

# CSC444 - Software Engineering I

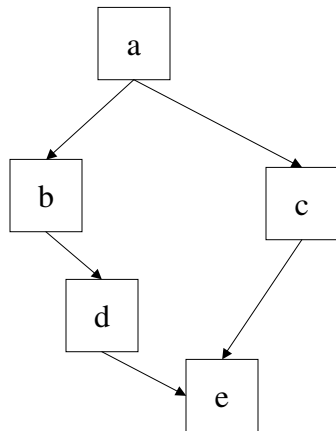
Week 8-2

## Unit testing

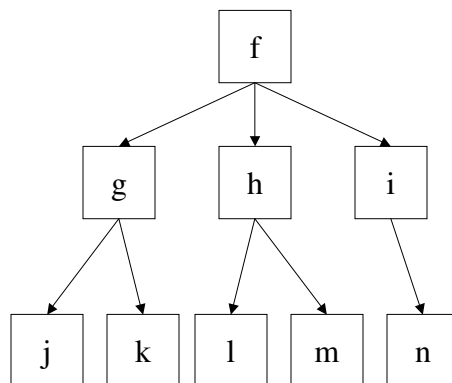
- White box
  - what is the task of a unit
  - test every paths/statement/independent path
- Black box
  - what is the responsibilities of the unit
  - test every path in the specification
    - boundary conditions, off-nominal (graceful degradation)
- Re-use for regression

# Integration testing

- Vocabulary not consistent on integration testing
- Objective: units tested together to check whether they work together
  - Top down: use of stubs for unit not yet implemented
  - Bottom up: based on dependency graph



- 1: e
- 2: d and c
- 3: b
- 4: a



- 1: f with stubs for g, h, and I
- 2: test g,h, and I with stubs for j, k, l, m, n
- 3: test the whole system

## System testing

- Objectives: test the whole system for its functionality
- A black box approach
  - create scenarios based on specifications
  - test error conditions
  - test graceful termination
  - test normal operation

## Other testing

- |                         |                        |
|-------------------------|------------------------|
| • Usability             | • Co-existence         |
| • Stress                | • Integration          |
| • Performance           | In any test activity,  |
| – regression            | establish              |
| • Regression            | – objective of testing |
| • Installation          | – adequacy criteria    |
| • Security              | – execution style      |
| • Function verification | – defect management    |

## Code review and inspection

- Principles:
  - testing is too expensive
  - common errors can be revealed through review
  - earlier an error is found, cheaper to fix
- Can be formal or informal
- Egoless programming
  - programming errors are not programmer's
- Review before integration

## Fagan inspections

- Team:
  - moderator: responsible for the organization of the meeting
  - reviewer(s)/reader(s)/inspector(s):
    - read and paraphrase the code constructing an abstraction
    - simulate/trace a simple test case (walk through)
    - go through a checklist
  - programmer
    - generally a silent partner
    - takes notes on identified problems

## Checklist

- Wrongful use of data: uninitialized variables, array index out of bound, dangling pointers
- faults in declarations: declaration of the same identified in nested blocks, undeclared variables
- faults in computations: division by zero, overflow, type mismatch, precedence of operators
- faults in relational expressions: > instead of >=
- faults in control flow: infinite loops, loop exits,
- faults in interfaces: wrong type of parameter, global variable inconsistency

## ISO 9000 and SEI CMM

- Repeatability of your success through process maturity
- ISO 9000 motto
  - say what you do      Document
  - do what you say      Follow
  - prove it      Demonstrate
  - improve it      No absolutes

... cont'd

- Five Levels of Capability Maturity Model
  - level 1: initial
    - no formalized procedures, plans, or processes
  - level 2: repeatable
    - control and more planning
  - level 3: defined
    - standard processes in place
  - level 4: managed
    - quantitative data is gathered
  - level 5: optimizing
    - focus on process optimization