



# Software Test Documentation

## (IEEE 829-1998 and 2008)

**Hans Schaefer**

[hans.schaefer@ieee.org](mailto:hans.schaefer@ieee.org)

<http://home.c2i.net/schaefer/testing.html>



# Contents

- 1 - To document or not to document**
- 2 - How much Documentation**
- 3 - IEEE Standard 829 - 1998 and 2008 and discussion**
- 4 - Key concepts in the new standard**
- 5 - Which Documents for What**

## **Part II**

**Standard 829-2008 templates**

**Test Policy and Strategy**



# 1 - To Document Or Not

**Most important task: To test!**

**Documentation is secondary.**

**If time pressure, to test is more important.**

# 2 - How Much Test Documentation Do We Need?



## Discussion



# Danger With Documentation Templates

- **Excuse to switch off the brain**
- **“One size fits all”**



## 3 - IEEE Standard 829 - Old (1998)

- **Standard for format and contents of test documentation**
- **Document based**
- **Test Plan, Test Design, Test Procedure, Test Case, Test Item Transmittal Report, Anomaly Report, Test Log, Test Summary Report**
- **Unchanged since 1983**
  
- **Criticism: Not flexible, waste of time, wrong focus, DETRIMENTAL!**
- **Pro: Good checklist! Good if used flexibly.**



## 3 - IEEE Standard 829 - 2008

- Significant changes from the prior version.
- Changed focus from being document-focused to being process-focused.
- New concept of an integrity level to assist organizations in determining a recommended minimum set of testing tasks and concurrent selection of test documentation needed to support the tasks. (adds flexibility)
- New process for choosing appropriate documents and contents.
- -> continued



# IEEE 829 - 2008

- New Master Test Plan (MTP) for documenting the actual management of the total test effort. (Level Test Plans for every level).
- New Level Interim Test Status Report to be issued during the test execution activity.
- New Master Test Report for when there are multiple Level Test Reports that need consolidation. The Master Test Report may also summarize the results of the tasks identified in the Master Test Plan, and the Level Test Reports.
- Sample metrics.
- Concept of independence.





## 4 - Key Concepts In The New Standard

- **Integrity levels.**  
The standard defines four integrity levels (from high integrity to low integrity) to describe the importance of the software to the user. (or the RISK).
- **Recommended minimum testing tasks for each integrity level.**  
Optional testing tasks for tailoring the test effort to meet project needs and application specific characteristics.

# Key Concepts

- **Differing intensity and rigor applied to testing tasks.**  
**Higher integrity levels -> greater intensity and rigor.**

**Intensity includes**

**greater scope of testing**

**across all normal and abnormal system operating conditions.**

**Rigor includes**

**more formal techniques and recording procedures.**

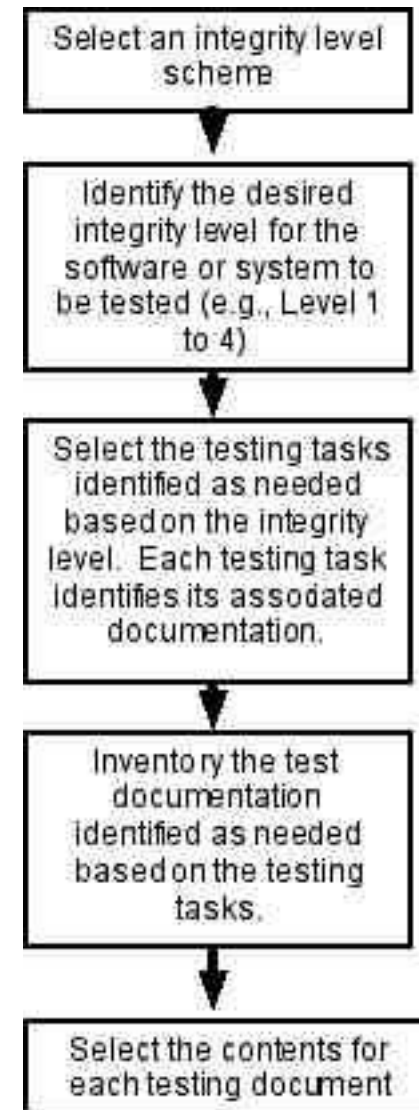


# Key Concepts

- **Systems viewpoint.**  
Recommended minimum testing tasks to respond to system issues.
- **Selection of test documentation.**  
Both the types of test documentation and the content within each document need to be selected based on the testing tasks associated with the identified integrity level.
- **Compliance with International and IEEE Standards.**  
Compliant with life cycle process standards such as ISO/IEC Std 12207, IEEE Std 1074-1997.  
Supports the full software life cycle processes including acquisition, supply, development, operation, and maintenance.  
The standard is compatible with all life cycle models.

# Usage Flow

- Find a integrity scheme
- Select an integrity level
- Select testing tasks, and depth and rigor -> documentation needs
- Collect test documentation needs
- Select contents



# 5 - Which Documents For What?

- **Strategic (company, many projects)**
  - Test policy, Test strategy (handbook) -> see other standards!
- **Test planning**
  - Master and Level Test Plans
- **Test design**
  - Test Design
- **Test cases/test procedures and their execution**
  - Test Cases, Test Procedure
- **Test results reporting**
  - Test Log, Anomaly Report, Level Interim Test Status Report, Level Test Report, Master Test Report.



# Literature And References

- Kaner, Falk, Nguyen, Testing Computer Software, 1999.
- IEEE Standard 829-1983, Standard for Software Test Documentation. Online at [www.wikipedia.org](http://www.wikipedia.org)  
-> search for IEEE 829.
- IEEE Standard 829-2008, Standard for Systems and Software Test Documentation (IEEE Standards Association)
- ISO/IEC Draft Standard 29119 (to be completed in 2012).



# Part 2: Test Documentation And IEEE Standard 829-2008

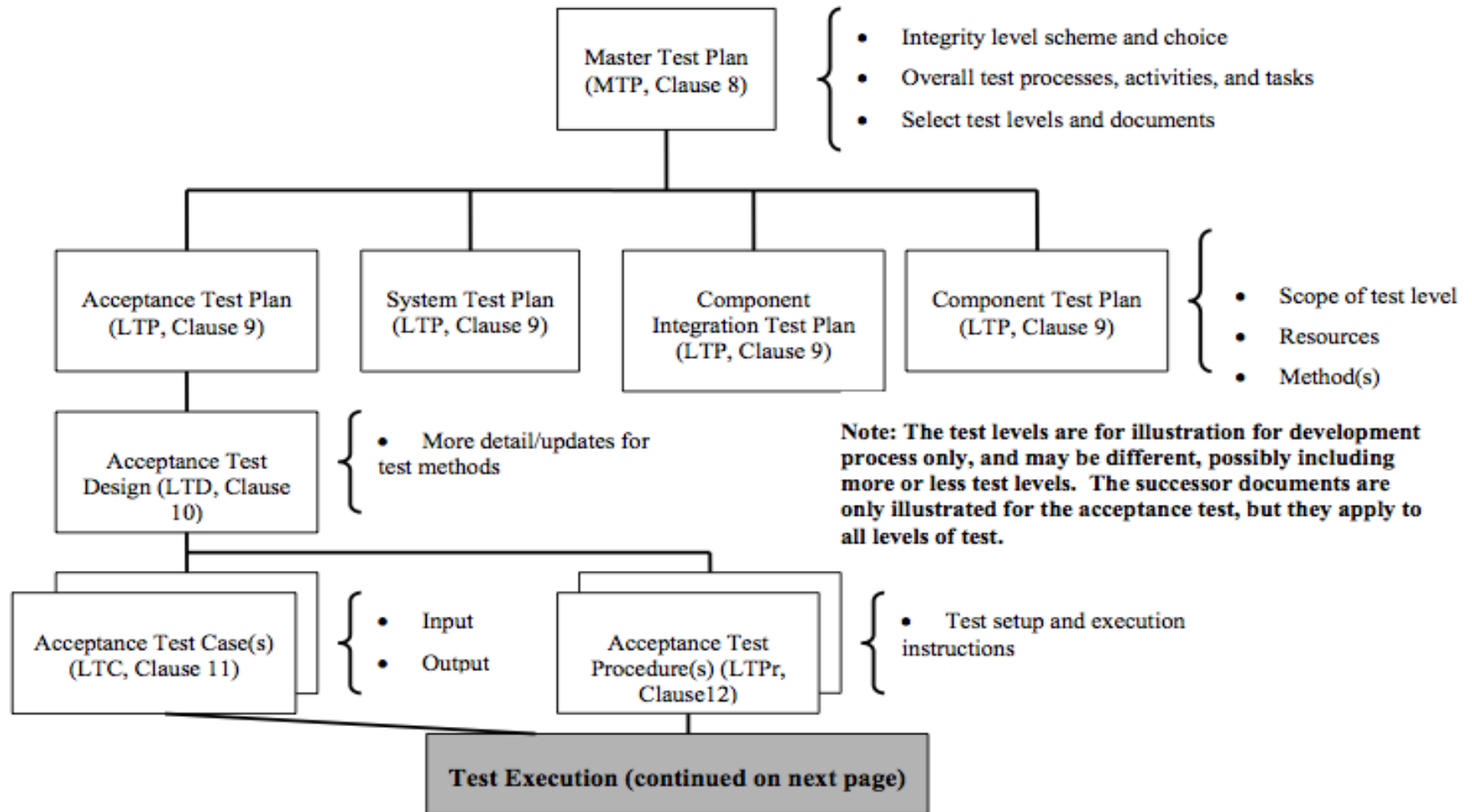
# Example: Recommended Documents per Integrity Level



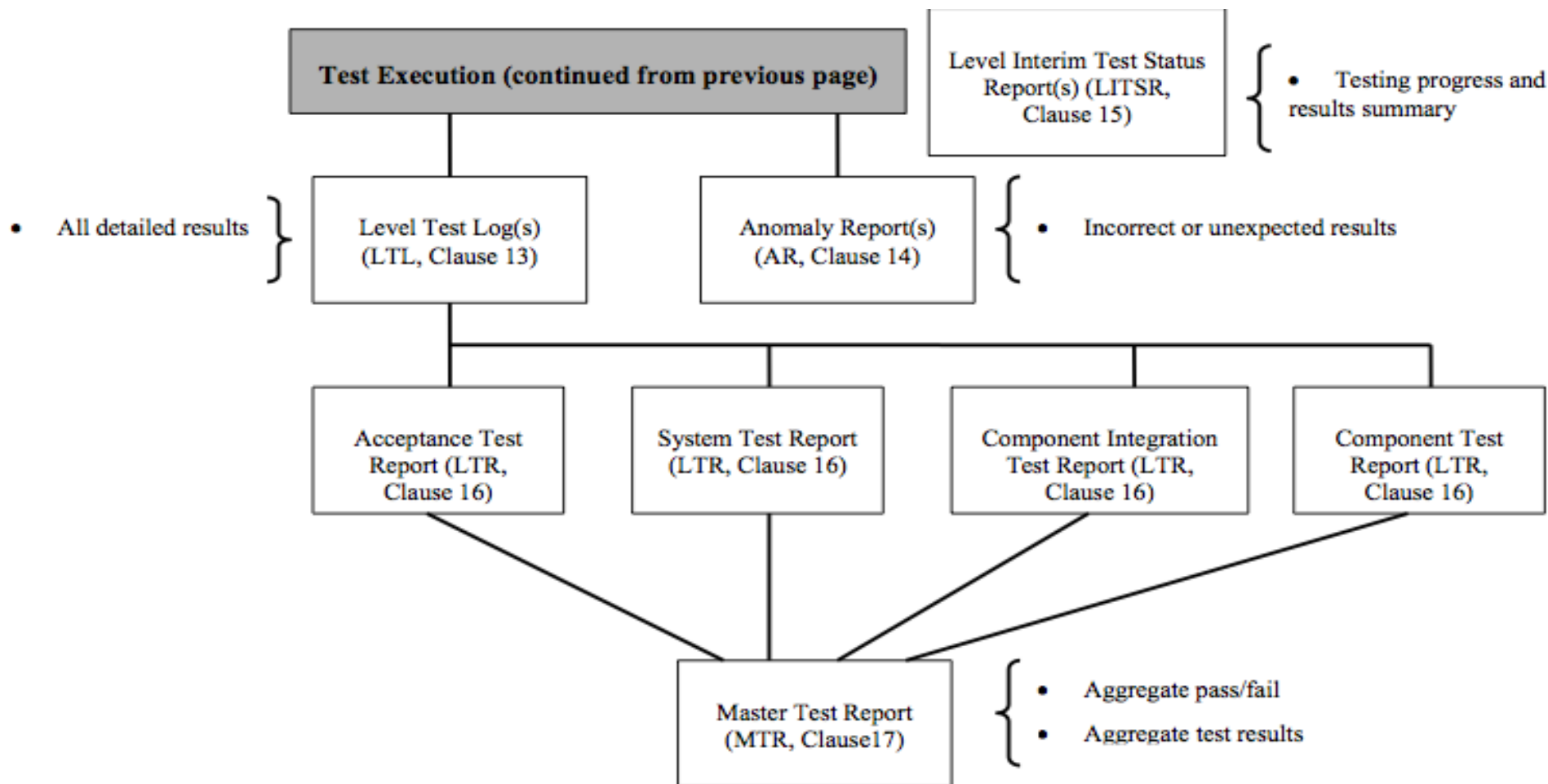
Integrity Level	Example Test documentation
4 Catastrophic	Master Test Plan
	Level Test Plan (Component, Component Integration, System, Acceptance)
	Level Test Design (Component, Component Integration, System, Acceptance)
	Level Test Case (Component, Component Integration, System, Acceptance)
	Level Test Procedure (Component, Component Integration, System, Acceptance)
	Level Test Log (Component, Component Integration, System, Acceptance)
	Anomaly Report
	Level Interim Test Status Report (Component, Component Integration, System, Acceptance)
	Level Test Report (Component, Component Integration, System, Acceptance)
	Master Test Report
3 Critical	...
2 Marginal	...
1 Negligible	Level Test Plan (Component Integration, System)
	Level Test Design (Component Integration, System)
	Level Test Case (Component Integration, System)
	Level Test Procedure (Component Integration, System,)
	Level Test Log (Component Integration, System)
	Anomaly Report (Component Integration, System)
	<ul style="list-style-type: none"> <li>The integrity level also determines the content and rigor of the test documents.</li> </ul>



# IEEE 829 Test Planning + Preparation



# IEEE 829 Test Execution And Reporting



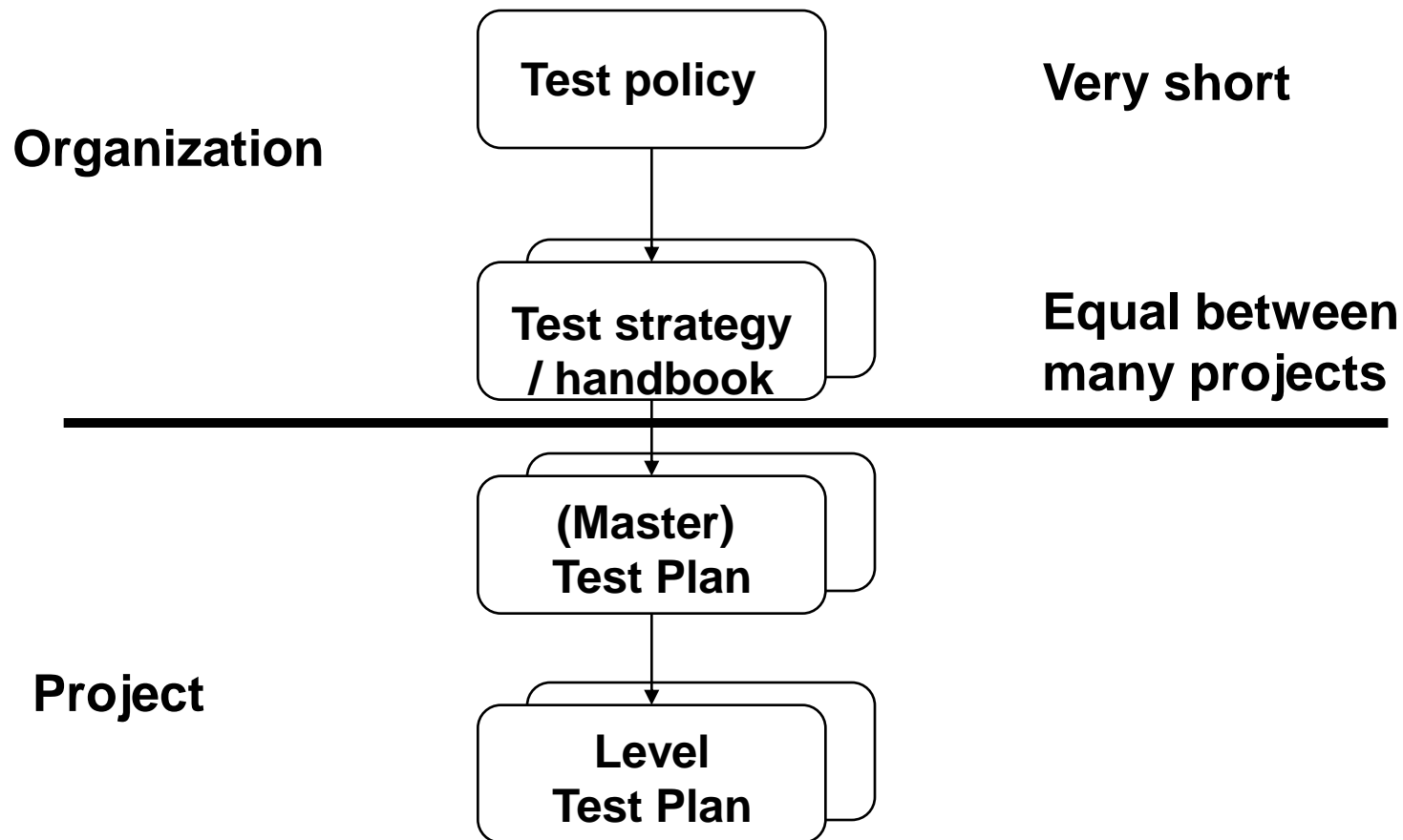
# General Principles

1. Do not duplicate information!
2. If information is somewhere else, refer to it.
3. Introductory items may be generally defined elsewhere.
4. Information may be in tools or databases, not in test documents.
5. Tailor the outlines from the standard to the actual needs!
6. Test documentation should be a tool. Most important is its usefulness.

The standard requires you to “address” every of its items.

Address = seriously consider.

# Planning Documents Inside And Outside A Project



# Master Test Plan Outline

- **1. Introduction**
  - 1.1. Document identifier
  - 1.2. Scope
  - 1.3. References
  - 1.4. System overview and key features
  - 1.5. Test overview
    - 1.5.1 Organization
    - 1.5.2 Master test schedule and responsibilities
    - 1.5.3 Integrity level schema
    - 1.5.4 Resources summary
    - 1.5.5 Responsible people (if not in 1.5.2)
    - 1.5.6 Tools, techniques, methods, and metrics
- **2. Details of the Master Test Plan - see next page**
  - 2.1. Test processes including definition of test levels
  - 2.2. Test documentation requirements
  - 2.3. Test administration requirements
  - 2.4. Test reporting requirements
- **3. General**
  - 3.1. Glossary
  - 3.2. Document change procedures and history

The test plan should discuss  
the product and project risks!

# Master Test Plan cont'd

## – 2. Details of the Master Test Plan

### 2.1. Test processes including definition of test levels and tasks

Describe the test levels and the relationship between them and the division of work. What is their focus?  
Describe deviations from test policy and strategy and their reasons.

#### 2.1.1 Process: Management

##### 2.1.1.1 Activity: Management of test effort

#### 2.1.2 Process: Acquisition

##### 2.1.2.1: Activity: Acquisition support test

#### 2.1.3 Process: Supply

##### 2.1.3.1 Activity: Planning test

#### 2.1.4 Process: Development

The V-model, for example.

#### 2.1.5 Process: Operation

##### 2.1.5.1 Activity: Operational test

#### 2.1.6 Process: Maintenance

##### 2.1.6.1 Activity: Maintenance test

### 2.2. Test documentation requirements

### 2.3. Test administration (and control) requirements

### 2.4. Test reporting requirements



# Task Description In MTP (Example)

1	Task	Generate System Test Design
2	Methods	Ensure that test design correctly emanates from the system test plan and conforms to IEEE Std 829-2008 regarding purpose, format, and content.
3	Inputs	System Test Plan, IEEE Std 829-2008
4	Outputs	System Test Design, provide input to Master Test Report
5	Schedule	Initiate (with all inputs received) 30 days after the start of the project. Must be completed and approved 120 days after start of project.
6	Resources	Refer to clause 1.5.4.
7	Risks & assumptions	Risk: adequacy and timeliness of the test plans Assumption: Timeliness is a primary concern because the team writing the test cases is dependent on the receipt of this the test plans
8	Roles & responsibilities	Refer to clause 1.5.5.



# Master Test Plan: Important Points

- **Define main objective (why)**
- **Define main test targets (what, criticality, risk)**
- **Determine resources**
- **Plan test actions**
- **Later: Execute according to plan, evaluate and change**

Unambiguous

specific

Measurable

how do you know you achieved it?

Reasonable costs

depends on risk

Benefits

no time wasted, no risk areas untested,  
goals reached with minimal resources



# Level Test Plan Outline

- **1. Introduction**
  - 1.1. Document identifier
  - 1.2. Scope
  - 1.3. References
  - 1.4. Level in the overall sequence
  - 1.5. Test classes and overall test conditions
- **2. Details for this level of test plan**
  - 2.1 Test items and their identifiers
  - 2.2 Test Traceability Matrix
  - 2.3 Features to be tested (high risk)
  - 2.4 Features not to be tested (and why) (low risk)
  - 2.5 Approach
  - 2.6 Item pass/fail criteria
  - 2.7 Suspension criteria and resumption requirements
  - 2.8 Test deliverables
- **3. Test management - see next page**
- **4. General**

Include any general info  
Into the Master Test Plan!



# Level Test Plan cont'd

- **3. Test management**
  - 3.1 Planned activities and tasks; test progression
  - 3.2 Environment/infrastructure
  - 3.3 Responsibilities and authority
  - 3.4 Interfaces among the parties involved
  - 3.5 Resources and their allocation
  - 3.6 Training
  - 3.7 Schedules, estimates, and costs
- **4. General**
  - 4.1 Risks and contingencies (project risks)
  - 4.2 Quality assurance procedures
  - 4.3 Metrics
  - 4.4 Test coverage
  - 4.5 Glossary
  - 4.6 Document change procedures and history

# Test Plan Risks

- **Too much detail**
  - **no possibility to use creativity**
  - **unrealistic**
- **Maintenance**
  - **not up to date, difficult to maintain**
- **Risks forgotten**
  - **no contingencies and solutions, no reserves**
- **Test coverage not addressed**
  - **how do we know we are done?**
- **Missing links**
  - **project management (tools)**
  - **configuration management**
  - **defect management**

# Test Plan Template from IEEE 829-1998

## For the exam!



1. Test Plan Identifier
2. Introduction
3. Test items
4. Features to be tested (product risks)
5. Features not to be tested (and why)
6. Approach
7. Item pass/fail criteria (test exit criteria)
8. Suspension criteria and resumption requirements
9. Test deliverables
10. Testing tasks
11. Environmental needs
12. Responsibilities / Authority
13. Staffing and training needs
14. Schedule
15. Risks and contingencies (project risks)
16. Approvals

# Test Design Outline

- **1. Introduction**
  - 1.1. Document identifier
  - 1.2. Scope
  - 1.3. References
- **2. Details of the Level Test Design**
  - 2.1. Features to be tested
  - 2.2. Approach refinements
  - 2.3. Test identification
  - 2.4. Feature pass/fail criteria
  - 2.5. Test deliverables
- **3. General**
  - 3.1. Glossary
  - 3.2. Document change procedures and history

**May be combined with  
test cases and test  
procedures!**

# Test Case Outline

- **1. Introduction (once per document)**
  - 1.1. Document identifier
  - 1.2. Scope
  - 1.3. References
  - 1.4. Context
  - 1.5. Notation for description
- **2. Details (once per test case)**
  - 2.1. Test case identifier
  - 2.2. Objective
  - 2.3. Inputs
  - 2.4. Outcome(s)
  - 2.5. Environmental needs

**May be combined with  
test design and test  
procedures!**

# Test Procedure Outline

- **1. Introduction**
  - 1.1. Document identifier
  - 1.2. Scope
  - 1.3. References
  - 1.4. Relationship to other procedures
- **2. Details**
  - 2.1. Inputs and outputs
  - 2.2. Ordered description of the steps to be taken to execute the test cases
- **3. General**
  - 3.1. Glossary
  - 3.2. Document change procedures and history

**May be combined with  
test design and test  
cases!**



# Test Procedure Main Points (ch. 2.2)

- Logging: any tools or methods for logging the results of test execution, the anomalies observed, and any other events pertinent to the test.
- Special requirements: tester user Ids, privileges, skills, permissions, information flows, tools, other things.
- Set-up: the sequence of actions necessary to prepare for execution of the procedure.
- Start: the actions necessary to begin execution of the procedure.
- Proceed: any actions necessary during execution of the procedure.
- Measurement: how the test measurements will be made.
- Shut down / suspension: the actions necessary to suspend testing, when unscheduled events occur.
- Restart / resumption: restart points and description of the actions necessary to restart the procedure at each of these points.
- Stop: the actions necessary to bring execution to an orderly halt.
- Tear-down / Wrap-up: the actions necessary after the execution of the procedure has been completed.
- Contingencies: the actions necessary to deal with anomalous events that may occur during execution.



# Test Log Outline

- **1. Introduction**
  - 1.1. Document identifier
  - 1.2. Scope
  - 1.3. References
- **2. Details**
  - 2.1. Description
  - 2.2. Activity and event entries
- **3. General**
  - 3.1. Glossary
  
- **Often automatically generated!**

Log any incident  
and its possible impact  
on schedule, resources  
etc.!



# Good practice for logs

Source: How we test at Microsoft

## Every entry should contain:

- Test ID, Test Name, Environment info, Application under test ID, Test result

## Good practices for logging (auto) tests:

- Passing tests -> minimal info in logs
- Failing tests -> enough info for debugging. Include the binary and function under test, functionality under test, expected and actual results
- Error codes -> understandable text
- Logging should be configurable without recompiling automated tests. No logging, normal logging, extensive logging.

# Anomaly Report / Incident Report, see also IEEE 1044



- **1. Introduction**
  - 1.1. Document identifier
  - 1.2. Scope
  - 1.3. References
- **2. Details**
  - 2.1. Summary
  - 2.2. Date anomaly discovered
  - 2.3. Context
  - 2.4. Description of anomaly
  - 2.5. Impact
  - 2.6. Originator's assessment of urgency (see IEEE 1044)
  - 2.7. Description of the corrective action
  - 2.8. Status of the anomaly
  - 2.9. Conclusions and recommendations
- **3. General**
  - 3.1 Document change procedures and history



# Test Reporting, What Is Important?

- **How was testing done**
- **With respect to plan**
- **Results**
- **Reference to system and test material**
  
- **Provide exact information about our knowledge about the system**



# Level Interim Test Status Report Outline

- **1. Introduction**
  - 1.1. Document identifier
  - 1.2. Scope
  - 1.3. References (**tested object, version, testware**)
- **2. Details**
  - 2.1. Test status summary
  - 2.2. Changes from plans (**what was not done as planned?**)
  - 2.3. Test status metrics
- **3. General**
  - 3.1. Document change procedures and history

# Test Level Report Outline

- **1. Introduction**
  - 1.1. Document identifier
  - 1.2. Scope
  - 1.3. References (**tested object, version, testware**)
- **2. Details**
  - 2.1. Overview of test results
  - 2.2. Detailed test results (**ref. interim reports, deviations**)
  - 2.3. Rationale for decisions
  - 2.4. Conclusions and recommendations
- **3. General**
  - 3.1. Glossary
  - 3.2. Document change procedures and history



# Master (Final) Test Report Outline

- **1. Introduction**
  - 1.1. Document identifier
  - 1.2. Scope
  - 1.3. References (**tested object, version, testware**)
- **2. Details of the Master Test Report**
  - 2.1. Overview of all aggregate test results (**ref. deviations!**)
  - 2.2. Rationale for decisions
  - 2.3. Conclusions and recommendations
- **3. General**
  - 3.1. Glossary
  - 3.2. Document change procedures and history

# How To Add Value

- **Test documentation should be a tool!**
- **Keep exploratory testing / ad hoc testing possible**
- **Avoid expensive documentation costs**
- **Beware of a paper intensive process** (Kaner, Petticord, Bach)
- **Keep your test documentation synchronized with requirements**
- **Categorize test doc. requirements by risk**





# References

- **IEEE Standard 829 - 1998 Standard for Software Test Documentation**
- **IEEE Standard 829 - 2008 Standard for System and Software Test Documentation**
- **Draft of ISO Standard 29119 Part 2 (August 2008): Software Testing - Test Process**
- **MIL-STD-498 Software Development and Documentation Standard, 1989, [www.pogner.demon.co.uk/mil\\_498/frindex.htm](http://www.pogner.demon.co.uk/mil_498/frindex.htm)**
- **ANSI/ISO/IEC TR 9294:1990: Information Technology: Guidelines for the Management of Software Documentation**
- **ANSI/ISO/IEC TR 6592:2000: Information Technology: Guidelines for the Documentation of Computer Based Application Systems**
- **TTCN-3: Testing and Test Control Notation, version 3. A notation for test systems. "An introduction into the TTCN", D. Hegrefe et al., Computer Networks Vol 42, No 3, pp 375-403, 2003.**