Software Architecture Beyond the Blueprints

"Aligning Software Architecture with the facets of Software Development - Business, Management, Engineering and Organization"

Eldo
Architect, Philips Healthcare
eldo.issac@philips.com

SATURN 2009, North America
May 4-7, Pittsburgh, Pennsylvania, USA

The Facts

• Every Software has an architecture.
• Every business dealing with Software has a Software Architecture Entity (SAE).
• Like Software Architecture, effective realization of SAE varies from instance to instance – it depends on the ecosystem.

• When it comes to software, it’s not about whether there is an architecture or not, but it’s more about whether its full potential is achieved or not.

• Spectrum of recognition of software architecture varies from ‘everything is code’ to ‘it is the lifeline’ of the organization.

• Hence the effectiveness/ROI of architecture varies from situation to situation.
Key Tenets of SAE are:
- Architecture and Design Process
- Architect(s)
- The Architecture

Effectiveness of an architecture needs to be measured as the effectiveness of SAE, not just the ‘ility’s of architecture.

This will also help to establish architecture as a valuable entity in organizational maps.

The ecosystem consists of:
- Business
- Engineering
- Management
- Organization

Each of these facets of the ecosystem will have different set of expectations from architecture and vice versa.

SAE needs to support and be supported by the ecosystem it is living in. It is a two way street to success!
Need for aligning SAE with its Ecosystem

Quality of the blueprint itself is not going to make the architecture or project successful.

An average architecture that fits well for the ecosystem may do better than an excellent architecture that can't be supported by the ecosystem.

If positioned, SAE has many qualities than can amplify the effectiveness of all facets.

Ecosystem play a role in constraining the architectural options and deliverables.

- If SAE is not aligned with facets of ecosystem
  - It will undermine the ability of SAE to contribute to the ecosystem (i.e. ROI). Hence weaken the ecosystem itself.
  - It will undermine the support architecture needs from the ecosystem and thereby chip way at the quality and success of the architecture itself.

- Each of the facets of ecosystem has different dimensions that affect SAE

Example: SAE and Organizational Facet - a mutual impact scenario

Both architecture cartoons are trying to solve the same problem.

This is a common problem in product acquisitions and portfolio management.

Each model puts different constraints/expectation on organizational facet supporting it (in addition to its impact on other facets and its dimensions).

Based on situation, a SOA based model may support more distributed development, organizational scalability, division of responsibility/ownership, parallel development...etc. This may also come with some disadvantage when it comes to low level leverage points.... 'but its all about trade-offs'
Example: Software Development Process impact on SAE

Software Development Process (SDP) has great impact on architecture deliverables, effort, effort distribution and roadmap.

SDPs are very ecosystem specific

Some SDP tries to solve arch (major or detailed) earlier where as others push it further down.

e.g. when it comes to agile models, refactoring and architecting needs to be coordinated across many sprint teams as it is being implemented by the teams. It needs a very different way of thinking and organizational and management support.

Agility Vs Planned models - an architecture perspective

1. Goal: Carry maximum load but reduce variability carried at anytime
2. Lesser the variability bigger the truck that can be used
3. Increase in variability is compensated by reducing the load and iterating more – drive smaller trucks. This can be done at phase levels or end to end.
4. Bigger the truck lesser the overhead, i.e. more efficiency
5. Variability in an inner circle is going to force iteration in the outer based on variability flow

Process settles when: Maximum amount of load is carried most efficiently with rational amount of variability for the business.
SAE and Ecosystem: Mutual Impact

To be successful, there needs to be a seamless marriage between SAE and the ecosystem. Each of the constraints/requirements needs to be analyzed and addressed.

Any assumption that just having an architect(s) is going to make a project successful, irrespective of the ecosystem, is not on solid ground.

On the flip side, every architecture cannot be supported by every ecosystem.

First step to success here is to understand these touch points between the SAE and its ecosystem.

• The Ecosystem puts constraints/requirements on the SAE
  – Each facets of the organization has different dimensions which initiates these constraints/requirements
• Also, SAE has constraints/requirements that needs to be met by the ecosystem to achieve its goals.
• Alignment of these requirements affects all elements of SAE – Architecture and design process, architects and the architecture – and the ecosystem.

Dimensions of Interest for SAE [1]

|----------------|-----------------|--------------|-------------------|---------------|----------------------|----------------|------------------|----------------------|----------------|------------------------|---------------------|----------------|--------------------------|-------------------------|

<table>
<thead>
<tr>
<th>Organization</th>
<th>Team Size</th>
<th>Team Distribution</th>
<th>Organizational Boundaries</th>
<th>Employee Skill</th>
<th>External Process Dependencies</th>
<th>Organizational Structure</th>
<th>Business Model</th>
<th>Team and Organizational Roles..etc.</th>
</tr>
</thead>
</table>

| Engineering/Architecture | Size | Complexity | Quality Attributes | Cost | Team Knowledge | Team Size and Distribution | Documentation | External Dependencies | Architecture Contribution to Other facets | Market and Requirement Stability | Organizational Culture | Resources (skills, time, money) | Roles and responsibilities Deliverables..etc. |
### Dimensions of Interest for SAE [2]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Team Knowledge</td>
<td>Team knowledge</td>
<td>Size</td>
<td>Release models</td>
<td>Support Model</td>
</tr>
<tr>
<td>Complexity</td>
<td>Complexity</td>
<td>Complexity</td>
<td>Customer update models</td>
<td>Release Model</td>
</tr>
<tr>
<td>Resources (skill, time, money)</td>
<td>Review Guidelines and enforcement Technology</td>
<td>Documentation</td>
<td>Support Models</td>
<td>Service pack, hot fix models</td>
</tr>
<tr>
<td>Coordination</td>
<td>Size</td>
<td>Automation</td>
<td>Software Delivery models</td>
<td>Team structure</td>
</tr>
<tr>
<td>Quality</td>
<td>Test Model</td>
<td>Roles, responsibilities and deliverables</td>
<td>Licensing models</td>
<td>Team skill levels</td>
</tr>
<tr>
<td>Size</td>
<td>Documentation</td>
<td>Interaction with other facets</td>
<td>Documentation</td>
<td>Delivery Models</td>
</tr>
<tr>
<td>Estimating and Planning</td>
<td>Team isolation</td>
<td>Requirements gathering and documentation</td>
<td>Size</td>
<td>Size</td>
</tr>
<tr>
<td>Review Model</td>
<td>Framework...etc.</td>
<td>Business Domain...etc.</td>
<td>Complexity</td>
<td>Complexity</td>
</tr>
<tr>
<td>Domain</td>
<td></td>
<td></td>
<td>Customization</td>
<td>Domain...etc.</td>
</tr>
<tr>
<td>Documentation</td>
<td></td>
<td></td>
<td>OEM models</td>
<td></td>
</tr>
<tr>
<td>Design Maintenance</td>
<td></td>
<td></td>
<td>Partnerships...etc.</td>
<td></td>
</tr>
<tr>
<td>Refactoring</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team Composition</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer Interaction Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements...etc.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Aligning SAE with dimensions of the ecosystem

- Expectations are generally two way.
- Some of these expectations may get addressed by traditional ‘utility trees’ (non functional attributes), some of them would not.
  - E.g. Team Distribution
- These requirements may affect different aspects of SAE – process, architects and/or architecture
  - E.g. Team distribution may put demands on, architecture and design process, architecture, documentation, architect(s), etc.
- This calls for a need for incorporating an Ecosystem Analysis in setting up an SAE.

- Traditional analysis of architecture focuses on the architecture, not on how it is done in an ecosystem.
- For example, how an architecture needs to be delivered to a team may depend on team size, distribution, knowledge level, team structure...etc.
- Also architects/architecture may need a better say on the SDP which will be used to build the system. Architecting a ‘fighter jet’ using agile refactoring model may be over the top.
Example: Ecosystem Analysis on SAE

Just like in architecting a system there will be different ways to address each constraint or requirement.

But it is more important to understand them and have an action plan.

In some cases, it would be more of a ‘nice to have’ than a mandate. For example you may not want to create a distributed architecture just to have teams distributed, that may be silly.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Direction</th>
<th>Constraint/Requirement</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Org-Team Distribution</td>
<td>Ecosystem → SAE</td>
<td>1. Need to support teams distributed in different locations 2. Needs to support teams operating in different continents 3. Etc.</td>
<td>1. Identify and incorporate representations of architecture which allows teams to be decentralized. 2. Identify local architecture proxies/architects. 3. Use SOA (some thing more distributed) as preferred guideline 4. Etc.</td>
</tr>
</tbody>
</table>

This could be part of Utility Tree ++!

Example: Steps to improve alignment of SAE with Ecosystem [1]: Process

When it comes to SDPs, the answer is ‘it depends’.

Every SDP has its home ground(s), none of them fit to all situations.

But there are common patterns of steps that you could take to improve the effectiveness of SAE with in a SDP.

• Create an Architecture and Design Process based on Ecosystem Analysis
  – It is not safe to assume any software development process could build any architecture.
  – Each process slices, emphasizes and connect tenets of software engineering differently – hence each has its home ground.
  – Amend engineering processes with Architecture and Design Process.
Example: Steps to improve alignment of SAE with Ecosystem [2]: Architecture

- Create Consumer Centric documentation model for architecture based on an ecosystem analysis
  - Architecture needs to empower a wide set of audience (from all facets) with different skill set and needs
  - Its effectiveness is only as good as the understanding of the consumer

Example: Steps to improve alignment of SAE with Ecosystem [3]: Process

- Stakeholder Centric Architecture Modeling
  - Conceptualization phases are most crucial for success of projects
  - Architecture can contribute and benefit greatly by participating and empowering all stakeholders.
Examples: Steps to improve alignment of SAE Tenets with Ecosystem [4]: Architects

- Create Architecture Team based on an ecosystem analysis
  - There are many architecture titles around, which one you need?
  - What kind of skills set and distribution you need?
  - All these depends on the ecosystem.
- When it comes to architecture ‘design by committee’ is the last thing you want.
- Also need to avoid architecture becoming a bottle neck at any cost as well.

Ecosystem would decide the structure and resources needed for an architecture team.

Example: Steps to improve alignment of SAE with Ecosystem [6]: Process

- Improve Planning Framework by leveraging SAE
  - Effective planning and business projection needs an evolving plan.
  - Plan without logical structure and ability to build confidence is volatile – irrespective of dev process.
  - Architecture and Plan, both benefit from each other
Example: Steps to improve alignment of SAE with Ecosystem [7]: Architecture/Process

- Improve Team Ownership of architecture
  - Reduce “Lost in Translation” Effect – most expensive problems in designs.
  - Reduce hand offs and manage/promote ownership of architecture
  - Get architecture deliverables to feature/subsystem level

Summary

- To be successful architecture has to go way beyond just the blue prints.
- This calls for an SAE (Software Architecture Entity), which defines
  - Architecture and Design Process
  - Architect(s) / Resource
  - The Architecture
- There are many influential factors in the ecosystem that has great impact on the SAE
- This ecosystem consists of different facets (Business, Engineering, Management and Organization) and their numerous dimensions
- So analyzing and aligning the SAE with the ecosystem and its dimensions is key to success – a ‘Utility Tree’ needs to cover all aspects of SAE.