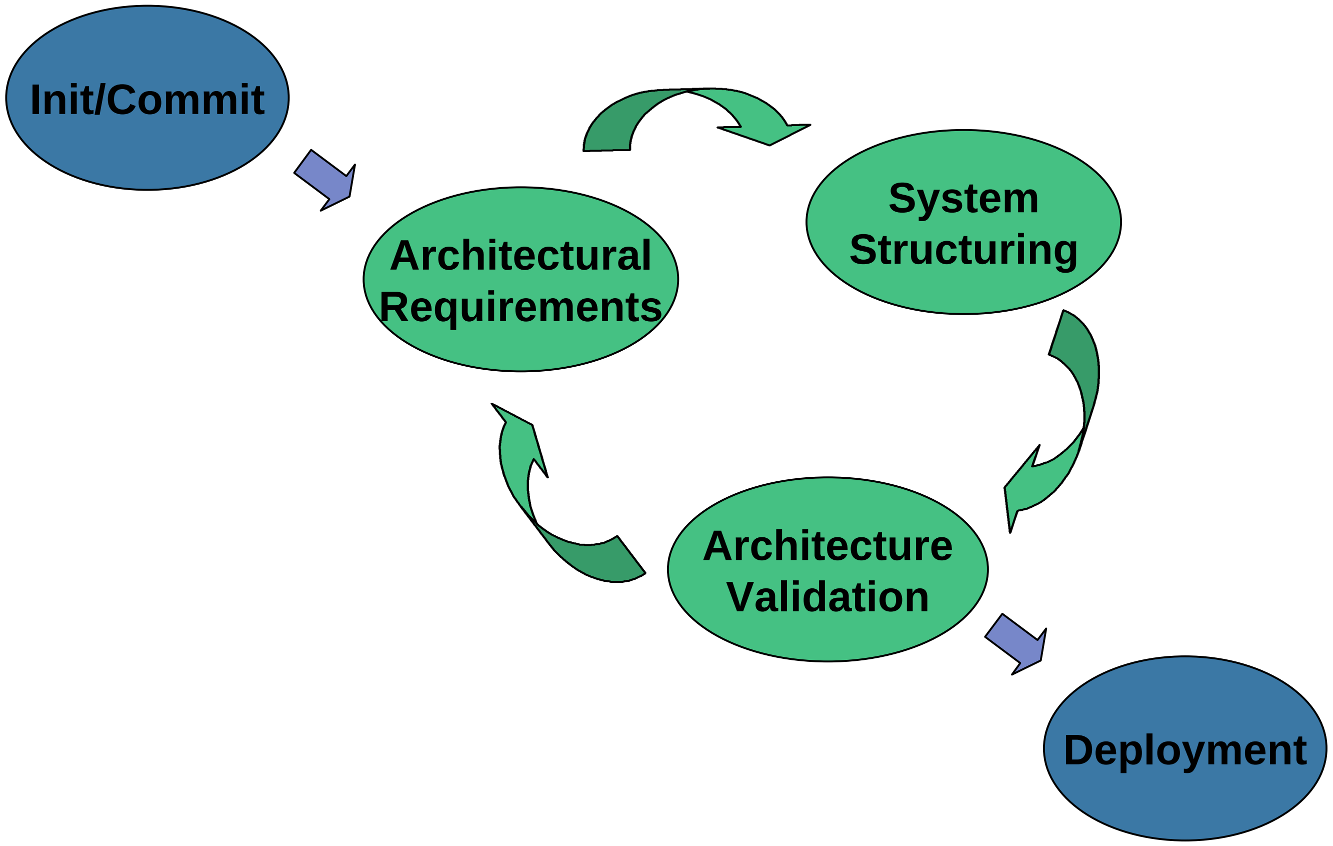
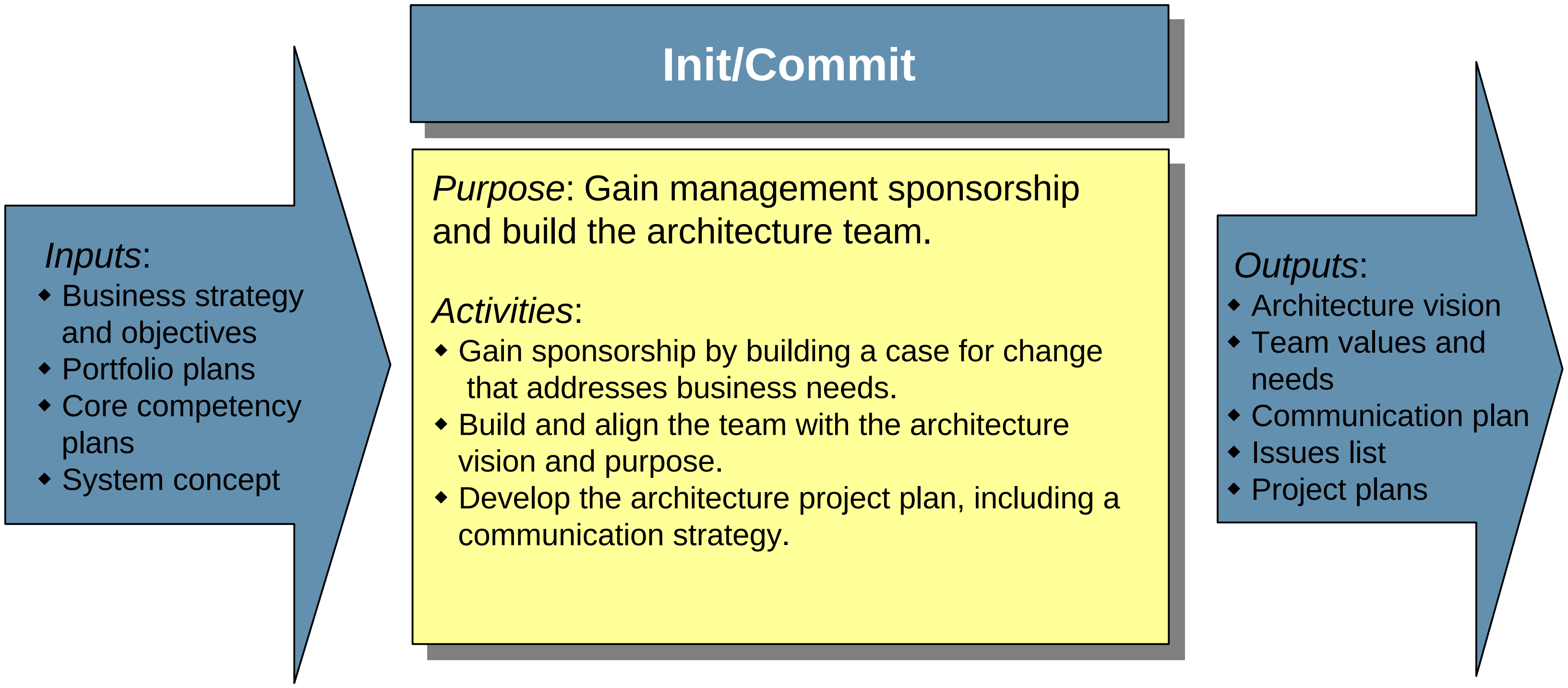
Architecting Process



Architecting Process: Init/Commit Phase



Purpose: Gain management sponsorship and build the architecture team.

Activities:

- Gain sponsorship by building a case for change that addresses business needs.
- Build and align the team with the architecture vision and purpose.
- Develop the architecture project plan, including a communication strategy.

Inputs:

- Business strategy and objectives
- Portfolio plans
- Core competency plans
- System concept

Outputs:

- Architecture vision
- Team values and needs
- Communication plan
- Issues list
- Project plans

Architecting Process: Architectural Requirements Phase

Inputs:
- Business strategy and objectives
- Portfolio plans
- Core competency plans
- System concept
- Architecture vision

Architectural Requirements

Purpose: Establish and document the architectural requirements.

Activities:
- Understand the system context, including key organizational, business, competitive and technical drivers affecting the architecture.
- Determine how the architecture will contribute to competitive advantage and business objectives
- Identify stakeholder goals and architecture scope.
- Document functional requirements by translating user goals into a set of use cases.
- Document the non-functional requirements, associating measurable qualities with use cases or creating scenarios.
- Model common/unique usage and infrastructure requirements across systems

Outputs:
- Architecture requirements
- Issues list
- Updated project plans

Architecting Process: System Structuring Phase

Inputs:
- Architectural requirements (use cases and qualities)
- Architectural patterns and styles

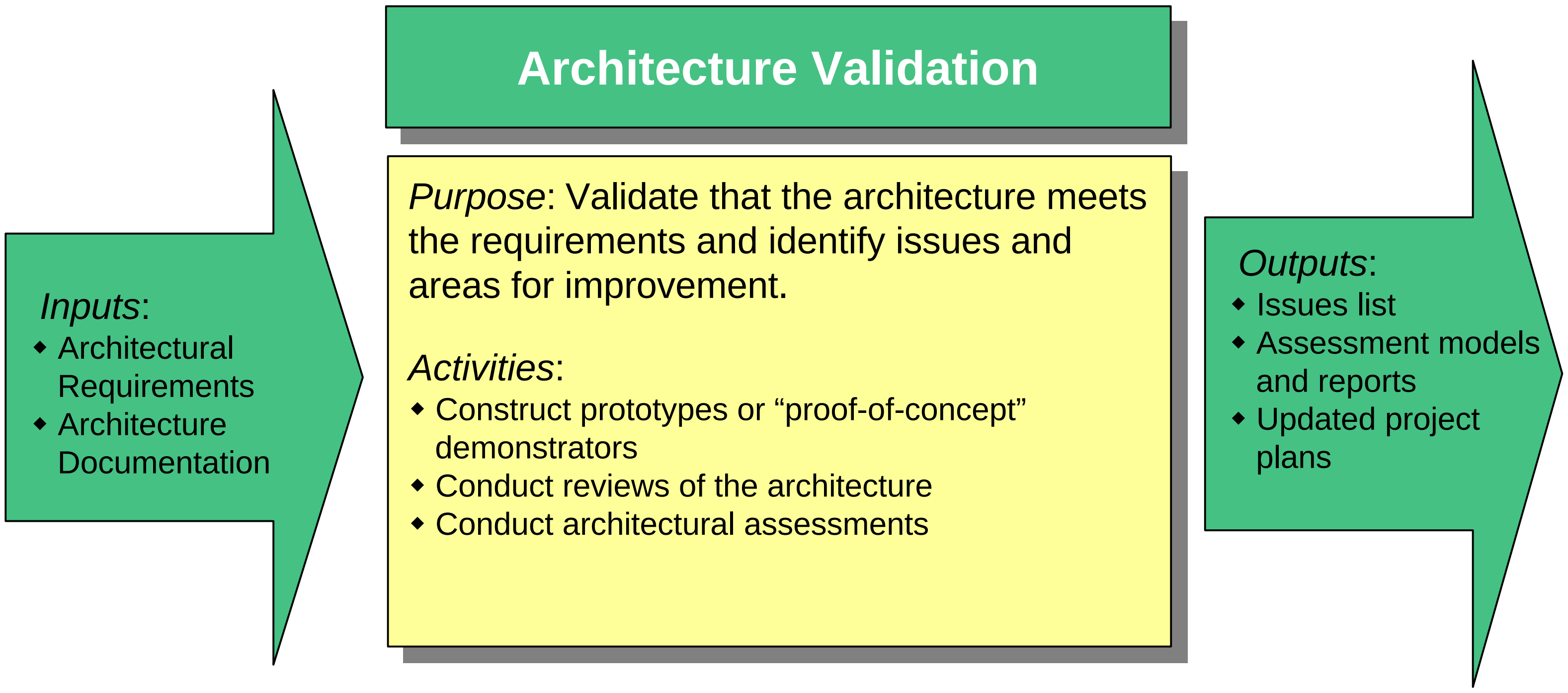
System Structuring

Purpose: Design the high-level system structure, and specify (broad-scoped) architectural guidelines for designers.

Activities:
- Define the meta-architecture, including principles, style and key mechanisms
- Define the conceptual architecture: partition the system and allocate responsibilities to components
- Define the logical architecture: model collaborations, design interfaces, complete component specifications
- Define the execution architecture: map components to processes and threads; determine location on physical nodes
- Specify architectural guidelines and standards
- Select key technologies

Outputs:
- Architecture documents and models
- Issues list
- Updated project plans

Architecting Process: Architecture Validation Phase



Architecture Validation

Inputs:
- Architectural Requirements
- Architecture Documentation

Purpose: Validate that the architecture meets the requirements and identify issues and areas for improvement.

Activities:
- Construct prototypes or “proof-of-concept” demonstrators
- Conduct reviews of the architecture
- Conduct architectural assessments

Outputs:
- Issues list
- Assessment models and reports
- Updated project plans

Architecting Process: Architecture Deployment Phase

Inputs:
- Architectural Requirements
- Architecture Documentation

Architecture Deployment

Purpose: Ensure that the architecture is followed in designing products or systems.

Activities:
- Communicate the architecture and its rationale
- Educate and consult with developers as they apply the architecture
- Review designs (to ensure that designs are consistent with the architecture); certify designs for compliance
- Identify needs for evolving the architecture

Outputs:
- Designs compliant with architecture
- Requirements for evolving the architecture
- Issues list

Architecting Process: Iterations

	Pass 1 From Business Strategy to Architectural Strategy	**Pass 2** From Strategy to Concept	**Pass 3** From Concept to Specification	**Pass 4** From Specification to Execution	**Pass 5** From Execution to Deployment
Architectural Requirements	Context Goals Scope	Use Cases Qualities	Refine Use Cases	Concurrency	Developer needs
System Structuring	Meta-Architecture	Conceptual Architecture	Logical Architecture	Execution Architecture	Architectural Guidelines
Architecture Validation	Reasoned Arguments	Impact Analysis	Estimates	Prototypes	Design reviews, Implementation

Architecting Process

Background

We have worked with and studied numerous architecting projects, distilling common best practices and identifying pitfalls and critical success factors. Our architecting process is based on these lessons from experience creating architectures for various industries.

Technical + Organizational Process

The core of the architecting process is a technical process focused on creating an architectural solution that is a good fit to the requirements placed on it. However, a technically sound architecture, though necessary for success, is not sufficient to ensure that the architecture is used as intended. The organizational process focuses on building support for and understanding of the architecture.

Architecting Process Steps

The software architecting process involves the following steps:

- **Init/Commit**: Gain management sponsorship and form the architecture team
- **Requirements**: Establish and document the architectural requirements
- **System Structuring**: Define the architecture
- **Validation**: Validate that the architecture meets the requirements
- **Deployment**: Deploy the architecture to the developer community

The technical process steps, namely Architectural Requirements, System Structuring and Validation, are best conducted iteratively.

Though activities aimed at ensuring support for the architecture predominate during Init/Commit and Deployment, they need to continue at some level throughout the life of the architecture.

Each of the steps is described in the action guides that follow.

Init/Commit

Gain management sponsorship

Purpose: Ensure management support through the life of the architecture project, so that management will remove obstacles to success and champion the architecture

Activities:

- Create/communicate the architecture vision (see the Vision Action Guide*) showing how the architecture contributes to long-term business success

Checks:

- Do you have the resources you need?
- Are architecture team members assigned full-time?
- Does management champion the architecture vision?

Build the architecture team

Purpose: Ensure a cohesive and productive team that is able to move quickly toward a sound architectural solution

Activities:

- Use the architecture vision to build team alignment (see the Vision Action Guide*)
- Assess team capabilities and needs (see Team Assessment Action Guide*)
- Establish the team operating model, including team roles and responsibilities, decision model and issue resolution strategy

Checks:

- Is there a strong and accepted leader?
- Is the team collaborative and creative?
- Do decisions get made effectively?

Start to build organizational buy-in

Purpose: Ensure the ultimate adoption and appropriate use of the architecture by building broad support among stakeholders

Activities:

- Understand stakeholder communication styles and needs
- Create a communication plan (see the Communication Plan Action Guide*) to ensure appropriate input, participation and understanding among the various stakeholder groups

Checks:

- How will you get the participation of influential developers?
- How will you avoid too much input (“architecture by committee”)?
- How will you get application/product managers’ buy-in to using the architecture even if it “slows their project down”?

* The complete set of our Action Guides are included in the materials given to participants in our Software Architecture Workshops. See http://www.bredemeyer.com/training.htm for more information on our workshops.

Architectural Requirements

Capture Context, Goals and Scope

Purpose: Ensure that the architecture is aligned with the business strategy and directions, and anticipates market and technology changes

Activities:

- Scan the environment, identifying factors, trends and forces that are likely to impact the architecture (see Context Map Action Guide*)
- Establish which business objectives apply to the architecture to ensure that the architecture is aligned with the business agenda
- Determine where the architecture will provide competitive differentiation, and where not
- Elicit and record stakeholder goals to discover opportunities to support stakeholder goals and focus on providing specific perceived benefit to the stakeholder (see the KJ Analysis and Stakeholder Profile Action Guides*)
- Determine the architecture scope to provide focus and direction and identify dependencies (see Scope Action Guide*)

* The complete set of our Action Guides are included in the materials given to participants in our Software Architecture Workshops. See http://www.bredemeyer.com/training.htm for more information on our workshops.

Capture Functional Requirements

Purpose: Document, communicate and validate the intended behavior of the system

Activities:

- Use *use cases* to capture *who* (actor) does *what* (interaction) with the system, for what *purpose* (goal), without dealing with system internals (see the Use Case Action Guide*)

Checks:

- Did you include use cases for all categories of users (including field support, system administrators, etc.)?

Capture Non-functional Requirements

Purpose: Explicit, documented non-functional requirements are needed to: define architectures so that they achieve the required qualities; form a basis for comparing alternatives and make tradeoffs; enhance communication; and evaluate the architecture

Activities:

- Identify non-functional requirements (qualities and constraints)
- Define each required quality attribute unambiguously
- State a measure or test that will be used to ensure that the quality attribute is met
- Prioritize the non-functional requirements

Checks:

- Is each requirement SMART (specific, measurable, attainable, realizable and traceable)?

System Structuring

Create the Meta-Architecture

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

Activities:

• Review other architectures, styles and patterns and gather lessons from past experience (see Graphic History Action Guide)
• Create architectural principles (see Principles Action Guide)
• Select/adapt applicable architectural style(s) or patterns
• Decide on concepts and mechanisms to ensure architectural integrity and consistency

Create the Conceptual Architecture

Purpose: Create conceptual models to communicate the architecture to management sponsors, project managers for team/individual work assignments and customers/users. Also allows for early validation of key architectural decisions and forms the starting point for the logical architecture.

Activities:

• Create an architecture diagram showing the system decomposition into components and connectors (see Architecture Diagram Action Guide*)
• Create informal component specifications, documenting each component's responsibilities, the components it collaborates with in accomplishing the responsibilities and the rationale for clustering the responsibilities in that component (see CRC-R Action Guide*)

Create the Logical Architecture

Purpose: Create detailed architectural specifications to document the architecture decisions and to communicate the architecture to designers, developers and contractors in a way that is directly actionable, clear and unambiguous

Activities:

• Use component collaboration diagrams (CCD) to explore and document system behavior--helpful in elaborating the component interfaces. (see CCD Action Guide*)
• Create detailed component specifications, documenting the interfaces (list of operations, descriptions of the operations, constraints represented as pre-post conditions on the operations or state diagrams, etc.) and component use model (concurrency model, constraints on component composition, lifecycle model, how the component is instantiated, how it is named, a test or performance suite, etc.) (see Component Specification and State Diagram Action Guide*)

Create the Execution Architecture

Purpose: Map the components to processes to explore distribution options and concurrency, to optimize the system's throughput, deal with simultaneous events, deal with scalability, etc.

Activities:

• Annotate the component collaboration diagrams (CCD) showing active components, flows of control, and (a)synchronous messages. (see Process View Action Guide*)

Architecture Validation

Validate the Architecture

Purpose: Assess the architecture to validate that it meets the requirements and identify issues and areas for improvement early

Activities:

• Construct prototypes or "proof-of-concept" demonstrators or build a skeletal architecture to validate communication and control mechanisms and interfaces

• Conduct reviews of the architecture to check that principles are upheld and other meta-architecture guidelines are met, and discuss how the architecture meets the goals and requirements placed on it

• Conduct architectural assessments to assess the architecture against use cases to see that it will support the required functionality, and scenarios to see that the system qualities are met (see the SAAM Action Guide*)

Checks:

i. Goodness of the architecture

• Did you satisfy yourself and others that the architecture as defined satisfies the stakeholder goals and requirements?

• Did you assess the conceptual integrity, correctness, and buildability of the architecture?

• Did you assess the degree to which the architecture will flex to meet future requirements?

ii. Goodness of the architecture documentation

• Did you assess the understandability and utility of the documentation?

* The complete set of our Action Guides are included in the materials given to participants in our Software Architecture Workshops. See http://www.bredemeyer.com/training.htm for more information on our workshops.

Deployment

Build Understanding

Purpose: Help the developer community understand the architecture, its rationale, and how to use it. Help the management community understand its implications for organizational success, work assignments, etc.

Activities:

- “Communicate, communicate, communicate” (Rechtin, 1996). Listen! Give presentations, keep an “open door”, write good documentation, etc.
- Consult: always be available to assist and consult
- Educate: create tutorials and demos

Ensure Compliance

Purpose: Ensure that designs and implementations adhere to the architecture and do not cause architectural drift

Activities:

- Review designs to ensure they are consistent with the architecture

Evolve the Architecture

Purpose: Ensure that the architecture remains current

Activities:

- Actively watch for and respond to need for changes to the architecture. Stay engaged!

Checks:

- Do all stakeholders "see" and describe the same picture of the architecture, what its components are and who owns them?
- Does each person in the delivery chain knows his/her role and impact on delivering (or eroding) the architectural value?
- Is the architecture is explicitly discussed when changes to the products it supports are evaluated or enacted?

* The complete set of our Action Guides are included in the materials given to participants in our Software Architecture Workshops. See http://www.bredemeyer.com/training.htm for more information on our workshops.

Iterations

Suggested Iterations

The software architecting process, and in particular the technical steps in the process, are best conducted iteratively. One way to do this, is to focus on each of the architecture model abstraction layers in turn. Thus, meta-architecture is the focus of the first iteration through requirements, structuring and validation. Next, an iteration focuses on creating the conceptual architecture, and then logical and execution architectures in turn.

In our workshops, we cover the appropriate kinds of requirements to be gathering during each iteration, the modeling tools used during system structuring in that iteration, and the validation techniques used to validate the level of the architecture created during that iteration. At each iteration, the results of previous iterations are reviewed and refined, and completely reworked if called for.

This approach discovers flaws in the architecture early and allows them to be rectified with minimal impact.

Bredemeyer Consulting specializes in training and mentoring software architects. We typically work with architecture teams, providing training and mentoring to accelerate their creation or migration of an architecture. However, we do offer a limited number of Software Architecture Workshops for open enrollment. The currently scheduled open enrollment workshops are:

• Bloomington, Indiana: Nov. 30-Dec. 3, 1999
• Palo Alto, California: February 22-25, 2000

See http://www.bredemeyer.com/training.htm for more information.