



## Book Review

### ***Software Architecture: Organizational Principles and Patterns***

by David M. Dikel, David Kane, and James R. Wilson

Prentice-Hall, 2001  
ISBN: 0-13-029032-7  
Cover Price: US\$49.00  
(308 Pages)

Dikel, Kane, and Wilson have written quite a useful book for software architects, project managers, or IT department managers. It is not about the tools and techniques for software architecture, but about the many organizational and social implications of taking (or not taking) an architectural focus for developing software.

The authors have organized their thoughts around a model (no, not a UML model) called VRAPS, which stands for *Vision, Rhythm, Anticipation, Partnering, and Simplification*. For each of these five, deeply interconnected principles, they have compiled advice in the form of organizational patterns (i.e., successful practices), a few anti-patterns (i.e., things that really do not work), and also criteria to help practitioners gauge how well they fare against the VRAPS model.

Here are the five principles in brief summary.

### **Vision**

*"Vision is the mapping of future values to architectural constraints, as measured by how well the architecture's structures and goals are clear, compelling, congruent and flexible."*

Developing an architectural vision permits a better definition of the scope of the architectural endeavor, and sets a reasonable level of expectations among the stakeholders, both internal and external. It also allows early detection when expectations fail to align, and makes normally tacit knowledge about the architecture visible to its users.

### **Rhythm**

*"Rhythm is the recurring, predictable exchange of workproducts within an architecture group and across their customers and suppliers."*

- [▶ subscribe](#)
- [▶ contact us](#)
- [▶ submit an article](#)
- [▶ rational.com](#)
- [▶ issue contents](#)
- [▶ archives](#)
- [▶ mission statement](#)
- [▶ editorial staff](#)

Rhythm is about the lifecycle and its cycles and iterations, the visibility and predictability of those cycles and iterations, how the organization will evolve an architecture over time, and how the various components will unfold and integrate. It is about timing, and also about the content of a release and the quality of that content. It deals with synchronization among stakeholders. It is about reevaluation.

## **Anticipation**

*"Anticipation is the extent to which those who build and implement the architecture predict, validate and adapt the architecture to changing technology, competition and customer needs."*

What is the architect's most difficult objective? How to balance long-term vision with immediate needs, in the context of high-technology churn? How to avoid the nasty track of too much "bleeding" edge, or avoid tunnel vision? Here you see how the various principles start influencing each other: A healthy rhythm will force regular review and reevaluation of the vision, for example.

## **Partnering**

*"Partnering is the extent to which architecture stakeholders maintain clear, cooperative roles and maximize the value they deliver and receive."*

Do not do it all by yourself -- build cooperative organizations, internally and externally. Maximize the ultimate value for critical stakeholders, not the amount of local invention and development. Make sure there is a clear and compelling agreement between the various stakeholders, in the sense of the "win-win" approach advocated by Barry Boehm and others.

## **Simplification**

*"Simplification is the intelligent clarification and minimization of both the architecture and the organizational environment in which it functions."*

The architecture continues to be used, but there is a cost to using it, and therefore its complexity must be reduced. No fluff, no overgeneralization of the problems leading to overly simplified solutions. The architects must understand the essential minimal requirements and build them into the core elements of the architecture. The overall business case for evolving the architecture remains clear.

## **VRAPS and the Rational Unified Process**

All these principles, criteria, and (anti) patterns resonated very clearly for me, based on my own personal experience and what I have witnessed in the software development field over the last twenty years. I also caught myself contrasting this model with the Rational Unified Process® or RUP® product, and trying to gauge how well the RUP's guidance would fare with regard to the VRAPS.

**Vision:** Vision is clearly articulated in the RUP's *Software Architecture Description Document*, which must tie to the overall project *Vision Document*. Many of the activities related to architecture in the RUP Elaboration phase contribute to this vision.

**Rhythm:** The RUP has a three-tier "tempo." The initial development cycle brings a clear focus on architecture through its Elaboration phase. Subsequent evolution cycles allow revisiting an Elaboration phase to evolve, refine, and simplify the architecture. Each cycle is decomposed into iterations, which provide organization-wide synchronization points, focusing on a release of products, initially leading to an architectural baseline, then evolving to a gradually more complete product of increasing quality. Then, in each iteration, regular builds force both a focus on value and closure on concrete issues.

**Partnering, anticipation, and simplification** are embedded in the RUP, mostly in the various activities and guidelines related to component-based development, architectural reviews, and architectural design. Again, these are supported by the fundamental iterative nature of the RUP, which brings these issues to the table at least once per cycle.

Altogether, I find *Software Architecture: Organizational Principles and Patterns* good companion reading for RUP-literate people. Practical and easy to read, it emphasizes the *social aspect* of developing software architecture. Architecture is not just about building the right UML models, with three, five, or seven views; it is also about people. As I used to tell the architecture team on an air-traffic control project I worked on: Fifty percent of the job of an architect is communications -- communications with other teams, internally, with the customers, with suppliers, with other parts of the company -- twenty percent is planning; and the rest is technical stuff.

-[Philippe Kruchten](#)

Rational Fellow



---

**For more information on the products or services discussed in this article, please click [here](#) and follow the instructions provided. Thank you!**