: The change request happened after the meeting, the development team will receive the feedback and have the solution of the bug. They will request to change the code, and make sure of the accuracy of code, also get the approval from the whole member.

7.4 Defect Reporting: It is a job for the testing teams. They will find the bugs from the source code. Testers will document defects using a standardized format, including descriptions, steps to, expected vs. actual behavior, and attachments. Defects will be categorized by severity for prioritized resolution, enabling efficient issue tracking and resolution.

8. Functions To Be Tested

a. The closest route function:

Testing the calculation function calculates the route correctly between destinations, avoiding buildings. Test cases will cover various situations with different start place structured paths by buildings, and edge cases.

b. The capacity of the package function:

Testing will confirm accurate determination of available truck capacity. Test will cover situation with various weight and box size combinations, reaching maximum weight, and accommodating trucks with different constraints.

c. Shipment Allocation Function:

Testing will confirm accurate shipment allocation to trucks, accounting for weight, box size, and destination. Test cases will cover various situations like valid/invalid inputs, multiple truck availability, and capacity constraints.

9. Resources and Responsibilities

9.1. Resources:

A group of testers who will do the tests, write down what happens, and tell everyone if there are any problems or mistakes.

9.2. Responsibilities

The testers are responsible for doing the tests, writing down what happens, and letting everyone know if they find any problems or mistakes while testing.

10. Deliverables

- a. Make sure it puts packages in trucks in the right way.
- b. It should look at how much space is left in the truck, how far the destination is, and if there's any need to take a different route.
- c. Figure out the shortest path from where the truck is to where it's supposed to go. Show which truck is making the delivery, where it's going, and if it's taking a different route.

11. Suspension / Exit Criteria

- a. When the total weight of products is more than 1000 kilograms.
- b. The volume is over than 36 cubic meters
- c. The calculation of the closest route is not correct.
- d. The display information or error message doesn't show as expected.

12. Resumption Criteria

After fixing any problems or bugs, testers need to test the testing again to make sure it's doing what it's supposed to. We'll use different situations and tests to check if it works right and is accurate. If we make changes, we'll test it again to be sure we didn't accidentally make new problems.

13. Dependencies

13.1 Personnel Dependencies:

Assign every member's job and their responsibility to finish personal job.

13.2 Software Dependencies:

Every member uses different software to develop or test, and we need to ensure the code tested correctly, and maintain the accuracy and stability of data.

13.3 Hardware Dependencies:

The hardware is like computers, we use it to make sure it works successfully.

13.3 Test Data & Database:

The data should be suitable for the test and store the data at the correct position like the weight, volume, the route, also maintain the accuracy of data conversion and performance works as expected.

14. Risks

14.1. Schedule

To avoid schedule time risks, plan the project realistically and communicate clearly with team members about task priorities.

14.2. Technical

There are potential problems like network issues, limits with the computer hardware, and problems with the tools we use.

14.3. Management

See how not good work doing in managing the project can affect how it moves forward.

14.4. Personnel

It may have time conflict with teammates.

14.5 Requirements

It may have unclear or incorrect output; we should know the requirement and these potential problem will impact on the code

15. Tools

15.1 Testing the program with the debugger tool.

15.2 Testing the program by print statement.

15.3 Testing the program by the log files.

16.Documentation: Collect all available documentation related to the software. This is the first step a developer needs to take to fully understand what the project asks. Also, this is what we are doing for milestones 2.

17.Approvals: Approvals in a test plan refer to the formal sign-off process. It is ensuring the testing activities align with the project's goal and requirements. So, in our group we need to make every teammate agree before we release.