Software Security: Building Security In

Gary McGraw, Addison-Wesley Professional, 2006, \$49.99, ISBN: 0321356705

Over the years, I have read several books covering software security from a system or programming language perspective. While most of them provided excellent overviews, I was hoping eventually to see a holistic approach. *Software Security* is just the kind of book I had in mind and is one of the best introductions to software security that I have seen.

One phrase summarizes the content of the book: “Software security is not security software.” One of the major errors we make in development is addressing software security by adding features; this is cost efficient in the short term, but raises major issues in the long run. In his book, McGraw shows that security is not a feature that can be added to extend the functionality of software but is an essential building block and key architectural design characteristic of reliable software.

Although the book is written at a high level of abstraction, going beyond simple code vulnerabilities and examples, the multiple sidebars with anecdotes provide illustrations and make it easy to read. The first part of the book defines the discipline and introduces the notion of risk management. It covers issues such as risk mitigation, risk measures, operations, and the major stages of applying risk management in practice. The second part covers touch points for software security: code review, architectural risk analysis, penetration testing, abuse cases, security requirements, security operations, and external analysis. The final section deals with enterprise-level security development cycles and the importance of knowledge-based management schemes for such purposes.

At first, the content of the book might seem dry and targeted to less technically oriented readers. For those more interested in technical and programming issues, however, my favorite chapter is the fourth, which addresses automated code review with the Fortify security tool. The author is one of the developers of this tool, and a CD containing a sample scenario to be worked out by the reader accompanies the book.

There is a final jewel at the end: an annotated bibliography covering most of the essential readings from academia and industry. The contents of the book are

intrinsically tied to both of these areas, and McGraw manages to provide a common view on software security from both perspectives. I highly recommend this book to all readers wishing to build security into their software.

—Radu State

Refactoring Databases: Evolutionary Database Design

Scott Ambler and Pramodkumar Sadalage, Addison-Wesley Professional, 2006, \$49.99, ISBN: 0321293533

The simplest way to explain this book’s evolutionary approach to database maintenance is with an analogy to the electrical rewiring of a home while keeping the lights on. The book discusses a number of basic operations, serving as effective and safe techniques for database maintenance. These techniques

cover not only the standardized steps of database schema transformation and related code refactorings, but also a number of relevant supporting techniques of interest to practitioners working with complex databases.

Overall, the book’s evolutionary approach to database application system maintenance jointly addresses software and data design. It explains 70 operations, with illustrations and code snippets, and is organized into 11 chapters.

Chapter 1 presents the general idea of evolutionary development. Chapter 2 introduces refactoring concepts in database situations. The third chapter explains recommended processes, and the next ties refactoring to deployment. Chapter 5 serves as the main entry point for the remainder of the book, succinctly addressing database refactoring strategies and providing online resources, discussion lists, and Web sites for updates. Chapters 6 through 9 make up the main reference section, presenting, respectively, structural, data quality, referential integrity, and architectural database refactorings. Chapter 10 is an overview of related code refactoring, and the final chapter discusses elementary database transformations.

This book is recommended for database system implementers, database design experts, and advanced students of database design, as it provides a unique perspective on database schema changes in a manner preserving application semantics. It is not a quick read, but it is an approachable one.

—Vladan Jovanovic

Reprinted from *Computing Reviews*, © 2006 ACM, <http://www.reviews.com>