Testing

Software Analysis and Design Project

Universidad Autónoma de Madrid

What is a software error?

- A software error happens when the software does not follow the user expectations, previously defined in the requirements analysis phase.
- The main causes are user communication or coding errors.

What is software testing?

- Testing is the process of executing the software in order to find errors

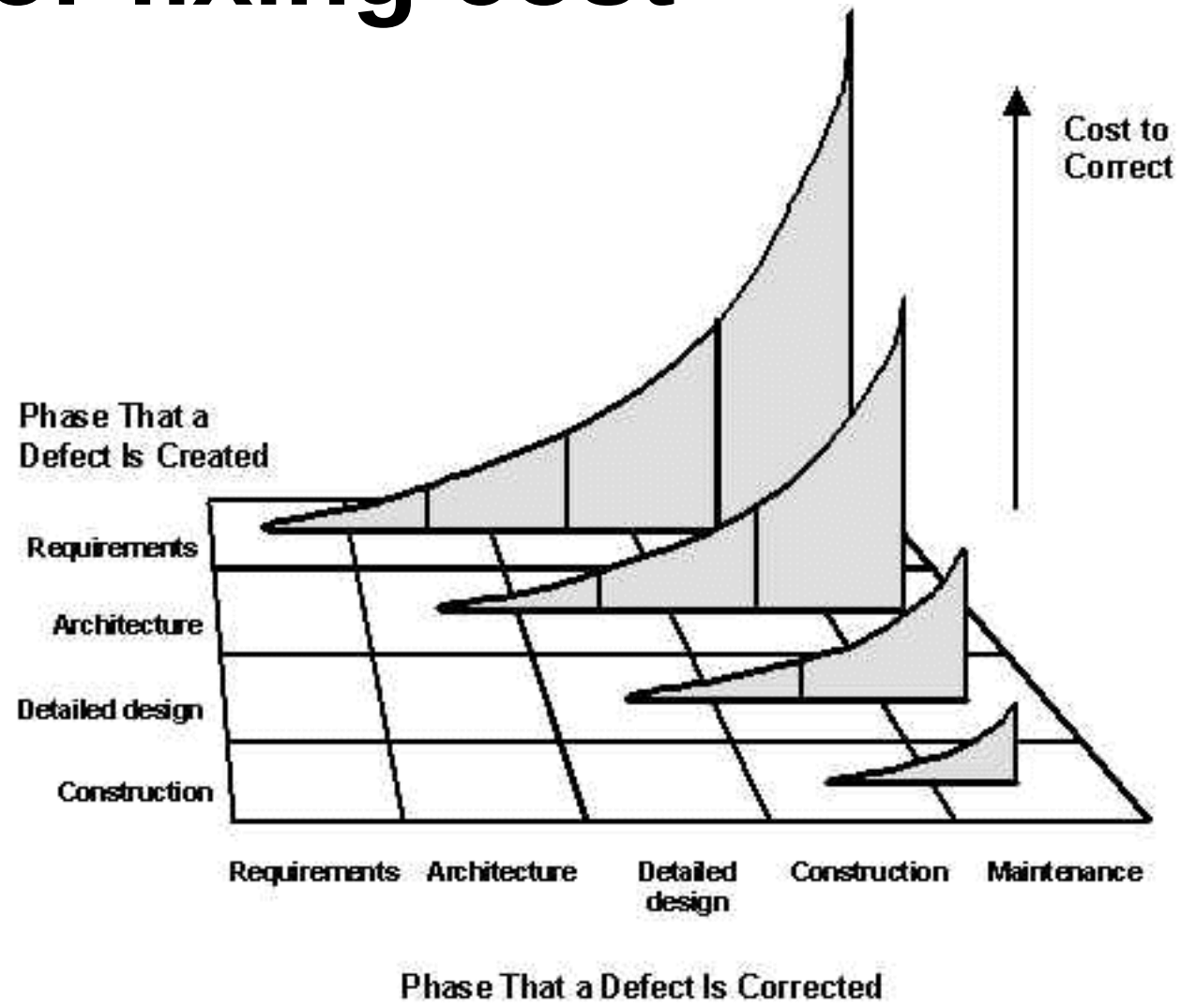
- Wrong definitions:
  - ✗ Testing consists of ensuring that there are no errors in the program.
  - ✗ Testing consists of showing that the program is executed correctly.

Validation and verification

- **Verification**
  - Have we built the system right?
  - Checks the software behavior. Ensures that the software implements properly a given functionality.

- **Validation**
  - Have we built the right system?
  - Confirms that the software will fulfill its intended use (the user requirements are met)

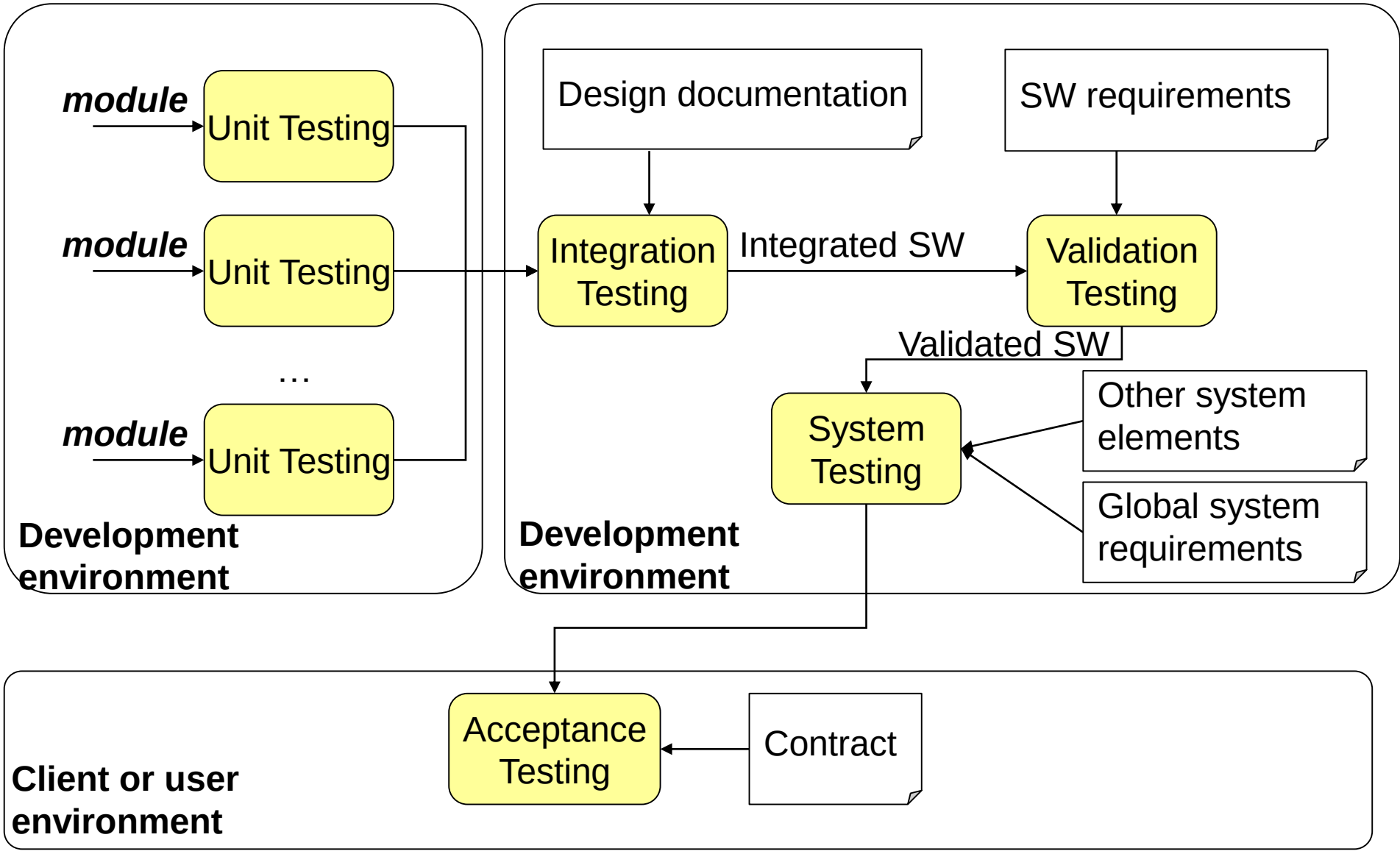
Error fixing cost



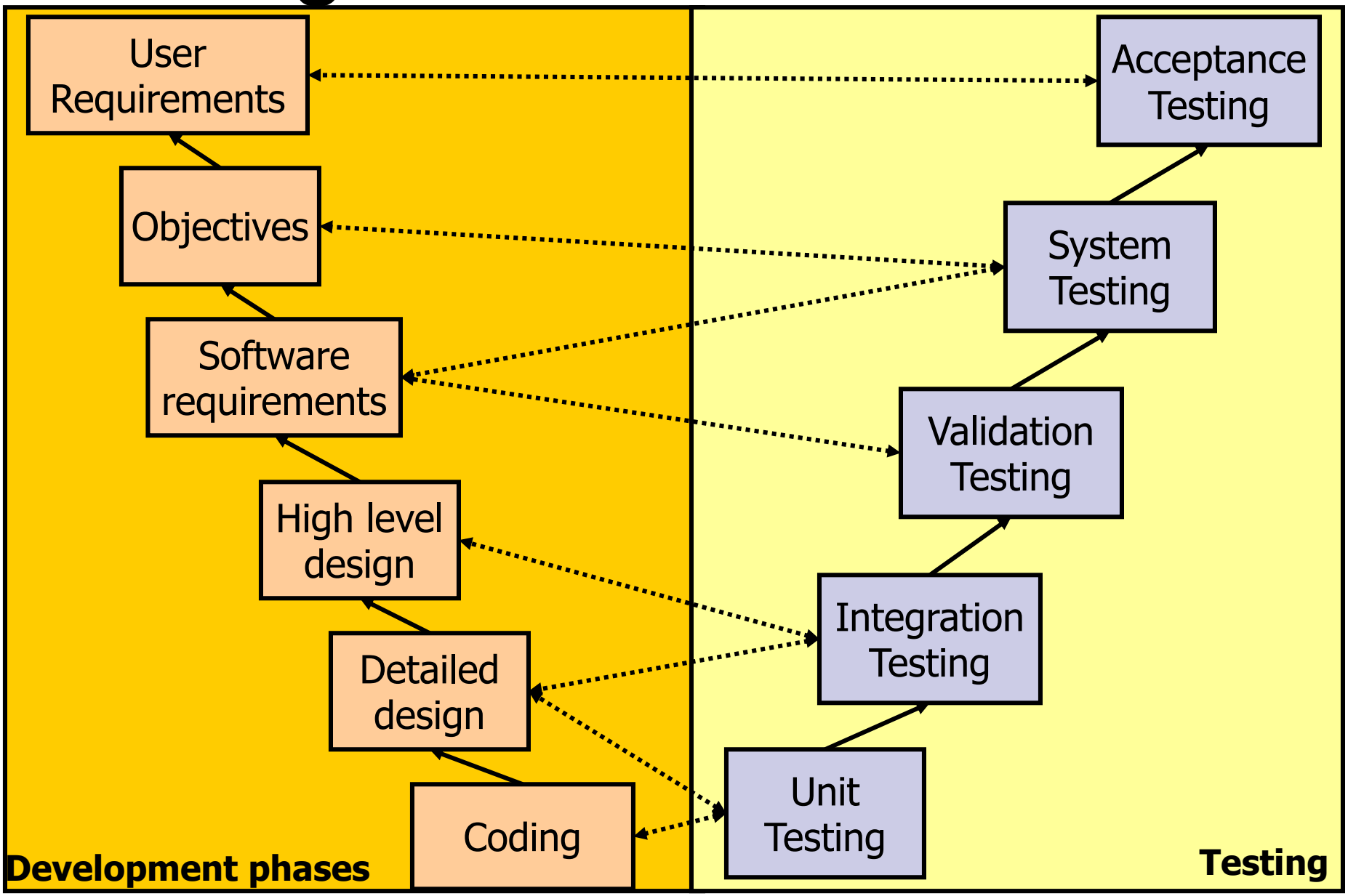
Testing levels

- Make tests from inside to outside, beginning with units and ending with the whole system:
  - **Unit testing**. Test the logic, functionality and correctness of each module or unit.
  - **Integration testing**. Verify the interfaces between components against a software design. Usually it is done following module grouping, in a iterative way.
  - **Validation testing**. Verify that the system complies with the SW requirements.
  - **System testing.** Tests a completely integrated system, with the right software and hardware environment.
  - **Acceptance testing.** Test if the user requirements are met.

Testing Levels



Testing Levels



In your Project…

- Design a set of validation tests:
  - Choose **two** Use Cases (scenarios) from the requirements specification document
  - Design, for each case, a validation test
  - Execute those validation tests in your application
- Deliverable:
  - A (brief) report, using provided template (in Moodle) with:
    - Selected Use Cases.
    - Tests description.
    - Results (screenshots and analysis).

Bibliography

- "*Software Engineering, a practitioner's approach, 7ª ed*". Roger Pressman. McGraw Hill Higher Education, 2010. INF/681.3.06/PRE. También disponible en castellano.
- "*El Arte de Probar el Software".* Glenford J. Myers. Editorial El Ateneo, 1983
- "*Testing Computer Software, 2nd Edition*". C. Kaner, J. Falk, H. Q. Nguyen. Wiley 1999.

Other interesting topics

(not covered in this course)

- Test case design (white/black box).
- Testing levels in OO systems (class testing, object grouping, etc.)
- Automatic User Interface testing
- Other techniques of validation and verification:
  - Formal methods
  - Model checking
  - Program analysis
  - …