Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement

Jeff Tian, tian@engr.smu.edu
www.engr.smu.edu/~tian/SQEbook

Chapter 7. Testing Activities, Management, and Automation

- Major Testing Activities
- Test Management
- Testing Automation
Test Planning and Preparation

- Major testing activities:
  - Test planning and preparation
  - Execution (testing)
  - Analysis and followup

- Test planning:
  - Goal setting
  - Overall strategy

- Test preparation:
  - Preparing test cases & test suite(s)
    (systematic: model-based; our focus)
  - Preparing test procedure
Test Planning

- Goal setting and strategic planning.

- Goal setting
  - Quality perspectives of the customer
  - Quality expectations of the customer
  - Mapping to internal goals and concrete (quantified) measurement.
  - Example: customer’s correctness concerns $\Rightarrow$ specific reliability target

- Overall strategy, including:
  - Specific objects to be tested.
  - Techniques (and related models) to use.
  - Measurement data to be collected.
  - Analysis and followup activities.
  - Key: Plan the “whole thing”!
Test Preparation

- Procedure for test preparation
  - Preparing test cases (model-based)
    - individual test cases
    - test case allocation
  - Preparing test procedure
    - basis for test procedure
    - order, flow, followup

- General concepts
  - Test run: operation instances
  - Input variable: test point
  - Input space:
    - all possible input variable values
  - Test case: static object + input to enable test runs to start-execute-finish.
Individual Test Case Preparation

- Individual test cases (micro-level) vs. test suite (macro-level)

- From multiple sources:
  - Actual runs (usage-based).
  - Implementation-based (white-box).
  - Specification-based (black-box).
  - May use similar/earlier products.
  - (direct) record and replay (less often).
  - (via) formal models (OP, CFT, BT, etc.)

- Defining input values (model \(\Rightarrow\) test cases):
  - Initial/intermediate/interactive input (expected output too?)
  - Exercise path/slice/track/etc
  - In testing terminology: sensitization
Test Cases Based on Formal Models

- Most organized, systematic test cases are derived from formal testing models:
  - Directly via newly constructed models.
  - Indirectly via exist test cases, etc.

- Model construction steps:
  - Information source identification and data collection
  - Analysis and initial model construction
  - Model validation and improvement

- Model usage:
  - Defining test cases.
    (details with individual models/techniques)
  - Indirectly in analysis/followup (Part IV).
Test Suite Preparation

- Test suite (macro-level)
  - Existing suite: what and where?
    - suitability? selection/screening?
  - Construction/generation of new ones
  - Organization & management: often hierarchical.

- Adding new test cases
  - Estimate # of new test cases
  - Specify new (individual) test cases
  - Integrate to existing test cases

- Allocation to systems/operations
  - OP-/structure-based allocation
  - Both old and new test cases in suite
Test Procedure Preparation

- Key consideration: sequencing:
  - General: simple to complex.
  - Dependency among test cases.
  - Defect detection related sequencing.
  - Sequence to avoid accident.
  - Problem diagnosis related sequencing.
  - Natural grouping of test cases.

- Other considerations:
  - Effectiveness/efficiency concerns.
  - Smooth transition between test runs.
  - Management/resource/personnel/etc.
Test Execution

- Major testing activities:
  - Test planning and preparation
  - Execution (testing)
  - Analysis and followup

- Test execution:
  - Execution planning and management
  - Related activities: important part
    - failure identification and measurement
    - other measurement
Test Execution

- General steps
  - Allocating test time (& resources)
  - Invoking test
  - Identifying system failures
    (& gathering info. for followup actions)

- Allocating test time
  - OP-based: systems/features/operations
  - Coverage concerns for critical parts
  - Coverage-based: func./struc. areas
  - Alternative: bottom-up approach
    - individual test cases ⇒ test time
    - sum-up ⇒ overall allocation
    - by OP or coverage areas
Test Execution

- Invoking test (OP-based)
  - OP $\Rightarrow$ input variables (test points)
  - Follow probabilistic distributions (could be dynamically determined)
  - Sequence (what to test first?): COTS, product, supersystem

- Invoking test (coverage-based)
  - Organize sensitized testcases
  - Sequence $\leftarrow$ coverage hierarchies

- Common part: Retest due to
  - Defect fix $\Rightarrow$ verify fix
  - Code-base or feature change
  - General regression test
Test Execution

- Identifying system failures (oracle problem):
  - Similar for OP-/coverage-based
  - Analyze test output for deviations
  - Determine: deviation $\neq$ failure?
  - Handling normal vs. failed runs
    - non-blocking failure handling

- Solving oracle problem:
  - Theoretically undecidable.
  - Some cases obvious: crash, hang, etc.
  - Practically based on heuristics:
    - product domain knowledge
    - cross-checking with other products
    - implementation knowledge & internals
    - limited dynamic consistency checking
Test Execution

- Failure observation and measurement:
  - When determining deviation = failure
  - Establish when failure occurred
    - used in reliability and other analysis
  - Failure information (e.g., ODC):
    - what/where/when/severity/etc.

- Defect handling and test measurement:
  - Defect status and change (controlled)
  - Information gathering during testing:
    - example template: Table 7.1 (p.93)
  - Followup activities:
    - fix-verification cycle
    - other possibilities (defer, invalid, etc.)
Testing Analysis and Followup

- Major testing activities:
  - Test planning and preparation
  - Execution (testing)
  - Analysis and followup

- Test analysis and followup:
  - Execution/other measurement analyzed
  - Analysis results as basis for followup
  - Feedback and followup:
    - decision making (exit testing? etc.)
    - adjustment and improvement.
Testing Analysis and Followup

- Input to analysis
  - Test execution information
  - Particularly failure cases
  - Timing and characteristics data

- Analysis and output
  - Basic individual (failure) case
    - problem identification/reporting
    - repeatable problem setup
  - Overall reliability and other analysis? (Module V)

- Followup activities
  - Defect analysis and removal (& re-test).
  - Decision making and management.
  - Test process and quality improvement.
Testing Analysis and Followup

- For individual test runs:
  - Success, continue with normal testing.
  - Failure: see below.

- Analysis and followup for failed runs:
  - Understanding the problem by studying the execution record.
  - Recreating the problem (confirmation).
  - Problem diagnosis
    - may involve multiple related runs.
  - Locating the faults.
  - Defect fixing (fault removal)
    - commonly via add/remove/modify code
    - sometimes involve design changes
  - Re-run/re-test to confirm defect fixing.
Testing Analysis and Followup

- Analysis and followup for overall testing:
  - Reliability analysis and followup.
  - Coverage analysis and followup.
  - Defect analysis and followup.
  - Focus of Part IV.

- Analyses: Different focuses:
  - Overall reliability and coverage for usage-based and coverage-based testing.
  - Detailed defect analysis.

- Followup activities: Similar.
  - Decision making and management.
  - Test process and quality improvement.
Test Management

- People’s roles/responsibilities in formal and informal testing.

- In informal testing:
  - “run-and-observe” by testers.
  - “plug-and-play” by users.
  - Informal testing with ad-hoc knowledge
  - Deceptively “easy”, but not all failures or problems easy to recognize.

- In formal testing:
  - Testers, and organized in teams.
  - Management/communication structure.
  - Role of “code owners” (multiple roles?)
  - 3rd party (IV&V) testing.
  - Career path for testers.
Test Management

• Test team organization:
  ▶ Vertical: Project oriented
    – product domain knowledge,
    – staffing/resource management hard.
  ▶ Horizontal: Task oriented
    – even distribution of staff/resources
    – lack of internal knowledge/expertise
  ▶ Mixed models might work better.

• Users and 3rd party testers:
  ▶ User involvement in beta-testing and other variations (e.g., ECI in IBM)
  ▶ IV&V with 3rd party testing/QA
  ▶ Impact of new technologies:
    – CBSE, COTS impact
    – security, dependability requirements.
Test Automation

- Basic understanding:
  - Automation needed for large systems.
  - Fully automated: Impossible.
  - Focus on specific needs/areas.

- Key issues to consider:
  - Specific needs and potentials.
  - Existing tools available/suitable?
    - related: cost/training/etc.
  - Constructing specific tools?
  - Additional cost in usage & support.
  - Impact on resource/schedule/etc.
Test Automation

- Automation by test activity areas:
  - Automated test planning & preparation.
  - Automated test execution.
  - Automated test measurement, analysis, and followup.
  - Slightly different grouping due to tightly coupling for measurement & analysis.

- Automation for test execution.
  - Many debuggers: semi-automatic.
  - Task sequencing/scheduling tools.
  - Load/test generator: script $\Rightarrow$ runs
  - Generally easier to obtain test scripts.
Test Automation

- Automation for test planning/preparation:
  - Test planning: Human intensive not much can be done (≈ inspection and FV).
  - Test model construction: similar to above.
    - automation possible at a small scale.
  - Test case generation: focus.

- Test case generation:
  - From test model to test cases.
  - Specific to individual techniques
    - e.g., cover checklist items, paths, etc.
  - Various specific tools.
  - Key: which specific testing technique supported by the specific tool?
Test Automation

- Test measurement, analysis, and followup.
  - Analyses dictate measurements needed.
  - Most common: reliability/coverage.
  - Defect measurement needed in most cases:
    - defect tracking tools.

- Reliability analysis related tools:
  - Analysis/modeling tools.
  - Collecting execution/input/etc. data.
  - More in Chapter 22.
**Test Automation**

- Coverage-based testing: measuring coverage and compare to pre-set goals.

- Test coverage steps:
  - Preparation: program instrumentation.
  - Measurement step: run and collect data.
  - Analysis step: analysis for coverage.
  - Example: Fig 7.1 (p.100).

- Test coverage tools:
  - Different levels/definitions of coverage ⇒ different tools.
  - Example tools:
    - McCabe: execution (control flow) path
    - S-TCAT: functional coverage
Summary

- Test activities:
  - Planning & preparation: focus of Part II.
  - Execution & measurement: common.
  - Analysis & followup: focus of Part IV.

- Test management:
  - Different roles and responsibilities.
  - Good management required.

- Test automation:
  - Set realistic expectations.
  - Specific areas for automation, esp. in execution, measurement, and analysis.