LECTURE 12: INTEGRATION TESTING

Outline

- Hierarchy of testing
- Integration Testing
  - Big bang
  - Top-down
  - Bottom-up
  - Sandwich
- Regression Testing
Tuesday’s Quote

- *This is our last lecture for CS 4311!!!!*
  - Dr. Salamah

- *You’ll get your exams back on Thursday!*
  - Dr. Salamah

In Class

- Suppose you have constructed and tested all of the units.
- What are the possible things you can do next?
Integration Testing

- Assumes the units have been tested individually
- Done between unit testing and system testing
- Objective: ensure assembled modules, that work fine in isolation, work together
  - Identifies errors in the *interfaces* between units
- Need a module call graph.

```
main
   A  B  C
      D  E  F
```
Review: Test Procedures

How low-level modules are assembled to form higher-level program entities

Strategy impacts
- the type of test tools to use
- the order of coding/testing units
- the cost of generating test cases
- the cost of locating and correcting detected defects
Integration testing strategies

- Several different strategies can be used for integration testing
- Comparison criteria:
  - fault localization
  - effort needed (for stubs and drivers)
  - degree of testing of modules achieved
  - possibility for parallel development
- Examples of strategies
  - Big-bang
  - Top-down
  - Bottom-up
  - Sandwich

Big Bang Integration - 1

- Non-incremental strategy
  - Unit test each module in isolation
  - Integrate as a whole

Q: Do we need test drivers and stubs for big-bang testing?
Big Bang Integration - 2

- **Advantages**
  - Convenient for small systems

- **Disadvantages**
  - Integration testing can only begin when all modules are ready
  - Fault localization difficult
    - Easy to miss interface faults

Top-Down Integration Testing - 1

- Modules are integrated by moving downward through control hierarchy.
- Modules subordinate to main control module are incorporated
  - Depth-first
  - Breadth-first

What gets tested first in breadth first?
Top-Down Integration Testing - 2

- In Top Down, Breadth First
  - M1 tested with stubs for M2, M3 and M4
  - Then M1-M2 tested with stubs for M3, M4, M5, and M6
  - Then M1-M2-M3 tested with stubs for M4, M5, M6, and M7
  - ...

Top-Down Integration Testing - 3

- The integration order can be modified to:
  - include critical modules first
  - leave modules not ready to later
Steps in Top-Down Integration Testing

- The main module is used as a test driver
- Stubs are substituted for all components directly subordinate to the main control module
- Stubs are replaced one at a time with actual components
- Tests are conducted as each component is integrated

Evaluation of Top-Down Strategy

- Advantages
  - Fault localization easier
  - Few or no drivers needed
  - Possibility to obtain an early prototype
  - Different order of testing/implementation possible
  - Major design flaws found first
    - in logic modules at the top of the hierarchy
- Disadvantages
  - Need lot of stubs/mock objects
  - Potentially reusable modules (in bottom of the hierarchy) can be inadequately tested (Why?)
Bottom-up Integration - 1

- Begins construction and testing with atomic modules
- Need for stubs is eliminated
- Operational modules tested thoroughly
- Steps
  - Low-level components are combined into clusters (builds) to perform specific sub-function
  - A driver is written to coordinate test case input and output
  - Cluster is tested
  - Drivers are removed and clusters are combined moving upward in program structure.

Bottom-up Integration - 2

- Incremental strategy
  - Test low-level modules, then
  - Modules calling them until highest level module

```
A   B   C
  D   E   F
  test D, E, A
  test D, E, A
  test D, E, A
  test D, E, A
  test D, E, A
  test D, E, A
  test D, E, A
  test D, E, A
  test D, E, A
```
Bottom-up Integration - 3

- Incremental strategy
  - Test low-level modules, then
  - Modules calling them until highest level module

1. M8 tested with drivers
2. M5-M8 is tested with drivers needed for M5
3. M5-M6-M8 is tested with drivers needed for M5-M6
   ...

Bottom-up Integration - 4

- Advantages
  - Fault localization easier (than big-bang)
  - Less need for stubs/fewer mock objects
  - Operational modules tested thoroughly
  - Testing can be in parallel with implementation

- Disadvantages
  - Need drivers
  - High-level modules (that relate to the solution logic) tested in the last (and least)
Sandwich Integration

- Combination of bottom-up and top-down integrations
- System is viewed as layers

**Approach 1:**
- Top-down approach is used for the top layer
- A bottom-up approach is used for the bottom layer
- Allows integration to begin early in the testing phase

Sandwich Integration – Cont.

**Approach 2:**
- Start with a layer in the middle
- Use drivers and stubs to check
- Work out from middle
- Allows integration to begin early in the testing phase
Risk Driven integration

- Integrate base on criticality
  - most critical or complex modules integrated first with modules called

Regression Testing

- Adding new or changing module impacts the system
  - New data flow paths established
  - New I/O may occur
  - New control logic invoked

- Regression testing is re-execution of subset of tests that have already been conducted
- Ensures changes have not propagated unintended side effects
Regression Testing – Cont.

- Approaches
  - Manual testing
  - Capture/Playback tools: capture test cases and results for subsequent playback and comparison

- Test suite contains following classes of test cases:
  - Representative sample of tests that exercises all software functions
  - Focus on functions likely affected by change
  - Focus on components that have been changed

This is it😊

- Congratulations! You’ve survived the boring lectures of CS4311.
Final Presentations

- Available slots (60 minutes for presentations and questions, with at least 30 minutes between presentations)
  - Wednesday 12/06/2017
    - 9:00-4:30
  - Thursday 12/07/2017
    - 8:00-10:30
    - 12:00-6:00