Refactoring: Improving the Design of Existing Code

Martin Fowler

Chief Scientist, ThoughtWorks, Inc.
fowler@acm.org

Abstract. Almost every expert in Object-Oriented Development stresses the importance of iterative development. As you proceed with the iterative development, you need to add function to the existing code base. If you are really lucky that code base is structured just right to support the new function while still preserving its design integrity. Of course most of the time we are not lucky, the code does not quite fit what we want to do. You could just add the function on top of the code base. But soon this leads to applying patch upon patch making your system more complex than it needs to be. This complexity leads to bugs, and cripples your productivity.

Introduction. Refactoring is all about how you can avoid these problems by modifying your code in a controlled manner. Done well you can make far-reaching changes to an existing system quickly, and without introducing new bugs. You can even take a procedural body of code and refactor it into an effective object-oriented design. With refactoring as part of your development process you can keep your design clean, make it hard for bugs to breed and keeping your productivity high.

In this tutorial we'll show you an example of how a lump of poorly designed code can be put into good shape. In the process we'll see how refactoring works, demonstrate a handful of example refactorings, and discuss the key things you need to do to succeed.

Duration: Half Day

Level: Intermediate

Presenter's Resume

Martin Fowler is the Chief Scientist for ThoughtWorks, Inc., a leading custom e-business application and platform development firm. For a decade he was an independent consultant pioneering the use of objects in developing business information systems. He's worked with technologies including Smalltalk, C++, object and relational databases, and EJB with domains including leasing, payroll, derivatives trading and healthcare. He is particularly known for his work in patterns, the UML, lightweight methodologies, and refactoring. He has written four books: Analysis Patterns, Refactoring, the award winning UML Distilled, and Planning Extreme Programming.

Martin has been giving this tutorial in various forms since around 1997. It has been presented at a wide range of commercial and academic conferences including OOPSLA (where it was one of the top 5 tutorials in 1999), Software Development, and JavaOne.

The tutorial is based on the early chapters of Martin's book "Refactoring: Improving the Design of Existing Code".

D. Wells and L. Williams (Eds.): XP/Agile Universe 2002, LNCS 2418, p. 256, 2002.