11

Writing Test Plans

Young people think quick thoughts. Old people think deep thoughts. Leaders think long thoughts.

A major emphasis for many organizations is documentation, including test plans and test plan reporting. Unfortunately, putting too much of a focus on documentation can lead to an environment where lots of meaningless reports are produced but nothing useful is done. That is why this book largely focuses on content, not form. The contents of a test plan include how the tests were created, why the tests were created, and how they will be run.

Producing test plans, however, is an essential requirement for many organizations. Companies and customers often impose templates or outlines. Rather than surveying many different types of test plans, we look at the IEEE standard definition. The original version was defined in 1983 (829-1983), with updates in 1990 and 1998, with the most recent being 829-2008, the "IEEE Standard for Software and System Test Documentation." A quick search on the Web will supply you with more test plans and test plan outlines than you could ever use. The 829-2008 standard defines a test plan as:

"(A) A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning. (B) A document that describes the technical and management approach to be followed for testing a system or component. Typical contents identify the items to be tested, tasks to be performed, responsibilities, schedules, and required resources for the testing activity."

The two major types of test plans in the current standard are:

1. A *Master Test Plan* (MTP) provides an overall test planning and test management document for multiple levels of test. An MTP can either apply to one project, or apply to multiple projects within the same organization.
2. A *Level Test Plan* (LTP) describes testing at a particular level, where the levels are as described in Chapter 1. Each LTP must describe the scope, approach, resources, and schedule of the testing activities for its level of testing. The LTP then defines the items being tested, the features to be tested, the testing tasks to be performed, who is responsible for each task, and any risks associated with that testing.

Below is an outline for a sample level test plan, provided as example only. The plan was derived from numerous samples that have been posted on the Web, so does not exactly represent a single organization. It is based on the IEEE 829 standard.

11.1 LEVEL TEST PLAN EXAMPLE TEMPLATE

1. **Introduction**: The introduction puts the test activities described in the document in the context of the overall project and test effort for the project.
   1.1. *Document identifier*: Each document must have a unique name, encoding information such as the document's date, the author, etc.
   1.2. *Scope*: The scope should describe what is being tested for this document level. Details about the portion of the software being tested may be included.
   1.3. *References*: Related documents should be referenced here. External and internal documents should be identified and listed separately.
   1.4. *Level in the overall sequence*: This should be a figure that shows how the testing described in this document fits into the overall project development and test structure.
   1.5. *Test classes and overall test conditions*: This section should describe what is unique about the testing activity being documented. This may describe how testing should proceed for components, integration testing, or the system. Generally what should be tested, or test criteria to be used, should be described

here.

2. **Details For This Level Of Test Plan**: The following subsections should be introduced here. The general test approach should be described here, along with criteria for test completion.
   2.1. *Test items and their identifiers*: This section should identify the system under test (or component or integrated subsystem). This will also document details about the software component under test, including how to install it, run it, and any environmental needs it has.
   2.2. *Test traceability matrix* : This section should document the origin of each test. This may be requirements, test coverage requirements, or design elements. Testers and test managers should be able to look up each test and understand **why** it was included and **what** it tests.
   2.3. *Features to be tested*: All features to be tested should be explicitly listed, using names that are referenced in other software documentation (such as the user manual, requirements document, or design document).
   2.4. *Features not to be tested*: Everything that will not be tested should be listed. This section should also explain why not.
   2.5. *Approach*: This section should describe how this testing should be carried out, including test criteria, level of automation, etc.
   2.6. *Item pass/fail criteria*: For each item to be tested, when can it be deemed to have passed testing? This may be stated in terms of remaining issues, or percentage of tests that pass. This can also be weighted by severity of the issues.
   2.7. *Suspension criteria and resumption requirements*: Some failures are severe enough that it makes no sense to continue testing. The criteria for when to suspend testing and wait for the development team to correct the problem should be clarified.
   2.8. *Test deliverables*: This section should list all documents and data that are to be delivered during testing.
3. **Test Management**: This section describes what will be done when and who will do them.
   3.1. *Planned activities and tasks; test progression*:
   This section should describe the tasks that must be done to plan for testing and carry out testing. Any inter-task dependencies and constraints should be identified.

3.2. *Environment* and *infrastructure*: This sections should described the test environment, including anything that the testers need before running tests. This should address facilities needed, hardware, software, database, support tools, results capturing tools, privacy issues to be address, and security issues to be address.
3.3. *Responsibilities and authority*: This section should identify who is responsible for managing, designing, preparing, executing, checking results, and resolving problems found during this testing. This section should also identify anybody else who may be needed during testing.
3.4. *Interfaces among the parties involved*: This section should describe how the people should communicate. Each person involved with testing should be able to look at this section and know who to contact when needs arise.
3.5. *Resources and their allocation*: This section should describe any needed resources that are not identified previously in the LTP.
3.6. *Training*: This section should identify what knowledge, skills, and training the test personnel need. It should also include how that knowledge can be obtained.
3.7. *Schedules, estimates, and costs*: This section should provide the schedule for testing, including preparation, design, and execution of tests. The major test milestones should be highlighted.
3.8. *Risks and contingencies*: This section should identify any risks that can be foreseen, and provide suggestions for how to avoid the risks, how to mitigate the risks, and how to recover if something happens.

4. **General**: This section contains general information that is needed for testing, including QA procedures, metrics, glossary, etc. The subsequent sections should be described here.
4.1. *Quality assurance procedures*: This section should describe the plan for quality assurance of the testing effort. If the project has a separate quality assurance plan, it can simply be referenced here.
4.2. *Metrics* : This section should describe how testing will be measured and reported.
4.3. *Test coverage*: This section should describe how coverage is measured and how much coverage is required.
4.4. *Glossary*: This section should provide a list of terms and their

definition, including acronyms.

4.5. *Document change procedures and history*: This section should document changes to the LTP document.

11.2 BIBLIOGRAPHIC NOTES

The primary source for test plans is IEEE's 829 document [IEEE, 2008]. The current version is 829-2008, which replaced 829-1998. The original was 829-1983. If the IEEE standards document is behind a paywall, Wikipedia has a reasonable introduction [Wikipedia, 2009]. A useful related document is BS 7925-2, the British Computer Society's Standard for Software Component Testing [British Computer Society, 1997].