Review of software testing

P David Coward

200911385 박기남
200911425 조서경
Presenter : 200911426 조성완
200911427 조아라

Index

I. What is software testing?

II. Dimensions of testing strategies.

III. Categories of testing techniques.

IV. Summary of software testing

0. Introduction

- Software testing arises from distrust in software developed.
- Testing is followed by static and dynamic analysis, and functional and structural testing strategies.
- Testing techniques are divided from strategies.

Ⅰ. What is Software testing?

1. Definition of software testing
2. Motive for using software testing
3. Verification & Velidation
4. Functional & Nonfunctional testing
5. Aims of software testing
6. Examples of software testing

1. Definition of software testing

- There is no agreed definition of testing.

- Checking software

whether

- Software meets with user requirements.
- Output values are based on specification.

2. Motive for using software testing

- For next reasons, People feel the necessity of software testing.

- Despite advances in formal method of specification and improved software creation tools, there is no guarantees that the software produced meets its functional requirement.

- Software's specification may not be correct.

3. Verification & Velidation

- Verification

  - Is software being developed well?

  - Correctness of the software development cycle.

  - Check the process of software development.

3. Verification & Velidation

- Velidation

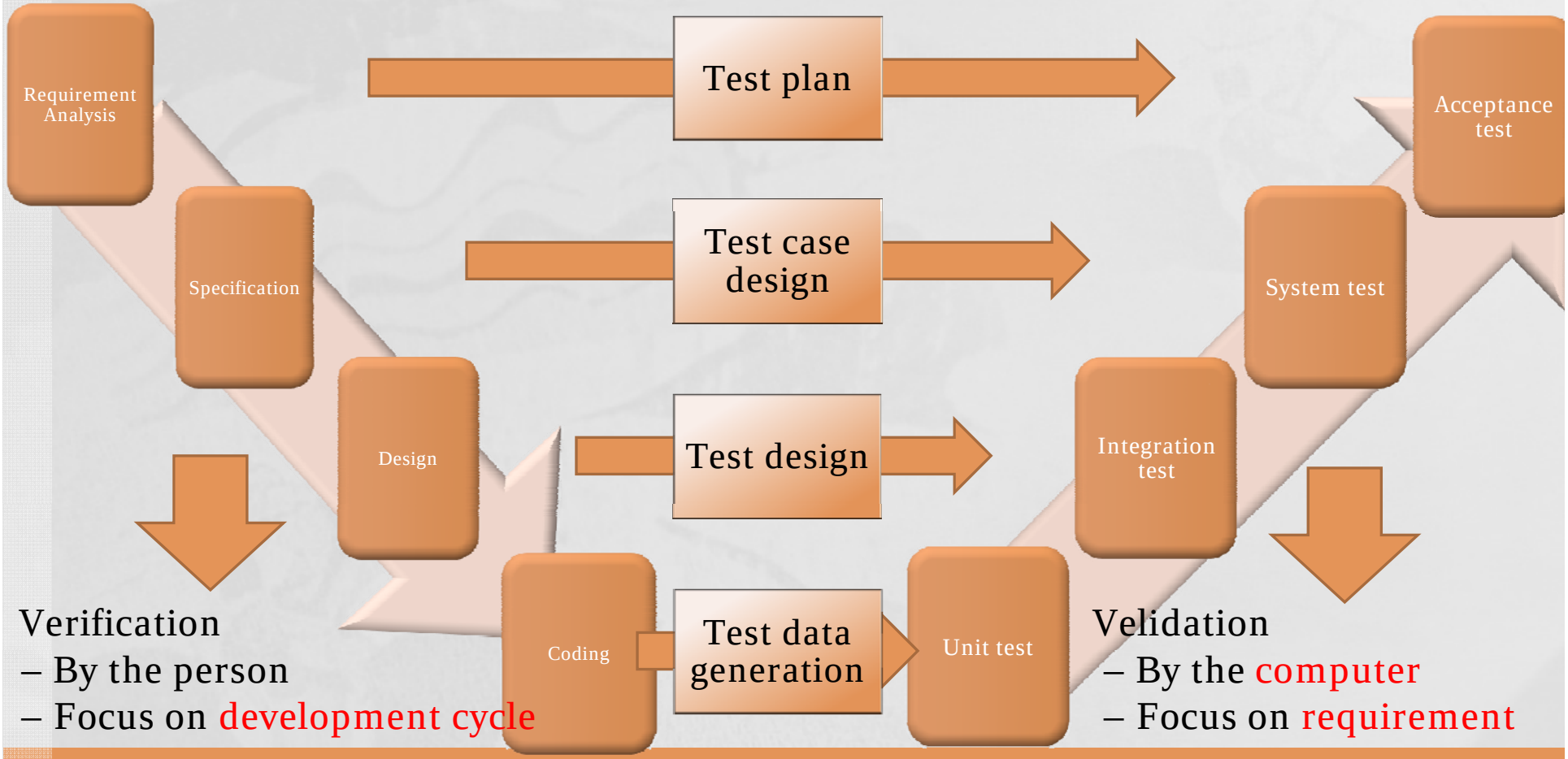
  - Is software developed well?

  - Checking the software against the requirements.

  - Check the result or output data of software.

3. Verification & Validation

- Comparison of V & V (V model)



Verification
– By the person
– Focus on development cycle

Velidation
– By the computer
– Focus on requirement

4. Functional & nonfunctional testing

- Functional testing
  - Checking whether output is correct.
  - It is used when testing new program
    or testing modified program
- Regression testing
  - Alter the function of the software that were intended to remain unchanged.
  - But, implement the functions required by the customer will not involve all requirements placed upon a software system

4. Functional & nonfunctional testing

- Nonfunctional testing
  - For omitted requirements

Checking

whether

- satisfies legal obligation
- performs within specified response time
- written in particular house style
- meets documents standards

5. Aims of software testing

There is two camps of aims

- Find faults in the software

  - ***Destructive*** process
  - Because of the probing attention, this will be more likely to uncover faults.

5. Aims of software testing

- Demonstrate that there are no fault in the software.

  - ***Constructive*** process
  - Cause the tester to be gentle
  - Have risks of missing inherent fault

6. Examples of software testing

- NASA
  - NASA established teams of software validators.

- Other large-software-development-organizations has established testing team, too.

- Not only in software organization,
  but also in other kinds of companies.

Ⅱ. Dimensions of testing strategies

1. Functional-structural testing
2. Static-dynamic testing

Dimensions of testing strategies

- The functional-structural dimension

  - Functional testing
  - Structural testing

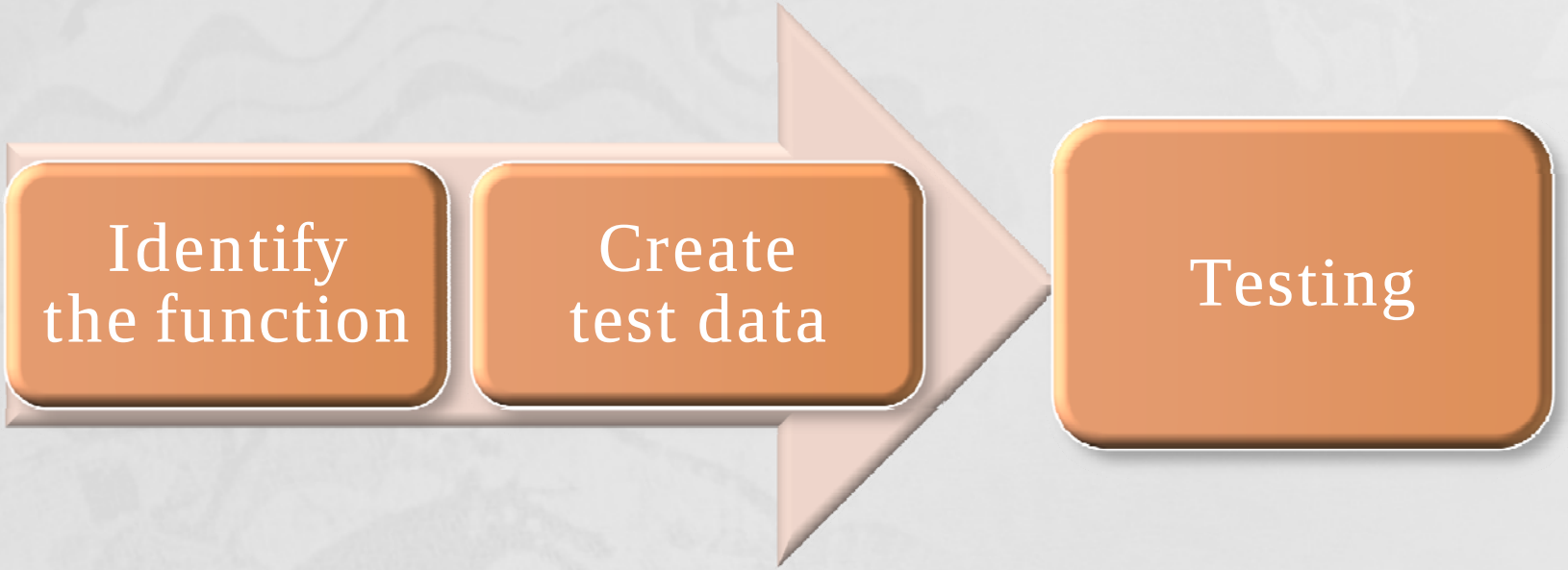
- The static-dynamic dimension

  - Static testing
  - Dynamic testing

1. Functional-structural testing

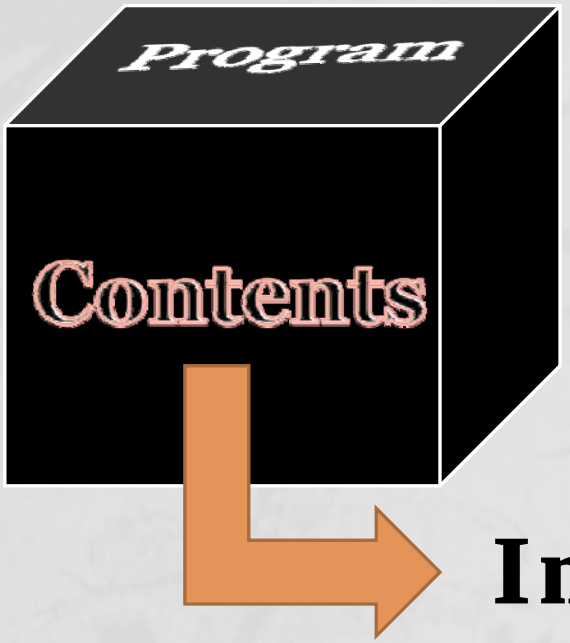
1-1. Functional testing

- Two main steps



1-1. Functional testing

- Black-box testing



- Based on requirement
  (function specification, interfaces)
- Understanding of function
- ***Oracle*** is important
- Simulation software

Invisible!

1. Functional-structural testing

1-2. Structural testing

- Two scenarios

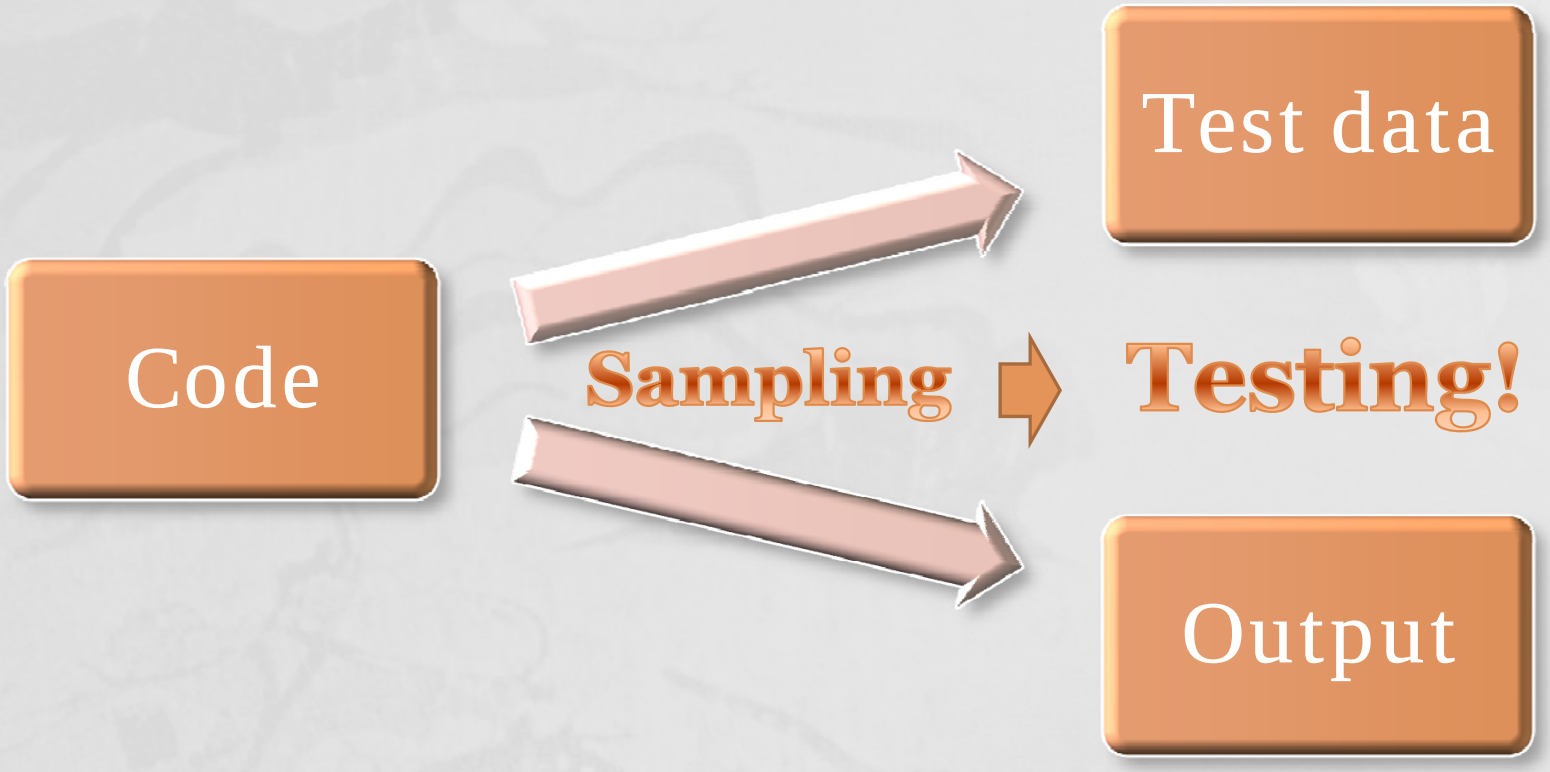
Execute program with ***test data***.

- Most commonly case

Compare with require function

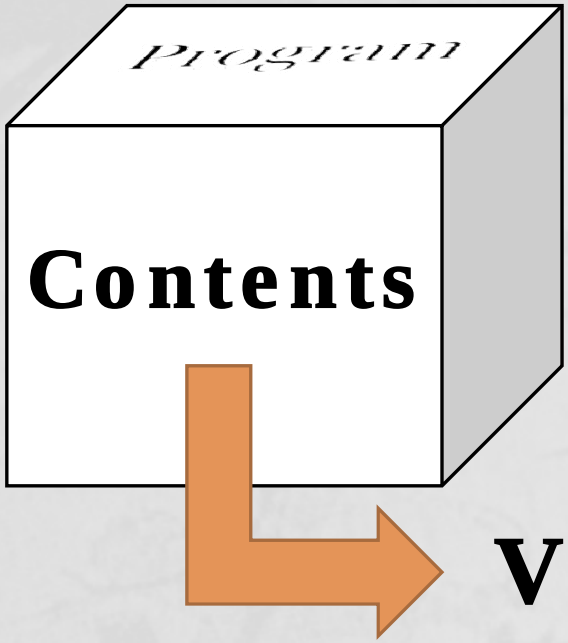
- Less common
- Symbolic execution
- Program providing

1-2. Structural testing



1-2. Structural testing

- White-box testing



- Based on implementation
  (Structure of code)
- Check the detail design

Visible!

1-2. Structural testing

- Trying to discover what is the minimum amount of testing that is required to ensure a degree of reliability.

  - Statement testing
  - Branch testing
  - LCSAJs testing
    (Linear Code Sequence And Jumps)

1-2. Structural testing

- The best test is exhaustive test.
- But, there is two obstacles.
  - The large number of possible path.
  - Exist of infeasible path.

- Island codes disturb metric of testing overage.
  - Island code
    - Procedure that isn't invoked.
    - Caused by error in the invocation of a required procedure.

1. Functional-structural testing

- Functional vs Structural

	Metaphor	Base	Testing tool	Type
Functional	Black-box	Requirement	- Function specification - Interfaces	Indirect
Structural	White-box	Implementation	-Structure of the code	Direct

2. Static-dynamic testing

- Static testing
  - Not involve the execution of software

  - Program providing
  - Symbolic execution
  - Anomaly analysis

2. Static-dynamic testing

- Dynamic testing
  - Require execution of software
  - Use of probes
  - Analysis routines

  - It can be the bridge between
    functional & structural testing.

2. Static-dynamic testing

- Comparison static and dynamic

	Program execution	Testing tool
Static	Needless	- Symbolic values - Input & output datas
Dynamic	Required	- Probes - Routines

Ⅲ. Categories of testing techniques

1. Static-structural
2. Dynamic-functional
3. Dynamic-structural

0. Categories

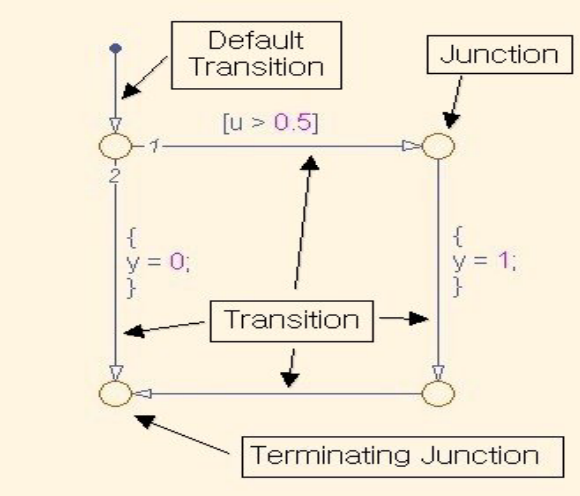
Testing techniques are divided
by the dimensions of testing strategies

	Structural	Functional
Static	- Symbolic execution - Program providing - Anomaly analysis	
Dynamic	- Computation testing - Domain testing - Automatic path-based test - data generation - Mutation analysis	- Random testing - Domain testing - Cause-effect graphing - Adaptive perturbation testing

2. Static-structural

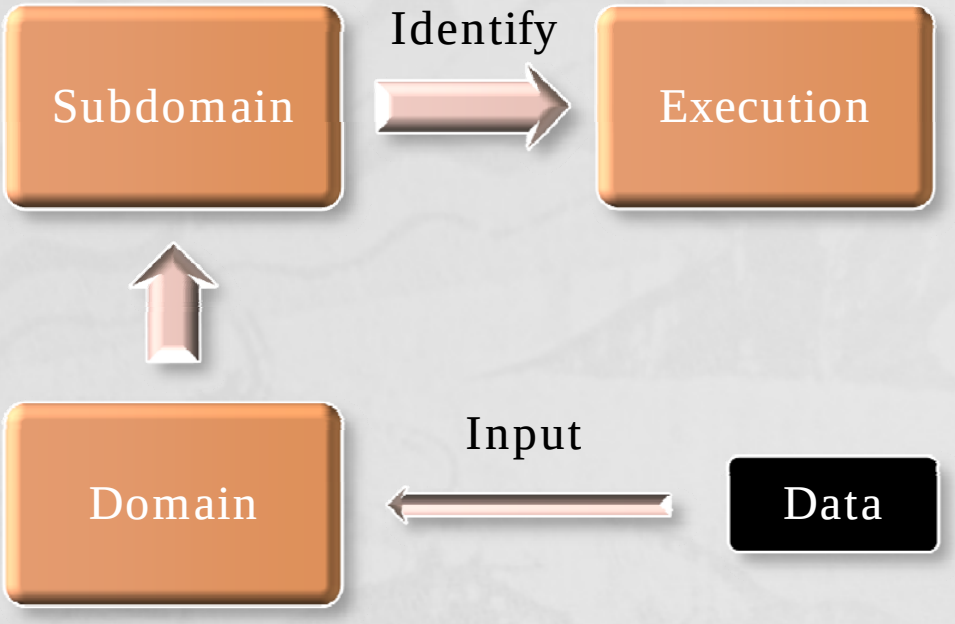
- Testing detailed design (no execution)
- Symbolic execution
  - Actual data values are replaced by symbolic.
  - Difficulty : handling of loops
  - Flow-graph

    ex) if(u > 0.5) { y = 1; }
    else { y = 0; }



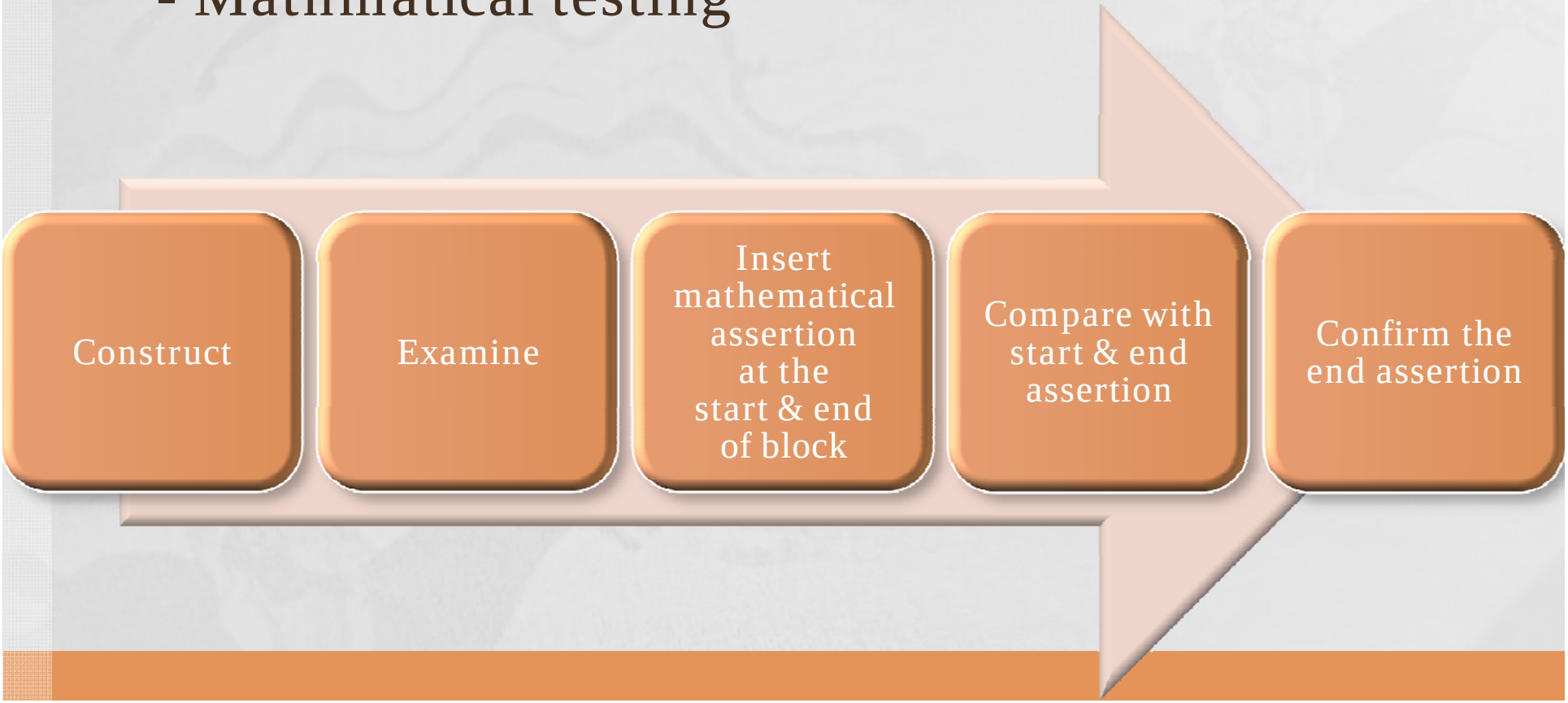
2. Static-structural

- Partition analysis



2. Static-structural

- Program providing
  - Check input and output data
  - Mathmatical testing



2. Static-structural

- Anomaly analysis
  - Checking language syntax
  - Search for anomalies

  ex) 1. Unexecutable code
  2. Array bounds
  3. Failed initializing
  4. Unused variables
  5. Failed in loops

2. Dynamic-functional

- Execute test cases (no detailed design)

- Domain testing
  - Test case is based on
    an informal classification of the requirements.
  - Execute test case and detect fault.

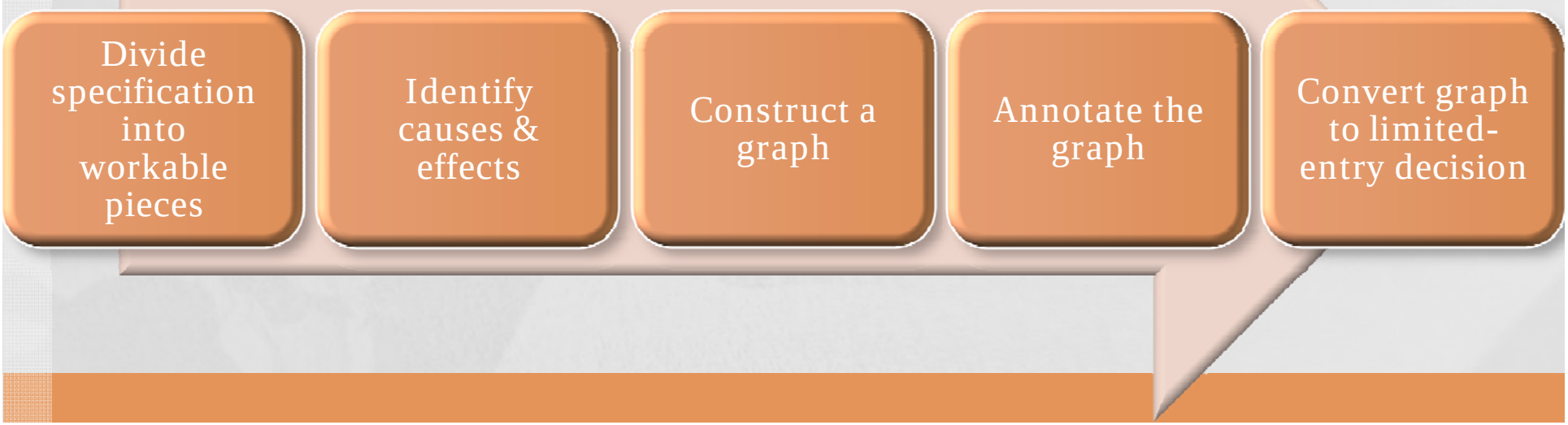
2. Dynamic-functional

- Random testing
  - Test data is produced without reference to the code or the specification.
  - Problem : No guarantee to complete coverage of the program.

    ⇒ Little practical ...
- Adaptive perturbation testing
  - Test data is based on effectiveness.
  - Cornerstone : Using executable assertions
  - Maximize the number of assertion violations.

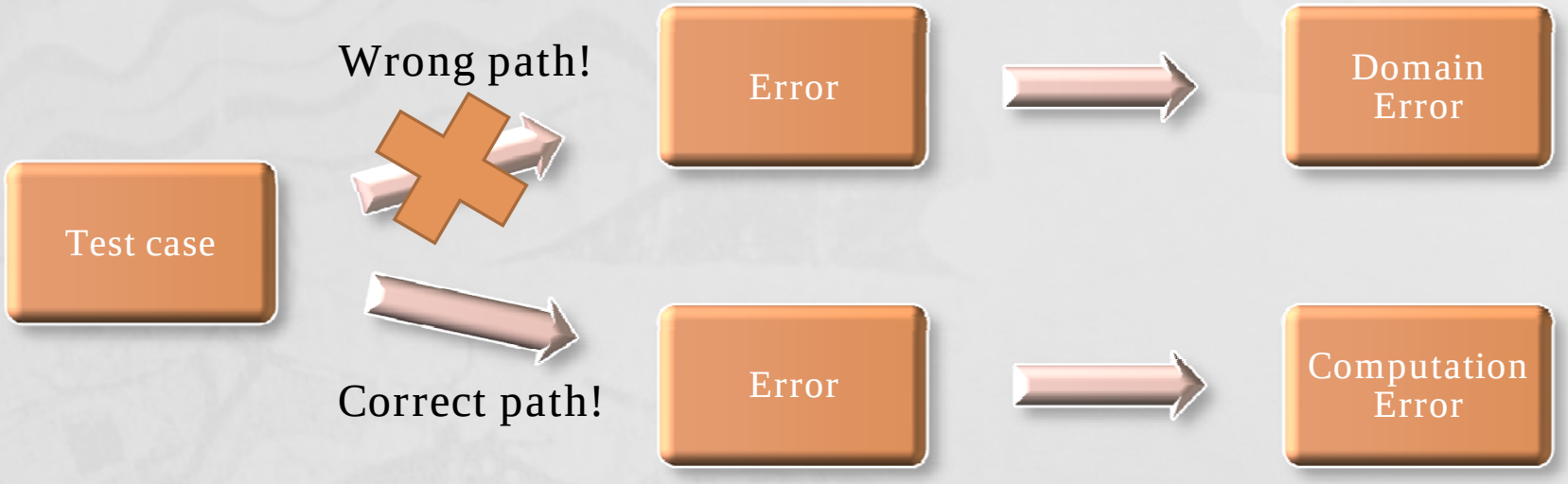
2. Dynamic-functional

- Cause-effect graphing
  - Test data is a combinational input.
  - Combinatorial logic network
  - Use Boolean logical operators.
  - To identify a small number of useful test cases.



3. Dynamic-structural

- Domain and computation testing
  - Using structure and select paths
    which are identify domain.



3. Dynamic-structural

- Automatic test data generation
  - Test data is generated from a syntactic description of the test data expressed in.
  - Repeated use of this method may produce the test data that has confidence.
- Mutation analysis
  - It doesn't create test data nor demonstrate that the program is correct.
  - Check the quality of a set of test data.
  - To create high quality test data.

Ⅳ. Summary

Summary

- Software testing gives us confidence.
- Testing has 2 strategies.
- From the strategies, testing techniques are divided to 4 categories.
- Thorough testing is a necessary for software development.

Thank You

A review of software testing

by team 9