ISTQB®
International Software
Testing Qualifications Board

ISTQB Foundation Level

Chapter5: Test Management

Test Management

- Test Organization and Independence
- Test Planning and Estimation
- Test Progress Monitoring and Control
- Configuration Management
- Risk and Testing
- Defect Management

Test Independence

- Degree of independence often makes the tester more **effective at finding defects** due to differences between the author's and the tester's.
- Developers can efficiently find many defects in their own code.

Testing's Degree of Independence (lowest to highest)

1.No independent testers; developers testing their own code

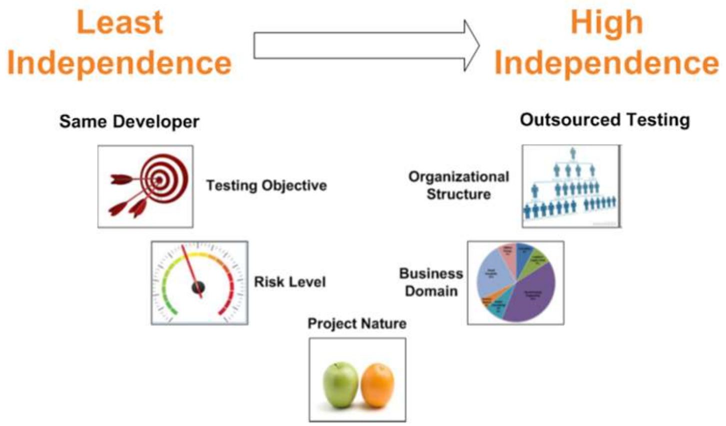
2.Independent developers or testers within the development teams or the project team

3.Independent test team or group within the organization

4.Independent testers from the business organization or user community, or with specializations in specific test types.

5.Independent testers external to the organization, either working on-site (insourcing) or off-site (outsourcing)

Independence Range



Least Independence

High Independence

Same Developer

Testing Objective

Risk Level

Project Nature

Business Domain

Organizational Structure

Outsourced Testing

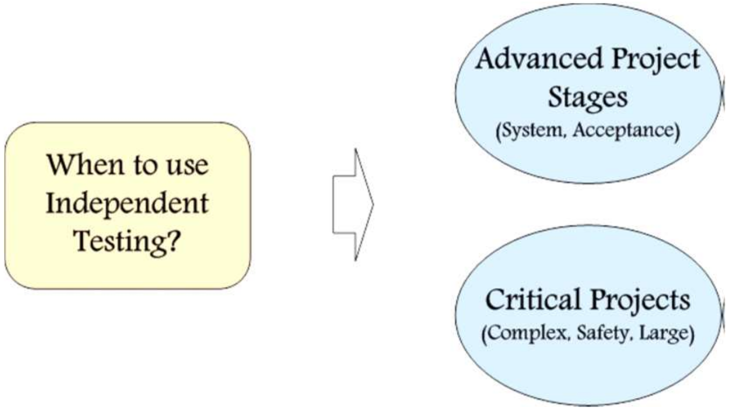
Benefits of Independence

- Independent testers are likely to **find different kinds of failures .**
- An independent tester can verify, challenge, or disprove assumptions made by stakeholders.

Drawbacks of Independence

- Isolation from the development team.
- Developers may lose a sense of responsibility for quality.
- Independent testers may be seen as a bottleneck or blamed for delays in release
- Independent testers may lack some important information

When to use Independent Testing?



When to use Independent Testing?

Advanced Project Stages (System, Acceptance)

Critical Projects (Complex, Safety, Large)

Test Leader and Tester

Test Leader & Testers are the Main Two roles in Testing Team

Test Leader’s Tasks

- Test policy-Test Strategy-Test Plan
- Test monitoring & Control (Test progress report-test summary report)
- Initiate the analysis, design, implementation, and execution of tests
- Configuration Management
- Metrics
- Tools selection
- Test Environment Implementation Decision
- Develop the skills and careers of testers

The Tester

A skilled professional who is involved in the testing of a component or system.

Tester’s Tasks

- Review and contribute to test plans
- Assess requirements for testability
- Test conditions, test cases, test procedures, test data, & test execution schedule
- Test Environment setup
- Test Execution
- Test automation
- Non-Functional Testing
- Review tests developed by others

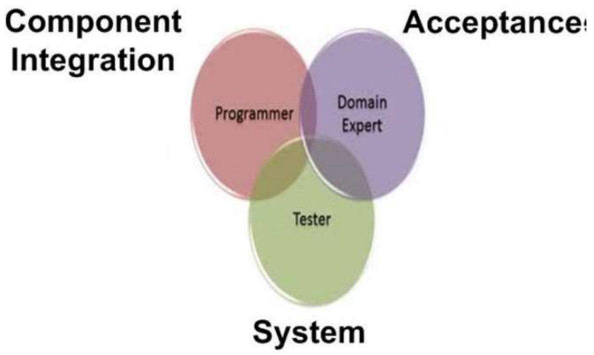
Specialized Testers

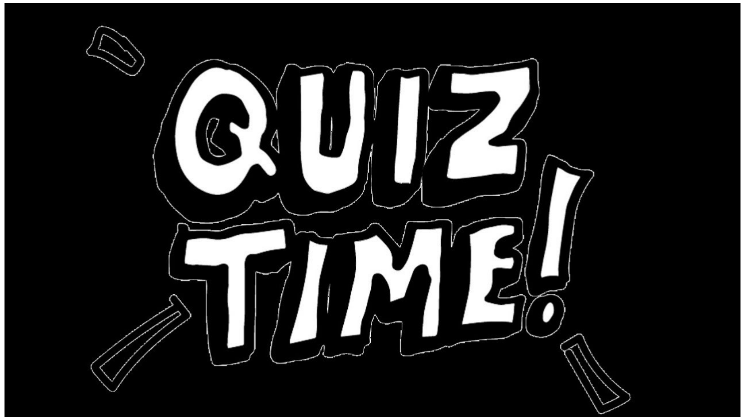
Performance

Security

Automation

Who take Testers Role?





Which of the following BEST describes how tasks are divided between the test manager and the tester?

A. The test manager plans testing activities and chooses the standards to be followed, while the tester chooses the tools and controls to be used

B. The test manager plans, organizes, and controls the testing activities, while the tester specifies and executes tests

C. The test manager plans, monitors, and controls the testing activities, while the tester designs tests and decides about automation frameworks

D. The test manager plans and organizes the testing and specifies the test cases, while the tester prioritizes and executes the tests

Who is normally responsible for the creation and update of a test plan for a project?

A. The project manager

B. The test manager

C. The tester

D. The product owner

Which of the following is a benefit of test independence?

A. Testers have different biases than developers

B. Testers are isolated from the development team

C. Testers lack information about the test object

D. Testers will accept responsibility for quality

Who normally writes the test plan for a project?

A. The project manager

B. The product owner

C. The test manager

D. The tester

What is the biggest problem with a developer testing his own code?

A. Developers are not good testers

B. Developers are not quality focused

C. Developers are not objective about their own code

D. Developers do not have time to test their own code

Test Planning and Estimation

Purpose of a Test Plan

A test plan **outlines test activities** for development and maintenance projects.

Content of a Test Plan

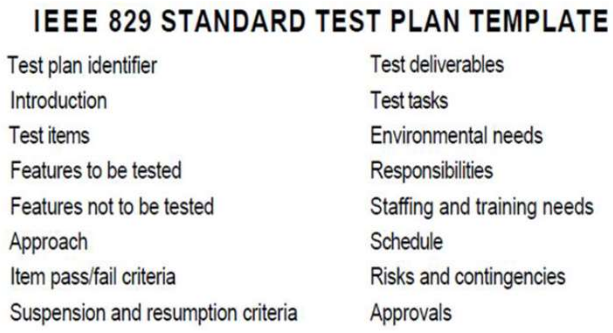
- Test planning is a continuous activity and is performed throughout the product's lifecycle
- Feedback from test activities should be used to recognize changing risks so that planning can be adjusted.
- Planning may be documented in a master test plan and in separate test plans for test levels or for separate test types.

How to plan our Testing?

Master Test Plan

Acceptance **Test Plan**
Integration **Test Plan**
System **Test Plan**
Unit **Test Plan**

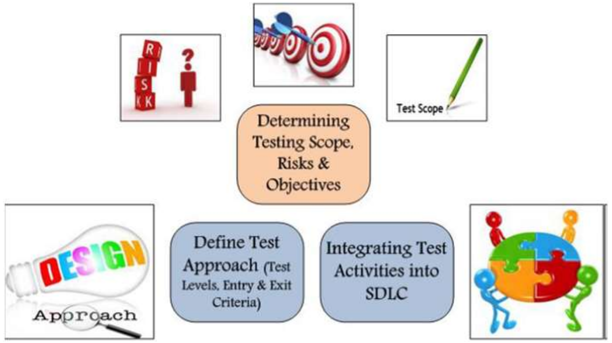
IEEE Template of Test Plan



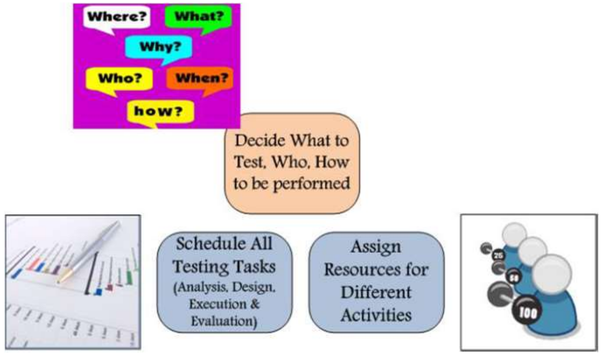
IEEE 829 STANDARD TEST PLAN TEMPLATE

- 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

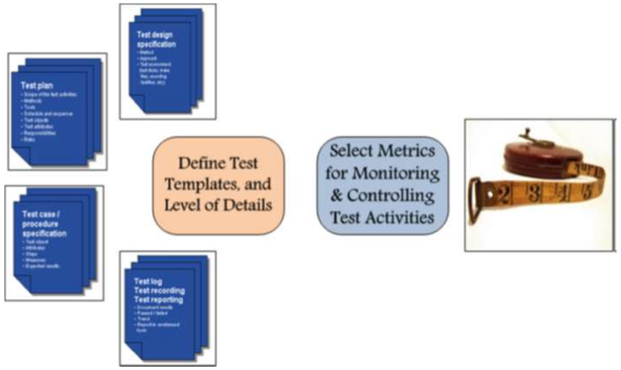
Test Planning Activities


RISK
Test Scope
Determining Testing Scope, Risks & Objectives
DESIGN
Approach
Define Test Approach (Test Levels, Entry & Exit Criteria)
Integrating Test Activities into SDLC

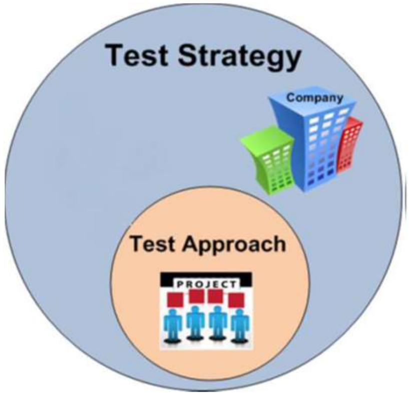
Test Planning Activities … Cont'd


Where?
What?
Why?
Who?
When?
how?
Decide What to Test, Who, How to be performed
Schedule All Testing Tasks (Analysis, Design, Execution & Evaluation)
Assign Resources for Different Activities
25
50
100

Test Planning Activities … Cont'd


Test plan
Test design specification
Test case / procedure specification
Test log
Test recording
Test reporting
Define Test Templates, and Level of Details
Select Metrics for Monitoring & Controlling Test Activities

Test Strategy & Test Approach



Test strategy types

- Analytical
- Model-based
- Methodical
- Process-compliant (or Standard Compliant)
- Directed (or Consultative)
- Regression-averse
- Reactive

1-Analytical:

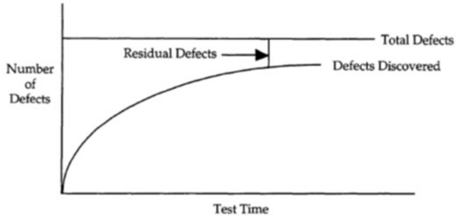
•This type of test strategy is based on an **analysis** of some factor (e.g., requirement or risk).

•**Risk-based testing** is an example of an analytical approach, where tests are designed and prioritized based on the level of risk.

2-Model-based

•In this type of test strategy, tests are designed based on some model of some required aspect of the product, such as a function, a business process, an internal structure, or a non-functional characteristic.

•Examples of models include business process models, state models, and reliability growth models.



3-Methodical

- This type of test strategy relies on making systematic use of some predefined set of tests or test conditions, such as likely types of failures or a list of important quality characteristics.

4-Process-compliant (or Standard Compliant)

- This type of test strategy involves analyzing, designing, and implementing tests based on **external rules and standards**, such as those specified by industry-specific standards
- Ex. ISO , IEEE ,

Standard

5-Directed (or Consultative)

- This type of test strategy is driven by the **advice, guidance, or instructions of stakeholders**, business domain experts, or technology experts, who may be outside the test team or outside the organization itself.
- Ex. Tester ask his friends on type of security testing and using tools.

6-Regression-averse

•This type of test strategy is motivated to avoid regression of existing capabilities.

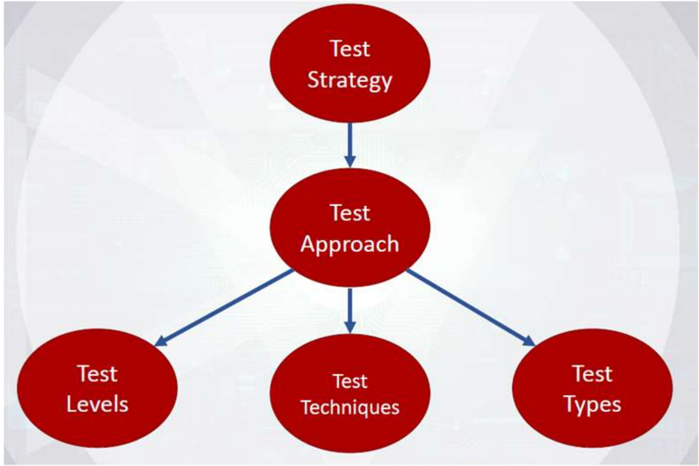
•This test strategy includes reuse of existing testware, extensive **automation of regression tests**, and standard test suites.

7-Reactive

- In this type of test strategy, testing is reactive to the component or system being tested, and the events occurring during test execution, **rather than being pre-planned** (as the preceding strategies are).

- Tests are designed and implemented, and may immediately be executed in response to knowledge gained from previous test results.

- **Exploratory testing** is a common technique employed in reactive strategies



Test Strategy

Test Approach

Test Levels

Test Techniques

Test Types

Entry and Exit Criteria

Goal: Specify When to Start and Stop Testing

Entry Criteria

- Entry criteria (definition of ready) define the preconditions for undertaking a given test activity.

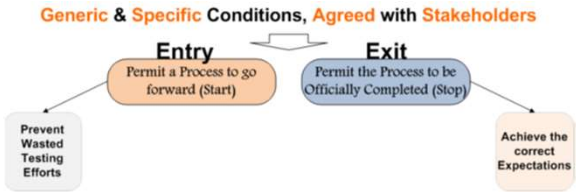
- Typical entry criteria include:
  - Availability of testable requirements, user stories, and/or models
  - Availability of test items that have met the exit criteria for any previous test levels
  - Availability of test environment
  - Availability of necessary test tools
  - Availability of test data and other necessary resources

Exit Criteria

- Exit criteria (definition of done) define what conditions must be achieved in order to declare a test level or a set of tests completed.

- Typical exit criteria include:
  - Planned tests have been executed
  - A defined level of coverage has been achieved
  - The number of unresolved defects is within an agreed limit
  - The number of estimated remaining defects is low
  - The evaluated levels of quality characteristics are sufficient

Entry and Exit Criteria


Generic & Specific Conditions, Agreed with Stakeholders

Entry

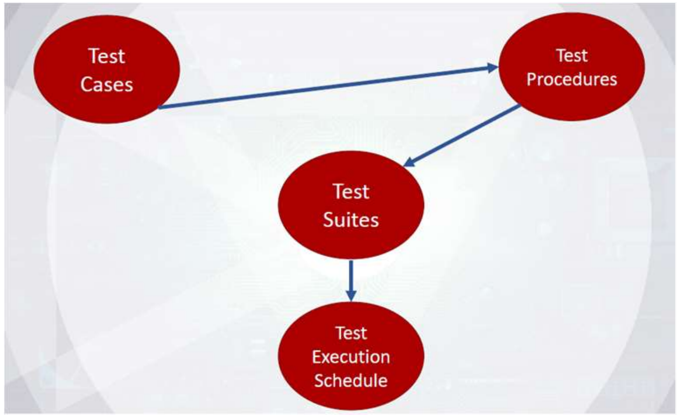
Permit a Process to go forward (Start)

Exit

Permit the Process to be Officially Completed (Stop)

Prevent Wasted Testing Efforts

Achieve the correct Expectations


Test Cases

Test Procedures

Test Suites

Test Execution Schedule

Test Execution Schedule

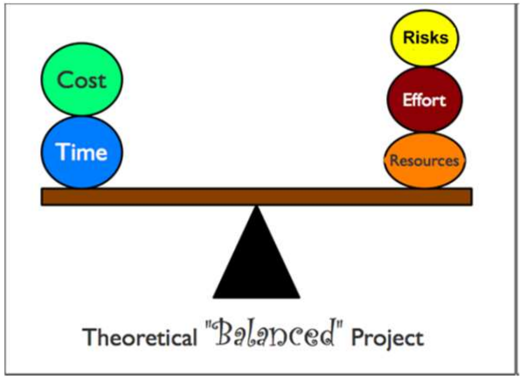
- Test cases would be ordered to run based on their priority levels

- If a test case with a higher priority is dependent on a test case with a lower priority, the lower priority test case must be executed first.

Test Effort Estimation

The Approximate calculation of the Results of different Testing Aspects … **Time, Resources and Costs**

Need for Test Estimation



Factors Influencing the Test Effort

- Product Characteristics
- Development Process Characteristics
- People Characteristics
- Test Results

Product Characteristics

- Risks
- Quality of the test basis (requirement , user store and use case) it's clear?
- Size-Complexity-Requirements
- Documentation Required ? (If required to write test case low level of the system the test effort will be an increase).

Development Process Characteristics

- The stability and maturity of the organization
- The development model in use
- The test approach <<
- The tools used
- The test process
- Time pressure

People Characteristics

- The skills and experience of the people involved, especially with similar projects and products.

- Team cohesion and leadership.

Test Results

- The number and severity of defects found.

- The amount of rework required.

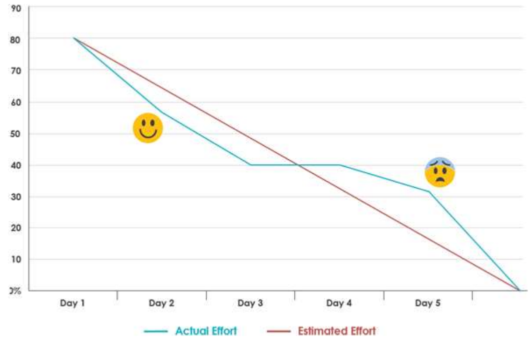
Login >> 2H >>15 Bug report !! >> re-testing

Test Estimation Techniques

1.Metrics-based :estimating the test effort based on metrics of former **similar projects**, or based on typical values

2.Expert-based :estimating the test effort based on the **experience of the owners of the testing tasks** or by experts

Burn-down Chart

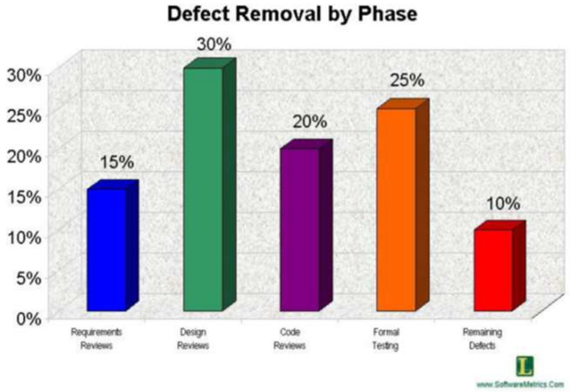


Agile/Metrics-based

Planning Poker

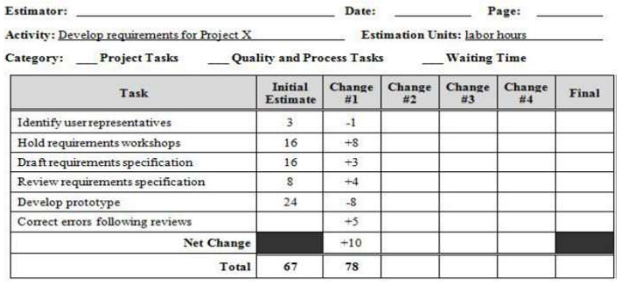
Agile/Expert-based

Defect Removal Models



Sequential/Metrics-based

Wideband Delphi



Estimator: ______________________ **Date:** __________ **Page:** _________

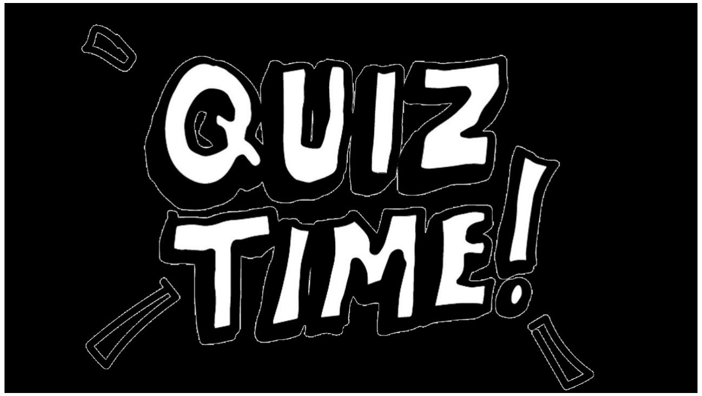
Activity: Develop requirements for Project X **Estimation Units:** labor hours

Category: ___ **Project Tasks** ___ **Quality and Process Tasks** ___ **Waiting Time**

Task	Initial Estimate	Change #1	Change #2	Change #3	Change #4	Final
Identify user representatives	3	-1				
Hold requirements workshops	16	+8				
Draft requirements specification	16	+3				
Review requirements specification	8	+4				
Develop prototype	24	-8				
Correct errors following reviews		+5				
Net Change	■	+10				■
Total	**67**	**78**				

Figure 2: Sample Wideband Delphi estimation form.

Sequential/Expert-based



Which TWO of the following can affect and be part of test planning?

a)Budget limitations

b)Test objectives

c)Test log

d)Failure rate

e)Use cases

You are working in a team of testers who are all writing test cases. You have noticed that there is a significant inconsistency with the length and amount of detail in the different test cases. Where should the test case guidelines have been documented?

A. The test plan

B. The test approach

C. The test case template

D. The project plan

Which of the following is an example of a good exit criterion from system testing?

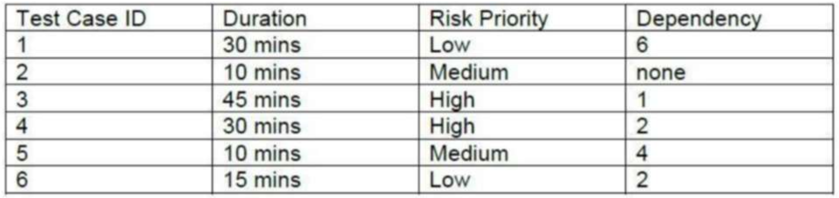
A. All tests should be completed

B. The project budget should be spent

C. All defects should be fixed

D. All severity 1 defects must be resolved

You have been given the following set of test cases to run. You have been instructed to run them in order by risk and to accomplish the testing as quickly as possible to provide feedback to the developers as soon as possible. Given this information, what is the best order in which to run these tests?



Test Case ID	Duration	Risk Priority	Dependency
1	30 mins	Low	6
2	10 mins	Medium	none
3	45 mins	High	1
4	30 mins	High	2
5	10 mins	Medium	4
6	15 mins	Low	2

A.2, 4, 5, 6, 1, 3

B.4, 3, 2, 5, 6, 1

C.2, 5, 6, 4, 1, 3

D.6, 1, 3, 2, 4, 5

There are several test strategies. Which strategy (1-4) is characterized by which description (A-D) below?

1.Analytical.

2.Methodical.

3.Model-based.

4.Consultative.

a)Tests are based on a state diagram of a required aspect of the product

b)Tests are designed and prioritized based on the level of risk.

c)Systematic use of some predefined set of test conditions.

d)Tests are chosen based on the views of business domain experts

a.1D, 2B, 3A, 4C

b.1A, 2C, 3D, 4B

c.1D, 2C, 3B, 4A

d.1B, 2C, 3A, 4D

Which one of the following is the characteristic of a metrics-based approach for test estimation?

A. Budget which was used by a previous similar test project.

B. Overall experience collected in interviews with test managers

C. Overall estimate agreed with the developers

D. Average of calculations collected from business experts.

If your test strategy is based off the list of the ISO 25010 quality characteristics, what type of strategy is it?

A. Regulatory

B. Analytical

C. Methodical

D. Reactive

You are getting ready to test another upgrade of an ERP system. The previous upgrade was tested by your team and has been in production for several years. For this situation, which of the following is the most appropriate test effort estimation technique?

A. Effort-based

B. Expert-based

C. Metric-based

D. Schedule-based

Consider the following test cases that are used to test an accounting system:

Test ID	Name	Dependency	Priority
1	Purchase Item	none	2
2	Receive Invoice	Test 1	3
3	Receive Goods	Test 1	2
4	Send Payment	Test 2	3
5	Report Payments	Test 4	1

➢Given this information, what is the proper order in which to execute these test cases?

A.5, 1, 3, 2, 4

B.1, 2, 4, 3, 5

C.1, 3, 2, 4, 5

D.3, 4, 5, 1, 2

Test Monitoring

A Test Management Task to check Periodically the Status of Test Activities

Actual Status is Compared to Planned

Test Control

- A Test Management Task to Develop & Apply Corrective Actions, when tasks Deviated from Planned
- To Return the Test Project On Track

Test Monitoring & Control

- The purpose of test monitoring is to gather information and provide feedback and visibility about test activities.
- Test control describes any guiding or corrective actions taken as a result of information and metrics gathered and reported.

Examples of Test Control Activities

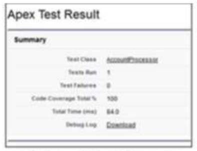
- Re-prioritizing tests when an identified risk occurs

•Changing the test schedule (4w)due to availability or unavailability of a test environment (1v) or other resources (1 tester)

•Re-evaluating whether a test item meets an entry or exit criterion due to rework

Test Monitoring Goals

Feedback about Test Progress



Apex Test Result

Summary

Test Class	AccountProcessor
Tests Run	1
Test Failures	0
Code Coverage Total %	100
Total Time (ms)	84.0
Debug Log	Download

Status of Test Coverage Vs. Exit Criteria

Visibility to Project Team

Assess Progress Vs. Planned & Budget

Estimate Future Test Results

Common Test Monitoring Metrics

•Percentage of planned work done in (test case preparation/implementation-Test environment preparation)
100/5 >>

•Test case execution (e.g., number of test cases run/not run, test cases passed/failed)

•Defect information >>H/L

•Test coverage of requirements, user stories, acceptance criteria, risks, or code
20 ?
10/20 >>

•Task completion, resource allocation and usage, and effort

•Cost of testing
7000/10000

Test Reporting

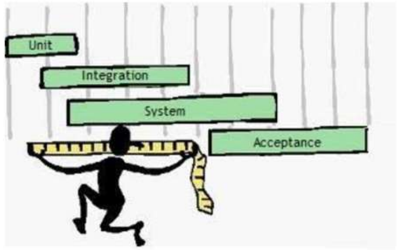
- The purpose of test reporting is to summarize and communicate test activity information, both during and at the end of a test activity.
- The test report prepared during a test activity may be referred to as a **test progress report**, while a test report prepared at the end of a test activity may be referred to as a **test summary report.**

Test progress reports include:

- The status of the test activities and progress against the test plan
- Factors impeding progress
- Testing planned for the next reporting period
- The quality of the test object (Non functional)

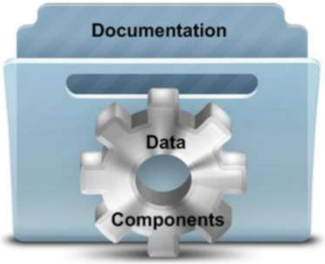
When to collect Metrics?

- Metrics should be collected During & At the END of each Test Level



Configuration Management

Establish & Maintain the Integrity of Software (Components, Data and Documentation) through the project

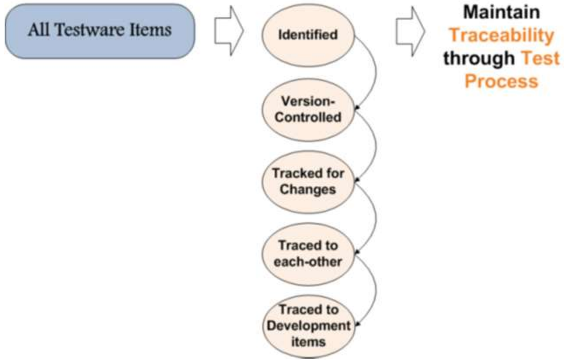


For All Project artifacts … Not only Testing

Configuration Management

- During test planning, configuration management procedures and infrastructure (tools) should be identified and implemented.

CM for Testing



If the developers are releasing code for testing that is not version controlled, what process is missing?

A. Configuration Management

B. Debugging

C. Test and defect management

D. Risk analysis

Risks and Testing

- Chance of an Event/Threat/Situation occurring & resulting in undesirable Results.
- The level of risk is determined by the **likelihood** of the event and the **impact** (the harm) from that event.
- Risk = Probability of trouble * Cost of trouble

Types of Risks

Risks at Product Level

Product risk involves the possibility that a work product may **fail to satisfy the needs of its users and/or stakeholders**, examples include:

1. Software not perform functions
2. A system architecture may not support some non-functional requirement(s)
3. A particular computation may be performed incorrectly.
4. A loop control structure may be coded incorrectly
5. Response-times may be inadequate for a high-performance transaction processing system
6. User experience (UX) feedback not meet product expectations

Risks at Project Level

Project risk involves situations that may have a **negative effect on a project's** ability to achieve its objectives, examples include :

1. Project issues

2. Organizational Issues
- Cost
- Shortage of Skill , training

3. Political Issues (issue between developer and tester).

4. Technical Issues
- test env is not ready
- Defining the right requirement
- Low quality of design or code

5. Supplier Issues
- Failure of Third party

Defect Report Objectives

- Provide developers with information about any event (defect) that occurred.
- Provide ideas for development and test process improvement
- Help test managers to tracking the quality of the work product

Defect Report Components

1. Identifier-Title-Summary-Date-Author-Test Item-Test Environment
2. The development lifecycle phase(s) in which the defect was observed
3. A description of the defect to enable reproduction and resolution, including logs, database dumps screenshots, or recordings (if found during test execution)
4. Expected and actual results
5. Scope or degree of impact (severity) of the defect on the interests of stakeholder(s)
6. Urgency/priority to fix
7. State of the defect report
8. Conclusions, and approvals
9. Global issues, such as other areas that may be affected by a change resulting from the defect
10. Change history
11. References including the test case that revealed the problem

Defect Management

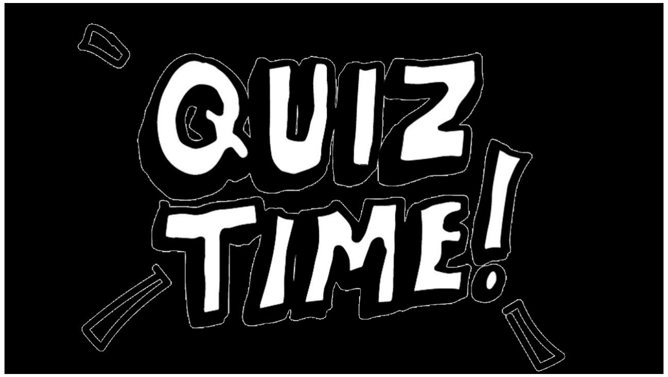
- During the defect management process, some of the reports may turn out to describe **false positives**, not actual failures due to defects.

- For example, a test may fail when a network connection is broken or times out.

Defect Management

- Some of these details may be automatically included and/or managed when using defect management tools, e.g., automatic assignment of an identifier, assignment and update of the defect report state during the workflow, etc.

- Defects found during **static testing**, particularly reviews, will normally be documented in a different way, e.g., in review meeting notes.



- You are testing a new version of software for a coffee machine. The machine can prepare different types of coffee based on four categories. i.e. coffee size, sugar, milk and syrup. The criteria are as follows:

● Coffee size (small, medium, large),

● Sugar (none, 1 unit, 2 units, 3 units, 4 units),

● Milk (yes or no),

● Coffee flavor syrup (no syrup, caramel, hazelnut, vanilla)

- Now you are writing a defect report with the following information:

Title: Low coffee temperature.

Short summary: When you select coffee with milk, the time for preparing coffee is too long and the temperature of the beverage is too low (less than 40 oC)

Expected result: The temperature of coffee should be standard (about 75 oC).

Degree of risk: Medium

Priority: Normal

What valuable information is MOST likely to be omitted in the above defect report?

a)The actual test result

b)Data identifying the tested coffee machine

c)Status of the defect

d)Ideas for improving the test case

- You have received the following description section in a defect report.

The report executed per the attached steps, but the data was incorrect. For example, the information in column 1 was wrong. See the attached screenshot. This report is critical to the users and they will be unable to do their jobs without this information. What is the biggest problem with this defect report?

A. The developer won’t know how important the problem is

B. The developer won’t know how to repeat the test

C. The developer won’t be able to see what the tester is saying is wrong

D. The developer doesn’t know what the tester expected to see

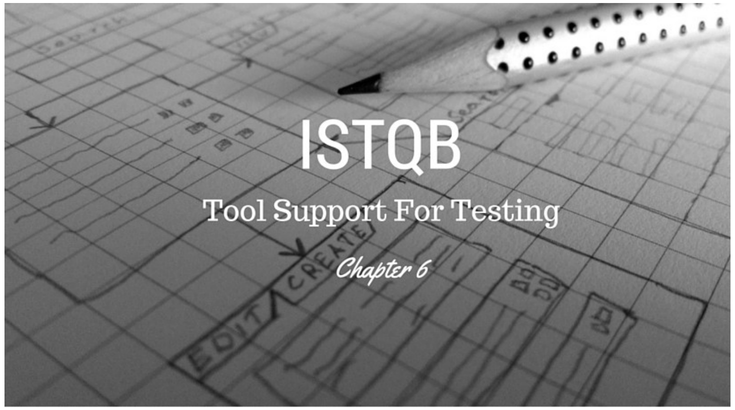
- You have been testing software that will be used to track credit card purchases. You have found a defect that causes the system to crash, but only if a person has made and voided 10 purchases in a row. What would be the proper priority and severity rating for this defect?

A. Priority high, severity high

B. Priority high, severity low

C. Priority low, severity low

D. Priority low, severity high



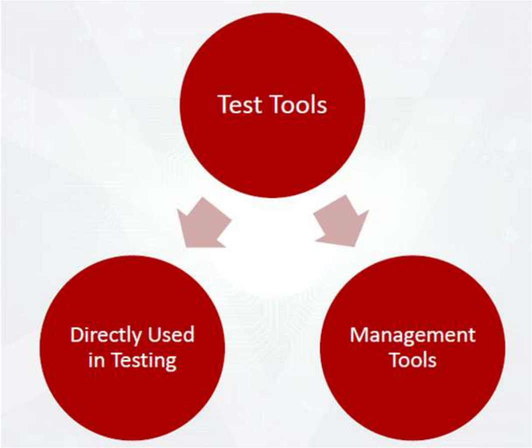
ISTQB

Tool Support For Testing

Chapter 6

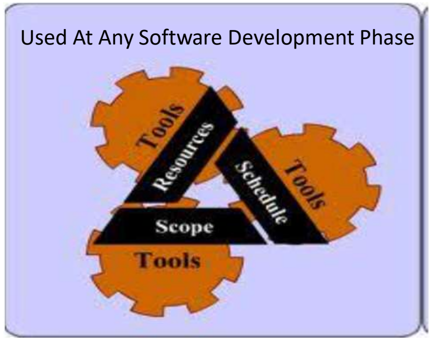
Tool Support for Testing

- Types of Test Tools
- Effective Use of Tools



Tool Support for Testing

Software Tools are used to Solve/Enhance the way of dealing with Project Constraints



Objectives of Test Tools

- Improve the efficiency of test activities by **automating repetitive tasks** or tasks that require significant resources when done manually
- Improve the efficiency of test activities by **supporting manual test activities** throughout the test process
- Improve the quality of test activities by **allowing for more consistent testing** and a higher level of defect reproducibility
- Automate activities that cannot be executed manually (E.g. Load testing).
- Increase reliability of testing

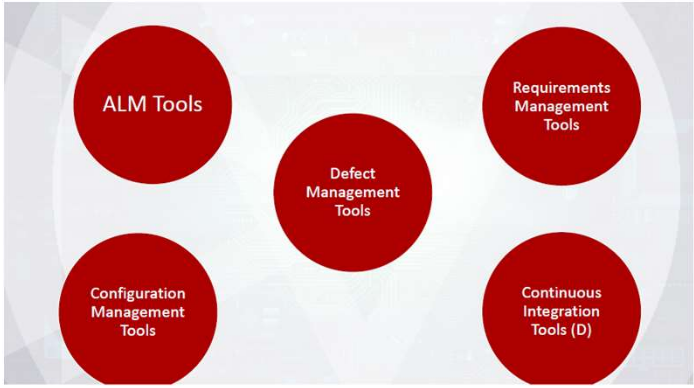
Intrusive Tools

- Intrusive Tools may affect the actual outcome of the test

- The consequence(result) of using intrusive tools is called the **probe effect**

E.g. using the simulator to test android application, this tool effect on app performance.

1) Tool Support for Management of Testing



ALM Tools

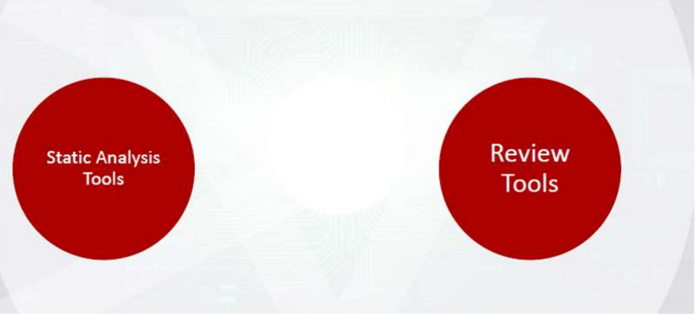
Requirements Management Tools

Defect Management Tools

Configuration Management Tools

Continuous Integration Tools (D)

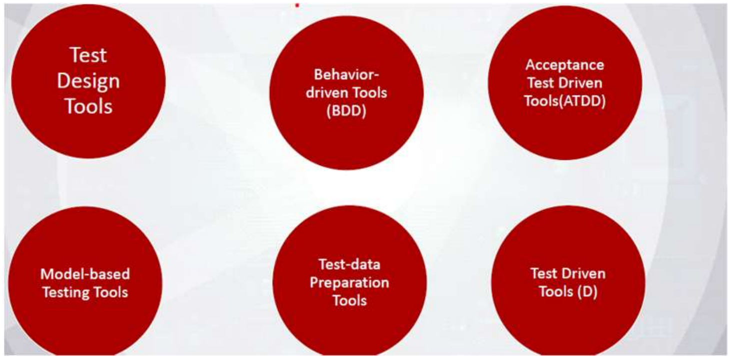
2) Tool Support for Static Testing



Static Analysis Tools

Review Tools

3)Tool support for Test Design & Implementation



Test Design Tools

Behavior-driven Tools (BDD)

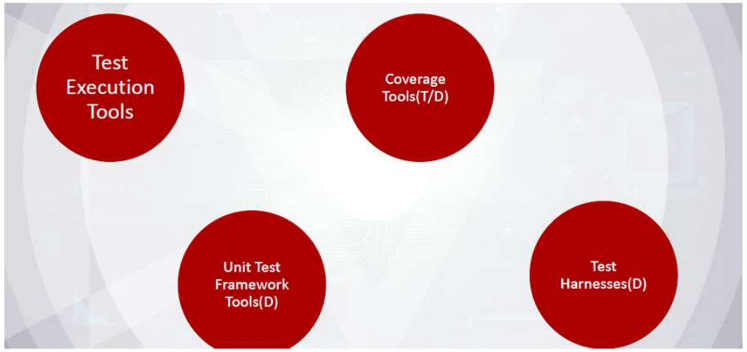
Acceptance Test Driven Tools(ATDD)

Model-based Testing Tools

Test-data Preparation Tools

Test Driven Tools (D)

4)Tool support for Test Execution & Logging



Test Execution Tools

Coverage Tools(T/D)

Unit Test Framework Tools(D)

Test Harnesses(D)

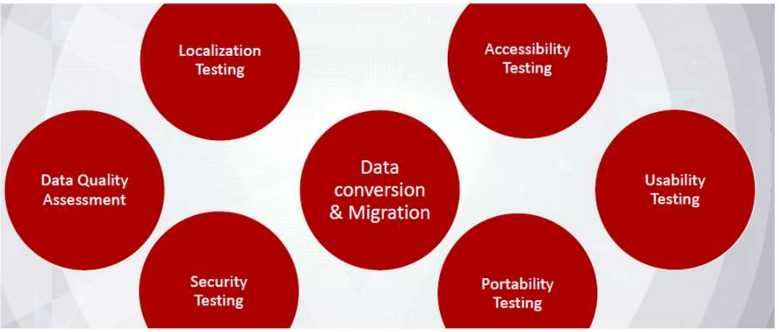
5)Tool support for Performance Measurement & Dynamic Analysis

Monitoring Tools

Performance Testing Tools

Dynamic Analysis Tools (D)

6)Tool Support for Specialized Testing Needs



Localization Testing

Accessibility Testing

Data Quality Assessment

Data conversion & Migration

Usability Testing

Security Testing

Portability Testing

Benefits of Test Execution Tools (Automation)

- Reduction in repetitive manual work, to saving time
- Greater consistency and repeatability
- More objective assessment (Not subjective).
- Easier access to information about testing

Risks of Test Execution Tools (Automation)

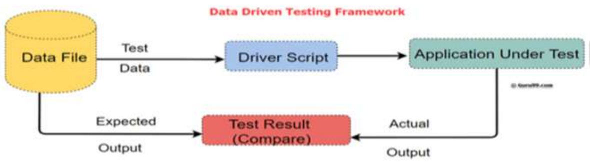
- Unrealistic expectations of the tool
- Underestimating time, cost, & effort for the initial introduction of a tool and also to achieve significant benefit from it
- The effort required to maintain the test assets generated by the tool may be under-estimated
- The tool may be relied on too much
- Relationships issues between critical tools may be neglected
- Vendor problems (Retire or sell the tool/ poor response/go out of business)
- An open source project may be suspended
- A new platform or technology may not be supported by the tool

Capture/Replay (Record/Playback) Tools

- Capturing tests by recording the actions of a manual tester seems attractive, but this approach does not scale to large numbers of test scripts.
- A captured script is a linear representation with specific data and actions as part of each script. This type of script may be unstable when unexpected events occur.
- Generated scripts still require ongoing maintenance as the system’s user interface evolves over time

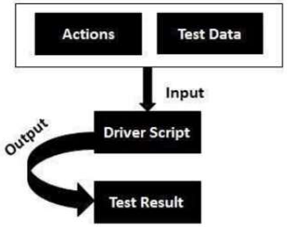
Data-driven Testing

- A data-driven testing approach separates out **the test inputs and expected results**, usually into a spreadsheet, and uses a more generic test script that can read the input data and execute the same test script with different data.
- Testers who are not familiar with the scripting language can then create new test data for these predefined scripts.



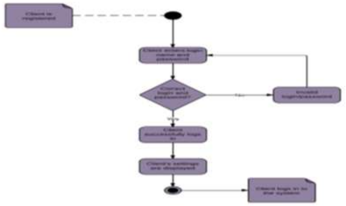
Keyword-driven Testing

- In a keyword-driven testing approach, a generic script processes keywords describing the actions to be taken (also called action words), which then calls keyword scripts to process the associated test data.
- Testers (even if they are not familiar with the scripting language) can then define tests using the keywords and associated data, which can be tailored to the application being tested.



Model-based Testing

- Model-Based testing (MBT) tools enable a functional specification to be captured in the form of a model, such as an activity diagram.
- This task is generally performed by a system designer.
- The MBT tool interprets the model in order to **create test case** specifications which can then be saved in a test management tool and/or executed by a test execution tool



Effective Use of Tools

Considerations for Tool Selection

- Assessment of the maturity of the organization, its strengths and weaknesses
- Identification suitable tool for an improved test process supported by tools
- Understanding of the technologies used by the test object(s), in order to select a tool that is compatible with that technology
- The build and continuous integration tools already in use within the organization, in order to ensure tool compatibility and integration
- Evaluation of the tool against clear requirements and objective criteria

Considerations for Tool Selection

- Consideration of whether or not the tool is available for a free trial period (and for how long)
- Evaluation of the vendor
- Identification of coaching, mentoring, & training needs
- Consideration of pros and cons of various licensing models (e.g., commercial or open source)
- Estimation of a cost-benefit ratio
- a proof-of-concept evaluation should be done

Pilot Project

introducing the selected tool into an organization generally starts with a pilot project, which has the following objectives:

- Gaining knowledge about the tool
- Evaluating how the tool fits with existing processes and practices
- Deciding on standard ways of using, managing, storing, and maintaining the tool and the test assets
- Assessing whether the benefits will be achieved at reasonable cost
- Understanding the metrics that you wish the tool to collect and report, and configuring the tool to ensure these metrics can be captured and reported

Success Factors for Tools

- Rolling out the tool to the rest of the organization incrementally
- Adapting and improving processes to fit with the use of the tool
- Providing training, coaching, and mentoring for tool users
- Defining guidelines for the use of the tool
- Implementing a way to gather usage information from the actual use of the tool
- Monitoring tool use and benefits
- Providing support to the users of a given tool
- Gathering lessons learned from all users