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.**

The project is a system that allocates additional delivery locations to delivery trucks along 3 different routes. The system must allocate the deliveries in an efficient manner while considering the capacity requirements of each truck and monitoring these requirements throughout the program. Essentially, a delivery must be assigned to the tuck that’s route is closest to the delivery location, so long that the truck has capacity to carry the package.

The testing will include various unit testing to ensure that no semantic errors are found within the code. There will also be functional testing so that the business requirements of the application are tested. This can be combined with more in-depth integration testing so that we ensure that the program’s individual units are combined correctly.

Talk about testing more in depth if necessary…

1. **Scope**
   1. **In this section you can describe what will be tested and what will not be tested.**
2. **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
   2. **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.**
3. **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.
4. **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.
5. **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.**
6. **Control Procedures**
   1. 6.1 Reviews  
      6.2 Bug Review Meetings  
      6.3 Change Request  
      6.4 Defect Reporting
7. **Functions To Be Tested**
8. **Resources and Responsibilities**  
   8.1. Resources  
   8.2. Responsibilities
9. **Deliverables**
10. **Suspension / Exit Criteria**
11. **Resumption Criteria**
12. **Dependencies**  
    12.1 Personnel Dependencies  
    12.2 Software Dependencies  
    12.3 Hardware Dependencies  
    12.3 Test Data & Database
13. **Risks**  
    13.1. Schedule  
    13.2. Technical  
    13.3. Management  
    13.4. Personnel  
    13.5 Requirements
14. **Tools**
15. **Documentation**
16. **Approvals**