8130 and 4130

Outline

- Test Execution
- Test Environments
- Test Data
- Configuration management/control
- Pass/Fail Criteria
- Test Records
- Test Metrics
- Test Reporting
- Regression Testing
- When is testing complete?
- Defect Management

Test Execution

- Execution is the key to the testing objectives
- The aim is to spend as much time executing as possible
- Execution can be manual or automated
- Execution can be formal (witnessed) or informal
- Formal test execution may need to allow for a number of “dry runs”
  - A dry run is a pass through all the tests. This way the formal “run” should not produce any surprises
  - Does this add value?

Test Execution

- Entry criteria
  - Item to test
    - Ready for test
    - Is defined
    - Under configuration management
    - Passed previous tests
      - Test coverage OK
      - Outstanding defects OK
  - Agreed process and procedure to test
  - Agreed expected results (requirements baseline)
  - Test environment
    - Ready and available
    - Under configuration management

Test Execution

- Entry Criteria (contd.)
  - Resources available
    - Testers
    - Witnesses (if applicable) e.g. QA, Customer
    - Test Administrator / DBA (if applicable)
  - Dry runs completed (if applicable) and known errors are acceptable
  - Can include a Test Readiness Review (TRR) to confirm all entry criteria are satisfied

Test Execution

- Use a Test Execution Matrix
  - List the system states and test cases are required

	Start-up	Day-to day	End-of-day	Roll-over	Balancing	Maintenance	Batch runs	Interface runs
Test 1	x							
Test 2		x	x	x		x	x	
Test 3					x			
Test 4						x		
Test 5		x			x	x		
Test 6				x				x
Test 7	x							

Test Execution

- Use a Test Execution Matrix (contd.)
  - Use the Matrix to determine:
    - If a state has too many tests (may need to split up the state). This should have been determined in the test design phase but can be reviewed here to ensure efficient execution
    - A logical order of testing

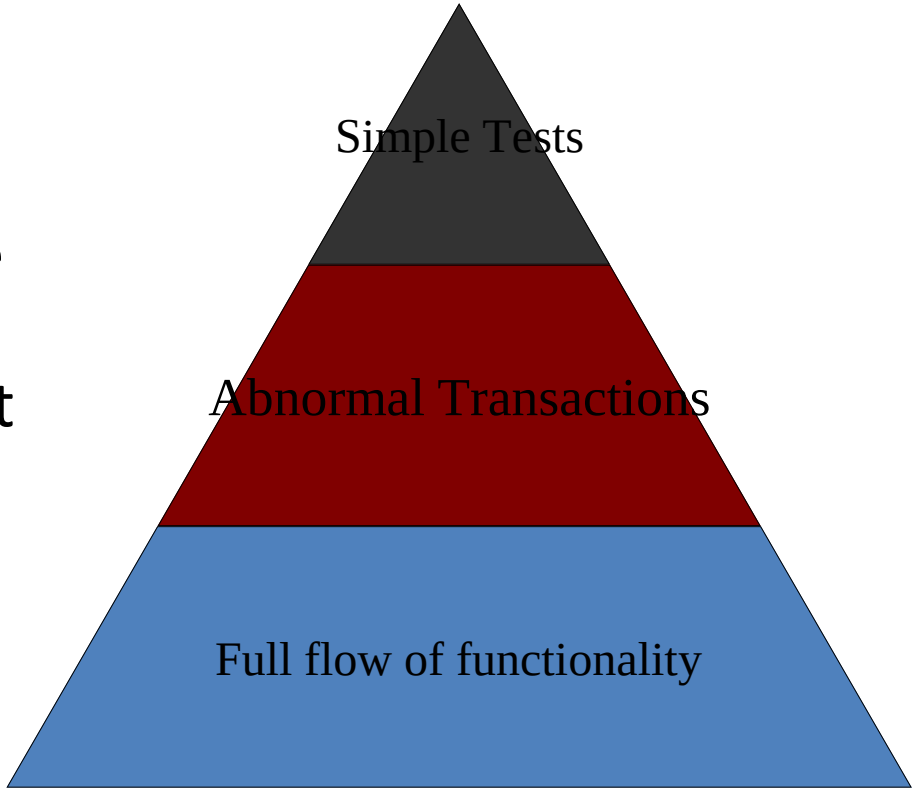
Test Execution

- Test Sequencing
  - Build up the complexity of testing
  - Few simple tests
  - Include some abnormal transactions
  - Add more normal and abnormal transactions
  - Then conduct full flow functionality testing (coverage of all transactions)

Test Execution

- Test Sequencing (continued)

  Most likely included in the test design, however may be adjusted in test execution planning.



Test Execution

Test Sequencing (continued)

- Order of functional availability (obvious)
- Order of functional priority (less obvious if not included in the test design)
- Allow time in the sequence (schedule) for regression testing

Test Execution

- Stick to the documented tests
- Manage any diversions
  - Restrict diversions
  - Document diversions
  - Not an issue with automated tests

- Record Test Results
- Capture additional information if issues arise (log files, screen captures, …)

Test Execution

- Exit Criteria

  Examples of some exit criteria:

  – Test Coverage (e.g. all test procedures completed) on any software build
  – Test coverage on a static build
  – No defects (not possible?)
  – No critical defects
  – System stability (how to measure?)
  – Scheduled release date has occurred (have to release the product!!)

Test Environments

Project Concept / Design

- Paper / Office environment (reviews / Inspections)
- Development environment (unit testing, reviews, inspections)
- Integration environment
- Pre-production environment
- Production environment

In Service (operation)

Test Environment

- Test environments should be under configuration control to ensure environmental impacts are understood
- Test environments include:
  - test hardware platforms,
  - test harnesses (hardware / software),
  - Operating systems,
  - test data, and
  - test management tools

Test Environment

- Consideration needs to be given to the verification of internally developed test harnesses – how is independence maintained if the testers develop these tools?
- Can use interface / network tools to monitor the communications between (sub)-systems

Test Data

- Test Data is part of the test environment
- Examples of test data include:
  - Spreadsheets used to verify the value tables in an office application
  - Customer data
  - Pre-recorded voice data (for voice recognition systems)
  - Simulated flight data

Test Data

- Test data can be fictitious or can be real data already in use by the system
- Fictitious data has the advantage that it can specifically target the test objective (e.g out of bounds values)
- Real data has the advantage that it represents the real world (particularly useful for systems where there is a large amount of data)
- Real world data has the disadvantage that it is sometimes difficult to find the exact conditions to satisfy the testing objective

Test Data

- In the test environment, there must be control over the test data

- Large data driven systems may require the use of a Database Administrator in the test team

Configuration Management/control

- Changes to the Test environment (and test data), and test documentation must be managed to ensure that any new result in the test execution is understood.
- Sources of change could come from:
  - Requirement changes (affects the test documentation)
  - New test data (particularly when using real world data)
  - Software patches
  - Operating system environment updates

Configuration Management/control

- Configuration Management tools include:
  - Configuration management tools used in the development environment
  - Test Management Tools (automated tracing to requirements)

- Manual Configuration Management processes are often used in the control of the physical environment

Configuration Management/control

- One of the challenges in configuration management is the handling of new software builds from development, particularly where these builds are available more frequently than the period of the complete test execution
- One solution is to allow frequent config changes in the early phases of the test execution, but to have one complete test run on a static configuration just prior to product release

Pass / Fail criteria

- Pass / Fail criteria must be understood prior to test execution
- Consideration must be given to:
  - The failure of a single step = failure of the test
  - What to do if a test step passes but some observation unrelated to the test step results in a potential defect
  - Handling errors in the test procedures (eg the system produced the correct result but the test procedure had an incorrect expected result)

Test Records

- Test records must be sufficient to determine:
  - What was tested
  - What passed and what failed
  - What defects occurred
  - Who tested
- Test records should also support:
  - Test scheduling and planning
  - Determination of product quality
  - Determination of product stability

Test Records

- Test records include:
  - Test log or running sheets
  - Test procedures (completed)
  - Discrepancy reports
  - Test Reports
  - Test data (system outputs)

Test Metrics

- Test Metrics provide data to be able to manage the verification and validation processes
- They provide information about the system under test and how testing is progressing
- The collection of data to support the metric should not be too onerous

Test Metrics

Metric	Person or Team Responsible	Reason	Method of Reporting
Test Cases completed (run)	Tester	Identify completion rate of validation test protocol	Graph of % test cases run
Date Discrepancies Raised and Date discrepancies closed	Tester	Indicates if quality is improving. Identifies if defects are being found and fixed at a rate consistent with the schedule	Graph of Discrepancies raised vs. Discrepancies closed over time (the lines should be converging approaching acceptance).
Total number of discrepancies raised	Tester	Part of completion criteria for testing (reliability) – looking for the rate of discrepancy raising to decrease.	Graph of total discrepancies over time (curve should flatten approaching acceptance)
Severity of Discrepancy	Tester	Prioritises discrepancies to assist in an assessment of product acceptability.	Graph discrepancies raised in each of the severity categories
Time to run each test	Tester	Identify how much time should be planned for re-testing	How long it takes to run each test
Software Version Tested	Tester	Identify how current the test results are.	Identify software version for each test run.

Test Reporting

- Test Reports are useful to communicate the outcome of a testing activity
- They provide a decision point in the testing activity (whether the product is ready for the next phase or not).
- The next phase after testing could be release of the software or moving to the next verification / validation activity

Test Reporting

- Test Reports
  - Identify:
    - **What was tested**
    - **When**
    - **By whom**
    - **On what**
    - **Outcomes (pass / fail)**
    - **Issues (reference defect number)**

Regression Testing

- Regression testing aims to verify a new version of software.
- This new version of software may be as a result of fixes to defects and / or the addition of new functionality.
- The issues with regression testing is how much testing of the new / fixed function(s) and how much of the existing functionality (note that it is rarely practical to retest everything).

Stopping

- The big issue is when to stop testing
- Is this when all test cases are finished?
- If so, what happens when the product quality is poor (lots of defects)?
- If this is when there are few defects? Does this imply that testing can be cut short when testing doesn't find defects?
- How do we handle changes in software to address defects near the end of testing?

Stopping

- The key to knowing when to stop testing is in being able to provide some prediction about the level or risk that releasing the product at any point in time is going to present.
- Testing metrics can help in providing some basis for these predictions.
- Next week's presentation should provide with more information in determining when to stop testing.

Summary

- Test Execution is the key to testing
- Execution matrices and sequencing can help to improve efficiency of execution, as does the whole test planning
- Test Environments need to be planned and managed
- Test Data is a part of the test environment and may be fictitious or real-world
- Changes in environment, data, procedures needs to be understood to mange the impacts to the test results
- Test Metrics provide information to manage the testing activities
- Test Reports communicate the outcome of a testing activity
- Regression Testing is used to verify new releases of software
- Stopping testing is accepting a level of risk and the decision to stop should be made on an estimate of that level of risk
- Defect Management involves reporting, investigation, correcting and re-verifying the correction.