Architecture Development Methodology for Business Applications

Santonu Sarkar, Riaz Kapadia, Srinivas Thonse and Ananth Chandramouli

The Open Group Architecture Practitioners’ Conference

April 2004
Topics

• Motivation
• Methodology Overview
• Language and Tool Overview
• Summary
Motivation
Architecture Development Methodology is vital for enterprises running large number of application development projects

**Goals**

1. Standardizing architecture process across the organization
   - Standard approach, vocabulary, work-products

2. Streamlining work assignments for multiple teams
   - Avoiding information loss, enforcing traceability

3. Strengthening architecture areas
   - QoS analysis, integration, architecture evaluation which are high-importance areas

4. Improving productivity through tools and automation

**Illustrative Profile**
*(Reference Infosys 2003)*

- 500 concurrent projects
  - 10-100 team size

- Projects have Requirements, Architecture and Development teams
  - 1-5 architects / team

- Architecture teams start with requirements team & handover to Development teams
Different Aspect of Architecture

- TOGAF, Zachmann
- SEI ATAM, SAAM
- ISO 10746 RM-ODP
- IBM enterprise architecture standard
- C4ISR
- US Treasury Architecture Development process
- Bredemeyer architecture process

- Network Queuing theory, MVA
- Markov modeling for availability analysis
- Formal verification techniques
- Rule based system for applying architecture heuristics
- Engineering best practices

- TOGAF, Zachmann
- SEI ATAM, SAAM
- ISO 10746 RM-ODP
- IBM enterprise architecture standard
- C4ISR
- US Treasury Architecture Development process
- Bredemeyer architecture process

- Network Queuing theory, MVA
- Markov modeling for availability analysis
- Formal verification techniques
- Rule based system for applying architecture heuristics
- Engineering best practices

- TOGAF, Zachmann
- SEI ATAM, SAAM
- ISO 10746 RM-ODP
- IBM enterprise architecture standard
- C4ISR
- US Treasury Architecture Development process
- Bredemeyer architecture process

- Network Queuing theory, MVA
- Markov modeling for availability analysis
- Formal verification techniques
- Rule based system for applying architecture heuristics
- Engineering best practices

- TOGAF, Zachmann
- SEI ATAM, SAAM
- ISO 10746 RM-ODP
- IBM enterprise architecture standard
- C4ISR
- US Treasury Architecture Development process
- Bredemeyer architecture process

- Network Queuing theory, MVA
- Markov modeling for availability analysis
- Formal verification techniques
- Rule based system for applying architecture heuristics
- Engineering best practices
Scope of Methodology

Defining an architecture development methodology for business applications

InFlux™ Architecture framework

- Define prescriptive architectures
  - Standard stereotypes and patterns for J2EE, .NET...
  - Reference Architecture for Domain problems

- Tools to support the framework
  - Modeling tools
  - Automation – areas such as code generation

- Define standard architecture representation
  - Viewpoints
  - Meta-model & language
  - Standard architecture doc

- Strengthen architecture areas
  - Methods/techniques for various areas (availability, integration, performance, variability etc.)

- Formalize Process for Architecture definition
  - Architecture track
  - Flavors of process

Architecture Development Methodology for Business Applications
Methodology Overview
While creating the Architecture Framework certain guiding principles have been followed:

**Adopt Industry Standards for normative set of terminologies and notations to define architecture**
- Architecture definition as per IEEE 1471
- Viewpoint definition as per RM-ODP 10746-3
- QoS framework definition as per ISO/IEC 9126

**Flexible for Distributed Development Model**
- Should have traceability of decisions
- Phase based approach - should define hand-offs from one phase to another

**Formalize Representation with Meta-model and Tool**
- Architecture creation through Views
- Report Generation
- Architecture Analysis
- Reasoning/Inference capability
- Architecture Dynamism

**Have Separation of Concerns to address architectural complexities**
- Each viewpoint addresses concerns of certain stakeholders

**Should facilitate strong Upstream and Downstream integration**
- Derisk downstream implementation
- Architecture should be executable for use by downstream team

**Provide Tailorability to address various project contexts**
- Create a generic Architecture process independent of technology
- Tailor the generic process to a Technology specific prescriptive one
  - J2EE
  - .NET
Architecture Process – Context

- Consists of 3 phases - each produces a deliverable
- Overlaps with requirement and development process
- Focuses on functional, information, technical and infrastructure aspects

Methodologies
- Performance Engineering
- Availability Analysis
- Product Selection

Strategize
- Solution Brief

Define
- System Architecture

Implement
- Architecture Realization

Source: InFlux architecture track, Infosys
A Phase is a part of the architecture lifecycle. Each phase defines certain aspects of the architecture by creating the views associated with the viewpoints.

- **Define**
  - Computational Viewpoint
  - Information Viewpoint
  - Engineering Viewpoint

- **Strategize**
  - Enterprise Viewpoint

- **Implement**
  - Technology Viewpoint
  - **Software Organization** Viewpoint (New addition)
  - Engineering Viewpoint

- **Solution Brief**
  - Enterprise Viewpoint

- **Architecture Realization**

- **Proof of Concept**

Source: InFlux architecture track, Infosys

Architecture Development Methodology for Business Applications
Architecture Process is driven by 3 broad factors

Based On
- IEEE 1471 recommended architecture
- ISO/IEC RM-ODP 10746 recommended viewpoints
- ISO 9126 recommended QoS Framework

System Architecture Document contains all viewpoints

Architecture Development Methodology for Business Applications

Source: InFlux architecture track, Infosys
Process Development Approach

Create Baseline Process

Review with Community

Prioritize & Layout roadmap

Refine Process with community

Piloting

Define Prescriptive Architectures

Define Meta-model

Develop Architecture tools

Rollout, Education & Support

Public Draft

Public Release

Deployment

Community Draft

• Quality team
  • Education team

• Core team
  • Architect Community

• Intranet SourceForge based development

Create Baseline Process

Review with Community

Prioritize & Layout roadmap

Refine Process with community

Piloting

Define Prescriptive Architectures

Define Meta-model

Develop Architecture tools

Rollout, Education & Support

Source: InFlux architecture track, Infosys

Note: Community refers to Infosys architects, Public refers to Infosys users and stakeholders

• Adopted concepts from SPEM, OMG
• Unique concepts based on Infosys experience
  – Track (collection of activities)
  – Building processes from disciplines
  – Formalization of framework types
  – Tailoring feature for project specific customization
  – Concept of reference work-products (knowledge elements)

Architectural Development Methodology for Business Applications
Process Development Learnings

• Community based development approach for quick adoption
  – Involvement of key stakeholders ensures easy deployment by the practitioners
  – Involvement of Quality department ensures faster broad-basing

• Lightweight process to cover most aspects of architecture and tailor it to suite the context

• Global development approach by introducing phase based development
  – Strategize phase can be at the client location whereas the detailed architecture development can be performed off-shore
Language and Tool Overview
Architecture is captured using a graphical modeling language – EAML

- Uses IEEE 1471 Conceptual Framework as the basis.
- Defines notations for all the views and viewpoints.
- Facilitates traceability among various views
- Uses OMG MOF to organize the metamodel notations
- Introduces component-connector concepts in modeling
- Provides notations for design rationale
Overview of some modeling concepts

• Models business processes
• Facilitates creation of functional architecture from business processes
• Allows functional decomposition
• Maintains traceability

Source: InFlux architecture track, Infosys
Architecture Modeling tool allows to draw views associated with viewpoints. Each view is a diagram.

<table>
<thead>
<tr>
<th>Viewpoint</th>
<th>View</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computational</td>
<td>Context Diagram</td>
<td>System, interconnection, protocol, service</td>
</tr>
<tr>
<td></td>
<td>Functional Diagram</td>
<td>System, subsystem, interface, messages</td>
</tr>
<tr>
<td>Engineering</td>
<td>Logical Diagram</td>
<td>Subsystems, components mapped into tiers</td>
</tr>
<tr>
<td>Technology</td>
<td>Application Diagram</td>
<td>Tier with COTS product and application mapping</td>
</tr>
<tr>
<td></td>
<td>Deployment Diagram</td>
<td>Hardware, devices, networks for production, QA and Development environment</td>
</tr>
<tr>
<td>Information</td>
<td>Domain Diagram</td>
<td>Domain entities and Relationships</td>
</tr>
<tr>
<td>Software Organization</td>
<td>Realization Diagrams</td>
<td>Application frameworks, patterns</td>
</tr>
<tr>
<td></td>
<td>Layer Diagram</td>
<td>Packages, layers, dependencies</td>
</tr>
</tbody>
</table>

Source: InFlux architecture track, Infosys
Illustration: *The Unified Billing Solution is a global effort to create a single billing solution applicable to all regions*

- Starts with understanding Architecture Principles, IT Gaps, IT Strategy....
- Uses business stereotypes to characterize the application
- Enterprise viewpoint is captured through a set of business processes
- Defines Computation, Information, Technology and Engineering viewpoints
- Software organization viewpoints for downstream integration
Summary

• Architecture methodology is vital for enterprises running large number of application development projects
  – To enforce standardization of work-products and process
  – To promote reuse, automation

• InFlux Architecture track is based on certain guiding principles –
  – separation of concerns
  – upstream and downstream integration
  – using community standards development model

• InFlux meta-model has semantics for representing architecture
  – combination of industry standards
  – use of views and viewpoints