From Code to Architecture

Summary

The code we write has a tremendous impact on our software architecture. In fact, the code is our architecture. Not all of it, of course, but a good share of it. The problem is that we often don't fully comprehend the architectural impact of our code at the time we create it. One poorly designed class or method can severely impact the resiliency, stability, extensibility, and maintainability of your software.

In this session, we examine how code impacts architecture. Extensive discussion is devoted to modularizing units of deployment, and how code can be created and organized to create more flexible, reusable, maintainable, extensible, and testable software components. Topics of discussion range from coupling between classes to the dependency structure between your modules of deployment. Examples gleaned from real world scenarios will be used in a pragmatic case study that examines the evolution of a system from its initial deployment through a number of refactorings that positively influenced the architectural resiliency of the final product. A suite of heuristics will be introduced that allow you to apply many of these concepts immediately.

Audience

From Code to Architecture will benefit **software architects** and **software developers**. They will learn new ways to structure large codebases, resulting in extensible and architecturally resilient software. Using an extensive case study, numerous examples will be shown that illustrate how a system's architecture can evolve while maintaining a high degree of stability. A suite of heuristics will be presented as we progress through the presentation, offering attendees ideas that can be applied immediately to their projects.

Content Outline (90 minutes)

1. **Introduction (15 minutes)**
   Goal: Define software architecture using popular definitions by Grady Booch and Ralph Johnson
   Topics Include:
   - Architecture as a structural entity
   - Architecture as a social construct
   - Subsystems and Interfaces
2. **Logical versus Physical Design (25 minutes)**
   Goal: Explore the differences between logical and physical design. Examine how logical design influences physical design and vice versa.
   Topics Include:
   - Logical Design: Maintenance and Extensibility
   - Physical Design: Modularity and Reuse
   - Physical Units of Deployment
   - Component Relationships
   - Direct versus Indirect Relationships
3. **System Case Study (50 minutes)**

Goal: Examine a system's initial architecture and explore through numerous refactorings how the architecture is modified. A suite of heuristics will be presented as each example is built on the previous, offering an experience very much like one would experience on a real world project.

Topics Include:

- Layering Modules
- Concrete versus Abstract Components
- Leveling Relationships
- Separating Abstractions
- Inverting Component Relationships

Presenter Background

Kirk is the Chief Technology Strategist at QWANtify, Inc., where he leads based on his firm belief in the pragmatic use of technology. In addition to his work on enterprise development projects, Kirk shares his experiences through courseware development and teaching, writing, and speaking at seminars and conferences. Kirk has provided training and mentoring to thousands of software professionals, teaching courses on object-oriented development, Java, software architecture, software process, and UML.

Kirk is the author of Java Design: Objects, UML, and Process, and the founder of www.extensiblejava.com, a growing resource of design pattern heuristics that emphasize greater component modularity in large scale enterprise software projects. He is also the creator of JarAnalyzer, a utility for identifying and managing the physical dependencies between .jar files. His personal website is www.kirkk.com.

Tutorial History

This tutorial is presented on the 2006 NoFluffJustStuff Java Software Symposium Tour, and was presented at Software Development Best Practices in 2005. It's origins are traced back to work conducted on multiple projects over a five year period where special emphasis was placed on minimizing dependencies between software components. The material is also based on my upcoming book, *From Code to Architecture.*

Presenter Contact Information

Kirk Knoernschild
email: kirk@qwantify.com
mobile: (608) 225-8647