Book review

Beyond Software Architecture: Creating and Sustaining Winning Solutions
by Luke Hohmann

Addison-Wesley, 2003
ISBN: 0-201-77594-8
352 Pages

In the Foreword to *Beyond Software Architecture*, Martin Fowler says that Luke Hohmann "talks about the kinds of things that are often sadly omitted from most architectural discussions — yet are every bit as important." And that is precisely what makes Hohmann's book a true gem.

Architecture is more than structure and design — in fact, it is much more. But it is also difficult to describe precisely; there is no one universally accepted definition of architecture. IBM Rational Unified Process® says that architecture encompasses the structure, abstraction, and decisions made about a system, and that you need multiple views of the software under construction to understand an architecture. In short, architecture is big. For the many people who think architecture relates only to the structural integrity of the systems they produce, Hohmann's book will be an eye-opener. For those who already understand what architecture comprises, the book will provide new perspectives and knowledge that will help them deliver winning software solutions.

The book is only 310 pages long, and it has something worthwhile to say on almost every page. You could dip in here and there and use it as a reference, but I urge you to read it from cover to cover. The writing style flows easily, and a fast reader can get through it in a couple of hours. It is well-designed for learning: Each of the sixteen chapters ends with a summary of key points, a list of items to check for on your projects, and a set of questions to help deepen your understanding of the material and apply it.

Architecture issues are business issues

Hohmann is a consultant who has clearly been through many different experiences and learned from them. A marvelous writer, he relates those experiences in the form of lessons he learned. Here's the primary lesson:
The technical issues of architecture cannot be separated from business issues. Although the book is written primarily about software-for-sale systems, this lesson applies as well to software developed for in-house use. To explain the lesson more fully, he examines architecture from two viewpoints: tarchitecture and marketecture.

Tarchitecture is the traditional, technical view of architecture. It represents decisions about the structure and technical qualities of the system. When we talk of architectural patterns, we are usually referring to the tarchitecture. But marketecture is just as important — if not more important — to the overall success of your project. Marketecture is the business perspective of the system's architecture. It encompasses many of the things we traditionally ignore but shouldn't: business and licensing models, technology in-licensing (i.e., licensing technology to put into your product), managing brands, installation, upgrades, logs, and release management.

**Detailed information for project managers**

Hohmann's examination of these often overlooked considerations should make product managers especially happy. Few books on the market describe the product manager's job as clearly and succinctly as this one does in Chapter Two, "Product Development Primer." I only wish I'd had this chapter as a guide during my brief tenure as a product manager. Most likely, I would have been more successful, and I certainly would have been more confident. Hohmann tells the reader not only what it means to be a successful product manager; he also provides details about how to fulfill the responsibilities of that role and steer development teams in productive ways.

The two chapters on licensing illustrate the tight coupling between marketecture and tarchitecture. In the first of these chapters, Hohmann begins by laying out common software business models. He then discusses many licensing methods and how they apply to the business models, and then offers advice on how to decide which method is best for you. He even has a short section to help you decide how to price your product. The second chapter on licensing focuses on technology in-licensing. Even experienced architects who have dealt with technology licensing issues will find very useful information here.

This book is not designed to replace the many excellent books on technical aspects of software architecture; instead, it is an outstanding companion for any of these books. If I were responsible for developing a new software product and couldn't hire Luke Hohmann to help my team, I'd certainly make sure that several copies of this book were available for my team to refer to regularly!

-Gary Pollice
Professor of Practice
Worcester Polytechnic Institute