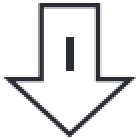
Test Analysis and Design

- The purpose of the test analysis and design activities is to produce test designs with test conditions and tests cases and the necessary test environment based on the test basis and the test goals and approach outlined in the test plan.
- The inputs on which this process is based are:



  - Level test plan
  - Basis documentation
- The activities are:

  - Analysis of basis documentation
  - Design of high-level test cases and test environment
- The output consists of:

  - Test design
  - Test environment design and specification

Input to Test Analysis and Design

- The input from the level test plan that we need for this process is:
  - Test objectives
  - Scheduling and staffing for the activities
  - Definition of test object(s)
  - Approach:- especially test case design techniques to use and structure and contents of the test specification
  - Completion criteria, not least required coverage
  - Deliverables

Documentation of Test Analysis and Design

- The result of the test analysis and design should be documented in the test specification.
- This document or series of documents encompasses
  - The test designs: also called test groups
  - The test cases: many test cases per test design
  - Test procedures: often many-to-many relationship with test cases.
- The overall structure of the test specification is defined in the level test plan.
- The test specification documentation is created *to document the decisions* made during the test development and *to facilitate the test execution*.

Activities in Test Analysis and Design

- The test specification activity can already start when the basis documentation is under preparation.
- The test specification aims at designing tests that provide the largest possible coverage to meet the coverage demands in the test plan.
- The work with the specification of the test groups, the test conditions, the test cases, and the test procedures are highly iterative.
- A side effect of the analysis is that we get an extra review of the basis documentation.

Activities in Test Analysis and Design

- In Test Analysis and Design phase , it performs the following basic activities.
  - Defining Test Designs
  - Identification of Test Conditions
  - Creation of Test Cases

Defining Test Designs

- In test design the testing task is broken into a number of test design or test groups.
  - This makes the test development easier to cope with, especially for the higher test levels.
- Test groups may also be known as test topics or test areas.

Activities in Test Analysis and Design Cont.

- A test design or test group specification should have the following contents according to IEEE 829:

1. Features to be tested (test conditions)
2. Approach refinement
3. List of high-level test cases
4. List of expected test procedures
5. Feature pass/fail criteria

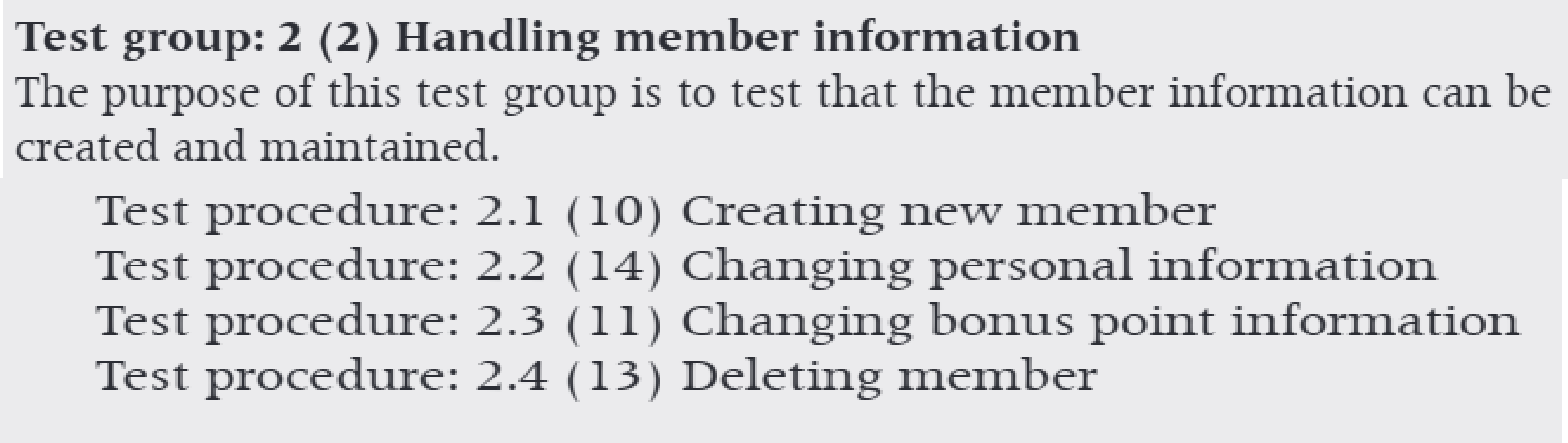
- Test design specification approvals
- The groups and the procedures must be uniquely identified.
- The number of test groups we can define depends on the test level and the nature, size, and architecture of the test object:

Activities in Test Analysis and Design Cont.

- In component testing we usually have one test group per component
- For integration testing there are usually a few groups per interface
- For system and acceptance testing we typically have many test groups.
- A few examples of useful **test groups** defined for a system test are:
  - ✓ Start and stop of the system
  - ✓ Functionality x
  - ✓ Nonfunctional attribute xx
  - ✓ Error situations

Activities in Test Analysis and Design Cont.

- To document the test designs ,often a list of groups with a short purpose description and list of the test procedures for each are sufficient. Example:-



Test group: 2 (2) Handling member information
The purpose of this test group is to test that the member information can be created and maintained.

Test procedure: 2.1 (10) Creating new member
Test procedure: 2.2 (14) Changing personal information
Test procedure: 2.3 (11) Changing bonus point information
Test procedure: 2.4 (13) Deleting member

- The unique identification is the number in brackets, for example (10).
- The number before the unique identifier is the sorting order.

Activities in Test Analysis and Design Cont.

❖ Identification of Test Conditions

- The features to be tested mentioned in the test design can be expressed as test conditions or test requirements.
- A test condition is a verifiable item or element.
- The nature of a test condition depends on the nature of the test basis documentation. It may for example be
  - ✓ a function,
  - ✓ a transaction,
  - ✓ a feature,
  - ✓ a requirement, or
  - ✓ a structural element like an interface parameter or a statement in the code

Activities in Test Analysis and Design Cont.

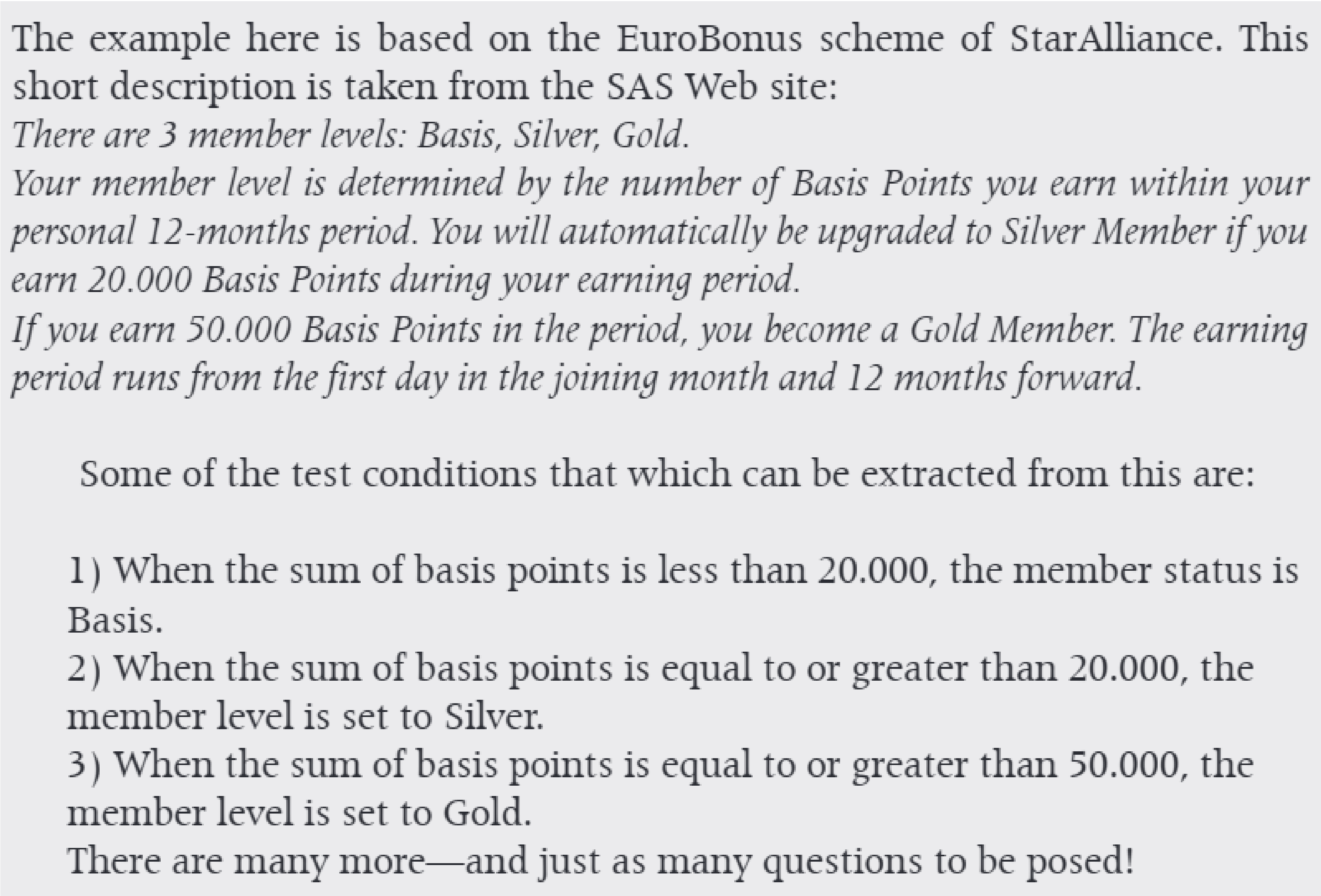
❖ Identification of Test Conditions

- The test conditions are based on or identical to our coverage items.
- We cannot expect to be able to cover 100% of all the relevant coverage items for our test.
- The completion criteria often include the percentage of the coverage items we must cover, called the coverage.
  - ➢ We select the test conditions to get the highest coverage.
- Prioritization criteria identified in the risk analysis and test planning may be applied in the analysis activity to pick out the most important coverage items.

Activities in Test Analysis and Design Cont.

- For example :The completion criteria for a component test could include a demand for 85% decision coverage.
- The documentation of a test condition must at least include:
  - Unique identification
  - Description
  - Reference to test basis documentation, if not taken from there directly
- Only if the test conditions are not clearly defined in the basis documentation do we have to document them ourselves.
- If we do so we must get the test conditions reviewed and approved by the stakeholders. **Example :-**

Activities in Test Analysis and Design Cont.



The example here is based on the EuroBonus scheme of StarAlliance. This short description is taken from the SAS Web site:

There are 3 member levels: Basis, Silver, Gold.

Your member level is determined by the number of Basis Points you earn within your personal 12-months period. You will automatically be upgraded to Silver Member if you earn 20.000 Basis Points during your earning period.

If you earn 50.000 Basis Points in the period, you become a Gold Member. The earning period runs from the first day in the joining month and 12 months forward.

Some of the test conditions that which can be extracted from this are:

1) When the sum of basis points is less than 20.000, the member status is Basis.

2) When the sum of basis points is equal to or greater than 20.000, the member level is set to Silver.

3) When the sum of basis points is equal to or greater than 50.000, the member level is set to Gold.

There are many more—and just as many questions to be posed!

Activities in Test Analysis and Design Cont.

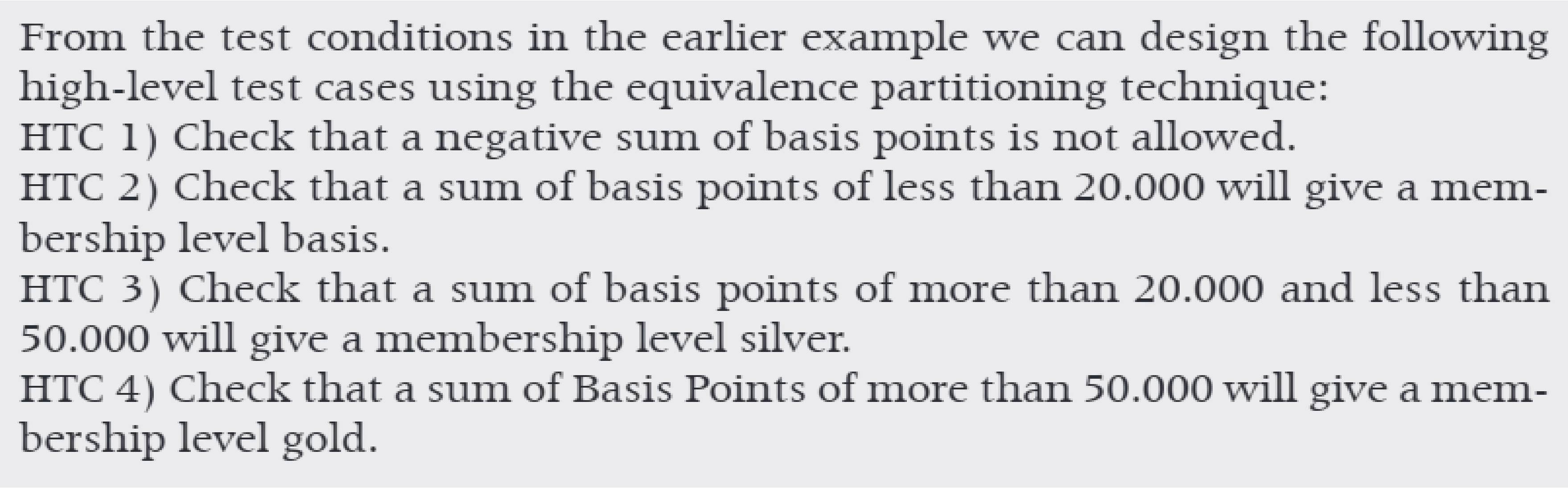
❖ Creation of Test Cases

- Based on the test conditions, we can now produce our first high-level test cases and subsequently low-level test cases.
  - ➢ A high-level test case is a test case without specific values for input data and expected results, but with logical operators or defining in general terms.
- The test cases we design should strike the best possible balance between being:
  - ➢ Effective: Have a reasonable probability of detecting errors
  - ➢ Exemplary: Be practical and have a low redundancy
  - ➢ Economic: Have a reasonable development cost and return on investment
  - ➢ Evolvable: Be flexible, structured, and maintainable
- The test case design techniques make it possible to create test cases that satisfy these demands.

Activities in Test Analysis and Design Cont.

- The documentation of a *test case* at this stage must at least include:
  - Unique identification
  - Description
  - References to test condition(s) on which the test case is based and to test design(s) to which the test case belongs.

- For example：high-level test cases



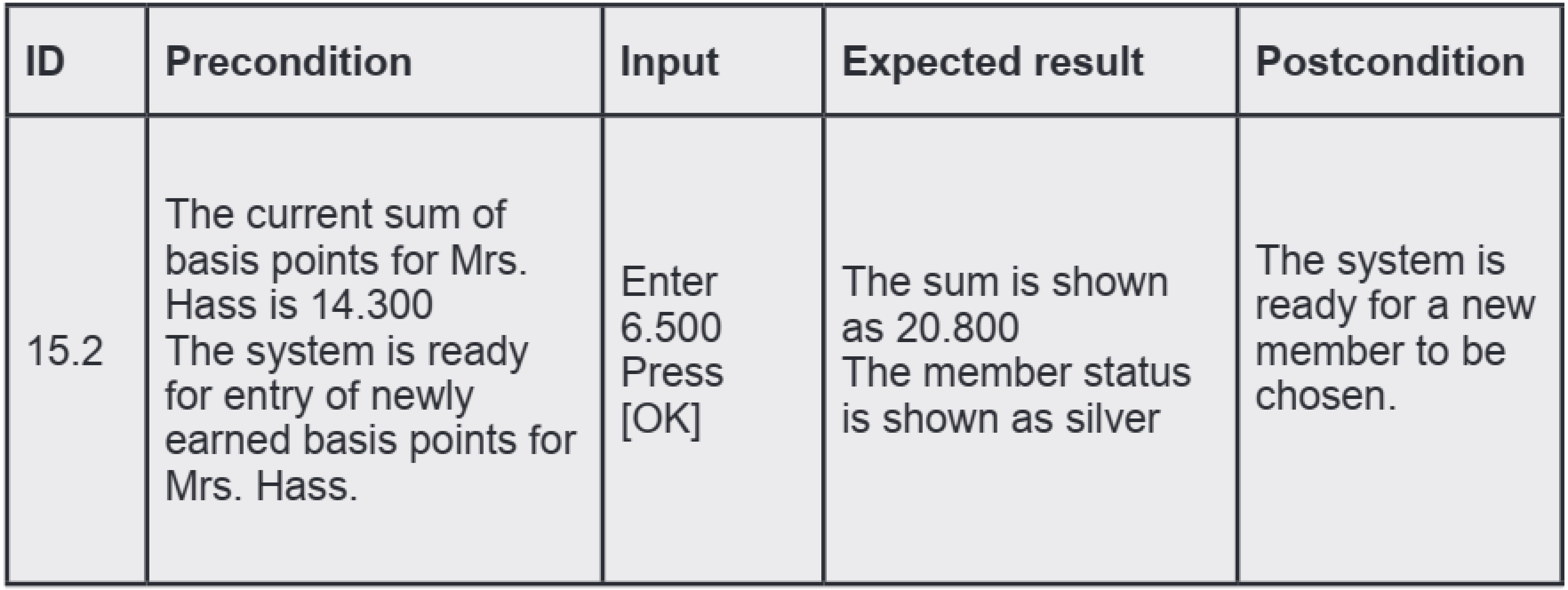
From the test conditions in the earlier example we can design the following high-level test cases using the equivalence partitioning technique:
HTC 1) Check that a negative sum of basis points is not allowed.
HTC 2) Check that a sum of basis points of less than 20.000 will give a membership level basis.
HTC 3) Check that a sum of basis points of more than 20.000 and less than 50.000 will give a membership level silver.
HTC 4) Check that a sum of Basis Points of more than 50.000 will give a membership level gold.

Activities in Test Analysis and Design Cont.

- From the high-level test cases we go on to define the low-level test cases.
- It is not always possible to execute all the test cases we have identified;
  - The actual test cases to be executed must be selected based on the risk analysis.
- A low-level test case is a test case with specific values defined for both input and expected result.
- The documentation of a low-level test case must at least include:
  - Unique identification
  - Execution preconditions
  - Inputs: data and actions
  - Expected results including post conditions
  - Reference(s) to test conditions and/or directly to basis documentation

Activities in Test Analysis and Design Cont.

- One low-level test case created from the list of these high-level test cases could be:



ID	Precondition	Input	Expected result	Postcondition
15.2	The current sum of basis points for Mrs. Hass is 14.300 The system is ready for entry of newly earned basis points for Mrs. Hass.	Enter 6.500 Press [OK]	The sum is shown as 20.800 The member status is shown as silver	The system is ready for a new member to be chosen.

- The expected result must be determined from the basis documentation where the expectations for the coverage items are described.
- you must never, ever just guess or assume.

Requirements

- All product development starts with the requirements. The higher level testing is done directly against requirements.
- The lower level testing is done against design that is based on the requirements.
  - All testing is hence based on the requirements.

Requirement Levels

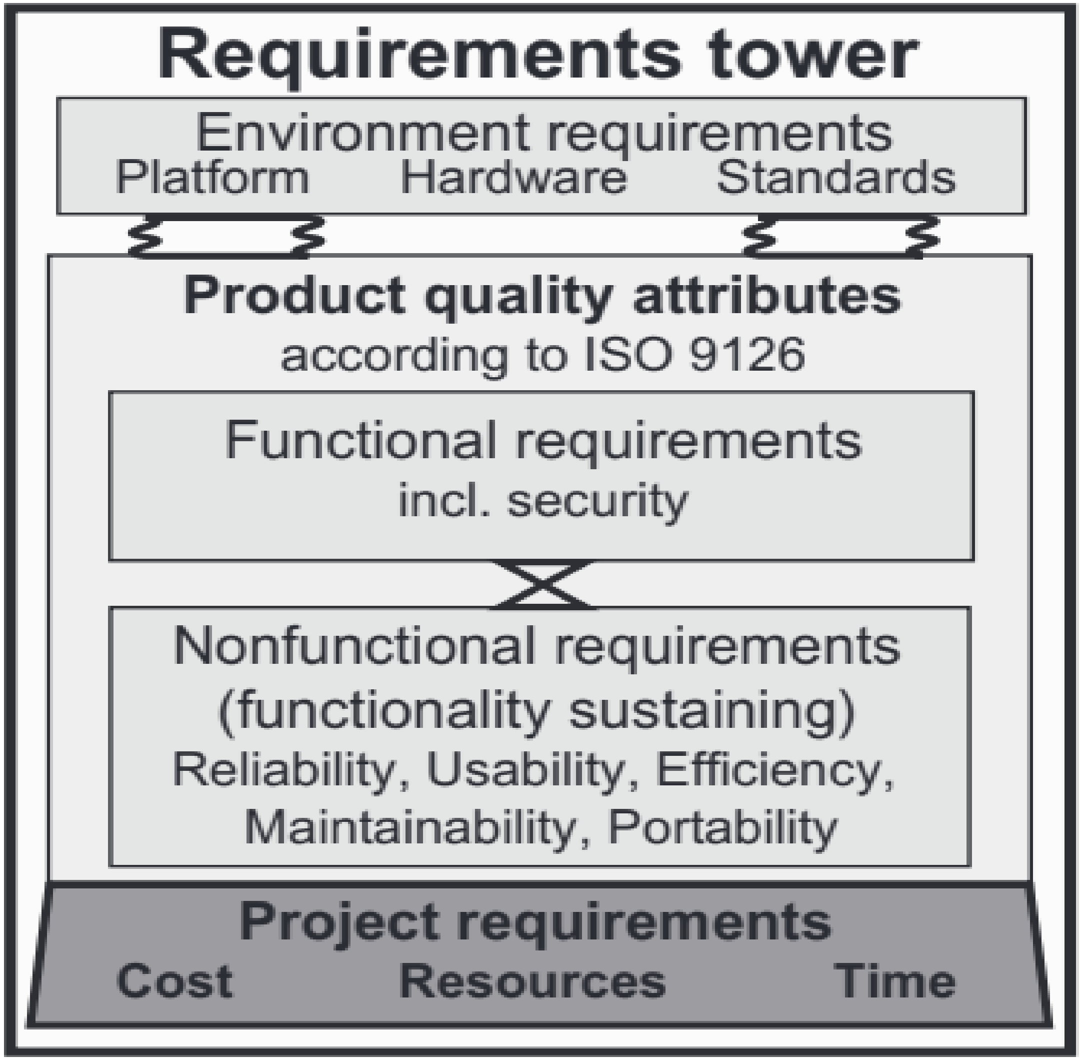
- Requirements should exist at different levels, for example:
  - Business requirements
  - User requirements
  - System requirements

Requirements Cont.

- ➢ Business requirements:-The organization and top management "speak" money: they express business requirements.
  - ✓ Business requirements may be tested, but most often they are not tested explicitly.
- ➢ User requirements:-he users speak "support of my work procedures"
  - ✓ User requirements are tested in the acceptance testing.
- ➢ System requirements:-product is split up in a software system and a hardware system.
  - ✓ they are tested in the system testing.

Requirements Cont.

- Requirement Types



Requirements tower

Environment requirements
Platform Hardware Standards

Product quality attributes
according to ISO 9126

Functional requirements
incl. security

Nonfunctional requirements
(functionality sustaining)
Reliability, Usability, Efficiency,
Maintainability, Portability

Project requirements
Cost Resources Time

Requirements Cont.

- Requirement Styles

- Requirements can be expressed in many ways. Typical styles are: statements, tasks, models and tables

  - Statements:-The most common style

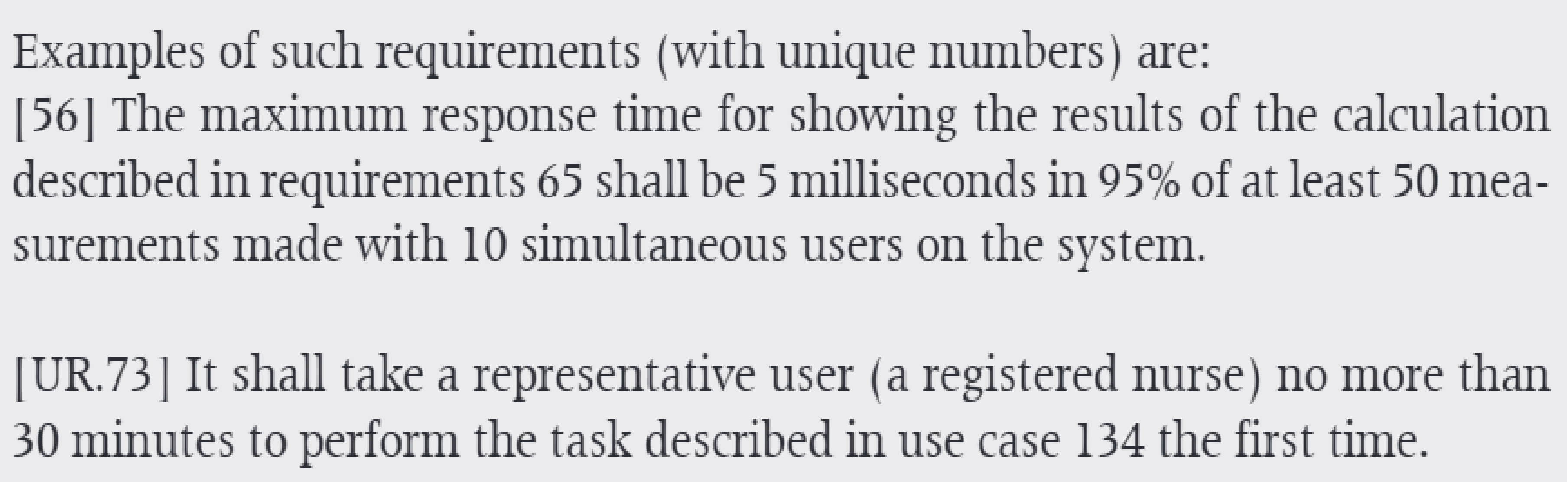
  - ✓ each requirement is expressed as a single (or very few) sentences in natural language.

  - ✓ To make statement requirements more precise and testable we can use metrics and include information such as

    - the scale to use,
    - the way to measure,
    - the target, and
    - maybe acceptable limits.

  - ✓ This is especially important for nonfunctional requirements!

Requirements Cont.



Examples of such requirements (with unique numbers) are:
[56] The maximum response time for showing the results of the calculation described in requirements 65 shall be 5 milliseconds in 95% of at least 50 measurements made with 10 simultaneous users on the system.

[UR.73] It shall take a representative user (a registered nurse) no more than 30 minutes to perform the task described in use case 134 the first time.

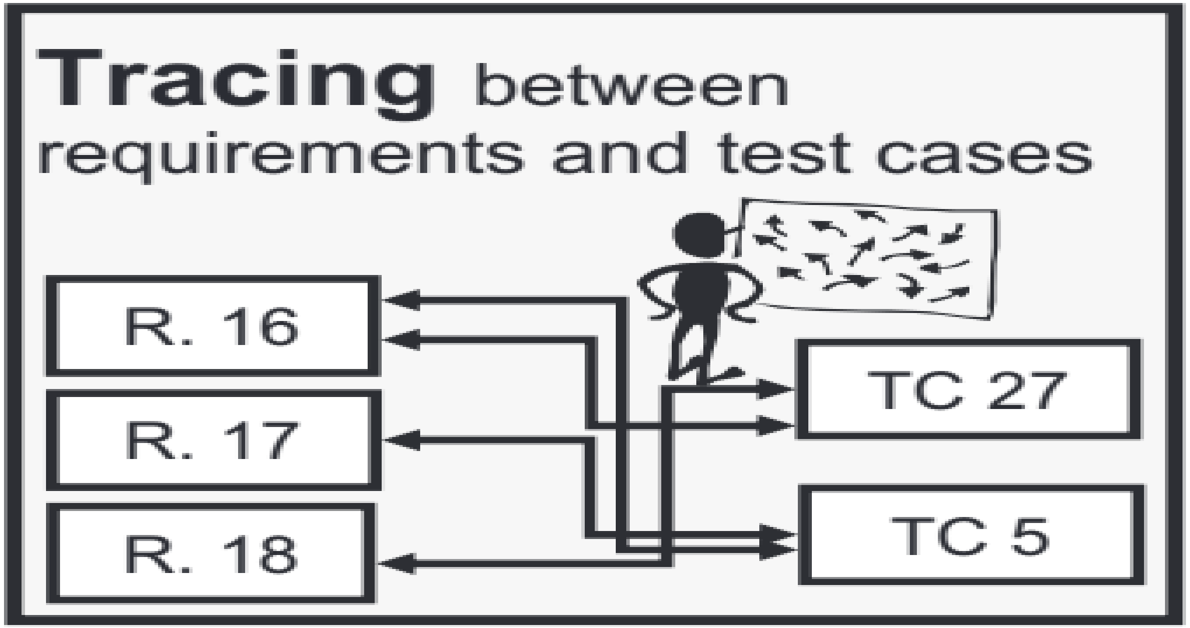
- A task is a series of actions to achieve a goal.
  - ✓ Task styles may be stories, scenarios, task lists, or use cases.
  - ✓ Requirements expressed in these ways are easy to understand, and they are typically used to express user requirements.
  - ✓ They are easy to derive high-level test cases and procedures from.

Requirements Cont.

- A model is a small representation of an existing or planned object.
  - ✓ Model styles may be domain models, prototypes, data models, or state machines.
- A table is a compact collection and arrangement of related information.
  - ✓ Tables may be used for parameter values, decision rules, or details for models.
- The collection of requirements for each level documented in the requirement specification is in fact a model of the product or the system.
- This model is the one the test is based on.

Traceability

- References are an important part of the information to be documented in the test specification.
  - A few words are needed about these.
- There are two sets of references:
  - References between test specification elements
  - References from test specification elements to basis documentation



Metrics for Analysis and Design

- Metrics to be defined for the monitoring and control of the test analysis and design activities may include:
  - Number of specified test conditions and high-level requirements over time
  - Coverage achieved in the specification (for example, for code structures, requirements, risks, business processes), over time.
  - Number of defects found during analysis and design
  - Other tasks commenced and completed over time, for example, in connection with test environment specifications
  - Time spent on each task over time

Test Implementation and Execution

- The purpose of the test implementation is to organize the test cases in procedures and/or scripts and to perform the physical test in the correct environment.

  - ➢ The inputs on which this process is based on:

    I

    - ✓ Level test plan
    - ✓ Test conditions and test design
    - ✓ Other relevant documents
    - ✓ The test object

  - ➢ The activities are:

    P

    - ✓ Organizing test procedures
    - ✓ Design and verify the test environment
    - ✓ Execute the tests
    - ✓ Record the testing
    - ✓ Check the test results

Test Implementation and Execution Cont.

- The output consists of:
  - ✓ Test specification
  - ✓ Test environment
  - ✓Test logs
  - ✓ Incident reports
  - ✓ Tested test object

Input to Test Implementation and Execution

- The input from the level test plan that we need for this process is:
  - Scheduling and staffing for the activities
  - Definition of the test object(s)
  - Specification of test environment
  - Entry criteria for the test execution
  - Exit criteria, including coverage criteria
- the test specification in its current state.
- We might need other documentation like a user manual, documentation of completion of preceding test work, and logging sheets.

Documentation of Test Implementation and Execution

- The test specification is finished in this process where the test procedures are laid out.
- The requirements concerning the test environment are finalized.
- The test environment must be established before the test execution may start.
- The test execution is documented in test logs.
- When failures occur these should be documented in incident reports..

Activities in Test Implementation and Execution

- Organizing Test Procedures
- Test Environment Specification and Testing
- Checking Execution Entry Criteria
- Test Execution
- Identifying Failures
- Test Execution Logging
- Confirmation Testing and Regression Testing
  - *Confirmation Testing*
  - *Regression Testing*

Activities in Test Implementation and Execution Cont.

Organizing Test Procedures

- The low-level test cases should now be organized and assembled in test procedures and/or test scripts.
  - The term “procedure” is mostly used when they are prepared for manual test execution.
  - the term “script” is mostly used for automatically executable procedures.
- The degree of detail in the procedures depends on who will be executing the test.

Activities in Test Implementation and Execution Cont.

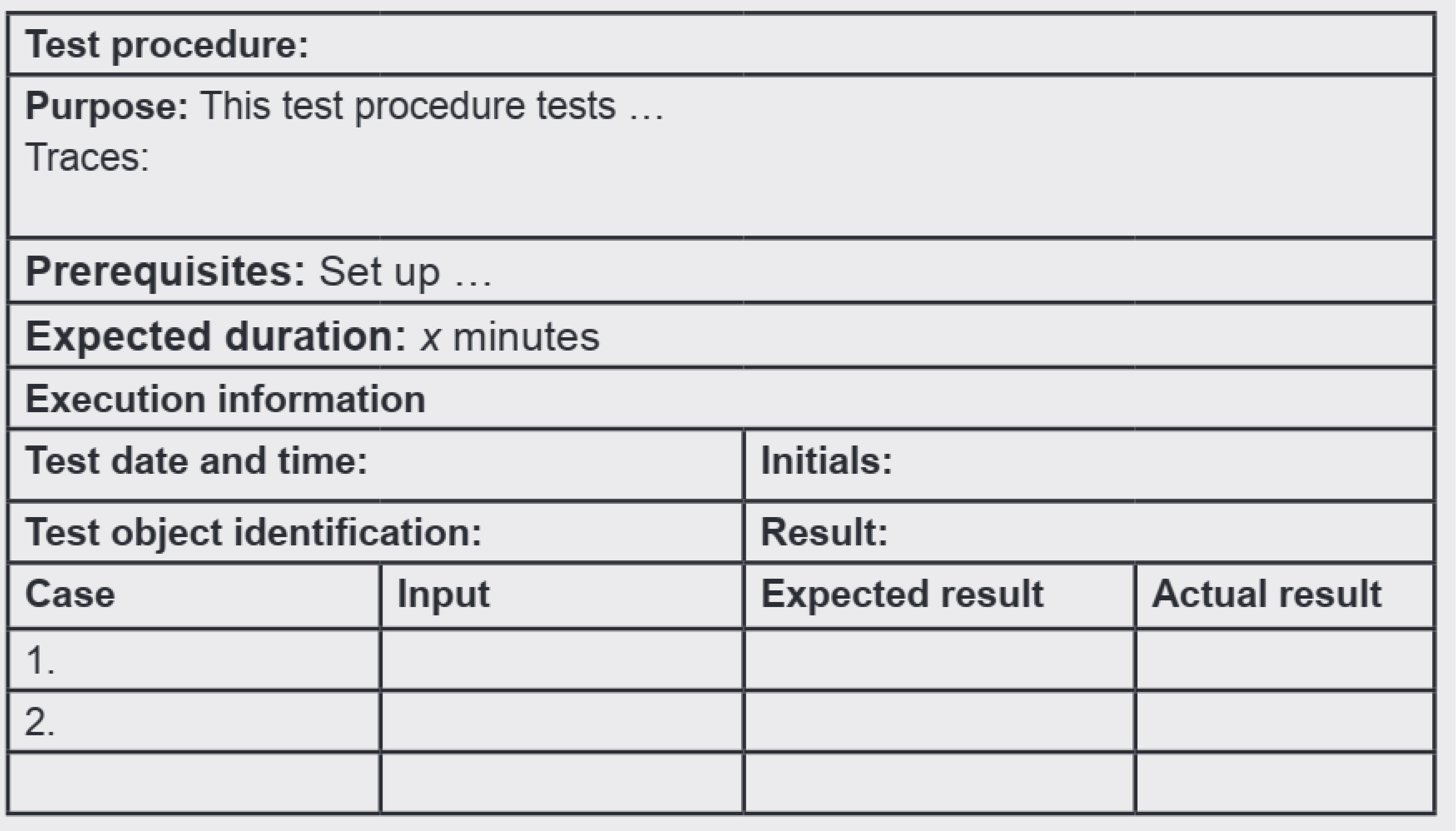
- The documentation of a test procedure must at least include:
  - Unique identification
  - Description
  - References to high-level test cases and/or to test conditions and/or directly to basis documentation to be covered by the procedure
  - An explicit description of the preconditions to be fulfilled before the actual test execution can start Included low-level test cases

Activities in Test Implementation and Execution Cont.

- The organization in test procedures could be looked at as the execution schedule.
  - ➢ It could be fixed, but it could also be dynamic. For specific purposes.
  - ✓ especially for regression testing, some of the test procedures may be selected and reorganized in other execution schedules that fit the specific purpose.
- A test procedure should not include too many or too few test cases
  - ➢ A maximum of 20 test cases
  - ➢ A minimum of 2–4 test cases is a good rule of thumb.
- The test procedure may also include facilities for logging the actual execution of the procedure

Activities in Test Implementation and Execution Cont.

- Here is an example of a template for a test procedure templet.
  - Test procedure: n.n (n)



Test procedure:			
Purpose: This test procedure tests … Traces:			
Prerequisites: Set up …			
Expected duration: *x* minutes			
Execution information			
Test date and time:		**Initials:**	
Test object identification:		**Result:**	
Case	**Input**	**Expected result**	**Actual result**
1.			
2.			

Activities in Test Implementation and Execution Cont.

- Note that the template indicates a unique identification of the test procedure (n), and a number indicating its position among all the other test procedures (n.n).
- To facilitate estimation the test designer is required to provide an estimate of the execution time for manual execution of the test procedure.

Activities in Test Implementation and Execution Cont.

Quality Assurance of the Test Specification

- Before the test specification is used in the test execution it should be reviewed.
- The review should ensure that the test specification is correct with respect to the test basis, including
  - any standards,
  - that it is complete with respect to the required coverage, and
  - that it can be used by those who are going to execute the test.

Activities in Test Implementation and Execution Cont.

- The review may be guided by a checklist, for example :
  - Is the test specification clear and easily understood?
  - Is the test structure compatible with automated test?
  - Is it easy to maintain?
  - Is it easy for others to perform a technical review?

Activities in Test Implementation and Execution Cont.

- Test Environment Specification and Testing

- The test environment is a necessary prerequisite for the test execution.
- The environment is first outlined in the test plan based on the strategy.
  - Test plan also describes by whom and when the test environment is to be created and maintained.
  - Some additional requirements for the environment may be specified in the test specification in the form of prerequisites for the test procedures, and especially for test data.
- The description of the test environment must be as specific as possible in order to get the right test environment established at the right time (and at the right cost).

Activities in Test Implementation and Execution Cont.

- The descriptions of the test environment must cover:
  - Hardware: on run on and/or to interface with
  - Software: to the test platform and other applications
  - Peripherals (printers including correct paper, fax, CD reader /burner)
  - Network: provider agreements, access, hardware, and software Tools and utilities
  - Data: actual test data, security, and rollback facilities.
  - Other aspects: security, load patterns, timing, and availability
  - Physical environment (room, furniture, conditions)
  - Communication (phones, Internet, paper forms, paper, word processor)
  - Sundry (paper, pencils, coffee, candy, fruit, water)

Activities in Test Implementation and Execution Cont.

- The testers should be verified that the test environment is complete according to the specifications and it works correctly before you start to execute the test procedures.

❖ Checking Execution Entry Criteria

- The tester to make sure that the execution entry criteria are fulfilled.
- If the test object has not passed the entry criteria defined for it, do not start the test execution.
- The test executors must be appropriately trained, and any stakeholders needed, for example, customers to witness the execution, must be present and briefed.

Activities in Test Implementation and Execution Cont.

- The configuration management is working well.
- The testing process and the configuration management process, including:
  - The ability to get the correct version of the test object, the test specification, and the ability to get the correct versions of any other necessary material.
  - The ability to be able to report the failures and other incidents found during the testing
  - The ability to follow the progress of the failures and plan any necessary confirmation testing and regression testing
  - The ability to register approval of successful removal of failures.

Activities in Test Implementation and Execution Cont.

- Test Execution

- The execution of the tests is what everybody has been waiting for: the moment of truth!
- During test execution is to follow the test specification and register all incidents on the way.
- If the execution is done by a tool, this is exactly what will happen

Activities in Test Implementation and Execution Cont.

- We have taken great care in writing the test procedures, and it is important to follow them.
  - We need to be able to
    - ✓ trust that the specified testing has actually been executed.
    - ✓ collect actual time spent and compare it with the estimates to improve our estimation techniques.
    - ✓ compare the progress with the plan.
    - ✓ repeat the tests exactly as they were executed before for the sake of confirmation testing and regression testing.
  - It should be possible to make a complete audit of the test.

Activities in Test Implementation and Execution Cont.

- Identifying Failures

- For each test case we execute the actual result should be logged and compared to the expected result.
  - a tick mark, √, is sufficient to indicate when the actual result matched the expected result.
  - If the actual outcome does not comply with the expected outcome we have a failure on our hands.

- Test Execution Logging

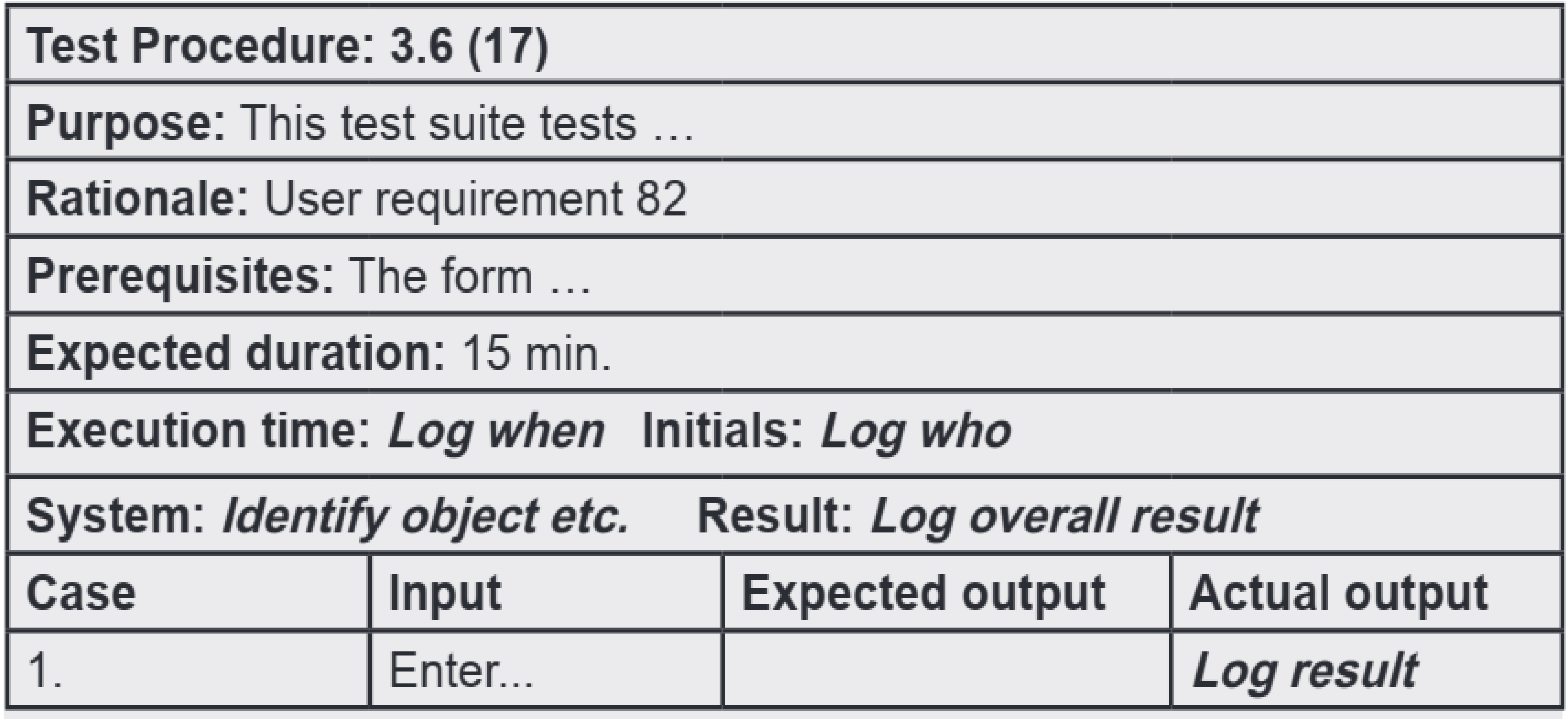
- As we execute, manually or by the use of tools, we must log what is going on.
- We must record the precise identification of what we are testing and the test environment and test procedures we use.

Activities in Test Implementation and Execution Cont.

- We must also log the result of the checking
- we must log any significant event that has an effect on the testing.
- The IEEE 829 standard suggests the following contents of a test log:
- Test log identifier
  - ➢ Description of the test
  - ➢ Activity and event entries

Activities in Test Implementation and Execution Cont.

- An example of this is shown here



Test Procedure: 3.6 (17)			
Purpose: This test suite tests …			
Rationale: User requirement 82			
Prerequisites: The form …			
Expected duration: 15 min.			
Execution time: ***Log when*** **Initials:** ***Log who***			
System: ***Identify object etc.*** **Result:** ***Log overall result***			
Case	**Input**	**Expected output**	**Actual output**
1.	Enter...		***Log result***

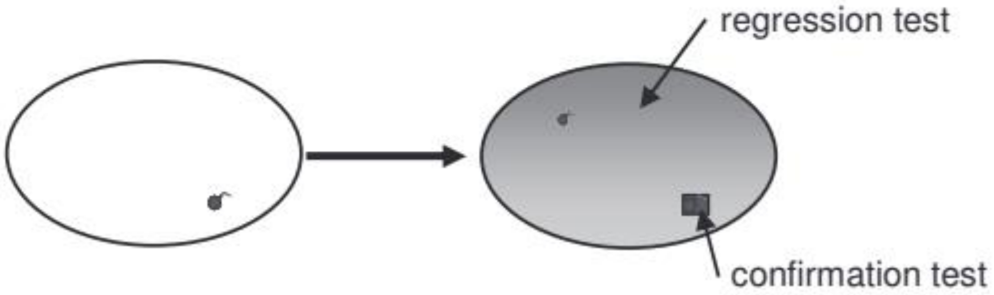
Activities in Test Implementation and Execution Cont.

- Confirmation Testing and Regression Testing

- During testing we get failures.
- In most cases the underlying defects are corrected and the corrected test object is handed over to the testers for confirmation.
- This is the situation where we iterate in the test process and go back to the test execution process.
- We go back to perform confirmation testing and regression testing.
- They can appear in all the test levels from component testing to acceptance testing and during maintenance of a product in operation.

Activities in Test Implementation and Execution Cont.

- These two types of change-related testing
- they are executed after defect correction.



Activities in Test Implementation and Execution Cont.

Confirmation testing

- Confirmation testing is the first to be performed after defect correction.
- It is done to ensure that the defect has indeed been successfully removed.

Regression Testing

- Regression testing is repetition of tests that have already been performed without problems to ensure that defects have not been introduced or uncovered as a result of the change.
  - ➢ External or environmental changes
  - ✓ Example : operating system ,database administration system

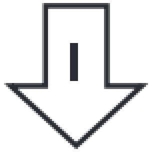
Metrics for Implementation and Execution

- Metrics to be defined for implementation and execution testing activities may include:
  - ➢ Number of created test
    - ✓ environments over time
    - ✓ data over time
    - ✓ procedures over time
  - ➢ Number of initiated
  - ➢ Number of passed
  - ➢ Number of failed

    test procedures over time
  - ➢ Number of passed confirmation tests over time.
  - ➢ Number of test procedures run for regression testing over time
  - ➢ Time spent on the various tasks

Evaluating Exit Criteria and Reporting

- Test execution, recording, control, retesting, and regression testing must be continued until the exit criteria have been achieved.
- The inputs on which this process is based are:



  - Test plan
  - Measurements from the test development and execution processes
- The activities are:

  - Comparing actual measurements with estimates and planned values
  - Reporting test results

Evaluating Exit Criteria and Reporting

- The output consists of:
  - Presentation of test progress
  - Test report

Input to Test Progress and Completion Reporting

- The input from the level test plan that we need for this process is:
  - Scheduling and staffing for the activities
  - Exit criteria

Documentation of Test Progress and Completion Reporting

- The documentation of the progress must be presented in various ways according to who is receiving it.
- The audience may be the customer, higher management, project management and participants, and testers.
- At the completion of each test level a test summary report should be produced.
- The ultimate documentation of completion is the final test summary report for the entire test assignment.

Activities in Test Progress and Completion Reporting

❖ Checking for Completion

- A check against the test exit criteria is mandatory before the testing is completed at any level.
- To warrant a stop it is important to ensure that the product has the required quality.
- The exit criteria are tightly connected to the coverage items for the test.
- The test case design techniques used, and the risk of the product.
- The exit criteria therefore vary from test level to test level

Activities in Test Progress and Completion Reporting Cont.

- Examples of exit criteria are:
  - Specified coverage has been achieved
  - Specified number of failures found per test effort has been achieved
  - No known serious faults
  - The benefits of the system as it is bigger than known problems
- Any changes to the test completion criteria must be documented.

Activities in Test Progress and Completion Reporting Cont.

- When all test completion criteria are met and the report approved, the test object can be released.
  - When the test is a static test the test object (usually a document) can be released to be used as the basis for further work.
  - When the test is a test level for dynamic test the test object is progressively released from one test level to the next.
  - Ultimately the product can be released to the customer.

Metrics for Progress and Completion Reporting

- It includes:
  - Number of tasks commenced over time
  - Task completion percentage over time
  - Number of task completed over time
  - Time spent on each task over time

Test Closure

- The purpose of the test closure activities is to consolidate experience and place test ware under proper control for future use.
- The inputs on which this process is based are:

I

  - Level test plan
  - Test ware, including test environment
- The overall procedure consists of the activities:

P

  - Final check of deliveries and incident reports
  - Secure storage/handover of test ware
  - Retrospection

Test Closure Cont.

- The output generated in this process is:
  - Test experience report
  - Configuration management documentation

Input to Test Closure Cont.

- The input from the test plan that we need for this process is:
  - Scheduling and staffing for the activities
  - Planned deliveries

Documentation of Test Closure

- an experience report or a retrospective report from the retrospective meeting.
- the organization’s and customer’s configuration management system.

Activities in Test Closure Cont.

❖ Check Completion Again

- Before we definitively close the door to the testing assignment we need to make extra sure that we have met the part of the exit criteria.
- This is both in terms of test coverage and deliveries we are to produce.

❖ Delivering and Archiving Test Ware

- For the sake of easy and economically sound future testing in connection with defect correction and development of new versions of the product we should keep the assets we have produced

Activities in Test Closure

- *It is a waste of time and money not to keep the test ware we have produced.*

❖ **Retrospective Meeting**

- The last thing we have to do is to report the experiences we have gained during our testing.
- This is also a very valuable activity since the results of the testing can be the main indicators of where processes need to be improved.
  - development processes
  - support processes
  - test process

Metrics for Test Closure Activities

- Metrics to be defined for these activities may include
  - number of tasks commenced over time,
  - task completion percentage over time,
  - number of tasks completed over time,
  - time spent on each task over time

as for the other processes.

Thank You !!!