Introduction to testing

Marco Di Natale

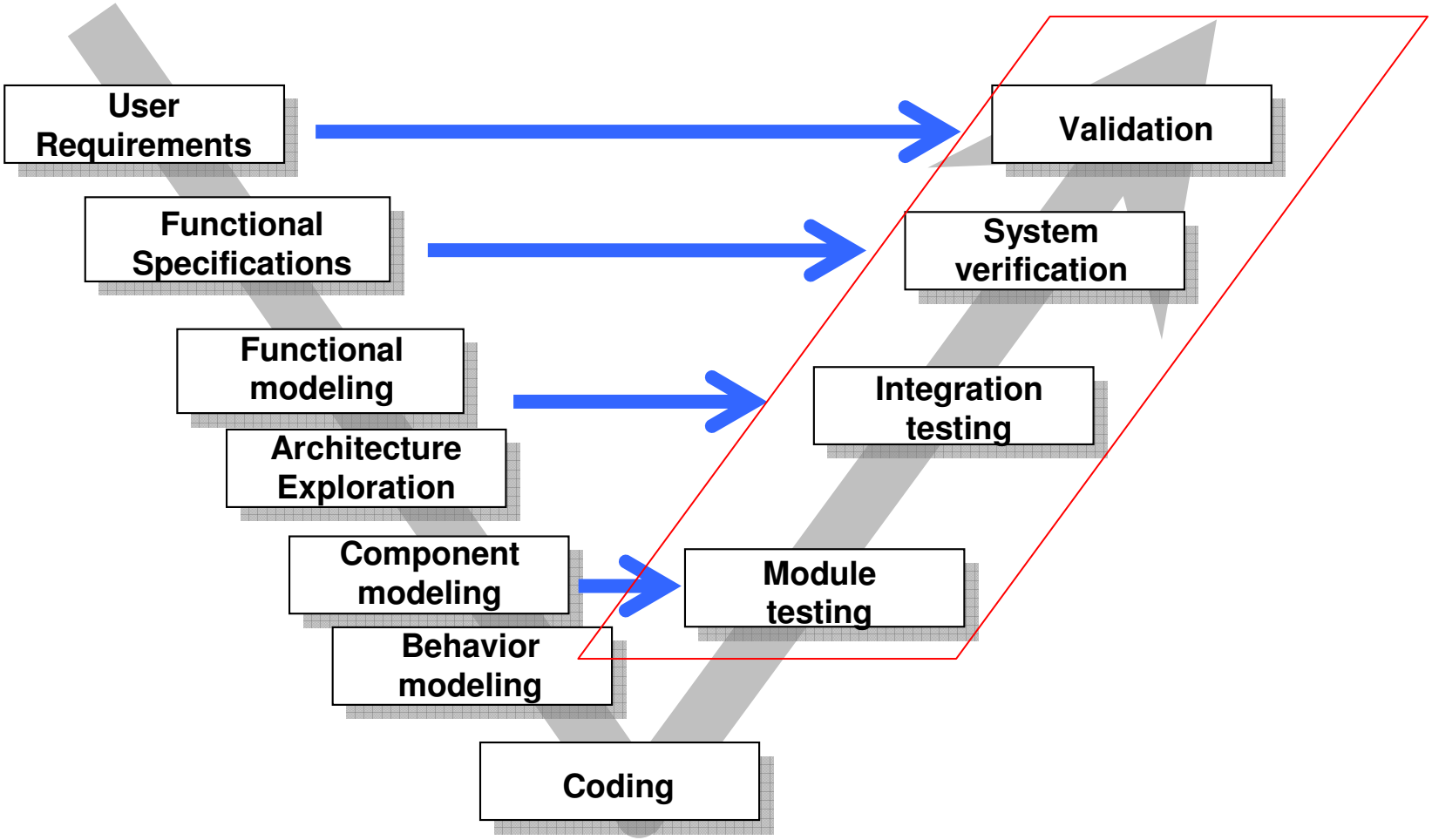
Scuola Superiore S. Anna- Pisa, Italy

Objectives

- Learn when, what and how to test
- Learn about practical testing techniques
- Learn theoretical basis of testing
- Define and use functional (black box) testing
- Understand structural (white box) testing
- Learn about coverage criteria
- Understand conformance testing
- Define methods for Unit testing, use tools
- Learn about the connections between them and how to choose among them

When, what and how to test

Plan early, test often



Functional (black-box) testing

Assumption: software/system is a function from inputs to outputs

Typically used for

- System-level testing
- Unit testing

It is requirements-driven …

- covering aspects of specification
- can be enhanced with information from the function

... and implementation independent:

- tests can be planned early

Not sufficient. Does not detect introduction of unintended functionality

Difficult to perform with decent coverage of the I/O space

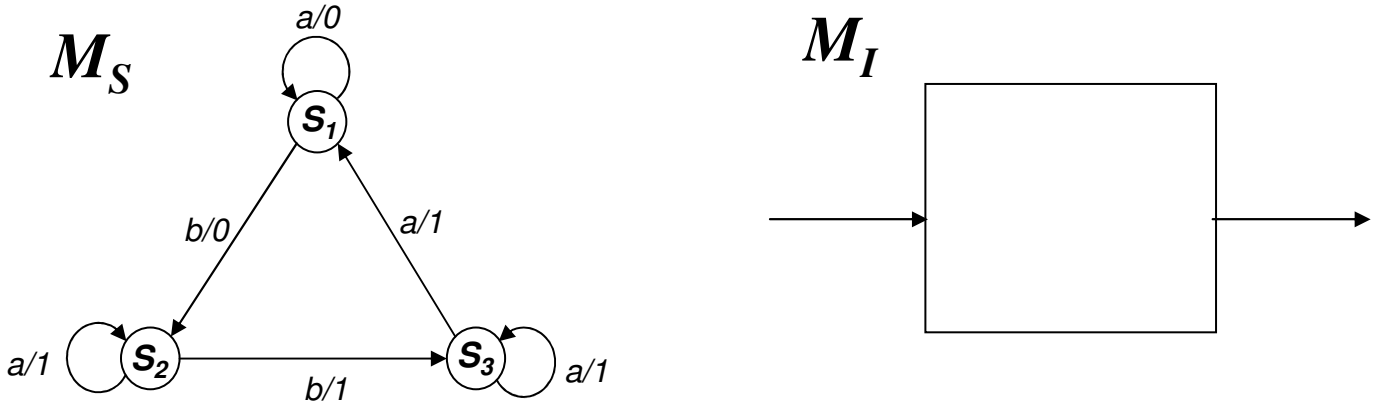
Structural (white-box) testing

- based on the structure of the program/model defining the solution to the requirements problem
- Tries to exercise the structure of the program/model
- Defined to satisfy a coverage metric, such as statement coverage, branch coverage ...
  - Allows measuring the progression of the test (the amount of coverage)
- Can be complex and long (100% coverage practically impossible in some cases)
- Incapable of detecting errors of omitted implementation or implementation out of specs
  - The algorithm is correct but does not solve the requirements problem

Conformance (model-based) testing

Given a model specification M_S, for which we know the transition diagram, and M_I, the (program) implementation of M_S and for which we can only observe the behavior, we want to know if M_I correctly implements M_S.

Also called *fault detection* or *machine verification*



Unit testing

Approach, programs and utilities to test software modules in isolation before integration