

Software Architecture Book Provides Practical Guidance about Documentation

Erin Harper

The construction of a successful system depends on an appropriate architecture—the way the system is decomposed into parts and the ways in which those parts interact. Yet even the best architecture is useless if others cannot understand it, making the documentation of software architectures crucial.

As part of the SEI Series in Software Engineering,¹ a new book called *Documenting Software Architectures: Views and Beyond* has been written by Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Robert Nord, and Judith Stafford.

Intended as a handbook for practitioners in the field, this book helps readers decide what information about an architecture is important to document and provides guidelines, notations, and examples for documenting that information. The authors provide an extended example of a software-architecture documentation package that demonstrates the concepts covered in the book. “Documenting an architecture is a formidable task without firm guidance,” Stafford says. “We break it down into three basic viewtypes, then we allow refinements, called *views*, that are based on architectural styles.”

Because the same system can be seen in many different ways, choosing the right views depends on who the audience will be and what they will need to do with the information. “Different stakeholders don’t have the time or energy to sift through a lot of unnecessary information,” Stafford remarks. For example, a project manager may be interested in the system’s overall purpose and constraints, but not in the detailed design. Members of the development team, on the other hand, may be given responsibility for elements they did not implement, and will need to know the details of those elements and how they interact.

A simple three-step process for choosing the best views to document for a particular system is included in the book, along with detailed templates for practitioners to use for documenting interfaces, views, and additional information beyond the views.

Documentation beyond views consists of three major aspects, which the authors summarize as “how/what/why”: *how* the documentation is organized (consisting of a roadmap and a view template); *what* the architecture is (consisting of a short system overview); and *why* the architecture is the way it is (consisting of background information, external constraints, and the rationale for decisions). All of these aspects of documentation are important. “Documentation speaks for the architect, today and 20 years from today,” says Clements. Many

¹ <http://www.sei.cmu.edu/products/publications/sei.series.html>

people may need to look at the documentation after a system is created: those maintaining or updating the system, those trying to expand it, or those building similar systems.

Although many individuals and organizations are beginning to understand the importance of software architecture, there is little practical guidance on how to capture an architecture independent of language or notation. “Box-and-line drawings aren’t enough. They’ve been masquerading as architecture for years now, but they need to be supported by additional information and explanations,” Clements says. Based on their years of research and experience in the field, the authors provide needed guidance on creating a complete documentation package.

More information about this book can be found on the SEI Web site at <http://www.sei.cmu.edu/ata/books.html>.

For more information, contact—

Paul Clements

Phone

512-453-1471

Email

pclement@sei.cmu.edu

World Wide Web

<http://www.sei.cmu.edu/ata/books.html>