



Sommerville Chapter 8

Software Testing: Definitions and Fundamentals

Announcements

- Homework 3 handout will be available later today.
- Week 7 lab starts Monday. You can drop in to any one of the sessions next week.

Topics for Today

- Some definitions
 - Let's get the language right
- What is a test?
- Testing strategies
 - How do we tackle a testing project

Verification vs. Validation

- Verification:

- ??????

- Validation:

- ??????

Verification and Validation: IEEE

■ Verification

- The process of evaluating a system or component to determine whether the products...satisfy the conditions imposed...

■ Validation

- The process of evaluating a system or component...to determine whether it satisfies specified requirements.

Verification and Validation: Kaner

■ Verification

- Checking a program against the most closely related design documents or specifications

■ Validation

- Checking the program against the published user or system requirements

Verification and Validation: Myers

■ Verification

- An attempt to find errors by executing a program in a test or simulated environment

■ Validation

- An attempt to find errors by executing a program in a real environment

Verification vs. Validation: Right Definition

■ Verification:

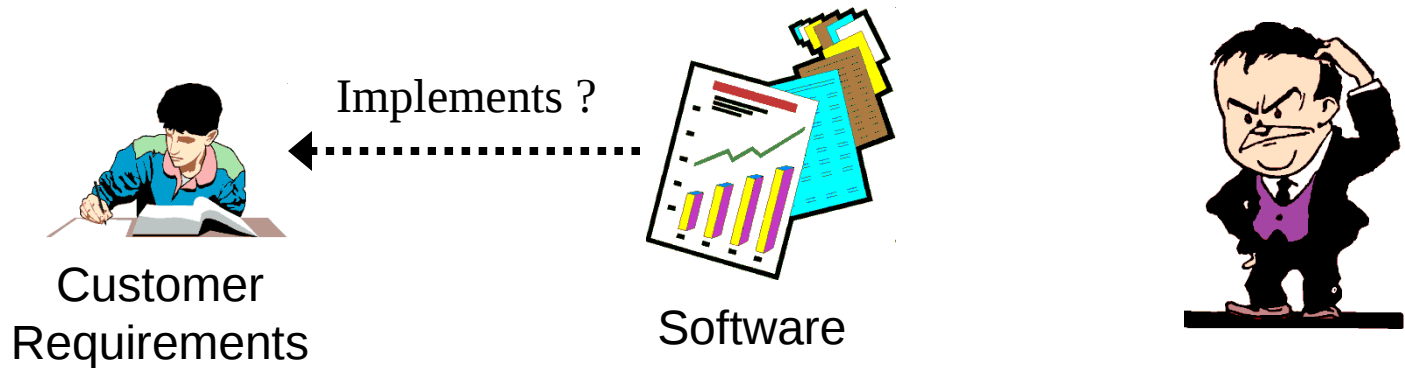
- “are we building the product right?”
- The software should conform to its specification

■ Validation:

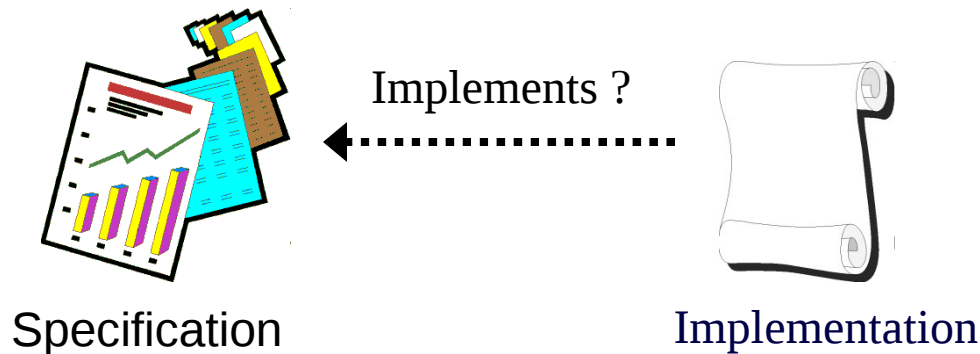
- “are we building the right product?”
- The software should do what the user really requires

Validation and Verification

Validation: Are we building the right product?



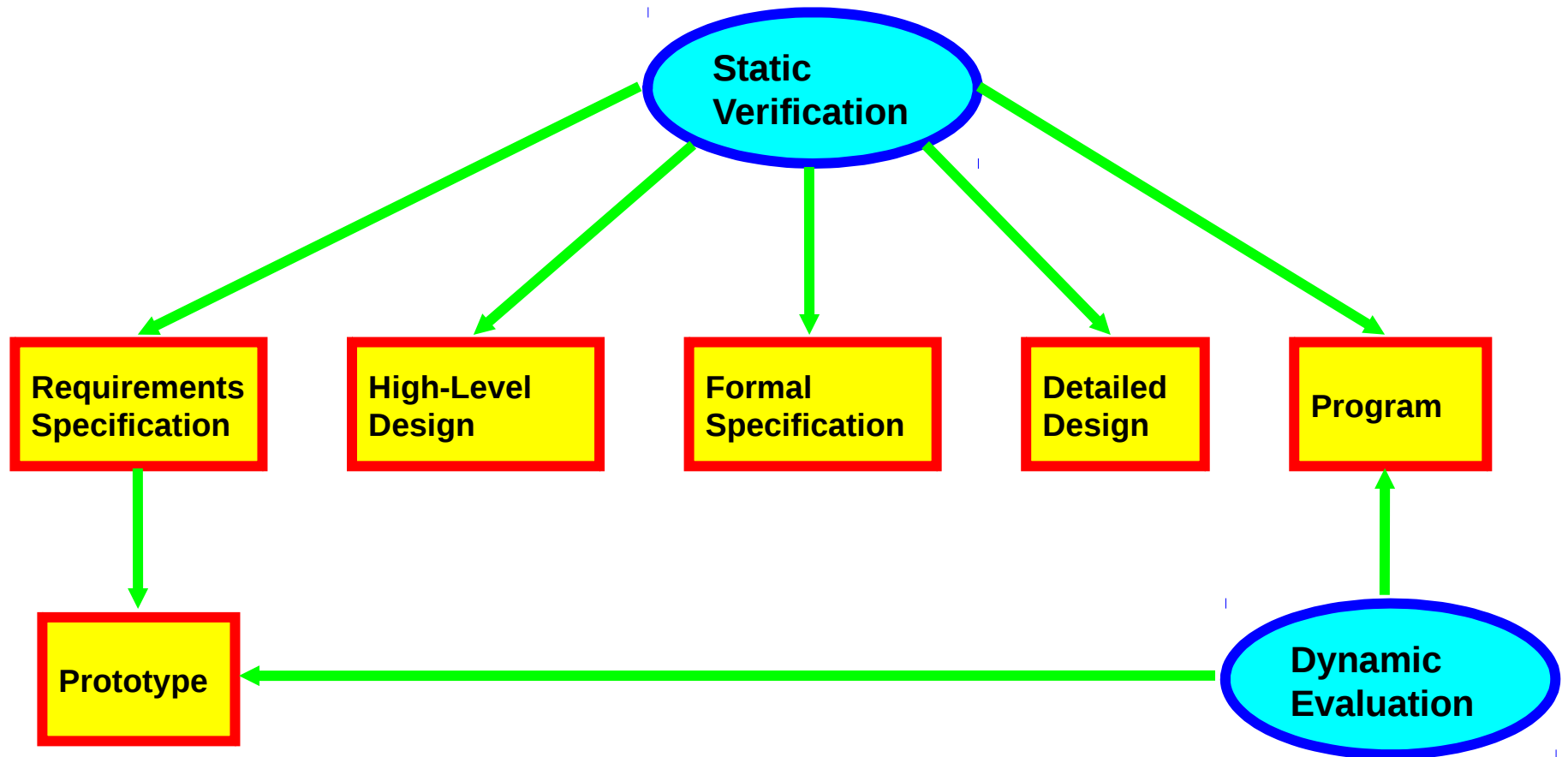
Verification: Are we building the product right?



Dynamic and Static Verification

- Dynamic V & V
 - Concerned with exercising and observing product behavior
 - Testing
- Static V & V
 - Concerned with analysis of the static system representation to discover problems
 - Proofs
 - Inspections

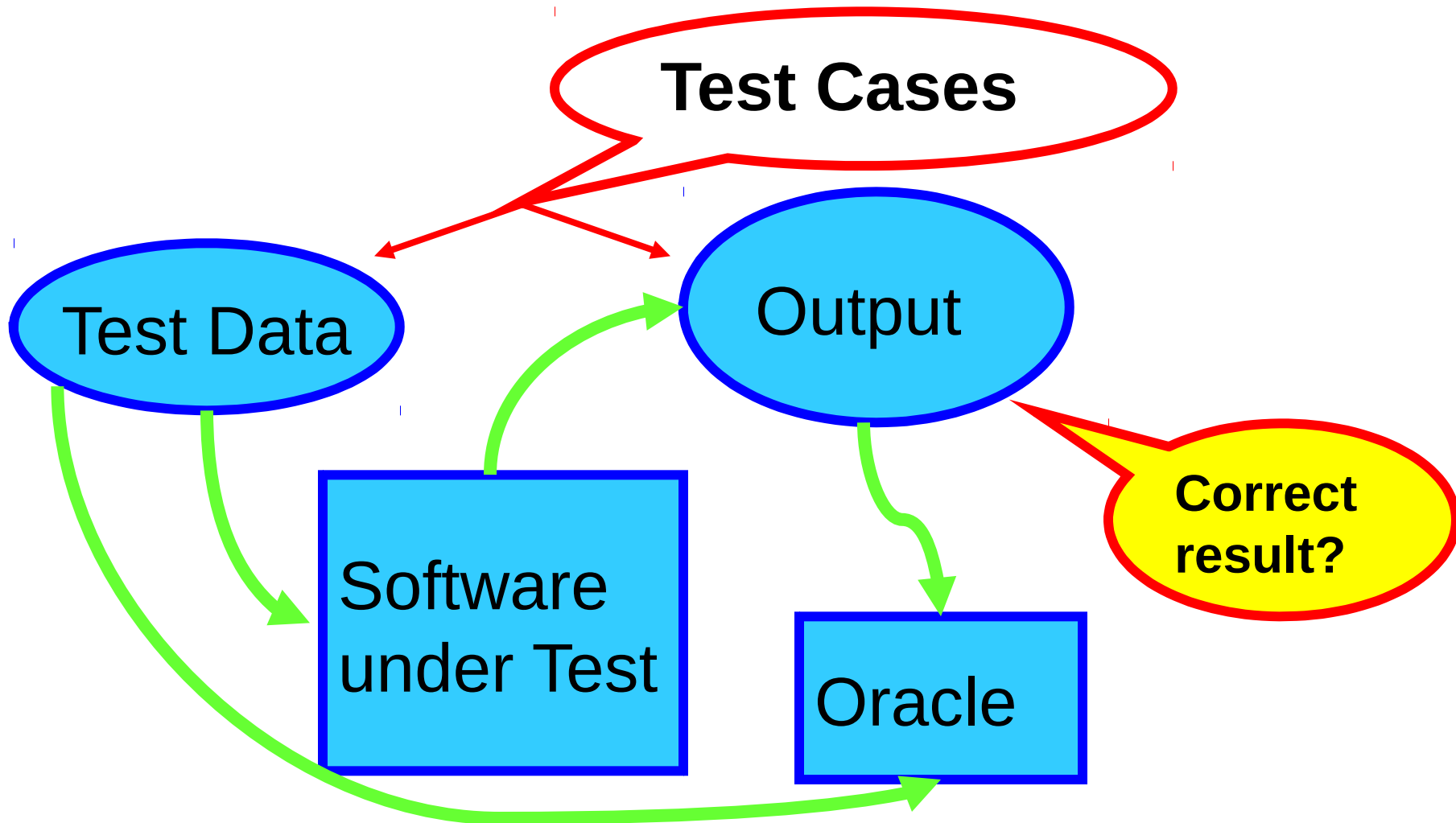
Static and Dynamic V&V



Definitions of Testing

- The process of executing a program (or part of a program) with the intention of finding errors (Myers, via Humphrey)
- The purpose of testing is to find errors
 - Testing is the process of trying to discover every conceivable fault or weakness in a work product (Myers, via Kit)
- The process of searching for errors (Kaner)

What is a Test?



Test Data and Test Cases

- Test data
 - Inputs which have been devised to test the system
- Test cases
 - Inputs to test the system and the predicted outputs from these inputs if the system operates according to its specification

Bugs? What is That?

- Failure

- An execution that yields an erroneous result

- Fault

- The source of the failure

- Error

- The mistake that led to the fault being introduced in the code

Axiom of Testing

“Program testing can be used to show the presence of bugs, but never their absence.”

- Dijkstra, 1969

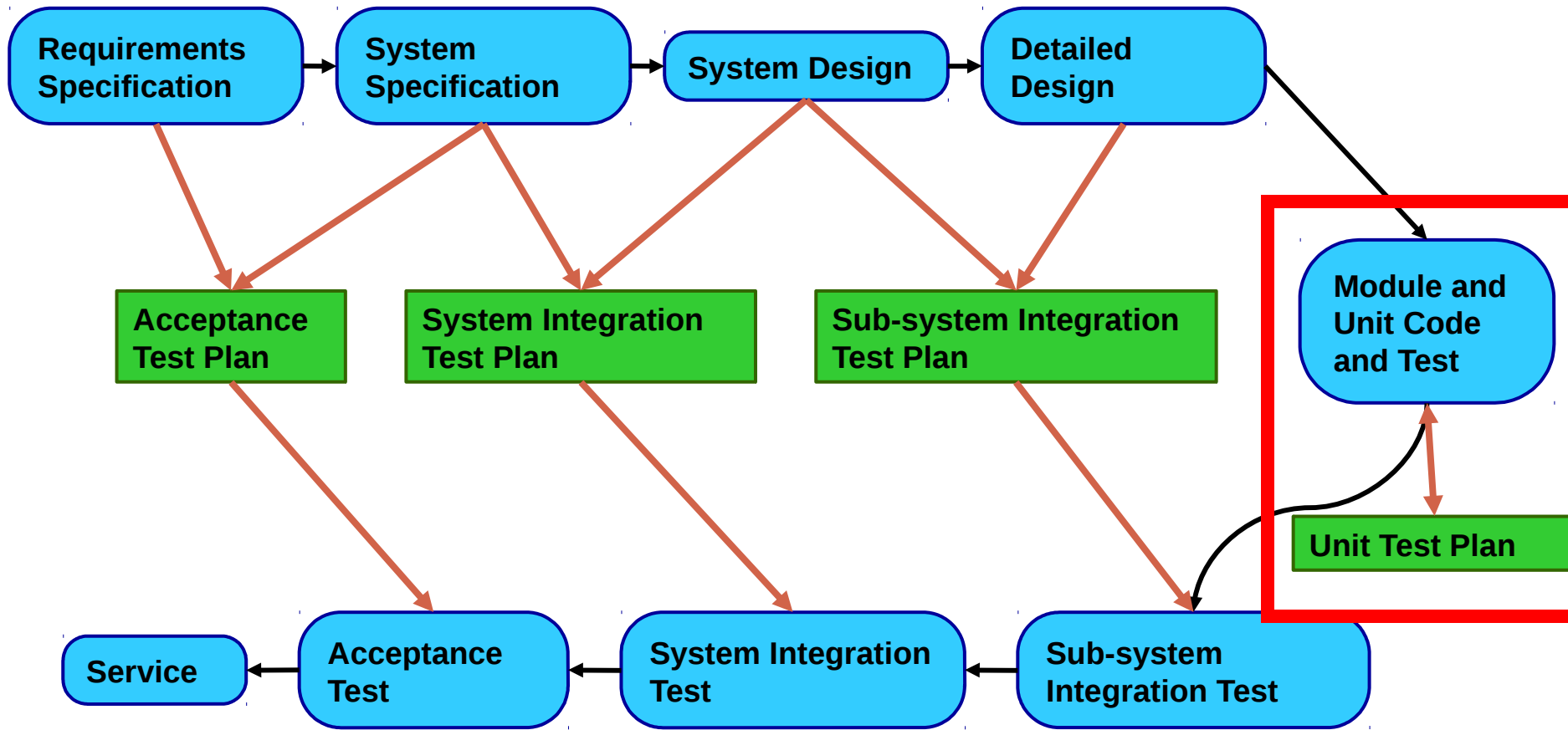
Black and White Box

- Black box testing:
 - Designed without knowledge of the program's internal structure and design
 - Based on functional requirements
- White box testing:
 - Examines the internal design of the program
 - Requires detailed knowledge of its structure

The V & V Process

- Is a whole life-cycle process
 - V & V must be applied at each stage in the software process
- Has two principal objectives
 - The discovery of defects in a system
 - The assessment of whether or not the system is usable in an operational situation

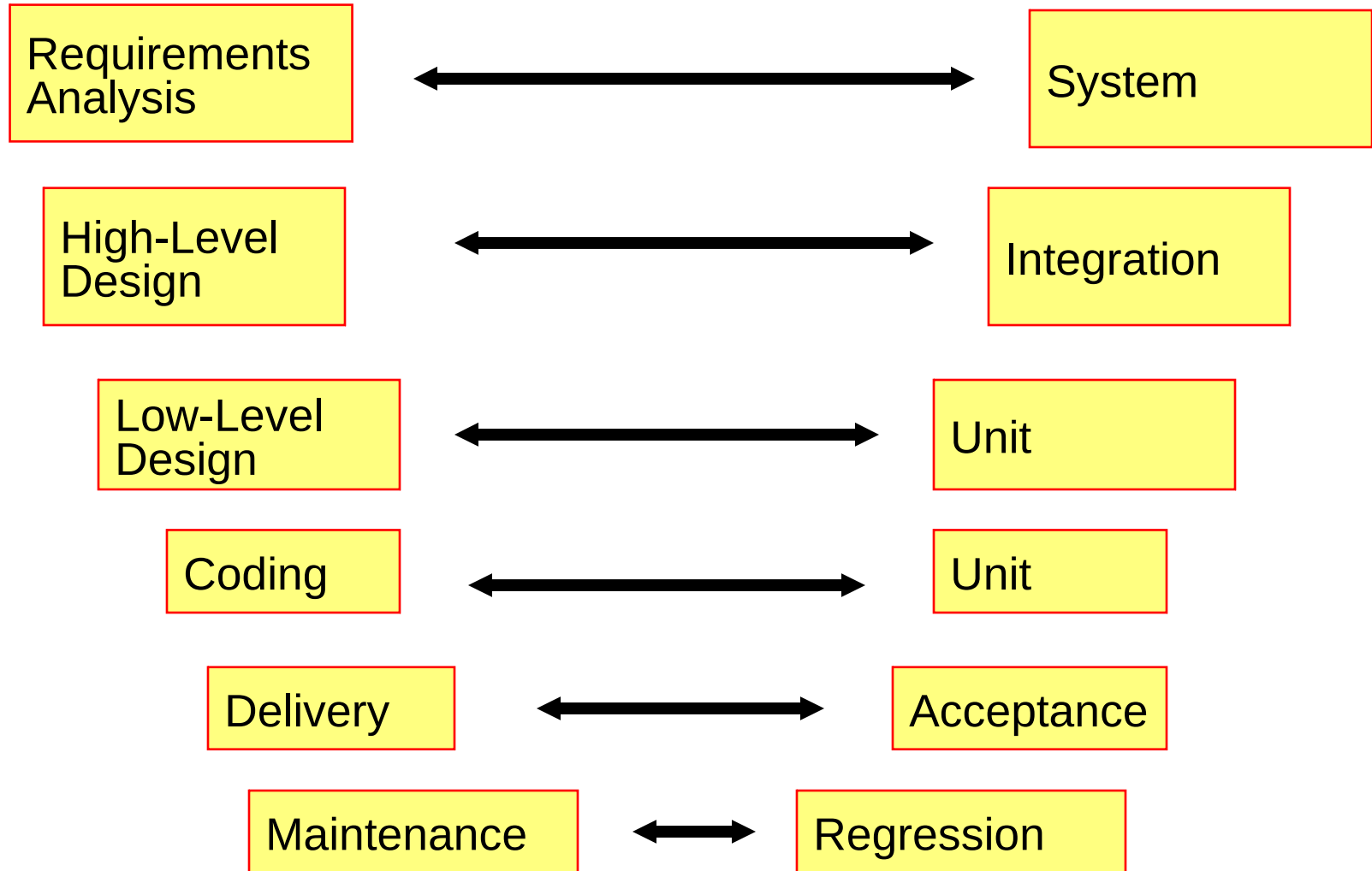
The V-Model of Development



Testing Stages

- Unit testing
 - Testing of individual components
- Module testing
 - Testing of collections of dependent components
- Sub-system testing
 - Testing collections of modules integrated into sub-systems
- System testing
 - Testing the complete system prior to delivery
- Acceptance testing
 - Testing by users to check that the system satisfies requirements
 - Sometimes called alpha and beta testing

V Graph



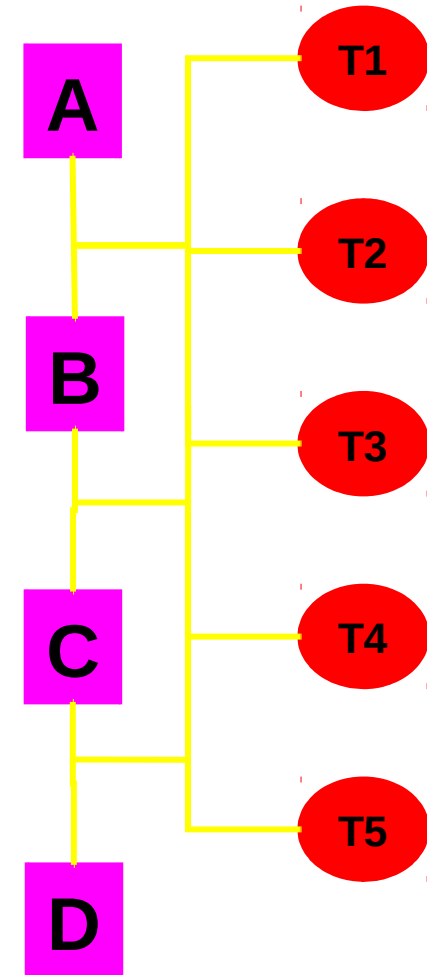
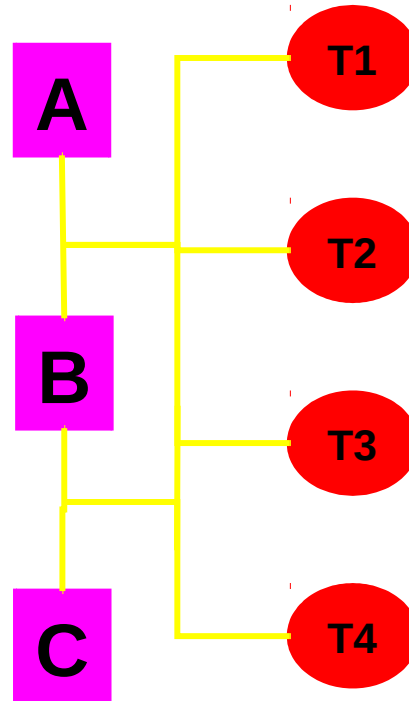
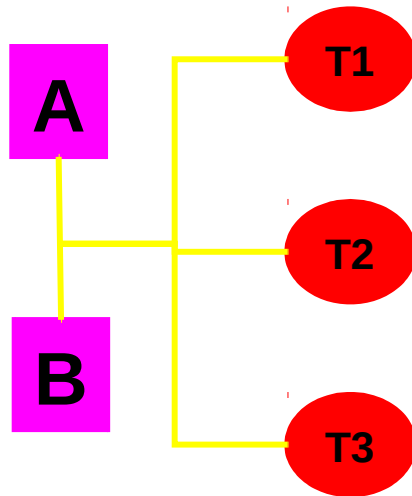
Types of Testing

- Statistical testing
 - Tests designed to reflect the frequency of user inputs
 - Used for reliability estimation
- Defect testing
 - Tests designed to discover system defects
 - A successful defect test is one which reveals the presence of defects in a system

Testing Strategies

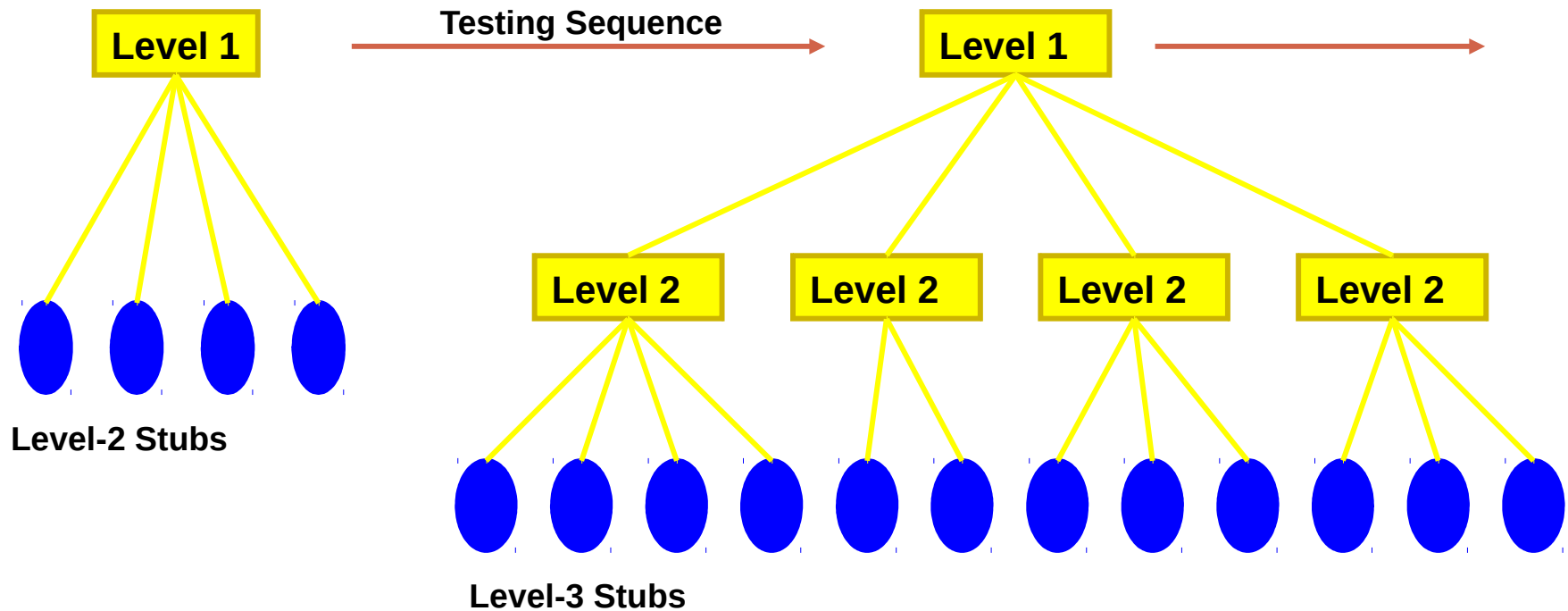
- Testing strategies are ways of approaching the testing process
- Different strategies may be applied at different stages of the testing process
- Strategies covered
 - Top-down testing
 - Bottom-up testing
 - Back-to-back testing

Incremental Testing



An integration testing strategy in which you test subsystems in isolation, and then continue testing as you integrate more and more subsystems

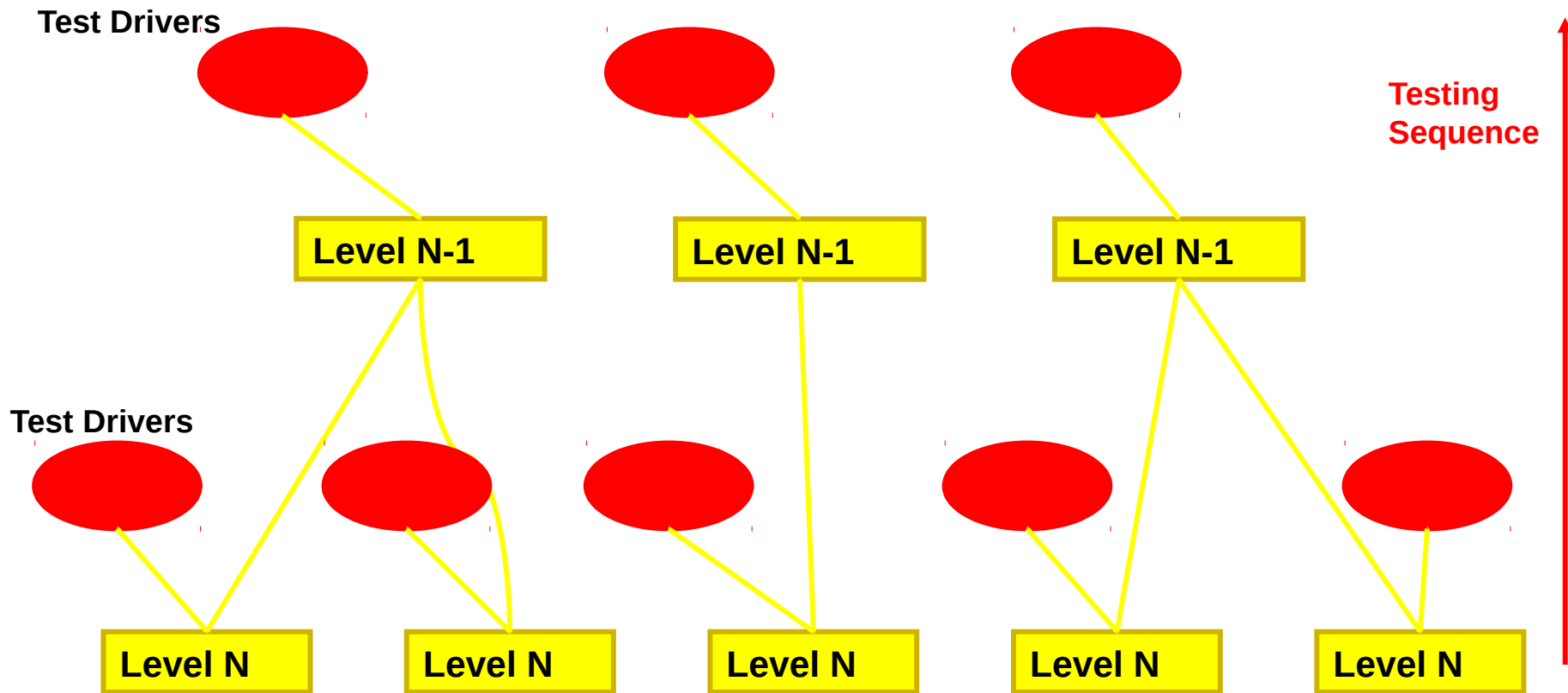
Top-down testing



Top-Down Testing

- Start with the high-levels of a system and work your way downwards
- Testing strategy which is used in conjunction with top-down development
- Finds architectural errors
- May need system infrastructure before any testing is possible
- May be difficult to develop program stubs

Bottom-Up Testing



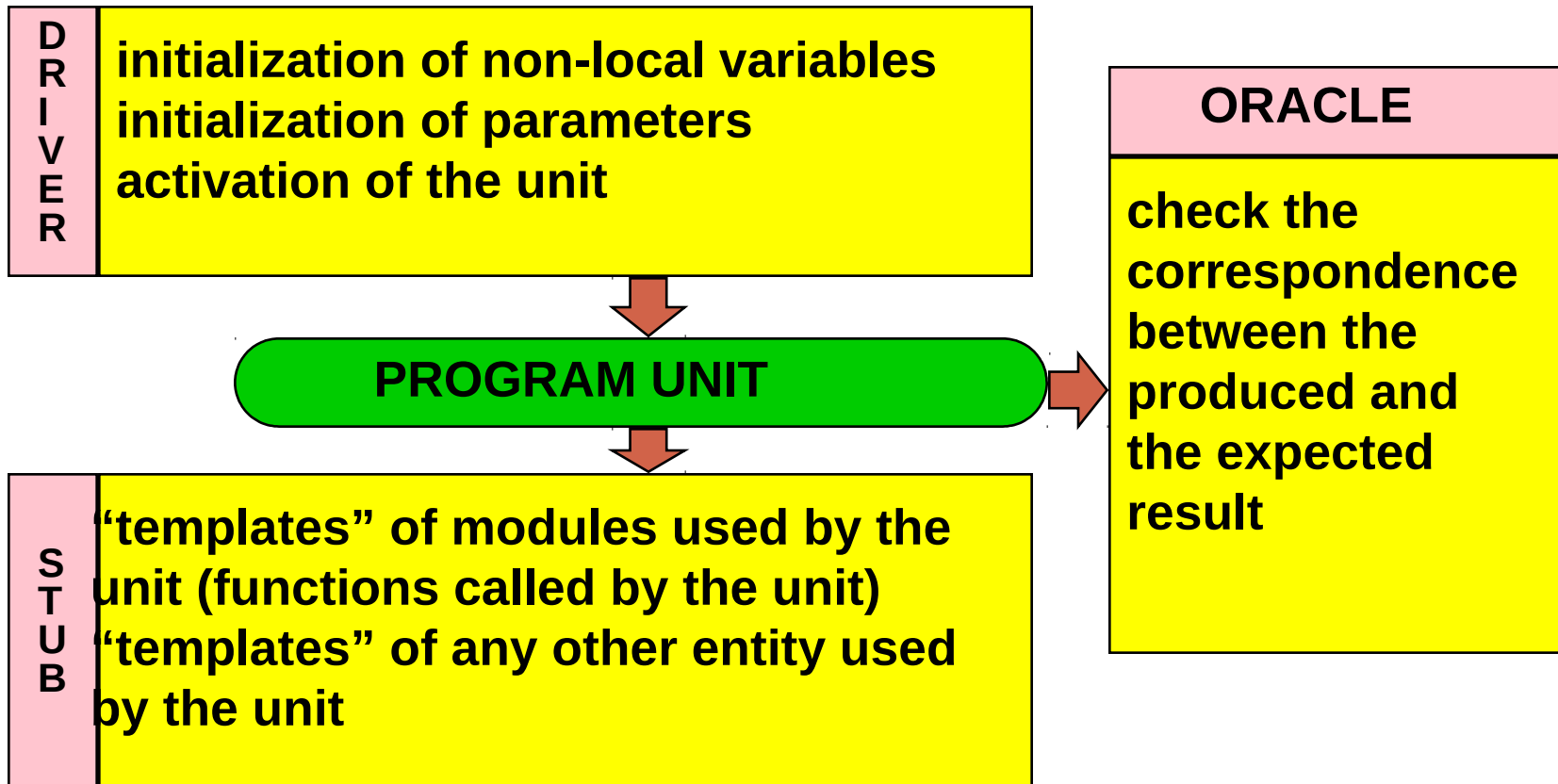
Bottom-Up Testing

- Necessary for critical infrastructure components
- Start with the lower levels of the system and work upward
- Needs test drivers to be implemented
- Does not find major design problems until late in the process
- Appropriate for object-oriented systems

Create Scaffolding

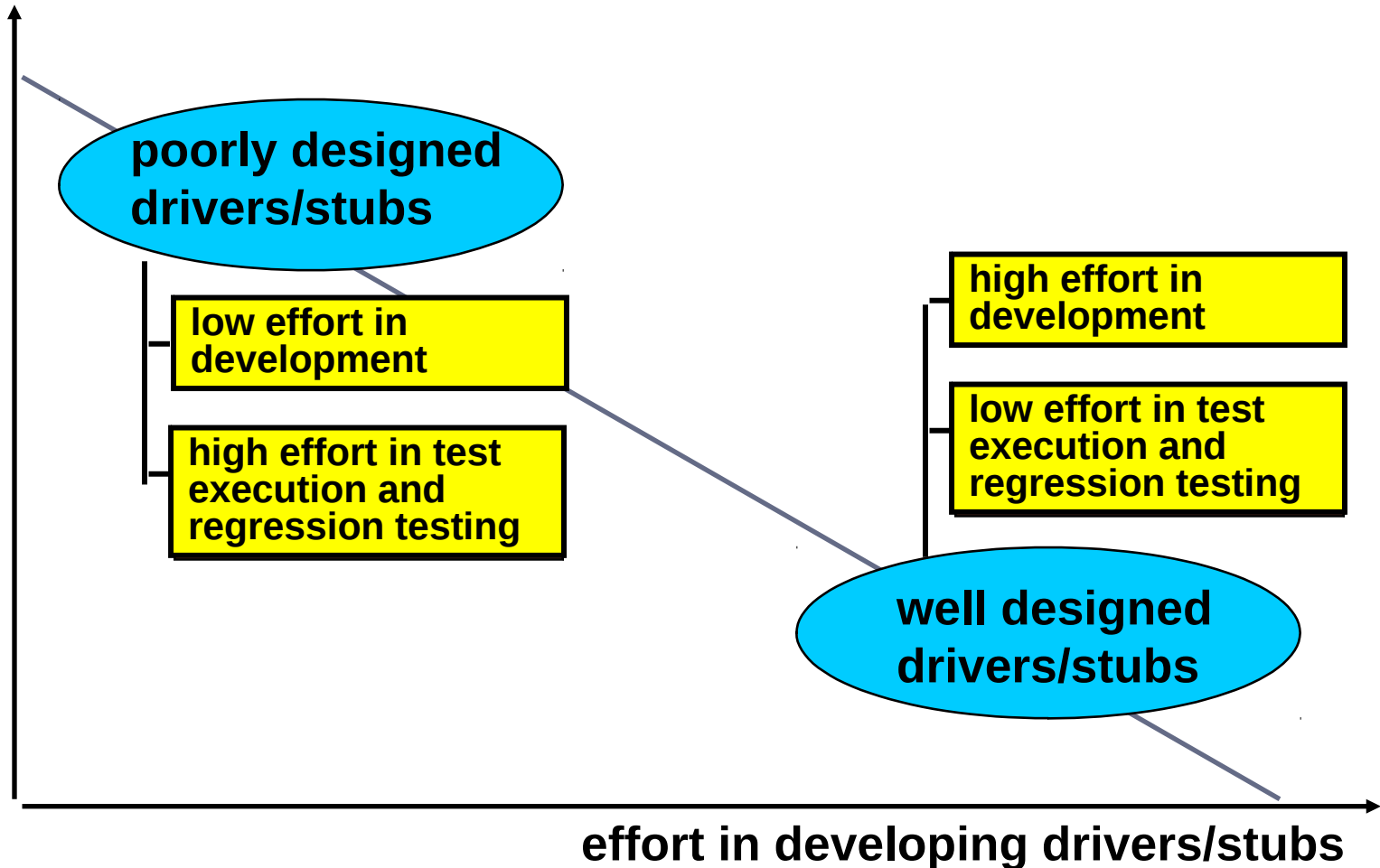
Goal

To setup the environment for executing the test

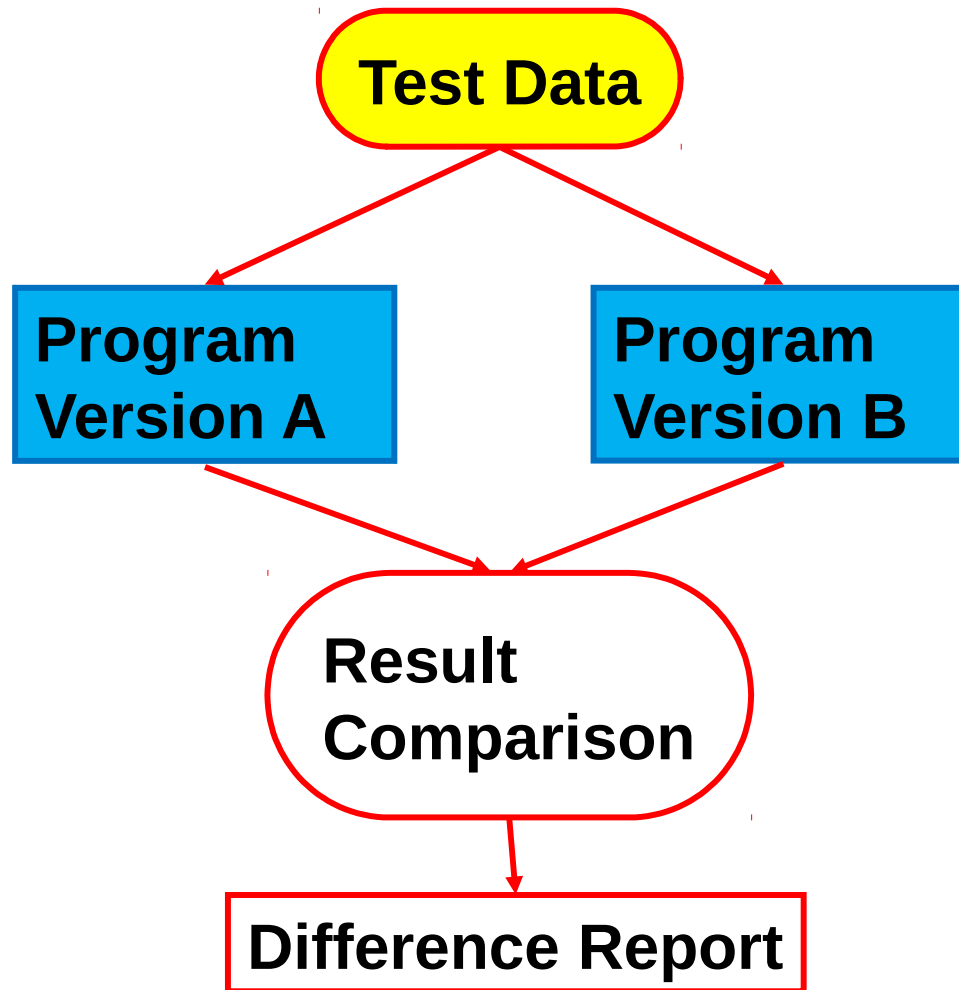


Problems and Tradeoffs

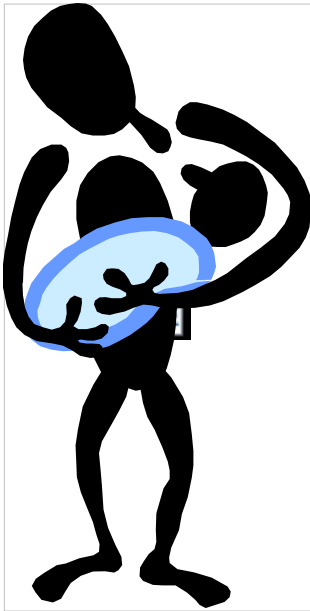
effort in test execution and regression testing



Back-to-Back Testing



Who Should Test?



■ Developer

- Understands the system
- But, will test gently
- And, is driven by deadlines



■ Independent tester

- Must learn system
- But, will attempt to break it
- And, is driven by “quality”

Axioms of Testing

- “As the number of detected defects in a piece of software increases, the probability of the existence of more undetected defects also increases”
- “Assign your best programmers to testing”
- “Exhaustive testing is impossible”

Axioms of Testing

- “You cannot test a program completely”
- “Even if you do find the last bug, you’ll never know it”
- “It takes more time than you have to test less than you’d like”
- “You will run out of time before you run out of test cases”

We Have Learned

- Test definitions and language
- Testing activities include unit testing, module testing, sub-system testing, integration testing and acceptance testing
- Testing should be scheduled as part of the planning process
 - Adequate resources must be made available
- Testing strategies include top-down testing, bottom-up testing, and back-to-back testing
- Some axioms about testing