

Refactoring Principles: Rejuvenating Object Oriented Software

Duration

2 days

Instructor

Steve Adolph

Class Limit

20 students

Prerequisite

It is assumed that all participants are familiar with basic object oriented concepts and at least one OO programming language such as C++ or Java.

Price

On-site:

Please contact SPC for pricing (contact information on page 2)

Public Training:

\$995 (2 days)

*Discount available for early registration

Materials Provided

- Student handouts with class exercises and class studies

Software does not age well. Often a piece of software starts out with a crisp well-defined vision and a clear distinct architecture, then expedient changes are made and the original clear vision is quickly lost. Software ages rapidly becoming rigid and fragile. Simple changes take excessively long to implement and are frequently unreliable. This seminar will present the concept of refactoring. Refactoring is a collection of techniques for rejuvenating software by improving the structure and performance of existing software without changing the behaviour of the software.

Object technology was supposed to solve the problem of geriatric software. Yet today, we see many legacy object oriented systems that are aging badly and in need of replacement. What went wrong? Software, like any other capital asset requires on going preventative maintenance to reverse the deleterious effects of wear and tear and keep the asset performing at its best. The everyday process of updates, changes, and patches wear and tear at software. Without the software equivalent of preventative maintenance, the software will age and wear out.

Intended Audience

This seminar will directly benefit anyone who is involved with the development of object oriented software, including developers and technical managers.



Instructor

Steve is an experienced software developer with 16 years of software development experience. He has an excellent grasp of the technology issues facing today's companies and works with them as teacher, consultant, and mentor. His software development experience includes call processing software for cellular telephone systems, railway signaling software, direct to plate systems for the graphics art industry, financial software, web based software, and e-commerce software.

Steve has extensive experience in software project planning, software estimation, requirements analysis, software architecture and risk management. Steve has also led many seminars in the area of software design and has given numerous talks at software development conferences. He has written many articles on software development and is a regular contributor to Software Development magazine. He is currently working on his first book on Use Case Patterns.

TRAINING

Refactoring Principles

Outline

Refactoring Overview

- Refactoring definition
- Refactoring versus rework
- Pay as you go
- Use no hooks

Software Engineering Basics and Software Quality

- Modularity
- Coupling
- Cohesion

How Good Software Goes Bad

- Practices that prematurely age software
- Big ball of mud
- Blobs
- Lava flows
- Spaghetti code

OO Design Principles

- Dependency inversion
- General Responsibility Assignment Patterns (GRASP)

Refactoring Principles

- Refactoring to generalize
- Refactoring to specialize
- Aggregate abstraction

Low Level Refactorings

- Large classes
- Duplicate code
- Large methods
- Feature envy
- Utility methods
- Complex conditionals
- Subclass proliferation
- Co-occurring parameters

Design Patterns and Refactoring

- Identifying what varies in an application

Refactoring and the Software Development Lifecycle

- Iterative software development processes
- Xtreme Programming
- Refactoring and design
- Refactoring and gold plating
- When not to refactor

For more information on this or other SPC Springboard courses, please visit www.spcspringboard.com or e-mail SPC at info@spc.ca

Software Productivity Center
Suite 460—1122 Mainland Street
Vancouver, BC V8M 4T8
www.spc.ca

Toll Free:
1.877.548.1948

Fax:
604.689.0141

Vancouver:
604.662.8181

Toronto:
416.885.0512



TRAINING

