Planning & structuring the testing of a large program:

- Defining the process
  - Test plan
  - Input and output documents
- Who is testing?
  - Developers / special testing teams / customer
- What test levels do we need?
  - Unit, integration, system, acceptance, regression
- Order of tests
  - Top-down, bottom-up, combination
- Running the tests
  - Manually
  - Use of tools
  - Automatically

Who tests

Any significant project should have a separate QA team
Why: the almost infinite human propensity to self-delusion

Unit tests: the developers
  - My suggestion: pair each developer with another who serves as “personal tester”
Integration test: developer or QA team
System test: QA team
Acceptance test: customer & QA team

Classifying reports: by severity

Classification must be defined in advance
Applied, in test assessment, to every reported failure
Analyzes each failure to determine whether it reflects a fault, and if so, how damaging
Example classification (from a real project):
  - Not a fault
  - Minor
  - Serious
  - Blocking

Classifying reports: by status

From a real project:
  - Registered
  - Open
  - Re-opened
  - Corrected
  - Integrated
  - Delivered
  - Closed
  - Not retained
  - Irreproducible
  - Cancelled

Assessment process (from real project)
Some responsibilities to be defined

- Who runs each kind of test?
- Who is responsible for assigning severity and status?
- What is the procedure for disputing such an assignment?
- What are the consequences on the project of a failure at each severity level? (e.g., “the product shall be accepted when two successive rounds of testing, at least one week apart, have evidenced fewer than m serious faults and no blocking faults”).

IEEE-829-conformant test elements

- Test plan:
  - Prescribes scope, approach, resources, & schedule of testing. Identifies items & features to test, tasks to perform, personnel responsible for each task, and risks associated with plan.
- Test specification documents:
  - Test design specification: identifies features to be covered by tests, constraints on test process
  - Test case specification: describes the test suite
  - Test procedure specification: defines testing steps
- Test reporting documents:
  - Test item transmittal report
  - Test log
  - Test incident report
  - Test summary report

IEEE 829: Test plan structure

- a) Test plan identifier
- b) Introduction
- c) Test items
- d) Features to be tested
- e) Features not to be tested
- f) Approach
- g) Item pass/fail criteria
- h) Suspension criteria and resumption requirements
- i) Test deliverables
- j) Testing tasks
- k) Environmental needs
- l) Responsibilities
- m) Staffing and training needs
- n) Schedule
- o) Risks and contingencies
- p) Approvals

Test planning: IEEE 829


Can be found at: [http://tinyurl.com/35pcp6](http://tinyurl.com/35pcp6)

For an overview, see the Wikipedia entry

IEEE 829-conformant test elements

- Test design/Test case specification

- The test design specification details:
  - Test conditions
  - Expected results
  - Test pass criteria

- The test case specification details:
  - Specifies the test data for use in running the test conditions identified in the test design specification
The test procedure specification details:
- How to run each test, including any set-up preconditions and the steps that need to be followed

The test item transmittal report:
- Reports on when tested software components have progressed from one stage of testing to the next

The test log records:
- which tests cases were run, who ran them, in what order, and whether each test passed or failed

The test incident report details:
- for any test that failed, the actual versus expected result, and other information intended to throw light on why a test has failed that will help in its resolution. The report will also include, if possible, an assessment of the impact upon testing of an incident.

A management report providing any important information uncovered by the tests accomplished, and including assessments of the quality of the testing effort, the quality of the software system under test, and statistics derived from Incident Reports.

The report also records what testing was done and how long it took, in order to improve any future test planning.

This final document is used to indicate whether the software system under test is fit for purpose according to whether or not it has met acceptance criteria defined by project stakeholders.

Consider a small library database with the following transactions:
1. Check out a copy of a book.
2. Return a copy of a book.
4. Get the list of books by a particular author or in a particular subject area.
5. Find out the list of books currently checked out by a particular borrower.
6. Find out what borrower last checked out a particular copy of a book.

There are two types of users: staff users and ordinary borrowers.

Transactions 1, 2, 4, and 5 are restricted to staff users, except that ordinary borrowers can perform transaction 4 to find out the list of books currently borrowed by themselves. The database must also satisfy the following constraints:
- All copies in the library must be available for checkout or be checked out.
- No copy of the book may be both available and checked out at the same time.
- A borrower may not have more than a predefined number of books checked out at one time.

Source*: Wing 88