

CSC 309 Lecture Notes Week 5

Testing Implementation

I. Highlights of Milestone 4

I. Highlights of Milestone 4

A. Due Fri Wk 7, May 15th.

I. Highlights of Milestone 4

A. Due Fri Wk 7, May 15th.

B. Choose testing framework, coverage tool

I. Highlights of Milestone 4

A. Due Fri Wk 7, May 15th.

B. Choose testing framework, coverage tool

C. Finish Spest for all methods

I. Highlights of Milestone 4

- A.** Due Fri Wk 7, May 15th.
- B.** Choose testing framework, coverage tool
- C.** Finish Spent for all methods
- D.** Finish approx 75% of model/view design & implementation

I. Highlights of Milestone 4

- A.** Due Fri Wk 7, May 15th.
- B.** Choose testing framework, coverage tool
- C.** Finish Spent for all methods
- D.** Finish approx 75% of model/view design & implementation
- E.** Unit tests for 4-6 methods per person

I. Highlights of Milestone 4

- A.** Due Fri Wk 7, May 15th.
- B.** Choose testing framework, coverage tool
- C.** Finish Spent for all methods
- D.** Finish approx 75% of model/view design & implementation
- E.** Unit tests for 4-6 methods per person
- F.** See what coverage looks like

II. Different Styles of Functional Testing

- A.** Pure end user.
- B.** Log-based testing.
- C.** X-Unit testing.

III. Pure End User Testing Pros and Cons

A. *Pros:*

- Ensures HCI evaluated by humans.
- Makes programmer involvement indirect.

End-User Testing Pros and Cons, cont'd

B. *Cons:*

- Difficult to verify coverage.
- Difficult to do stress testing.

End-User Testing Pros and Cons, cont'd

C. *Other considerations:*

- Works well with suitable work force.
- Should always be done in some form.

IV. Log-Based Testing Pros and Cons

A. *Pros:*

- Expected test results defined concretely.
- Reports only differences.
- External oracle implementation.

Log-Based Testing Pros and Cons, cont'd

B. *Cons:*

- Generating expected results may be tedious
- External oracle implementation.

Log-Based Testing Pros and Cons, cont'd

C. *Other considerations:*

- Useful when spec is data-oriented, e.g., a compiler.
- Can be used in combo with X-Unit.

V. X-Unit Pros and Cons

A. *Pros:*

- Formal spec oracle directly implemented.
- No need to generate expected output data.

X-Unit Testing Pros and Cons, cont'd

B. *Cons:*

- Oracle imple'n may itself be buggy.
- Oracle execution may take longer than diff.

X-Unit Testing Pros and Cons, cont'd

C. *Other considerations:*

- Becoming an industry standard.
- Can support log-style if desired.

VI. What We'll Do in 309

- A.** Complete X-unit testing of model and process packages, with some use of logging.
- B.** Pure end-user testing of model+view.
- C.** Examination of automated view testing.

VII. Unit test development steps:

- A.** Finish Sppest specs.
- B.** Use specs to generate/write tests.
- C.** Refine specs to get better tests, based on coverage results.

VIII. Anatomy of a unit test plan.

A. Javadoc comment for a unit test method.

B. In the form of a table:

```
<pre>
```

Test Case	Inputs	Expected Outputs	Remarks
1	x=1	this.x == 1	Set x
2	...		

```
</pre>
```

C. Examples coming up.

IX. Anatomy of a unit test method.

A. Class and method under test:

```
class X {  
  
    // Method under test (with Spest spec)  
    public Y m(A a, B b, C c) { ... }  
  
    // Data field inputs  
    I i;  
    J j;  
  
    // Data field output  
    Z z;  
  
}
```

B. Testing class and method:

```
class XTest {  
    /** Test plan goes here */  
    public void testM() {  
  
        // Set up  
        X x = new X(...);  
        ...  
  
        // Invoke  
        Y y = m(aValue, bValue, cValue);  
  
        // Validate  
        assertEquals(y, expectedY);  
    }  
}
```

Anatomy of a unit test method, cont'd

C. Common core of unit test method:

1. *Setup* -- set up inputs necessary to run test
2. *Invoke* -- invoke the method under test and acquire its actual output
3. *Validate* -- validate that actual output equals expected output

*During lecture we'll go through,
in some detail,
the Milestone 4 code examples for*
`Schedule.java`
and
`ScheduleTest.java.`