Unit Testing

Dr. Mark C. Paulk

SE 4367 – Software Testing, Verification, Validation, and Quality Assurance

Jonsson School of Engineering and Computer Science

Phases of Testing Topics

Part IV. Phases of Testing

9. Test Selection, Minimization, and Prioritization for Regression Testing

10. Unit Testing

11. Integration Testing

A Software Unit

Could be a class or collection of classes in an object-oriented language

Could be a function or collection of functions in a procedural language

Typically coded by a single programmer

Units → Components → Items

JUnit

A popular tool for unit testing of Java programs
 • other X-Unit tools for other languages

For each test, JUnit informs the tester whether
the test passed or failed.

An example of using JUnit
 • each test case is a class that extends TestCase
 • several tests named testX, where X denotes
    the specific feature to be tested
 • JUnit first calls the method setUp()
 • next it runs the tests
 • finally it calls the method tearDown()

Phases of Testing Topics

Integration Testing

Performed when two or more components are combined to create a new subsystem or a system.

Focus is on errors that may exist during communications among the components being integrated.
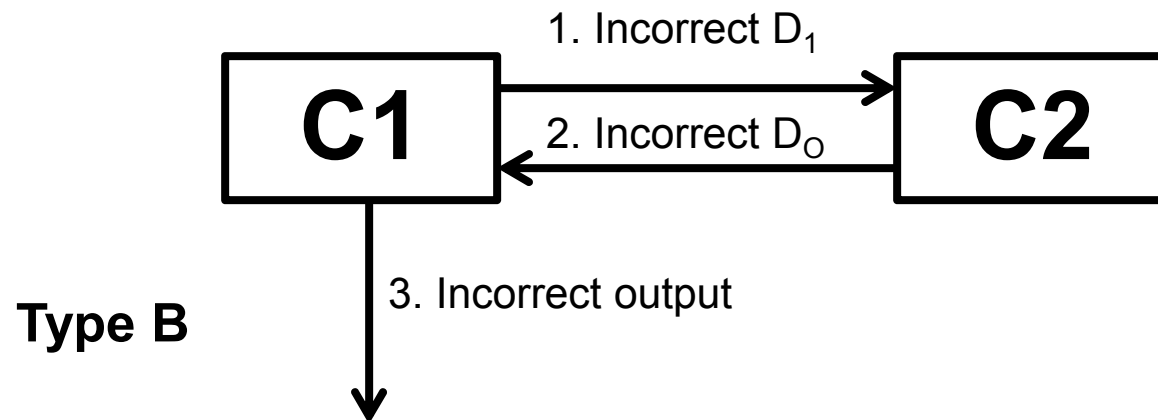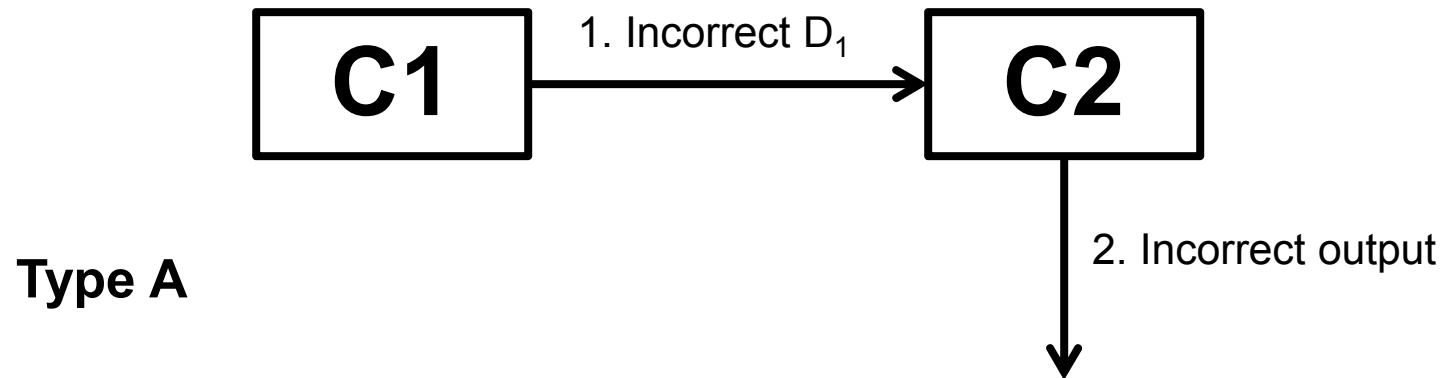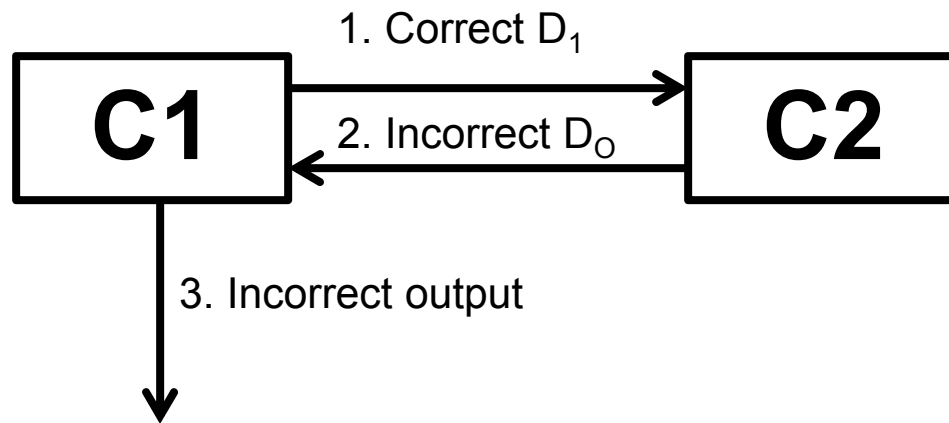 • Object Relation Diagram (ORD) represents relationships among components

When many components are being integrated, it may be important to determine a (near) optimum integration sequence.
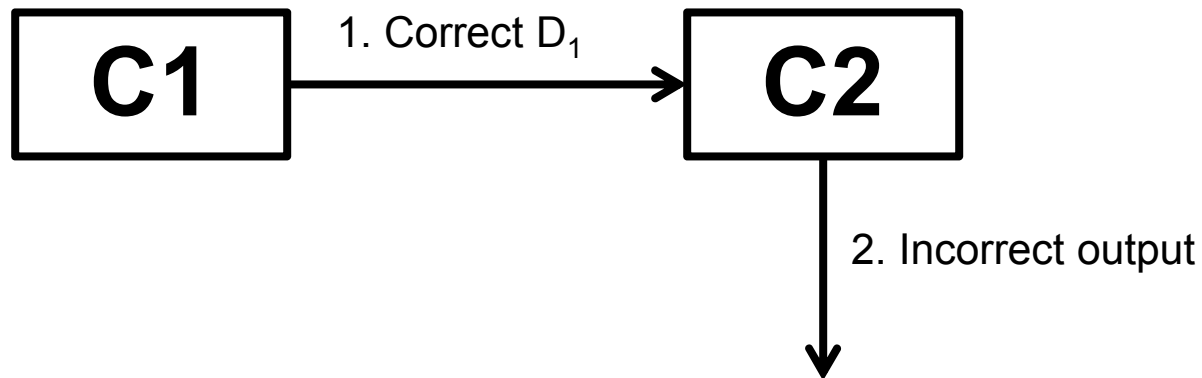 • top-down? bottom-up? a mix?

Types of Integration Errors
(Mathur, Figure 11.1)

```
┌─────────┐                          ┌─────────┐
│         │   1. Incorrect D₁        │         │
│   C1    │ ───────────────────────► │   C2    │
│         │                          │         │
└─────────┘                          └─────────┘
                                          │
                                          │  2. Incorrect output
                                          ▼
```

Type A

```
┌─────────┐   1. Incorrect D₁        ┌─────────┐
│         │ ───────────────────────► │         │
│   C1    │   2. Incorrect D_O       │   C2    │
│         │ ◄─────────────────────── │         │
└─────────┘                          └─────────┘
     │
     │  3. Incorrect output
     ▼
```

Type B

1. Correct D_1

2. Incorrect D_O

3. Incorrect output

Type C

1. Correct D_1

2. Incorrect output

Type D

Choosing an Integration Strategy

Choosing an integration strategy (and sequence) depends on:
  - **availability of code**
  - **difficulty or ease of constructing stubs**
  - **difficulty or ease of constructing drivers**
  - **size of the test team**

Finding a Near-Optimal Test Order

The problem of finding an optimal integration
sequence is NP-complete.

Methods for finding a near-optimal sequence
  • Tai-Daniels (TD)
  • Traon-Jeron-Jezequel-Morel (TJJM)
  • Briand-Labich-Wang (BLW)

Questions and Answers