ISTQB – Foundation Level

CHAPTER 5: TEST MANAGMENT

Prepared by: Vu Nguyen
Feb 2010
Updated by: Minh Ha
Dec 2012

AGENDA

- 5.1 Test organization (K2)
- 5.2 Test planning and estimation (K2)
- 5.3 Test progress monitoring and control (K2)
- 5.4 Configuration management (K2)
- 5.5 Risk and testing (K2)
- 5.6 Incident Management (K3)

5.1 Test organization (K2)

- Objectives
  - LO-5.1.1 Recognize the importance of independent testing. (K1)
  - LO-5.1.2 List the benefits and drawbacks of independent testing within an organization. (K2)
  - LO-5.1.3 Recognize the different team members to be considered for the creation of a test team.
  - (K1)
  - LO-5.1.4 Recall the tasks of typical test leader and tester. (K1)

Independent and integrated testing

- The effectiveness of finding defects can be improved by using independent testers.
- Options for independence are:
  - No independent testers. Developers test their own code.
  - Independent testers within the development teams.
  - Independent test team or group within the organization.
  - Independent testers outsourced or external to the organization.

Independent and integrated testing

- Other Options for independence are:
  - Independent testers from the business organization or user community.
  - Independent test specialists such as usability testers, security testers or certification testers.

Independent and integrated testing

Benefits of independence	Drawbacks
○ Independent testers see different defects, and are unbiased. ○ An independent tester can verify assumptions people made during specification and implementation of the system. ○ More credibility in the organization	○ Isolation from the development team ○ Independent testers may be the bottleneck as the last checkpoint. ○ Developers may lose a sense of responsibility for quality.

Component testing

- Testing should be performed in close conjunction with the development activities
- Independent testing organized like model 1 would certainly improve testing quality
- Management sets testing standards and rules, prepares testing schedules, and requires test logs from the developers
- Testing specialists should, at least temporarily, be called in as coaches

Integration testing

- Can be organized analogous to component testing if same team performs and tests integration
- Components originating from several teams:
  - A mixed integration team is required
  - Or an independent integration team should be responsible

System testing

- Final product shall be considered from viewpoint of customer/end user
- Independence from the development is crucial
- Only models 3, 4 as professionally acceptable choices

Working as a test leader

- Test leader : Plan, monitor, and control
  - Devise the test objectives, test policies, test strategies and test plans
  - Estimate the test and resources.
  - Plan for test automation if applicable.
  - Schedule, lead/guide, monitor, measure and evaluate its quality.
  - Ensure configuration management.
  - Report test progress, summary report.
  - Take action to compensate for problems.

Working as a tester

- Tester:
  - Review and contribute to test plan.
  - Analyze, review and assess requirements/design specifications.
  - Create test specifications.
  - Setup test environment
  - Execute and log the tests, evaluate results and document problems found.
  - Automate tests.
  - Review tests developed by others.

Defining the skills test staff need

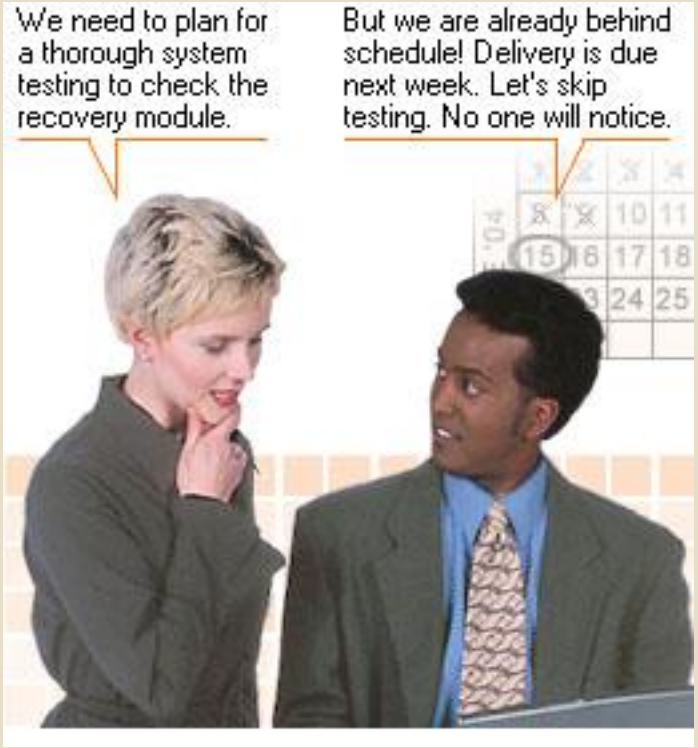
- Application or business domain
- Technology
- Testing

5.2 Test planning and estimation (K2)

- Objectives
  - LO-5.2.1 Recognize the different levels and objectives of test planning. (K1)
  - LO-5.2.2 Summarize the purpose and content of the test plan, (IEEE 829). (K2)
  - LO-5.2.3 Differentiate between conceptually different test approaches(K2)
  - LO-5.2.4 Differentiate between the subject of test planning for a system and for scheduling test execution. (K2)

Introduction of planning

- Planning is required to achieve success.
- A plan describes a management approach, which, if followed, will ensure that the project achieves its objectives.



Quality can be compromised if testing is not planned for and implemented.

What is project success?

- Project management activities should aim at ensuring that the project meets the following.
  - It is completed within the scheduled time.
  - It is completed within the budgeted cost.
  - It results in a good-quality product that fulfills customer requirements.

Test planning (K2)

- Defining overall approach and strategy for testing
- Deciding about the test environment
- Definition of the test levels
- Deciding how to evaluate the test results
- Selecting metrics for monitoring and controlling test work, defining test exit criteria
- Determining how much test documentation shall be prepared and deciding about templates
- Writing the test plan
- Estimating test effort and test costs

Test planning (K2)

- Test Plan is a document describing the scope, approach, resources and schedule of intended test activities.

Project management involves project planning, monitoring, and control to ensure project success.

IEEE 829 Standard Test Plan Template

- Test Plan Identifier
- Introduction
- Test Items
- Features to be tested
- Features not to be tested
- Approach
- Item pass/fail criteria
- Suspension and resumption criteria
- Test Deliverables
- Test Tasks
- Environmental needs
- Responsibilities
- Staffing and training needs
- Schedule
- Risks and contingencies
- Approvals

Exit criteria (K2)

- The purpose of exit criteria is to define when to stop testing.
- Typically exit criteria may consist of:
  - Thoroughness measures, such as coverage of code, functionality or risk.
  - Estimates of defect density or reliability measures.
  - Cost.
  - Residual risks, such as defects not fixed or lack of test coverage in certain areas.
  - Schedules such as those based on time to market.

Test estimation (K2)

- Two approaches for the estimation of test effort are covered in this syllabus:
  - The metrics-based approach: estimating the testing effort based on metrics of former or similar projects or based on typical values.
  - The expert-based approach: estimating the tasks by the owner of these tasks or by experts.

Test estimation (K2)

- The testing effort may depend on a number of factors, including:
  - **Maturity** of the software development process
  - **Testability** of the software
  - **Test infrastructure**
  - **Qualification of employees**
  - **Quality requirements**
  - **Test approach and strategy**

Test Approach (Test Strategy)

- Defines the project's testing objectives and means to achieve them
- Determines testing effort and costs
- Selecting appropriate test strategy is one of the most important planning task decisions of test manager
- Goal is to choose test approach that optimizes the relation between costs of testing and costs of defects as well as minimizes the risk
- Test costs should be less than the costs of defects and deficiencies in the final product

Preventative vs. Reactive Approach

- Preventive approaches:
  - Testers are involved from the beginning
  - Test manager can optimize testing and reduce testing costs
  - The use of the general V-model with emphasis on design reviews, will contribute a lot to prevent defects
  - Reduced defect density during test execution
  - Especially, in safety critical software, a preventive approach may be mandatory.

Preventative vs. Reactive Approach

- Reactive approaches:
  - Testers are involved (too) late
  - When a preventive approach cannot be chosen
  - Test manager must react appropriately
  - "Exploratory testing" strategy can be used:
    - Tester "explores" the test object and the test design
    - Execution and evaluation occurs nearly concurrently

Analytical vs. Heuristic Approach

- During test planning and test design, TM may use different sources of information
- Analytical approach:
  - Test planning is founded on data and (mathematical) analysis of these data
  - Amount and intensity of testing are chosen based on individual or multiple parameters (costs, time, coverage, etc.)
- Heuristic approach:
  - Test planning is founded on experience of experts and/or on rules of thumb.
  - When no data is available

Analytical vs. Heuristic Approach

- Approaches used in practice use both analytical and heuristic elements
  - Model-based testing
  - Statistical or stochastic (model-based) testing
  - Risk-based testing
  - Process- or standard-compliant approaches
  - Reuse-oriented approaches
  - Checklist-based (methodical) approaches
  - Expert-oriented approaches
- The above approaches are seldom used stand-alone
- TM uses a combination of several approaches

5.3 Test progress monitoring and control

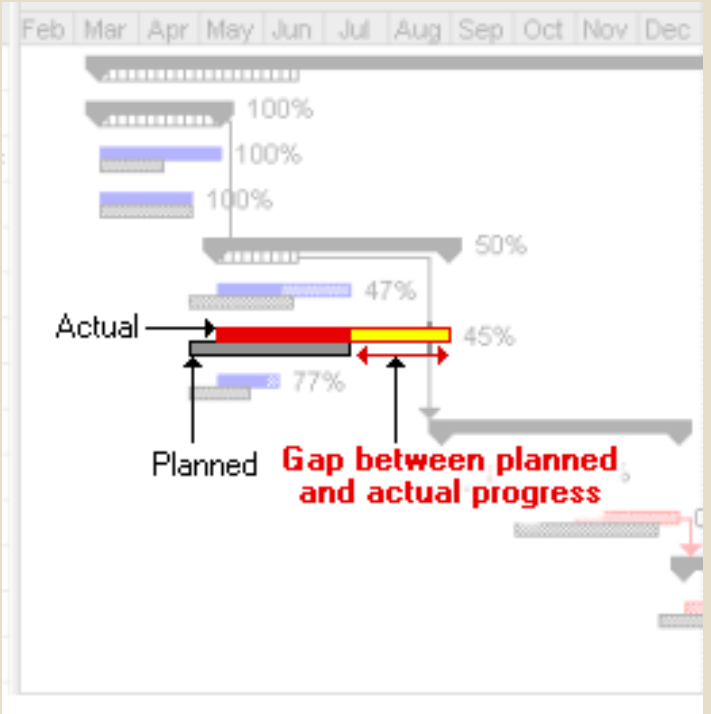
- Objectives
  - LO-5.3.1 Recall common metrics used for monitoring test preparation and execution. (K1)
  - LO-5.3.2 Understand and interpret test metrics for test reporting and test control (e.g. defects found and fixed, and tests passed and failed). (K2)
  - LO-5.3.3 Summarize the purpose and content of the test summary report document according to the 'Standard for Software Test Documentation' (IEEE 829). (K2)

Monitoring the test activities

- Test monitoring:
  - Is a task of test management
  - Periodically checking the status of a test project
  - Report to compare the actual and planned
- Test Monitoring provides:
  - The feedback on how testing is going
  - Test results
  - The status of testing
  - Data for future test estimation

Test progress monitoring

- Common test metrics include:
  - Percentage of work done in test case preparation.
  - Percentage of work done in test environment preparation.
  - Test case execution
  - Defect information
  - Test coverage.
  - Subjective confidence of testers in the product.
  - Dates of test milestones.
  - Testing costs.



Test Summary Report

- Summary Report template IEEE 829
  - Test summary report identifier
  - Summary
  - Variances
  - Comprehensive assessment
  - Summary results
  - Evaluation
  - Summary of activities
  - Approval

Test Summary Report

- Test summary report identifer: A unique label so you can refer to that document.
- Summary. Summarize what was tested and what happened. Point to all relevant documents.
- Variances: If any test items differed from their specifications, the testing process didn't go as planned, why things were different.
- Comprehensiveness assessment. How thorough was testing, in the light of how thorough the test plan said it should be? What wasn't tested well enough? Why not?

Test Summary Report

- Summary of results. Which problems have been dealt with? What problems remain?
- Evaluation. How good are the test items? What's the risk that they might fail?
- Summary of activities. In outline, what were the main things that happened? What did they cost (people, resource use, time, money)?
- Approvals. Who has to approve this report? Get their signatures.

Test control (K2)

- Test control describes any guiding or corrective actions.
- Examples of test control actions:
  - Making decisions based on information from test monitoring.
  - Re-prioritize tests.
  - Change the test schedule due to availability of a test environment.
  - Set an entry criterion.
- Changes to test plan must be communicated clearly

5.4 Configuration management (K2)

- Objectives
  - LO-5.4.1 Summarize how configuration management supports testing. (K2)

Configuration management

- Purpose:
  - All items of software are identified, version controlled, tracked for changes, related to each other and related to development items
  - All identified document and software items are referenced unambiguously in test document

Configuration management

- Configuration control (version control)
  - An element of configuration management, consisting of the evaluation, co-ordination, approval or dis-approval and implementation of changes to configuration items after formal establishment of their configuration identification

5.5 Risk and testing (K2)

- Objectives
  - LO-5.5.1 Describe a risk as a possible problem that would threaten the achievement of one or
  - more stakeholders' project objectives. (K2)
  - LO-5.5.2 Remember that risks are determined by likelihood (of happening) and impact (harm
  - resulting if it does happen). (K1)
  - LO-5.5.3 Distinguish between the project and product risks. (K2)
  - LO-5.5.4 Recognize typical product and project risks. (K1)
  - LO-5.5.5 Describe, using examples, how risk analysis and risk management may be used for test planning. (K2)

Project risks (K2)

- Project risks are the risks that surround the project's capability to deliver its objectives, such as:
  - Logistics or product quality problem that block the tests
  - Excessive change to the product that invalidates test results
  - Insufficient or unrealistic test environments that yield misleading results

Project risks (K2)

- Other Project risks
  - Organizational factors:
    - Skill and staff shortages;
    - Personal and training issues;
    - Political issues
    - Improper attitude toward or expectations of testing
  - Technical issues:
    - Problems in defining the right requirements;
    - The quality of the design, code and tests.
  - Supplier issues

Product risks (K2)

- Product risks: Potential failure areas in the software or system such as:
  - Failure-prone software delivered.
  - The potential that the software/hardware could cause harm.
  - Poor software characteristics (e.g. functionality, reliability, usability and performance).
  - Software that does not perform its intended functions.

Steps of the risk management model

- The five steps of the risk management model are given below:
  - Identifying risks
  - Analyzing risks
  - Planning for risks
  - Tracking risks
  - Controlling risks

5.6 Incident Management

- Test Log:
  - Test log analysis needs to be done after test run
  - Document incidents if problem found caused by the test object, an incident report is initiated.
  - Cause-analysis (debugging) is developers' task
- Incident Reporting:
  - Central database should be established for each project
  - Personnel involved in development, as well as customers and users can report incidents
  - Reports can refer to problems in the tested programs, errors in specifications, user manuals, or other documents.
  - Document all information relevant to reproduction and correction

Incident Report Format

Table 6-1. Incident report template

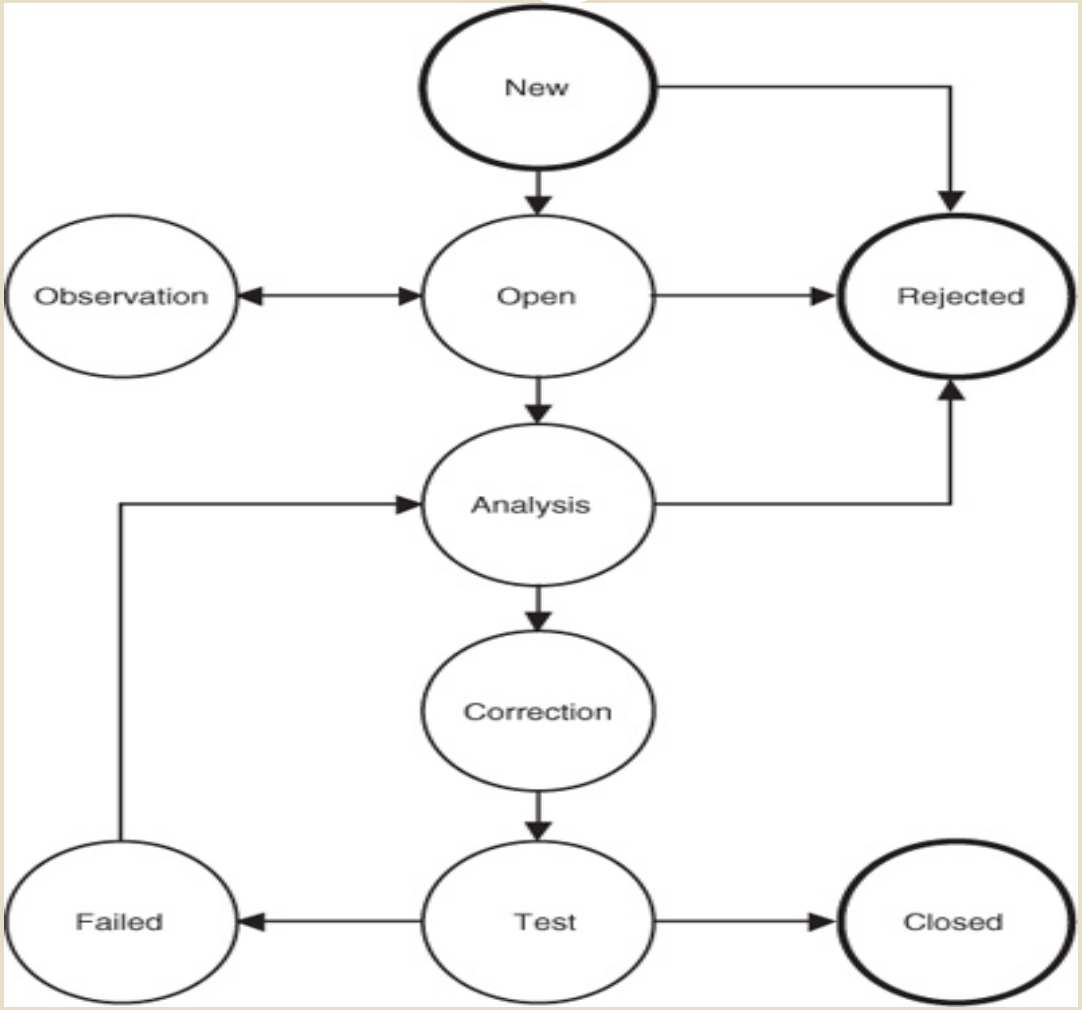
	Attribute	**Meaning**
Identification	Id/Number	Unique identifier/number for each report
	Test object	Identifier or name of the test object
	Version	Identification of the exact version of the test object
	Platform	Identification of the HW/SW platform or the test environment where the problem occurs
	Reporting person	Identification of the reporting tester (possibly with test level)
	Responsible developer	Name of the developer or the team responsible for the test object
	Reporting date	Date and possibly time when the problem was observed

Incident Report Format

Table 6-1. Incident report template

	Attribute	Meaning
Classification	Status	The current state (and complete history) of processing for the report
	Severity	Classification of the severity of the problem
	Priority	Classification of the priority of correction
	Requirement	Pointer to the (customer-) requirements which are not fulfilled due to the problem
	Problem source	The project phase, where the defect was introduced (analysis, design, programming); useful for planning process improvement measures
Problem description	Test case	Description of the test case (name, number) or the steps necessary to reproduce the problem
	Problem description	Description of the problem or failure occurred; expected vs. actual observed results or behavior
	Comments	List of comments on the report from developers and other staff involved
	Defect correction	Description of the changes made to correct the defect
	References	Reference to other related reports

Incident Status



IEEE 829 Standard: Test Incident Report Template

- Test incident identifier
- Summary
- Incident description
  - Input
  - Expected results
  - Actual results
  - Anomalies
  - Date and time
  - Procedure step
  - Environment
  - Attempts to repeat
  - Testers and observed
- Impact

References

- ISTQB Foundation Syllabus
- Foundation of Software Testing: ISTQB Certification
  - Dorothy Graham, Erik Van Veenedaal, Isabel Ivan, Rex Black
- Software Testing Practice: Test Management (Advance ISTQB)
  - Andreas Spillner, Thomas Rossner, Mario Winter, Tilo Linz
- IEEE Software Test Documentation, a summary
  - http://www.cs.otago.ac.nz/cosc345/lecs/lec22/testplan.htm

Q & A