

# Ross Collard's Software Unit/Component Testing

## **Duration**

2 days

## **Instructor**

Roland Stens

## **Class Limit**

20 students

## **Prerequisite**

None

## **Price**

On-site:

Please contact SPC for pricing

Public Training:

\$995 (2 days)

\*Discount available for early registration

## **Materials Provided**

- Student manual containing the course slides
- Student handouts with class exercises and class studies

Delivering bug-free software is hard, and the consequences of bugs can be catastrophic. Do you want to deliver more reliable software? Produce work that you can be proud of? And also reduce the hassles of debugging and fixing defects? Then this workshop is for you: it focuses on how to build quality into software and how to unit test software.

The main question addressed by this workshop is: how do I take a software component (or a set of related software components) and test it effectively? Effective component unit testing is often intellectually challenging, and even intriguing. It also is complicated by deadline pressures, incomplete information and shifting requirements. It is an opportunity to excel and worthy of our best attention.

## **Workshop Objectives**

In this workshop, participants will learn how to:

- Test components, modules and programs effectively
- Build reliable software and code defensively
- Modify software in a risk-controlled manner

## **Intended Audience**

System testers & QA engineers, Web developers and Web site administrators, software engineers, project leaders, systems analysts and designers, data base administrators, and users who are involved in testing software systems.

---

For more information on this or other SPC Springboard courses, please visit [www.spcspringboard.com](http://www.spcspringboard.com) or e-mail SPC at [info@spc.ca](mailto:info@spc.ca)

Software Productivity Center  
Suite 460—1122 Mainland Street  
Vancouver, BC V8M 4T8  
[www.spc.ca](http://www.spc.ca)

Toll Free:  
1.877.548.1948

Fax:  
604.689.0141

Vancouver:  
604.662.8181

Toronto:  
416.885.0512

---

TRAINING

# Software Unit/Component Testing

## Instructor

*Ross Collard* is president of Collard & Company, and specializes in software testing and quality assurance. His consulting assignments have included strategic planning for technology, managing large software development projects, and development of software engineering practices. His clients have included Amazon.com, American Express, Boeing, General Electric, Hewlett Packard, IBM and NASA. Ross has an MS in Computer Science from the California Institute of Technology and an MBA from Stanford University.

*Roland Stens* is an independent consultant based in Vancouver who specializes in QA and testing—particularly performance and robustness testing. His broad background in programming, system analysis, database and network management, testing and project/test management enables him to relate to all the issues that play during the software development lifecycle.



## Outline

### I. Introduction to Unit Testing

The Nature of Unit Testing

How Unit Testing fits into the System Development Framework

The Nature and Sources of Defects

Test-Driven Development (TDD)

### II. Challenges and Complications

Practical Limitations of Testing

Trade-Offs: Risk versus Economics

Surfacing “Hidden” Problems

Factors which Influence Testing

The Testing Framework

Documentation is a Drag (in more ways than one)

Responding Adroitly to Changes

Third-Party Software

Component Re-Use

### III. Unit Testing Methodology and Practices

Steps in Software Component Development & Testing

Developing Test Cases

The Test Work Flow

Writing Test Drivers

Analyzing and Understanding Existing Software

Determining the Side Effects of Changes

Debugging Techniques and the Debugging Process

### IV. Unit Test Techniques

Specification-Based Techniques

Sampling Techniques

Model-Based Techniques

Experience-Based Techniques

Selecting the Best Techniques for Your Project

Reviews of Specifications: Checklist of Review Questions

Code-and Data-Based (White Box) Testing

Path Analysis and Decision-Driven Testing

Boundary Value Conditions

Risk-Based Testing

Exception Handling

Regression Testing

### V. Specialized Test Techniques

Complexity and Coverage Analysis

State-Based Testing

Table & Array Testing

### VI. Prevention

Defensive Programming for Defect Prevention and Detection

Adding Diagnostics and Testability to Code

Programming to Facilitate Recovery

Programming to Facilitate Modifications



TRAINING

