東北大學
Northeastern University

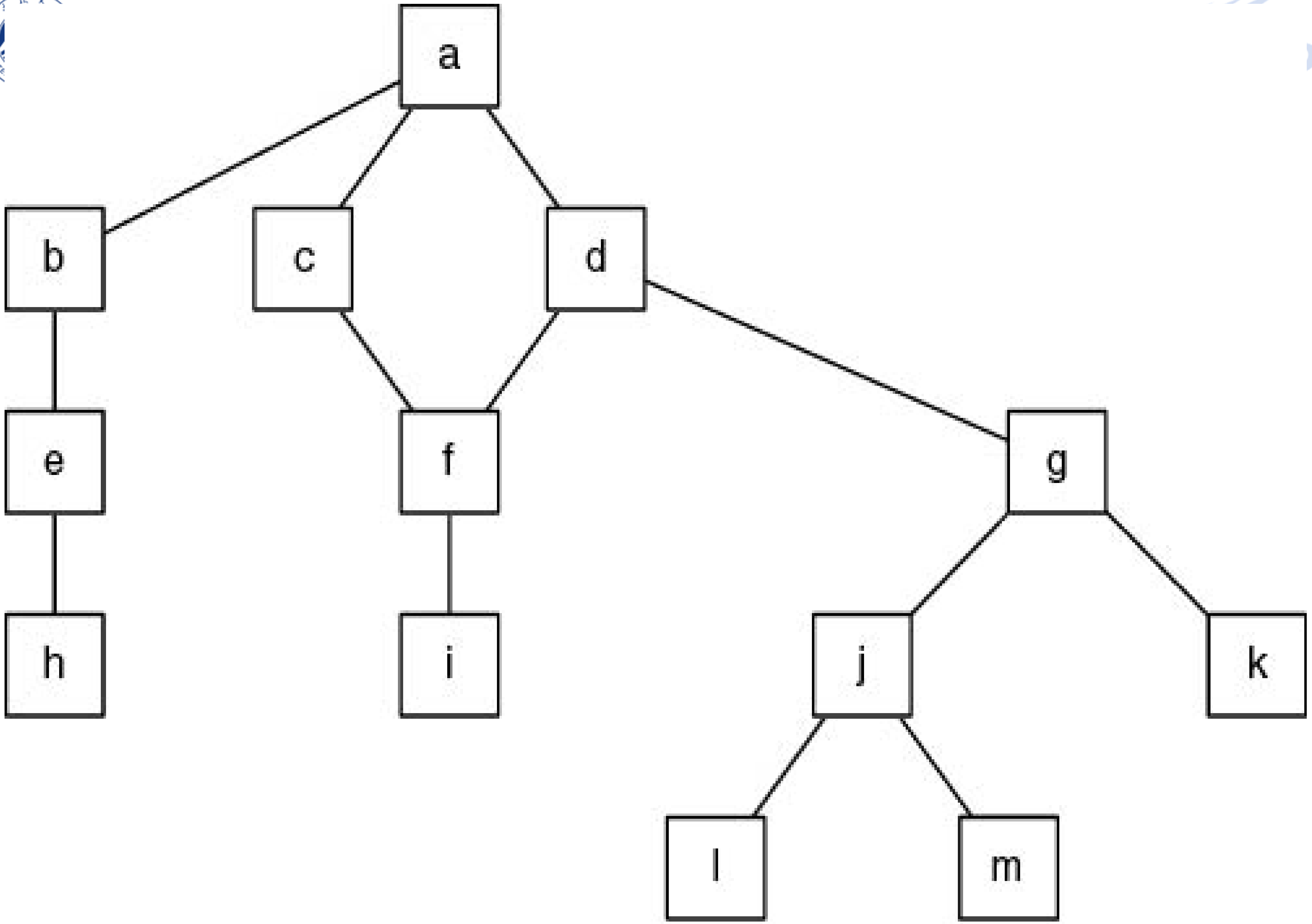
软件工程

张爽

东北大学软件学院

7.3

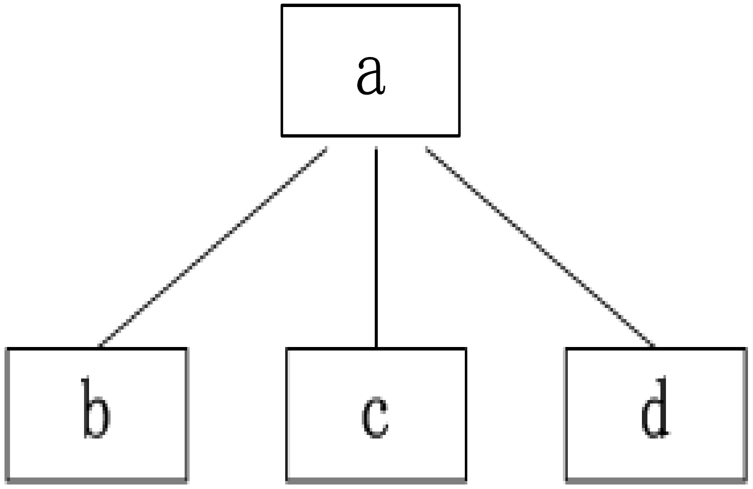
Implementation & Integration


a
b
c
d
e
f
g
h
i
j
k
l
m

Stubs

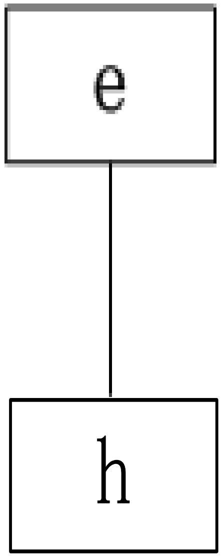
- **To code and test module *a*, modules *b*, *c*, *d* must be stubs**
  - **Empty module, or**
  - **Prints message ("b is called"), or**
  - **Returns precooked values from preplanned test cases**

```
// b() is a stub
b ( ){
    print("b is called.");
}
```



Drivers

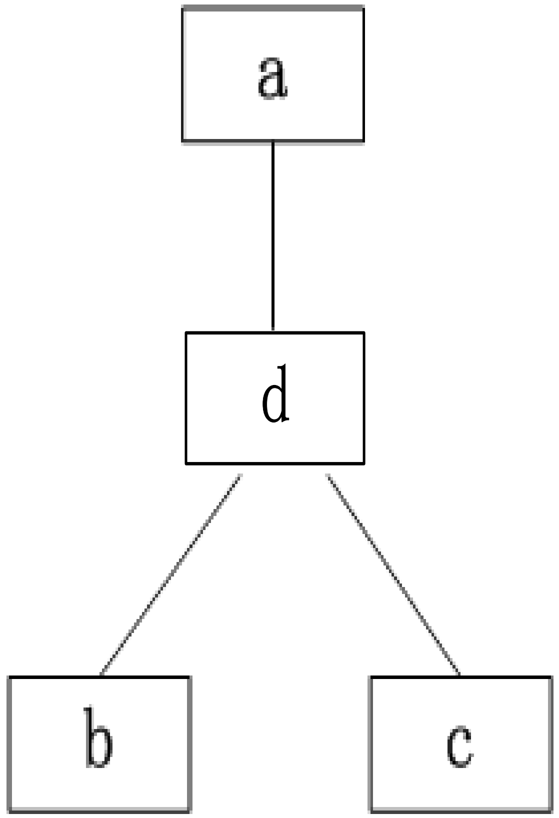
- **To code and test module *h* on its own requires a driver, which calls it**
  - **Once, or**
  - **Several times, or**
  - **Many times, each time checking value returned**



```
// e() is a driver
e ( ){
        call h( );
}
```

Drivers and Stubs

- To code and testing module ***d*** requires a driver and two stubs.



Implementation, then Integration

- Code and test each module separately
- Link all 13 modules together, test product as a whole.

Implementation, then Integration

- **Problem 1**
  - **Stubs and drivers must be written, then thrown away after module testing is complete.**
- **Problem 2**
  - **Lack of fault isolation.**
  - **A fault could lie in any of 13 modules or 13 interfaces.**
  - **In a large product with, say, 103 modules and 108 interfaces, there are 211 places where a fault might lie.**

Implementation, then Integration

- Solution to both problems
  - Combine implementation and integration methodically
  - "Implementation and integration phase"

Top-down Implementation and Integration

- **If module *mAbove* calls module *mBelow*, then *mAbove* is implemented and integrated before *mBelow*.**
- **One possible top-down ordering is**

  a, b, c, d, e, f, g, h, i, j, k, l, m
- **Another possible top-down ordering is**

	a
[a]	***b, e, h***
[a]	***c, d, f, I***
[a, d]	***g, j, k, l, m***

Top-down Implementation and Integration

- **Advantage 1: Fault isolation**
  - **Previously successful test case fails when *mNew* is added to what has been tested so far, the fault certainly lies within *mNew* or in the interface.**

Top-down Implementation and Integration

- **Advantage 2: Major design flaws show up early**
  - **Logic modules include decision-making flow of control**
    - ✓ **In the example, modules *a, b, c, d, g, j***
  - **Operational modules perform actual operations of module**
    - ✓ **In the example, modules *e, f, h, i, k, l, m***
  - **Logic modules are developed before operational modules.**

Top-down Implementation and Integration

- **Problem**
  - **Reusable modules are not properly tested.**
  - **Lower level (operational) modules are not tested frequently.**
  - **The situation is aggravated if the product is well designed.**
- **Defensive programming (fault shielding)**
  - **Example**

```
if (x >= 0)
     y = computeSquareRoot (x, errorFlag);
```

  - **Never tested with** *$x < 0$*

Bottom-up Implementation and Integration

- **If module *mAbove* calls module *mBelow*, then *mBelow* is implemented and integrated before *mAbove***
- **One possible bottom-up ordering is**

  l, m, h, i, j, k, e, f, g, b, c, d, a
- **Another possible bottom-up ordering is**

  h, e, b

  i, f, c, d

  l, m, j, k, g [d]

  a [b, c, d]

Bottom-up
Implementation and Integration

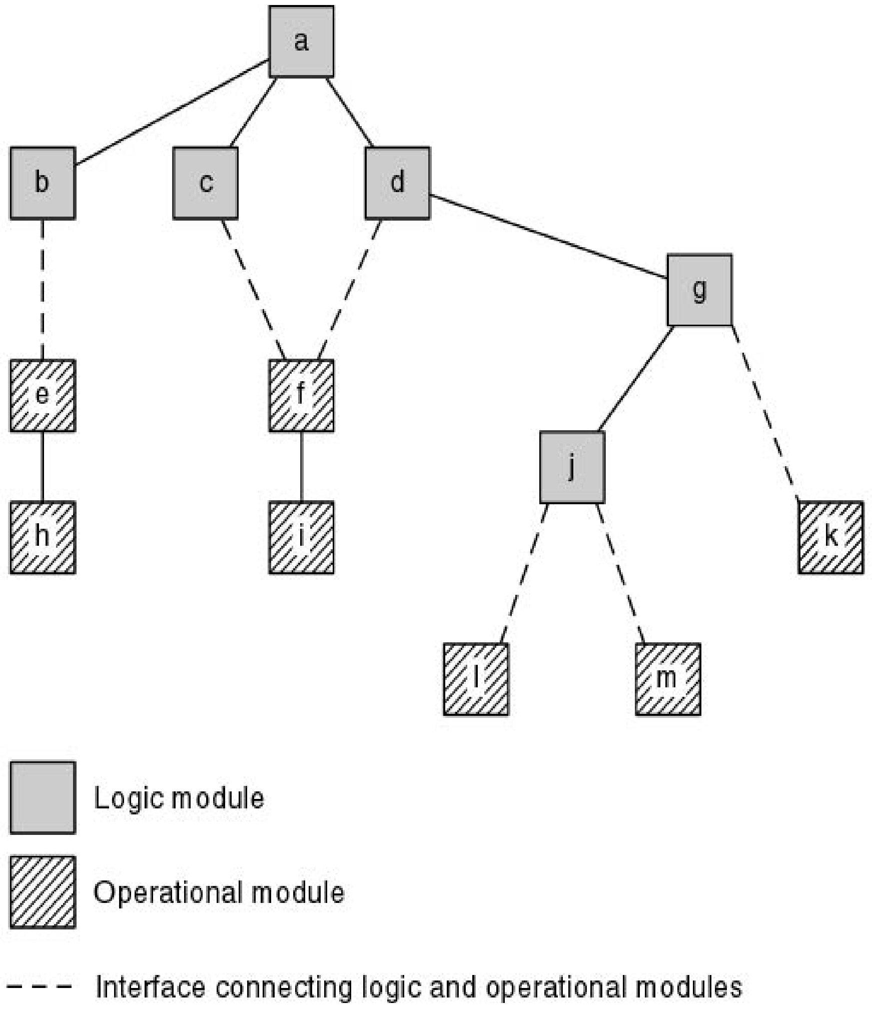
- **Advantage 1**
  - **Operational modules are thoroughly tested.**
  - **Operational modules are tested with drivers, not by fault shielding, defensively programmed calling modules.**
- **Advantage 2**
  - **Fault isolation**

Bottom-up Implementation and Integration

- **Problem**
  - **Major design faults are detected late.**
- **Solution**
  - **Combine top-down and bottom-up strategies making use of their strengths and minimizing their weaknesses**

Sandwich Implementation and Integration

- **Logic modules are implemented and integrated top-down.**
- **Operational modules are implemented and integrated bottom-up.**
- **Finally, the interfaces between the two groups are tested.**



Sandwich Implementation and Integration

- **Advantage 1**
  - **Major design faults are caught early.**
- **Advantage 2**
  - **Operational modules are thoroughly tested.**
  - **They may be reused with confidence.**
- **Advantage 3**
  - **There is fault isolation at all times.**