




A Brief essay on software testing

Antonio Bertolino and Eda Marchetti

test **★★**

미국 영국 [test]   | 영국식  


 발음듣기 단축키

파생형 명사형 **testability** | 형용사형 **testable** | **testing**

동사 과거 **tested** | 과거분사 **tested** | 현재분사 **testing** | 3인칭 단수 현재 **tests**

유의어/반의어 [명사] **trial, research, check, ...** [동사] **check, investigate, assess, ...** [더보기](#)

+ 단어장에 추가

 인쇄

[명사](#) | [동사](#) | [영영사전](#)

뜻+예문 함께 보기 뜻만 보기

명사

- 1. OF KNOWLEDGE/ABILITY** | ~ (on sth) (지식·능력 등을 알아보기 위한) 시험 [테스트/검사] [예문](#)
- 2. OF HEALTH** | (의료적인) 검사 [테스트] [참고](#) blood test, breath test [예문](#)
- 3. OF MACHINE/PRODUCT, etc.** | 실험, 테스트 [참고](#) acid test, blind test, field test, means test, road test [예문](#)
- 4. OF STRENGTH, etc.** | 시험대, 시금석 [예문](#)
- 5. IN CRICKET, etc.** | Test [동의어](#) Test match

Type of Test

- Static Technique test
- Dynamic Technique test

Type of Test

■ Static Technique

- Can be employed throughout development.
Ex) requirement stage, design phase
- Examination : project documentation, Models, code, etc...
- Include : software inspection, review, code reading.
- Have to enforce rigorous, unambiguous for formal verification.
Ex) model checking
- Yield generally valid result.
- But maybe weak in precision.

Type of Test

■ Dynamic Techniques

– Observing some executions.

Ex) profiling, simulation, prototyping...

– Testing

- Finding failure.
- Execution of the code on input value,
- On a **finite** set of test cases,
- Suitably **selected** from usually infinite execution domain,
- Against the specified **expected** behavior,

Terminology

- Failure

- Inability of program to perform the function.
- Incorrect output.
- Abnormal termination.

- Fault

- Cause of failure.
- Missing or incorrect piece of code.

- Error

- Intermediate unstable state.
- Eventually causes the failure.

Type of Test

■ Dynamic Techniques

– Observing some executions for finding failure.

Ex) profiling, simulation, prototyping...

– Testing

- Execution of the code on input value,
- On a **finite** set of test cases,
- Suitably **selected** from usually infinite execution domain,
- Against the specified **expected** behavior,

Index

1. Test Level

2. Strategies for test case selection

3. Test Design

4. Test Execution

5. Test Documentation

6. Test Measurement

7. Test Management

1. Test Level (1/3)

- Unit test

- The smallest testtable piece.
- To ensure unit satisfies function or structure matches design.

- Integration test

- For a larger component(aggregated pieces or unit).
- Focus on component interacts.
- Test for traditional system : focus on hierarchy.
- Thread-based testing : integrating based on functional thread.
- Other types : focus on structure or architecture.

1. Test Level (2/3)

■ System test

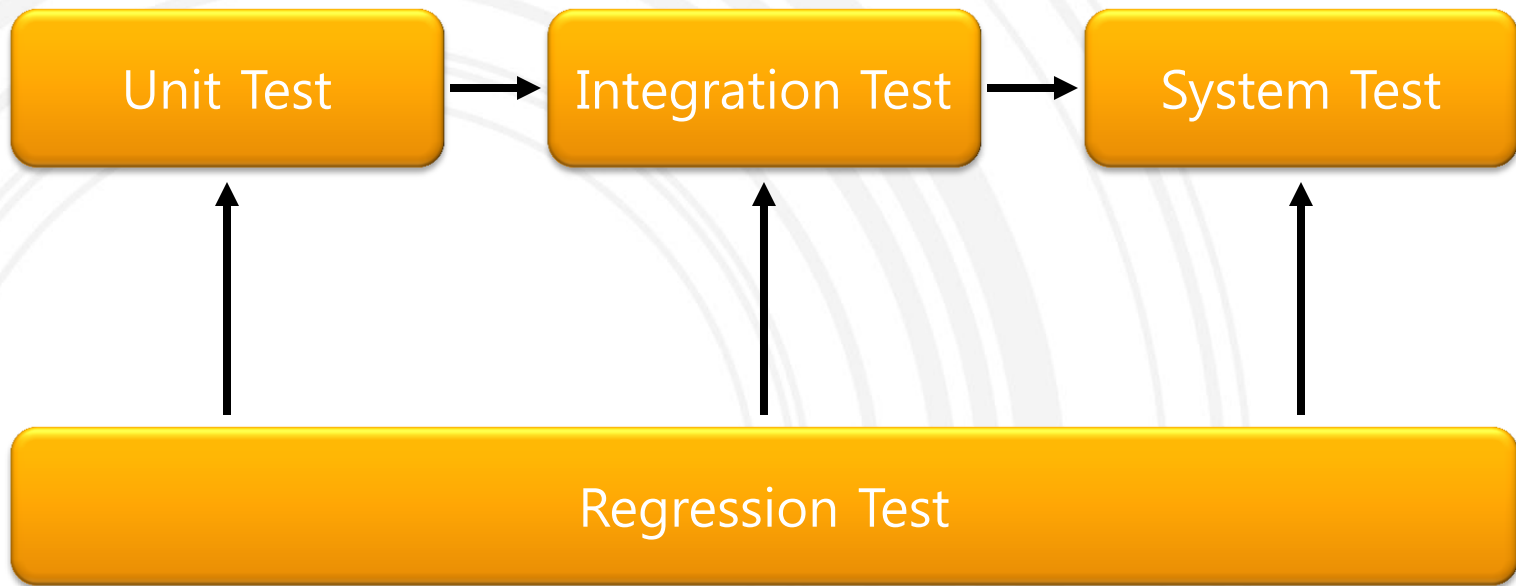
- Involves the whole system.
- According to the user requirement.
- Reveal bug, discovering failures.
- Collecting information useful for deciding release.
- Include functional and non-functional properties.

■ Regression test

- After modification.
- To check that the change has not made new faults
- Prohibitively expensive.

1. Test Level (3/3)

- Logical Schema of software testing levels

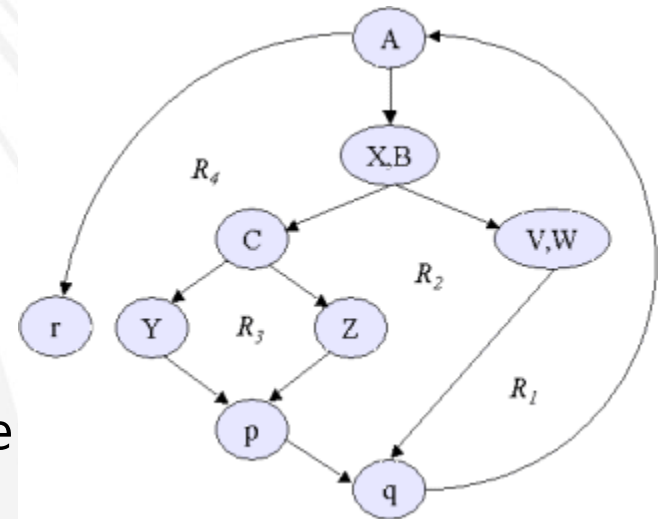


2.1 Strategies for Test Case Selection

- Lack of commercial automated tool.
- Tester's intuition affect selecting criterion.
- Criterion
 - Defined on triples,
 - Program
 - Reference model related Program
 - Test suite
 - ☞ Provides a decision procedure.

2.2 Based on Code

- Called "structural testing" or "white-box testing"
- Executing code that related to fault
- Code coverage criteria
 - Flow-graph(*)
 - Statement coverage
 - Each statement be exercised at least once
 - Branch coverage
 - True and false be tested at least once
- Limitation
 - Legacy complex system



2.3 Based on Specification

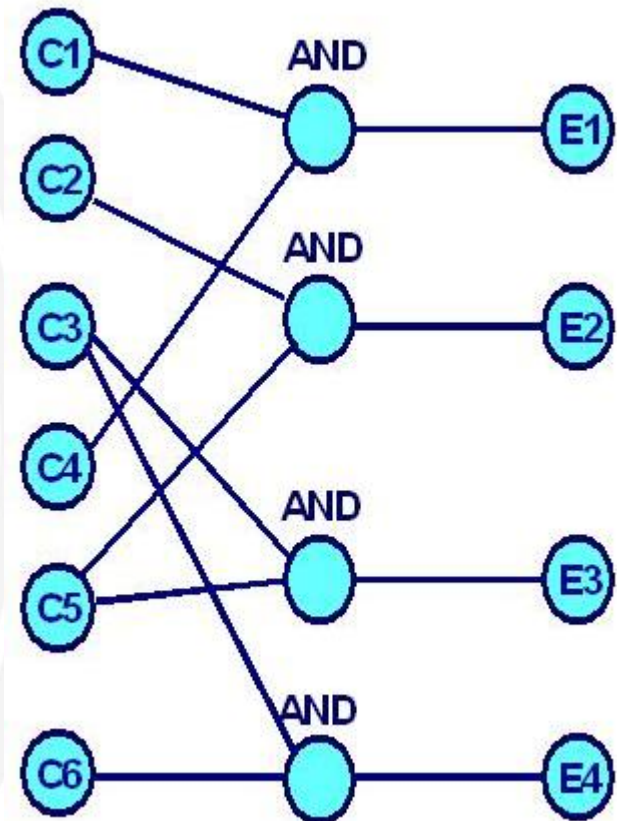
- Treat software as black-box
- Approaches
 - Equivalence classes
 - Boundary conditions
 - Cause-effect graph
 - Category-partition

2.3 Based on Specification - Approaches(1/2)

- Equivalence partitioning
 - First, identify the equivalence classes
 - Second, partitioning input domain into subdomains
 - Third, choosing the inputs representative of subdomain
- Boundary condition
 - Base : faults are more likely to be found at the boundaries

2.3 Based on Specification - Approaches(2/2)

- Cause-effect graph
 - The relevant input condition, causes.
 - The consequent transformations and output, effects.
 - Modeled into graphs.
- Category-partition
 - Automated generating functional test.
 - Analyze functional requirement.
 - Divide system into functional unit.
 - Identify condition & parameters.



2.4 Other Criteria

- Based on Tester's intuition and experience
 - Ad-hoc testing
 - Useful for those not easily captured by formalized technique
 - Exploratory testing
 - Simultaneous learning, test design, execution
 - Dynamical process
- Fault-Based
 - Mutant testing
- Based on operational usage
 - Testing for reliability evaluation

3. Test Design



- Establishing test objective
- Define the test case specification
- Design test procedure
- Define pass/fail criteria

4. Test Execution(1/2)

- Launching Testing

- Forcing execution according to the criteria.

- Oracle

- Determining whether a test has passed or failed.
- Observe failure : *Reject*, Otherwise : *Approve*.
- Doesn't reach a decision : *Inconclusive*.
- Can be a person(tester),
- Can be earlier version,
- But generally derived from specification.

4. Test Execution(2/2)

- Test tools
 - Test harness
 - Test generators
 - Capture / replay
 - Oracle / file comparator / assertion checking
 - Coverage analyzer / instrumenter
 - Tracer

5. Test Documentation (1/2)

■ Documentation

- Formalization of the test process
- For the coordination and control of the testing phase
- Type
 - Test plan : defines test item
 - Test design specification : describes the features to be tested
 - Test case specification : defines the i/o required for test case
 - Test log : the result of a test execution, including failures
 - Test incident or problem report : describes all problem during the test

5. Test Documentation (2/2)

- Documentation example – test case specification

Test case	Specification
Test case ID	The unique identifier
Test item & purpose	The items and features exercised
Input data	The list of the input
Test case Behavior	Description of the expected test case behavior
Output	The list of the outputs admitted for each feature
Environmental Setup	The hardware / software configuration required
Specific procedural Requirement	The constraints and the special procedure required
Test case dependencies	The IDs of test cases that must be executed prior to this case

6. Test Measurement(1/3)

■ Objective

- Generating descriptions of key process and product
 - Controlling software behavior and result
 - For understanding the nature and impact of proposed changes
 - Monitoring the effect of activities and changes
- ☞ We can check how final outcome differs

6. Test Measurement(2/3)

- Evaluation of the program under test
 - Program measurement
 - Linguistic measures : based on properties or specification text
 - Structural measures : based on structural relationship
 - Hybrid measures
 - Fault density
 - Counting discovered faults and Classifying by type
 - The ratio : number of faults / size of program
 - Life Testing
 - Evaluating reliability

6. Test Measurement(3/3)

- Evaluation of the test performed
 - Coverage / thoroughness measure
 - The ratio : the covered element / total number.
Ex) the number of functionalities exercised during testing.
 - Effectiveness
 - Associated with test case or an entire test suite.
- Measures for monitoring the test process
 - Statistical measure
 - Basis of software reliability

7. Test Management

▪ Manager's Activity

- Schedule the timely completion of tasks
- Estimate required resources (including tools & people)
- Quantify the risk associated with the task
- Estimate cost using models
 - Static model : use historical data to derive empirical relationship
 - Dynamic model : use project resource requirement
- Decide quality control measures

Finish

Thank you for listening
My presentation