Component-Oriented Programming

Andy Ju An Wang and Kai Qian, Wiley-Interscience, 2005, \$89.95, ISBN: 0471644463

This book, which describes a hands-on, experimental approach to COP (component-oriented programming), could be used in a course on CBSE (component-based software engineering). Chapters 1 and 2 place COP in context and present a framework for organizing treatment of the various component technologies covered in subsequent chapters. This framework introduces three complementary models: a component model for developing components, a connection model for assembling components, and a deployment model for producing working applications.

Each of the remaining chapters covers a particular component technology and a supporting development or deployment environment: JavaBeans (BDK [Beans Development Kit] 1.1 and Bean Builder 1.0); Enterprise JavaBeans (J2EE [Java 2 Platform, Enterprise Edition] 1.4.x); CORBA (OpenCCM 0.6); .NET (Visual Studio); OSGi (Open Services Gateway initiative); JES (Java Enterprise System 2.0); and Web services (Apache Tomcat Axis and .NET IIS [Internet Information Server]).

Each chapter surveys a technology's history and architecture and describes its component, connection, and deployment models. The discussion is tied closely to concrete examples and includes working code and step-by-step instructions for configuring the selected environment and for building and deploying an application. Each chapter includes several labs, self-review questions, exercises, and key references.

The book misses an opportunity to raise concerns that could be used to further explicate the technologies and assess their relative merits. How does each technology support the separation between a component's public interface and private implementation? Is meta-information about a component stored in the component or in the environment, and how is this information accessed? There is also little attempt to compare technologies. Indeed, the chapters are themselves components, with few references to the other technologies covered in the book.

I can recommend this book for the survey it provides of major COP technologies. Its tutorial style and level of detail allow one to sample and even experiment with

these technologies, although it is not a comprehensive reference to any of them.—*Michael Laszlo*

Agile Estimating and Planning

Mike Cohn, Prentice Hall PTR, 2005, \$44.99, ISBN: 0131479415

The principal contribution of this book is that it helps to destroy the myth that agile development need not be estimated or planned. This myth, which is widespread among inexperienced software engineers, is detrimental to the construction of software that meets customer expectations in terms of functionality and deadlines. Do not forget that these expectations have to be met whether or not the development life cycle is agile.

Mike Cohn discusses the different aspects of estimating and planning a project following an agile life cycle from a practical perspective. He gives developers a range of real alternatives, each with its pros and cons, rather than a magic (and therefore imaginary) wand for estimating and planning such projects. In many chapters, the discussion of alternatives is accompanied by advice from the author backed by experience. Readers can therefore appreciate the strengths and weaknesses of the different approaches and make the right decision for their case.

The book is divided into seven parts. Part 1 discusses why estimating and planning are important and common problems encountered when doing so. Part 2 addresses estimation and introduces two measures of size. Part 3 tackles project preplanning issues in an agile environment. Part 4 looks at the essential steps of planning a release and iteration in an agile project. Part 5 deals with monitoring a release and an iteration plan. In Part 6, Cohn explicitly groups reasons why the agile estimating and planning strategies presented in the book are successful for project planning. Part 7 presents a fictional case study that summarizes the key points of the book and illustrates how to use the techniques.

Despite the practical focus of the text, Cohn gives a number of references to software engineering literature justifying the different claims discussed in the book. This raises readers' confidence in the content and makes it easier for them to pursue the subject further. —*A. Moreno*
Reprinted from *Computing Reviews*, © 2006 ACM, <http://www.reviews.com>