A Federated Approach to Enterprise Architecture Model Maintenance

2nd EMISA Workshop, 8–9/10/2007, St. Goar
Agenda

1. Necessity and Challenges of Enterprise Architecture Maintenance
2. Approaches to Maintain Enterprise Architecture
3. Case Study
4. Conclusion
Enterprise Architecture Goal Hierarchy

- Creation and preservation of transparency
  - Reengineering and redesign of functionality clusters
  - Extension of outdated, partial, inconsistent design documents
  - Measurability of dis-alignment, un-supported business requirements, unnecessary IT functionalities

- IT simplification and consolidation
  - Creation of fine granular, reusable functionality chunks
  - Reuse of functionalities

- Creation and preservation of flexibility
  - Easier adaptation of applications to specified changes

- Creation and preservation of agility
  - Preparation of applications for future, yet unspecified changes
Why Enterprise Architecture matters
Enterprise Architecture

Business Engineering Framework

- **Strategy Layer**
  - Designing the Strategy
    - Business network models
    - Customer process models
    - Output models
    - Corporate objectives

- **Organization Layer**
  - Designing the Organization
    - Process map
    - Process models
    - Organizational structure
    - Information map

- **Integration Layer**
  - Designing the Integration
    - Application map
    - Enterprise services

- **Software Layer**
  - Designing the Software
    - Software components
    - Software services
    - Data models

- **Infrastructure Layer**
  - Designing* the IT Infrastructure
    - Platform infrastructure
    - Network infrastructure
EA: A Cross-layer View Comprised of ALL Aggregate Business Engineering Artefacts
Agenda

1. Necessity and Challenges of Enterprise Architecture Maintenance
2. Approaches to Maintain Enterprise Architecture
3. Case Study
4. Conclusion
Approach (I)
A Holistic EA Model

remodeling

model interpretation and consolidation
Approach (II)
A Federated EA Model

- Data delivery
- Contract
- Transformation
- Rules
- Maintenance schedule
EA Maintenance Process

1. initiate maintenance cycle
2. deliver model data
3. check consistency
4. revise inconsistencies
5. prepare change report
6. check intended changes
7. coordinate vetoes
8. authorize repository update
9. perform repository update
10. communicate repository update

Data Owner

(1) initiate maintenance cycle (2) deliver model data

Repository Manager

(3) check consistency (5) prepare change report

(4) revise inconsistencies

(6) check intended changes

(7) coordinate vetoes

(8) authorize repository update

(9) perform repository update

(10) communicate repository update
# RACI Matrix

## Mapping Roles and Activities

<table>
<thead>
<tr>
<th>Activities</th>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Initiate maintenance cycle</td>
<td>Chief Enterprise Architect</td>
</tr>
<tr>
<td>(2) Deliver model data from specialized architecture</td>
<td>EA Coordinator</td>
</tr>
<tr>
<td>(3) Check data consistency</td>
<td>EA Repository Manager</td>
</tr>
<tr>
<td>(4) Revise inconsistencies</td>
<td>EA Stakeholder</td>
</tr>
<tr>
<td>(5) Prepare change report &amp; notify affected stakeholders</td>
<td>Data Owner</td>
</tr>
<tr>
<td>(6) Check intended changes</td>
<td>Responsible Position working on the activity</td>
</tr>
<tr>
<td>(7) Coordinate vetoes</td>
<td>Accountable Position with yes/no authority</td>
</tr>
<tr>
<td>(8) Authorize repository update</td>
<td>Consult Position involved prior to decision or action</td>
</tr>
<tr>
<td>(9) Perform repository update</td>
<td>Inform Position that needs to know of the decision or action</td>
</tr>
<tr>
<td>(10) Communicate repository update</td>
<td></td>
</tr>
</tbody>
</table>

### Responsible
Position working on the activity

### Accountable
Position with yes/no authority

### Consult
Position involved prior to decision or action

### Inform
Position that needs to know of the decision or action
Case Study
The Company

- Large financial retailer in Switzerland

- Products and services for every situation:
  payments, investments, financial security, credits

- Private customers, associations, companies, and public corporations

- 2006
  total assets > 50 billion CHF
  number of accounts > 3 million
Case Study
Initial Situation – EA History

- Numerous detailed but isolated models/management systems (products, processes, applications, data, ...)
- No consistent overall picture of the company

- Jan 05: EA gets introduced to management body – decision for a proof of concept
- Aug 05: 1st prototype: business continuity planning
- Feb 06: 1st release: business continuity planning and IT operations
- Jun 06: 2nd release: strategy and project selection
- Oct 06: 3rd release: data architecture
- Jan 07: standard operations
Impact Analysis

**Customer process**
- service activity: claim management
- customer activity: claim report
- customer process: motor insurance

**Process map**
- process: claim management

**Process flow**
- process flow: claim management

**Application repository**
- applications: claims core
- application domain: claim/benefit management

**Components and platforms**
- platform: J2EE
- software components: order, events, partner, claims catalogue
Case Study
Defining EA Application Goals and Scenarios

- EA is a **high-level** model of the company
- Supporting EA relevant decisions which require knowledge across multiple layers of the EA stack
- Application scenarios supported by EA may be:
  - Business continuity planning (BCP)
  - Portfolio management (applications, partners, projects)
  - Evaluation of strategic options
  - IT service management
Case Study

EA Program

- EA combines and consolidates existing specialized architecture models:
  - To achieve a consistent overall picture, maintenance roles and processes need to be defined
  - EA has a passive role – EA as a service
  - EA does not own any EA models

![Diagram showing EA-REPO and its connections to Strategic goals, Process models, Application repository, Hardware repository, and Controlling data.]
## Agenda

1. Necessity and Challenges of Enterprise Architecture Maintenance
2. Approaches to Maintain Enterprise Architecture
3. Case Study
4. Conclusion
Conclusion

Lessons Learned

- Using existing models fosters acceptance and saves costs

- Aiming at well defined and concrete as opposed to abstract problems allows for a pragmatic approach

- It is a waste of time to introduce EA in divisions which are not willing to create transparency

- Using/aligning at established meta-models allows for a switch to commercial EA products after starting with an individual solution
Contact

Dr. Stephan Aier
stephan.aier@unisg.ch
www.iwi.unisg.ch
+41 71 224 3360