

If you are doing a careful and professional job of testing, and if failures of the program could be expensive or dangerous to the customer, keep records. Ask your company's attorney what records she would find most useful if there were a liability suit, and provide them.

We discuss these issues in more detail in Chapter 14.

## TYPES OF TEST DOCUMENTS

This section describes some types of documents that you can develop for test materials. Many of these descriptions summarize *IEEE Standard 829-1983 for Software Test Documentation*, which attempts to define a common set of test documents, to be used across the industry. Schulmeyer (1987) summarizes many other test documentation specifications.

You can order Standard 829-1983, which includes examples and much more detailed definitions, for a few dollars from:

Computer Society of the IEEE  
P.O. Box 80452  
Worldway Postal Center  
Los Angeles, CA 90080

Or call the IEEE Standards Sales office in New Jersey: 201-981-0060.

Standard 829 does not specify which documents you should write for each project. We won't either, except to say that you probably don't want to write one of each. Also, you might choose to omit some of the detail required by the Standard. We urge you not to feel bound to make your documents conform to the IEEE standard. We describe the Standard because it provides a background of careful thought, which you should adapt to your needs. Finally, don't feel compelled to write everything at the start of testing. Try to publish your qualifying acceptance test before testing begins. It also helps to write the first draft of the test plan up front. Write and refine the rest as you go.

### Test plan

The test plan provides an overview of the testing effort for the product. You can put everything into this one document (some people do), but it's more common to write many documents and reference them in the appropriate sections. Here are the sections of the test plan, as defined by IEEE Standard 829:

- **Test plan identifier.** A unique name or number, useful if you store all documents in a database.
- **Introduction.** Include references to all relevant policy and standards documents, and high level product plans.
- **Test items.** A test item is a software item (function, module, feature, whatever) that is to be tested. List them all, or refer to a document that lists them all. Include references to specifications (e.g., requirements and design) and manuals (e.g., user, operations, and installation).
- **Features to be tested.** Cross-reference them to test design specifications.
- **Features not to be tested.** Which ones and why not.

12

TEST PLANNING &  
DOCUMENTATION

## DOCUMENTING TEST MATERIALS

### TYPES OF TEST DOCUMENTS

#### Test plan

- **Approach.** Describe the overall approach to testing: who does it, main activities, techniques, and tools used for each major group of features. How will you decide that a group of features is adequately tested? The Standard also says that this section, not the Schedule section, is the place to identify constraints, including deadlines and the availability of people and test items.
- **Item pass/fail criteria.** How does a tester decide whether the program passed or failed a given test?
- **Suspension criteria and resumption requirements.** List anything that would cause you to stop testing until it's fixed. What would have to be done to get you to restart testing? What tests should be redone at this point?
- **Test deliverables.** List all of the testing documents that will be written for this product.
- **Testing tasks.** List all tasks necessary to prepare for and do testing. Show dependencies between tasks, special skills (or people) needed to do them, who does each, how much effort is involved, and when each will be done.
- **Environmental needs.** Describe the necessary hardware, software, testing tools, lab facilities, etc.
- **Responsibilities.** Name the groups (or people) responsible for managing, designing, preparing, executing, witnessing, checking, fixing, resolving, getting you the equipment, etc.
- **Staffing and training needs.** How many people you need at each skill level, and what training they need.
- **Schedule.** List all milestones with dates, and when all resources (people, machines, tools, and facilities) will be needed.
- **Risks and contingencies.** What are the highest risk assumptions in the test plan? What can go sufficiently wrong to delay the schedule, and what will you do about it?
- **Approvals.** Who has to approve this plan? Provide space for their signatures.

#### Function list

IEEE Standard 829 does not discuss this document. For its details, see "Components of test planning documents: Outlines—the function list" in this chapter. You could include a function list in the test plan's section on test items, or treat it as a separate document.

#### Criteria for acceptance into testing

IEEE Standard 829 does not discuss this document.

This acceptance test is a brief test that the program must pass when submitted for testing. If it passes, the Testing Group runs the item through a full test cycle. Otherwise they reject it as too unstable for testing. Such tests should take less than half an hour—never more than two hours.

If you use an acceptance test, write a document that describes it *exactly*. Circulate it to programmers, preferably before the first cycle of testing. Make the document detailed enough for programmers to run the tests themselves before submitting the product for testing. Let them catch their most obvious blunders in private.

### Test design specification

This specifies how a feature or group of features will be tested. According to Standard 829, it includes the following sections:

- **Test design specification identifier.** This is a unique name or number.
- **Features to be tested.** Describe the scope of this specification.
- **Approach refinements.** Expand on the approach section of the test plan. Describe the specific test techniques. How will you analyze results (e.g., visually or with a comparison program)? Describe the boundary or other conditions that lead to selection of specific test cases. Describe any constraints or requirements common to all (most) tests.
- **Test identification.** List and briefly describe each test associated with this design. You may list a test case under many different designs if it tests many different types of features.
- **Feature pass/fail criteria.** How can the tester decide whether the feature or combination of features has passed the test?

### Test case specification

This defines a test case. According to Standard 829, the test case specification includes the following sections:

- **Test case specification identifier.** A unique name or number.
- **Test items.** What features, modules, etc., are being tested? References to specifications and manuals are in order.
- **Input specifications.** List all inputs by value, by range of values, or by name if they are files. Identify anything else that's relevant, including memory-resident areas, values passed by the operating system, supporting programs or databases, prompt messages displayed, and relationships between the inputs.

Describe any timing considerations. For example, if the tester should enter data while the disk light is flashing, or within half a second after a certain message, say so. For very short intervals, describing the rhythm can be more effective than describing the exact times involved.

- **Output specifications.** List all output values and messages. Consider including response times.
- **Environmental needs.** List special requirements, including hardware, software, facilities, and staff.
- **Special procedural requirements.** List anything unusual in the setup, tester's actions, or analysis to be done of the output.
- **Inter-case dependencies.** What tests have to be executed before this one, why, and what if the program fails them?

# 12

TEST PLANNING &  
DOCUMENTATION

## DOCUMENTING TEST MATERIALS

### TYPES OF TEST DOCUMENTS

#### Test procedure specification

##### Test procedure specification

This describes the steps for executing a set of test cases and analyzing their results. According to Standard 829, it includes the following sections:

- **Test procedure specification identifier.**
- **Purpose.** What is this procedure for? Cross-reference all test cases that use this procedure.
- **Special requirements.** List any prerequisite procedures, special tester skills, and special environmental needs.
- **Procedure steps.** Include the following steps as applicable:
  - **Log:** any special methods or formats for logging results or observations.
  - **Setup:** preparations for execution of the procedure.
  - **Start:** how to begin execution of the procedure.
  - **Proceed:** any actions necessary during procedure execution.
  - **Measure:** how test measurements (e.g., response times) are made.
  - **Shut down:** how to suspend testing in the face of unscheduled events (or when the tester goes home for the night).
  - **Restart:** where to restart and how, after a shut down.
  - **Stop:** how to bring execution to an orderly halt.
  - **Wrap up:** how to restore the environment to its original state.
  - **Contingencies:** what to do when it all goes wrong.

##### Test item transmittal report

This report accompanies anything submitted to you for testing. The report tells you what you're getting. According to Standard 829, it includes the following sections:

- **Transmittal report identifier.**
- **Transmitted items.** Names the submitted program or modules, along with their version identifiers or revision levels. Names the people responsible for this submission.
- **Location.** Where is the submitted material—on a disk or tape, in a shared directory, in a binder? How is it labeled?

- **Status.** How has this changed since the last time you tested it? Which Problem Reports were resolved? Did the specification or visible program behavior change? What invisible changes were made and how might they affect program reliability? How does this material differ from the published specification or manual and which is correct? What significant changes are yet to come?
- **Approvals.** The people who have to agree that this material is ready to test should sign the transmittal before you accept it for testing.

### Test script

IEEE Standard 829 does not discuss this document. It is described above, in "Test script for the inexperienced tester." It should include the following components:

- **General Instructions.** These tell the tester how to read and use the script, how and when to fill out Problem Reports, where to find them, etc. You might provide this material in a separate binder, rather than pad the script with it, but you must provide it to the inexperienced tester.
- **Getting started.** Setup information.
- **Step by step procedural description for each test.**
- **Check-off boxes for each step and result.**
- **Ample room to describe behavior that was odd or just not understood,** and questions that prompt these descriptions. An experienced tester should review these answers later, examine the behavior herself, and probably file many new Problem Reports on the basis of them.

12

TEST PLANNING &  
DOCUMENTATION

### Test log

This is a chronological record of the test executions and events that happened during testing. According to Standard 829, it includes the following sections:

- **Test log identifier.**
- **Description.** What's being tested, including Version ID, where testing is being done, what hardware (printer, amount of available memory, type of computer, etc.), and all other configuration information (for example, operating system revision level).
- **Activity and event entries.** What happened, including:
  - **Execution description:** The procedure used, who witnessed the test, and their role in testing.
  - **Procedure results:** What happened. What did you see, and where did you store the output?
  - **Environmental information:** Any changes (e.g., hardware substitutions) made specifically for this test.
  - **Anomalous events:** Unexpected events (usually due to bugs). What happened before and after they occurred.
  - **Incident report identifiers:** Problem Report numbers.

## DOCUMENTING TEST MATERIALS

### TYPES OF TEST DOCUMENTS

#### Test incident report

#### Test incident report

This is a Problem Report. The IEEE Standard report has different fields from the report in this book. The IEEE report has these fields: test incident report identifier, summary, inputs, expected results, actual results, anomalies, date and time, procedure step, environment, attempts to repeat, testers, observers, and impact on test plans and specifications.

#### Test summary report

This is a summary of a series of tests, of the type that you might issue after completing a cycle of testing. It briefly describes the testing done and evaluates the results. According to Standard 829, it includes the following sections:

- *Test summary report identifier.*
- *Summary.* Say what was tested (including Version ID), in what environment, and summarize your evaluation of it. Refer to test case specifications.
- *Variances.* Report any deviation of test procedures from the specified ones, and explain why.
- *Comprehensiveness assessment.* Was testing as comprehensive as the test plan called for? What modules, features, or feature combinations weren't tested enough, and why?
- *Summary of results.* What problems were reported, which were resolved, and what were the resolutions? Which are still outstanding?
- *Evaluation.* Overall evaluation of each item (program or module) tested, based on the test results. Optionally, estimate the risk and probability of failure of the item in actual use.
- *Summary of activities.* Summarize such things as the number of staff who worked on the tests summarized in this report, the total machine time used, total elapsed time, and any special events or other resource uses that deserve mention.
- *Approvals.*

#### Documentation embedded in data and control files

When you create a file of input data for a test, if you can, embed comments in these files to explain why you chose each data value.

Control files execute a test. If comments are possible in the file, use them to explain each step in the file.

During a test, it pays to show the expected results onscreen or in a printout. The tester can compare these to the obtained results. This is convenient, since she has to make this comparison anyway. Don't display