Chapter 6. Testing Overview

- Testing: Concepts & Process
- Testing Related Questions
- Major Testing Techniques
Testing and QA Alternatives

- Defect and QA:
  - Defect: error/fault/failure.
  - Defect prevention/removal/containment.
  - Map to major QA activities

- Defect prevention:
  Error blocking and error source removal.

- Defect removal:
  - Testing – Part II, Ch.6-12.
  - Inspection, etc.

- Defect containment: Fault tolerance and failure containment (safety assurance).
QA and Testing

• Testing as part of QA:
  ▶ Activities focus on testing phase
  ▶ Related act. throughout dev. process
    – see Fig 4.1 (p.45) and Fig 4.2 (p.48)
  ▶ One of the most important part of QA
    – QA/defect context: Fig 3.1 (p.30)

• Testing: Key questions:
  ▶ Why: quality demonstration vs. defect detection and removal
  ▶ How: techniques/activities/process/etc.
  ▶ View: functional/external/black-box vs. structural/internal/white-box
  ▶ Exit: coverage vs. usage-based
Testing: Why (and How)?

- Original purpose: demonstration of proper behavior or quality demonstration.
  ≈ “testing” in traditional settings.
  ▶ Evidence of quality or proper behavior.

- New purpose: defect detection & removal:
  ▶ Mostly defect-free software manufacturing vs. traditional manufacturing.
  ▶ Flexibility of software (ease of change; sometimes, curse of change/flexibility)
  ▶ Failure observation ⇒ fault removal.
    (defect detection ⇒ defect fixing)
  ▶ Eclipsing original purpose

- How? Run-observe-followup
  (particularly in case of failure observations)
Testing: Activities & Generic Process

- Generic Process: Fig 6.1 (p.69).
  - Instantiation of SQE in Fig 5.1, p.54.
  - Planning-execution-analysis-feedback.
  - Link major testing activities.
  - Entry criteria: typically external.
  - Exit criteria: internal and external.
  - Some (small) process variations
    - but we focus on strategies/techniques.

- Major testing activities:
  - Test planning and preparation
  - Execution (testing)
  - Analysis and followup
    (decision making and management too)
Testing: Planning and Preparation

- Test planning:
  - Goal setting based on customers’ quality perspectives and expectations.
  - Overall strategy based on the above and product/environmental characteristics.

- Test preparation:
  - Preparing test cases/suites:
    - typically based on formal models.
  - Preparing test procedure.

- More details in Chapter 7.
**Testing: Execution**

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

- Key to execution: handling both normal vs. abnormal cases

- Activities closely related to execution:
  - Failure identification:
    - test oracle problem
  - Data capturing and other measurement

- More details in Chapter 7.
Testing: Analysis and Followup

- Analysis of testing results:
  - Result checking (as part of execution)
  - Further result analyses
    - defect/reliability/etc. analyses.
  - Other analyses: defect \sim other metrics.

- Followup activities:
  - Feedback based analysis results.
  - Immediate: defect removal (& re-test)
  - Other followup (longer term):
    - decision making (exit testing, etc.)
    - test process improvement, etc.

- More details in Chapter 7 (for activities) and Part IV (for mechanisms/models/etc.).
Testing: How?

• How to test?
  ▶ Refine into three sets of questions.
  ▶ Basic questions.
  ▶ Testing technique questions.
  ▶ Activity/management questions.

• Basic questions:
  ▶ What artifacts are tested?
  ▶ What to test?
    – from which view?
    – related: type of faults found?
  ▶ When to stop testing?
  ▶ Addressed in this Chapter.
Testing Technique Questions

- Testing technique questions:
  - Specific technique used?
  - Systematic models used?
    - related model questions (below)
  - Adapting technique from other domains?
  - Integration for efficiency/effectiveness?

- Testing model questions:
  - Underlying structure of the model?
    - main types: list vs. FSM?
  - How are these models used?
  - Model extension?
  - Details in Chapters 8–11.

- Major techniques: Chapters 8–11.
Test Activity/Management Questions

- Addressed already: Generic process and relation to QA and software processes.

- Other activity/management questions:
  - Who performs which specific activities?
  - When can specific activities be performed?
  - Test automation? What about tools?
  - Artifacts used for test management?
  - General environment for testing?
  - Product type/segment?

- Most questions answered in Chapter 7. Integration issues addressed in Chapter 12.
Functional vs. Structural Testing

- Key distinction: Perspective on what need to be checked/tested.

- Functional testing:
  - Tests external functions.
    - as described by external specifications
  - Black-box in nature;
    - functional mapping: input $\Rightarrow$ output
    - without involving internal knowledge

- Structural testing:
  - Tests internal implementations.
    - components and structures.
  - White-box in nature;
    - “white” here $=$ seeing through
      $\Rightarrow$ internal elements visible.
  - Really clear/glass/transparent box.
Black-Box vs. White-Box View

- Object abstraction/representation:
  - High-level: whole system \(\approx\) black-box.
  - Low-level: individual statements, data, and other elements \(\approx\) white-box.
  - Middle-levels of abstraction:
    - function/subroutine/procedure, module, subsystem, etc.
    - method, class, super-class, etc.

- Gray-box (mixed black-/white-) testing:
  - Many of the middle levels of testing.
  - Example: procedures in modules
    - procedures individually as black box,
    - procedure interconnection \(\approx\) white-box at module level.
White-box Testing

- Program component/structure knowledge (or implementation details)
  - Statement/component checklist
  - Path (control flow) testing
  - Data (flow) dependency testing

- Applicability
  - Test in the small/early
  - Dual role of programmers/testers
  - Can also model specifications

- Criterion for stopping
  - Mostly coverage goals.
  - Occasionally quality/reliability goals.
Black-box Testing

- Input/output behavior
  - Specification checklist.
  - Testing expected/specified behavior
    - finite-state machines (FSMs)
  - White-box technique on specification
    - functional execution path testing.

- Applicability
  - Late in testing: system testing etc.
  - Suitable for IV&V
  - Compatible with OO/Reuse paradigm

- Criteria: when to stop
  - Traditional: functional coverage
  - Usage-based: reliability target
When to Stop Testing

- Resource-based criteria:
  - Stop when you run out of time.
  - Stop when you run out of money.
  - Irresponsible $\Rightarrow$ quality/other problems.

- Quality-based criteria:
  - Stop when quality goals reached.
  - Direct quality measure: reliability
    - resemble actual customer usages
  - Indirect quality measure: coverage.
  - Other surrogate: activity completion.
  - Above in decreasing desirability.
Usage-Based Testing and OP

- Usage-based statistical testing:
  - Actual usage and scenarios/information
  - Captured in operational profiles (OPs)
  - Simulated in testing environment
    (too numerous ⇒ random sampling)

- Applicability
  - Final stages of testing.
  - Particularly system/acceptance testing.
  - Use with s/w reliability engineering.

- Termination criteria: reliability goals
Coverage-Based Testing

• Coverage-based testing:
  ▶ Systematic testing based on formal models and techniques.
  ▶ Testing models based on internal details or external expectations.
  ▶ Coverage measures defined for models.
  ▶ Testing managed by coverage goals.

• Applicability
  ▶ All stages of testing.
  ▶ Particularly unit and component testing.
  ▶ Later phases at high abstraction levels.

• Termination criteria: coverage goals
Systematic Testing Steps

- Instantiation of Fig 6.1 (p.69), but,
  - with a formalized strategies/goals,
  - based on formal models and techniques,
  - managed by termination criteria.

- Steps in model construction and usage:
  - Define the model, usually represented as graphs and relations.
  - “Check” individual elements:
    - “Test”: derive (sensitize) test cases and then execute them.
  - Result checking and followup.

- Specifics on model construction and usage in individual testing techniques: Ch.8–11.