Test Plan Template

1. **Introduction**
   1. Test Plan Objectives
      1. **This section should describe the project that is being tested and what are the objectives for the test. You might give a quick overview of the project to tell people what it's about, and then describe the testing at a very high level and what do you expect to get as a result of the testing.**

**Answer:**

Test Plan Objectives

The goal of this testing plan is to ensure the local delivery company's package selection algorithm works successfully. This algorithm aims to optimize truck selection based on factors such as weight, container size, destination, and truck route.

1. **Scope**
   1. **In this section you can describe what will be tested and what will not be tested.**

**Answer:**

1. This pilot plan will include testing the bid selection algorithm, including validation of input parameters, truck selection process, diversion calculations, and handling of various scenarios.

It will not include the actual truck coordination or delivery process beyond truck selection for each package.

1. **Test Strategy**
   1. **This section describes the approach you will take to performing the tests. There are sections below where you can elaborate on different types of tests. Not all these types of tests will be in every project and some projects might have tests which are not listed below. This would be a good section to describe where the test data is being obtained from. You could also describe the different levels of testing which might be used. For example, testing is often broken up into exploratory testing which attempts to make sure that critical defects are removed before next level of testing begins. After exploratory testing catches some of the big critical defects you can go on to functional testing as the next testing cycle to make sure that all the prime functions of the application are being delivered correctly. You can continue to describe all the test deliverables and what roles are responsible for producing and delivering these. You could also include an estimate of how long it is going to take to do the testing.**  
      3.1. System Test  
      3.2. Performance Test  
      3.3. Security Test  
      3.4. Automated Test  
      3.5. Stress and Volume Test  
      3.6. Recovery Test  
      3.7. Documentation Test  
      3.8. Beta Test  
      3.9. User Acceptance Test

Answer:

This section outlines the testing approach, including different types of tests to be conducted.

* Functionality Test Cases: These tests ensure that the application's functions operate as intended. They validate user inputs, outputs, and interactions with various features.
* Unit Test Cases: Unit tests verify individual units or components of the code to ensure they perform correctly in isolation. They are typically conducted by developers during the development phase.
* Performance Test Cases: Performance tests assess the application's responsiveness, scalability, and stability under various load conditions. They measure factors such as response time, throughput, and resource utilization.
  1. **You could describe the test design process and give an overview of how it will be conducted. You could provide a broad overview of** 
     1. **how to understand requirements,**
     2. **build a traceability matrix,**
     3. **prepare test cases,**
     4. **and have them reviewed by another member of the quality assurance team.**

**Answer:**

**The test design process includes:**

* **Understanding Requirements: Analyzing project specifications to identify testable features and functionalities.**
* **Building a Traceability Matrix: Mapping test cases to requirements to ensure comprehensive test coverage.**
* **Preparing Test Cases: Developing detailed test scenarios and steps to validate functionality.**
* **Reviewing Test Cases: Conducting peer reviews within the quality assurance team to verify test completeness and accuracy.**

1. **Environment Requirements**
   1. This section will typically define the hardware and software environment necessary for the tests to be conducted. This could involve specifying that a test computer is necessary to run the tests in a continuous integration process or it might say that all testing is done on the developers workstations. Test harness is might need to be built to conduct the test or you might be using a pre-existing set of testing tools. All of this needs to be laid out with all its requirements so that the testing environment can be set up before the testing begins.

Answer:

a. Test Environment Setup

This section outlines the hardware and software requirements for testing:

Hardware:

* Test computers or workstations.
* Specific hardware devices if needed.

Software:

* Supported operating systems.
* Development tools, frameworks, and testing tools.
* Test harness, if required.
* Version control systems and repositories.
* Continuous integration tools, if applicable.
* Virtualization or containerization tools, if needed.

Setup Instructions:

* Provide step-by-step setup instructions.
* Include troubleshooting guidelines.

Configuration:

* Define environment settings.
* Specify test data or cases.

Validation:

* Procedures for validating the environment.
* Ensure readiness for testing.

This ensures proper setup before testing.

1. **Execution Strategy**
   1. this is the section where you will describe heavy chests are actually executed. You can describe what the entry and exit criteria for the tests are. For example you might be able to exit a test if it passes 95% of test scripts. In another situation, you might want to pass 100% of the tests. Or perhaps you want to declare but a test is completed if there are no severe or critical defects.
   2. You can describe the severity of defects in this section and break them down into severity levels of:
      1. **critical** which cause the system to crash or produce anomalous results,
      2. **high** which causes lack of program functionality and might have a work around,
      3. **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
      4. **Low** which might be an unclear error message or some other minor error that has minimum impact on functionality
      5. **Cosmetic** which is something that makes the user interface less than optimal but still perfectly functional.
   3. **Test Reporting**
      1. This action will describe what sort of reports should be produced as a result of testing, how often these reports should be produced, and to whom the reports should be sent. It should give some indication of the contents of the reports and under what conditions the reports are generated. You might say that a manager receives a daily report of the number of tests conducted, passed, and failed with a brief description of the areas being tested and the areas which are failing.
      2. This section could also have details of how the testers are going to feed information back to the project managers so that they can assign developers to fix the bugs. This section can detail the communication to occur between management, the development team, and the quality assurance team.
   4. You can also explain how the quality assurance team we'll be able to interact with the developers and how they will be able to work with the developers to resolve the defects found in the software.

**Answer:**

* 1. Test Execution & Criteria:
* Entry: Conditions to start testing.
* Exit: Completion criteria, e.g., passing 95% of tests.
* Process: Steps for test execution, result recording.
  1. Defect Severity Levels:
* Critical: Causes crashes or anomalous results.
* High: Lack of essential functionality.
* Medium: Degrades quality with workarounds.
* Low: Minor impact on functionality.
* Cosmetic: Aesthetic issues, no functional impact.
  1. Test Reporting

i. Reports produced as a result of testing should include:

1. Daily reports detailing the number of tests conducted, passed, and failed.

2. Brief descriptions of the areas being tested and the areas which are failing.

3. Reports on critical bugs or issues discovered during testing.

4. Final test reports at the end of each testing phase or milestone, summarizing all test results and findings.

ii. Testers will feed information back to project managers through:

1. Regular status updates during stand-up meetings or through written communication channels.

2. Detailed bug reports documenting issues found during testing, including steps to reproduce, severity, and impact.

3. Collaborative discussions with team to prioritize and assign fixes for identified bugs.

4. Providing feedback on the overall quality of the software and suggesting improvements or adjustments to testing strategies.

* 1. Test reporting will include daily reports on test execution progress, passed and failed tests, and defective summaries.

Reports will be sent to team leaders, and quality assurance team members.

Testers will communicate defect information to the project leader for bug prioritization and assignment to developers.

1. **Test Schedule**
   1. **This is the section where you wrote layout a schedule for the testing and be able to give an estimate of how long the testing will take and approximately when it will be complete.**

**Answer:**

**A detailed test schedule will be created outlining the testing phases, estimated duration, and completion timelines.**

1. **Control Procedures**
   1. 6.1 Reviews  
      6.2 Bug Review Meetings  
      6.3 Change Request  
      6.4 Defect Reporting
2. **Functions To Be Tested**

Answer:

Truck Selection Algorithm: We’ll make sure the algorithm picks the right truck based on the package’s weight, size, and where it’s going. We’ll also test tricky scenarios, like what happens if a package is too big or the destination is invalid. Diversion Calculation: We’ll check that the algorithm figures out the best way to get around buildings and deliver the package. We’ll also test what happens if there’s no valid path. Input Validation: We’ll test what happens if someone enters something weird, like a negative weight or a destination that doesn’t exist.

1. **Resources and Responsibilities**  
   8.1. Resources : QA team members responsible for test case preparation, execution, and defect reporting. Development team for fixing reported defects and implementing algorithm changes.
2. 8.2. Responsibilities : QA team: Test case preparation, execution, defect reporting, and test documentation. Development team: Fixing reported defects and implementing algorithm changes based on QA feedback.
3. **Deliverables**

Test-plan-template, test cases, test execution reports, defect report, test-description-template.docx, function-description-template

1. **Suspension / Exit Criteria**Testing may be suspended if critical defects affect testing progress or if significant changes are made to the algorithm. Testing will be considered complete when all test cases are executed, and exit criteria are met.

**Answer: We’ll consider testing done when: We’ve run all the test cases. All the critical and high-priority bugs have been fixed. We’ve met all the goals we set out in the test plan.**

1. **Resumption Criteria**

Testing will resume once critical defects are fixed, and the system is stable for testing.

Answer: We’ll start testing again when: The really bad bugs have been fixed, and the system is stable. Any big changes to the algorithm have been made and tested. The team members we need are available to keep going.

1. **Dependencies**12.1 Personnel Dependencies: Availability of team members for testing and defect fixing.  
   12.2 Software Dependencies: Availability of the C environment and necessary libraries for algorithm implementation.  
   12.3 Hardware Dependencies: Availability of standard PCs or laptops for testing purposes.  
   12.3 Test Data & Database: Availability of synthetic test data for simulating various scenarios.
2. **Risks**  
   13.1. Schedule: Delays in defect fixing or algorithm implementation may impact the testing schedule.  
   13.2. Technical: Complexity of the algorithm may lead to unforeseen issues during testing.  
   13.3. Management: Changes in project priorities or resource allocation may affect testing timelines.  
   13.4. Personnel: Enthusiastic participation and discussion, checking test cases and bugs is essential  
   13.5 Requirements: Timely communication and updates will be the key factor to be able to correct errors and give the best results for project development.
3. **Tools**

Testing will be conducted using standard C development tools and libraries for algorithm implementation and testing.

Answer:

Development Tools: We’ll use Visual Studio for coding and debugging. Testing Tools: We’ll use the Visual Studio Test Framework for running our tests.

Collaboration Tools: We’ll use GitHub for version control and Jira for tracking bugs and tasks.

1. **Documentation**

Comprehensive documentation will be maintained for test-plan-template, test cases, test execution reports, defect report, test-description-template.docx, function-description-template

1. **Approvals**

Test plan and associated documents will be reviewed and other team members and leader by project stakeholders before testing commences.

Answer: The test plan and all related documents will need to be approved by: The QA lead. The project manager.