The Role of the Software Architect

Peter Eeles
peter.eeles@uk.ibm.com
Agenda

- Architecture
  - Architect
  - Architecting
    - Requirements
    - Analysis and design
    - Implementation
    - Test
    - Project management
    - Configuration and change management
    - Environment
  - Summary
Architecture, Architect, Architecting
Architecture

- Architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution. [IEEE 1471]

- The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them. [Bass]

- [Architecture is] the organizational structure and associated behavior of a system. An architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Parts that interact through interfaces include classes, components and subsystems. [UML 1.5]
An architecture defines structure
An architecture defines behaviour
An architecture is concerned with significant elements

- The element relates to some critical functionality of the system
  - E.g. monetary transactions
- The element relates to some critical property of the system
  - E.g. reliability
- The element relates to a particular architectural challenge
  - E.g. external system integration
- The element is associated with a particular technical risk
- The element relates to a capability that is considered to be unstable
- The element relates to some key element of the solution
  - E.g. login mechanism
An architecture meets stakeholder needs

- The end user is concerned with intuitive and correct behavior, performance, reliability, usability, availability and security.
- The system administrator is concerned with intuitive behavior, administration and tools to aid monitoring.
- The marketer is concerned with competitive features, time to market, positioning with other products, and cost.
- The customer is concerned with cost, stability and schedule.
- The developer is concerned with clear requirements, and a simple and consistent design approach.
- The project manager is concerned with predictability in the tracking of the project, schedule, productive use of resources and cost.
- The maintainer is concerned with a comprehensible, consistent and documented design approach, and the ease with which modifications can be made.
An architecture comes in many forms
And …

- An architecture is part of a broader system
- An architecture embodies decisions based on rationale
- An architecture conforms to an architectural style
- An architecture is influenced by its environment
- An architecture influences organizational structure
- An architecture is present in every system
Agenda

- Architecture

  Architect

  Architecting
  - Requirements
  - Analysis and design
  - Implementation
  - Test
  - Project management
  - Configuration and change management
  - Environment

- Summary
Architect

- The architect is a technical leader
- The architect understands the software development process
- The architect has knowledge of the business domain
- The architect has technology knowledge
- The architect has design skills
- The architect has programming skills
- The architect is a good communicator
- The architect makes decisions
- The architect is a mentor
- The architect is aware of organizational politics
- The architect is a negotiator
- The architect role may be fulfilled by a team

“The life of a software architect is a long and rapid succession of suboptimal design decisions taken partly in the dark.” [Kruchten]
Agenda

- Architecture
- Architect

Architecting
- Requirements
- Analysis and design
- Implementation
- Test
- Project management
- Configuration and change management
- Environment

- Summary
Architecting

Diagram showing relationships between architectural components:

- Development Project
- Team
- Development Process
- Architect
- Architecting
- Concern
- Stakeholder
- Architecture
- Environment
- System
- Mission
- Rationale

Relationships include:
- Uses
- Has
- Performs
- Identifies
- Inhabits
- Provides
- Described by
- Develops
- Modifies
- Responsible for

Diagram outlines the process of architecting a software system, highlighting key components and their interconnections.
Architecting

- Architecting is a science
- Architecting is an art
- Architecting spans many disciplines
- Architecting changes emphasis over time
- Architecting involves many stakeholders
- Architecting is involved in tradeoffs
- Architecting considers reusable assets
- Architecting is both top-down and bottom-up
The benefits of architecting

- Architecting addresses system qualities
- Architecting drives consensus
- Architecting ensures architectural integrity
- Architecting helps manage complexity
- Architecting provides a basis for reuse
- Architecting reduces maintenance costs
- Architecting supports impact analysis
- Architecting supports the planning process
Documenting a software architecture
Architecture description – Kruchten 4+1 views
Architecture description – Siemens

Conceptual View

Module View

Execution View

Code View
Architecture description – RM-ODP
# Architecture description – Zachman framework

<table>
<thead>
<tr>
<th>Perspectives</th>
<th>Data</th>
<th>Function</th>
<th>Network</th>
<th>People</th>
<th>Time</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planner</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>contextual</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enterprise Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>conceptual</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Designer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>logical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technology Constrained Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Builder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>physical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detailed Representations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subcontractor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>out-of-context</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functioning Enterprise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Agenda

- Architecture
- Architect
- Architecting
  - Requirements
    - Analysis and design
    - Implementation
    - Test
    - Project management
    - Configuration and change management
    - Environment
- Summary
Requirements

- Concepts
  - Functional requirements
  - Non-functional requirements
    - Qualities
    - Constraints
- Work products (i.e. artifacts, deliverables)
  - Stakeholder requests
  - Use-case model
  - System-wide requirements
  - Changes cases
  - Prioritized requirement list
  - Software architecture document
## System-wide requirements example

<table>
<thead>
<tr>
<th>Category</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality</td>
<td>Online help</td>
<td>The system shall provide online help that can be downloaded from the auction site and installed locally, or accessed via a browser.</td>
</tr>
<tr>
<td>Reliability</td>
<td>Availability</td>
<td>The system shall be 99.9% available. Backup and maintenance operations shall not require system shutdown.</td>
</tr>
<tr>
<td>Performance</td>
<td>Response time</td>
<td>The response time for any query shall be less than 3 seconds when measured on a 100Mb local-area connection. The response time for all transactions, such as creation of an auction, shall be less than 5 seconds, where response time represents the time from submitting the transaction data to the time transaction results are reported to the user.</td>
</tr>
<tr>
<td>Supportability</td>
<td>Scalability</td>
<td>The system shall support 100,000 users and 5,000 concurrent users.</td>
</tr>
<tr>
<td>Constraint</td>
<td>Error management</td>
<td>All errors shall be time-stamped and logged in the system error file. Exception messages should identify the system element that threw the caught exceptions.</td>
</tr>
</tbody>
</table>
Analysis and Design

- Concepts
  - Architectural style
  - Design elements
    - Layers
    - Components/services
  - Reusable elements
    - Reference architectures
    - Patterns
  - Requirement realizations (functional and non-functional)
  - Architectural proof-of-concept

- Work products
  - Design model
  - Deployment model
  - Data model
  - Software architecture document
## J2EE Architecture

### Client

<table>
<thead>
<tr>
<th>Presentation Tier</th>
<th>Business Tier</th>
<th>Integration Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Device</td>
<td>Web Server</td>
<td>JDBC</td>
</tr>
<tr>
<td>HTML, XML, WML</td>
<td>Web Container</td>
<td>Relational Database</td>
</tr>
<tr>
<td></td>
<td>JSP</td>
<td>JavaMail</td>
</tr>
<tr>
<td></td>
<td>Servlet</td>
<td>Mail Server</td>
</tr>
<tr>
<td>Client Device</td>
<td>EJB Server</td>
<td>JNDI</td>
</tr>
<tr>
<td>Applet Container</td>
<td>EJB Container</td>
<td>Directory Service</td>
</tr>
<tr>
<td>Applet</td>
<td>J2EE Services</td>
<td>JMS</td>
</tr>
<tr>
<td>J2EE Services</td>
<td>EJB</td>
<td>Message Queue</td>
</tr>
<tr>
<td></td>
<td>J2EE Services</td>
<td>RMI</td>
</tr>
<tr>
<td>Application Client</td>
<td>J2EE Services</td>
<td>Java Application</td>
</tr>
<tr>
<td>Application Client</td>
<td>J2EE Services</td>
<td>CORBA Server</td>
</tr>
<tr>
<td></td>
<td>J2EE Services</td>
<td>RMI-IOP</td>
</tr>
</tbody>
</table>
J2EE Patterns – Presentation Layer
J2EE Patterns – Business Layer

Diagram showing the architecture of the J2EE Business Layer with components like Client, UseCaseDispatcher, BusinessDelegate, SessionFacade, and ValueObject interacting with the Web and EJB Containers.
Implementation

- Concepts
  - Implementation structure
  - Programming language constructs
  - Integration of implementation subsystems
- Work products
  - Implementation model
  - Software architecture document
Test

- Concepts
  - Unit testing
  - Integration testing
  - System testing
  - Acceptance testing

- Work products
  - Test suite
  - Test cases
  - Test scripts

The architecture must be both testable and tested
Project management

- Concepts
  - Schedule
  - Cost
  - Effort
  - Risk

- Work products
  - Project plan
  - Risk list
Configuration and change management

- Concepts
  - CM structures
  - Workspaces
  - Releases

- Work products
  - CM plan
Environment

- Concepts
  - Standards
  - Guidelines
  - Process

- Work products
  - Development process
  - Standards and guidelines
Summary

- The role of the architect is both broad and deep (in places 😊)
- Successful software architects treat “architecting” as an engineering discipline
- The role of the architect is more than “high-level design”
- Fulfilling the role of the architect is one of the most challenging … and rewarding … roles in the software industry