Optimizing Your Test Documentation using The New IEEE Standard 829

Hans Schaefer
Software Test Consulting
N-5281 Valestrandsfossen, Norway
Hans.schaefer@ieee.org

1. The problem

Test documentation should not be the same in every project. Criticism against the old IEEE 829 standard was that it was too rigid. A lot of paperwork would be created with low value, partially duplicating information.

The new draft standard explicitly encourages tailoring in order to avoid this.

2. Varying needs for test documentation: one size does not fit all

Generally, when looking into how much test documentation is necessary, one has to look into different factors:

The integrity level:
•If the risk in using the system is great, then the integrity requirements are high. Standards for developing safety critical systems require more test levels, a higher degree of rigor and more tangible evidence, i.e. documentation, from testing. Even authorities are a new stakeholder in software of higher integrity. This means that the integrity level should influence the degree of test documentation.

How long time will your system be alive:
If a system lives for a long time, some of the people who have developed it will move on, and new personnel may enter the product support, maintenance and further development. These new people need information about the system and its tests. One way is coaching, but it is not always possible to transfer all necessary information this way. In many cases, documentation is a more useful and more complete way to transfer information. The test may also need some tailoring and adaptation, new tests need to be developed and others adapted or deleted. This means the tests must be understood.

Will there be many people who need to understand the testing done?
•If projects are reorganized, or people are changed, or projects are transferred, outsourced etc, the new people enter. They need documentation. If such people are far away (offshoring), documentation will help more.

On the other hand, if people have the same background, both culturally and educationally, they may need less documentation, as silent understanding does not need to be documented. People with a thorough knowledge of the application area, programming and program design, or testing techniques, may be less in need of written documentation.

How large is your system?
•Documentation demands grow with system size. (As there would normally be larger numbers of people, again). Offshoring and outsourcing increases demand. The larger the system, the more test levels could exist. They in turn need to be coordinated, which increases the need for plans etc.

How are your tools and how is your existing documentation?
Tools are used in order to keep track of information. For example, project management tools, test management tools, test automation tools and configuration management tools may include a lot of test documentation items. Such items should then rather be included in the tool, rather than written in separate documents. The same is true if test planning is an integrated part of project or quality planning, and test reporting an integrated part of usual project reporting.

•Do you have standards for working?
•Finally, an organization may have standard ways of working. Thus, what is standard needs no extra documentation. Only deviations need to be documented. This may greatly reduce needs for special documents. If the standards are followed, indeed...

The result: Test documentation needs always vary. This means the test documentation needs to be flexible, and it needs to be tailored to the organization and the project. The new IEEE 829 draft support these needs.

3. Choosing which documents to have and not to have and what to include in them

The new standard requires the user to choose an integrity level scheme, or use the one given as an example in the standard itself. From that scheme, the user should choose which documents to include and which not, and which documents could be combined. For example, for high integrity software, all documents mentioned in the standard should be used, and a full document set for every level of testing. For lower integrity, there may be fewer levels of testing and less documents. Documents may be combined, and test levels may be done informally or undocumented. There is a lot of flexibility from this.

Finally, there is only one requirement left in the standard: Every document and every item in it need to be ”addressed”. This means there must be a documented serious consideration about what to include and what not, and what to move where. If you know what you do, and you give a good reason and document it, you can do what you want! The whole standard turns into a checklist of what may be necessary.

4. The new master test plan and report

When using the old standard, the user started with writing a test plan. The test plan was typically well suited for one test level. If used to document several test levels, one had to modify its structure, duplicate chapters or somehow improvise. The old test plan layout was not optimal. Projects needed a master test plan or test strategy. IEEE Standard 1012 (software verification and validation planning) could provide help, showing how to plan not only testing, but even reviews and other analysis activities during the project. However, that standard was explicitly directed at so-called critical software, i.e. software with high integrity. In many cases, this was too bureaucratic. The two standards were not well coordinated either.

Thus, the new standard includes a Master Test Plan (MTP) and a corresponding Master Test Report (MTR), for planning and reporting the testing of the whole project. There is room in it to define the different levels of testing and their organization, and the total resources necessary for testing.

For every test level, there is now the Level Test Plan (LTP), which to a large degree contains the information found in the old standard’s test plan. A new item is quality assurance procedures for test stuff plus a few minor changes. A level test pan may be used to plan the whole testing for this level, like all integration testing, or one part of this test, like one integration step or testing of one module only. Correspondingly, there is a Level Test Report (LTR). The collected Level Test Reports lead to the Master Test Report.

The other documents are more or less the ones known before: Test design (often called test specification), Test Case, Test Procedure, Test Log, and Anomaly Report.

A new item has a horrible name: Level Interim Test Status Report (LITSR). Someone must have gone berserk in inventing this name. But it is important! After all, managers want to know how testing is going, trace it more or less continuously. Thus, this document is necessary, but probably in high need of tailoring. But tailoring is part of the new standard.

5. No more test item transmittal report

This item in the old standard has not been continued. Is it necessary anyway? Yes and no. Yes, because it is necessary to know exactly what is being tested and when and where. No, because this information should be known to configuration management – if that works. But with modern configuration management tools procedures and traditions, the test configuration should be know and under control. Thus, this item is not longer necessary. It is implicit.

6. May there be something missing?

Probably, things are missing. No project is equal. One consideration is a difficult test environment. Test environment needs are addressed in several places, but there may be a need for a though and complete test environment specification. However, the test environment varies just too much. It is probably better to generate such a description individually at your organization.

Another consideration could be a higher-level test procedure, describing how a whole large test should be executed, in parallel and sequentially. It would be about like a production plan for a film. But again, this would be highly dependent on context.

There may be the need for a lot of utilities, scripts, specially written simulators and corresponding instructions. Test automation also may mean different needs. Again, this is extremely context dependent.

Such items are left to the user to define.

7. Moving information from test documentation to somewhere else

Information may be found in the configuration management plan or system, in the project plan or other planning documents, in general development information and, last but not least, in tools and databases. Thus, test documents should not look the same in every project, but need to be adapted. This adaptation, tailoring, should be outlines in one single source, to be produced at the beginning of the project or early iterations.

First, there should be a project **Glossary**. The **Glossary** may be put into a central place in the project, for example published as an internal web page.

Another typical item to be put aside is change information and document change procedures. Even document status, author and approvals could be put into a Configuration Management Plan or similar document.

Section no.	Test Document(s)	Contents
1.1	All	Date of issue, issuing organization, author, approvals, status
1.3	All	References
Last	All	Document change procedures and history
1.5	Test cases	Notation for description
2.8	Anomaly / Incident report	Anomaly status

A Project Plan or other planning documents may cover the following topics:

Section no.	Test Document(s)	Contents
1.5	MTP	Test overview
2.1.1	MTP	Process: Management
1.5.2	MTP	Master test schedule
1.5.4	MTP	Resources summary

2.1	MTP	Test processes
2.8	LTP	Test deliverables
3.1	LTP	Planned activities and tasks; test progression
3.2	LTP	Responsibilities and authority
3.4	LTP	Interfaces among the parties involved
3.5	LTP	Resources and their allocation
3.6	LTP	Training
3.7	LTP	Schedules, estimates, and costs
3.8	LTP	Risks and contingencies
2.5	LTD (Design)	Test deliverables

Additionally, the master test plan may contain the level test plan information, especially for minor projects.

Sections 4.2 to 4.4 in a Level Test Plan (LTP) (Quality control procedures, Metrics and Test coverage) may be covered by general quality assurance plans or documentation.

Sections 2.2. through 2.4. (Requirements vs. test traceability matrix, Features to be tested, Features not to be tested) could be combined into one table telling which requirements are tested in more or less thoroughness by which test cases.

The following topics may be covered by general development documentation:

Section no.	**Test Document(s)**	**Contents**
1.4	MTP	System overview and key features
2.3, 2.4	LTP	Features to be tested with priorities
2.1	LTD	Features to be tested

The following topics may be fully or partially eliminated. Instead, the information may be contained in or generated by tools, for example a specification database, a test management tool or other tools.

Section no.	**Test Document(s)**	**Contents**
2.2	LTP	Test traceability matrix
2.3	LTP	Features to be tested and priorities
2.4	LTP	Features not to be tested
2.1	LTD	Features to be tested
2.3	LTD	Test identification
1.4	LTC (Case)	Context
1.5	LTC	Notation for description
2 (2.1-2.7)	LTC	Details of the Test Case(s)
	LTC	(Database, spreadsheet, test automation tool)

All	Test procedure	Script in test automation tools
All	Test log	May be automatically generated by logging tools
All	Anomaly report	Part of incident management system
2.2	LTR	Detailed test results, from logging tool

8. Status

The new standard has been balloted and is being updated for some minor, but important, comments. There will be another ballot soon, where after the IEEE editors have a last review. The new standard should replace the old one later this year. It may be obtained through the IEEE Computer Society.

9. References

IEEE Std 829-1998 (Revision of IEEE Std 829-1983)
IEEE Standard for Software Test Documentation (currently valid standard)

ANSI/IEEE Std P829-2008, January 21, 2008
Draft IEEE Standard for software and system test documentation (probable future standard)