Testing Techniques

Test Organization : TMap

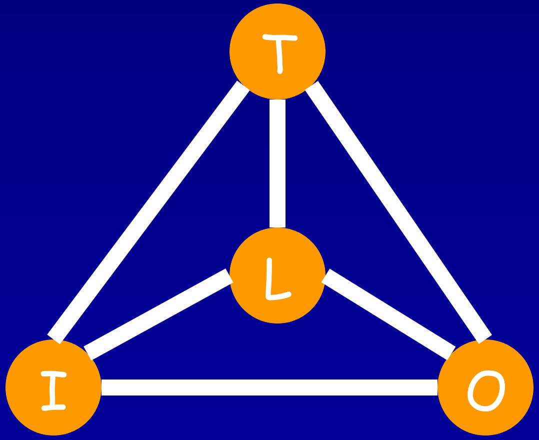
Jan Tretmans

ESI
Eindhoven, NL

Radboud University
Nijmegen, NL

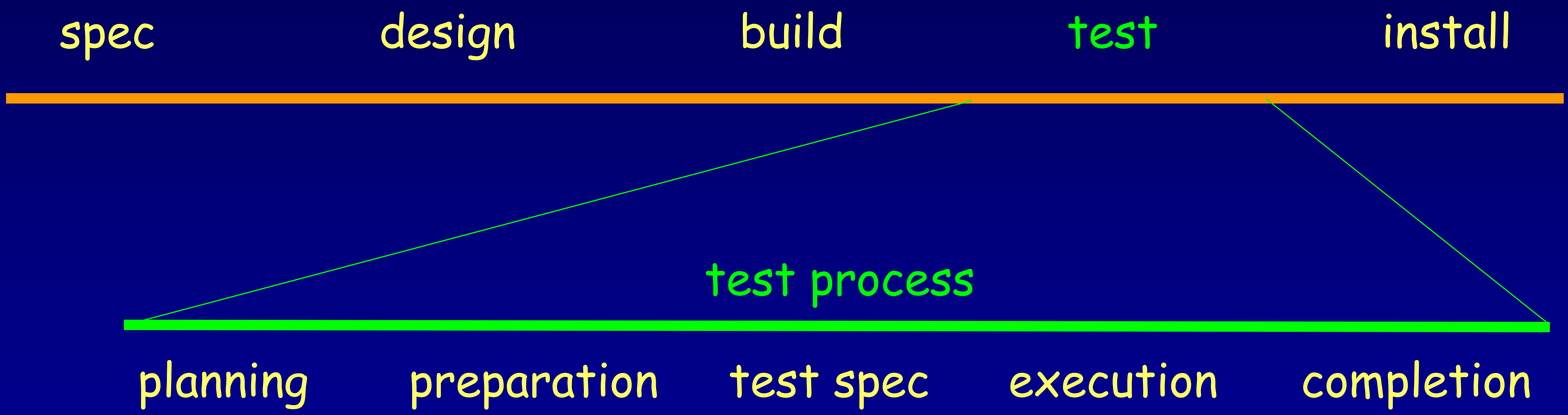
TMap®: Test Management approach

- Approach to organization and structuring of testing
- Developed and promoted by IQUIP Informatica B.V. (NL) - now Sogeti - and others
- Mainly applied to administrative software testing
- Testing as a process - in addition to development process

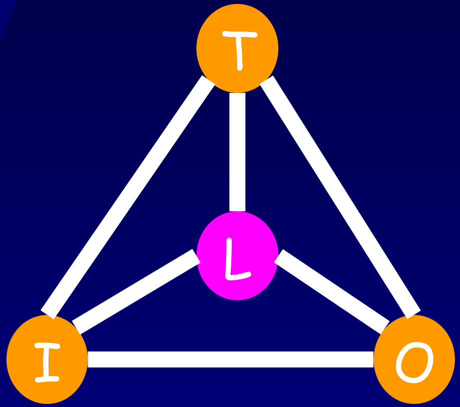


L	:	Life-cycle for testing
O	:	Organization
I	:	Infrastructure and tools
T	:	Techniques

The Software Development Trajectory

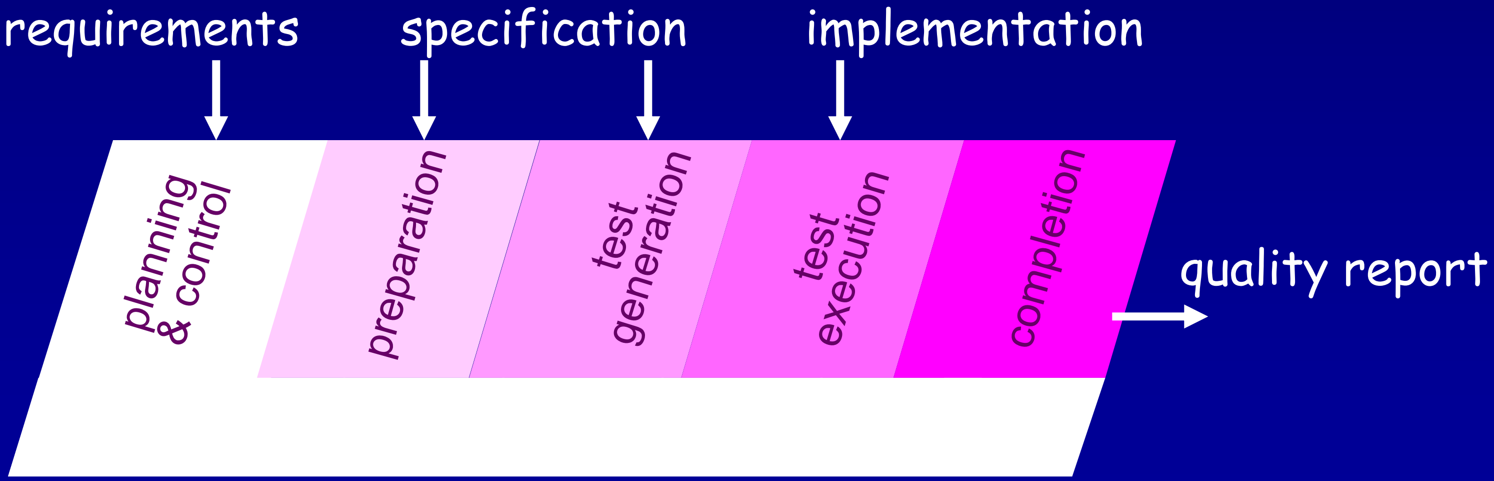


Life Cycle : Testing as a Process

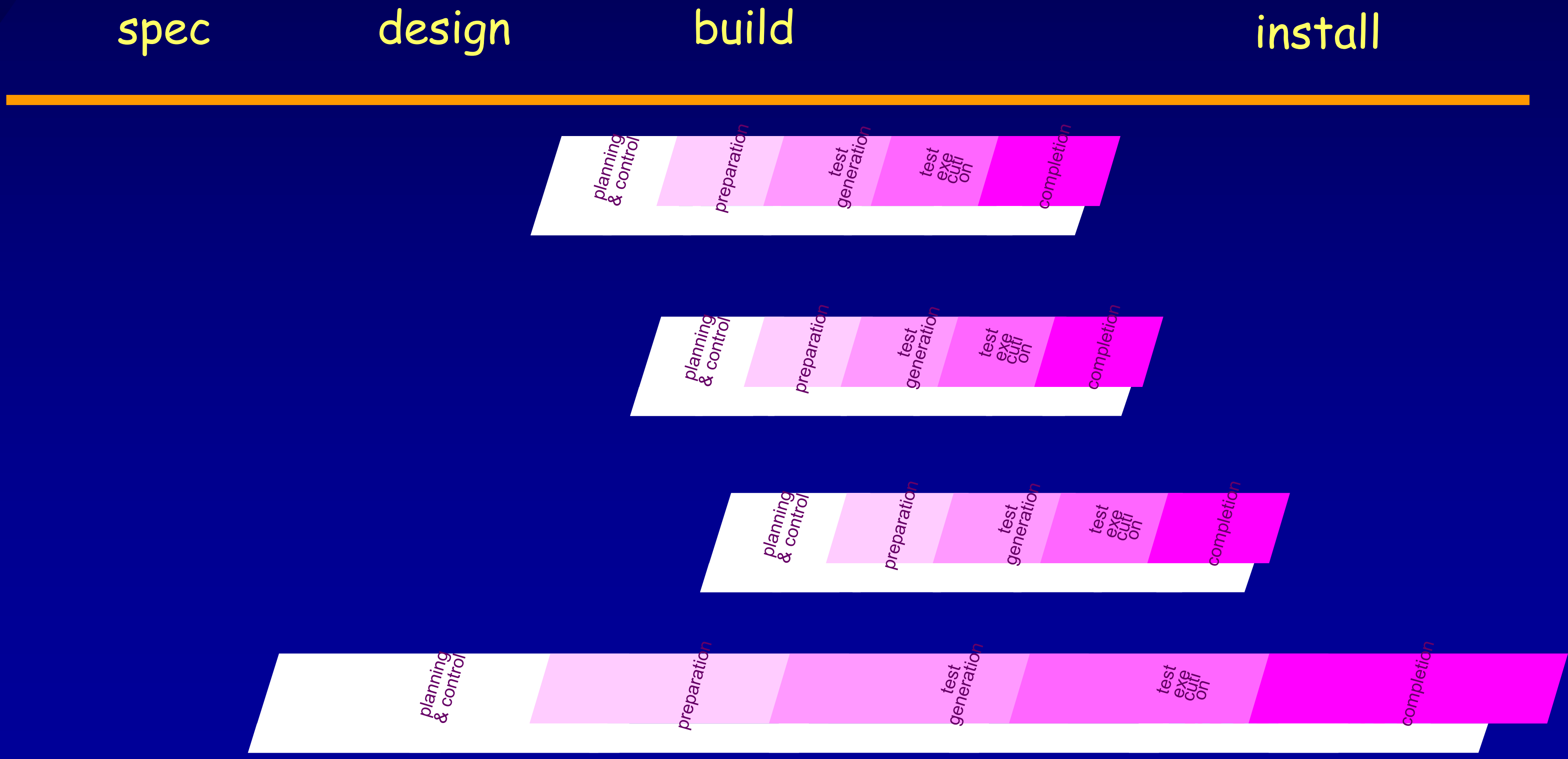


Testing as a process itself

- with its own phases
- in parallel with development process



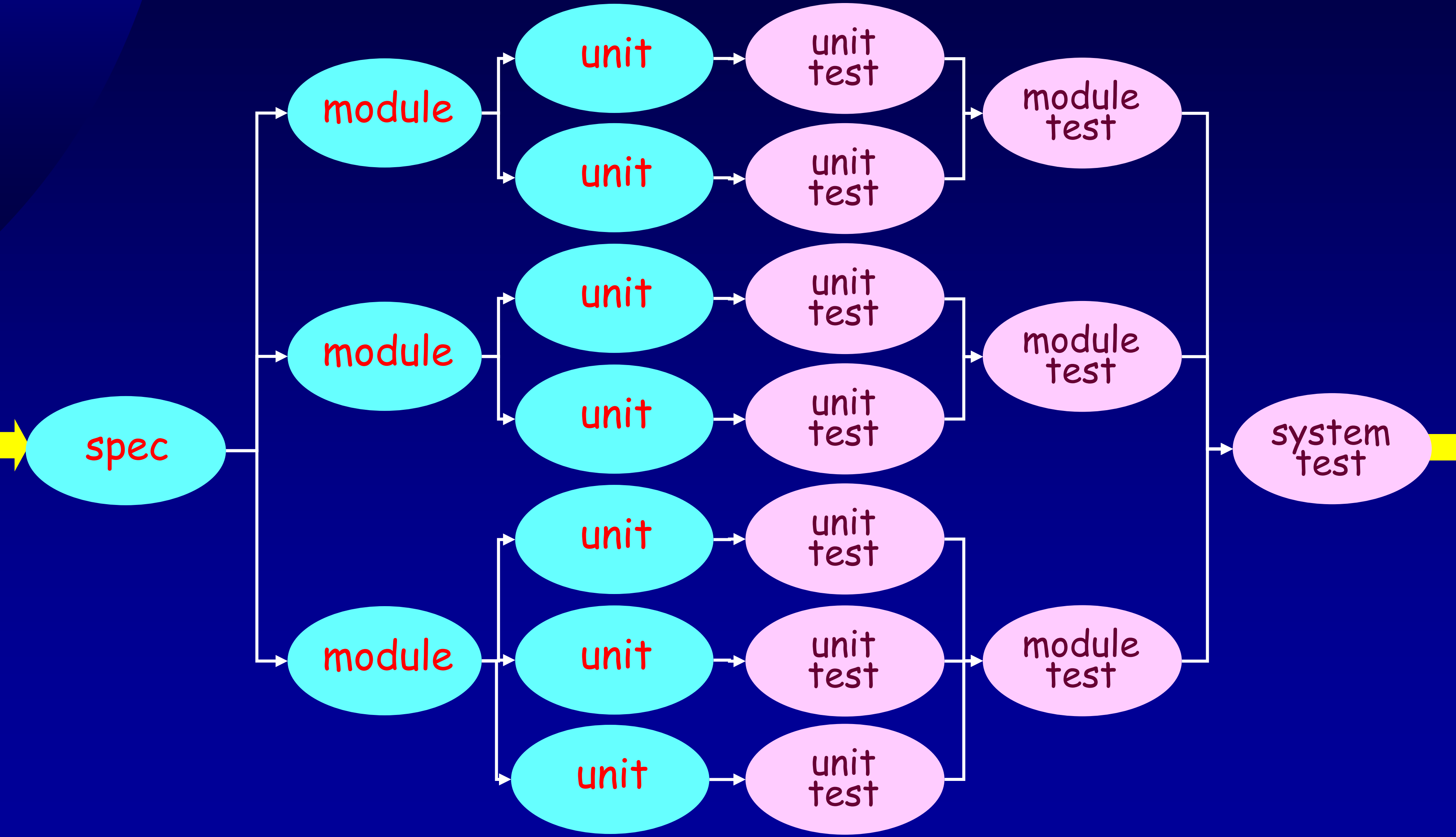
Testing in the Development Trajectory



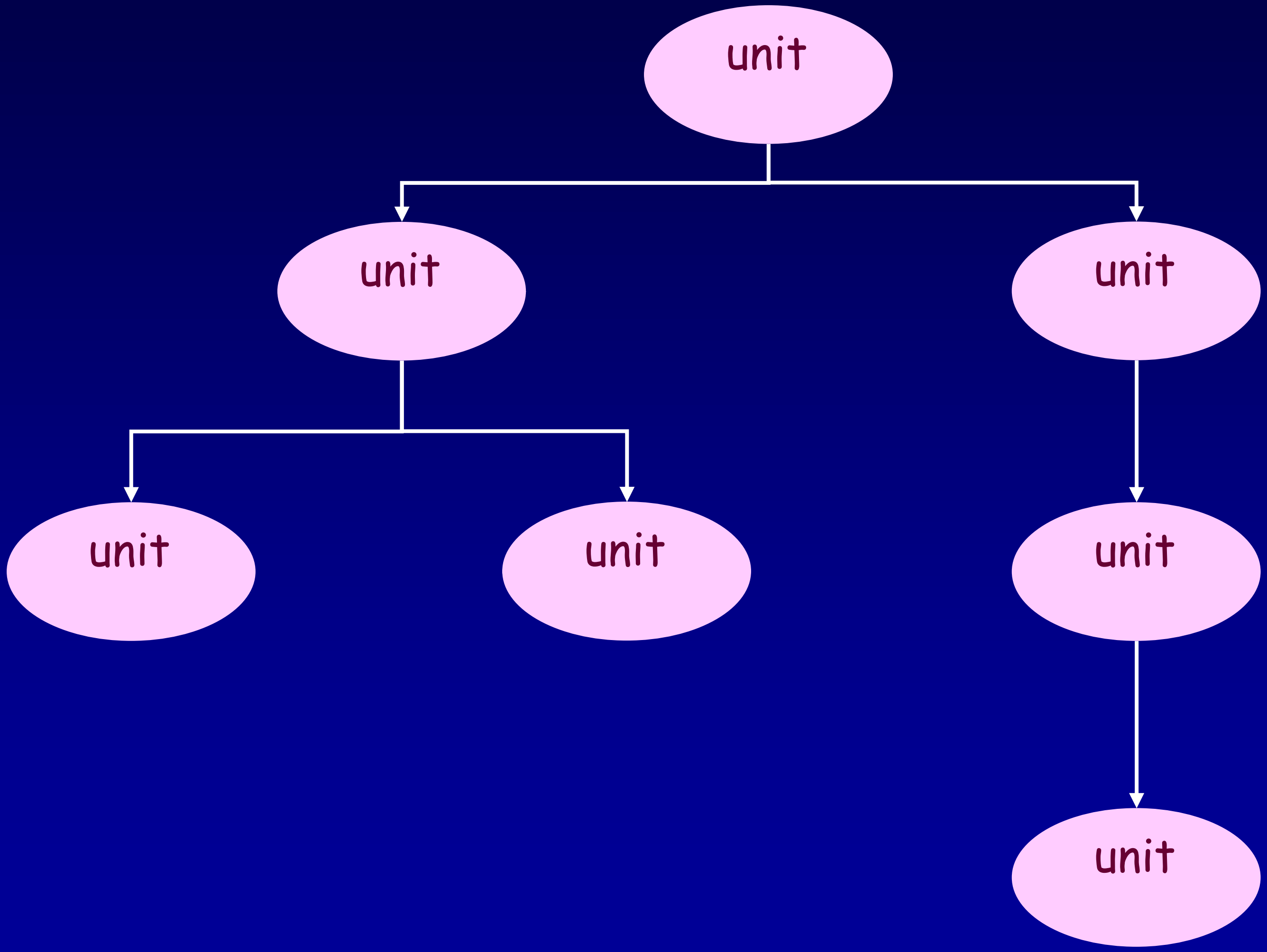
Phase 1: Test Planning and Control

- Start at requirements phase of system development
- Development of master test plan and derived test plans
- Under responsibility of test manager
- Description of
  - what: objectives, tasks, deliverables
  - by whom: personnel, responsibilities
  - with what: infrastructure
  - in which time: planning
- Risk assessment : what and how thoroughly to test
- Control and management during remaining phases

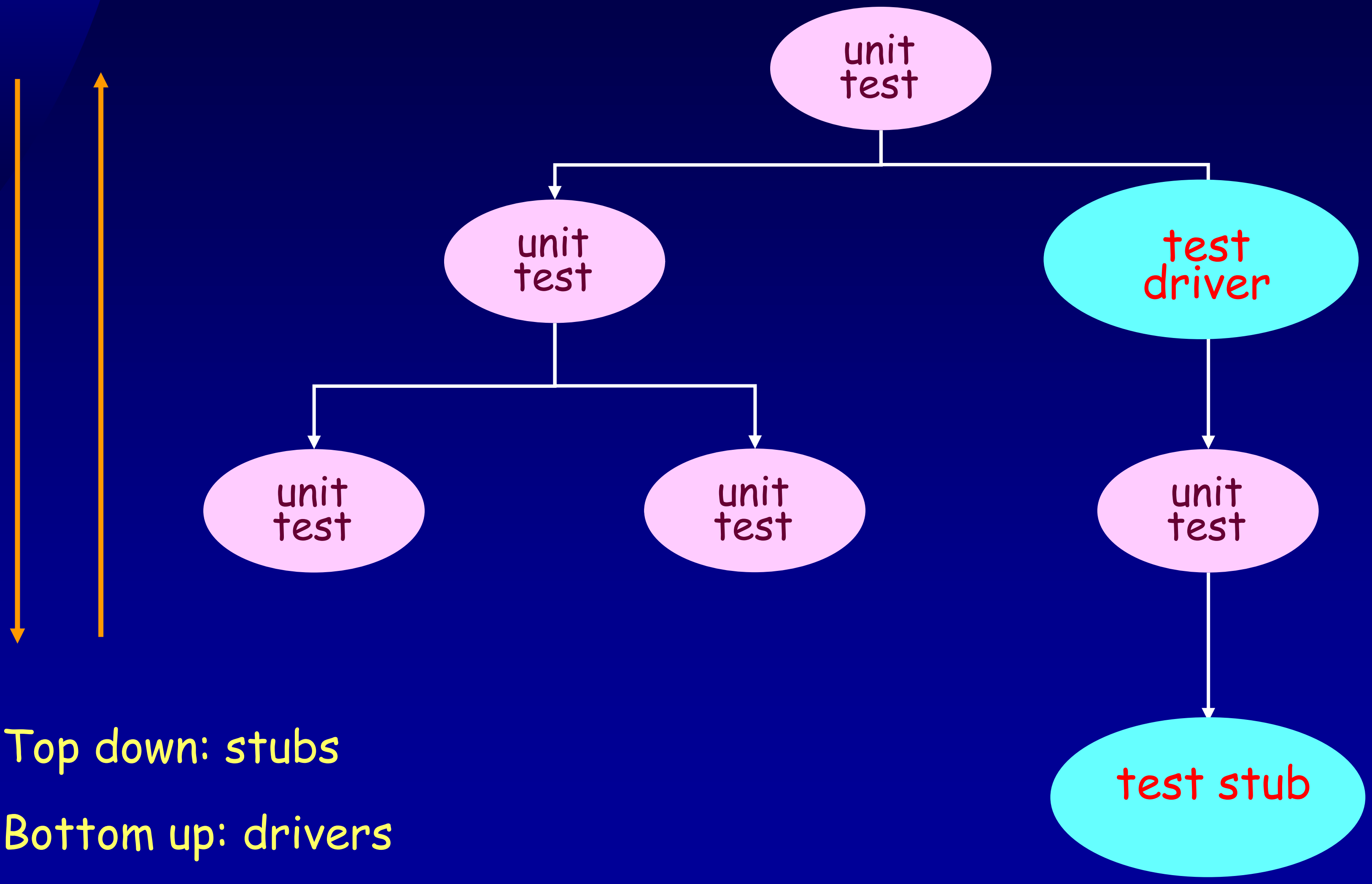
Integration and Test Planning



Modules using Each Other



Module Testing: Drivers and Stubs



Top down: stubs

Bottom up: drivers

Phase 1: Test Strategy-- Plan

- Test Mission/Vision
  - global, politically oriented, goal of testing for company
- Test Strategy
  - high level test approach for company, department
  - which levels of testing, techniques for testing, ...
  - For:
    - external communication: what is testing, business, IT, auditors
    - intro to new team members
    - internal communication: shared common understanding
- Test Approach
  - implementation of test strategy for project
  - risk assessment, test projects goals, starting points for testing, ...
- Test Plan
  - implementation of test approach:
    who is doing what and when, with what, what costs, ...

Phase 2: Preparation

- ☞ Study of test basis
  = specification and other documentation as basis for testing
- ☞ Reviewing of specifications
- ☞ Check of testability of specifications
- ☞ Specifications under formal change and configuration control
- ☞ Division of system into sub-systems which will be separately delivered and tested

Phase 3: Test Generation (Specification)

- ☞ Generation and specification of test cases
- ☞ Test case (= logical test case design = abstract test case) :
  - ♦ purpose
  - ♦ starting situation
  - ♦ input and changes to be performed
  - ♦ expected output
  - ♦ expected resulting situation
- ☞ Development of infrastructure for test execution
- ☞ Implementation of test cases on infrastructure
  (= physical test case design = test scripts = executable tests)

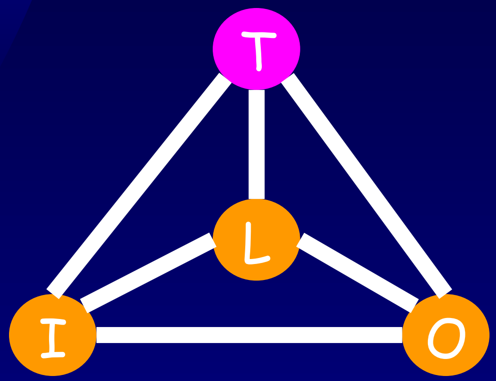
Phase 4: Test Execution

- Starts when testable code components are available
- Testing in phases:
  - static "tests" = (completeness) checks
  - basic functionality tests (pre-tests)
  - full functional testing
- Test execution:
  - Test
  - Repair
  - Re-test
- Discrepancy between actual and expected result:
  - defect in implementation
  - ambiguity in specification
  - invalid test case
  - error in test infrastructure
- Reporting about testing and quality
  - defects found
  - what has been / needs to be tested
  - trends

Phase 5: Completion

- Final reporting : remaining risks
- Preservation of testware
  - reuse during regression / maintenance testing
- Evaluation
- After completion:
  - defects found by users
  - continuous testing, management and control
  - keep consistency between different configurations of specifications, implementations and testware

Techniques



Standardized / known techniques:

- standard and known way of working
- allows checking by management / auditors
- reproducibility

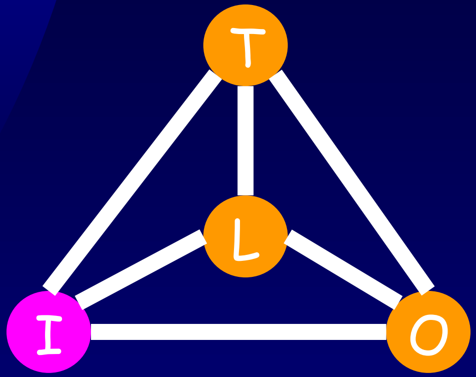
☞ Templates

☞ Checklists

☞ Test generation techniques - derive test cases from specification / test basis :

- equivalence partitioning, boundary value analysis, decision trees, . . .

Infrastructure and Tools



Test environment

- laboratory environment
- production environment

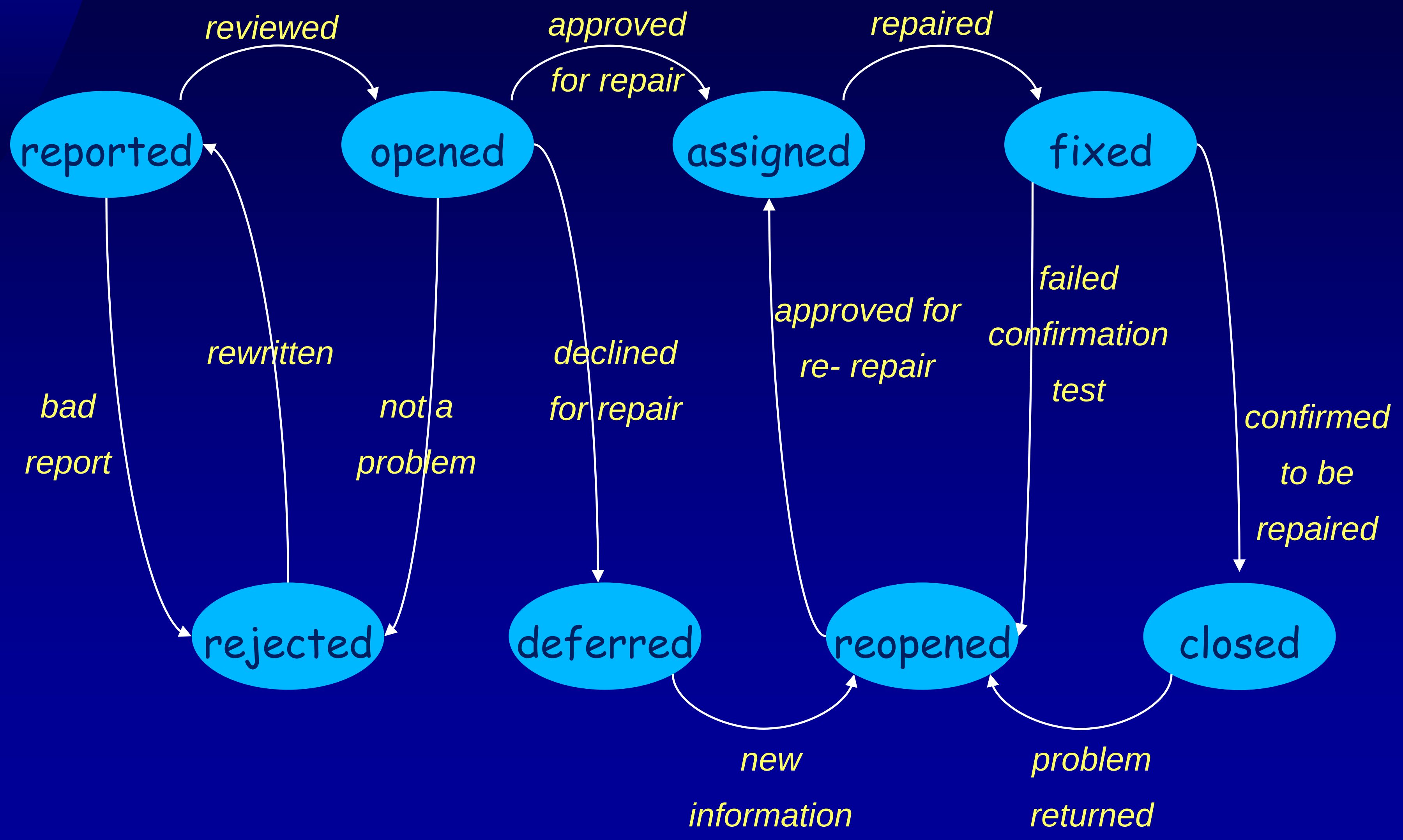
☞ Tools - classified according to life-cycle phase that is supported :

- Planning and control : standard planning and tracking tools, configuration management, traceability, defect administration and tracking, . . .
- Test specification : editor, spreadsheet, load generation, . . .
- Test execution : load generation, capture & playback, “diff”, test (code) coverage, monitor, . . .
- Completion : reporting tools

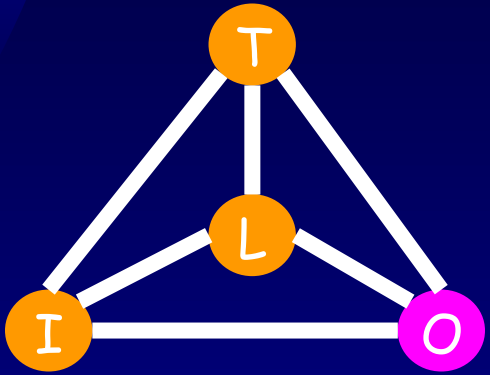
Configuration Management

- About the items that constitute the system and building it
  - source code
  - object code
  - third party software
  - hardware
  - compilers
  - build scripts
  - delivered systems
  - parameters
  - documentation
  - test environments
  - test scripts
  - test results
- All versions of these items
- Management of these items throughout project and system life cycle
- Testing:
  - manage testware in same configuration management system,
  - test proper versions
  - test cases attached to proper versions
  - defect reports attached to proper versions
  - interface to development process

Incident Life Cycle



Organization



Organization of test process itself:

- control of resources
- availability of tools
- people
- education and competences

☞ Embedding in project

- independent test team
- status of test manager

☞ Embedding in organization

- separate test department
- test support department
- relation to quality assurance and auditing