Lecture 13: Introduction to Testing

Kenneth M. Anderson

Foundations of Software Engineering

CSCI 5828 - Spring Semester, 2000

Today's Lecture

- Introduction to
  - Testing concepts
  - Testing terminology
  - Testing Strategies

Testing

- Experiments with Behavior
- Requires Execution Model
- Executing a System to Observe its Behavior Can be Expensive
- Testing is "Easy" if the System is Deterministic and Takes No Inputs
- Exhaustive Testing is Usually Impractical

Modeling for Software Testing

- Formal Models of Programs Are Employed
  - To make the process of testing programs systematic
  - To increase the probability that testing will reveal faults

Testing Formalized: Basics

- Let *P* be a program, *D* be the input domain of *P*, and *R* be the output range of *P*; *P* acts as a function $P : D \rightarrow R$
- Let R_O denote the requirements on output values of *P*, as stated in *P*'s specification; *P* is *correct* iff for all $d \in D$, *P(d) satisfies* R_O

Testing Formalized: Basics

- An *error* (or *defect*) is demonstrated by showing that *P(d)* is incorrect for some *d*
- A *failure* is a symptom of an error
- A *fault* is an incorrect intermediate state
- A failure occurs only if a fault occurs, and a fault occurs only if an error exists

Testing Formalized: Test Cases

- A test case is an element d of D
- A test set T is a finite subset of D
- P is correct for T if it is correct for all elements of T; T is called successful for P
- T is ideal if, whenever P is incorrect, there exists $d \in T$ such that P is incorrect for d
- If T is ideal and T is successful for P, then P is correct

Testing Formalized: Test Selection

- A test selection criterion C is a subset of 2^D (the set of all finite subsets of D); C gives a condition that must be satisfied by a test set
- T satisfies C if it belongs to C
- C is consistent if, for any pair T1 and T2, both satisfying C, T1 is successful iff T2 is successful

Testing Formalized: Test Selection

- *C* is *complete* if, whenever *P* is incorrect, there is an unsuccessful *T* that satisfies *C*
- If *C* is consistent and complete, then any *T* satisfying *C* could be used to decide the correctness of *P*
- C_1 is *finer* then C_2 if, for any *P*, for all T_1 satisfying C_1, there exists T_2 subset of T_1 and T_2 satisfies C_2

Testing Strategies

- Based on Experience and Intuition
  - Empirical basis for "good" testing criteria
  - Automated support for clerical/repetitive chores
- Testing Criteria are Used to Choose Representative Test Cases
  - Criteria group inputs into equivalence classes
  - Reduces the number of test cases

Testing Strategies

- Principle of Complete Coverage
  - If all the classes together exercise the whole input, then coverage is complete
- The Partition Advantage
  - If classes are a *partition* of *D*, then any element of a class will do
- Partition Overlap
  - If a criteria overlaps more than one partition, then a good representative test case can reduce the number of test cases needed overall

Testing Approaches

- Black Box Testing
  - Tests are selected based on specification of intended functionality
  - Tester can only see interface to test subject
  - Emphasis on proper *use* of test subject
- White Box Testing
  - Tests are selected based on internal structure
  - Tester can see inside test subject
  - Emphasis on proper *structure* of test subject