The Role of the Architect in Software Development

By Dana Bredemeyer, Bredemeyer Consulting
and Ruth Malan, Hewlett-Packard Company

We often find it useful to look at building architecture and see if lessons learned there apply to the software domain. Though there have been building architects for as long as we have built structures, the regulated profession of building architecture is less than 150 years old. Ancient, traditional cultures and languages used the same word for both builder and architect. Construction was an integrated craft. The master mason or carpenter knew how to design structures, estimate costs, assemble labor and materials, and manage the construction process from foundation to roof. With the industrial revolution came new materials, machines, techniques, regulations, etc. And along with all this came a proliferation of highly specialized subcontractors**,** who handled each specialized problem. This redefined the role of the general contractor, whose labor force built less and less of the building. The specialized details of construction became matters for experts while the role of the architect became more clearly focused on providing overall conception of structures, and managing the relationship between the client and the builder/contractor (Lewis, 1998).

It is really easy to see the parallels in software. It wasn't that long ago that an individual or very small group might conceive of and develop an operating system or an entire application. Increasing product complexity, project size, distributed teams, high levels of integration within and even between different product lines, and product lines sharing a common code base, have changed the processes and roles associated with software development. In particular, over the past few years the role of software architect has been created in many organizations to ensure the overall integrity and critical characteristics of systems and development processes.

Although the history of the software architecture discipline is short in comparison with its analogous counterpart in the building domain, we have been able to establish several success factors for the role of the software architect. In this paper, we concentrate on the competencies the architect must have to be successful in the role. First, we consider the responsibilities of the architect in the product generation context. Next, we explore a competency framework for the architect role.

Product Generation Context

Figure 1 is a high-level diagram of the product generation process for a single product. The central responsibility of the architecture team is to create the architecture of the system, identifying the large-grained components or subsystems and their responsibilities, and designing the interfaces. Correspondingly, management will primarily hold the architects accountable for producing an architecture document or specification.

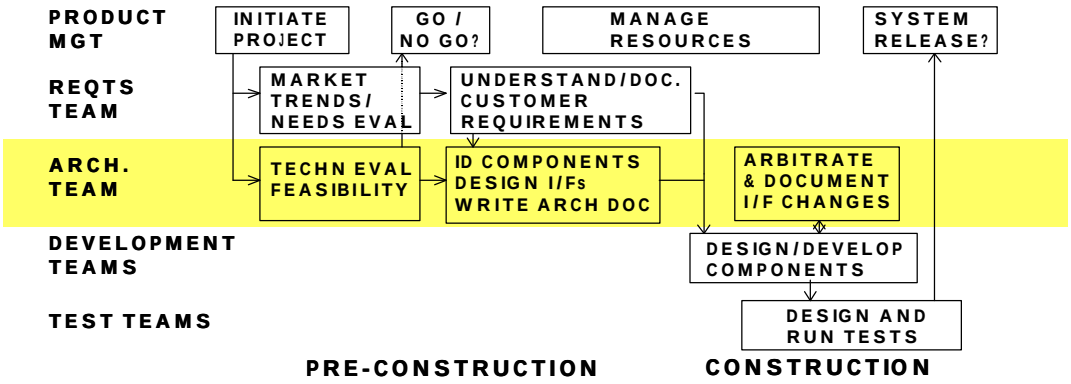


Figure 1. Single-product Development Process

Though this is the case, when we have been called in to conduct architecture assessments late in the development cycle, we usually find that the architecture is no longer apparent and our first activity is to help the team re-document their architecture. There are a number of forces that cause this erosion of the architecture. First, once the architecture is initially defined, quite often the architecture team is dispersed to become the subsystem leads and don't continue to have bandwidth or the authority to overview the architecture as a whole. Second, when an architecture document is "thrown over the wall" to development teams, that provokes resistance behaviors that subvert the purpose of the architecture. And third, as requirements are clarified or new ones emerge that were not taken into account in the architecture, accommodations are made by development teams that are not reflected in the architecture.

To counter these problems, management obviously has to make some changes. First, acknowledge the strategic nature of architecture by committing permanent resources to it. Protect the architects' bandwidth so they have time to review designs and requests for changes to the architecture. Second, they need to support the architects in making their decisions stick. One of HP's leading multi-function device architects calls this the "architecture crying towel"--if you don't like the decision, use the crying towel.

Moreover, the architects need to help the developers understand the architecture and the rationale behind their decisions. They need to act as *consultants* to the developers, and *lead* the development community, helping them to rally behind the architectural vision.

So, in the case of architecting for a single product, the architects need to be consultants and leaders, as well as technologists. Now let us turn to the more complex case of platform development.

A platform is the architecture, frameworks and reusable components that form the foundation of a number of products in a product family. It is a development style that is gaining media attention in the automotive industry, and that has been used in various HP hardware and software businesses over the past several years. Now the architecture has broad business impact across different products and reaches well into the future. For this reason, the architect's role is not just that of creating a good architecture, but of creating a technical strategy that informs the business strategy (Figure 2).

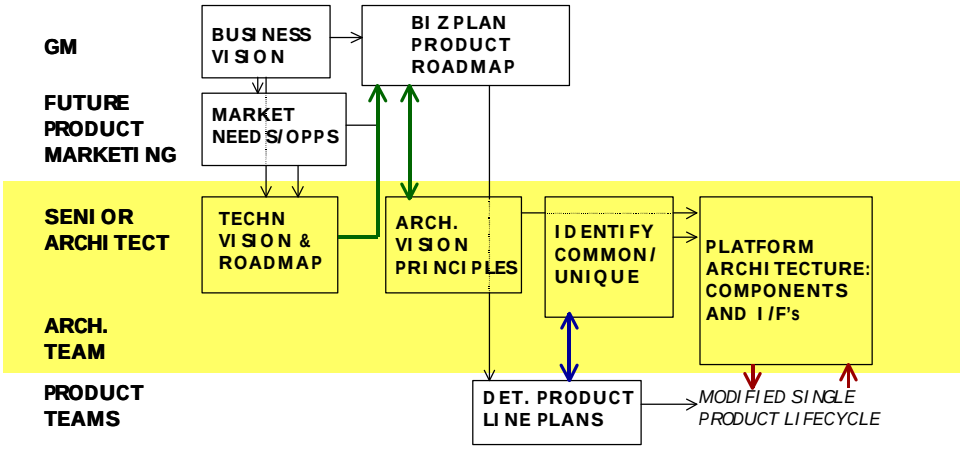


Figure 2. Platform Development Process

In addition to being a *strategist*, the architect needs to work with the management teams across the various product lines, helping them to buy into the architectural vision and maintain their support for the architecture through all the schedule pressures they will face in getting products to market. This is largely a *political* endeavor, as the architect has no direct authority over these chains of management.

And the architect still has to be concerned with the deployment of the architecture to the various development teams. This requires the architecture team to be *consultants* to the engineering community.

Above all, the senior architect has to be seen as a *leader* across these different communities of management, marketing, engineering, etc., as well as by the architecture team itself. If this is not the case, there will quickly be divergence from the architectural vision as each of the product teams go their own direction regardless of the organization's considerable investment in the platform.

An Architect's Domains of Competency

When we created an architecture workshop for HP internal use three years ago, we studied many architecture projects in HP and in the industry, as well as the literature on software architecture and systems architecture (e.g., Rechtin, 1991), and even looked into some of the work on building architecture. Since then, we have worked with many leading architects within HP, and held technical exchanges with other industry leaders such as the architecture team at the SEI who have just published a book on software architecture.

Based on this understanding, and looking at the software architect in the context of the product development process, we have identified several critical areas of activity, or domains of competency, that figure prominently in the architect role. These are technology, business strategy, organizational politics, consulting and leadership. Below, we take a look at the knowledge and experience, activities and personal characteristics it takes to be successful in each of these aspects of the architect role.

Technology

As an architect, you need a thorough knowledge of the product domain, relevant technologies and development processes. But even in the technical area your key activities are different than those of the developers. The problems are less well defined, often with unclear or conflicting objectives, and you play a significant role in clarifying what the objectives are. Your focus is more on the implications of organizational objectives on technical choices. You take an overall system view. You are building models of the problem and solution space, exploring alternative approaches, preparing documents and explaining the architecture to sponsors and stakeholders.

The personal characteristics really essential to success in this domain are a high tolerance for ambiguity and a lot of skill working consistently at an abstract level. We know of at least one case where an otherwise qualified junior architect did not get the senior architect position because of his need for clear and unambiguous objectives.

Often this is the extent of how people see the architect role, and this, along with consulting, is in fact the primary role of a junior architect. But as a senior architect you also need to be an effective strategist.

If the junior architect is primarily a technologist, the senior architect is primarily a strategist, contributing to the business strategy and having primary responsibility for defining the technical strategy.

Table 1. Technology Competency Summary

What you KNOW	What You DO	What You ARE
In-depth understanding of the domain and pertinent technologies Understand what technical issues are key to success Development methods and modeling techniques	Modeling Tradeoff analysis Prototype/experiment/simulate Prepare architectural documents and presentations Technology trend analysis/roadmaps Take a system viewpoint	Creative Investigative Practical/pragmatic Insightful Tolerant of ambiguity, willing to backtrack, seek multiple solutions Good at working at an abstract level

Business Strategy

To succeed in this aspect of the architect role, you need a solid understanding of your organization's business strategy and the rationale behind it, as well as your company or division's business practices, planning cycles, and decision making processes. You have a good understanding of the business context of your organization. You understand your competitors, their products, strategies and product generation processes. You are familiar with the key factors in the business environment that affect your organization's success, and you are able to distill all these business factors into architectural requirements and architectural choices. But the overriding characteristic that fuels your success in this domain is that of an entrepreneurial visionary who can translate well between the business and technical domains.

As a skilled technologist you create good architecture. As a skilled strategist, you create the right architecture for your organization. The next three domains of competency are more about getting the architecture realized. The first is about gaining support for the architecture among the management community. Rob Seliger, the principal architect for the Concert Architecture (Seliger, 1997) for medical information systems said, the single thing architects most need to learn is how to sell, sell, sell.

Table 2. Strategy Competency Summary

What you KNOW	What You DO	What You ARE
Your organization's business strategy and rationale Your competition (products, strategies and processes) Your company's business practices	Influence business strategy Translate business strategy into technical vision and strategy Understand customer and market trends Capture customer, organizational and business requirements on the architecture	Visionary Entrepreneurial

Organizational Politics

As we discussed in the product generation process section, software architectures almost always have many and diverse stakeholders, and are ultimately meant to be used by many developers. Often they are used across divisions and by developers in other companies. To gain and maintain the sponsorship of your management and the enthusiastic support of other key influencers, you will need to do a good deal of influencing yourself.

You really need to understand both the business and personal objectives of key players, and get them personally committed to the success of the architecture. This means listening, networking, articulating and selling a vision, and doing all this continuously over the life of the project.

The people doing this well are extremely articulate and confident. They are resilient and driven, and they are sensitive to where the real power is and how it flows. They look for and see the organization behind the organization, and they use this insight to build and maintain support for their projects.

This domain of competency generates the organizational support to get the architecture created. The next one supports getting it deployed into use.

Table 3. Organizational Politics Competency Summary

What you KNOW	What You DO	What You ARE
Who the key players are in the organization What they want, both business and personal	Communicate, communicate, communicate! Listen, network, influence Sell the vision, keep the vision alive Take and retake the pulse of all critical influencers of the architecture project	Able to see from and sell to multiple viewpoints Confident and articulate Ambitious and driven Patient and not Resilient Sensitive to where the power is and how it flows in your organization

Consulting

The actual users of architecture are development teams creating products or components, and their goal is not to make your architecture successful, but rather to satisfy their specific functionality, schedule and quality requirements. While using the architecture may be the best overall approach for the organization, this is often not apparent to product teams. Consequently, your task as an architect includes recognizing first that developers are a primary customer, and that the architecture must provide value to them in generating good products. Second, you need to enable product teams to quickly understand and effectively use the architecture. You are functioning here more as a mentor and teacher, preparing and making presentations, consulting to individuals and teams, and also mentoring junior architects.

What really contributes to your success here is to be truly committed to others' success and to have a good understanding of change management and how groups adopt new processes.

Table 4. Consulting Competency Summary

What you KNOW	What You DO	What You ARE
Elicitation techniques Consulting frameworks	Build "trusted advisor" relationships Understand what the developers want and need from the architecture Help developers see the value of the architecture and understand how to use it successfully Mentor junior architects	Committed to others' success Empathetic, approachable An effective change agent, process savvy A good mentor, teacher

So now we have a good architecture. It is the right architecture for the organization. It has got sufficient organizational support to actually get created. And it has been effectively deployed to the developer community. It's a wrap! Well, not quite!

Leadership

The domain of competency which organizes all the others and gives them dynamic force, is leadership. An architecture team without leadership goes nowhere. It thrashes and diverges. We've seen this too many times. A leader is required to infuse the team with a common vision, and to motivate the core team and associated teams to do their best work.

This requires dedication and passion, and a strong belief that you can lead the effort. You must see yourself, and others must see you, as a credible leader.

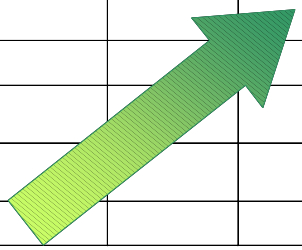
Table 5. Leadership Competency Summary

What you KNOW	What You DO	What You ARE
Yourself	Set team context (vision) Make decision (stick) Build teams Motivate	You and others see you as a leader Charismatic and credible You believe it can and should be done, and that you can lead the effort You are committed, dedicated, passionate You see the entire effort in a broader business and personal context

The diagram below shows that while technology and business strategy skills form a foundation for you as an architect, the real challenges (and ones that are not always acknowledged) as those in organizational

politics, consulting and leadership. Also, as you become more senior in this role, it is less about what you know and more and more about who you are--your personal characteristics.

	What you KNOW	What you DO	What you ARE
Leadership			
Consulting			
Organizational Politics			
Business Strategy			
Technology			



Conclusion

As we have seen, the architect role is very challenging. A lot of what this role is about is not technical, so if this is what you enjoy doing--great! If not, you may not want the role of senior architect.

Before choosing the role, you should also be aware that there are other risks that you should consider. You will have more responsibility without corresponding authority and control, you will encounter a lot of resistance and disappointments--we have seen many an architecture project canceled along the way. And from every angle you will encounter others that believe they have a better idea.

However, if the challenges inherent in architecting are the kind that appeal to you, then the role has great rewards. These include a focus on interesting and complex problems, the opportunity to advance very high in the organization with a continued focus on technical rather than personnel and fiscal issues, and the opportunity to make an enormous difference to the company.

References

Lewis, R., *Architect? A Candid Guide to the Profession*. MIT Press, 1998. (*Note: This book is about the building architect profession.)*

Rechtin, E. *Systems Architecting: Creating and Building Complex Systems*. Prentice-Hall, 1991.

Seliger, R. “An Approach to Architecting Enterprise Solutions”. *HP Journal*, Feb 1997