

# SM-515

## Software Verification & Validation

Fall 2002 Semester

Lecture 11: Test Documentation  
23/12/2002

Dr. Kivanç DINÇER

TÜBİTAK-UEKAE/ILTAREN  
<http://www.uao.tubitak.gov.tr/>

## Test Planning

- Organizations need to plan a testing activity in the same way they plan a development activity
  - Test Plan (similar to development plan)
    - defines the scope of the work to be performed.
    - defines what must be done and who must do it.
  - Test Procedure
    - is a container document that holds all of the individual tests (test scripts) that are to be executed.
  - Test Report
    - documents what occurred when the test scripts contained in the test procedure were run.
    - (include bugs found and fixed during the testing effort, as well as relevant data)

23/12/2002 Lecture 11 SM-515 Software Verification & Validation (Fall 2002)  
© Dr. Kivanç Dinçer

3

## Examples of Software Units

- Function, Procedure, Routines, or Subroutine:
  - primarily encapsulate software control flow
  - are associated with procedural programming
  - each represent a callable software component
- Module or Compilation Unit:
  - encapsulate both data and control flow
  - correspond to modular programming
  - normally consists of an interface and an implementation
- Class
  - encapsulate both data and control flow
  - correspond to object-oriented programming

(Classes can be derived from each other, and thus can inherit attributes and methods.)

23/12/2002 Lecture 11 SM-515 Software Verification & Validation (Fall 2002)  
© Dr. Kivanç Dinçer

2

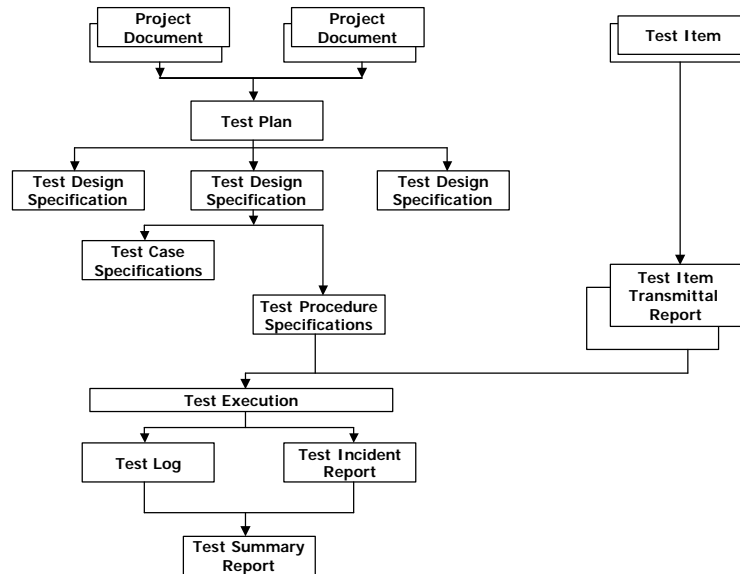
## IEEE Std.829-1998

- IEEE Standard for Software Test Documentation defines a standardized approach for documenting software testing and encourages software testing to be performed more systematically.
- Numerous small documents are produced in different phases of SDLC
- The following are the types of Software Test Documentation:
  - Test Plan
  - Test Design Specification
  - Test Case Specification
  - Test Procedure Specification
  - Test Item Transmittal Report
  - Test Log
  - Test Incident Report
  - Test Summary Report

Depending on the size of the project, or the nature of the testing activities, some of the above documents can be simplified, combined, or discarded.

© Dr. Kivanç Dinçer

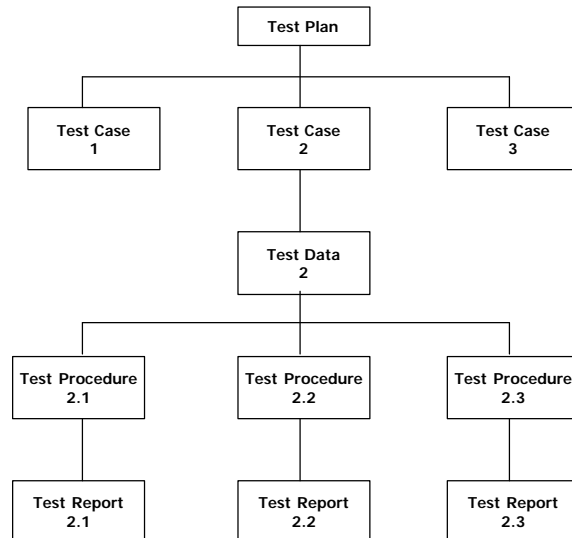
### Relationship of test documents, adapted from IEEE Std. 829



## Benefits of a Systematic Testing Process

- More effective testing
  - By developing test plans, test designs, test cases and test procedures earlier, test phase will be conducted more effectively (better test coverage, etc.)
- More effective follow-up
  - By test incident reports, software defects can be tracked more easily and rework can be done properly.
  - Test summary reports show the readiness of the software for installation.
- Reuse of test cases and test procedures
- Proof of testing
  - Might be necessary for an audit, or to provide evidence of regulatory compliance.

### Typical Testing Flow



## Test Plan

- describes all testing activities performed for the project.
- Acc.to IEEE Std.829, it is a document
  - describing the scope, approach, resources, and schedule of intended testing activities and
  - identifying test items, the features to be tested, the testing tasks, who do each task, and any risks requiring contingency planning.

- IEEE Std. 1012 provides guidelines for the required levels of testing as a function of the software integrity level.
  - For a software integrity level 1 (low criticality), only system testing is required.
  - For higher software integrity levels (more critical software), unit and integration testing are also required.
- According to IEEE Std. 1012, low criticality is described as follows:
  - Selected function has noticeable effect on system performance, but only creates inconvenience to the user if the function does not perform in accordance with requirements.

- The following topics must be addressed in the test plan regarding the validation testing:
  - Test estimation
    - How many tests do we need?
    - How long will it take to develop those tests?
    - How long will it take to execute those tests?
  - Test development and informal validation
  - Validation readiness review and formal validation
  - Test completion criteria

## Test Plan Outline

- |   |                               |
|---|-------------------------------|
| • Test plan identifier                            | • Test deliverables           |
| • Introduction                                    | • Testing tasks               |
| • Test items                                      | • Environmental needs         |
| • Features to be tested                           | • Responsibilities            |
| • Features not to be tested                       | • Staffing and training needs |
| • Approach  | • Schedule                    |
| • Item pass/fail criteria                         | • Risks and contingencies     |
| • Suspension criteria and resumption requirements | • Approvals                   |

## Test Estimation

- Estimating the number of tests that need to be developed
- Estimating the test development time
- Estimating the test execution time.

## Estimating the Number of Tests

- select a sufficient set of practical tests
  - the objective is to test reasonably completely all valid classes for normal operation and to exhaustively test unusual behavior and illegal conditions.
- based on factors such as:
  - Testing all features and functions defined in the SRS and related documents
  - Including appropriate number of ALAC tests
  - Achieving some test coverage goal
  - Achieving a software reliability goal

## Test Matrices

- Requirement traceability matrices (RTMs) are used to prepare test matrices.
  - We need to track each requirement to the tests that demonstrate software compliance with the reqs (validation).
  - Each req., both functional and interface, is traced to the primary (P) test that demonstrates its correct implementation.
  - (I) indicates other tests in which the reqs. are involved.
  - As the reqs. are modified, this matrix can offer clues to unexpected and usually undesirable results if a req. is changed or eliminated.
  - As the reqs. evolve throughout the development of the software system, each change is going to affect the test program.

## Estimating the Number of Tests

SRS Reference	Estimated Number of Tests Required	Notes
4.1.1	3	2 positive and 1 negative test
4.1.2	2	2 automated tests
4.1.3	4	4 manual tests
4.1.4	5	1 boundary condition, 2 error conditions, 2 usability tests
...		
Total	165	

## Requirements Traceability Matrix

Tests Requirements	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	...
Interface 1	P				I		I	
Interface 2		P						
Interface 3			P					
Function 1			I	P				
Function 2				I	P			
...								

**P:** Primary subject or test  
**I:** Involved in test

## Estimating the Number of Tests

- ALAC Tests:
  - Do it wrong
  - Use wrong or illegal combinations of inputs
  - Don't do enough
  - Do nothing
  - Do too much
- Test estimate should also reflect the complexity of tests as well as whether they are manual or automated tests.
  - Automated Tests
    - developed like manual tests, but can be executed repeatedly under computer control.
    - particularly well suited for testing user interfaces.

## Estimating the Number of Tests

- When estimating the number of tests required for each feature of function take into account issues such as:
  - Test complexity
  - Different platforms
  - Automated and manual tests

## Estimating the Number of Tests

- Developing tests should be viewed as an investment:
  - The time and effort required to identify, write and debug a test can be more than recouped based on the costs required to find and fix bugs once the product is released.
- Estimates
  - First time you try it, your estimate and the actual number of tests developed may be very different.
    - Do a postmortem analysis at the end of the project.
    - Learn from past experience.

## Estimating the Test Development Time

- Rely on the past experience
  - or try a few and get a measure
- Average time (person-hours/test) can be used as a metric
  - write the first draft of the test script
  - run the test once
  - find any errors in the test and revise the test script accordingly.

Estimated Number of Tests	Average Test Development Time (person-hours/test)	Estimated Test Development Time (person-hours)
165	3.5	577.5

## Estimating the Test Execution Time

- Execution time includes the time required to get systems set up for tests, execute tests, and report problems.
  - use past experience

Estimated Number of Tests	Average Test Execution Time (person-hours/test)	Estimated Test Execution time (person-hours)	Estimated Regression Testing (%50) (person-hours)	Total Estimated Test Execution Time (person-hours)
165	1.5	247.5	123.75	371.25

## Test Development and Informal Validation

- The test development task includes the work required to write the tests identified in the test plan.
  - Different test methods and types should be used as appropriate.
- The test plan should describe how the test development work will be partitioned among the QA people on the project team.
  - This task should also be synchronized with the software development plan.
- Informal validation involves developing and executing tests concurrently with feature development.
  - How this will be performed
  - How problems found will be reported to developers
  - How QA will track problems and verify resolution.

- Plan on adding an additional %25-50 to the estimated execution time to allow for regression testing.

It will depend on

- how many inspections were held,
- integrity of bug-fixing process,
- amount of new code vs. reused/modified code,
- amount of unit and integration testing performed,
- experience of developers.

Based on the above mentioned three measures, you can now develop a realistic validation testing schedule.

## Validation Readiness Review & Formal Validation

- The validation readiness criteria that must be satisfied in order to start formal validation should be specified in the test plan.
- The formal validation testing process should also be described in the test plan.
  - bug tracking, baseline change assessments, etc.

### Test Completion Criteria

- The test plan needs to include the criteria that must be met to complete the testing activity.
  - objective, easily measurable, and agreed to by the PM early on in the project.

## Test Cases

- The first step in function testing, and often in input/output testing, is to construct situations that mimic actual use of the software.
- These situations, or test cases (scenarios), should represent actual tasks that the software user might perform.
- Once the test cases have been developed, the software requirements that are involved in each test case are identified by using RTM.

## Test Case Specification Outline

- Test case specification identifier
- Test items
- Input specifications
- Output specifications
- Environmental needs
- Special procedural requirements
- Intercase dependencies

## Test Cases for a POS Terminal

Consider the case of testing the software in a point-of-sale terminal for a convenience store. The store stocks both grocery and fuel products.

- Open the store the very first time.
- Sell products.
  - Sell only fuel
  - Sell only grocery items
  - Sell both fuel and grocery items
- Restock the store.
- Close the store for the last time.

## Test Procedure

- A container document for the collection of test scripts that are to be run.
  - Test scripts define the detailed steps that determine if the software meets a specific requirement.
  - An integral part of each script is the expected results.
- Acc.to IEEE Std.829, it is a document
  - specifying a sequence of actions for the execution of a test.
- Test Procedures tell
  - which buttons to push
  - what data to input
  - what responses to look for
  - what to do if the expected response is not received
  - etc.

## Test Procedure Specification Outline

- Test procedure specification identifier
- Purpose
- Special requirements
- Procedure steps

## IEEE Std.829 Test Reports

- Test Item Transmittal Report
  - A document identifying test items. It contains current status and location information.
- Test Log
  - A chronological record of relevant details about the execution of tests.
- Test Incident Report (~Software Program Report)
  - A document reporting on any event that occurs during the testing process which requires investigation.
- Test Summary Report
  - A document summarizing testing activities and results. It also contains an evaluation of the corresponding test items.

## Test Report

- Documents the results of the formal software validation testing process. It typically includes
  - Completed copy of each test script with evidence that this script was executed (with the tester's signature)
  - Copy of each SPR showing resolution
  - List of open or unresolved SPRs
  - Identification of SPRs found in each baseline along with total number of SPRs in each baseline.
  - Regression tests executed for each software baseline.

## Test Incident Report Outline

- Test incident report identifier
- Summary
- Incident description
- Impact



## Test Summary Report Outline

- Test summary report identifier
- Summary
- Variances
- Comprehensiveness assessment
- Summary of results
- Evaluation
- Summary of activities
- Approvals

## Sample parts of a test log, adapted from IEEE Std. 829

### 2. Description

This log records the execution of the data conversion test procedure (AP08-0101)

### 3. Activities and Event Entries

June 10, 19xx	Incidents
2:00 PM – Dick started testing.	
2:15 PM – Began to generate old database.	
3:30 PM – Discovered a possible bug in the test database.	AP11-14
6:00 PM – Completed the old test database generation.	
6:15 PM – Disk stopped testing.	
June 11, 19xx	Incidents
9:45 PM – Dick started testing.	
10:15 PM – Began to create the random number file.	AP11-15
11:30 PM – Jane started testing.	
11:40 PM – Jane ran the database auditor against the new database.	AP11-16
16:15 PM – Disk stopped testing.	

...

## Logging Test Execution

- The mechanism for logging test execution specified by IEEE Std. 829 differs from the technique used by many organizations.
  - It is common in industry to use a copy of a test procedure, and to check off procedure steps as they are executed, sometimes filling in measures values in blanks provided for this purpose.
  - A test log in the IEEE format is a document entirely separate from a test procedure specification.

## Test Data Input

- Input of test data is the key to testing and comes from a variety of sources.
  - Traditionally, test data inputs have been provided by a test driver software, or tables of test data that are input at the proper time by an executive test control module specially written for the purpose.
  - In the case of interactive computing, a more flexible type of test environment is required.

## Test Environments for Interactive Computing

- Simulators: test software packages that perform in the same manner as some missing piece of hardware or other software.
  - Ex. a complete spacecraft or radar installation.
- Stimulators: represents an outside software or hardware unit that presents input data independently from the activities of the system under test.
  - Ex. input of a warning message that interrupts the processing
- Use of a keyboard or a terminal operated by a test user.

## Test Tools

- Many automated and manual test tools are available to assist in the various test activities.
  - Test Data Provision: Many commercial tools can help in the creation and insertion of test data.
    - Test data generators
    - General-purpose simulators
  - Data Recording
    - Large-scale event recorders
      - for recording and later repeating of tests
    - General- and specific-purpose data reduction packages
      - sorting and categorizing large volumes of data
    - Powerful analysis packages
  - Path Analyzers
    - monitor the progress of the test program and track the exercising of the various paths through the software.