

16.35

Aerospace Software Engineering

More on Testing

Prof. Kristina Lundqvist
Dept. of Aero/Astro, MIT

Test Techniques

- **Static** analysis techniques
 - analyses source code to detect and identify errors
- **Dynamic** analysis techniques
 - subjects software to real world conditions in an instrumented environment that detects errors and helps track down the cause of errors

Today

- Manual test techniques
 - Reading
 - Peer review
 - Egoless programming
 - Walkthroughs and Inspections
- Unit testing
- Integration testing
- System testing

FTR (Formal Technical Reviews)

Manual Test Techniques

- Reading
- Peer review
 - Anonymously assessing programs
 - Hand in 'best' program and one of lesser quality
- Egoless programming

FTR (Formal Technical Reviews)

Objectives of Formal Reviews

- Uncover errors in any representation of software
 - Most reviewed application programs run correctly first time
- Verify that
 - software meets its requirements
 - software follows predefined standards
 - software is developed in uniform manner
- Make projects more manageable
- Educate new team members

What are Software Reviews?



Everyone knew exactly what had to be done until someone wrote it down!

“Walkthroughs”

developer technique

used by development teams to improve quality of product

focus on finding defects

“(Fagan) Inspections”

a process management tool

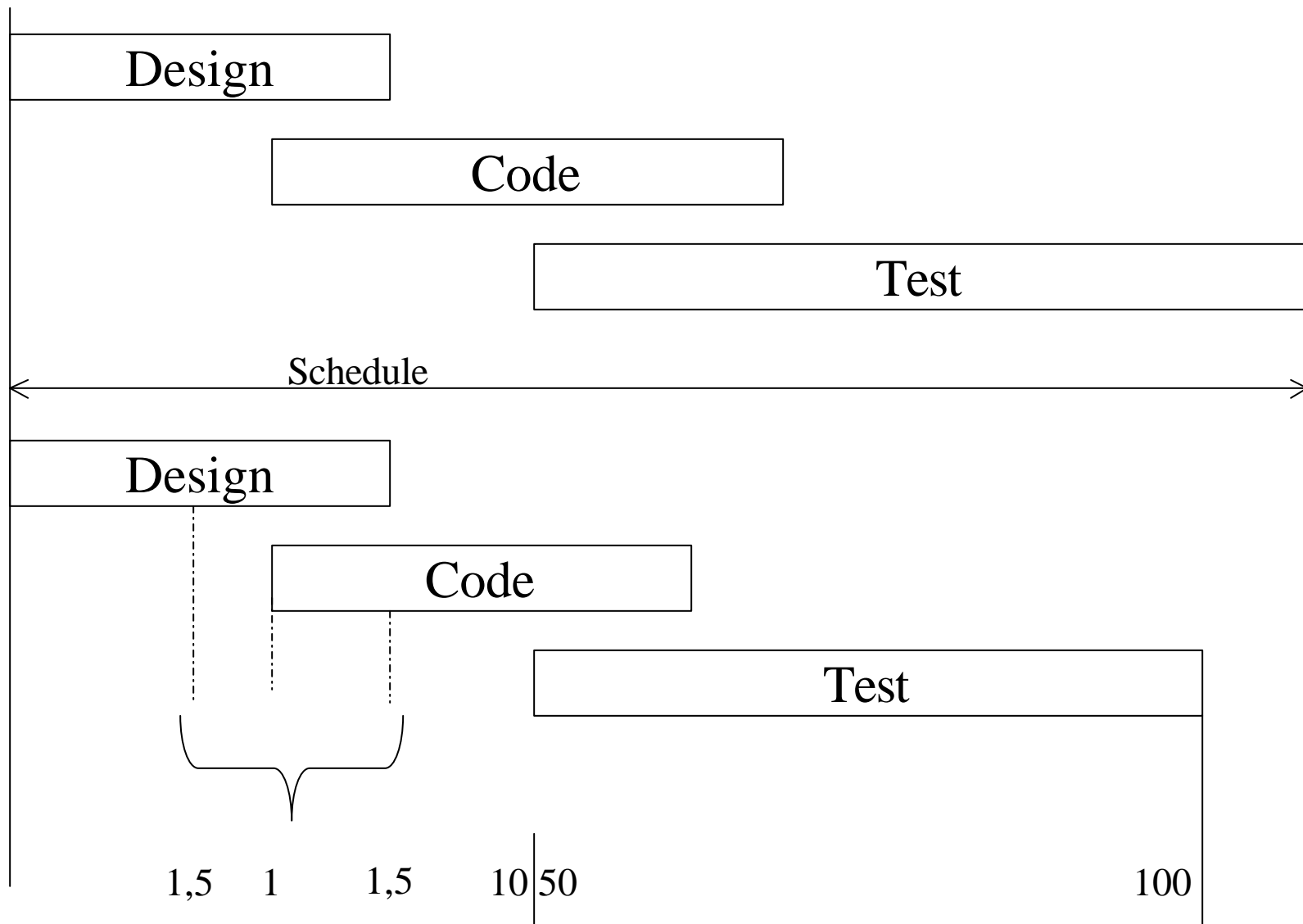
used to improve quality of the development process

collect defect data to analyze the quality of the process

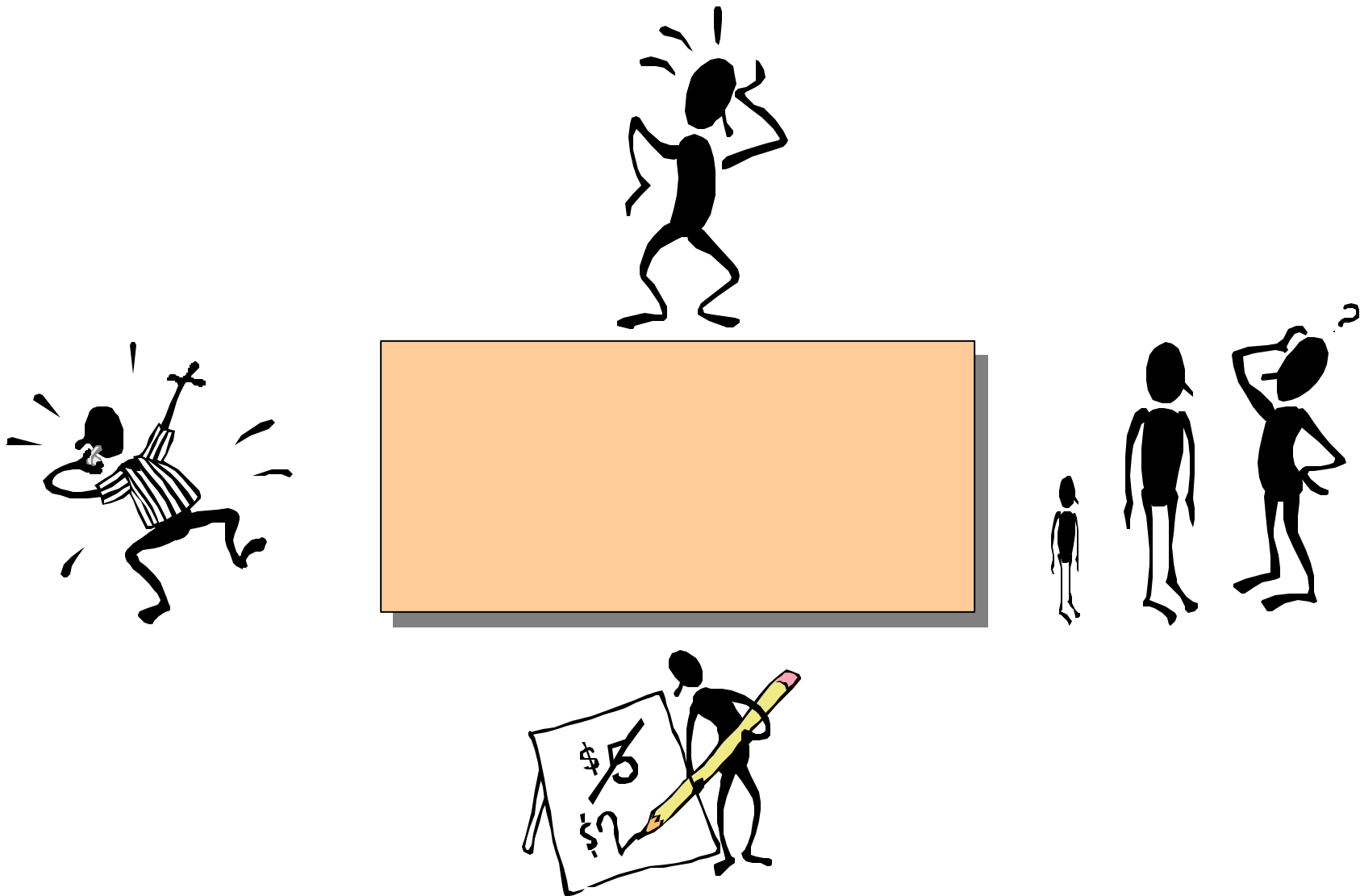
written output is important

major role in training junior staff and transferring expertise

Effect of Inspection



How to carry out reviews: The review team



Roles

Formal Walkthrough

- **Review Leader**
 - Chairs meeting
 - Ensures preparation is done
 - Keeps review focused
 - Reports the result
- **Recorder**
 - Keeps track of issues raised
- **Reader**
 - Summarizes the product piece by piece during the review
- **Author**
 - Should actively participate (may be the reader)
- **Other reviewers**
 - Task is to find and report issues

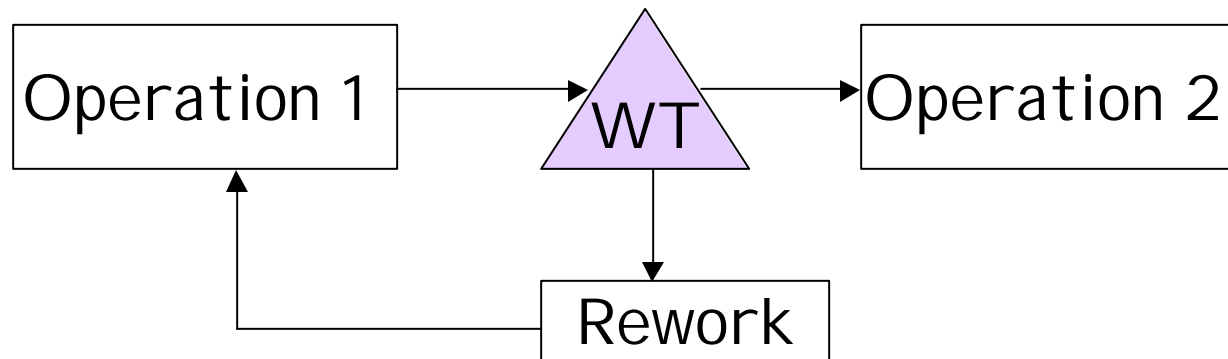
Fagan Inspection

- **Moderator**
 - Must be a competent programmer
 - Should be specially trained
 - Could be from another project
- **Designer**
 - Programmer who produced the design being inspected
- **Coder/implementor**
 - Programmer responsible for translating the design to code
- **Tester**
 - Person responsible for writing/executing test cases

Basic Guidelines

- 3-6 people (typical)
 - experienced senior technical staff
 - representatives of
 - team that created the document
 - client representative
 - team for next development phase
 - software quality assurance group
- IEEE Standard for Software Reviews and Audits [IEEE 1028, 1988]

Walkthrough Process

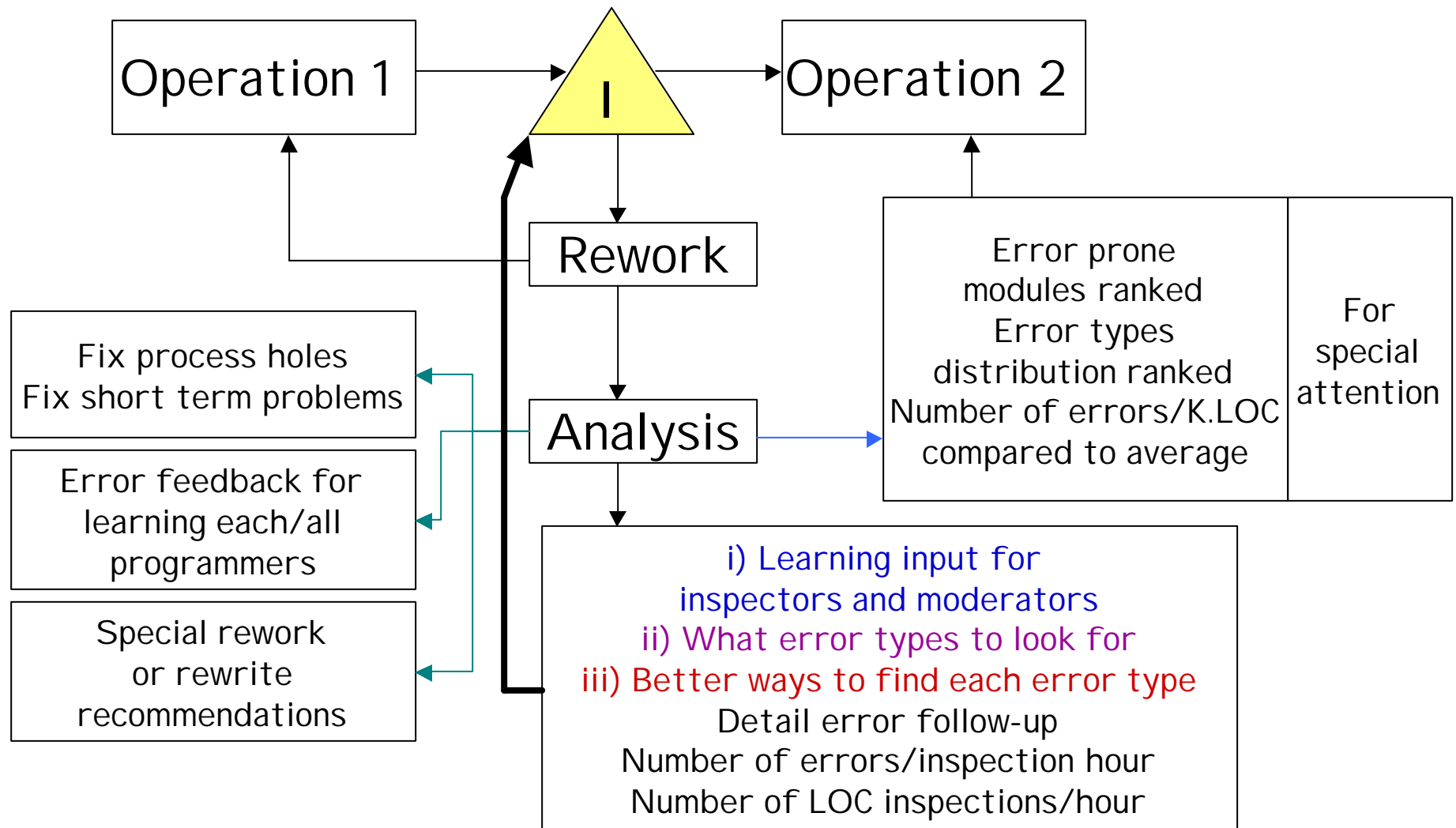


Result: one-time improvement due to error removal in proportion to error detection efficiency of walk through

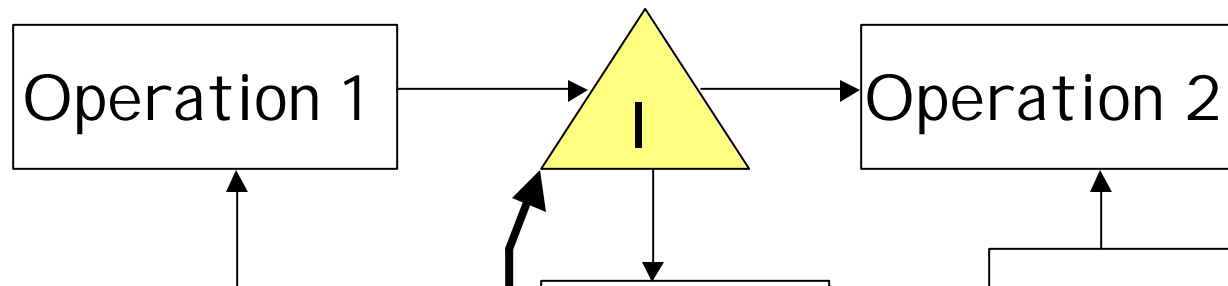
Code Walkthrough Guidelines

1. Have the review meeting chaired by the project manager or chief programmer, who is also responsible for scheduling the meeting, reserving a room, setting the agenda, inviting participants, and so on.
2. The programmer presents his or her work to the reviewers. Discussion should be general during the presentation.
3. Following the general discussion, the programmer walks through the code in detail, focusing on the logic of the code rather than on specific test cases.
4. Reviewers ask to walk through specific test cases.
5. The chair resolves disagreements if the review team cannot reach agreement among themselves and assigns duties, usually to the programmer, for making specific changes.
6. A second walkthrough is then scheduled if needed.

Inspection Process



Inspection Process



Result: one time improvement + iterative improvement due to improvements in *Operation 1*, *I*, and *Operation 2* enabled by analyzed feed-back/forward + error detection efficiency improved for reasons i-iii

Learning each/ all programmers

Special rework or rewrite recommendations

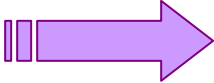
- i) Learning input for inspectors and moderators
- ii) What error types to look for
- iii) Better ways to find each error type

Detail error follow-up
Number of errors/inspection hour
Number of LOC inspections/hour

Inspection Guidelines

- Review leader should be SQA representative
 - has the most to lose
 - creator: eager to get approval (to start next job)
 - client: can wait for acceptance testing
- Review leader distributes material
- Advance preparation of max. 2 hours before the meeting
- Duration: less than 2 hours

Result of FTR

- Decision about the product
 - **accept** without further modification
 - **reject** the work due to severe errors (review must be repeated)
 - **accept with minor modifications** (that can be incorporated into the document by the producer)
- **All participants have to sign-off**
 - shows participation  responsibility
 - shows their concurrence with the findings

Reviewer's Preparation

- be sure that you understand the context
- first, skim all the product material to understand location and format of the information
- next, read product material and annotate hardcopy
- pose your *written comments* as questions
- avoid issues of style
- inform the review leader if you can't prepare

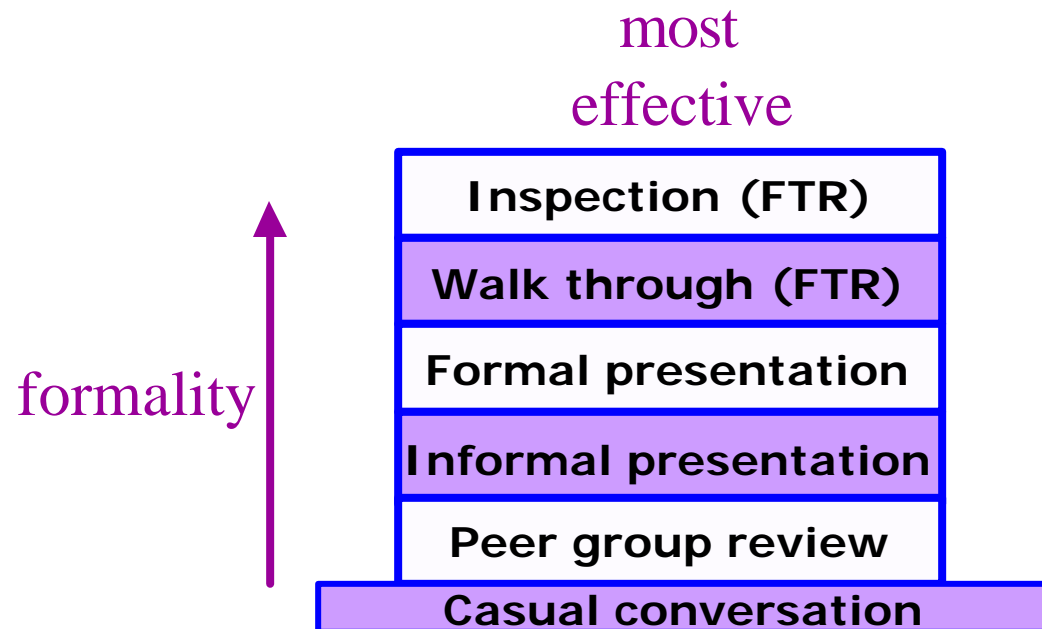
Conducting the Review

- Be prepared - evaluate product before review
- develop check list for each kind of work product
- review the product, not the producer
- keep your tone mild, ask questions instead of making accusations
- stick to the review agenda
- raise issues, don't resolve them!
- limit discussions (do them off line!)
- avoid discussions of style - stick to technical correctness
- schedule reviews as project tasks (allocate resources)
- record and report all review results

FTR Reporting

- Review summary report
 - What was reviewed?
 - Who reviewed it?
 - What were the findings and conclusions?
- Review issues list
 - Identify problem areas within the product
 - Serve as an action item checklist

Reviews: an Effectiveness Scale



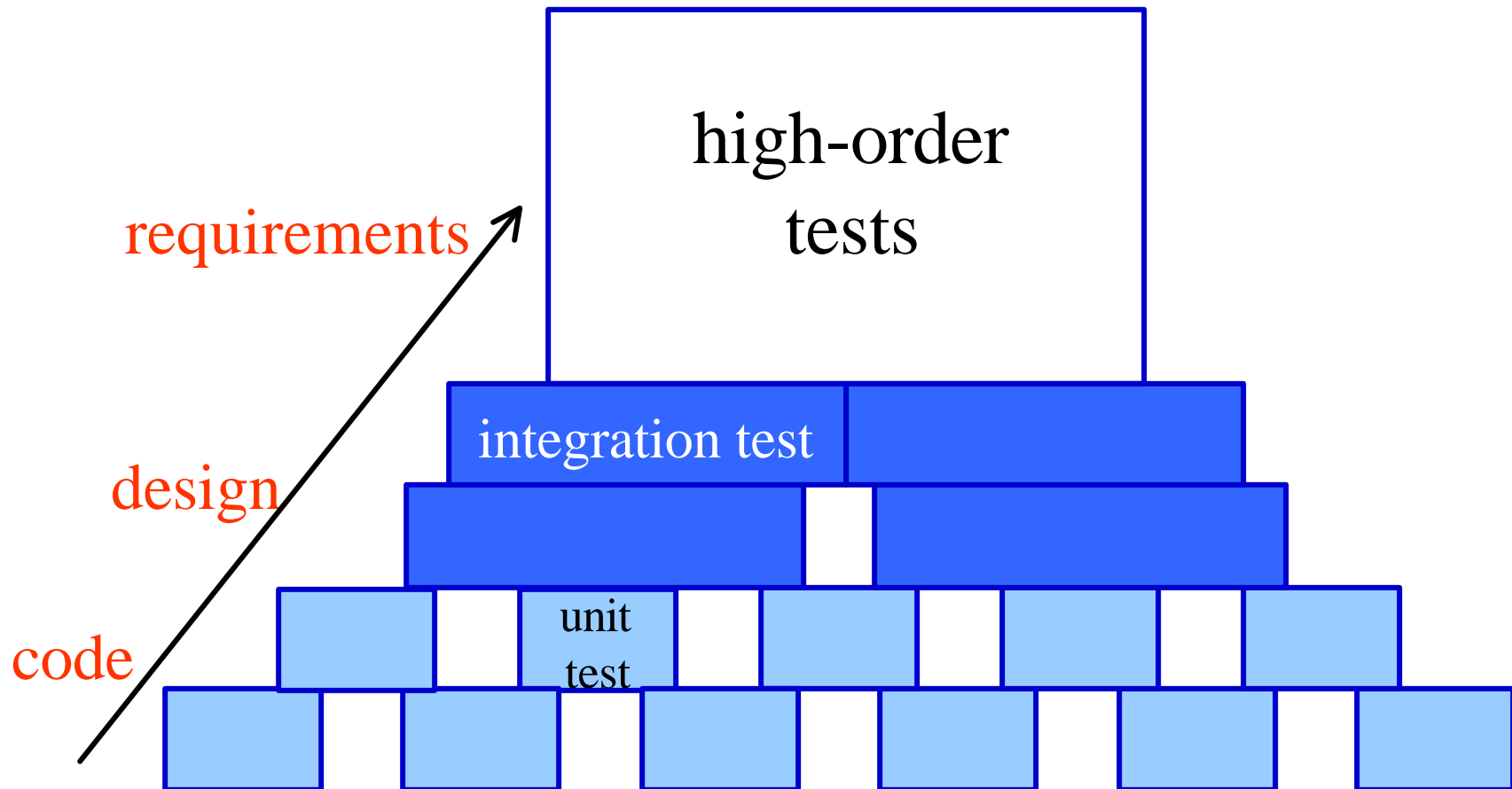
Effectiveness of inspections

- [Fagan 1976] inspections of design & code
 - 67%-82% of all faults were found by inspections
 - 25% time saved on programmer resources (despite inspections)
- [Fagan 1986]
 - 93% of all faults were found by inspections
- Cost reduction for fault detection (compared with testing)
 - [Ackerman, Buchwald, Lewski 1989]: 85%
 - [Fowler 1986]: 90%
 - [Bush 1990]: 25.000US\$ saved PER inspection

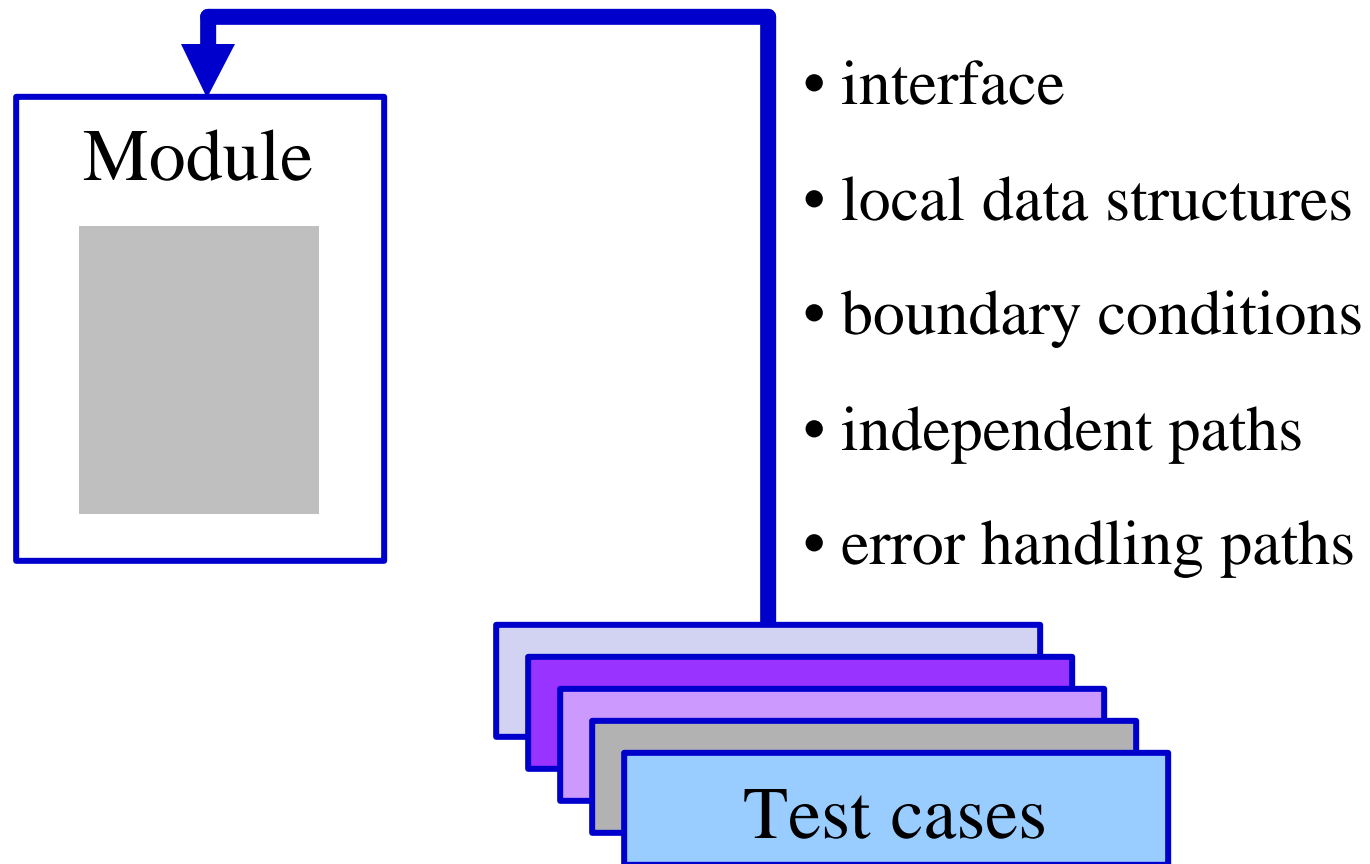
Today

- Manual test techniques
 - Reading
 - Peer review
 - Egoless programming
 - Walkthroughs and Inspections FTR (Formal Technical Reviews)
- Unit testing
- Integration testing
- System testing

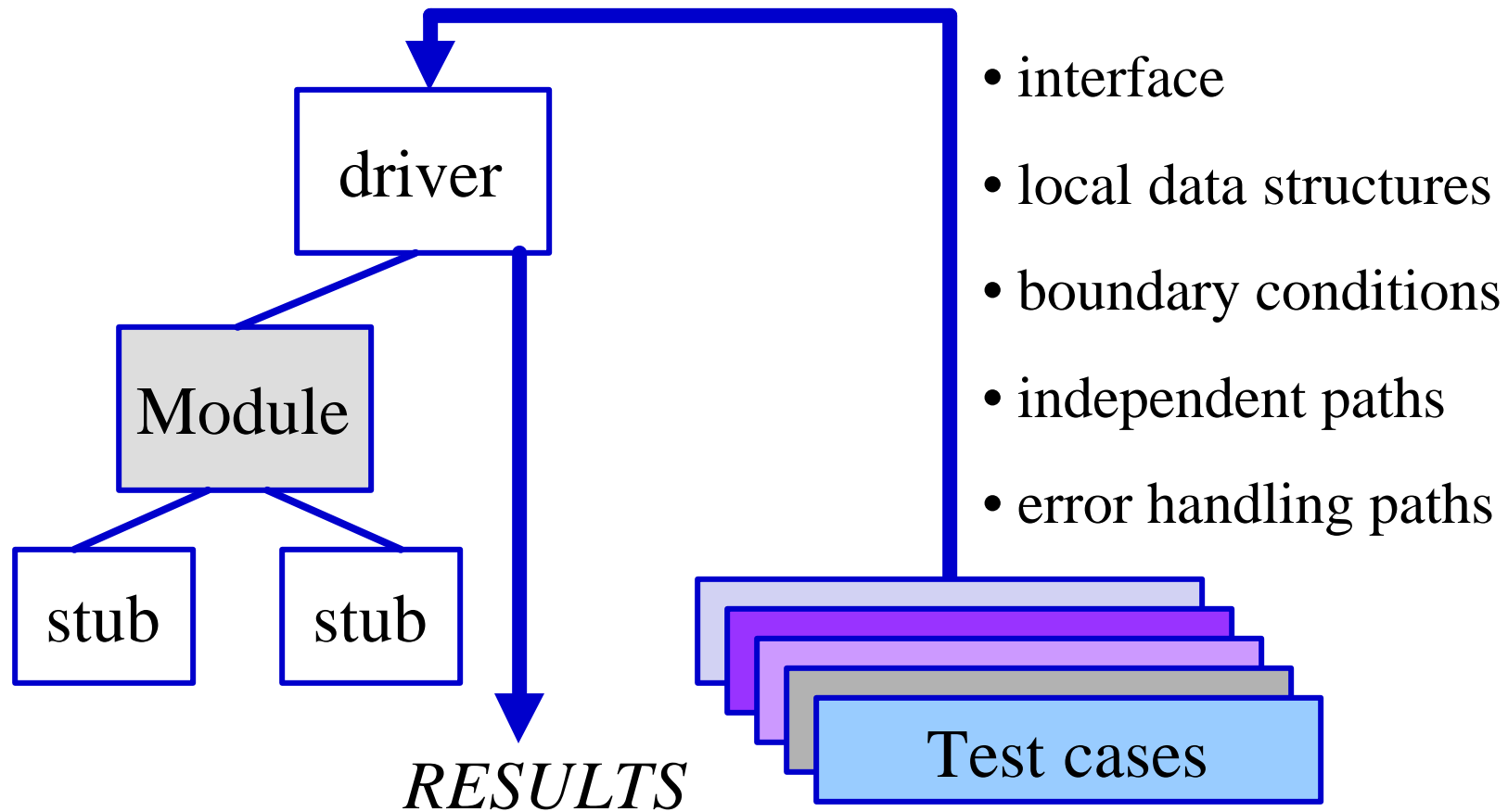
Testing strategy: hierarchy



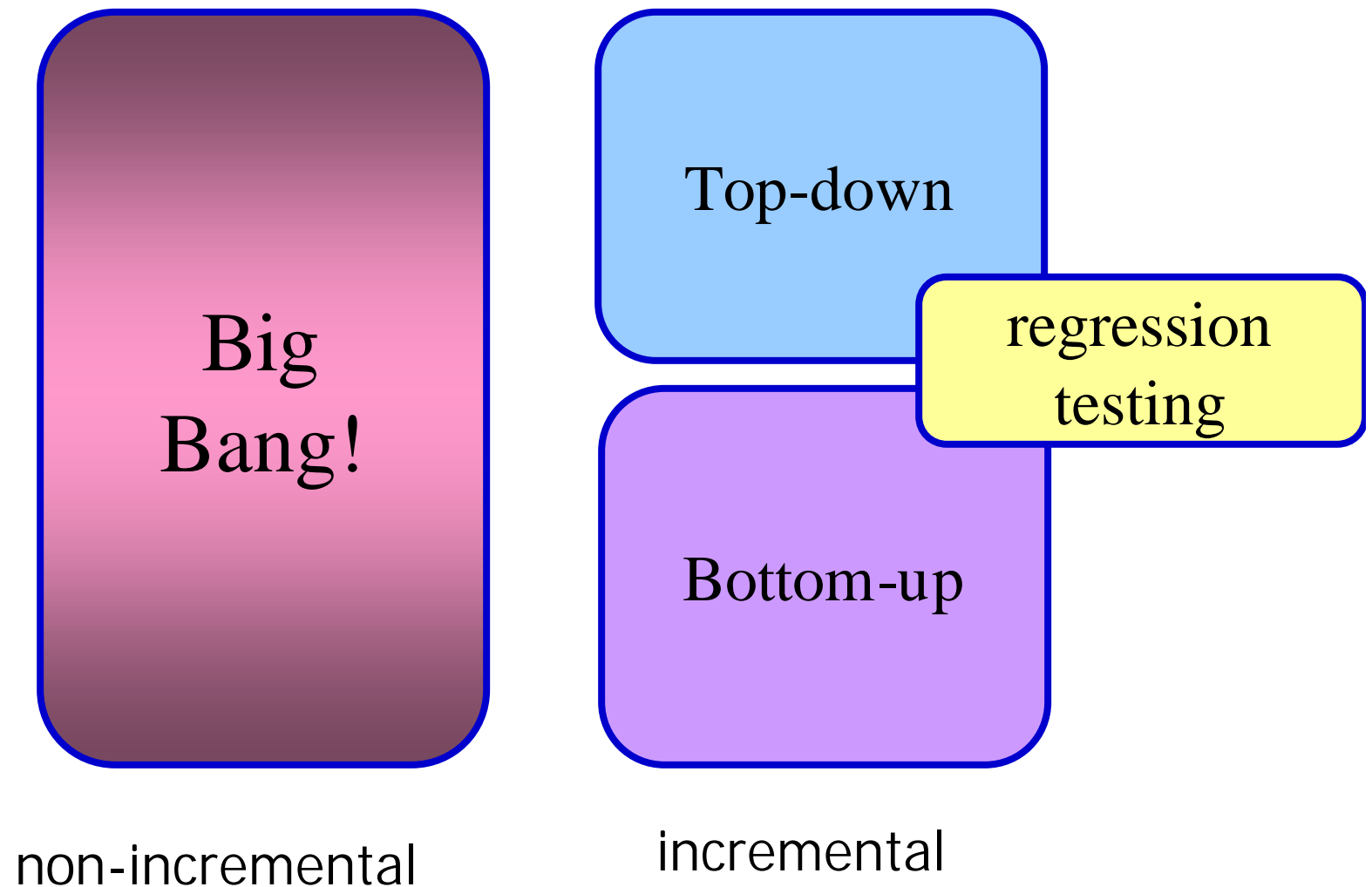
Unit Testing



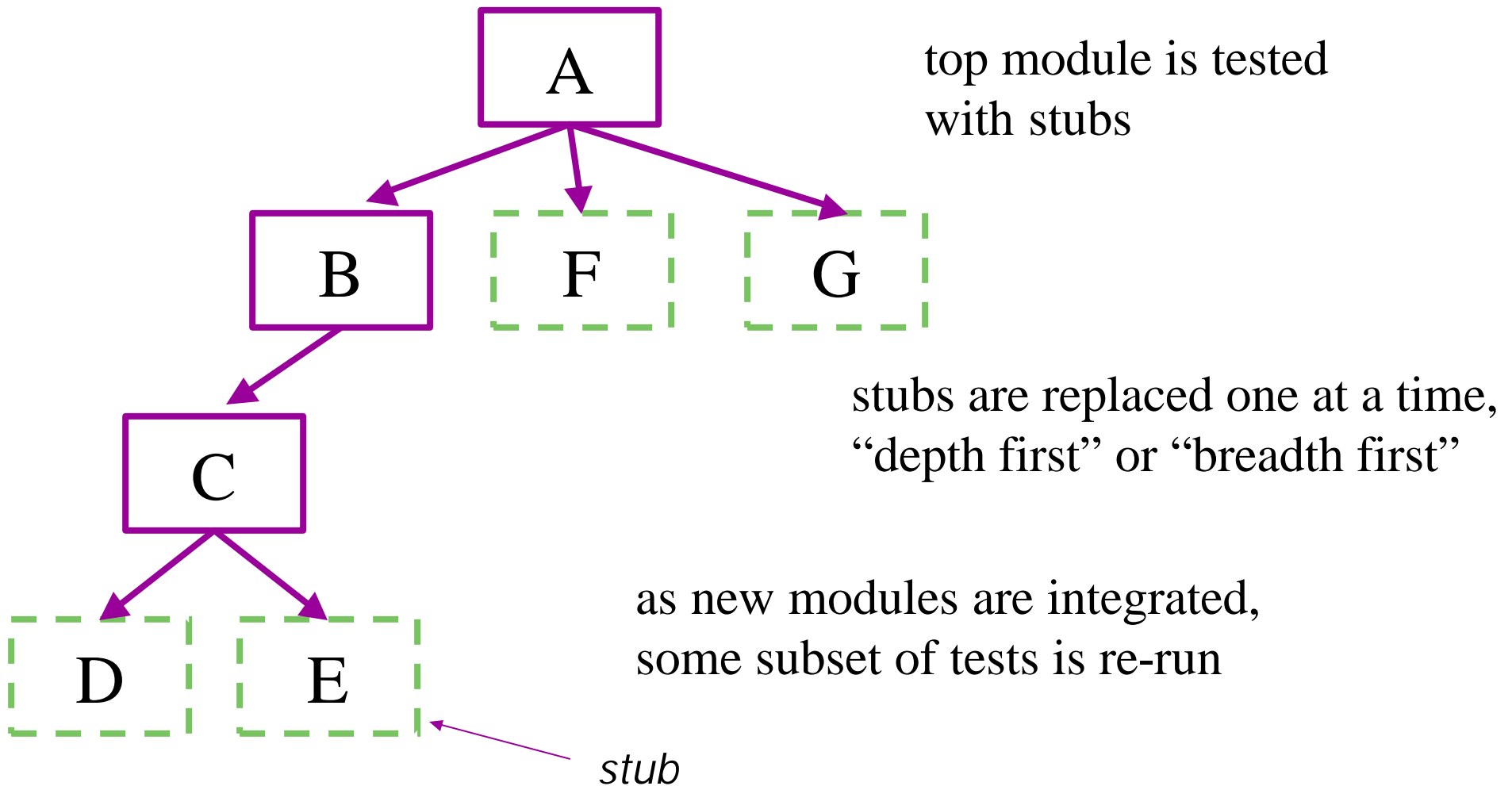
Unit Testing Environment



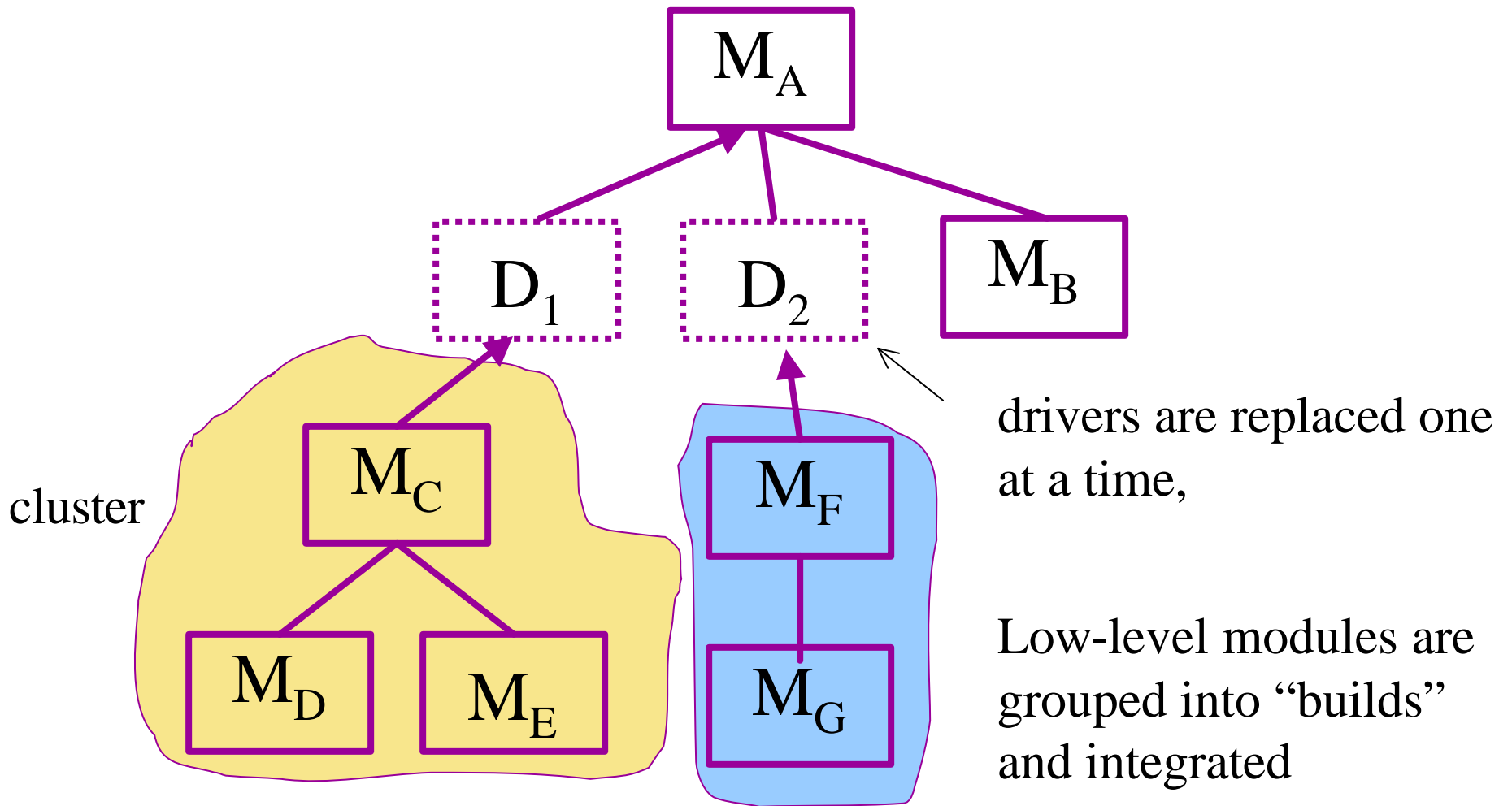
Integration Testing



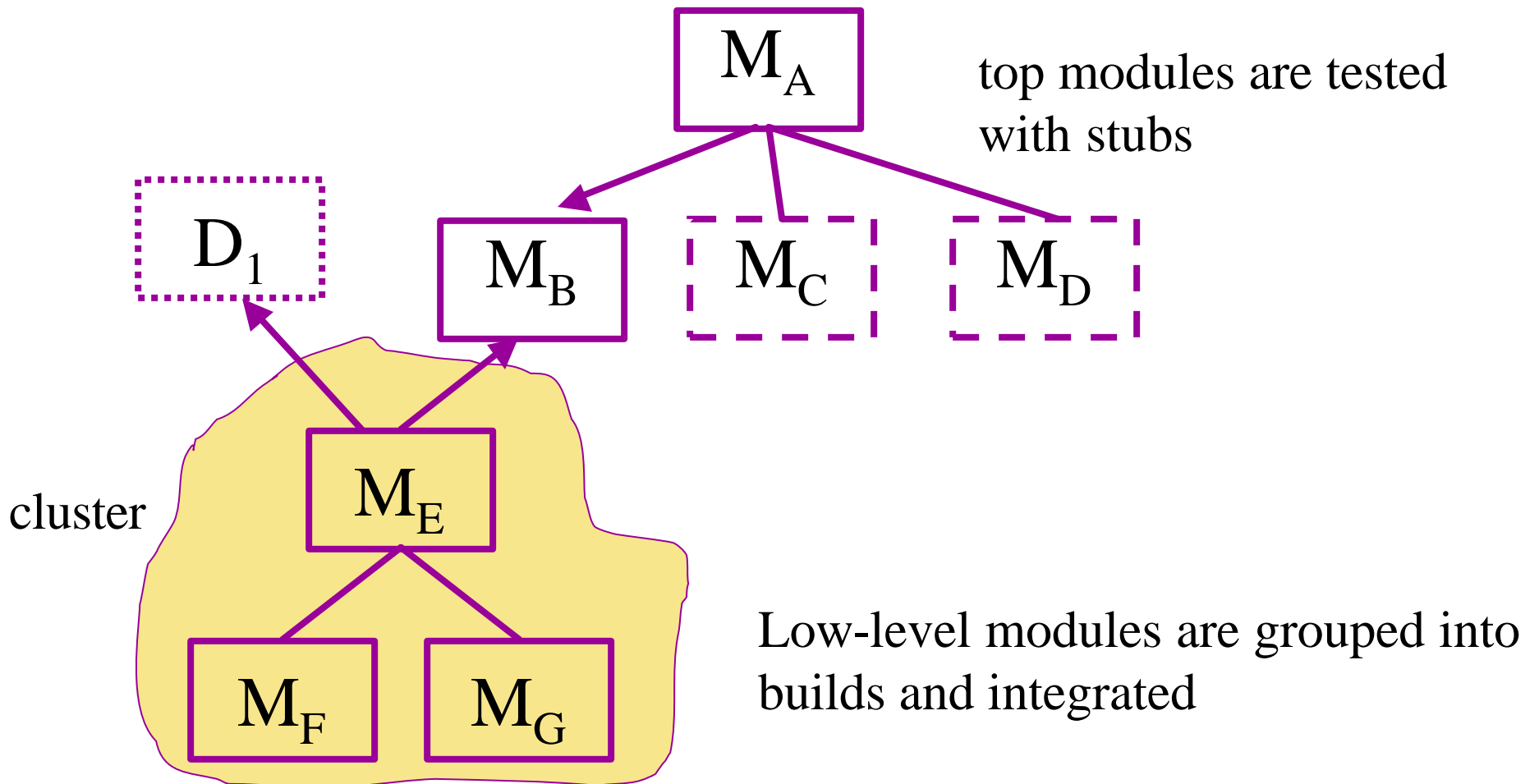
Top Down Integration



Bottom Up Integration



Sandwich Testing

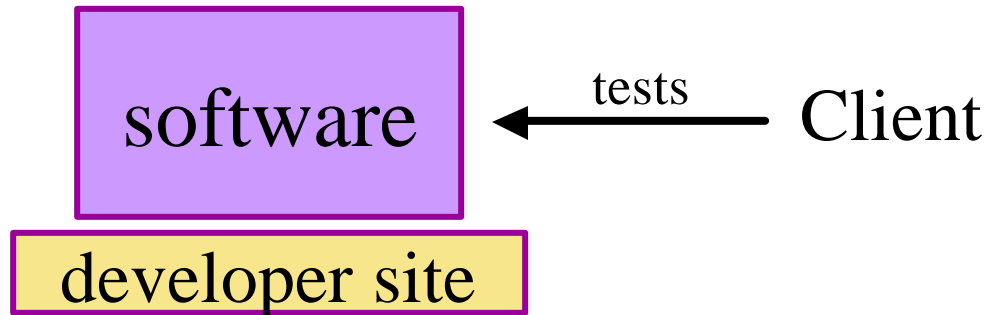


Higher Order Testing

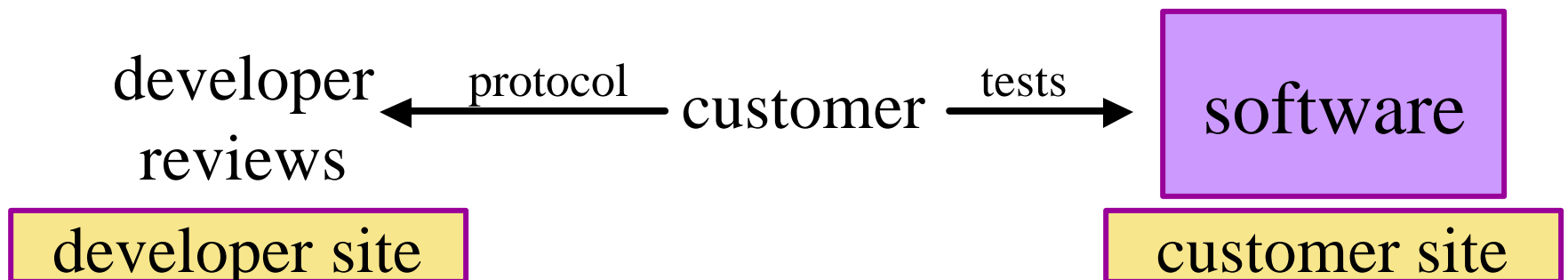
- Validation Testing
 - Alpha and Beta testing
- System Testing
- Performance Testing
- Security Testing

Alpha & Beta test

Alpha test



Beta test



Alpha Testing

- Performed by “special” users and the development organization
- Assess if the system meets the design requirements
- Make sure system is not overtly destructive to itself or environment
 - **recovery** testing -- force system to fail
 - **stress** testing -- exert excessive loads
 - **performance** testing -- determine performance
 - **security** testing -- test protection mechanisms

Beta Testing

- performed by subset of real users
- real environment and real data
- test system support mechanisms
 - system help
 - documentation
 - training
 - technical support

Acceptance Testing

The process whereby actual users test a completed software, the end result of which is the users' acceptance of the system.