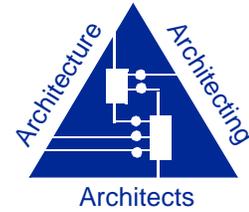


ARCHITECTURE RESOURCES

For Enterprise Advantage

<http://www.bredemeyer.com>



BREDEMEYER CONSULTING, Tel: (812) 335-1653

Meta-Architecture Action Guide

Purpose: Make strategic architectural choices that will guide the architecting effort.

Key Activities:

- Define architecture objectives.
- Review other architectures, styles and patterns and gather lessons from past experience.
- State architecture principles.
- Select/adapt applicable architectural style(s) or patterns.
- Decide on concepts and mechanisms to ensure architectural integrity and consistency.

*by Ruth Malan and Dana Bredemeyer
Bredemeyer Consulting
ruth_malan@bredemeyer.com
dana@bredemeyer.com*

From "Chapter 4: Meta-Architecture". The book is titled *Software Architecture Action Guide*, by Ruth Malan and Dana Bredemeyer.

Introduction

In this Action Guide, we distill the essentials of the Meta-architecture phase of the Visual Architecting Process. The meta-architecture collects together decisions relating to your architecture strategy. It sets direction for your architecture effort, with high-level decisions that will shape the architecture and guide the architects. These include architecture principles, statements of philosophy, metaphors and organizing concepts that will guide system decomposition and design of architectural mechanisms.

Ideally, these decisions are made early on, but pragmatically, some will arise later and simply be documented together with the other strategy-level architecture decisions that have their place in the meta-architecture document(s).

Purpose of Meta-Architecture

First and foremost, meta-architecture establishes the architecture strategy. This includes the architecture objectives, which should be clearly tied to business strategy or business imperatives.

Meta-architecture formulates the high-level choices that are made to achieve the architecture objectives and hence the business strategy. Thus, meta-architecture forms the critical bridge between business strategy and technical strategy, and provides the means to communicate this strategic connection to senior management.

The meta-architecture is a set of high-level decisions that will strongly influence the structure of the system, but is not itself the structure of the system. The meta-architecture, through style, patterns of composition or interaction, principles and guidelines, rules certain structural choices out, and guides selection decisions and tradeoffs among others. It is intended to shape the architecture, guiding the architecture team, but also guiding technical decisions throughout the life of the architecture.

The lead architect or architecture core team is encouraged to establish guiding principles and central organizing concepts or mechanisms, and when this is done in advance of specifying the system structure we consider this part of meta-architecture. Batman notes that a key to successful architecting is creating a small number of system concepts that are effective in reducing complexity and abstracting system characteristics so that designers can retain intellectual control of the system design (Batman, SEI 1999). For example, by choosing communication or co-ordination mechanisms that are repeatedly applied across the architecture, a consistent approach is ensured and this simplifies the architecture.

Inputs to Meta-Architecture

Business Strategy. Architects translate business strategy into technical strategy and lead the implementation of the technical strategy. Meta-architecture is where the technical strategy is set. Clearly, then, business strategy is a vital input to meta-architecture. If you are working at the level of an application or product, you need to pay most attention to the service or product strategy you are supporting. When working on a solution or product family architecture, you are paying attention to the strategy at that level.

Enterprise Architecture. In both business and government, many organizations have established an Enterprise Architecture group, generally within IT. If your organization has an Enterprise Architecture (EA), that needs to be taken into account in your meta-architecture work. You need to be familiar with the EA so that your architecture decisions align fully with the EA (and follow the established exception process if your system truly warrants exceptions, presumably because it lies outside the mainstream systems covered by the EA). Given the Minimalist Architecture Principle (Malan and Bredemeyer, 2002), you would only add to these EA decisions if doing so is necessary to achieve the vision and high-priority goals of the system you are architecting.

Domain, Portfolio or Product Family Architecture. In product development organizations, and in more and more IT groups, broader-scoped architectures are being established to create greater consistency, integration and leverage across products or services within a business domain. If your organization has established an architecture initiative at a higher level of scope that impacts your system, that needs to be taken into account in your meta-architecture work.

Artifacts and learnings from the Initiate and Gain Commitment Phase. The architecture vision is formulated in Init/Commit to help “sell” the architecture, and to act as a beacon guiding decisions during system structuring. This vision may valuably be viewed as a key part of your meta-architecture.

The learnings and artifacts from Init/Commit are also important input to meta-architecture and the creation of the technical strategy. These include stakeholder interviews and work done to understand the strategic direction and context.

Meta-Architecture Activities

Meta-Architecture Requirements: Understanding Context and Goals, and Setting Scope

In accordance with our *just enough* principle, we need to gather sufficient input to make quick but informed progress toward a first-cut meta-architecture. During subsequent cycles we will refine our understanding of the requirements, and mature the architecture. What, then, do we need to know to set architectural direction? Our primary outcome from this stage of requirements gathering needs to be an assessment of the scope of the architecture. In order to do a good enough job of setting scope at this point, we need an understanding of the system context and goals that the architecture will support.

Understand Context. In the Init/Commit phase, much of what we did was geared to understanding the organizational context (considering where we have come from, what our current context is, and what our desired state is), building up to a shared vision. The objective in Meta-architecture Requirements is to get more understanding of the context of use. The preliminary decisions about the system, and in particular determining system scope, rely on this understanding of context.

So, we are interested in the usage context, such as the workflow or business process that the system will support, and the system context, meaning other related systems. For the former, we find process diagrams (like *rich pictures*, Rummler-Brache process diagrams or *UML activity diagrams*) most useful. Often, we also create a *domain concept model* (stereotyped high-level class diagram) to model the real-world problem domain. These models are useful in setting system scope, and we use them downstream.

Understand Goals. In the Init/Commit phase, we identified the architecture stakeholders. These included users of the system as well as business stakeholders. We need to understand the important goals that various stakeholders have for the system (in use) and the architecture (supporting systems under development and evolution). We use *stakeholder profiles* to capture the business goals and system/architecture goals of the stakeholders (influential individuals, and groups of stakeholders). The input to the stakeholder profiles is gathered formally in interviews with the stakeholders, and informally by remaining attentive to that stakeholder during meetings and hallway conversations, and taking notes on their goals as and when they become apparent.

Set Scope. The stakeholder profiles are analyzed to extract goals relating to system functionality, and system properties. System functionality can be shown as use cases on a *use case diagram* for the system. Users, as well as related systems, outside the scope of the current system, are placed as actors outside the system boundary. While the use case diagram shows system functionality that is determined to be in scope, a narrative or other mechanism is needed to record functionality and properties that are ruled out of scope and the rationale for doing so.

The models used to explore system context are also useful for exploring scope. In rich pictures, the system boundary is drawn around the pieces of the picture that are determined to be in scope. In UML activity diagrams, the system is explicitly added as a “swimlane”, and all activities carried out by the system are placed in the system swimlane.

Meta-Architecture Specification: Setting Architecture Strategy

Gather lessons learned. It is a good practice to allocate time to research documented architectural styles, patterns, dominant designs and reference architectures, other architectures that you can get access within your own organization, or that partners or suppliers are willing to share, as well as any that your competitors happen to have documented in the literature, or presented at conferences or user group meetings, etc. Talk to your peer architects and review their architectures with an eye to extracting organizing structures or mechanisms and key concepts that work for your system. Also, reflect on your own experience, so that you can make explicit the lessons that you have personally learned.

Define the *architecture strategy*: Define the architecture *objectives*, and *principles* or *strategies* to guide achievement of these objectives. The business strategy establishes what capabilities need to be built or improved to achieve the business objectives. The architecture objectives establish how technology will be used to deliver these business capabilities, setting direction for the architects and development community. Thus, the architecture objectives are established by the senior architect(s) taking the business strategy, business imperatives and business stakeholder goals (stated on the stakeholder profiles) into account.

The architecture objectives are defined in the *architecture scorecard*, which links the objectives to strategic themes, identifies an owner who is responsible for ensuring the objective is met, and identifies intermediate measures to ascertain progress against the objective, as well as measures that identify when the objective is achieved. We use *strategy maps* to visually link architecture objectives to business objectives.

Select an *architectural pattern or style*: Quite analogous to style in building architecture, an *architectural style* defines a family of systems in terms of a pattern of structural organization. More specifically, an *architectural pattern* or style defines a vocabulary of components and connector types, and a set of constraints on how they can be combined (Shaw and Garlan, 1996). A growing number of architectural patterns applicable to broad classes of systems are published in the literature.

Your enterprise architecture team, or product family architecture team, may already have identified architectural patterns, or created a Reference Architecture, to be used by your organization. If so, this would form your starting point.

Define unifying and simplifying *theme, metaphor, or system concept*: Metaphors, analogies and symbols convey rich meaning. Metaphors and analogies enhance system integrity and understandability. Examples include the “rugby player in a business suit” metaphor for a Honda driver used in designing the new (in the late 1980’s) Honda Accord (Clark and Fujimoto, 1990), and “blackboard”, “pipe-and-filter”, and “broker.” These latter are names of architectural patterns. The use of metaphoric names conveys richly yet succinctly the intent and structure of the pattern.

Create Documentation. Architecture is, by nature, created by the few for the consumption of the many. You cannot bring “the many” into the process directly, so documentation is an important vehicle for communicating the architecture and educating the development community during design ramp-up, and clarifying and resolving issues later on.

It is important to record the reasoning behind architectural choices at the time that these choices are being weighed. This reasoning includes providing traceability to the driving forces behind the decisions (including business strategy, product requirements, etc.), the tradeoffs that were explicitly dealt with, and the experience, modeling and analysis that was brought to bear as alternative approaches were weighed. Later you can tailor your documents for different audiences, but if you neglect to provide a record as you

go, you will have more work to do trying to resurrect the assumptions and assertions, rationale and reasoning that explain the architectural decisions. In short, **don't** leave documentation until the end!

Meta-Architecture Validation: Checking the Direction

Early validation exercises are informal, and focus on validating assumptions and interpretations of stakeholder input, and getting feedback from a close community of “friends of the architecture” on early architecture strategy decisions. As the strategy starts to “gel,” decisions are validated with the architecture sponsor and other business managers impacted by the architecture, as well as key influencers in the development community. A formal review of the meta-architecture may be scheduled to form a check-point allowing the meta-architecture to be formally signed-off. The main objective of such a review is to ensure that the architecture strategy meets the business strategy and high-priority stakeholder goals.

A side-benefit is that participants in the validation process are brought into the architecting process early, and so tend to feel more ownership for ensuring a positive outcome. For this reason, it is useful to pay attention to who participates and when.

VAP Activity Sequence

We have listed a set of activities, but do not intend to convey a *strict* sequence. There is no easily identifiable “horse” to put in front of the “cart” here! For example, your survey of “history” needs to be focused by your understanding of the architecture objectives. But as you reference your own experience and that of other leading architects in your organization and elsewhere, it is quite likely that objectives will emerge to steer the architecture team away from the pitfalls history makes evident.

Further, if you discover the need for an architectural principle when you are drafting the conceptual or logical architecture, it is not too late to add it to the meta-architecture. Later on, you will need to put the whole architecture under change control, but early on, the process needs to be fluid and open to discovery.

The meta-architecture documents provide the structure for keeping track of architecture strategy and high-level decisions that impact the architecture. Thus, if you do discover that you are in fact applying an architectural principle that has not been articulated, and you retrospectively document it, the natural place for it is in the meta-architecture.

Just bear in mind that, when adapting the architecture, changes and additions must be recorded. Thus the various architecture views can be held in suspension for a while, and adapted and matured. At some point, though, each architecture view must be declared “done” so that it can be formally validated, progress can be metered and it can be broadcast to the community it impacts.

Outcomes and Deliverables

For an architecture to be good, right and *successful*, we cannot focus on deliverables alone. In particular, we need to ensure that intangible outcomes like attitude shifts and understanding do not get brushed aside in the effort to produce “deliverables.” We have encountered architects who believe that their manager is judging their effectiveness based on the number and size of documents produced, since this is the visible workproduct of architecture. Indeed, in the absence of all other feedback, this is what your manager will have to fall back on. But the more positive “buzz” you generate in the organization, the more your manager will rely on intangible signals of intangible progress. And the longer you have been in this architecting game, the more you recognize that it is the intangibles that lead to success or failure.

Yes, we need documents that provide a formal “organizational memory,” that serve as a reference, and a medium to judge whether the architecture is good and right. They help determine whether we are ready to move to the next stage. They also serve as medium for broadcast to achieve broader understanding and buy-in than we might have the bandwidth to achieve through one-on-one contact. Some time in the future, the architecture specification may be enough, but in the foreseeable future, success for architecture is most

likely going to mean you have to overcome resistance, educate, negotiate, coax and cajole, working formal and informal avenues of influence to achieve acceptance and readiness to apply the architecture.

Deliverables

Consider the concerns of each stakeholder group, as well as concerns of individual stakeholders in the case of those who have high impact on the success or failure of the architecture. Also, consider the communication styles of these stakeholders. Tailor views that address specific concerns, and tailor communication formats to match communication styles and needs. This process (consistent with the IEEE 1471 Standard for Architecture Description), will help you target your architecture deliverables most effectively. In the sections below, we cover deliverables that are generally useful.

Documents

Architecture Strategy Document. The audience is senior management, and the purpose is to show how the architecture strategy supports the business strategy, and indicate how the technical strategy will be implemented. Key artifacts to include in this document are the strategy map visually linking architecture objectives to business objectives, the architecture scorecard, and architecture principles.

Meta-Architecture Specification Document. The audience is the architecture team, and the purpose is to provide a complete record of decisions (and their rationale) made to guide the architects during the remainder of the architecting process. This record includes technical implications of the architecture strategy and high-level architecture decisions including architectural style, metaphors, early decisions around architectural mechanisms, together with the rationale for these decisions. It is helpful to include alternatives that were considered, and why they were ruled out.

White Papers. The audience is the technical community, and the purpose is to explain and achieve buy-in. White papers address a focused topic, such as discussing, motivating and illustrating an architecture principle or a key architectural mechanism.

Presentations

Here is a piece of advice from Rob Seliger that we have found useful: In communicating the architecture to upper management, start with the business strategy and work towards the technical implications. But in communicating with a technical audience, start with the technical content and work towards the business motivations driving the selection among technical alternatives. In other words, start in your audience's mindspace, and provide the bridge to the implementation strategy, in the case of the business audience, or the business drivers, in the case of the technical audience.

Here is another piece of advice from a chief software architect, Kristen Sanderson: If your organization is large and the architecture strategy must be bought into at several higher levels, it is important to create a concise executive overview—ideally, just one slide—to communicate the architecture strategy, plan and concepts. This will help you communicate to the stakeholders above you and, if you can clearly show the value and direction, they will use this slide as part of their own presentations.

Web Site

In this age of the Internet, it is imperative that you have an architecture web site! But, you need a site that is well-designed to present the architecture according to the needs of different stakeholders. Simply adding a web interface to a repository of architecture documents is going to confuse and overload your various audiences. Rather than supporting your effort to communicate, it will subvert this goal. Apply your architecting skills and techniques to the structure of the web site: identify stakeholders and their concerns, and structure views into the architecture documentation set accordingly. Do this early, so that your stakeholders will become eager followers of the unfolding architecture work, and your job during architecture deployment will be easier.

Outcomes

Given these deliverables and all the influencing you do informally in conversations, meetings, and the like, you will have laid the groundwork for achieving the following outcomes:

Sponsorship. The architecture has a strong sponsor and champion in senior management. The commitment of the sponsor is evidenced when the sponsor:

- fully resources the architecture project (e.g., the architecture team is not made up of “part-timers” who have competing responsibilities on other projects);
- actively removes barriers to architecture progress;
- gives the architecture group the authority to enforce architecture decisions; and
- visibly champions the architecture among managers and the developer community.

Architecture Team Alignment. The team of architects is aligned and has strong forward momentum. This is evidenced by:

- strong and accepted leadership within the architecture team;
- creative, open and collaborative exploration of architectural alternatives; and
- effective decision-making.

Community Buy-In. There is broad buy-in among impacted managers to the architecture strategy. Key influencers in the technical communities buy-in to the architectural style, principles, central concepts, metaphors, and architectural mechanisms. This buy-in is evidenced by:

- others (not on the architecture team) are promulgating the architecture strategy, architectural style and architectural principles, and presenting the rationale you would give when someone challenges them.

Conclusion

Meta-architecture lays an important foundation, laying out the high-level path toward the architectural vision, before diving into system decomposition and design of architectural mechanisms.

This phase is especially important for large projects, with an architecture team rather than a single architect. It is where the chief architect, with the help of a small core team of architects, lays out the architecture strategy that aligns and guides the architecture team during the core phases of architecture specification.

We strongly encourage architects to allocate time to work on architecture strategy, and just as strongly, we encourage architects to quickly start to explore Conceptual Architecture, and even Logical and Execution Architecture. This is because direction needs to be set, so that we can head out on the architectural journey with focused intent. But it is far better to learn soon that the direction needs to be adapted, than to plan intensely and meet with failure because we spent so much time planning the trip that we have too little time left for the journey!

References

- Batman, Joe, Characteristics of an Organization with Mature Architecture Practices, *Essays on Software Architecture*, Software Engineering Institute, 1999.
- Bredemeyer, Dana and Ruth Malan, “The Role of the Software Architect,” published on the *Resources for Software Architects* web site at <http://www.bredemeyer.com/papers.htm>.
- Clark and Fujimoto, “Power of Product Integrity”, *Harvard Business Review*, 1990.
- Kaplan, Robert, and David Norton, *The Strategy Focused Organization*, Harvard Business School Press, 2001.
- Malan, Ruth, and Dana Bredemeyer, “Less is More with Minimalist Architecture”, *IEEE's IT Professional*,

September/October 2002.

Malan, Ruth, and Dana Bredemeyer, “Architecture Strategy”, published on the *Resources for Software Architects* web site at <http://www.bredemeyer.com/ArchitectingProcess/ArchitectureStrategy.htm>

Shaw, Mary and David Garlan, *Software Architecture: Perspectives on an Emerging Discipline*, Prentice-Hall, 1996.

Acknowledgments

We would like to thank all the architects that we have worked with over the past decade for freely sharing their insights and hard-won lessons with us. We are most privileged to be able to pass on their learnings through our papers and workshops.

We would like to especially thank Kristen Sanderson, who has provided truly useful feedback on this chapter. We have incorporated much of her input already, and this chapter is stronger now. One manifestation of our response to Kristen’s advice, is the Table of Contents that we have added on the next page.

Restrictions on Use

This paper and all other material that is published on the *Resources for Software Architects* web site (<http://www.bredemeyer.com>), may be downloaded and printed for *personal* use. If you wish to quote or paraphrase fragments of our work in another publication or web site, please properly acknowledge us as the source, with appropriate reference to the article or web page used. If you wish to republish any of our work, in any medium, you must get written permission from the lead author or the site editor. Also, any commercial use must be authorized in writing by Bredemeyer Consulting.

A Note About the Forthcoming Book

We are writing a book for software architects that is short and oriented to guiding action. It has two parts, with the first part providing context and a guide to the process. The second part is the full set of Action Guides, one for each discrete technique, model or template that is used in the Visual Architecting Process.

We have other books in mind, but a distilled guide to our Visual Architecting Process is overdue! Brevity is a driving goal for this book. It has caused us to make choices, like pulling details on techniques and models out of the process overview and placing them in pithy “action guides”, one per technique, model or template. This means that if you are not familiar with a technique or model that we make reference to, you will have to wait with baited breathe for the related Action Guide to appear.

Joking aside, we look forward to your feedback, but please take into account that you don’t have all the pieces yet. We have chosen to put chapter drafts “out there” as quickly as possible so that you can have access to more information on the Visual Architecting approach, and we can have the benefit any input you are kind enough to take the time to give us.

Please join our mailing list to receive notice of new chapters and Action Guides as they are added to the site. To do so, complete the form on our web site at <http://www.bredemeyer.com/Forms/subscrib.htm>, or click the envelope icon in the sidebar of most pages on our web site.

Table of Contents

Part I: Software Architecture and the Visual Architecting Process

Chapter 1. Software Architecture: Central Concerns, Key Decisions

Chapter 2. The Visual Architecting Process: Good, Right and Successful

Chapter 3. Initiate and Gain Commitment: Getting Started

- Chapter 4. Meta-Architecture: Getting Strategic
- Chapter 5. Conceptual Architecture: Getting the Big Chunks Right
- Chapter 6. Logical Architecture: Getting Precise, Making Actionable
- Chapter 7. Execution Architecture: Getting Physical
- Chapter 8. Architecture Guideline and Policies: Getting Specific
- Chapter 9. Architecture Deployment: Getting Real

Part II: Software Architecture Action Guides

Here are some examples of what we call Action Guides:

- Software Architecture Principles Template (http://www.bredemeyer.com/pdf_files/Principles_Template.PDF, 24kb)
- Stakeholder Profile Action Guide (http://www.bredemeyer.com/pdf_files/Stakeholder_Profile.PDF, 222kb)
- Use Case Template (http://www.bredemeyer.com/pdf_files/UseCase_Template.PDF, 25kb)
- Interface Specification Template (http://www.bredemeyer.com/pdf_files/Interface_Template.PDF, 24kb)

Resources

Software Architecture Workshop. This class focuses on the Visual Architecting Process (VAP). It is organized around the process. As the workshop progresses, small teams of participants take their project from vision to architecture. This format, punctuating lectures with exercises that build on one another, gives participants the opportunity to learn and practice techniques used in each of the process steps.

Open Enrollment Classes:

Las Vegas, Nevada on **March 14-17, 2006**

London, UK on **June 6-9, 2006**

See http://www.bredemeyer.com/architecture_workshop_overview.htm

Enterprise Architecture Workshop. This class focuses on the Visual Architecting Process for the *Enterprise* (VAP-Enterprise). Following a couple of context-setting modules, the core sections of the course are organized around the process. It follows a workshop format, with lecture modules followed by team exercises to practice techniques and solidify concepts and models. The Visual Architecting Process for the Enterprise starts with Business Strategy, identifies and refines the Business Capabilities Architecture, and uses this to drive the Information (data), Application Solution, and Technology (Infrastructure) Architectures at the enterprise level of scope.

Open Enrollment Classes:

Las Vegas, Nevada on **March 21-24, 2006**

London, UK on **June 6-9, 2006**

See http://www.bredemeyer.com/Enterprise_Architecture/Enterprise_Architecture_Workshop.htm

Role of the Architect Workshop. Excellent class for architects—according to those who have taken it, that is. This class helps you identify what skills you need to strengthen, and starts you along the road to building them, while providing options for what to do next to further develop needed skills.

Open Enrollment Class: Indianapolis, IN on **March 28-30, 2006**

See http://www.bredemeyer.com/role_of_architect_workshop_overview.htm

About Bredemeyer Consulting

Bredemeyer Consulting provides a range of consulting and training services focused on Enterprise, System and Software Architecture. We provide training and mentoring for architects, and typically work with architecture teams, helping to accelerate their creation or renovation of an architecture. We also work with strategic management, providing consulting focused on developing architectural strategy and organizational competency in architecture.

We manage the *Resources for Software Architects* web site (see <http://www.bredemeyer.com>). This highly acclaimed site organizes a variety of resources that will help you in your role as architect or architecture program manager. A number of Bredemeyer Consulting's *Action Guides*, presentations and white papers are on the Papers and Downloads page (<http://www.bredemeyer.com/papers.htm>). You may also be interested in our *Software Architecture* and *Enterprise Architecture* Workshops, as well as our *Architectural Leadership* class. For more information, please see <http://www.bredemeyer.com/training.htm>.

BREDEMEYER CONSULTING

Bloomington, IN 47401

Tel: 1-812-335-1653

<http://www.bredemeyer.com>