
PREFACE

Building software has almost always involved fitting together products and organizations as well as developing code. Software architecture is fundamental to both activities, especially today. For example, an ordinary business transaction will traverse many layers of software architecture, leveraging shared platforms such as the Internet, client browsers, Web servers, business logic components, security systems, and back-end databases. In this environment, many partners must not only agree on a core set of interfaces and standards, but they must also agree on how to use those standards. Partners must also agree on the value they will add and receive for their contribution. All these agreements must remain workable and stay in place in the face of rapid changes in technologies, re-alignments of partners, shifts in business goals and requirements, as well as the ever-present mergers and acquisitions. If these agreements do not remain in place, the product and its architecture may fail, causing pain for developers and customers, as well as their managers and sponsors.

This book focuses on the interrelationship between software architecture and the organization. Software architecture serves as a framework that defines and orders not only the technical interactions needed to develop and implement a product, but also group and personal interactions. The ability of software architecture to fulfill this role over time relies on organizational factors.

It has long been observed that the structures of architectures and the organizations that build and use them influence one another. A close look reveals an extensive and complex relationship. Real-life architecture often is far removed from the intended structure, including hidden chunks of software,

odd connections, hard-coded “shortcuts,” missing pieces, and other irregularities. The same types of surprises come from an organization’s culture, its people, their beliefs, abilities, and behaviors. In practice, architecture and organization form a sensitive and highly volatile matrix. Done right, organization and architecture can deliver great value; failure can melt the core of the enterprise.

We have written this book to help people who have a critical stake in the success of software architecture understand and overcome the challenges of architecture and organization. These stakeholders form a large interdependent group that is getting larger as software products cross more organizational boundaries in their development and use. This group includes people who manage, develop, implement, maintain, acquire, and use software architecture. Each stands to benefit from a better understanding of how software architecture and organization interact. For example, partnering skills can decrease the time it takes for developers to find out about changes to a release of an architecture or platform, and increase the chances that they can negotiate to restore features that are critical to their continuing use of the architecture. Without these skills, the architect would soon lose customers; the customers would soon be maintaining a lot more code and creating a lot less product.

Our book is based on more than five years of research within some of the country’s best-known large-scale software development organizations and numerous workshops, as well as our work architecting product lines and implementing architectures for small, medium, and very large organizations. We also drew on work in other disciplines, especially organizational development. Our research yielded a model composed of five organizational principles that affect software architecture success—Vision, Rhythm, Anticipation, Partnering, and Simplification (VRAPS). We call this the VRAPS Model.

Together, the VRAPS principles provide a model you can use to get your arms around what to do and to improve your personal and corporate ability to get lasting value from products and ventures that depend on software architecture. The VRAPS model will help you to organize and interpret your observations and relate them to practices and patterns that others have used successfully. You can also use the model and principles to identify strengths and weaknesses, to communicate insights, and to encourage actions across roles, boundaries, and levels within and external to your organization.

You can take several paths through our book:

- ▶ You can get a quick overview of the book in Chapter 1. Then go to Chapter 8 to find a case study that illustrates how the VRAPS principles

worked within Allaire Corporation, as it grew from a small startup to become a leading provider of tools for building Internet applications.

- ▶ You can find out more about each VRAPS principle in Chapters 3 through 7. After defining and describing the principle, these chapters provide criteria to help you gauge how well your organization applies the principle. Patterns, stories, and antipatterns provide practical guidance about what to do and what not to do to benefit from the principle.
- ▶ You can get a detailed understanding of the VRAPS model and how it relates to other work in Chapter 2. Chapter 9 provides a real-world illustration of how to use the model, along with nine specific templates, tools, and guidance you can use to assess your organization and compare it with others. We describe how the templates were used in a commercial benchmark.

We invite you to read on and visit our Web site at www.VRAPS.com.

David M. Dikel
David Kane
James R. Wilson