Software testing

Basic concepts

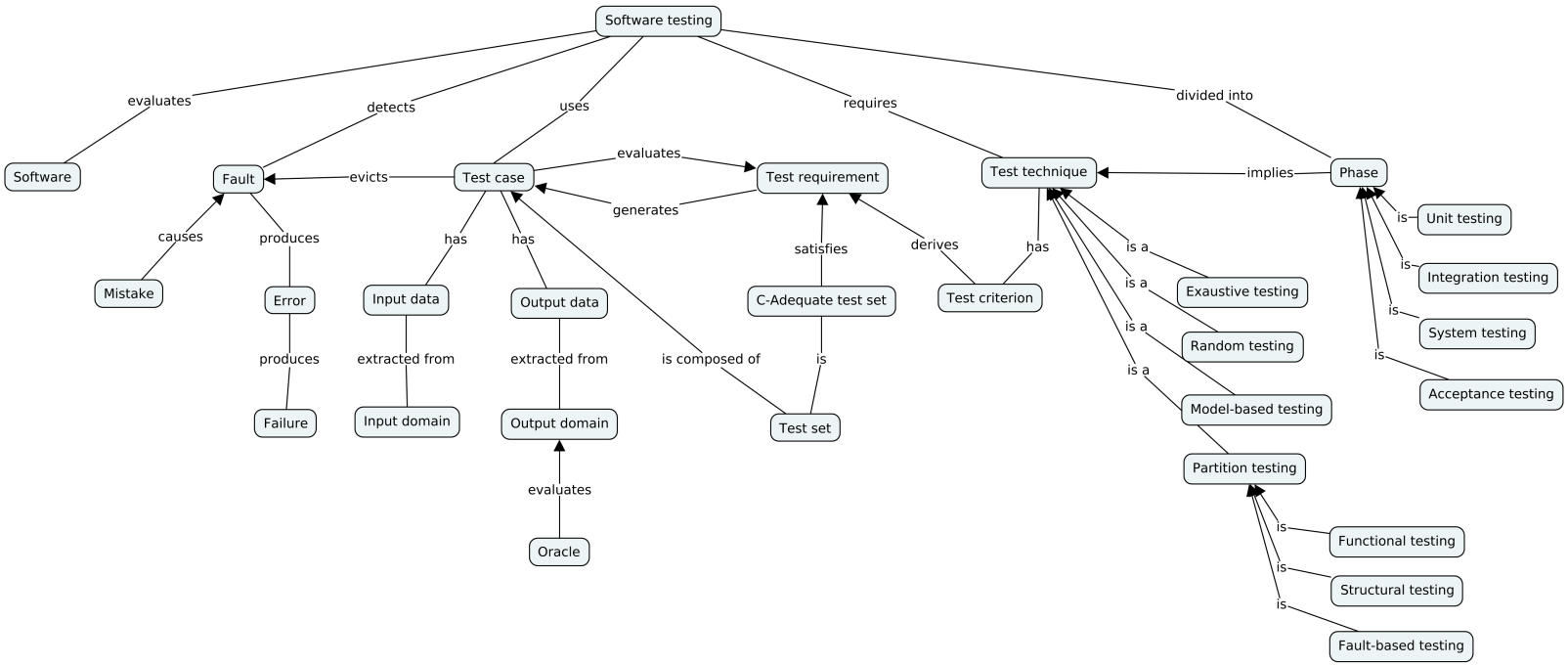
Marco Aurélio Graciotto Silva[1], Ellen Francine Barbosa[2], José Carlos Maldonado[2]

[1]**Department of Computing**
Federal University of Technology – Paraná (UTFPR)
Campo Mourão, PR, Brazil

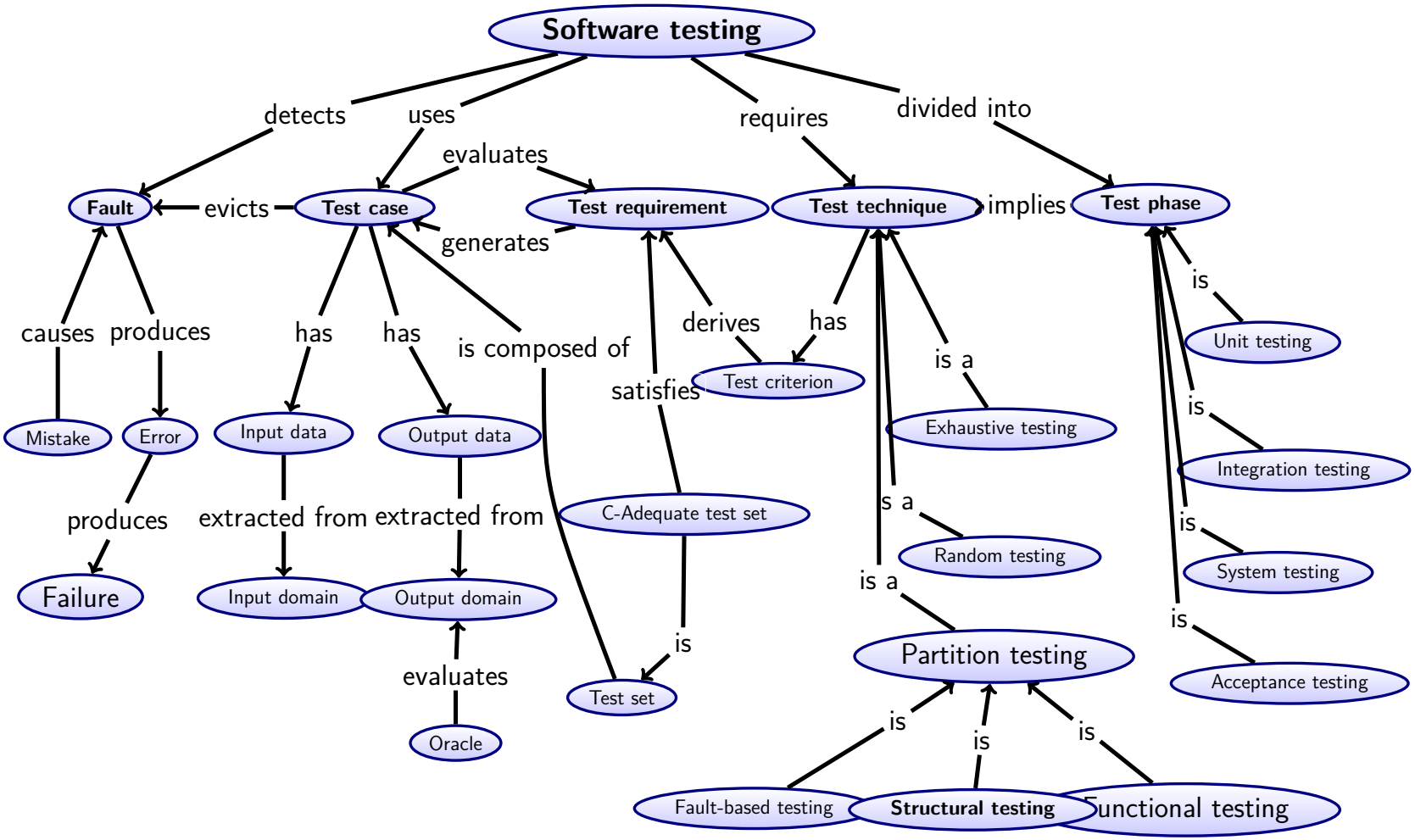
[2]**Institute of Mathematical Sciences and Computing**
University of São Paulo (USP)
São Carlos, SP, Brazil

2017

Software Testing



Software testing



Defect

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Defective products

- No product is defect-free.
  - And that applies to any kind of product, be it software or hardware.

Example: Product failures

Defective software

- No software module is defect-free when implemented and only 40% of software modules are defect-free when released [1].

Example: Software failures

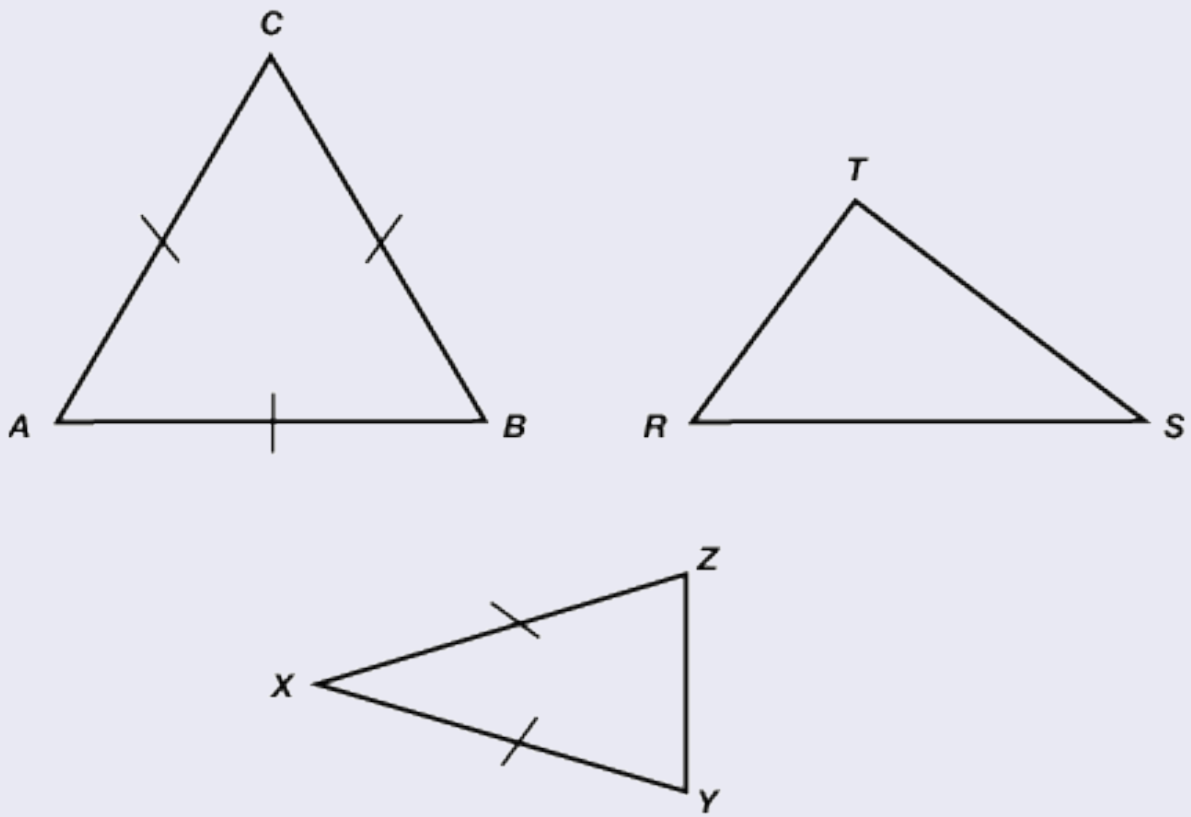
Defect

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Every software has defects

- Even small, simple, trivial products are error-prone.

Example: Triangle



Defect

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Defect avoidance and detection

- Nevertheless, products failures could and must be avoided and corrected, as they negatively impacts software quality.

Why we should care about it?

- Rising demand for higher software quality.
- Software defect reduction improves software quality.

Software quality and software testing

- The main goal of software testing is to find defects.
- Systematic software testing, carried out using proper techniques, criteria, and tools, improves software reliability (and quality).

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Motivation

- Software contains fault.
- Testing the software can reveal several faults.

Software testing as a discipline

- *Ad hoc* testing is insufficient and ineffective at fault detection.
  - It is hard to think of enough test cases even for simple software (such as Triangle).
- Software testing must be taken as a discipline.
  - Test cases must be designed to assess the quality of software artifacts (mainly the requirements specification and the source code) regarding given criteria.
  - Techniques must be developed to detect as much faults as possible in the software.

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

Software testing is the process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component [**?**].

Software testing is. . .

- a verification and validation activity,
- important for maintenance, reliability assessment,software process improvement, debugging,
- a dynamic activity,
  - It requires the execution of the program (static analysis is not enough),
- expensive [2]

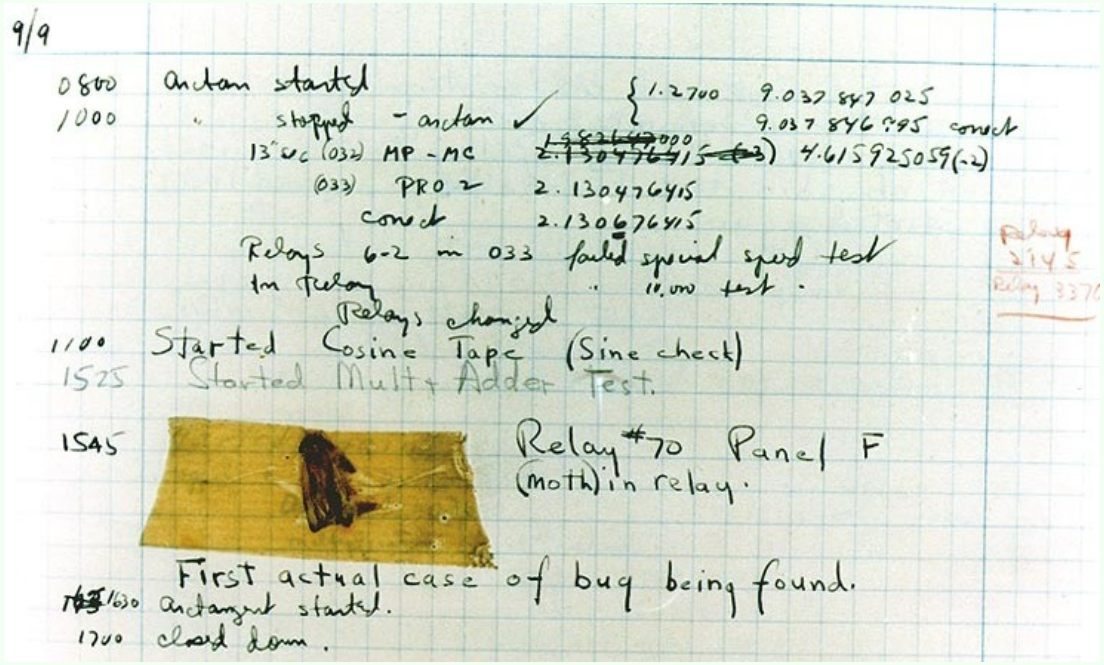
Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Goal

- The goal of software testing is to detect faults in the product under testing.

First error ever detected (by Grace Hopper). It is an actual bug (that is why we call errors of bugs till today).



Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Contradiction

- Thus, software testing aims at virtually destroying the software?
  - After all, the programmer spent hours implementing it and test activities will find errors in the work just done by him.

Rationale

- Humans are highly goal-oriented (and proper goals has an important psychological effect [3, p. 6]).
- If the goal is to demonstrate that a program has no error, then the tester will subconsciously be steered toward this goal.
  - There will be a tendency to select test data that have a low probability of causing the program to fail.
- If the goal is to demonstrate that a program has error, the test data will have a higher probability of finding error.

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Rationale

The process of demonstrating that errors are not present is impossible to achieve for virtually all programs (even trivial ones).

Undecidable problems

- The assessment of the correctness of a program is an undecidable problem.
  - Undecidable = non-computable.
- Errors can be masked by other errors due to coincident correction.
  - And assessment of coincident correct is also an undecidable problem.
- Several other undecidable problems plague software testing: software equivalence, executability.

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

How are faults detected?

- Faults are detected by running the software against a set of test cases.

Test case execution

1. Define the input data to be fed to the software under testing.
2. Run the software.
3. Check if the result the software produced was the expected one (for the given input data).
   - If the result is correct, keep defining different input data.
   - If the result is incorrect, a fault was found! Success!

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Exhaustive testing

If the software is executed against all possible input data, any fault will be found!

Feasibility and computability

Unfortunately, it is often not possible to run exhaustive testing due to some computational problems:

- the input domain may be so big that it is impossible to run the software against it in a reasonable time,
- computability (undecidable problems).

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Domain partitioning

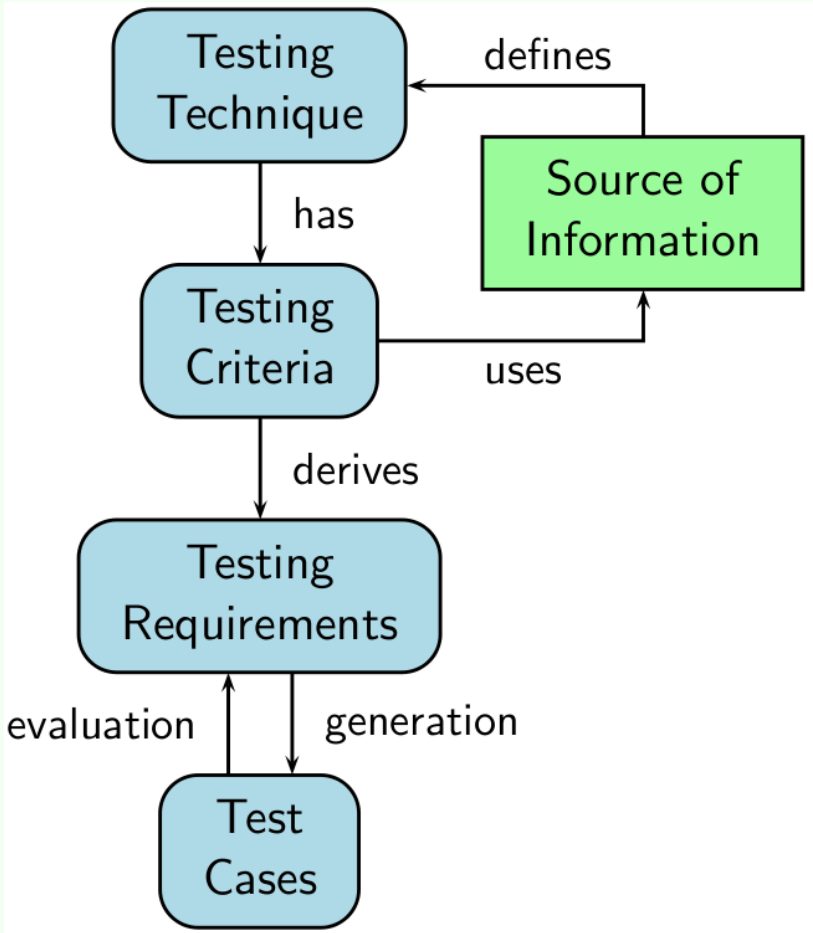
Nonetheless, it is possible to partition the input domain and, instead of using all elements within each partition, just a few ones are selected (and accepted as a good representation of every element in the set).

Techniques and criteria

- Test criterion defines rules regarding how a given input domain is partitioned.
- Such rules, when applied to the software under testing, generates test requirements.
  - Test requirement is a combination of elements of the program under testing that must be satisfied by the execution of a test case.
- Test technique defines the rationale and source of information that guides the development of test criteria.

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process



Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

What is a fault?

A fault is an incorrect data definition, step, or process in a software.

How a fault is created?

- A fault is inserted by a mistake.

Example: Incorrect statement

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

What is a mistake?

A mistake is a human action that produces an incorrect result.

How a mistake takes place?

- Lack of attention when implementing the software?
- Omitted information in the software requirement specification?
- Compiler fault (which by itself was caused by a mistake)?

Example: Incorrect statement

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

What is an error?

An error is the difference between a computed, observed or measured value or condition and the true, theoretically correct or specified value or condition.

How an error occurs?

- An error is produced by the execution of a fault.
  - Thus, if a fault is never executed, it never produces an error.

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

What is a failure?

A failure is the inability of a component or a system to fulfill its required functions within specified performance requirements.

How a failure occurs?

- A fault and its associated errors may cause one or more failures.

Example: Incorrect statement

Software testing

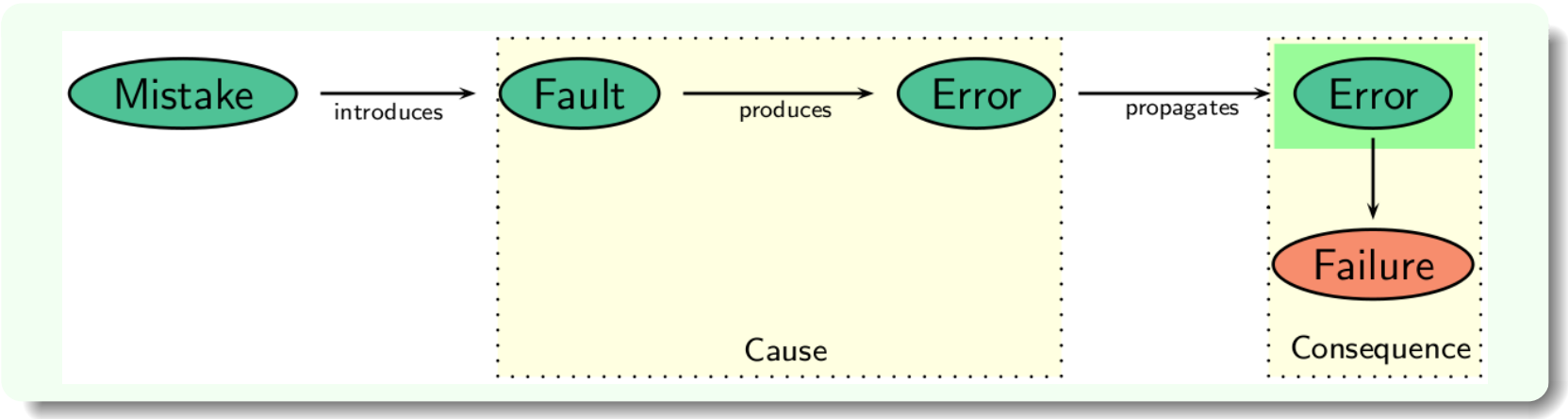
Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Summary

1. A programmer makes a **mistake**.
   - One single $<$ is replaced by a $>$ in a single statement.
2. Due to the mistake, the statement is **faulty** (it does not implement the behavior described in the software requirements specification).
3. The software is compiled and run. The given statement is run. As it belongs to a function in charge of creating the checksum of the data being processed, the output is produces is incorrect (**error**).
4. The user compares the checksum produces by the software with the expected one. No matter what, the checksum always **fails**.

Defect taxonomy

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process



Example: Physician analogy for defect taxonomy

Example: Defect taxonomy example (numZero)

Test case

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Simplified definition

A test case is a pair consisting of test data (a set of values, one for each input variable) to be input to the program and the expected output.

A better definition

A test case is usually defined as a tuple $(d, S(d))$, where:

- $d \in D$ (and D is the input domain), and
- $S(d)$ represents the expected output for the input d according to specification S.

Example: Test cases for a sort method
Example: Test cases for the numZero method

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Successful test cases

- At first, a well-constructed and executed test case is successful when it find errors [3, p. 7]
- It is also successful when it eventually establishes that there are no more errors to be found (as when applying a test criteria and satisfying all the test requirements).

Unsuccessful test cases

- A unsuccessful test case is one that causes a program to product the correct result without finding any error.

Analogy for test cases success and failure

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Execution order

- There are two styles of test case design regarding order of test execution:
  - cascading test cases, and
  - independent test cases.

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

Cascading test cases are test cases that build on each other.

Advantages and disadvantages

- The advantage of cascading test cases is that each test case is typically small and simple.
- The disadvantage of cascading test cases is that if one test fails, the subsequent tests may be invalid.

Example: Cascading test cases

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

Independent test cases are entirely self contained.

- Independent test cases neither build on each other nor require that other tests have been successfully executed.

Advantages and disadvantages

- The advantage of independent test cases is that any number of tests can be executed in any order.
- The disadvantage of independent test cases is that each test tends to be larger and more complex and thus more difficult to design, create, and maintain.

Oracle

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Is it correct?

- Given a set of input conditions and the observable results of the computation, who decides whether the results are correct?
- Somebody or something must check whether the software, for a given test case, has operated correctly.

Oracle

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

An oracle is any software, process, or data that provides the test designer with the expected result of a test case.

- An oracle decides whether output values are correct against what is specified.
  - An oracle is required to determine whether a fault was revealed.
- Some examples of oracle: human guess (kiddie oracle), regression test suite, validated data, purchased test suite, existing software.

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

A kiddie oracle is obtained from running the software and seeing the output. If it looks about right, it must be right.

Why should I use a kiddie oracle?

- Actually, you should not use it, as it is error prone.
- Nonetheless, it is better than nothing.
  - And, if the expected behavior of the application is not documented, it is up to the user to detect the correct result anyway.

Example: Human guess (kiddie) oracle

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

A regression test suite oracle is obtained from running the test case and comparing the output to the results of the same test cases run against a previous version of the software.

Why should I use a regression suite oracle?

- Regression test ensures that the modified system functions as per its specification.
- It ensures that old errors will not appear again (or that, at least, they will be early detected).

Example: Regression test suite oracle for Mozilla Firefox

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

A purchased test suite oracle consists in running the software against a standardized test suite that has been previously created and validated.

Why should I use a purchased oracle?

- Usually it is required that a software passes a test suite oracle in order assess its compliance to a particular technology or standard.
- Sometimes its simply easier to buy a test suite instead of developing your own.

Example: Java TCK

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

What I should test first?

- What should I target when testing a software?
- The definition of a single target for each test activity is key for systematic software testing.
  - Focus effort and restrict available techniques (thus lowering costs and, probably, increasing efficacy).
- A common sense approach would be to begin testing the smallest module possible, and then proceed to test the interactions between these modules and, finally, the system as a whole.

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

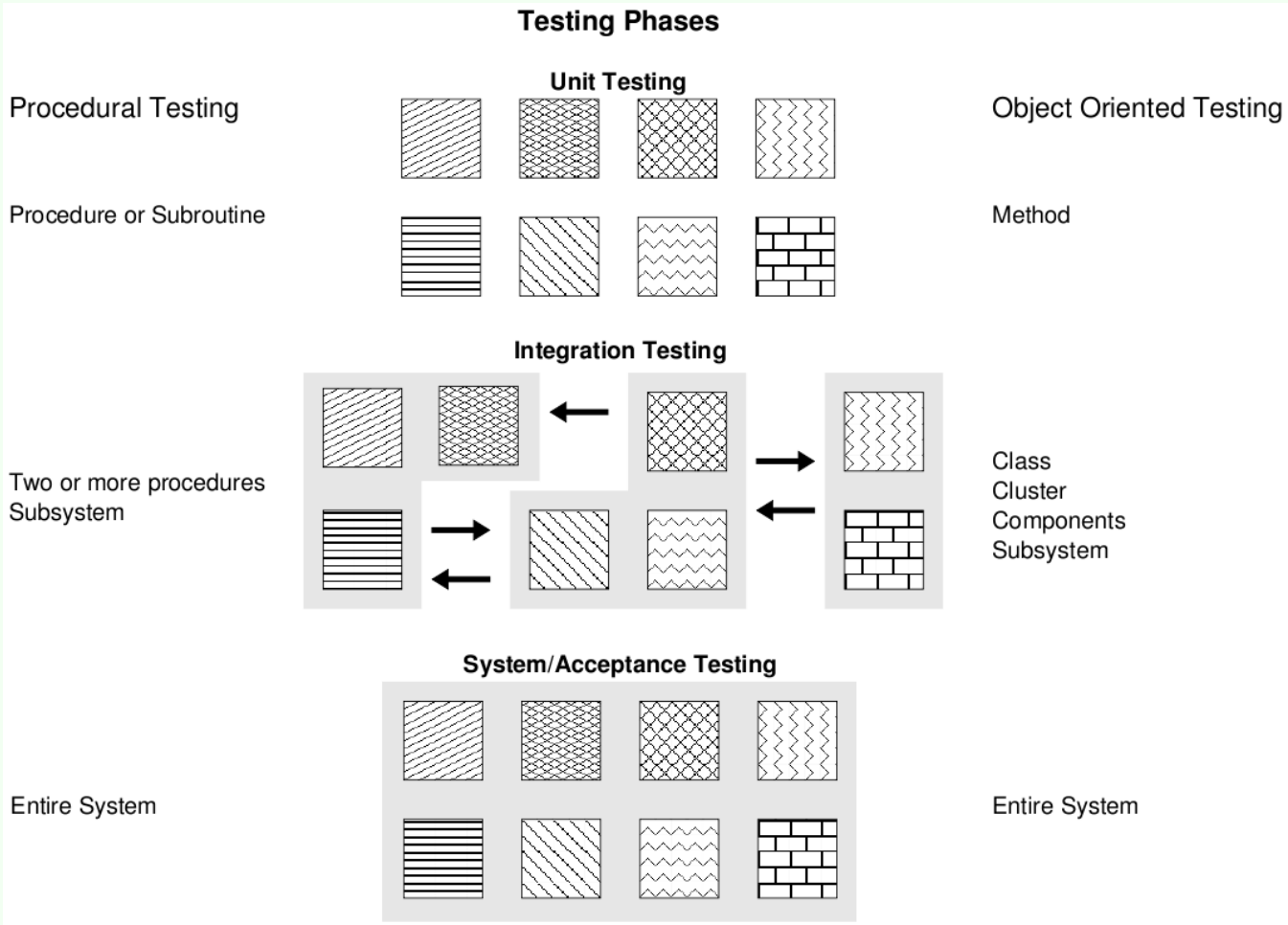
Test phase is a categorization of test activities which is directly related to the software life cycle the activities take place in.

Definitions

- Test phases allow the tester to concentrate on various aspects of the software and use different test criteria at each one.
- Test activities are organized in test phases such that testing starts with the smallest executable unit until reaching the software as a whole:
  - Unit testing.
  - Integration testing.
  - System testing.
  - Acceptance testing.

Test phases

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

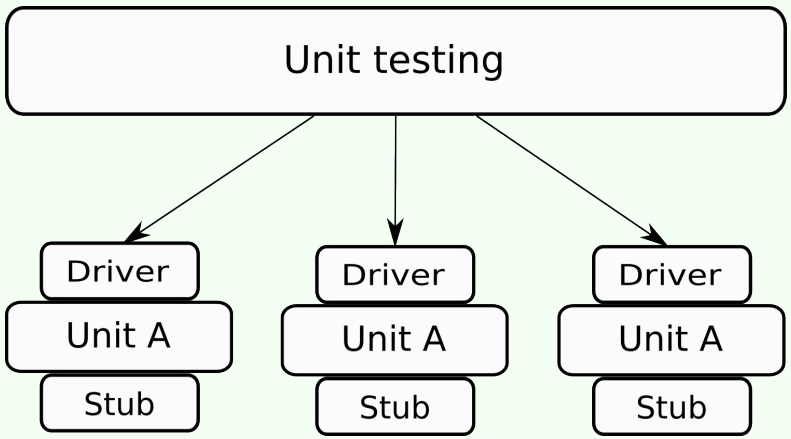


Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

Unit testing verifies the functioning in isolation of software pieces which are separately testable.



Example: Pentium FDIV bug

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

What is an unit?

- In procedural testing, the unit is the procedure or subroutine.
- In object oriented testing, the unit is the method or the class.
  - For the time being, we will consider that the unit is the method!

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

How to test an unit?

- Although the definition calls for units that are separately testable, this is not always possible.
  - Mosts units requires data from other units.
- Coupling between units is a threat for unit testing. Thus, it is desirable to replace some units by others more simple and predictable (just for software testing sake).

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

A stub is a unit that replaces another unit used by the unit under test.

Why should we use a stub?

- Stubs simulate the behavior of another unit not yet implemented but called by the unit under test.
- Usually, a stub simulates the expected behavior of the used unit with minimum computation effort or data manipulation.

Example: Stub

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

How to test a unit?

- Even tough stubs can be used to replace complex units for a given unit under testing, something must setup and wire the stubs into the unit.

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

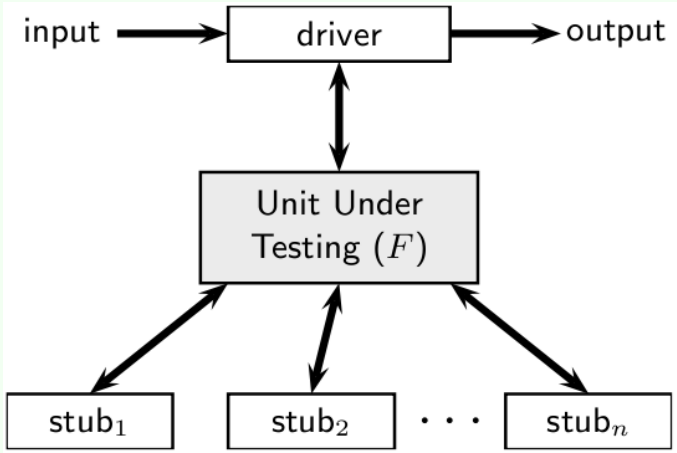
Driver is the software responsible for coordinating the testing of a unit.

What a driver does?

- Drivers are used to test a unit which requires input data provided by another unit:
  1. it gathers the data provided by the tester,
  2. it passes them to the unit under test in the form of arguments,
  3. it collects the results produced by the unit, and
  4. it shows them to the tester.

Test phases
Unit testing - Driver and stubs

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

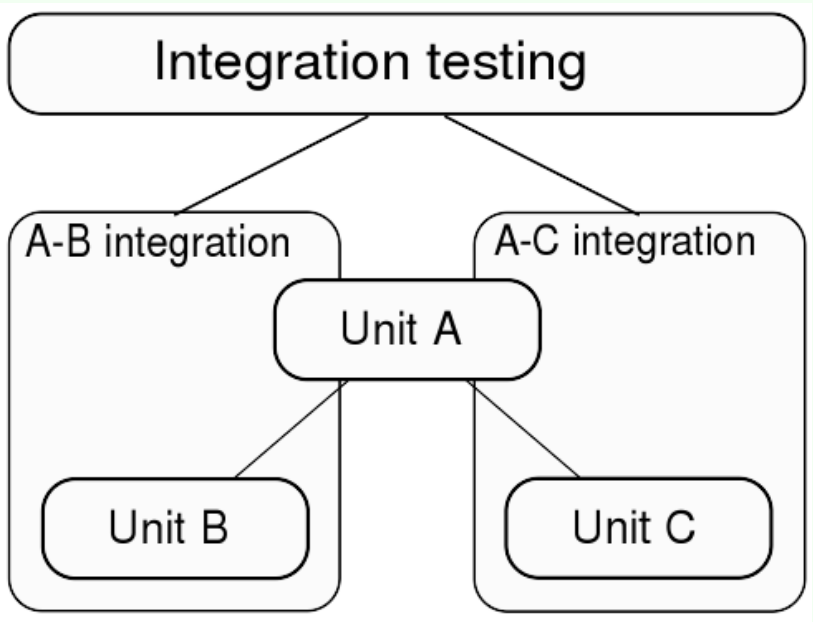


Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

Integration testing verifies whether the units tested individually communicate accordingly when integrated.



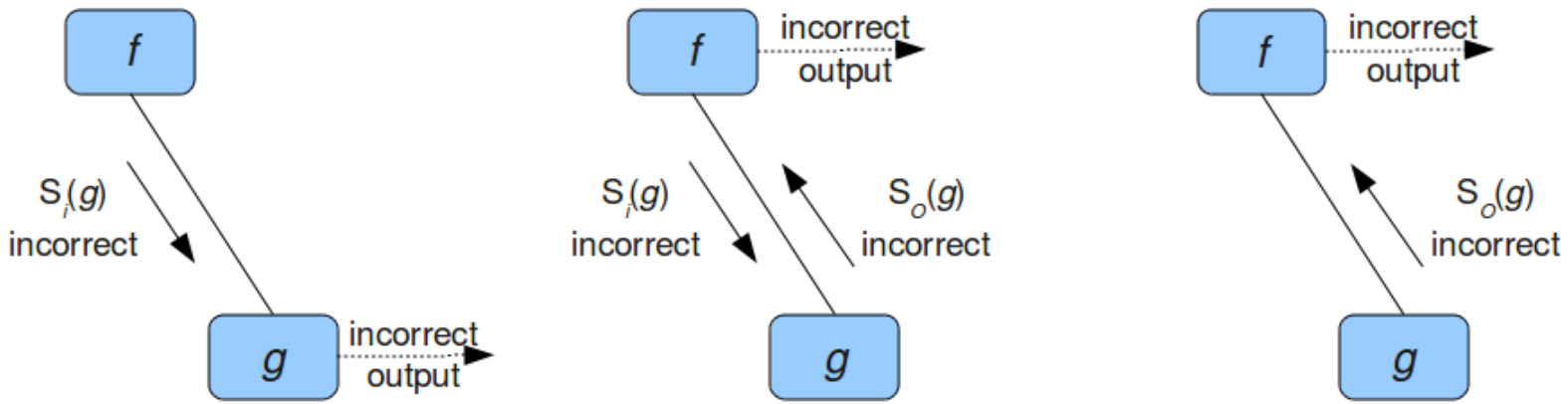
Example: Mars climate orbiter

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Why integration testing is important?

- Integration testing must be performed because:
  - Data can be lost at the unit's interface.
  - Global variables can suffer undesirable interferences.



Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

System testing ensures that the software and the other elements that are part of the system (hardware and database, for instance) are adequately combined and behave as expected.

Types of system testing

- Typically system testing includes many types of testing:
  - functionality,
  - usability,
  - security,
  - localization,
  - reliability,
  - availability,
  - ...

Example: Ghost train

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

Acceptance testing refers to the tester, in general performed by the user himself, who verifies whether the product satisfies his expectation.

Alpha/Beta testing

- Informally, it can be defined alpha and beta testing.
  - Alpha testing: software is installed and used internally (in the company that developed).
  - Beta testing: software is installed and tested by external users.

Why is acceptance testing important?

- Acceptance testing, when completed successfully, will result in the customer accepting the software.

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Test techniques and criteria

- Test techniques provides a grounded theory on how we should test our software:
  - What are the sources of test requirements?
  - What features of the sources should be exploited?
- However, in order to successfully test a software, it must be throughly specified how the techniques should be applied.
  - Any by successfully we mean: to find errors.

How to increase efficacy

- Test requirements should be established so they explore not only what the software should do, but also what it should **not** do.
- A measure should also be provided, so that a stop criteria can be defined for the test activity.

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

A test criterion systematizes the way test requirements are generated from the source of information (specification, source code, historical fault database, etc.).

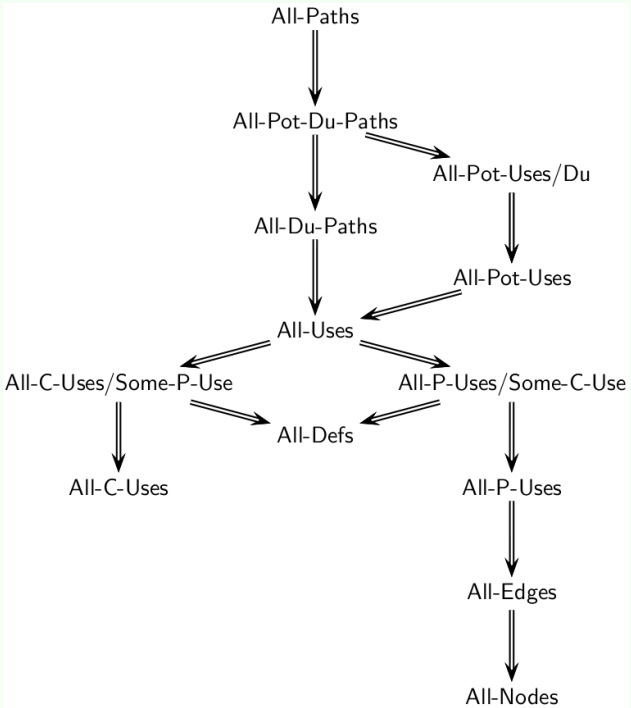
- A test criterion provides a systematic way to select test cases.
  - A test criterion divides the input domain.
- When no faults are found, test criterion provides an indication of how test cases should be selected in order to establish a high level of confidence of product correction.

Example: Test criterion example

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Test criterion attributes

- A test criterion can be compared based on cost, efficacy and strength.



Test criterion

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Test coverage

- A test criteria can be used either to evaluate the program under testing or to evaluation the thoroughness of the test set.

Test selection criterion

The test criterion is used to select test cases to evaluate the program under testing.

Test evaluation criterion

The test criterion is used to evaluate the test sets.

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

A test requirement is a specific element of a software artifact that a test case must satisfy or cover.

How a test requirement is created?

- Test requirements are derived from the program under testing using a specific test criterion.

What are test requirements for?

- A test requirement can:
  - evaluate a test set, and
  - generate a test set.

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

A test set is a set of test cases.

Test sets and test requirements

- A test set can be improved by adding test cases that exercise uncovered requirements.
  - The best test set is the smallest one that indicates the largest set of faults.

C-adequate test sets

- When a test set T satisfies all the test requirements derived from a program using a given criterion C, T is said $C - adequate$.

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

Test techniques are types of testing defined according to the source of information used to carried out the testing activity.

Test techniques and test criteria

- Each test technique has a set of associated test criteria.

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Software test techniques

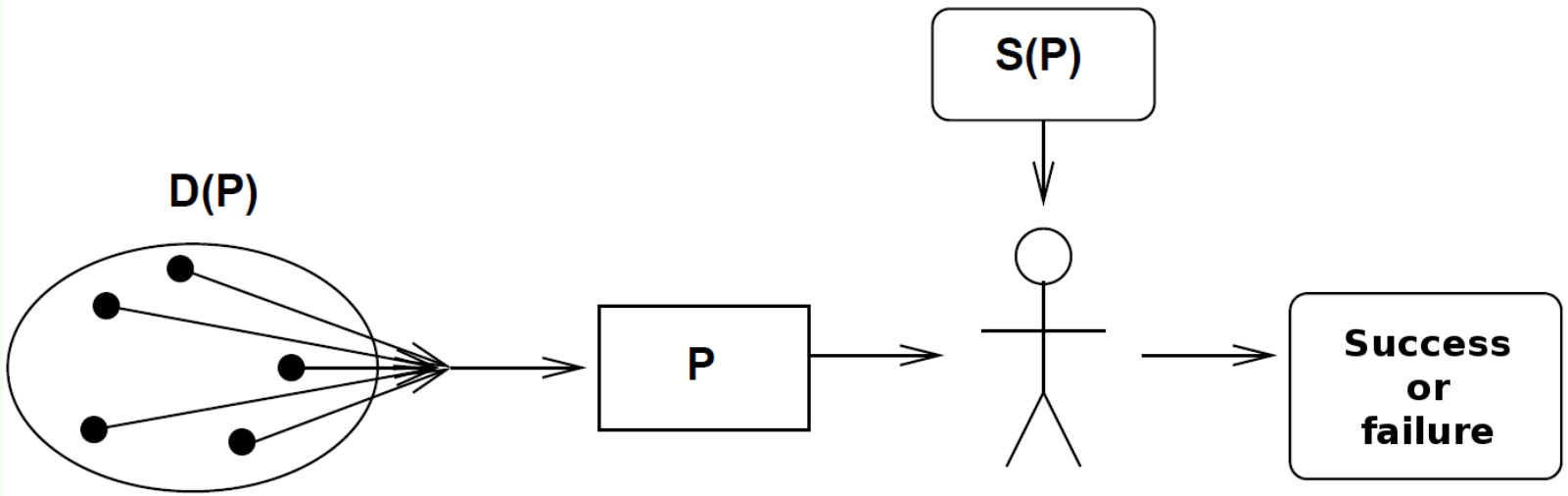
- Exhaustive testing
- Random testing
- Partition testing
  - Fault-based testing
  - Functional testing
  - Structural testing

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

An exhaustive test executes the software with all possible value from its input domains.



Example: Exhaustive testing of the blech function

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Exhaustive testing limitations

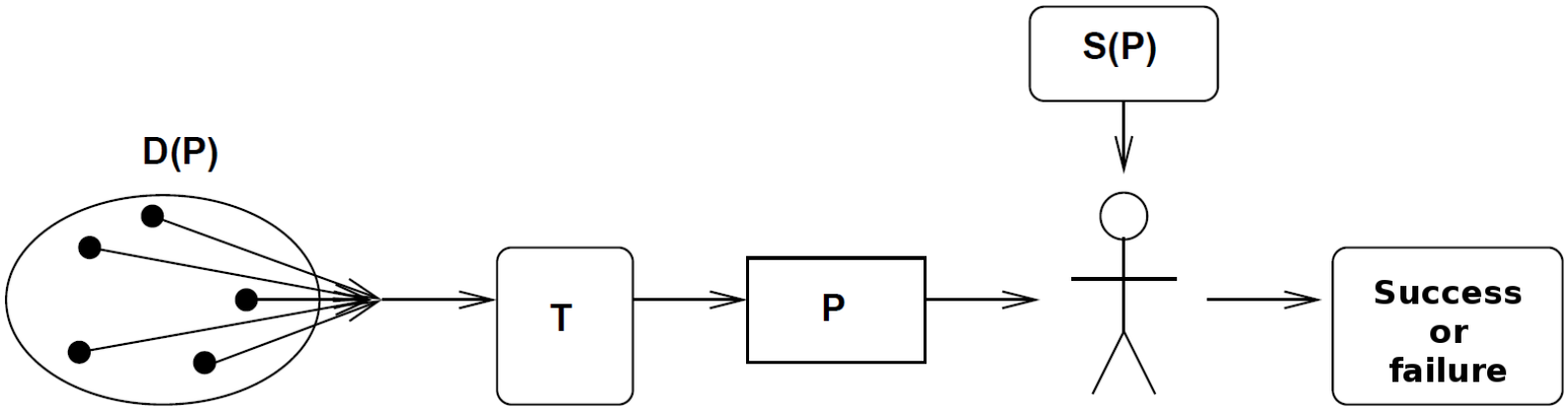
- Can be prohibitive due to time and cost constraints for finite but large input domain.
- Impossible if the input domain is infinite.
- Infeasible in general.

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

Random testing uses a systematic method to generate test cases: it requires modeling the input space and then sampling data from the input space randomly.



Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Reliability

- Using random testing, statistical measures of reliability can be achieved according to an operational profile.
- For every (input data of a) test case, it is assigned a probability distribution according to their occurrence in actual operation.

Effectiveness

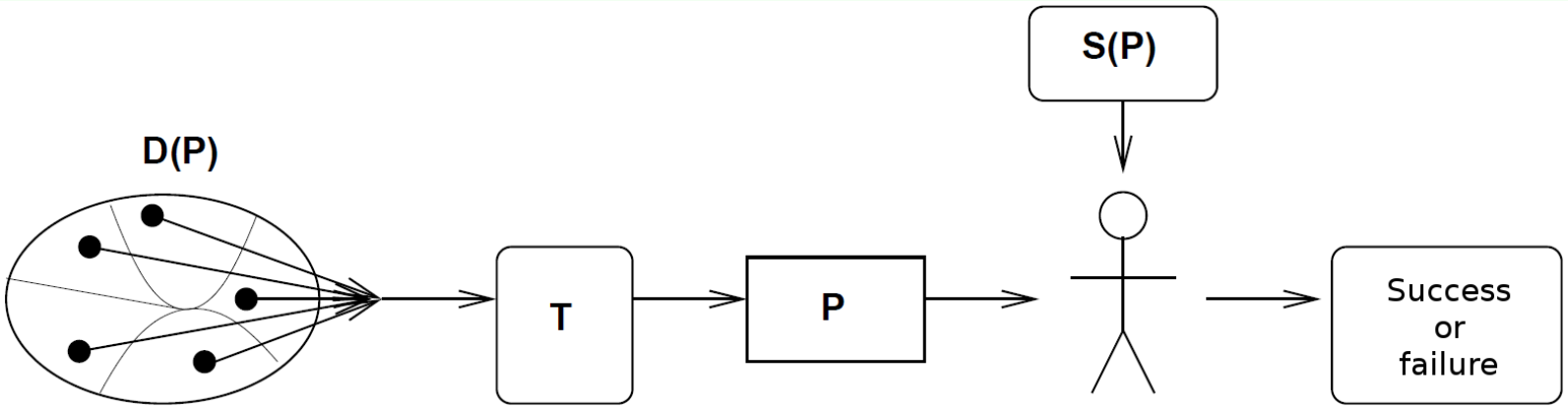
- It depends on the correctly definition of the operational profile.
- If the probability of occurrence of each input data is the same, random testing is regarded as the least effective technique for software testing [3, p. 43].

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

Partition testing is meant any testing scheme which forces execution of at least one test case from each subset of a partition of the input domain



Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

Functional testing is a technique based solely on the requirements and specifications.

- Functional testing is also known as black box testing.
- Functional testing obtains test requirements from the software specification.
  - Functional testing requires no knowledge of the internal paths, structure, or implementation of the software under test.

Example

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Functional test criteria

- Equivalence partition.
- Boundary-value analysis.
- Cause-effect graph.
- ...

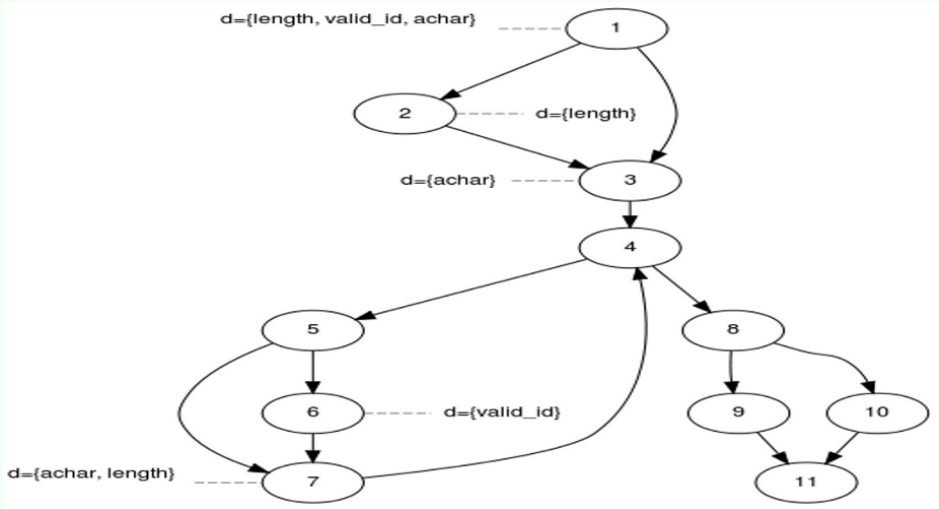
Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

Structural testing is a technique based on the internal paths, structure, and implementation of the software under test.

- Structural testing is also known as white box testing.
- Structural testing obtains test requirements from implementation features.



Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Control-flow based criteria

- Criteria based on the flow of control within a program:
  - Statement coverage.
  - Decision coverage.
  - Condition coverage.
  - ...

Data-flow based criteria

- Criteria based on the usage of data (variable creation, definition, and use):
  - All-uses.
  - All-potential-uses
  - ...

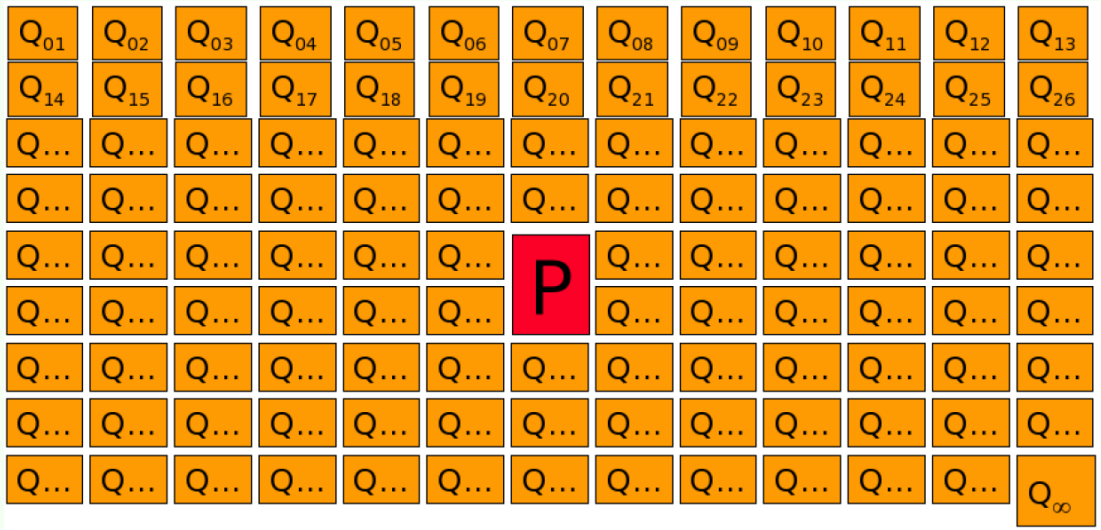
Example

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Definition

Fault-based testing is a technique in which testing is based on historical information about common faults detected during the software development life cycle.



Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Fault-based test criteria

- Error seeding.
- Mutation:
  - Mutation analysis.
  - Interface mutation.
  - ...

Example

Software testing

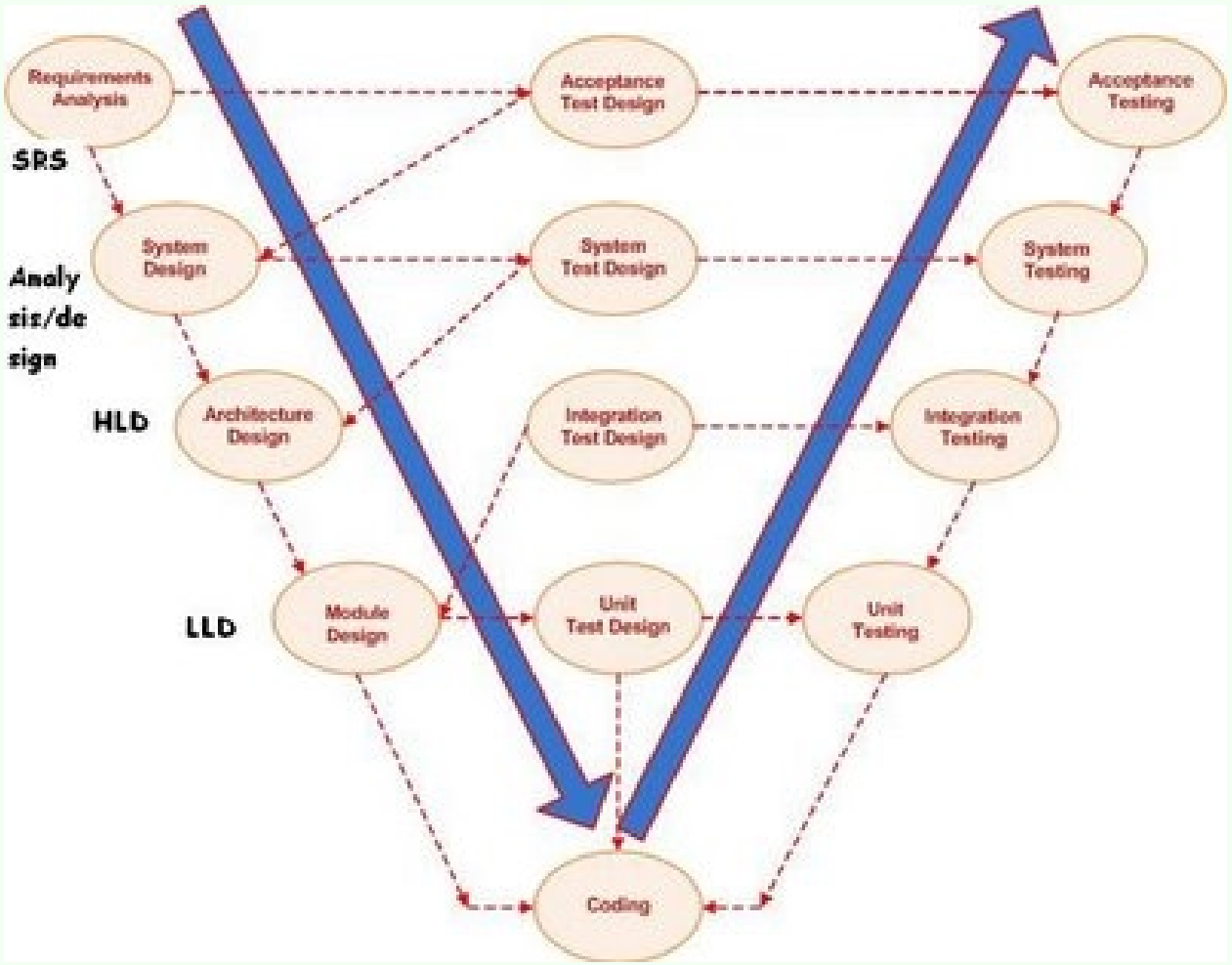
Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process

Software testing process

- When conducted in a systematic and clear-sighted way, software testing helps to:
  - Increase confidence that the product behaves according to its specification.
  - Highlight minimal characteristics of product quality.

Software testing

Software testing
Defect
Software testing
Defect taxonomy
Test case
Oracle
Test phase
Test criterion
Test requirement
Test technique
Software testing process



References


SHULL, F. et al. What we have learned about fighting defects. In: *8th International Symposium on Software Metrics*. [S.l.]: IEEE Computer Society, 2002. p. 249–258. ISBN 0-7695-1043-4.


HARROLD, M. J. Testing: A roadmap. In: *International Conference on Software Engineering*. New York, NY, USA: ACM, 2000. (ICSE), p. 61–72. ISBN 1-58113-253-0.


MYERS, G. J. *The Art of Software Testing*. 2. ed. New York, NY,EUA: John Wiley & Sons, 2004. 234 p. Revised and updated by Tom Badgett and Todd M. Thomas, with Corey Sandler. ISBN 0471469122.

Credits

- Reviewers:
  - Fabiano Cutigi Ferrari
  - Otávio Augusto Lazzarini Lemos