



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

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

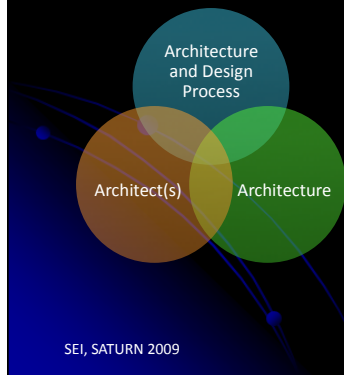
Spectrum of recognition of software architecture varies from 'everything is code' to 'it is the lifeline' of the organization.

Hence the effectiveness/ROI of architecture varies from situation to situation.

- 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.

Software Architecture Entity (SAE)

'Body Surface Area' of architecture goes way beyond the blue prints. There are lot more touch points (impact zones) between SAE and its ecosystem.



SEI, SATURN 2009

Architecture Beyond Blueprints

3

- 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.

SAE's Ecosystem

Software Architecture is not just an Engineering problem – which has been a traditional view.

Value/success of architecture would depend on SAE alignment with all facets of the ecosystem



SEI, SATURN 2009

Architecture Beyond Blueprints

4

- 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 blue print 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 there by chip way at the quality and success of the architecture itself.
- Each of the facets of ecosystem has different dimensions that affect SAE

SEI, SATURN 2009

Architecture Beyond Blueprints

5

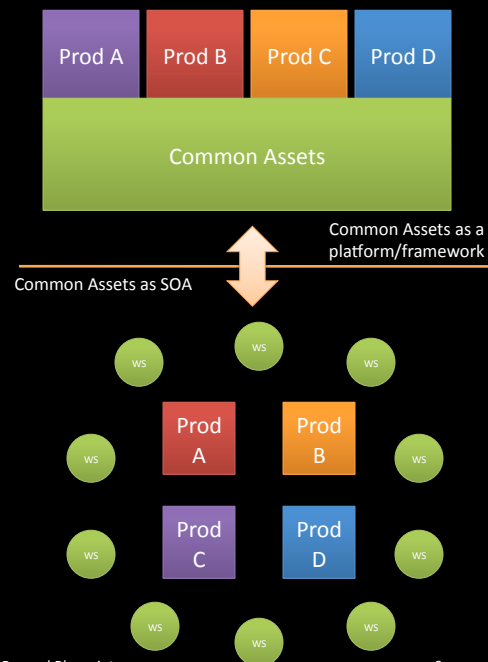
Example: SAE and Organizational Facet - a mutual impact scenario

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'



SEI, SATURN 2009

Architecture Beyond Blueprints

6

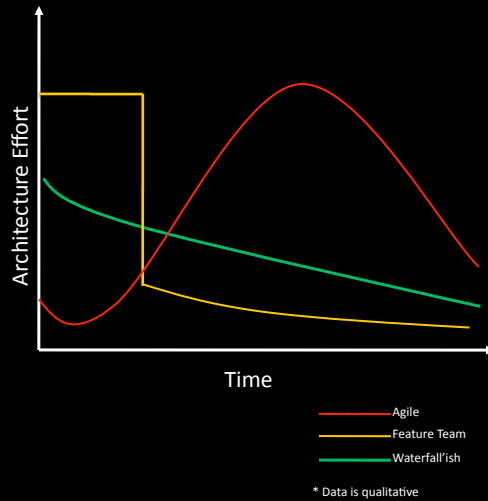
Example: Software Development Process impact on SAE

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.



SEI, SATURN 2009

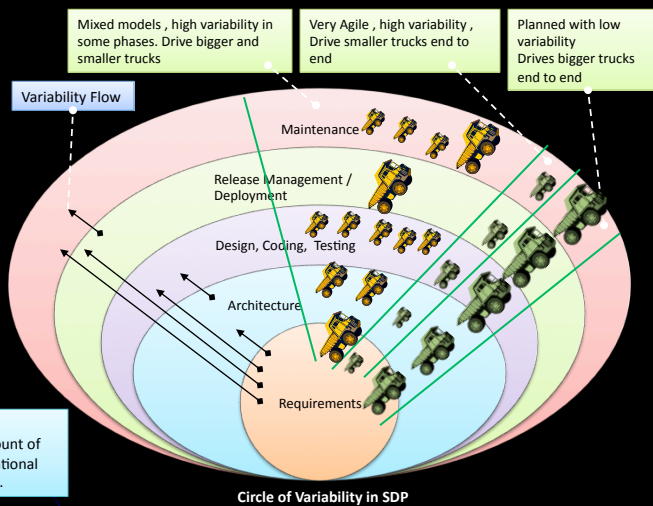
Architecture Beyond Blueprints

7

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.



SEI, SATURN 2009

Architecture Beyond Blueprints

8

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 project successful, irrespective the ecosystem, is not on solid grounding.

On the flip side, every architecture can't 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 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

SEI, SATURN 2009

Architecture Beyond Blueprints

9

Dimensions of Interest for SAE [1]

Business	Management	Organization	Engineering/Requirement Analysis	Engineering/Architecture
Business Model	Software Development	Team Size	Market Maturity and Stability	Size
Market Maturity	Process	Team	Validation	Complexity
Project Size	Planning	Distribution	Resource Needs	Quality Attributes
Domain Complexity	Estimation	Organizational	Tools support	Cost
Quality Needs	Risk Management	Boundaries	Customer Interaction Model	Team Knowledge
Customer Involvement & Accessibility	Compensating for deviation	Employee Skill	Prioritization	Team Size and Distribution
Release Models	Predictability and	External Process	Availability of Domain	Documentation
Delivery Models	Reliability of plan	Dependencies	Expertise	External Dependencies
Implementation	Business Plan Support	Organizational	Domain Size	Architecture Contribution to Other facets
Models	Roles and Responsibility	Structure	Domain Complexity	Market and Requirement
Upgrade Models	Team Structure	Business Model	Traceability	Stability
Interaction with other facets	Skills and Resource Needs	Team and	Interaction with other facets.	Organizational Culture
Business Plan Support	Training	Organizational	Deliverables	Resources (skills, time, money)
Risk Management	Traceability	Roles..etc.	Roles and	Roles and responsibilities
Skills and Resource	Scope Management..etc.		Responsibilities ..etc.	Deliverables..etc.
Market/Domain knowledge..etc.				

SEI, SATURN 2009

Architecture Beyond Blueprints

10

Dimensions of Interest for SAE [2]

Engineering/Design	Engineering/Coding	Engineering/Testing	Engineering/Release Management	Engineering/Support
Team Knowledge	Team knowledge	Size	Release models	Support Model
Complexity	Complexity	Complexity	Customer update models	Release Model
Resources (skill, time, money)	Review	Documentation	Support Models	Service pack, hot fix models
Coordination	Guidelines and enforcement	Customer interactions	Software Delivery models	Team structure
Quality	Technology	Automation	Licensing models	Team skill levels
Size	Size	External Bodies/Standards	Documentation	Delivery Models
Estimating and Planning	Team distribution	Release models	Size	Size
Review Model	Test Model	Roles, responsibilities and deliverables	Complexity	Complexity
Domain	Documentation	Interaction with other facets	Customization	Domain ...etc.
Documentation	Team isolation	Requirements gathering and documentation	OEM models	
Design Maintenance	Framework...etc.	Business Domain ...etc.	Partnerships ...etc	
Refactoring				
Team Composition				
Customer Interaction Model				
Requirements ...etc.				

Aligning SAE with dimensions of the ecosystem

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.

- 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.

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.

Dimension	Direction	Constraint/Requirement	Action
Org -Team Distribution	Ecosystem → SAE	<ol style="list-style-type: none"> 1. Need to support teams distributed in different locations 2. Needs to support teams operating in different continents 3. Etc. 	<ol style="list-style-type: none"> 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.
Mgmt-Process	SAE → Ecosystem	<ol style="list-style-type: none"> 1. Needs upfront platform work 2. Needs to carry 'medium size load' as subsystems are too expensive to break 3. Needs to enforce guidelines 	<ol style="list-style-type: none"> 1. Use 'feature team' models for development process. 2. Establish review points and hand off points and invest in tools. 3. Etc.

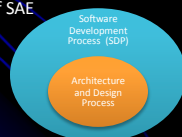
This could be part of Utility Tree ++ !

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.



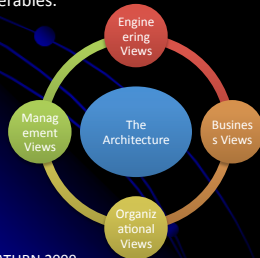
• Create an Architecture and Design Process based on Ecosystem Analysis

- 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

Documenting architecture in a way that consumers can understand is a key to effectiveness.

Types of representations of architecture and level of details in different representation depend a lot on the consumers of the deliverables.



SEI, SATURN 2009

Architecture Beyond Blueprints

15

- 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

Irrespective of software dev process, closely bind initial phases of software to architecture.



SEI, SATURN 2009

Architecture Beyond Blueprints

16

- 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

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

- 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.

SEI, SATURN 2009

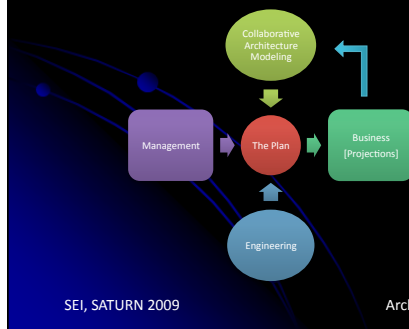
Architecture Beyond Blueprints

17

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

From initial draft to a high confidence plan, architecture can provide structure, estimates(time & resource), guidance, validation and mitigates risk to the plan....

- 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



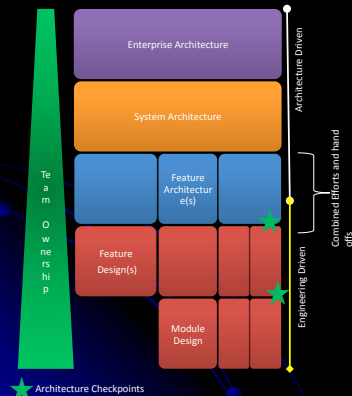
SEI, SATURN 2009

Architecture Beyond Blueprints

18

Example: Steps to improve alignment of SAE with Ecosystem [7] : Architecture/ Process

Increases knowledge and ownership of architecture at all levels....



SEI, SATURN 2009

Architecture Beyond Blueprints

19

- 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.



Eldo
eldo.issac@philips.com

SEI, SATURN 2009

Architecture Beyond Blueprints

20