Pointer

Test_Phase Test Plan

Version 3.1
Version_Date

Table of Contents

Introduction

1.1 Using This Template

Stop - Read This First!!
Before you begin filling out information in this template, first update the document codes that are embedded throughout this template (this will save you time).Go to File, Properties and then go to the Custom tab.You'll see the following entries – just update them with information pertinent to your project:
•Project – Project Name
•Company – Company Name
•TestPhase – The name of the test phase this plan covers
•Version – Version number of this document -
•VersionDate – The date this version was published -
Delete this portion (Section 1.1) once you've completed your version of the document.

1.2 Purpose

Describe the purpose of this plan.

1.3 Intended Audience

Describe the intended audience of this plan. Depending on the test phase, potential audience groups would include:

•Test team

•Development team

•Project Management/PMO

•Stakeholders

•Quality Assurance or other compliance groups

1.4 Test_Phase Objectives

List the test objectives for this specific test phase. For example, if the plan was for Unit Test, objectives might include: stabilizing the test environment, ensuring that all developed functionality performs per spec, etc… Think about secondary objectives that may occur during the test phase. For example, during the System Test phase you may execute all UAT scripts to ensure that they run successfully, or validate training material.

1.5 Relationship to Other Plans

Describe how this test plan fits into the overall document structure that exists. This may include:

- Overall test strategy
- Other test phase test plans
- Company or group/department-wide test documents (e.g. Software Quality Assurance Plan)

1.6 References

List all reference documents that were used to create this plan, will be used in test case development, and/or will be utilized during test execution.

Document	Author	Version	Location

1.7 Revision History

Revision	Author	Date	Comments
1.0			
1.0			

1.8 Test Plan Approvals

List everyone who has to either review or approve the test plan document in this table, along with their project role.

Name	Project Role	Approval Date	Responsibility
			Approver
			Reviewer

2 Scope and Test Approach

2.1 Scope Inclusions

List everything that will be tested during this test phase. This is very important for the test plan since it prevents someone saying later “I thought that was being tested in test phase X”. You don’t need to list detailed requirements – providing a list at the feature level is usually sufficient.

2.2 Scope Exclusions

Equally important is the list of items that will **not** be tested. Be as explicit as possible when producing this list.

2.3 Test_Phase Approach

This is a high level description of the approach for this test phase. You should not necessarily discuss all test phases in this section (you can refer readers to the Test Strategy document for that), but you may want to highlight important dependencies. Items to include in this section include:

- How requirements traceability will be met
- How test cases will be produced, reviewed, and approved
- The use of an automated test tool vs. manual testing
- Who will be executing the tests (e.g. testers, end users, developers)

3 Quality Risks

If you haven't already done a quality risk assessment on the system, this section will force you to do it now. It is important to analyze quality risks because they shape your testing approach and define the areas that you need to pay special attention to.

This section should highlight only system quality risks – not every project risk from the project risk register (see Section 9 for where some of these risks are addressed). System quality risks might include:

- Feature A is being implemented with new technology
- User adoption is critical and several UI screens were not prototyped during requirements or design
- Requirements for feature M changed significantly during the design phase

For every item in the table you should identify one (or more) strategies that will directly address and mitigate these quality risks.

Risk ID	Risk Description	Impact	Test Strategy

4 Schedule and Resources

4.1 Planned Schedule

This section should include a list of key milestones for this test phase, including (but not limited to):

- Approval of test plan
- Development of test case list
- Development of test cases
- Development of automated test scripts
- Preparation of test environment
- Test execution dates

4.2 Required Resources

List all resources required for test planning and execution. Do not forget to include non-test team members who need to be involved (e.g. stakeholders to sign off on the test case list, IT operations who need to maintain the test environment(s), requirements analysts who may need to clarify documentation during test case development).

5 Test_Phase Transition Criteria

The following describe required criteria in order for testing to move from one state to another.

5.1 Entry Criteria

List all criteria that must be met in order for test execution to begin. Possible items to list include:

- Test plan approved
- Test environment stable and ready
- Test cases written and approved
- Test tools ready
- Previous test phase's exit criteria met
- Test resources available

5.2 Exit Criteria

List all criteria that must be met in order for this test phase to be considered complete. Possible items for inclusion are:

- Test case completion
- Number and severity of open defects
- Passing of test objectives listed in Section 1.4

5.3 Suspension Criteria

This section should include criteria or conditions that if they occur, testing should be stopped. This section is very important – many times the test team is asked to continue testing in some vain attempt to meet published schedules when in reality the software is not ready for testing.

5.4 Resumption Criteria

In this section list criteria that must be met before testing can resume, in the event testing is suspended per the criteria in Section 5.3.

6 Test Environment and Configurations

6.1 Test Environment

Describe the test environment(s) required for this test phase. Be as specific as possible, particularly with requirements on data setups and operations support requirements. Make note if the environment will be significantly different than other test environments and/or the production environment.

6.2 Test Configurations

Describe the specific test configurations that will be utilized during this test phase. What you list here will vary by project, but some items that might be included are:

- Operating System
- Client hardware
- Client Java version
- Browser
- Connection type (connected, disconnected)

If there are specific configurations that you know are not supported and/or won't be tested, it is helpful to list those as well. You may also wish to call out which configuration will be used for which types of testing, if they vary. You can summarize this information in a chart, or use the pre-defined sections below.

6.2.1 Test Configuration A

Details on test configuration A.

6.2.2 Test Configuration B

Details on test configuration B.

7 Test_Phase Execution

7.1 Roles and Responsibilities

Describe each role/responsibility in this section. At a minimum, make sure you have roles identified for the following:

- Test environment support
- Development liaison
- Release manager (person responsible for deploying code/configuration changes to the test environment)
- Test manager
- Project manager (or other point of contact for escalating issues)
- Person producing test metrics
- Approver(s) of test plan
- Approver(s) of test scripts
- Approver(s) of test phase deliverables

7.2 Test Case Tracking

In this section describe how the status of each test case will be tracked for each test iteration. Also include the criteria for deciding how to determine the status of each test case (e.g. what needs to happen for a test case to be passed, failed, blocked, etc…)

If you have a test management tool, you can make a reference to the documented processes for the tool.

7.3 Test Defect Tracking

In this section you need to detail:

- What tool is utilized for defect tracking
- Statuses used and what they mean
- Classification definitions
- The defect workflow, including roles (i.e. who makes the determination that a defect is ready to test, or can be closed)

Since a good defect tracking process can be leveraged over and over again with little variation, I strongly suggest documenting a process and simply referencing it in each test plan. For a suggested defect workflow, visit CarnegieQuality.com.

7.4 Release Management

This is an often overlooked process – so make sure it is worked out prior to the start of testing. The release management process should describe (at a minimum):

- Who (and how) makes the decision to move code/configuration changes to the test environment
- Who is responsible for moving the code/configuration changes to the test environment
- How does notification of a new release occur
- Who produces the release report (which should include the new tag for the release as well as the defects addressed by the release)
- If required, who updates defect statuses based on the new release
- If there is a Configuration Management System, there should be a reference to the documentation as well

7.5 Planned Test Iterations

Describe the number of planned test iterations and (at a high level) what will get tested in each iteration. 2If this information is not known yet, place best guess estimates and refer readers to where this information can be found when it is developed.

For a template that can assist in planning test iterations, visit CarnegieQuality.com.

7.6 Planned Test Hours

List the planned test hours (and days). If there are specific non-testing days (e.g. holidays), list them here also.

8 Test_Phase Deliverables and Metrics

8.1 Deliverables

List all deliverables for this test phase, along with who is responsible for producing each one.

8.2 Metrics

Describe the metrics that will be produced during test execution, the frequency of publication, how they will be published (e.g. via email, within a testing tool, on a shared drive), and who is responsible for publication.

Some suggestions for test metrics may be found at CarnegieQuality.com.

9 Risks and Contingencies

This section should contain risks that are specific to testing. Note that quality risks of the application should be specified in Section 3 and not duplicated here. Some examples of risks include:

- Loss of key test personnel
- Inexperienced personnel
- Possibility of catastrophic failure to the test environment
- Overly demanding test schedule
- New/un-proven testing tools
- Etc…

For each risk, list one or more contingencies that can be prepared for ahead of time. For example, if you have inexperienced testing personnel you may decide on training prior to test execution. Or for the possibility of test environment failure, you may plan on a backup server.

Each risks impact to the project should be quantified as much as possible.