UES
Unlimited Educational Services®
www.eltturkey.com

COTS: commercial of the shelf product

— broad level — — narrow level —

* Test scenario is what to be tested. Test case is how it is tested.
* Scenario is the journey of user. Is linear statements. Reduces complexity & repetition
* Test scenario is a detailed documentation of test cases that cover end to end functionality of software. It is a high level classification of testable requirements and these are grouped on the basis of the functionality of a module, obtained from use cases. Looking from the user perspective is crucial
* Examine (Read) BLS, SRS, FRS and create traceability matrix for requirements (scenarios &)

—

Difference between Validation & Verification

Validation is the process of checking whether the specification captures the customer needs, Verification is the process of checking that the software meets the specification.

—

* Debugging is the development activity that finds, analysis and fixes defects.
* Quality management covers: QA & QC
* Error (mistake) → Defect (Bug, ~~failure~~ fault) → Failure.

—

False positives: Not all unexpected test results are failures. false positives may occur due to errors in the way tests were executed or due to ~~error~~ defect in the test data environment. They are reported as defects, but they are not actually.

False negatives: Tests do not detect defects that they should have detected.

False positives: Tests detect defects that they should not have detected

Richmond
ZB Zaner-Bloser

Root cause of failures (Klisa Slaytton) of projects

Inaccurate requirement specification %56
Under project goals, business process complexity, wrong schedule

Inefficient implementation %22
high complexity, high risks, reinventing the wheel, inadequate team

Ineffective testing %22
Shedule pressure, insufficient automation, application dependence, insufficient real data

---

Seven principles of testing

1-Testing shows the presence of defects, not their absence
2-Exhausting testing is not possible
3-Early testing saves time & money shift left
4-Defects cluster together 80/20 pareto rule
5-Beware of pesticide paradox
6-Testing is context dependent
7-Absence of errors is a fallacy

Test activities

they may be sequential or iterative
Test planning: defining objectives, approach, constraints, techniques, schedule
Test monitoring & control: checking exit criteria, DoD, assessing quality, checkup results
Test analysis: "what to test" identify testable features and test conditions.
Test design: "how to test" high level test cases. design test cases, test environment
analysis test basis
Test implementation: having everything to start to run the tests. capturing bidirectional traceability
creating test suits, building test environment preparing test data
Test execution: reporting defects, bidirectional traceability
Test completion: lessons learned, summary report

Test planning: test plans (exit criteria, test basis)
Test monitoring & control: progress reports, summary reports
Test analysis: prioritized and defined test conditions, test charters.
Test design: high level test cases, design of test data / tool / environment
Test implementation: test procedures, test suites, test schedule
refinement of analysis
low level test cases, test scripts, test data

} produced

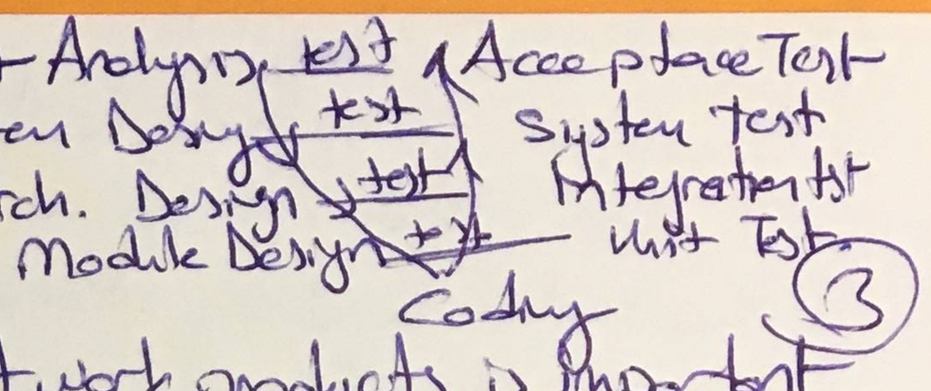
---

test basis: source of info that is needed to write test cases
test suite: collection of test cases to test a SW
to check a set of behavior
exploratory testing: test cases are created on the fly
thinking

Test execution: defect report documentation
Test completion: summary report
close report requests,

UES
Unlimited Educational Services®
www.eltturkey.com

V model



(3)

* Maintaining traceability between test basis & test work products is important to check test coverage, improve understandings, evaluate quality and progress.
requirement check

* Confirmation bias makes it difficult to accept information that disagrees with currently held beliefs. His a human trait to blame the bearer of bad news

* Tester mindset: curiosity, professional pessimism, critical eye, attention to detail positive & good communication-relationship.

* Rational unified process: long (2-3 months) increments — Self organized teams.
* Kanban: maybe both fixed-unfixed length iterations
* Spiral (prototyping) : experimental increments.

— Test Levels —

Component Testing	Integration Testing	System Testing	Acceptance Testing
aka unit/module testing	interactions between	behaviour of system/product	behavior & capabilities of full system
Test basis: detailed design	components & systems, and	end-to-end tasks	validates that system work as expected
Component specs	interfaces between integrated	provides release decision checks.	verifys functional & nonfunctional behavior as specified.
may cover both functional	components.	leg/regulatory req.	checks readiness for deployment
and non-functional characteristics	design	use cases	Test basis: contracts
and structural properties	Test basis: software & system	Test basis: risk analysis	user docs.
TDD: building automated tests	sequence diagrams, use cases	user stories	standards regulations
building & integrating code	workflows	Requirement specs.	
correcting issues and refactoring	communication failures,	state diagrams	
code.	inconsistencies, incorrect data	user manuals	
coders developers responsible	unhandled messaging	static testing	
	testers responsible	defects in	
	integration should be	reduces system	
	incremental.		

Acceptance Test Types:
UAT User Acceptance Testing
OAT Operational Acceptance Testing
Contractual or Regulatory Acceptance Testing

Alpha Testing: not by development team
By potential/existing customers/operators
independent test team at developing organization's site

Beta Testing: potential/existing customers operators at their own locations

Test Types

*Functional Testing
Can be performed at all test levels. Black-box tests can be used.

*Non-Functional Testing
Testing of how well the system behaves. Should be performed at all test levels.
Black box tests can be used.

*White Box Testing
Based on system's internal structure or implementation. Includes code, architecture, work flow, data flow
Code coverage may be tested. It may be called structural testing. At any test level.

*Change Related Testing (Regression Testing): At all test levels.
Confirmation Testing: Confirm original defect is successfully fixed.
Regression Testing: Running tests to detect unintended side effects.

Maintenance Testing
Modification: updates, patches, enhancements
Migration:
Retirement: for long retention periods.

Impact Analysis in maintenance testing is difficult;
- Specifications are out of date or missing
- Test cases are not documented or out of date
- Bidirectional traceability test-test basis has not maintained
- Tool support is weak or non-existent
- People do not have domain or system knowledge
- Insufficient attention to maintainability during development

Test Design Techniques
- Dynamic Test Design: Focus on externally visible behavior
- Static Test Design: Testing specs, guides, codes, user stories
  provides early detection of defects. Less cost.
  improves communication.
- Requirement defect
- Design defect
- Coding defect
- Deviation from standard
- Incorrect Interface spec
- Security vulnerabilities
- Gaps/inaccuracy in test basis traceability or coverage
- Maintainability defect

Non Functional Tests

*Reliability
- fault tolerance
- recoverability

*Usability
- understandability
- learnability
- operability
- attractiveness

*Efficiency
- time behavior
- resource utilization

*Portability
- adaptability
- installability
- co-existence
- replaceability

*Maintainability
- analyzability
- changeability
- stability
- testability

UES
Unlimited Educational Services®
www.eltturkey.com

5

A dry run is usually a test performed to see how a failure mitigation system will behave in the event of a failure (jet engine is started on porting. e.g.)

Product review process: planning/ individual review/ issue communication & analysis/ fixing/ reporting

Roles & responsibilities in formal review: Author/ management/ facilitator (moderator)/ Review leader/ Reviewers/ Scribe (Recorder)

Review Types

Informal Review:	detecting potential defects, results may be documented	generating solutions/ ideas	common in Agile	
Walkthrough:	find defect, exchange ideas, training participants	evaluate performance, conformance	scribe is mandatory	in the format scenarios
Technical Review:	detect potential defect	gain consensus, evaluating quality, building confidence	preparation is optional / preparation is required	experts
Inspection:	motivate authors, detect potential defect	evaluate quality, prevent future defect	scribe is mandatory, preparation is required	authors not be review leader

Review Technique:

Ad hoc / Checklist based / Scenarios & Dry-runs / Role Based / Perspective Based *Best for reviewing

Test Techniques

Black-Box Testing: (Behavior-Based Techniques) concentrate on input/output without intend structure

White Box Testing: (Structure-Based Techniques) concentrate on structure & process within object

Experience Based Testing: Combination of Black Box & White Box techniques

Black Box Test Techniques

aims to reduce time required for software testing

- Equivalence Partitioning: Dividing the input data into partitions of equivalent data
- Boundary Value Analysis
- Decision Table Testing
- State Transition Testing

White Box Test Techniques

Statement Coverage & Decision Coverage

Achieving %100 decision coverage guarantees %100 statement coverage but not vice versa.

Richmond HELBLING BLACK CAT ZB Zaner-Bloser

Experience-Based Test Techniques

- Error guessing
- Exploratory testing: when there is not enough specification
- Checklist-based testing

Test Organization

Degrees of Independence
- No independent testers
- Independent developers or testers
- Independent test team
- Independent testers from organization
- Independent testers external to org.

drawbacks of independence
- isolation from development team
- lack of collaboration
- developers lose sense of quality
- may lack of important info
- testers may be blamed of delays

Test planning

Defining overall approach, scheduling, budgeting, scope, objectives, risk

Test Strategies:

*Analytical: risk-based or requirement-based
*Model based: business process model, state model, reliability-growth model.
*Methodical: types of failures, look-and feel standards, list of quality features.
*Process or Standart Compliant: based on external rules or standarts imposed or/by organizations
*Directed (consultative): driven by expert & stakeholders outside organization or team
*Regression averse: avoid regression of existing capabilities. reuse of existing testware
*Reactive: events occur during test execution rather than pre-planning (exploratory)

- Test strategy provides a generalized ~~approach~~ description of test process
- Test approach tailors the test strategy for a particular project or release
- Test approach is the starting point for selecting test techniques, test levels, test types, defining entry & exit criteria (definition of ready, definition of done)

- Entry Criteria: availability of requirements, test items, test environment, test tools, test data.
- Exit Criteria: planned tests are executed, level of coverage achieved, remained defects are in an agreed limit, non-functional requirements are at sufficient level.

Estimation Techniques
- Metric-based technique: burndown chart
- Expert-based technique: planning poker, delphi estimation technique

---

Test progress reports: status, quality, planned actions
Test summary reports: residual risks, coverage, summary, metrics, find status.

Definition of Ready focus on user story level, DoD focus on the sprint or release level.

UES
Unlimited Educational Services®
www.eltturkey.com

Configuration management: version control for changes.

Product risks: functional suitability, reliability, performance efficiency, usability, security, compatibility, maintainability, portability, quality.

Project risks: delays, late changes, inaccurate estimates, lacking of resources
organization related: insufficient skills, conflicts, priorities
political issues: improper attitude towards testing, communication
technical issues: not ready test environment, not well defined requirements
supplier issues: contractual problem, fail to deliver equipment

Risk-based testing: Analyse, prioritize risks, implement mitigation plans, make contingency plans
Testing helps to identify new risks and lowers uncertainty about risks, focus on mitigation
Defect report should include; traceability, screenshot, video recording, expected result
identifier, date, short summary, description to enable reproduction, impact, urgency
change history, test case,

Test Tools

automating repetitive tasks, improving efficiency of manual testing, increasing reliability
Lifecycle Management Tools (ALM) / Requirement Management Tools (for traceability)
Defect Management Tools / Configuration Management Tools / Continuous Integration Tools

*There are test tools for Test design & implementation, static testing, execution, logging
performance management, dynamic analysis

*Test tools provide greater consistency & repeatability, objective assesment

*Test tool (disadvantages): unrealistic expectations, time/cost/effort underestimated,
interoperability issues with other tools, vendor may provide poor support, training, ownership problem

used in exploratory testing
test charter: Statement of test objectives & possibly test ideas about how to test
probe effects: actual response time of a test tool differs due to extra instructions it perform

Richmond
HELBLING
BLACK CAT
ZB Zaner-Bloser

Test charters are produced => a part of test analysis

Functional quality characteristic : completeness, correctness, appropriateness

Non-functional quality characteristic : reliability, efficiency, security, compatibility, usability

Reviews are conducted in the scope of Static testing

Entry criteria: Definition of Ready
Exit criteria: Definition of Done

Test charters are produced as a part of test analysis.

~

Review Process

Activities of review process:

1-Planning : defining entry & exit criteria for formal review
2-Initiate review : distributing materials,
3-Individual review : noting potential defects, recommendation
4-Issue communication and analysis : analysing defects & assigning ownership
5-Fixing & Reporting : creating defect reports.

---

*Test conditions are static rules should be followed to test an application. Test scenarios are an input for the creation of test cases. It gives the main goal to test an application. Test scenario covers all possible cases to test an application

*Test scenario is a way to test an application whereas test condition is a constraint that should be followed for testing an application. Test scenario can be a single or group of test cases whereas test condition is a piece of functionality.

* With two boundary value testing, the boundary value (on the boundary) and the value that is just over the boundary (by the smallest possible increment) are used 1-10 (increment 0,5) upper: 10; 10,5 lower 1,0,5

* For three value boundary testing, the values before, on and over the boundary are used. 0,5, 1, 1,5 (lower) 9,5, 10, 10,5 (upper)