

## How This Book Is Organized

We have organized the contents of this book around five thematic areas: overviews, enterprise applications, systems, end-user applications, and programming languages. There is an obvious, but not deliberate, lack of chapters on desktop software architectures. Having approached more than 50 software architects, this result was another surprise for us. Are there really no shining examples of beautiful desktop software architectures? Or are talented architects shying away from an area often driven by a quest to continuously pile ever more features on an application? We are really looking forward to hearing from you on these issues.

### Part I: On Architecture

Part I of this book examines the breadth and scope of software architecture and its implications for software development and evolution.

Chapter 1, *What Is Architecture?*, by John Klein and David Weiss, defines software architecture by examining the subject through the perspectives of quality concerns and architectural structures.

Chapter 2, *A Tale of Two Systems: A Modern-Day Software Fable*, by Pete Goodliffe, provides an allegory on how software architectures can affect system evolution and developer engagement with a project.

### Part II: Enterprise Application Architecture

Enterprise systems, the IT backbone of many organizations, are large and often tailor-made conglomerates of software usually built from diverse components. They serve large, transactional workloads and must scale along with the enterprise they support, readily adapting to changing business realities. Scalability, correctness, stability, and extensibility are the most important concerns when architecting such systems. Part II of this book includes some exemplar cases of enterprise software architectures.

Chapter 3, *Architecting for Scale*, by Jim Waldo, demonstrates the architectural prowess required to build servers for massive multiplayer online games.

Chapter 4, *Making Memories*, by Michael Nygard, goes through the architecture of a multistage, multisite data processing system and presents the compromises that must be made to make it work.

Chapter 5, *Resource-Oriented Architectures: Being “In the Web”*, by Brian Sletten, discusses the power of resource mapping when constructing data-driven applications and provides an elegant example of a purely resource-oriented architecture.