Anglia Ruskin University

TESTING

Evaluation of program or system correctness

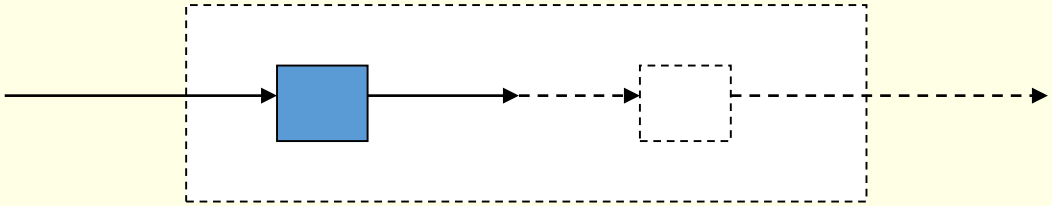
1. Requirements
2. Design
3. Implementation
4. Testing
5. Maintenance

- System Testing
- Regression Testing
- Acceptance Testing
- Alpha Testing
- Beta Testing
- Unit Testing
- Integration Testing

Anglia Ruskin University

White box (unit) testing

White = ☐ = transparent = see inside the system



Test individual components of a program or system

A unit = method OR a class (if a multi-class system)

Boundary checking, Type error checking

Anglia Ruskin University

EG: test a method functionality

```
public static double convert( double F )
{
   double C = 3.0 * ( F – 32.0 ) * 5.0 / 9.0;
   return C;
}
```

White box test using a *Test Harness*

Anglia Ruskin University

```
class TestHarness
{
   public static void Main( string[] args )
   {
     double output = TestHarness.convert( 100.0 );
     Console.WriteLine("T(C) = " + output );
   }

   public static double convert( double T )
   {
     double C = ( T - 32.0 ) * 5.0/9.0;
     return C;
   }
}
```

Anglia Ruskin University

Test Plan

Unit test 1: positive F returns correct C

Unit test 2: negative F returns correct C

Unit test 3: Integer input returns correct C

Test Results

Unit Tests				
No.	*Input*	*Expected*	*Output*	*Comment*
1	57.6	14.22	14.22	pass
2	-73.3	-58.5	-58.5	pass
3	100	37.77	37.77	pass

Anglia Ruskin University

Black box (integration) testing

Black = ■ = opaque = cannot see inside the system

Test whole program or system

Overall product input/output

Anglia Ruskin University

```
class Tdifference
{
  public static void Main(string[] args)
  {
    Console.WriteLine("T1 in F:");
    double T1 = double.Parse( Console.Readline() );
    Console.WriteLine("T2 in F:");
    double T2 = double.Parse( Console.Readline() );
    double C1 = Tdifference.convert( T1);
    double C2 = Tdifference.convert( T2);
    double ans = C1 - C2;
    Console.WriteLine("Temp diff in C is: “ + ans);
  }

  public static double convert( double F )
  {
   double C = ( F - 32.0 ) * 5.0/9.0;
    return C;
  }
}
```

Anglia Ruskin University

Test Plan

Integration test 1: Two input T's return correct difference

Integration test 2: Lower 2nd T outputs absolute result

Test Results

Integration Tests				
No.	*Input*	*Expected*	*Output*	*Comment*
1	20.0, 10.0	5.55	5.55	pass
2	10.0, 20.0	5.55	-5.55	fail

Anglia Ruskin University

Test Plan and Results – date/version 1

Test Plan

Unit Tests

Unit test 1: positive F returns correct C
Unit test 2: negative F returns correct C
Unit test 3: Integer input returns correct C

Integration Tests

Integration test 1: Two input T's return correct difference
Integration test 2: Lower 2nd T outputs absolute result

Anglia Ruskin University

Test Results

Unit Tests				
No.	*Input*	*Expected*	*Output*	*Comment*
1	57.6	14.22	14.22	pass
2	-73.3	-58.5	-58.5	pass
3	100	37.77	37.77	pass

Integration Tests				
No.	*Input*	*Expected*	*Output*	*Comment*
1	20.0, 10.0	5.55	5.55	pass
2	10.0, 20.0	5.55	-5.55	fail

Cannot test everything!

Anglia Ruskin University

Report Document

1. Introduction – describe requirements
2. Design – written sentences +/- pseudocode
3. Testing – test plan and results
4. Conclusion – eg assess if requirements met
5. References – eg give source of hacked code
6. Appendix I – user instructions
7. Appendix II – source code print out + disk

Testing: further info see:
software_testing_quick_guide.pdf