Architecture-Centric Procurement

SATURN Conference
April 29 – May 3, 2013
Minneapolis, MN

John Bergey
Larry Jones

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-2612
Presentation Outline

**Essential Architecture Practices**

Incorporating Architecture Practices into the Procurement Context

Conclusion
What is Architecture-Centric Engineering?

Architecture-Centric Engineering (ACE) is the discipline of using architecture as the focal point for performing ongoing analyses to gain increasing levels of confidence that systems will support their missions.

Architecture is of enduring importance because it is the right abstraction for performing ongoing analyses throughout a system’s lifetime.

The SEI Architecture Practices Group develops principles, methods, foundations, techniques, tools, and materials to support creating, fostering, and stimulating widespread transition of the ACE discipline.
Formal Definition of Software Architecture

“The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them and properties of both.”

If function were all that matters, any monolithic implementation would do, \textit{but other things matter}…

The important quality attributes and their characterizations are key.

- Modifiability
- Interoperability
- Availability
- Security
- Predictability
- Portability

\textbf{The Non-functional Requirements}
Specifying Quality Attributes

Quality attributes are rarely captured effectively in requirements specifications; they are often vaguely understood and weakly articulated.

Just citing the desired qualities is not enough; it is meaningless to say that the system shall be “modifiable” or “interoperable” or “secure” without details about the context.

The practice of specifying quality attribute scenarios can remove this imprecision and allows desired qualities to be evaluated meaningfully.

A quality attribute scenario is a short description of an interaction between a stakeholder and a system and the response from the system.
Parts of a Quality Attribute Scenario

- **SOURCE**
- **ENVIRONMENT**
  - Artifact: Process, Storage, Processor, Communication
  - Stimulus
- **RESPONSE**
  - Response
  - Measure
Example Quality Attribute Scenario

A “performance” scenario: A remote user requests a database report under peak load and receives it in under 5 seconds.

Stimulus: Remote user

Environment: Database under peak load

Artifact: Process, Storage, Processor, Communication

Response: under 5 seconds
Architecture-Centric Activities

Architecture-centric activities include the following:

• creating the **business case** for the system
• understanding the **requirements**
• creating and/or selecting the architecture
• documenting and communicating the architecture
• analyzing or evaluating the architecture
• implementing the system based on the architecture
• ensuring that the implementation **conforms** to the architecture
• evolving the architecture so that it **continues to meet business and mission goals**
## Some SEI Techniques, Methods, and Tools

<table>
<thead>
<tr>
<th>Activity</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>creating the <strong>business case</strong> for the system</td>
<td>Quality Attribute Workshop (QAW) *</td>
</tr>
<tr>
<td>understanding the <strong>requirements</strong></td>
<td>Mission Thread Workshop (MTW) *</td>
</tr>
<tr>
<td>creating and/or selecting the <strong>architecture</strong></td>
<td>Attribute-Driven Design (ADD)</td>
</tr>
<tr>
<td>documenting and communicating the <strong>architecture</strong></td>
<td>Views and Beyond Approach; AADL</td>
</tr>
<tr>
<td>analyzing or evaluating the <strong>architecture</strong></td>
<td>Architecture Tradeoff Analysis Method (ATAM) *; SoS Arch Eval *; Architecture Readiness Review (ARR) ; AADL</td>
</tr>
<tr>
<td>ensuring <strong>acquisition strategy is architecture-centric</strong></td>
<td>Acquisition Planning Workshop (APW)</td>
</tr>
<tr>
<td>ensuring that the implementation <strong>conforms</strong> to the <strong>architecture</strong></td>
<td></td>
</tr>
<tr>
<td>evolving the architecture so that it <strong>continues to meet business and mission goals</strong></td>
<td>Architecture Improvement Workshop (AIW)* and ArchE</td>
</tr>
<tr>
<td>ensuring <strong>use of effective architecture practices</strong></td>
<td>Architecture Competence Assessment</td>
</tr>
</tbody>
</table>

* = indicates a software engineering method that has been extended to systems engineering
Building the Business Case for the System

Some common business / mission drivers for systems include

- Reduce total cost of ownership
- Improve capability/quality of system
- Improve market position
- Support improved business processes
- Improve confidence in and perception of system

Results gleaned from

- 25 architecture evaluations
  - 18 government systems, 7 commercial systems
- 190 distinct business goals

http://www.sei.cmu.edu/reports/05tr021.pdf
Understanding the Requirements – The SEI’s Quality Attribute Workshop

The purpose of the SEI Quality Attribute Workshop (QAW) is to discover, early in the life cycle, the driving quality attribute requirements of a software-intensive system.

**QAW Steps**

1. QAW Presentation and Introductions  
2. Business/Programmatic Presentation  
3. Architectural Plan Presentation  
4. Identification of Architectural Drivers  
5. Scenario Brainstorming  
6. Scenario Consolidation  
7. Scenario Prioritization  
8. Scenario Refinement

http://www.sei.cmu.edu/library/abstracts/reports/03tr016.cfm
Analyzing the Architecture – SEI’s Architecture Tradeoff Analysis Method® (ATAM®)

The ATAM is an architecture evaluation method that focuses on multiple quality attributes.
View-Based Documentation

Views give us our basic principle of architecture documentation

Documenting an architecture is a matter of documenting the relevant views, and then adding documentation that applies to more than one view.

The choice of views used depends on the nature of the system and the stakeholder needs.
Presentation Outline

Essential Architecture Practices

Incorporating Architecture Practices into the Procurement Context

Conclusion
What is Architecture-Centric Procurement?

Architecture-Centric Procurement is the act of using \textit{architecture and architectural practices} as a \textit{contractual means} to reduce risk and gain added confidence that the system being procured will be capable of achieving its intended mission.  

[Bergey 2010]

It is a \textit{quality attribute} and \textit{architecture-driven} approach to reducing software procurement risk.

It involves \textit{being proactive} and \textit{selectively incorporating architectural practices} in a procurement* in a \textit{prescribed manner}.

* In an RFP/Contract or Task Order
Why Architecture-centric Procurement?

An architecture-centric approach raises the "performance bar" on both the acquirer’s and supplier’s side by:

- specifying architecturally significant requirements
- reinforcing a ‘quality-attribute-driven’ architectural design
- ensuring suitable software architecture documentation is procured
- conducting architecture evaluations to identify risks in achieving quality attributes and business goals
- requiring mitigation of identified architectural risks early in development to avoid costly rework and schedule delays
Selling the Concept with “Architecture Stories”

We have had success in getting procuring organizations buy into the concept through the use of “architecture stories.”

An architecture story is expressed in the following form:

As a <stakeholder role>, I want to <the goal> so that <reason why>

A well crafted architecture-centric story has these three components:

Who -- title denoting the acquisition stakeholder’s role
What -- the goal the acquisition stakeholder wants to achieve
Why -- the reason the stakeholder wants to achieve the goal
As a program manager, requirements manager, or architect …
I want to ensure the architecturally significant requirements are appropriately specified

so that:
• they unambiguously define the system’s non-functional requirements which will drive the design of the software architecture
• they accurately express the system’s desired qualities (e.g., security, performance, availability, modifiability, openness) in stakeholder terms
• they can later be used as the basis for conducting a software architecture evaluation

When should it be done?
Should the development contractor be involved?
As a program manager, software life cycle manager, or architect …
I want to be sure the software architecture is appropriately documented and included as a contract deliverable

so that:

• stakeholders can understand the system’s construction and behavior
• an architectural baseline can be established and subsequently placed under configuration management and control
• up-to-date architecture documentation consisting of multiple views (e.g., decomposition, run-time, allocation) will be available
• better estimates can be made of the cost and schedule of implementing a new software capability, planned upgrade, or ECP
• support can be provided to third party component developers
• software development and sustainment can be conducted more effectively

Will this happen on its own?
Architecture Evaluation Story

As a program manager, risk manager, or architect …

I want an independent team to conduct a software architecture evaluation in collaboration with the development contractor in order to:

• obtain visibility into the efficacy of the development contractor’s proposed software design
• determine if the architecture is suitable for achieving the program’s business and mission drivers and desired system qualities
• create a “win-win” situation by having an objective evaluation that also involves the development contractor
• identify architecture-related risks early in the development cycle so they can be mitigated in a timely and cost effective manner
• have a good understanding of the nature of the risks so they can be prioritized and a risk mitigation plan can subsequently be developed

Why not just leave it up to the contractor to do it?
Example Implementation of an Architecture-Centric Procurement
Example Architecture-Centric Procurement

Milestones

PHASES

Material Solution Analysis
Technology Development
Engineering and Manufacturing Development
Production and Deployment
Operations and Support

By an Independent Arch Eval Team

Initial SWARD Delivery
SWARD Readiness Review
Arch Risk Mitigation Report

PDR
Arch Demo

Contract Option

Contractually-specified architecture practices

Government Stakeholders

Government and Contractor Stakeholders

Evaluation Report

Proposed Mitigation Approaches

SW Arch Evaluation

SW Arch Readiness Review

Arch Demo – Architecture Demonstration
SRR – System Requirements Review
APW – Acquisition Planning Workshop
PDR – Preliminary Design Review
SWARD – SoftWare ARchitecture Description Document
CDR – Critical Design Review

Legend

Example Architecture-Centric Procurement

Milestones

Acquisition Planning
RFP / SOW
Contract Performance Phase
with Government Oversight

Scheduled by Program Office
N days after contract award

P
Z

X days before Arch Eval

W days before PDR

Q days before CDR

Government and Contractor Stakeholders

Initial SWARD Delivery

SRR

Arch Demo

Contract Option

Contractually-specified architecture practices

Legend

Arch Demo – Architecture Demonstration
APW – Acquisition Planning Workshop
PDR – Preliminary Design Review
SRR – System Requirements Review
SWARD – SoftWare ARchitecture Description Document
CDR – Critical Design Review
Architecture-Centric Activities and Events in a Procurement Context

- Quality Attribute Workshop
- Quality Attribute Scenarios

Software Architecture Development

SWARD

Software Architecture Readiness Review

ATAM Evaluation Report

Architecture Risks

Risk Mitigation Analysis

Risk Mitigation Report

AGB Architecture Governance Board

PDR Program Review

Program Review

Contract Deliverable

GFI Plan

Contractor Task

Collaborative Task

Program Office

Team QAW Plan

Team ARR Plan

Team ATAM Plan

Go Decision

No-Go Decision

Architecture documentation needs to be revised

Updated SWARD

Architecture risks mitigated and documentation updated

Detailed Design and CDR

Software Architecture Development

Program Office

SOW Task

Team

Contractor

Software Architecture Description

Contractor

Program Office

SOW Task

SOW Task

SOW Task

SOW Task

Architecture-Centric Procurement
© 2013 Carnegie Mellon University

Software Engineering Institute
Carnegie Mellon
Presentation Outline

Essential Architecture Practices

Incorporating Architecture Practices into the Procurement Context

Conclusion
Conclusion

Our experience has shown that a quality attribute and architecture-centric approach:

• provides early insight into critical requirements and design decisions that drive the entire development effort
• provides a proven and effective means for discovering architecture risks and risk themes
• results in fewer test and integration problems and costly rework downstream
• provides right level of abstraction that aligns with limited program office resources and time and effort
• provides a product focus that complements process-focused activities
• enables an acquisition organization to perform its contract management responsibilities with greater effectiveness.
Your Choice

An Architecture-Centric Acquisition Approach
Questions
References - 1

*Software Architecture in Practice, Third Edition*

*Evaluating Software Architectures: Methods and Case Studies*

*Documenting Software Architectures: Views and Beyond, Second Edition*

*Software Product Lines: Practices and Patterns*
References - 2

You can find a moderated list of references on the “Software Architecture Essential Bookshelf”
http://www.sei.cmu.edu/architecture/start/publications/bookshelf.cfm

Grady Booch: Handbook of Software Architecture (currently only an on-line reference):
http://www.handbookofsoftwarearchitecture.com/index.jsp?page=Main
Contact Information

John Bergey
Research, Technology, and Systems Solutions Program
Telephone: 215-348-0530
Email: jkb@sei.cmu.edu

Larry Jones
Research, Technology, and Systems Solutions Program
Telephone: 719-481-8672
Email: lgj@sei.cmu.edu

U.S. Mail:
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-3890

World Wide Web:
http://www.sei.cmu.edu/productlines
SEI Fax: 412-268-5758
Backup Slides
As a program manager, quality assurance manager, or architect …

I want to conduct an in-situ review of the architecture documentation in collaboration with the system developer

in order to:

• determine if the documentation is sufficient to enable the follow-on architecture evaluation to proceed as scheduled

  ✓ identify any significant architecture documentation deficiencies early so they can be corrected in a timely fashion

  ✓ allow enough time for the development contractor to correct the deficiencies prior to conducting the architecture evaluation

  ✓ avoid potential expense and delay that would otherwise be incurred if the planned evaluation had to be aborted

• familiarize the developer with the method that will be used to conduct the follow-on architecture evaluation

**Is this review really necessary?**

**What happens with out it?**
Architecture Risk Mitigation Story

As a program manager, risk manager, or architect …

I want the development contractor to analyze the risks identified during the architecture evaluation and propose solutions

in order to:

• mitigate risks as early as practical in the development cycle when they can be mitigated in a timely and cost effective manner

• understand the risk impact, what risk mitigation approaches were explored, and the rationale for the recommended approach

• have the development contractor present and facilitate a discussion of its risk mitigation approach during a Preliminary Design Review (PDR):
  ✓ avoids perfunctory design reviews
  ✓ provides forum for reviewing risks and understanding risk impact
  ✓ enables discussion of recommended risk mitigation approach and alternative approaches that were considered

Why do this? Isn’t the contractor required to have a risk management plan?
Architecture Governance Story

As a program manager, quality assurance manager, or architect …
I want to establish an effective means of architecture governance throughout the life cycle
so that:

• the software architecture is under configuration management (CM) and control and the architecture documentation is kept up-to-date
• there is an architecture IPT or governing board to provide architectural input and consultation to management and the CM control board to:
  ✓ ensure new requirements and planned software changes are reviewed and analyzed to determine architectural impact
  ✓ estimate cost and schedule impact of architectural changes
  ✓ participate in architecture evaluations (and conformance reviews) and ensure architectural risks are appropriately mitigated
• changes to the system architecture and software architecture are appropriately documented and coordinated

Why do we have to tell the contractor what to do?
Architecture Conformance Story

As a program manager, quality assurance manager, or architect …
I want the development contractor to conduct periodic architecture conformance reviews and report on the findings in order to:

• determine if the detailed software design and software implementation are in compliance with the baseline software architecture
  – prior to a critical design review (CDR)
  – before each major software release

• identify any significant discrepancies between the software architecture description and the “as built” software design and code implementation

• understand the impact of any discovered discrepancies or anomalies and how the contractor’s proposes to remedy them

What difference will it make?