

Unit Testing-Approaches, Tools, Traps

Unit Test Approaches Tools Traps

Presented

By

Tom Rooker

Of

Abbott Analytical Products

Unit Testing-Approaches, Tools, Traps

Abbott Analytical Products

- **The region's leading resource for:**
 - **Software reliability**
 - **Unit test**
 - **Software verification/validation**
- **238 users of assurance engineering toolkit <FMEA, FTA, Mil-Hdbk-217, QSR 820, etc.> .**
- **Compliance/Regulated Industries Specialist: FAA, FDA, DCASPRO, commercial, banking, and insurance.**
- **Licensed and certified professionals.**
- **Established in 1988.**
- **Web site: <http://qed1.home.mindspring.com>.**

Unit Testing-Approaches, Tools, Traps

Opening Thoughts:

- **Unit testing is not for the faint of heart.**
- **Programming is easy. It is just ones and zeroes, highs and lows. It is the order that is critical.**
- **Sometimes it just takes more than doughnuts and pizza to get the job done.**
- **Code is self documenting.**

Unit Testing-Approaches, Tools, Traps

Presentation

- **Survey style: Minimal code then demo**
- **The Why, What, Who of Unit Testing**
- **Static Approaches**
- **Dynamic Approaches**
 - **Behavioral Model/Tools**
 - **Motivational Model/Tools**
- **Common Attributes of Getting It Done**
- **What Works / What Does Not Work**

Unit Testing-Approaches, Tools, Traps

Why Unit Test:

A simple equation

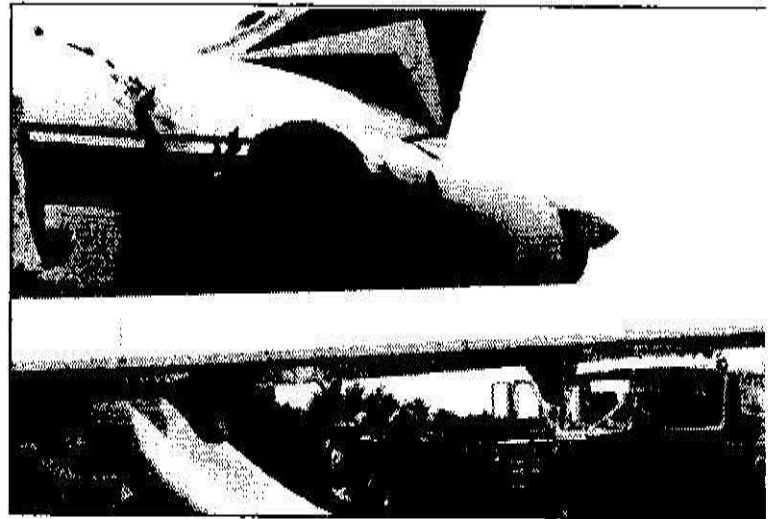
Raleigh, NC

Pensacola, FL

Bad engine grounds jet at RDU

By JEN GÓMEZ
STAFF WRITER

MORRISVILLE—A Continental Airlines jet remained grounded Tuesday at Raleigh-Durham International Airport after one its engines shut down shortly after takeoff Monday afternoon.



Firefighters examine the damaged left engine of Delta Flight 1288 in Pensacola, Fla. This photo was taken by a passenger.

THE ASSOCIATED PRESS

Unit Testing-Approaches, Tools, Traps

A simple equation

Engine
Fire/OS + **Sensor** + **SW** = **Engine** =====> **Walk**
Event **Suppression** **Away**

Unit Test

.....when the SW really has to work every time!

Unit Testing-Approaches, Tools, Traps

Untested Code Truisms¹

- **For untested code: Residue Bugs = Size of Untested Code * Probability of Bugs.**
- **Leave high probability paths to testing by others.**
- **Logic errors/fuzzy thinking inversely proportional to the probability of a path's execution.**

Unit Testing-Approaches, Tools, Traps

Untested Code Truisms¹

Continued

- **The subjective probability of executing a path as seen by the implementer and its objective probability are typically far apart.**
- **The subjective evaluation of the importance of a code segment by its implementer is strongly biased.**

Unit Testing-Approaches, Tools, Traps

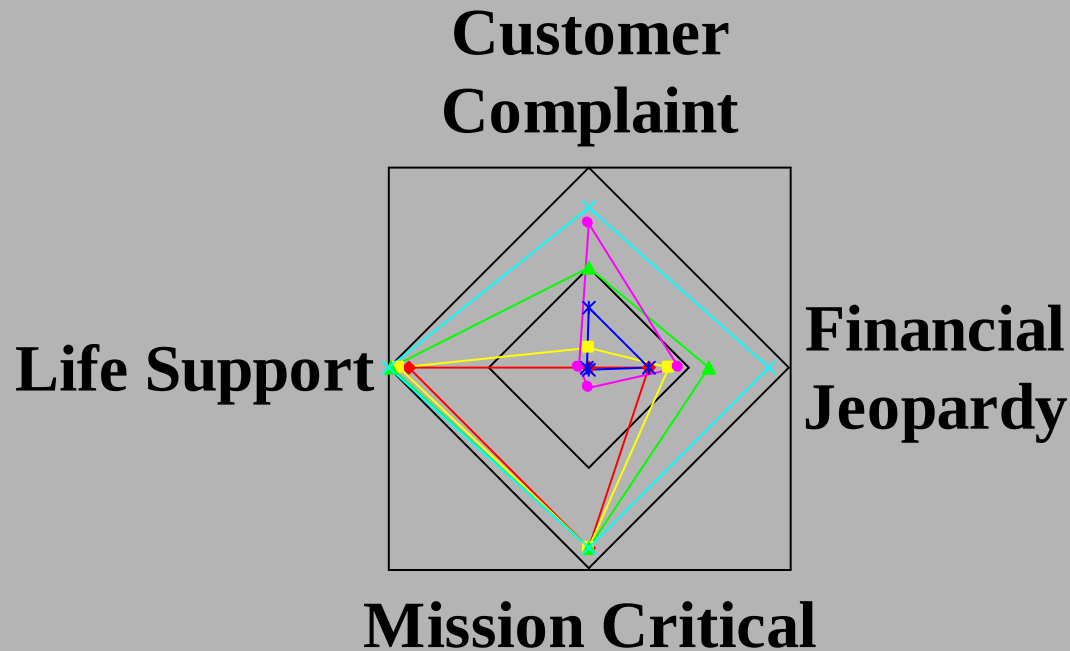
Organizational Influence

Unit testing depends on organizational focus²

- Schedule driven
- Requirements driven
- Document driven
- Quality driven
- Architecture driven
- Technology driven

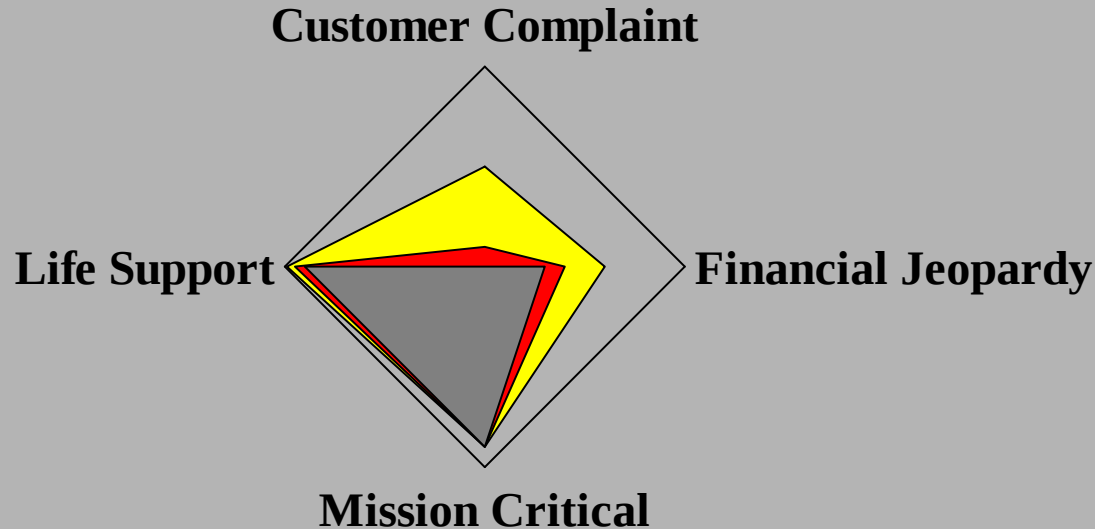
Unit Testing-Approaches, Tools, Traps

Warm-Fuzzy About Level of Commitment to Unit Testing



Unit Testing-Approaches, Tools, Traps

Warm-Fuzzy About Level of Commitment to Unit Testing



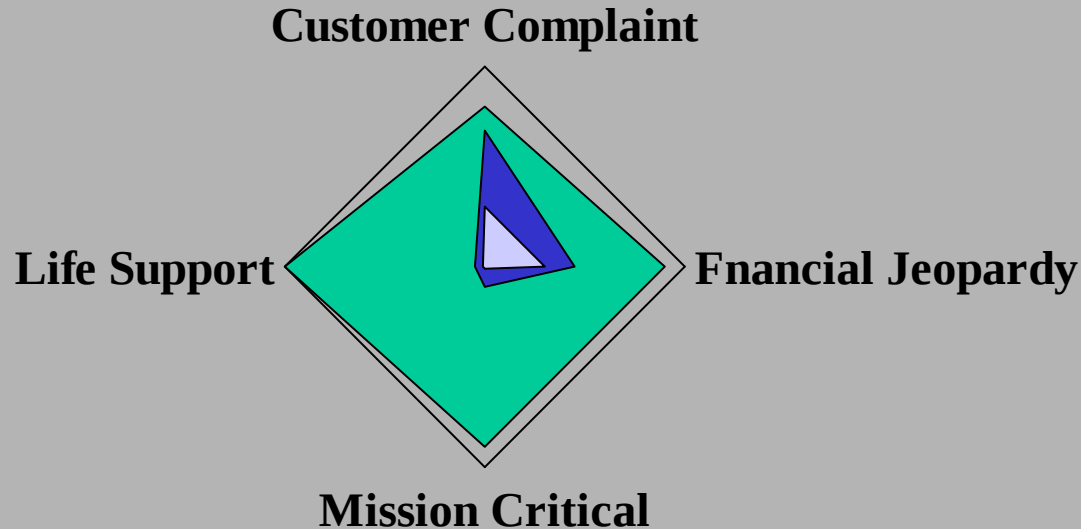
■ CMM3

■ CMM2

■ CMM1

Unit Testing-Approaches, Tools, Traps

Warm-Fuzzy About Level of Commitment to Unit Testing



■ FDA/FAA

■ Data Warehouse

■ Gamer

Unit Testing-Approaches, Tools, Traps

Unit Test and the Development Cycle

- **Requirement**
- **Design**
- **Code Implementation**
- **Unit Test**
- **Integration Test**
- **System/Verification/Validation Test**

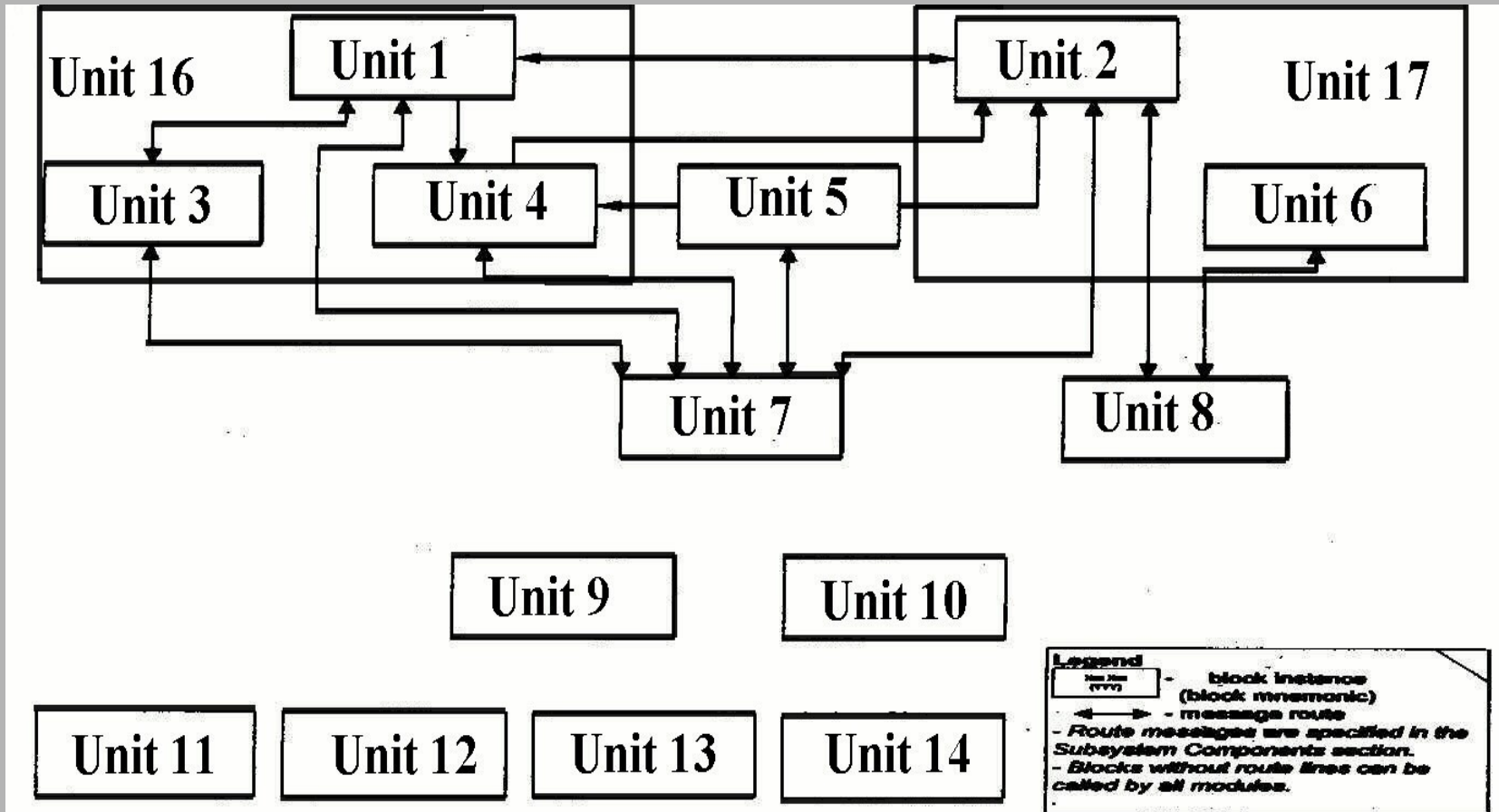
Unit Testing-Approaches, Tools, Traps

What is a unit?

- **There is considerable ambiguity in the definition of a unit.**
- **A design can usually be partitioned into one or more functional blocks or responsible entities.**
- **A unit is a software module or component of a design**
- **The smallest piece of software that can be independently tested (i.e. compiled or assembled, loaded, and tested). Usually the work of one programmer consisting of a few hundred lines of source code.** ³
- **For OOP languages such as C++, Java, Delphi a class is equivalent to a unit.**⁵

Unit Testing-Approaches, Tools, Traps

System Decomposed into Many Units



Unit Testing-Approaches, Tools, Traps

Unit Test Terms

- **Requirements Traceability**
- **Structural Test**
- **Code Review/Walkthrough**
- **Functional/Black Box Test**
- **Coverage Analysis/White Box Test**

Requirements Traceability

Requirements drive testing

- ◆ **Explicit and implied requirements**
- ◆ **Uniquely identified requirements**
- ◆ **An associated risk**

Unit Testing-Approaches, Tools, Traps

Requirements Mapping

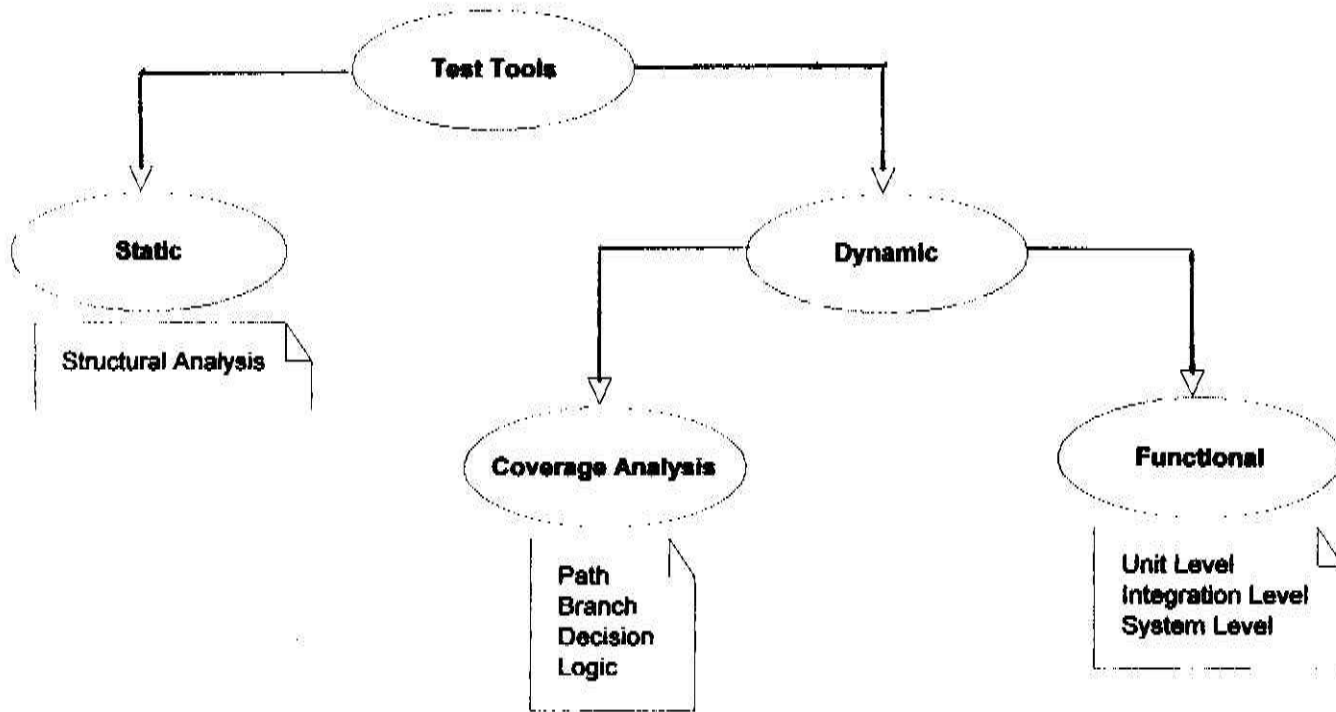
Requirement: Note: Only the requirement number has been given for this example.	Risk Analysis	Parent	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10	Unit 11	Unit 12	Unit 13	Unit 14	Unit 15	Unit 16	Unit 17	Interface	Hardware	Subtotal	
BNCM00001:		1				1	1	1			1	1						1				7	
BNCM00002:	\$E3	1	1	1							1	1							1			6	
BNCM00003:		1	1			1				1	1	1										1	7
BNCM00004:		1		1	1				1			1							1	1		1	7
BNCM00005:	!R5	1		1								1						1	1		1	6	
BNCM00006:	!R5	1					1				1	1						1				5	
BNCM00007:		1		1								1										1	4
BNCM00008:	!T6	1			1					1		1								1	1	1	6
BNCM00009:		1	1				1					1						1		1		1	6
BNCM000010:	+G2	1							1			1								1		1	4
BNCM000011:		1										1								1		1	4
BNCM000012:	+G2	1							1		1			1									4
BNCM000013:		1	1											1	1	1	1			1			7
BNCM000014:	#T5	1					1								1					1			4
BNCM000015:		1										1								1	1	1	4
BNCM000016:	#Y5	1																				1	2
Subtotal		16	4	4	2	2	4	1	3	2	5	12	1	2	2	1	0	4	5	6	7	83	
		!	\$!	!		!		+	!	!	!		+	#			!	!	+	!		
Risk Severity Category:																							
! : Catastrophic. \$: Critical.																							
+ : Major. # : Minor.																							

Unit Testing

- **Static: Code Analyzed**
 - ◆ **Complexity Measures: McCabe**
 - ◆ **Code Review/Walkthrough**
- **Dynamic: Code Runs**
 - ◆ **From a Behavior Vantage Point**
 - ◆ **From a Motivational Perspective**

Unit Testing-Approaches, Tools, Traps

Unit Test Tools



Code Review/Walkthrough

- Adherence to project coding standards
- General code readability
- Sufficient and correct commenting
- A lack of constructs that could compromise functionality
- Second order effect
 - Test plan
 - Test cases
 - Readiness for tests

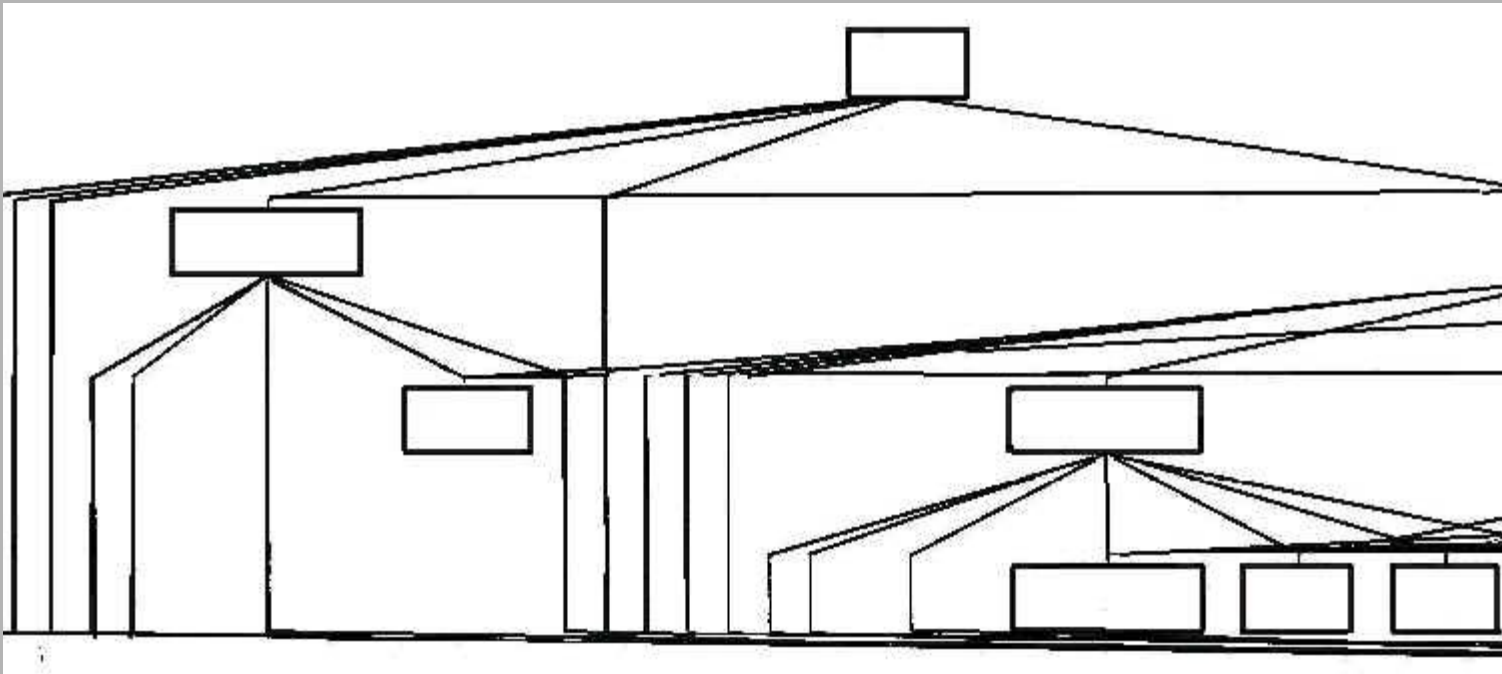
Unit Testing-Approaches, Tools, Traps

Complexity Measures: McCabe

- **Code paths mapped**
- **Indices produced**
- **Typical indices value serve as guide**

Unit Testing-Approaches, Tools, Traps

McCabe, Function Mapping



Unit Testing-Approaches, Tools, Traps

McCabe, Basic Metrics

Branch Metrics

Program: mcaabe.0111.01

Module Name	# Branches	# Covered	% Covered
-----	-----	-----	-----
.	5	1	20
.	37	0	0
.	17	0	0
.	17	0	0
.	37	0	0
.	5	0	0
.	3	0	0
.	15	0	0
.	7	0	0
.	7	0	0

Unit Testing-Approaches, Tools, Traps

McCabe, Metrics

01/16

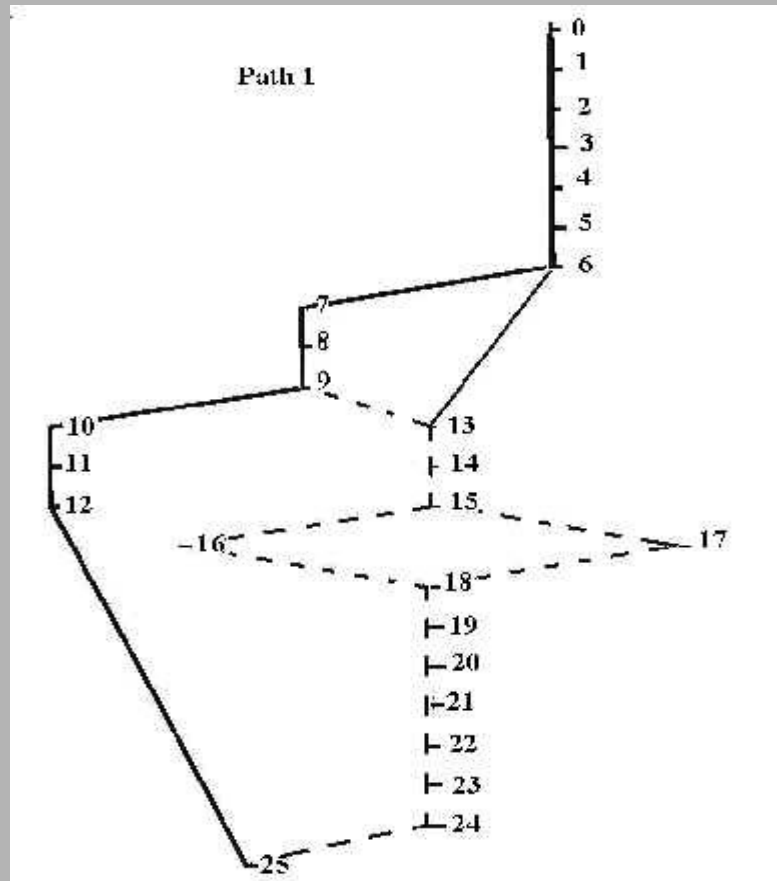
Module Metrics

Program: mcaabe.0111.01

Module Name	Mod #	v(G)	ev(G)	iv(G)	pv(G)	# Lines	Line #
	4	3	3	3	1	31	20328
	20	19	7	11	1	51	20371
	41	9	3	5	1	63	20433
	10	9	3	5	1	50	20329
	26	19	8	9	1	68	20390
	47	3	1	2	1	28	20469
	72	2	1	2	1	30	20336
	59	8	4	5	1	41	20377
	76	4	3	3	1	34	20428
	73	4	3	3	1	34	20473

Unit Testing-Approaches, Tools, Traps

McCabe, Path 1



Unit Testing-Approaches, Tools, Traps

Dynamic Unit Test

Functional Testing

or

Black Box Testing

{Behavior}

Unit Testing-Approaches, Tools, Traps

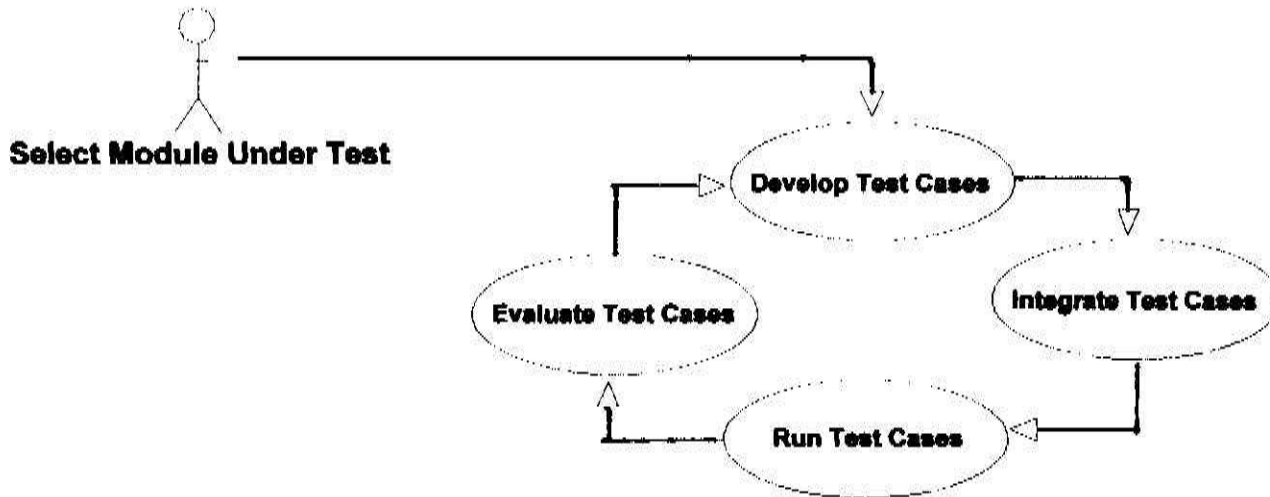
Dynamic Unit Testing: Behavior Vantage

- **Black Box Test**
- **Functional Test**
- **External Stimulus Yields
Observable External Response**

Unit Testing-Approaches, Tools, Traps

Dynamic Unit Test Flow

Functional and Coverage Analysis



Unit Testing-Approaches, Tools, Traps

Extreme Programming

Advocates Unit Testing

File Edit View Search Go Bookmarks Tasks Help

Back Forward Reload Stop <http://www.xprogramming.com/software> Go Search Print

Home Netscape Search Bookmarks Instant Message Internet Lookup New&Cool

XProgramming.com

an Extreme Programming Resource

[XProgramming.com](#) · [XP Magazine](#) · [What is XP?](#) · [Downloads](#) · [Services](#) · [Community](#) · [Publications](#) · [Q and A](#) · [C3's XP Practices](#) · [Recommended Reading](#) · [SEARCH](#)

XProgramming > Software Downloads

06/28/2002

Testing Framework (xUnit, unit testing) software is now hosted via anonymous FTP on XProgramming.com. Please note that some users have difficulty accessing anonymous FTP from behind firewalls. Links on the left side are direct file downloads where possible.

Recent Additions: C, C++, Flash, Web Applications, AppleScript, HTML, Manage JUnit Test Data, JavaScript, Linux Kernel, RealBasic, WebObjects, XML

Kent Beck's Original Testing Framework Paper	
For information on use of the testing framework, see http://c2.com/cgi/wiki?TestingFramework .	
For questions, post on the extremeprogramming discussion group on egroups. Sign up at XP Home .	
ADA	Ed Falis
ANT Continuous Integration	Info on the project
AppleScript NEW 2002/06/10	Anthony Adachi
C Language	Project
	Arien Malec

Unit Testing-Approaches, Tools, Traps

JUnit,CxxTest,DUnit

- **Benefits**
 - Freeware
 - Object oriented
 - Common framework
 - Leading edge thought/technology
- **Problems**
 - Variable level of documentation
 - Support limited/varied
 - Life cycle control

Unit Testing-Approaches, Tools, Traps

Framework⁴

- **TestCase**
 - **setUp** fixture
 - **tearDown** fixture
- **TestSuite**
 - **addTestCase**
 - **run**
- **TestResults**
 - **error**. An anomaly beyond the test scope.
 - **failure**. An anticipated event.

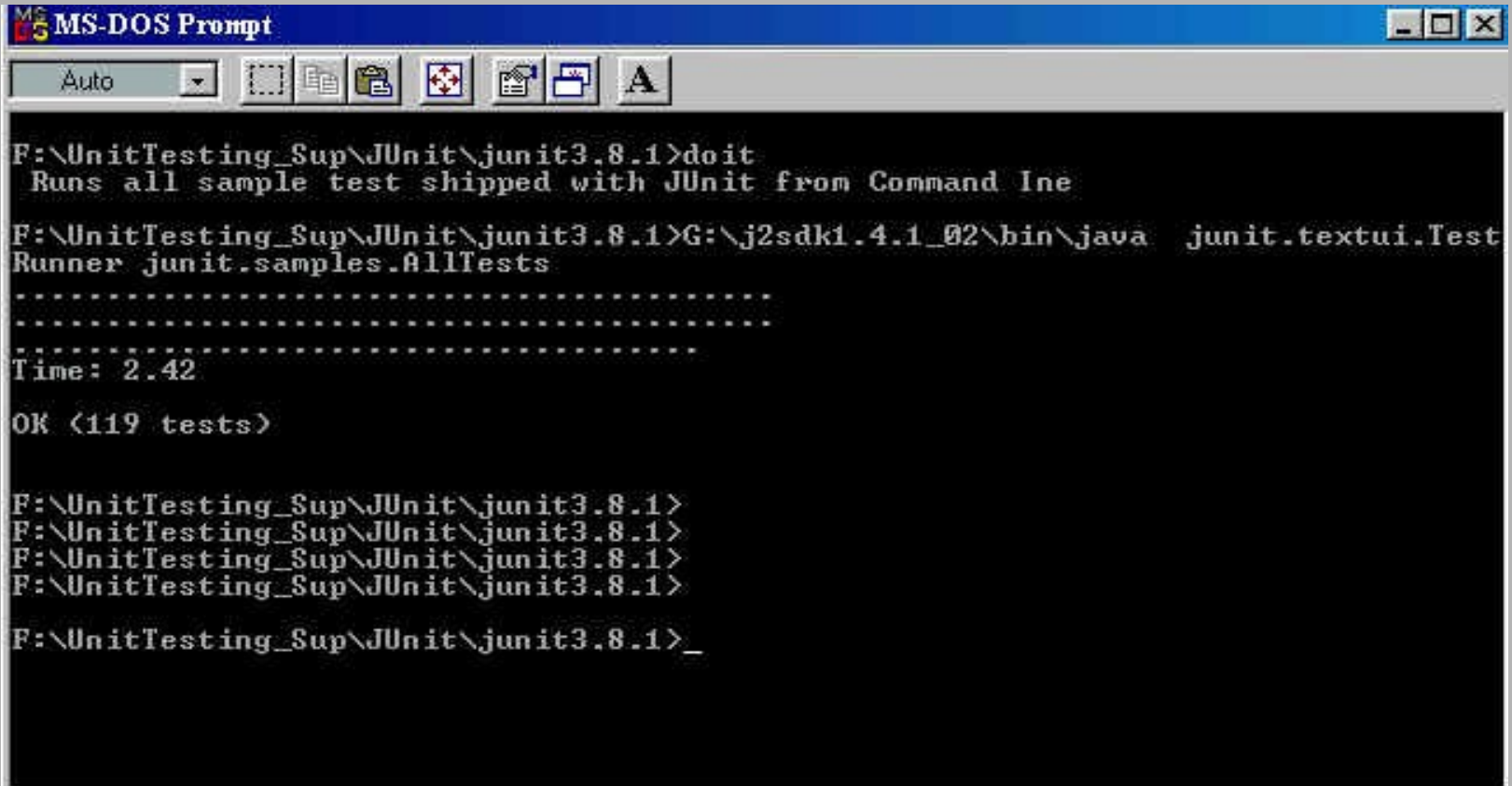
Unit Testing-Approaches, Tools, Traps

JUnit Instrumentation⁵

```
public void testAdd() {  
    double result= fValue1 + fValue2;  
    // forced failure result == 5  
    assertTrue(result == 6);  
}  
  
public void testDivideByZero() {  
    int zero= 0;  
    int result= 8/zero;  
}
```

Unit Testing-Approaches, Tools, Traps

JUnit Command Line



The image shows a screenshot of an MS-DOS Prompt window. The title bar reads "MS-DOS Prompt". The window contains the following text:

```
F:\UnitTesting_Sup\JUnit\junit3.8.1>doit
Runs all sample test shipped with JUnit from Command Line

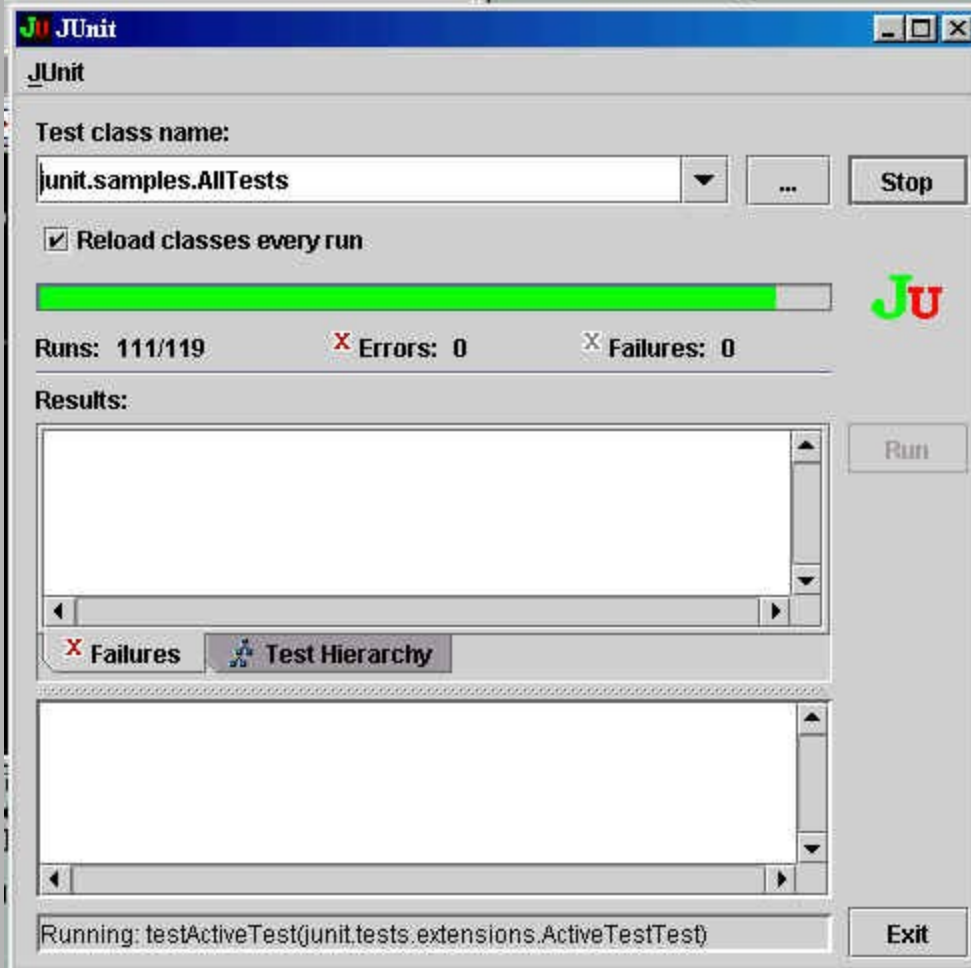
F:\UnitTesting_Sup\JUnit\junit3.8.1>G:\j2sdk1.4.1_02\bin\java  junit.textui.Test
Runner junit.samples.AllTests
.....
.....
Time: 2.42

OK (119 tests)

F:\UnitTesting_Sup\JUnit\junit3.8.1>
F:\UnitTesting_Sup\JUnit\junit3.8.1>
F:\UnitTesting_Sup\JUnit\junit3.8.1>
F:\UnitTesting_Sup\JUnit\junit3.8.1>
F:\UnitTesting_Sup\JUnit\junit3.8.1>_
```

Unit Testing-Approaches, Tools, Traps

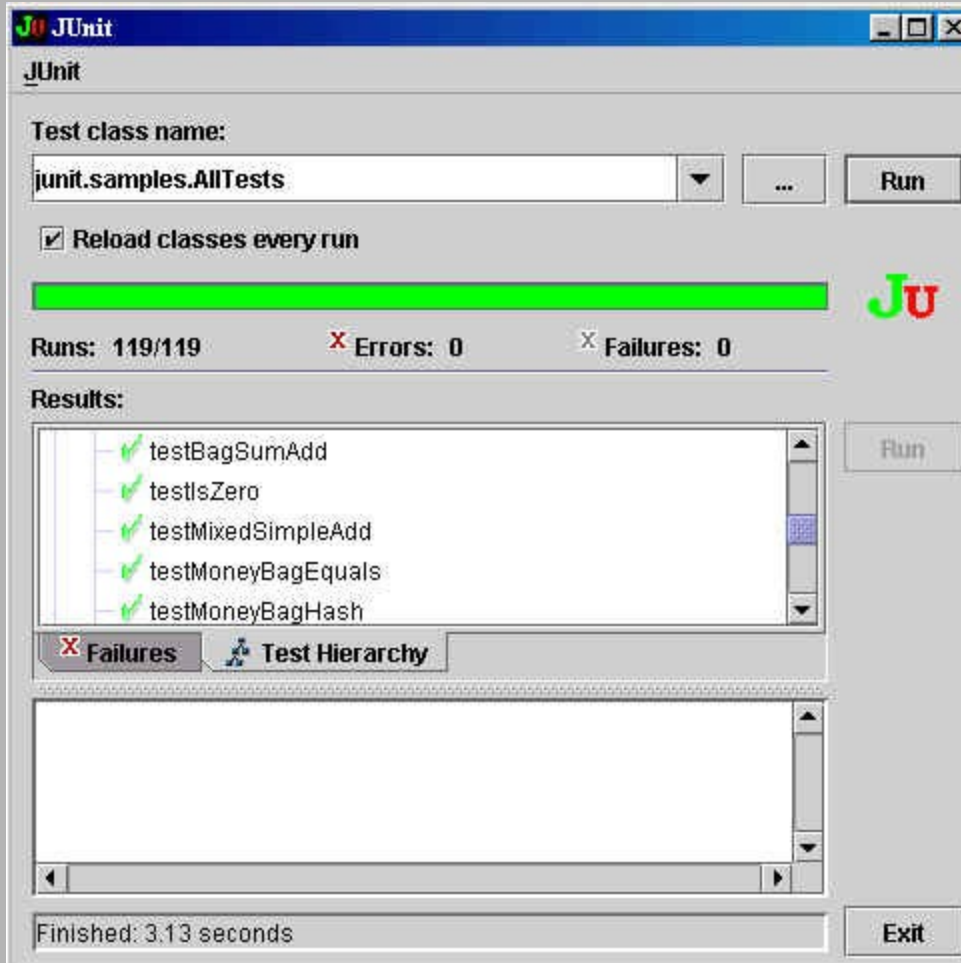
JUnit GUI



- Java
- TestCases
- Test Suites
- Metrics
- Behavioral

Unit Testing-Approaches, Tools, Traps

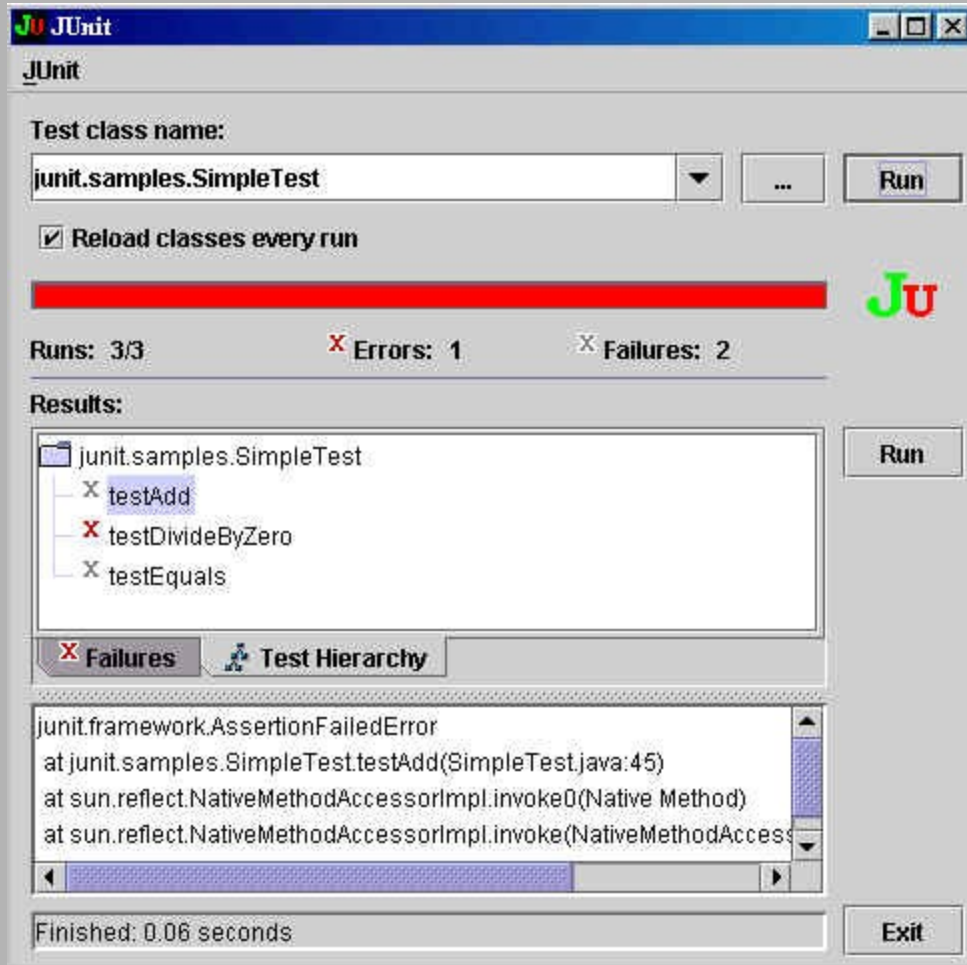
JUnit GUI



- JUnit3.8.1
- Executes from GUI
- Total of 119 test
- All okay

Unit Testing-Approaches, Tools, Traps

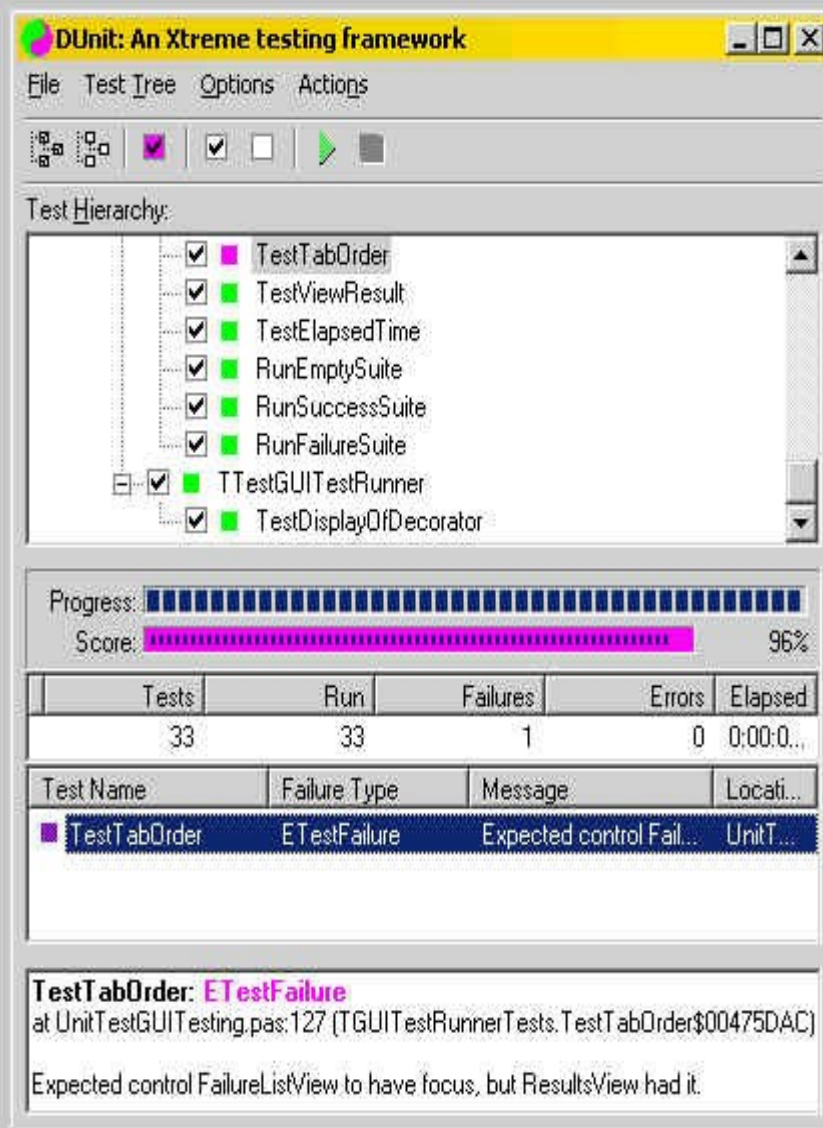
JUnit GUI



- JUnit3.8.1
- Forced divide by zero error
- Assert false failure

Unit Testing-Approaches, Tools, Traps

DUnit



- Delphi
- TestCases
- Test Suites
- Metrics
- Behavioral

Unit Testing-Approaches, Tools, Traps

Dynamic Unit Test

Coverage Analysis

or

Black Box Testing

{Motivation}

Unit Testing-Approaches, Tools, Traps

Dynamic Unit Testing: Motivational Perspective

- **Coverage Analysis**
- **White Box Test**
- **Given Stimulus Yields a Propagation of Observable Responses Within the Borders of the Unit of Interest**

Unit Testing-Approaches, Tools, Traps

Coverage Analysis

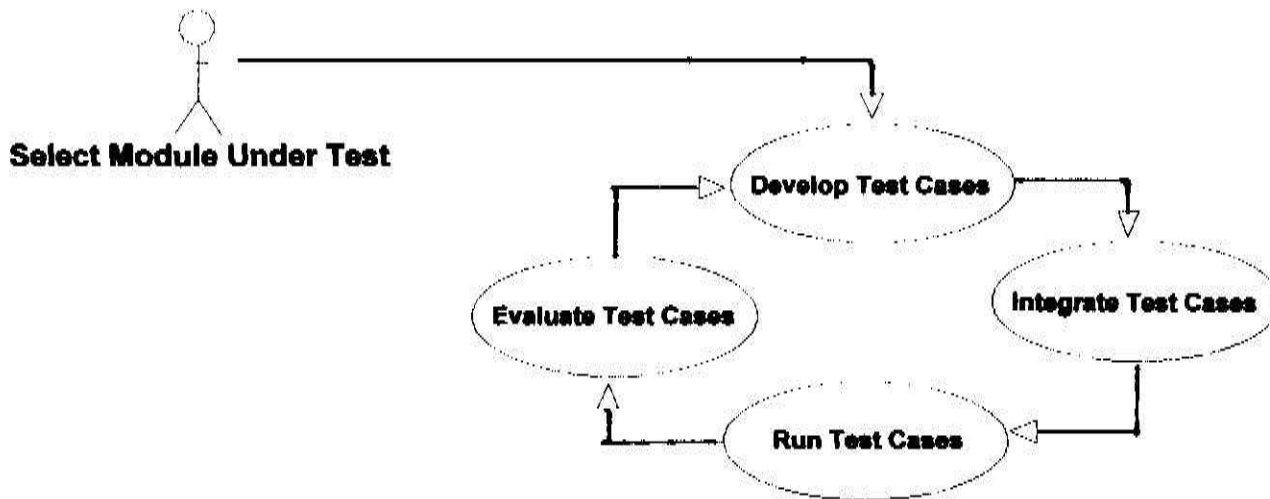
Strategy⁶

- **Path:** Control flow paths through the program, .i.e. all entry/exit points. **Toughest to achieve.**
- **Branch/Decision: P2:** Execute test to assure that every branch alternative has been exercised at least once. **$C2 = \% \text{ of } P2$**
- **Line by Line:** Execute all statements or lines of code at least once. **$C1 = \% P1$**
- **Coverage:** Percent metric in the unit test plan. **$C = C1 + C2$**

Unit Testing-Approaches, Tools, Traps

Dynamic Unit Test Flow

Functional and Coverage Analysis



Unit Testing-Approaches, Tools, Traps

Coverage Tools

- **jcoverage: Freeware for java**
 - May, 2003
 - Allows instrumentation down to line of code
 - Employs Junit
- **quilt, cactus: Freeware for java**
- **clover: \$250 to \$2500+ for java**
- **Roll-your-own:**
 - C/C++
 - Delphi

Unit Testing-Approaches, Tools, Traps

Jcoverage Instrumentation⁷

Path and BranchCoverage

```
public int square(int x) {  
    if(logger.isDebugEnabled()) {  
        logger.debug("x: "+x);  
    }  
    int result=x*x;  
    if(logger.isDebugEnabled()) {  
        logger.debug("result: "+result);  
    }  
    return result;  
}
```

Unit Testing-Approaches, Tools, Traps

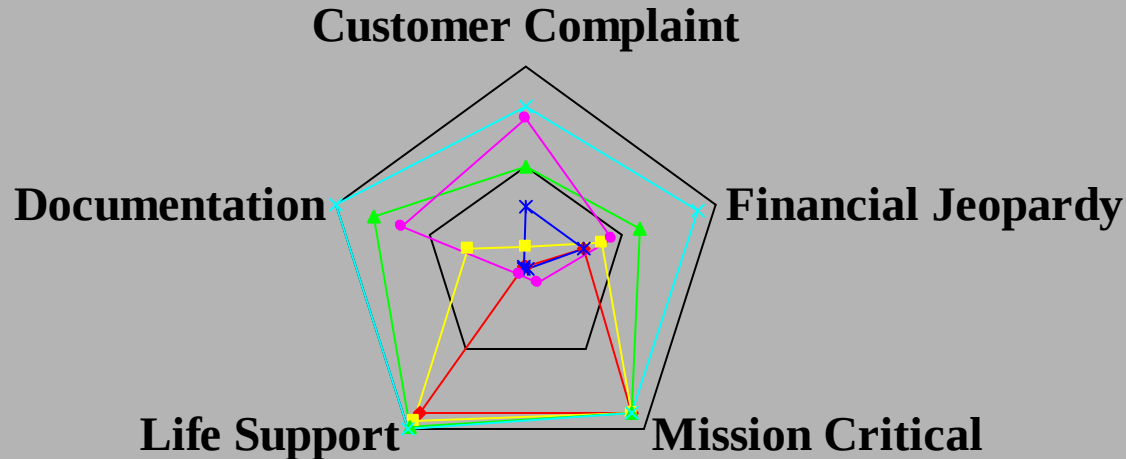
Coverage Analysis

Harness

- **Hooks/instrumentation**
- **Definitions**
- **Stub functions**
- **Test cases**
- **Post test assertions**
- **Administrative functions**
- **Log functionality**
- **Root function**
- **Main function**

Unit Testing-Approaches, Tools, Traps

How Much Energy Should Be Commitment to Unit Testing



Unit Testing-Approaches, Tools, Traps

Unit Test

“How To”

- Unit Test Project Strategy
 - Project Decomposition into Units
 - Requirements Allocation across Units
 - Unit Test Conventions/Templates/Harnesses
- Individual Unit Test Plans
- Individual Unit Test Data
- Individual Unit Test Report
- Project Unit Test Review/Report

Unit Testing-Approaches, Tools, Traps

Unit Test

“How To” Continued

- **Individual Unit Test Plan**
 - **Description**
 - **Risk Analysis**
 - **Mix of Unit Test Tools**
 - **Test Setup/Harness/Environment**
 - **Test Cases**
 - **Test Run Documentation**
 - **Test Report Template/Conventions**

Unit Testing-Approaches, Tools, Traps

Risk Analysis

- **Typical Effect Categories**
 - Catastrophic
 - Critical
 - Major
 - Minor
 - Not Applicable
- **Domain Experts**
- **Employ Requirements Traceability**
- **Documented Rationale**

Unit Testing-Approaches, Tools, Traps

Test Run Documentation

- **Objective evidence**
- **Test description/objective**
- **Test setup/environment/harness**
- **Test step**
- **Closure pass/fail/tester/artifacts**

Unit Testing-Approaches, Tools, Traps

Test Step

- **Unique identifier/requirement/task**
- **Procedural step**
- **Expected results**
- **Actual results**
- **Pass/fail decision**
- **Ancillary artifacts**
 - **Screen shots**
 - **Output files**
- **Tester initials or stamp**
- **Date**

Unit Testing-Approaches, Tools, Traps

Ancillary Artifacts

- **Reference unique test step**
- **Pass/fail status**
- **Tester stamp or signature**
- **Date**

Unit Testing-Approaches, Tools, Traps

Execution Errors

- **Pre-test review process**
- **Mistakes happen/procedure ambiguity**
- **Objective evidence paramount**
- **Handling corrections/changes**
 - Ink
 - Cross through error
 - Enter correction/change
 - Date/tester/reviewer per review process

Anomalies

- **Any observation or event which occurs during the course of a test step that does not conform a defined expectation.**
 - Document as if it were an execution error except that no cross through or reviewer is required.
 - Judgement about how to handle must be made according to previous agreed upon process.
 - It may be acceptable to continue testing with an announced anomaly.

Unit Testing-Approaches, Tools, Traps

Grunt Level Traps

- **No asserts⁸: Just relies on calling a method.**
- **Unreasonable asserts⁸: Assert overkill.**
- **Console based⁸: Using print statements.****
- **Failure to isolate each test⁸: Setup & tear down.**
- **Failure to isolate unit⁸: Stubs/drivers**
- **Ignore bug pathology: Root cause of bug**
- **Put too much burden on unit testing:**

**** May need objective evidence of unit testing.**

Unit Testing-Approaches, Tools, Traps

Code Review/Walkthrough

Bug Finding⁹

- **Effective Against:**
 - Math overflow/underflow
 - Interrupt handling
- **In-Effective Against:**
 - Non-implementation errors
 - Version control errors
 - Resource sharing problems
 - Interface

Unit Testing-Approaches, Tools, Traps

Functional Testing

Bug Finding⁹

- **Effective Against:**
 - Resource mapping
 - Instrumentation problem
 - Interface
- **In-Effective Against:**
 - Math overflow/underflow
 - Reentrance problem
 - Improper variable initialization
 - Interrupt handling/Interrupt suppression
 - Memory allocation/de-allocation

Unit Testing-Approaches, Tools, Traps

Coverage Analysis

Bug Finding⁹

- **Effective Against:**
 - Off by “1”
 - Parameter passing
 - Logic/math processing error
 - Incorrect control flow
- **In-Effective Against:**
 - Data synchronization error
 - Interrupt handling/Interrupt suppression
 - Task synchronization
 - Stack overflow/underflow
 - Resource mapping
 - Instrumentation problem

Unit Testing-Approaches, Tools, Traps

Tactical Traps

- **Silver bullet**
- **Balance resources/time**
- **Numbers game**
 - **Bug count**
 - **Percent tested**
- **Lessons learned/continuous improvement**
- **Owner's ego**

Unit Testing-Approaches, Tools, Traps

Strategic Traps

- **Balance cost with benefits**
- **Defect pathology/mitigation/root cause**
- **Sub-optimize view**
- **Management ego/style**

Unit Testing-Approaches, Tools, Traps

Unit Testing: Conclusion

- **Unit testing is not for the faint of heart.**
- **There must be an organized/balanced plan.**
- **There are valid engineering approaches.**
- **Results will vary.**
- **Not the silver bullet.**

Unit Testing-Approaches, Tools, Traps

Endnotes

1. Boris Beizer, Software Testing Techniques, page 75,76.
2. Grady Booch, Object Solutions, Managing the Object-Oriented Project, pages 11-22.
3. Boris Beizer, Software Testing Techniques, page 550.
4. Ken Beck, “Simple Smalltalk Testing With Patterns”.

Unit Testing-Approaches, Tools, Traps

Endnotes Continued

5. Modified code from SimpleTest.java Junit3.8.1, GNU General Public License
6. Boris Beizer, Software Testing Techniques, page 74,75.
7. Modified code from sample.java from jcoverage ltd, GNU General Public License
8. Glenn Niemyer & Jeremy Poteet, Extreme Programming with Ant, page 127

Unit Testing-Approaches, Tools, Traps

Endnotes

Continued

9. Sean Beaty, “Sensible Software Testing,
Embedded Systems Programming, August,
2000, pages 98-121.

Unit Testing-Approaches, Tools, Traps

WWW Links

- **<http://www.xprogramming.com/software.htm>**: Table of software tools and authors.
- **<http://www.junit.org>**: The information center for junit.
- **<http://www.quilt.sourceforge.net/overview.html>**: Reference site for quilt.
- **<http://www.thecortex.net/clover>**: Clover web site.

Unit Testing-Approaches, Tools, Traps

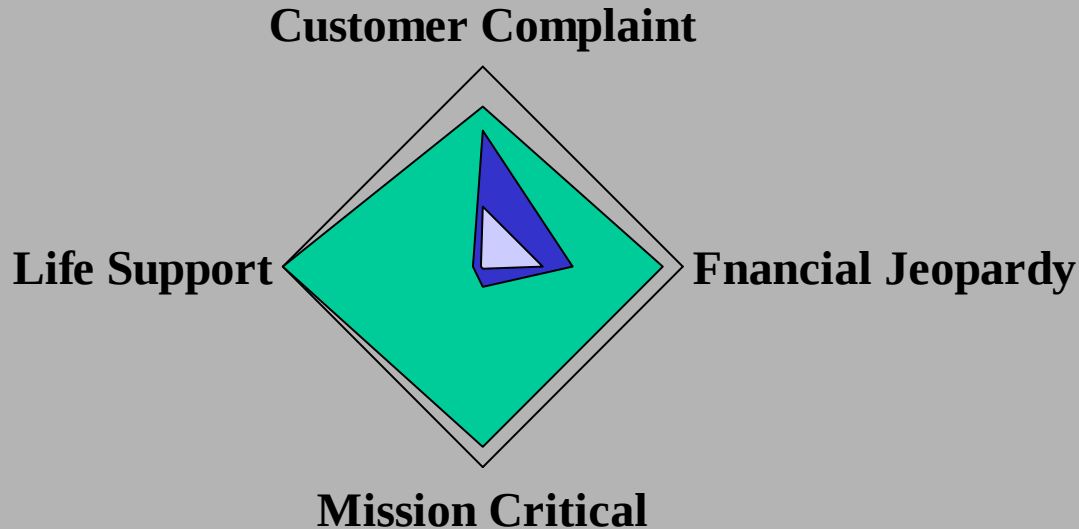
WWW Links

{Continued}

- **<http://jakarta.apache.org/cactus>**: Simple unit test framework extending junit test framework.
- **<http://www.software.org/quagmire>**: Rosetta stone of specifications.
- **<http://qed1.home.mindspring.com>**: Links to standards, papers, and unit test points of interest.

Unit Testing-Approaches, Tools, Traps

Where Should Your Level of Commitment Be to Unit Testing



■ FDA/FAA

■ Data Warehouse

■ Gamer

Unit Testing-Approaches, Tools, Traps

The End

- **Questions**
- **Demo/Lab**
 - **JUnit: Command Line. AllTest**
 - **JUnit: GUI. AllTest. SimpleTest**
 - **Coverage: Jackpot**

Unit Testing-Approaches, Tools, Traps

The End

- **Questions**
- **Demo/Lab**
 - **JUnit: Command Line. AllTest**
 - **Junit: GUI. AllTest. SimpleTest**
 - **Coverage: Jackpot**