CS496: Software Testing and Quality Assurance

Soha Makady

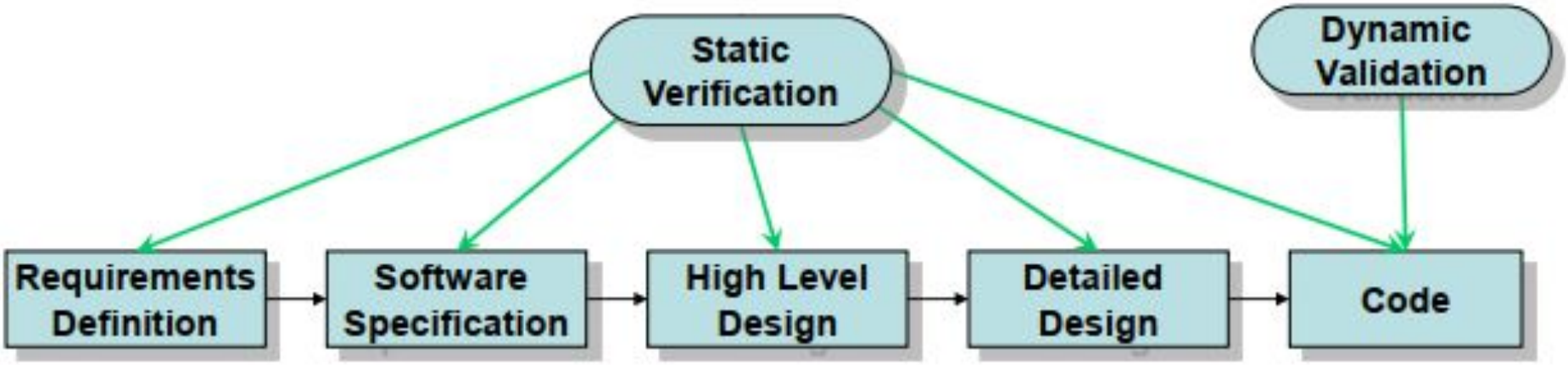
Some of the material are retrieved from a previous course offering by Prof. Amr Kamel

Outline

- Types of Testing
- Testing Levels
- Test Plans
- Dynamic Verification Techniques

Types of Testing

- Static
  - Analysis of the static system representation to discover problems.
- Dynamic
  - Exercising and observing the software behaviour.



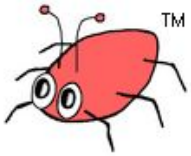
Commonly, testing refers to dynamic testing.

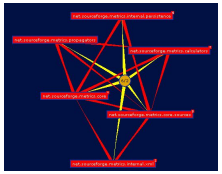
Static Testing Techniques

- Static Testing [before compile time]
  - Static Analysis
  - Review
  - Walk-through [informal]
  - Code inspection [formal]
- Dynamic Testing [at run time]
  - Black-box testing!!
  - White-box testing!!
  - Testing Scope

Static Analysis with Eclipse

- Compiler Warnings and Errors
  - *Possibly uninitialized Variable*
  - *Undocumented empty block*
  - *Assignment has no effect*
- Checkstyle
  - Check for code guideline violations
  - http://checkstyle.sourceforge.net
- FindBugs
  - Check for code anomalies
  - http://findbugs.sourceforge.net
- Metrics
  - Check for structural anomalies
  - http://metrics.sourceforge.net

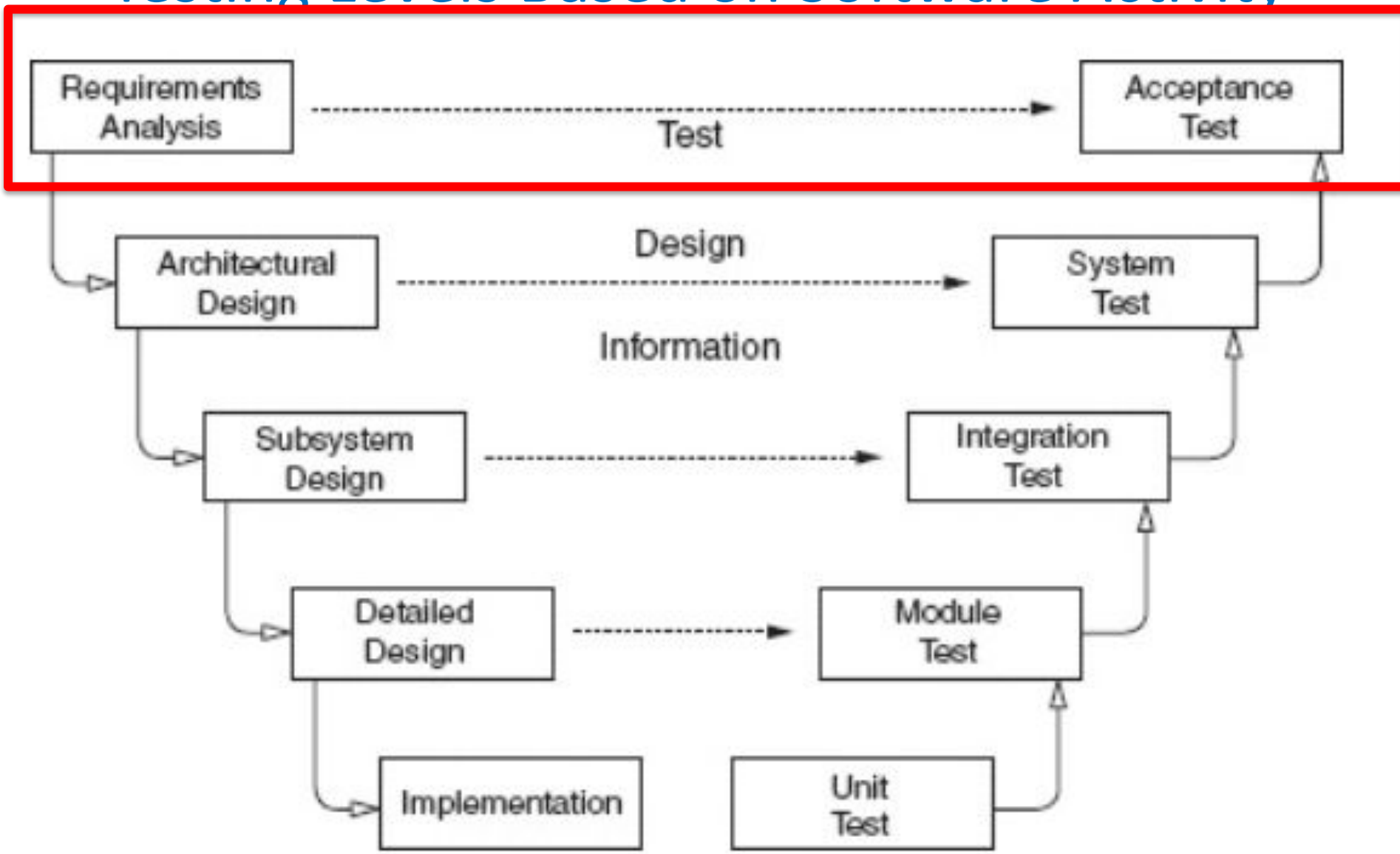




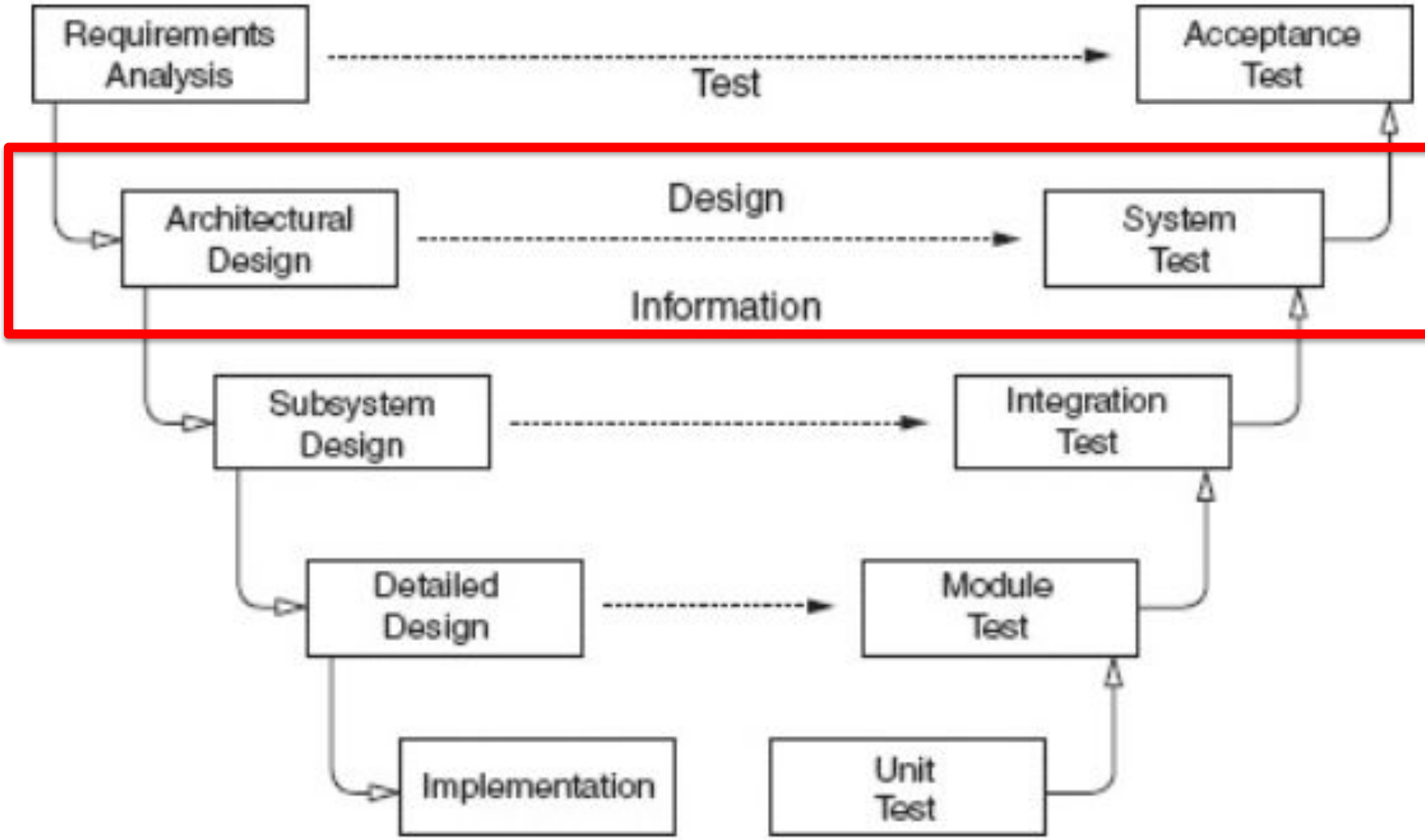
Testing Levels Based on Software Activity

- Tests can be derived from requirements and specifications, design artifacts, or the source code
  - Acceptance testing
  - System testing
  - Integration testing
  - Module testing
  - Unit testing
  -

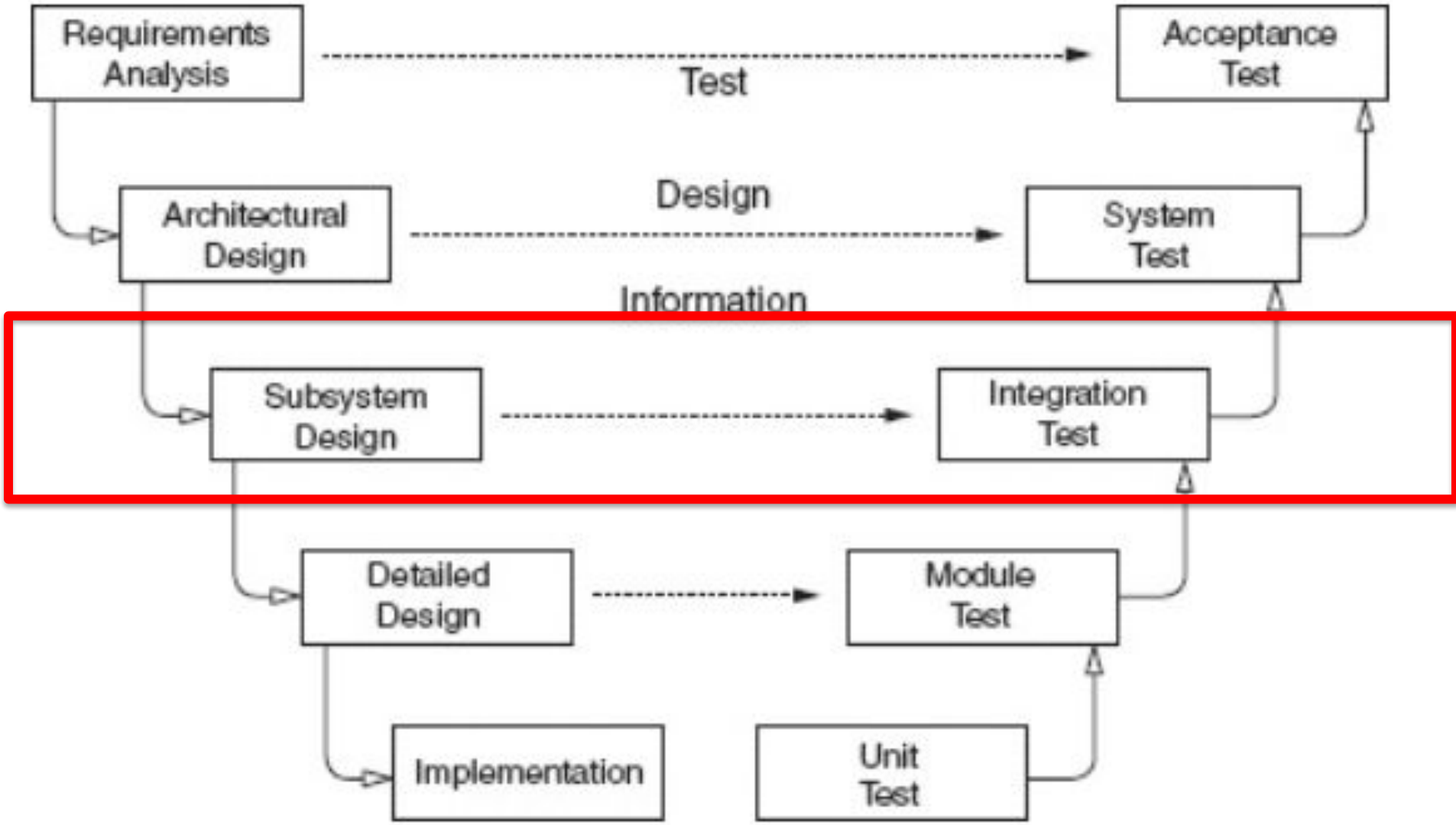
Testing Levels Based on Software Activity



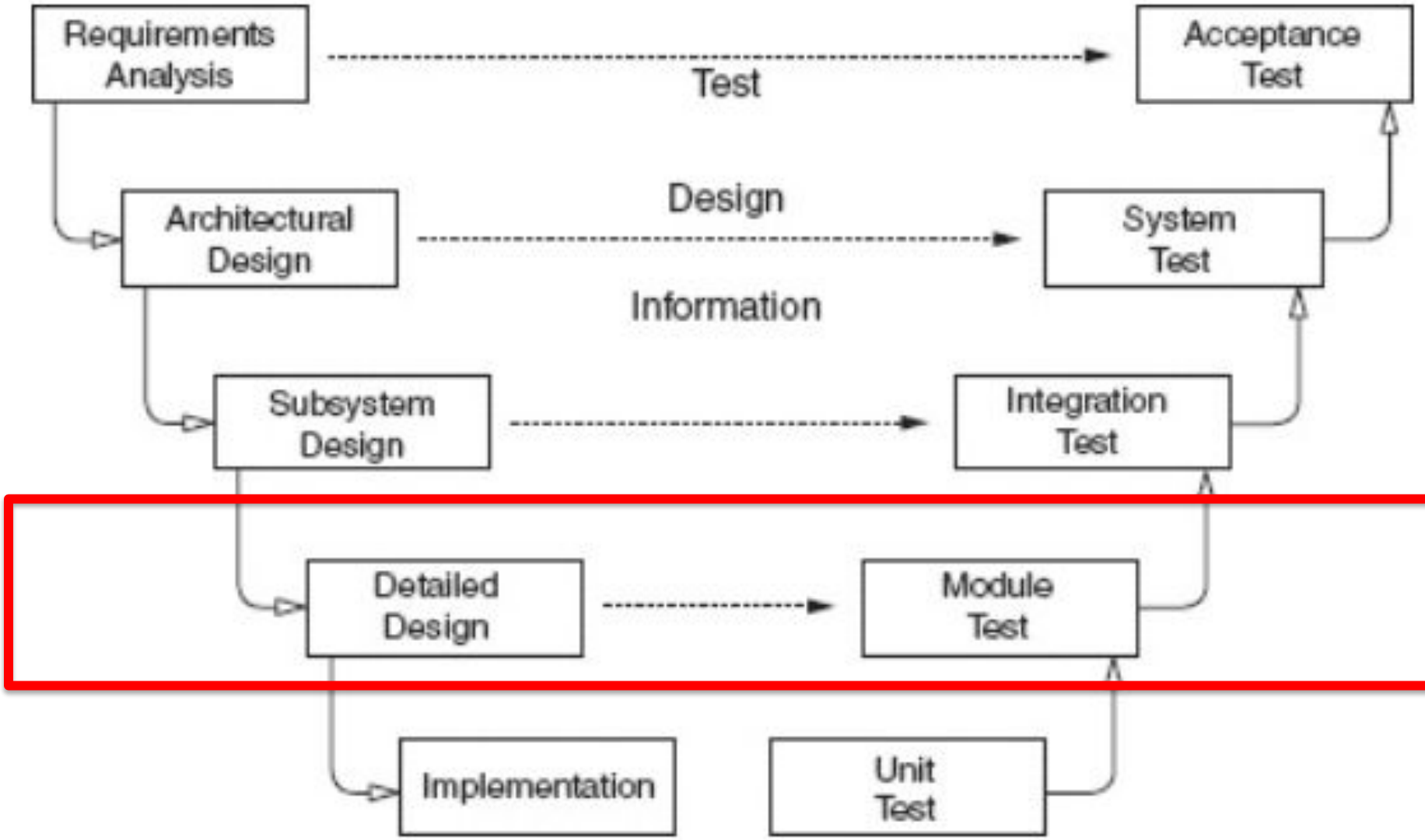
Testing Levels Based on Software Activity



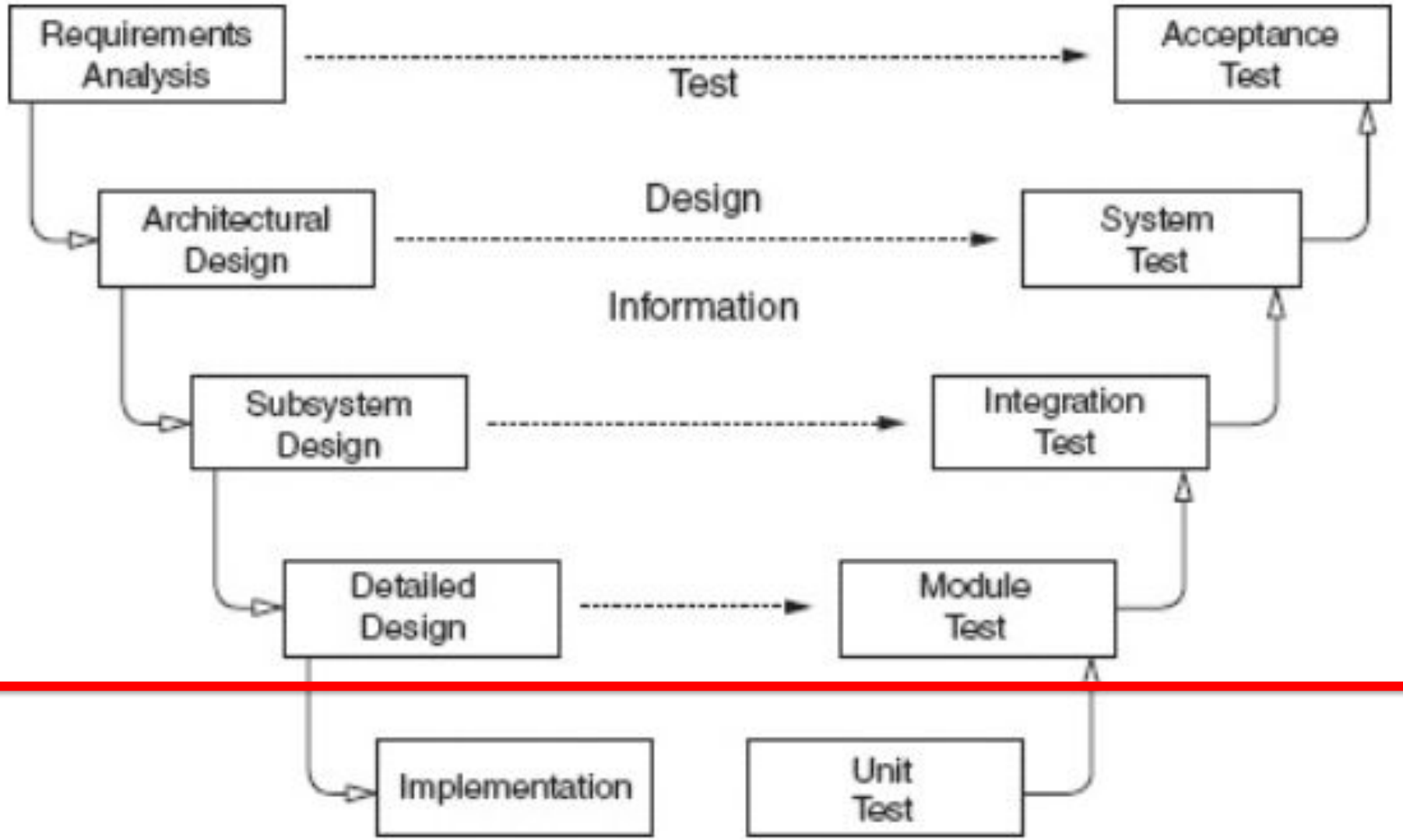
Testing Levels Based on Software Activity



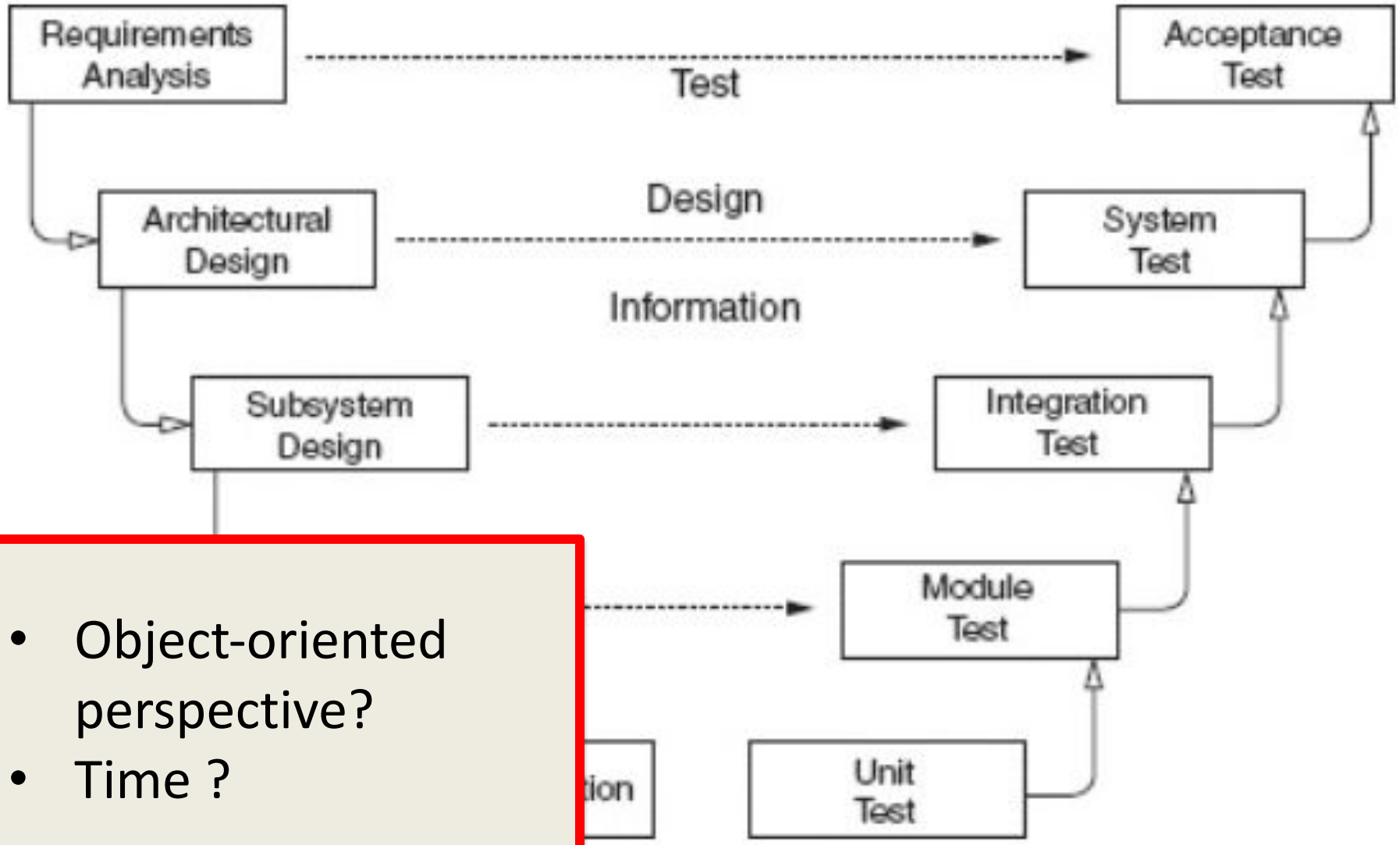
Testing Levels Based on Software Activity



Testing Levels Based on Software Activity

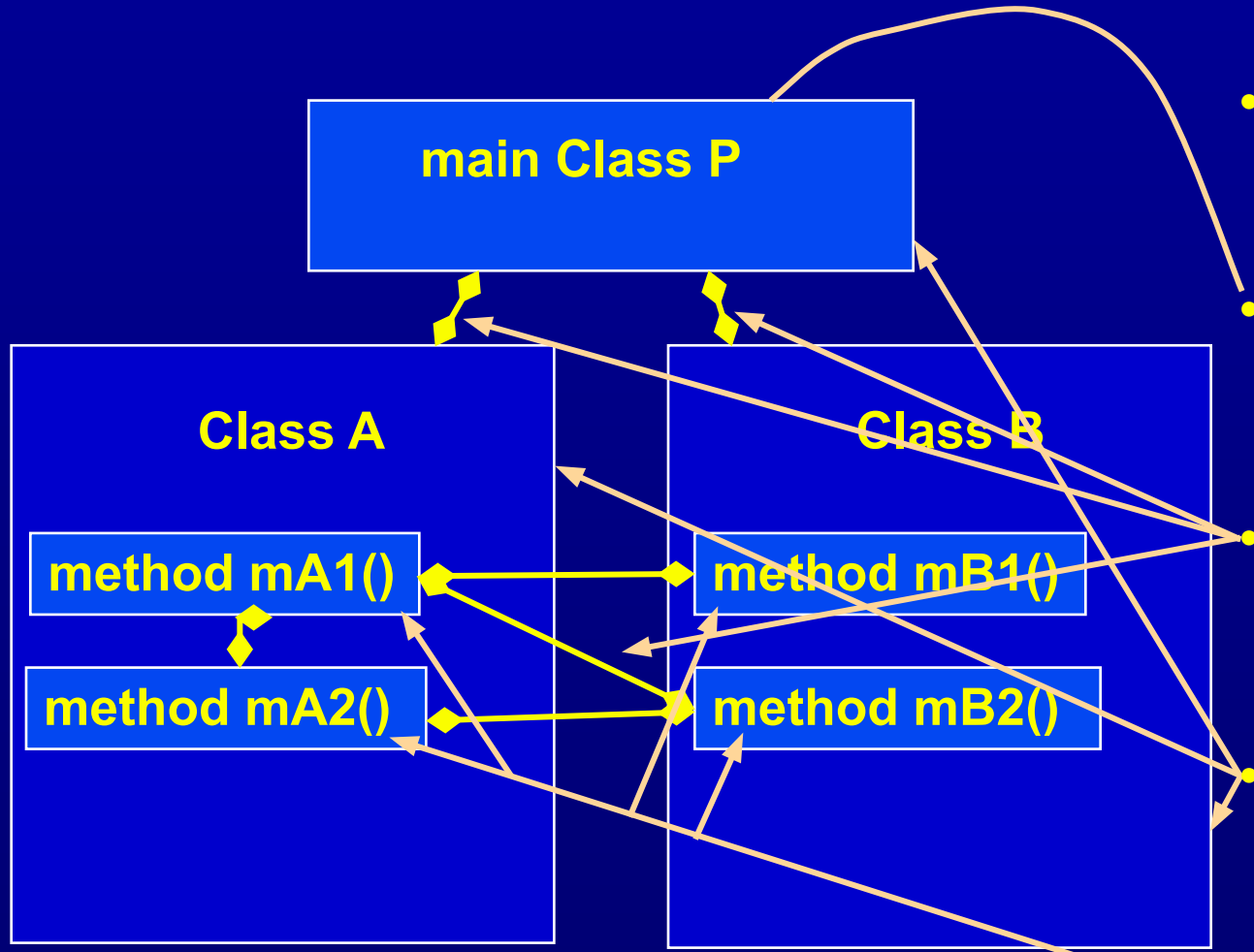


Testing Levels Based on Software Activity



- Object-oriented perspective?
- Time ?

Traditional Testing Levels

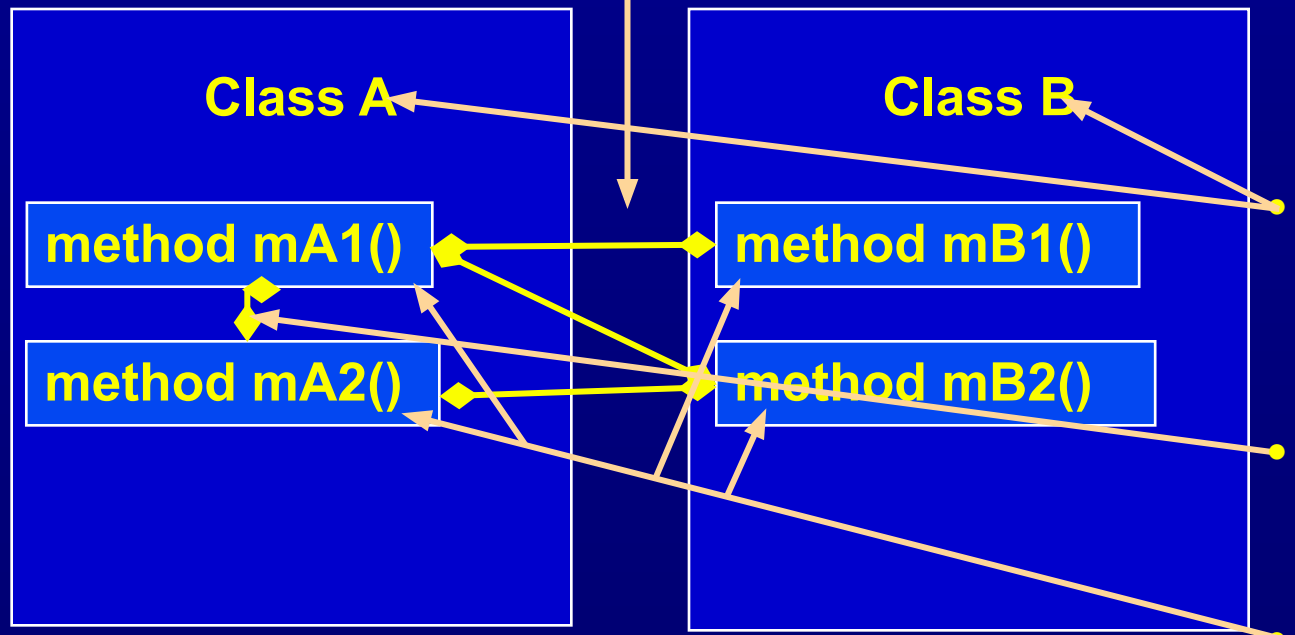


- **Acceptance testing :** Is the software acceptable to the user?
- **System testing :** Test the overall functionality of the system
- **Integration testing :** Test how modules interact with each other
- **Module testing (developer testing) :** Test each class, file, module, component
- **Unit testing (developer testing) :** Test each unit (method) individually

This view obscures underlying similarities

Object-Oriented Testing Levels

- **Inter-class testing :** Test multiple classes together

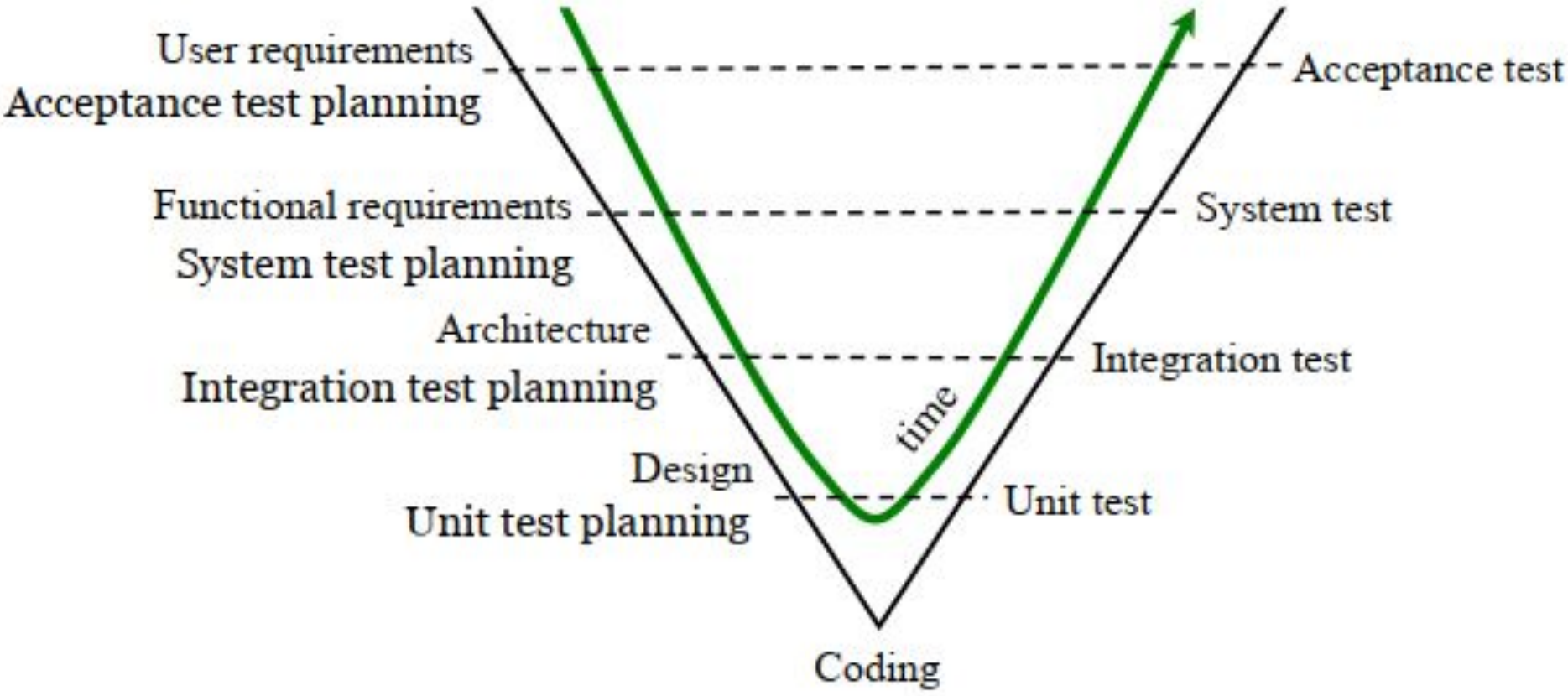


Intra-class testing : Test an entire class as sequences of calls

Inter-method testing : Test pairs of methods in the same class

Intra-method testing : Test each method individually

The V-Model


User requirements
Acceptance test planning
Acceptance test
Functional requirements
System test planning
System test
Architecture
Integration test planning
Integration test
time
Design
Unit test planning
Unit test
Coding

Test Plans

- A plan is a document that provides a framework or approach for achieving a set of goals.
- Test plans are detailed documents, including an essential set of items:
  1. Overall test objectives:
  2. What to test (scope of the tests).
  3. Who will test.
  4. How to test.
  5. When to test.
  6. When to stop testing.

Test Plans

1. Overall test objectives.

Test Plan

- Introduction
- Risks and contingencies

Test Plans

- 2. Scope of the tests.

Test Plan

- Introduction
- Items to be tested (e.g., procedures, classes, modules, libraries).
- Features to be tested (e.g., functional requirements, performance requirements).
- Features not to be tested (with reasons for exclusion).
- Risks and contingencies

Test Plans

- 3. Who will test.

Test Plan

- Introduction
- Items to be tested.
- Features to be tested
- Features not to be tested.
- Responsibilities
- Staffing and training needs
- Risks and contingencies

Test Plans

•4. How to test.

- What strategies, methods, hardware, software tools, and test techniques will be applied? E.g., what percentage of test coverage is expected?
- What test deliverables and documents should be produced? These include::
  - Test design specifications
  - Test cases
  - Test logs
  - Test summary reports

Test Plan

- Introduction
- Items to be tested.
- Features to be tested
- Features not to be tested.
- Approach
- Test deliverables
- Responsibilities
- Staffing and training needs
- Risks and contingencies

Test Plans

- 5. When to test.

Test Plan

- Introduction
- Items to be tested.
- Features to be tested
- Features not to test
- Approach
- Test Deliverables
- Responsibilities
- Staffing and training needs
- Schedule
- Risks and contingencies

Test Plans

- 6. When to stop testing.

Test Plan

- Introduction
- Items to be tested.
- Features to be tested
- Features not to test
- Approach
- Test Deliverables
- Responsibilities
- Staffing and training needs
- Schedule
- Risks and contingencies

Test Plans

- Test plan identifier
- A unique identifier

- Test plan identifier
- Introduction
- Items to be tested.
- Features to be tested
- Features not to test
- Approach
- Test Deliverables
- Responsibilities
- Staffing and training needs
- Schedule
- Risks and contingencies

Test Plans

- Item Pass/Fail Criteria

- Test plan identifier
- Introduction
- Items to be tested.
- Features to be tested
- Features not to test
- Approach
- Test Deliverables
- Item pass/fail criteria
- Responsibilities
- Staffing and training needs
- Schedule
- Risks and contingencies

Test Plans

- Suspension/resumption criteria

Test Plan

- Test plan identifier
- Introduction
- Items to be tested.
- Features to be tested
- Features not to test
- Approach
- Item pass/fail criteria
- Suspension/resumption criteria
- Responsibilities
- Staffing and training needs
- Schedule
- Risks and contingencies

Test Plans

- The testing tasks
  - Describes all the testing related activities

- Test plan identifier
- Introduction
- Items to be tested.
- Features to be tested
- Features not to test
- Approach
- Item pass/fail criteria
- Suspension/resumption criteria
- Test deliverables
- Testing tasks
- Responsibilities
- Staffing and training needs
- Schedule
- Risks and contingencies

Test Plans

- The testing environment
  - Describes the software/hardware neede the testing effort.

- Test plan identifier
- Introduction
- Items to be tested.
- Features to be tested
- Features not to test
- Approach
- Item pass/fail criteria
- Suspension/resumption criteria
- Test deliverables
- Testing tasks
- Environmental needs
- Responsibilities
- Staffing and training needs
- Schedule
- Risks and contingencies

Test Plans

- The testing costs

Test Plan

1. Test plan identifier
2. Introduction
3. Items to be tested.
4. Features to be tested
5. Features not to test
6. Approach
7. Item pass/fail criteria
8. Suspension/resumption criteria
9. Test deliverables
10. Testing tasks
11. Environmental needs
12. Testing costs
13. Responsibilities
14. Staffing and training needs

Required Readings

- Practical Software Testing
  - Chapter 2: Testing Fundamentals
- An Introduction to Software Testing
  - Chapter 2