

CSC 7003 : Basics of Software Engineering

J Paul Gibson, A207

paul.gibson@int-edu.eu

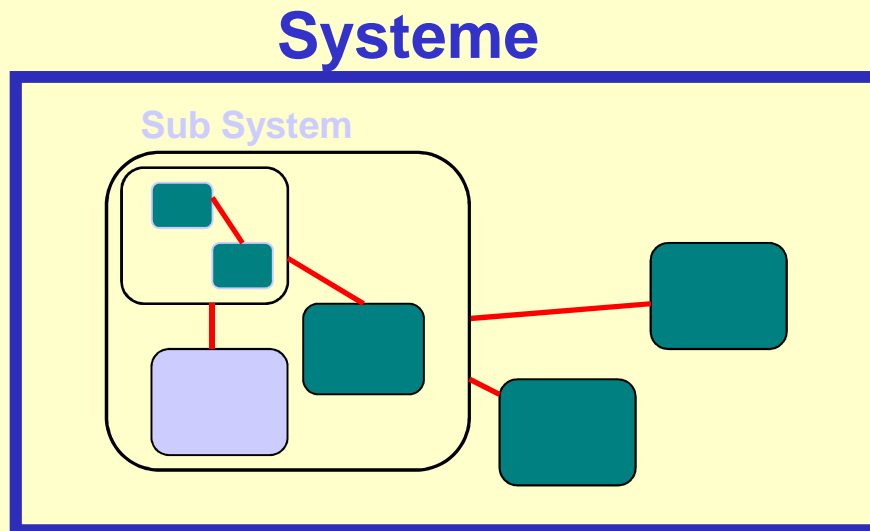
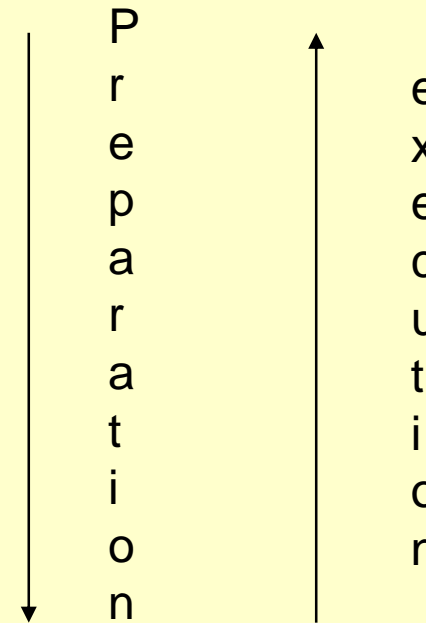
<http://www-public.it-sudparis.eu/~gibson/Teaching/CSC7003/>

Unit Tests

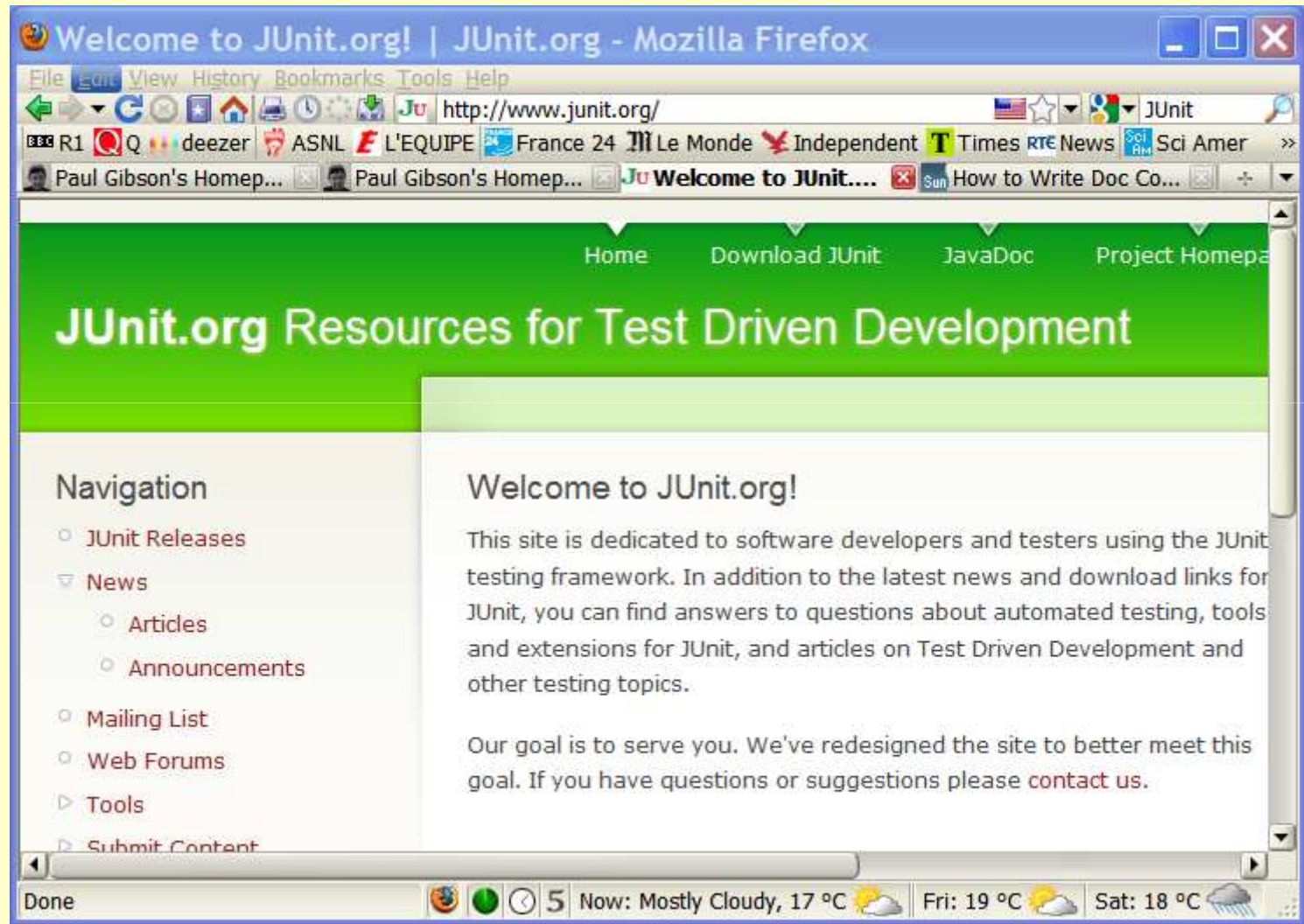
<http://www-public.it-sudparis.eu/~gibson/Teaching/CSC7003/L10-UnitTests.pdf>

Where do **unit tests** fit in the typical development process?

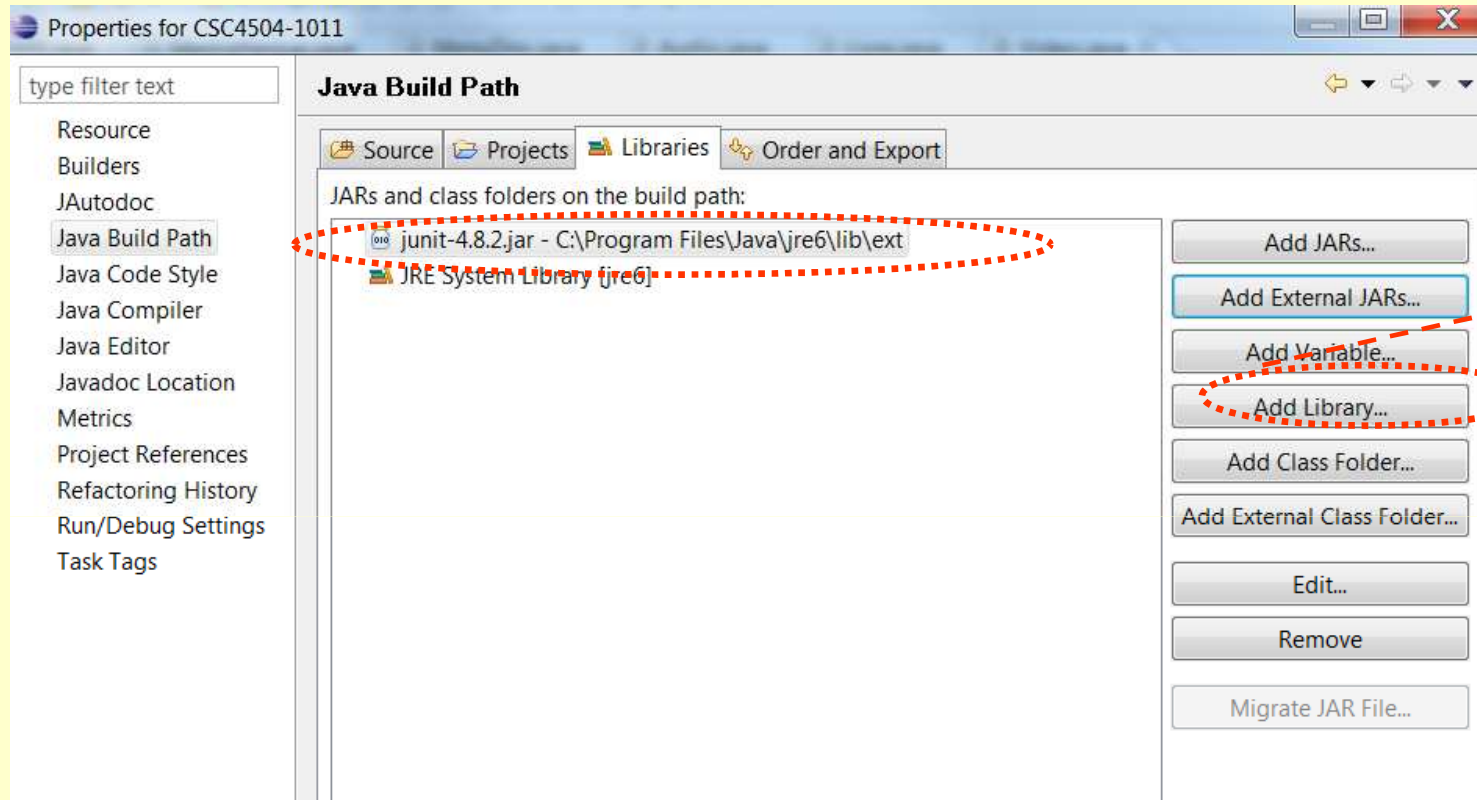
- **Validation tests**
- **Integration tests**
- **Unit tests**



Using **JUnit** test tool/classes to automate testing process.

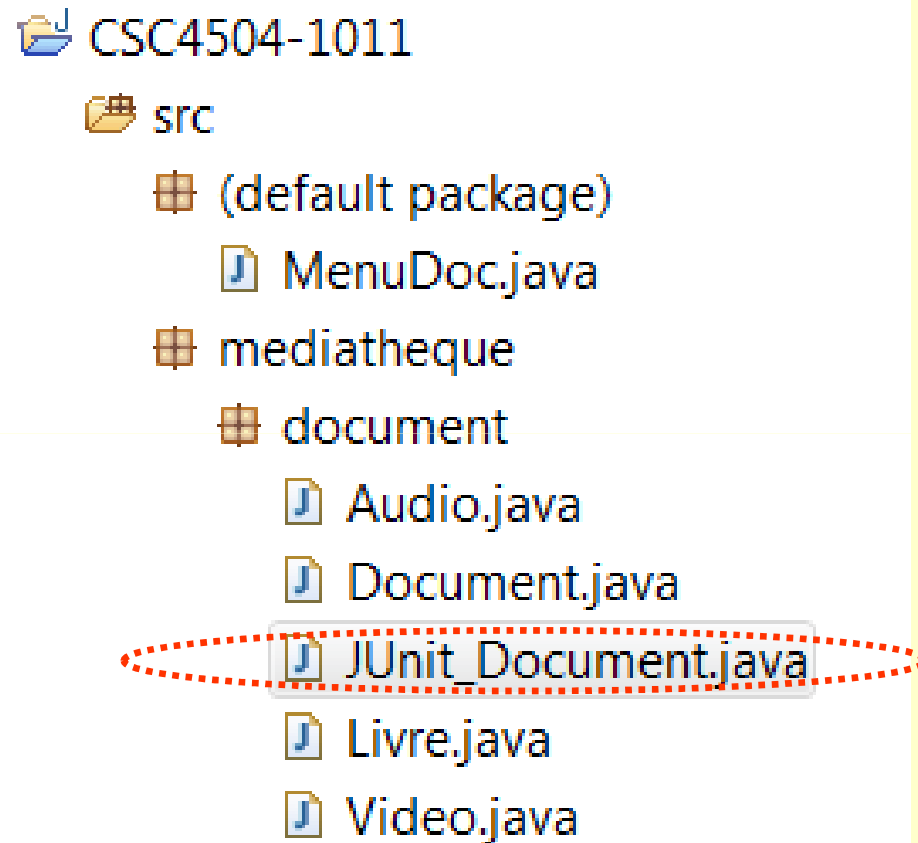


Running JUnit on test(s) through Eclipse interface



Make sure that JUnit is in buildpath of project

Running JUnit on test(s) through Eclipse interface



Choose option

Run as JUnit test

Example: Testing a Document class

```
package mediatheque.document;

import static org.junit.Assert.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;

import mediatheque.Genre;
import mediatheque.Localisation;
import mediatheque.OperationImpossible;
import util.InvariantBroken;

public class JUnit_Document {

    protected Localisation l;
    protected Genre g;
    protected Document d1;

    // setUp           Annotation @Before
    // tearDown        Annotation @After
    // tests            Annotation @Test
}
```

Using JUnit tool – assertions and failures

(see <http://junit.org/apidocs/org/junit/Assert.html>)

`assertTrue(boolean)`

`assertFalse(boolean)`

`assertArrayEquals(_ , _)`

`assertEquals(_ , _)`

`assertNull(_)`

`assertSame(_ , _)`

`fail()`

`fail(java.lang.String)`

`...`

Test1: check the construction of a Document respects the invariant

```
@Before
public void setUp() throws Exception{
    g = new Genre("Test_nom1");
    l = new Localisation("Test_salle1", "Test_rayon1");
    d1 = new Video ("Test_code1", l, "Test_titre1", "Test-
auteur1", "Test-annee1", g,
    "Test_dureel", "Test-mentionLegale1");
} //END JUnit_Document.setUp
```

Test1: check the construction of a Document respects the invariant

```
@After  
public void tearDown() throws Exception {  
    l=null; g=null; d1=null;  
} // END JUnit_Document.tearDown
```

Test1: check the construction of a Document respects the invariant

- `void mediatheque.document.JUnit_Document.constructorInvariant()` throws `OperationImpossible`, `InvariantBroken`

`@Test`

Document TEST 1

Construct valid concrete subclass (video) and check invariant

Throws:

[OperationImpossible](#)

[InvariantBroken](#)

```
@Test
```

```
public void constructorInvariant( ) throws  
OperationImpossible, InvariantBroken{
```

```
    assertTrue(d1.invariant());
```

```
} //END JUnit_Document.constructorInvariant
```

Test2: Check all reachable states to be SAFE, ie respect the invariant

- `void mediatheque.document.JUnit_Document.reachableStates()` throws `OperationImpossible`, `InvariantBroken`

`@Test`

Document TEST 2

Check all reachable states to be SAFE, ie respect the invariant Complete coverage - 3 states - from single test trace

Throws:

[OperationImpossible](#)

[InvariantBroken](#)

```
@Test
public void reachableStates () throws OperationImpossible,
InvariantBroken {

    assertTrue(d1.invariant());
    d1.metEmpruntable();assertTrue(d1.invariant());
    d1.emprunter();assertTrue(d1.invariant());

} // END JUnit_Document.reachableStates
```

« All other tests are successful »

```
Package Explorer | JUnit
Finished after 0,01 seconds
Runs: 5/5 | Errors: 0 | Failures: 0
mediatheque.document.JUnit_Document [Runner: JUnit]
  constructorInvariant (0,000 s)
  reachableStates (0,000 s)
  statistics (0,000 s)
  restituerBeforeEmprunter (0,000 s)
  constructorException (0,000 s)
Failure Trace

Document.java | JUnit_Document.java | JUnit_Mediatheque.java
*/
@Test
public void statistics () throws OperationImpossible, InvariantBroken {

    assertEquals(Audio.getStat(), 0);
    assertEquals(Livre.getStat(), 0);
    assertEquals(Video.getStat(), 1);

    d1.metEmpruntable();
    for (int i=0; i<5; i++){ d1.emprunter();d1.restituer();
    }
    assertTrue(d1.invariant());
    assertEquals(Audio.getStat(), 0);
    assertEquals(Livre.getStat(), 0);
    assertEquals(Video.getStat(), 6);

    } // END JUnit_Document.statistics

/**
 * Document TEST 4 1 <br>
<terminated> JUnit_Document [JUnit] C:\Program Files\Java\jre6\bin\javaw.exe (6 d c. 2010 13:02:14)
Attention: Test-mentionLegale1
Attention: Test-mentionLegale1
Document: ranger "Test_titre1" en Salle/Rayon : Test_salle1/Test_rayon1
Attention: Test-mentionLegale1
Document: ranger "Test_titre1" en Salle/Rayon : Test_salle1/Test_rayon1
Attention: Test-mentionLegale1
Document: ranger "Test_titre1" en Salle/Rayon : Test_salle1/Test_rayon1
Attention: Test-mentionLegale1
Document: ranger "Test_titre1" en Salle/Rayon : Test_salle1/Test_rayon1
Attention: Test-mentionLegale1
Document: ranger "Test_titre1" en Salle/Rayon : Test_salle1/Test_rayon1
```

QUESTION: What other sort of tests need to be executed?

TO DO: Read *Software Unit Test Coverage and Adequacy* , H Zhu, P Hall and J May, 1997