

Chapter 1

Failure(失败): Event that occurs when the delivered service deviates from correct services.

Fault(故障): Adjudged or hypothesized cause of an error.

Error(错误): Part of the system state that may cause a subsequent failure.

Why test generate is hard to automate: Oracle problem – A test oracle is a mechanism that determines whether a software executed correctly for a test case. We define a test oracle to contain two essential parts: oracle information that represents expected output, and an oracle procedure that compares the oracle information with the actual output.

Test case: Number of the case, module to test, severity, input, expected output.

Format of test case:

1	A. Register and Login 003, Bad Register 002, Severity 2 B. <ul style="list-style-type: none">• Enter information with wrong format• Click on “Done” C. Popup box appears saying “Wrong format”
---	--

Severity levels: 1. Critical impact (can't perform normal function)
2. General / Functional (functional limitations but not critical)
3. Quality / Usability (not frequently used)

Regression testing:

Defination: Uncover new software bugs in existing functional and non-functional areas of a system after changes such as enhancements, patches or configuration changes have been made to them. (检验软件原有功能在修改后是否保持完整)

Procedure: After modifying from P to P', reusing tests derived for P. Ensure that P' works correctly.

Law of conservation of bugs: The number of bugs remaining in a large system is proportional to the number of bugs already fixed.

Code inspection:

Procedure: Planning → Overview meeting → Preparation → Inspection meeting → Rework → Follow-up

Roles: Author, Moderator (leader of inspection), Recorder (documents the defects), Inspector (examines the work product)

Inspection vs. Testing: Different kinds of bugs; which one go first? ← **INSPECTION**

Testing: mock up a production environment to be used to see bugs

Inspection: involve testing and code review, efficient, readability and maintainability.
Chance of mistakes are reduced if both activities are performed.

Three different test generation:

Mutation testing: mutant, killing mutant (tests detect and reject mutants by causing the behavior of the original version to differ from the mutant), test suites are measured by the percentage of mutants that they kill.

Search based test generation: (genetic test generation) input A + input B \rightarrow input C

Dynamic test generation (symbolic execution): prog \rightarrow logic formula \rightarrow solver produces the input

```
E.g. int A[21]
x = input
if (x > 10)
    if (x < 13)
        A[10+x] = ...
x > 10 and x < 13 and 10 + x > 21
```

Chapter 2

What is software testing: A process of executing a program or application with the intent of finding the software bugs.

What does it mean by a test is successful: expose a bug.

Why test is expensive:

Black Box Testing: input domain is huge

White Box Testing: number of paths in the program is huge

Testing principles:

1. A necessary part of a test case is a definition of the expected output or result.
2. A programmer should avoid attempting to test his or her own program.
3. A programming organization should not test its own programs.
4. Any testing process should include a thorough inspection of the results of each test.
5. Test cases must be written for input conditions that are invalid and unexpected, as well as for those that are valid and expected.
6. Examining a program to see if it does not do what it is supposed to do is only half the battle; the other half is seeing whether the program does what it is not supposed to do.
7. Avoid throwaway test cases unless the program is truly a throwaway program.
8. Do not plan a testing effort under the tacit assumption that no errors will be found.
9. The probability of the existence of more errors in a section of a program is proportional to the number of errors already found in that section.
10. Testing is an extremely creative and intellectually challenging task.

Chapter 3

Chapter 4

Black Box Testing:

Equivalence class partitioning:

Definition: Partition the input domain of a program into a finite number of equivalence classes such that you can reasonably assume that a test of a representative value of each class is equivalent to a test of any other value.

Procedure: Identify the equivalence classes; Define the test cases.

Equivalent: equal classes representing input conditions.

Input condition: valid classes (one input as many valid class a possible)

invalid classes (each invalid class needs to be covered by a unique input)

Bounding value analysis: input range: $4n/6n$ (min, min-1, max, max+1)
output boundaries

White Box Testing:

SC: statement coverage (weakest), execute every statement at least once

DC: decision coverage, write enough test cases that exercise each possible outcome of all decisions at least once.

CC: condition coverage, write enough test that ensure each condition in a decision takes all possible outcomes at least once.

MCC: modified condition coverage

PC: path coverage, by examine which lines of executable code are visited and also the ways of getting to each line of code and subsequent sequence of execution.

E.g. if (x > 10 && y < 5)
 if (x > 20 && x - y > 30)

 S1

 else

 S2

else

 S3

Input: t1: x = 9, y = 0;

 t2: x = 25, y = -4

SC? DC? CC?

Limitations of BBT (hidden implementation logic) and WBT (missing code)

Chapter 5

Unit Testing: testing part individually.

Incremental Testing and Big Bang Testing:

Incremental: top-down (link from top) – need drivers / bottom-up (combined from bottom) – need stubs.

Big Bang: integration all modules in one shot.

Chapter 6

Higher Order Testing:

Module test → Integration test → Function test → System test → Acceptance test → Installation test

Function test: a process of attempting to find discrepancies between the program and the external specification.

System test: compare program with objectives and user language.

(usability / volume / stress testing)

Stress testing: subjecting the program to a peak volume of data over short span of time.

Chapter 7

Usability Testing:

Is BBT, testing program in real life: test plan → select user → document → analyze.

Hallway testing: data collection: tape, observer

Remote user testing

Chapter 8

Agile Testing: A form of collaborative testing, in that everyone is involved in the process through design, implementation, and execution of the test plan.

Chapter 9

Internet Apps:

T1 (representation layer: font, color, web layer): the layer of user interface, front end.

T2 (basic logic layer): model the business process, like user authentication and transactions. Focus finding logic on internet application.

T3 (data layer): the layer that collected from user, testing the database.

Security Testing:

SQL injection: the untrusted input is used to construct dynamic SQL queries.

XSS: cross-site scripting, inject client-side script into web pages viewed by other users.

Selenium: software testing framework for web applications, IDE + driver

Chapter 10

Mobile Testing: unique challenges: connectivity (network), device diversity