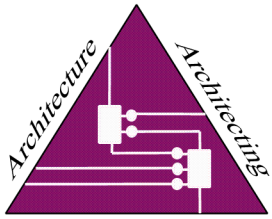
Software Architecture Workshop

from Bredemeyer Consulting



London, UK
April 8-11, 2002

Key Benefits

You will learn how to:

- Express architectural principles
- Capture architectural patterns using UML
- Create structural and behavioral views of the architecture using UML models
- Graphically model components (using UML) to understand/communicate their structure and behavior
- Design and document components and their interfaces
- Capture architectural requirements using Use Cases for run-time requirements and Test Cases for qualities
- Perform architecture analyses and validate the architecture
- Use our Action Guides to guide you through each key architecting activity

Background

Software architecture is the high-level structure of a software system. Its importance derives from the role it plays in system development:

- it is the translation of business strategy into technical strategy, and is the foundation for competitive advantage
- it helps address system complexity, making the system more understandable and more manageable

The *Unified Modeling Language* (UML) is the industry-standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. We focus on those UML models that are useful for capturing and communicating software architecture.

Action Guides are short (fit on an index card) and designed to solve a specific architecting problem (e.g., develop an architectural vision, capture non-functional requirements or document interfaces). Taken together, they provide a comprehensive architecting process that uses UML as the modeling language. Used selectively, they provide a mechanism for teams to tailor their own lightweight architecting method. Used individually, they produce some key result for the architecting team.

Benefits

This class covers the concepts and concerns central to software architecture. With that groundwork laid, the majority of the time is focused on the architecting process. Each architecting activity is motivated with stories. The concepts, techniques and models for each phase of requirements, structuring and validation are explained and demonstrated, then practiced in small teams. Action Guides provide handy references as teams fill out templates or create graphical models using UML.

The workshop addresses such questions as:

- What is a software architecture? (There are multiple definitions out there.)
- How do I use UML to model the system? Which models best capture the structure of the system? the behavior of the system? and the mapping of software components to the physical system?
- What have others learned in creating architectures? What works? What does not?
- What should we think about when designing the architecture?
- How should we document the architecture?
- What should we communicate, to whom?

Content

This 4-day workshop covers the following topics:

Introduction to Software Architecture

- What is software architecture?
- Central concerns of software architecture
- The architecting process

Architectural Requirements

- System context and scope
- Functional (or behavioral) requirements and Use Cases
- Non-functional requirements (i.e., system qualities including run-time qualities such as performance and reliability, and development-time qualities such as

evolvability/extensibility and reusability) and test cases

Architecture Modeling with UML and Action Guides

- *Meta-architecture*: the architectural vision, style, principles, key communication and control mechanisms, and concepts that guide the team of *architects in the creation of the architecture.*
- *Architectural patterns*: the role of patterns; documented patterns such as layers and client/server, brokers and bridges; how to capture patterns
- *Architectural views*: Structural views that help document and communicate the architecture; Behavioral views that are useful in thinking through how the components interact to accomplish their assigned responsibilities and in evaluating the impact of what-if scenarios on the architecture.
- *Conceptual Architecture*: decomposing the system into subsystems and components; assigning component responsibilities
- *Logical Architecture*: creating the architectural blueprint; interface design and specification; component documentation
- *Execution Architecture*: mapping components to processes and threads, and to nodes in the physical system to evaluate distribution options and document decisions
- *Key architectural design principles* including abstraction, separation of concerns, postponing decisions, and simplicity, and related techniques such as interface hiding and encapsulation, as well as system decomposition principles and good interface design.
- Heuristics for system decomposition

Architecture Validation

- Architecture validation techniques (reviews, assessments, prototypes)
- Impact assessment and SAAM

Format. Exercises form a large component of this four-day workshop, which is oriented toward building skills rather than simply exposing students to new concepts. Also, *case studies and stories* are used to convey real-world lessons.

Audience

This workshop is designed for architects and senior system design engineers. Managers of architecture teams will also benefit.

About the Instructors

The workshop leader, Dana Bredemeyer, has over 20 years experience architecting, designing and developing software systems, including 16 years with Hewlett-Packard. He developed Hewlett-Packard's internal Software Architecture Workshop, and continues to be its principle instructor. He has provided architecture consulting and training to architects, architecture teams and their management at the project, organization and business unit levels, and has helped teams develop software, firmware and system architectures for products, product families and information systems. Dana is currently co-authoring a book on software architecture for Prentice-Hall.

Our UK-based senior consultant, David Redmond-Pyle, will be co-teaching the workshop. David worked with IBM's architecture community on software architecture best practice and architectural reuse, and co-authored IBM's "Standard for Architecture Description", *IBM Systems Journal*, Vol 38 No 1, http://www.research.ibm.com/journal/sj/381/youngs.html. He is also a recognized expert in user interface design and usability evaluation, having published an influential book on the subject. David was previously Principal Consultant and Chief Methodologist at software method and tool vendor LBMS. David is technical director of PostModern Solutions Ltd..

Pricing

The cost for the 4-day workshop is $2000 per student.

Cancellation Policy. Refunds will be made in full if cancellations are made more than 4 weeks in advance of the workshop start date. Cancellations made within 4 weeks of the workshop start date are not refundable. However, student substitutions may be made any time prior to the start of the workshop.

Venue

The workshop will be held at Elvetham Hall, near London. For more information, see http://www.bredemeyer.com/Workshops/London.htm

To Register

To register for the workshop, you can do one of the following:

- register on-line on our web site http://www.bredemeyer.com/training.htm, or
- call Bredemeyer Consulting at 1-812-335-1653

About Bredemeyer Consulting

Bredemeyer Consulting specializes in training and mentoring enterprise and software architects. We typically work with architecture teams, helping them to accelerate their creation or migration of an architecture. However, we do offer a limited number of Software Architecture and other workshops for open enrollment.

For more information, please

- see our web site: http://www.bredemeyer.com
- call 1 812 335-1653.