

Architecting Software the SEI Way

Essential Steps Toward Mastery

February 28, 2012 • 1:00pm–4:30pm EST



Analyzing and Evaluating Enterprise Architectures

John Klein

Senior Technical Staff

John has over 20 years experience developing systems and software. He joined SEI in 2008. Before joining SEI, John was a chief architect at Avaya, Inc. There his responsibilities included development of multimodal agents, architectures for communication analytics, and the creation and enhancement of the Customer Interaction Software Product Line architecture. Prior to that, John was a software architect at Quintus, where he designed the first commercially successful multi-channel integrated contact center product and led the technology integration of the product portfolio as Quintus acquired several other companies.

See his full bio at:

www.sei.cmu.edu/go/architecting-software-the-sei-way



Software Engineering Institute

CarnegieMellon

Architecting Software the SEI Way

Twitter **#SEIArchitecture**

© 2012 Carnegie Mellon University

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 28 FEB 2012		2. REPORT TYPE		3. DATES COVERED 00-00-2012 to 00-00-2012	
4. TITLE AND SUBTITLE Analyzing and Evaluating Enterprise Architectures				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Carnegie Mellon University,Software Engineering Institute,Pittsburgh,PA,15213				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 29	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Outline

Essential problems for architects -

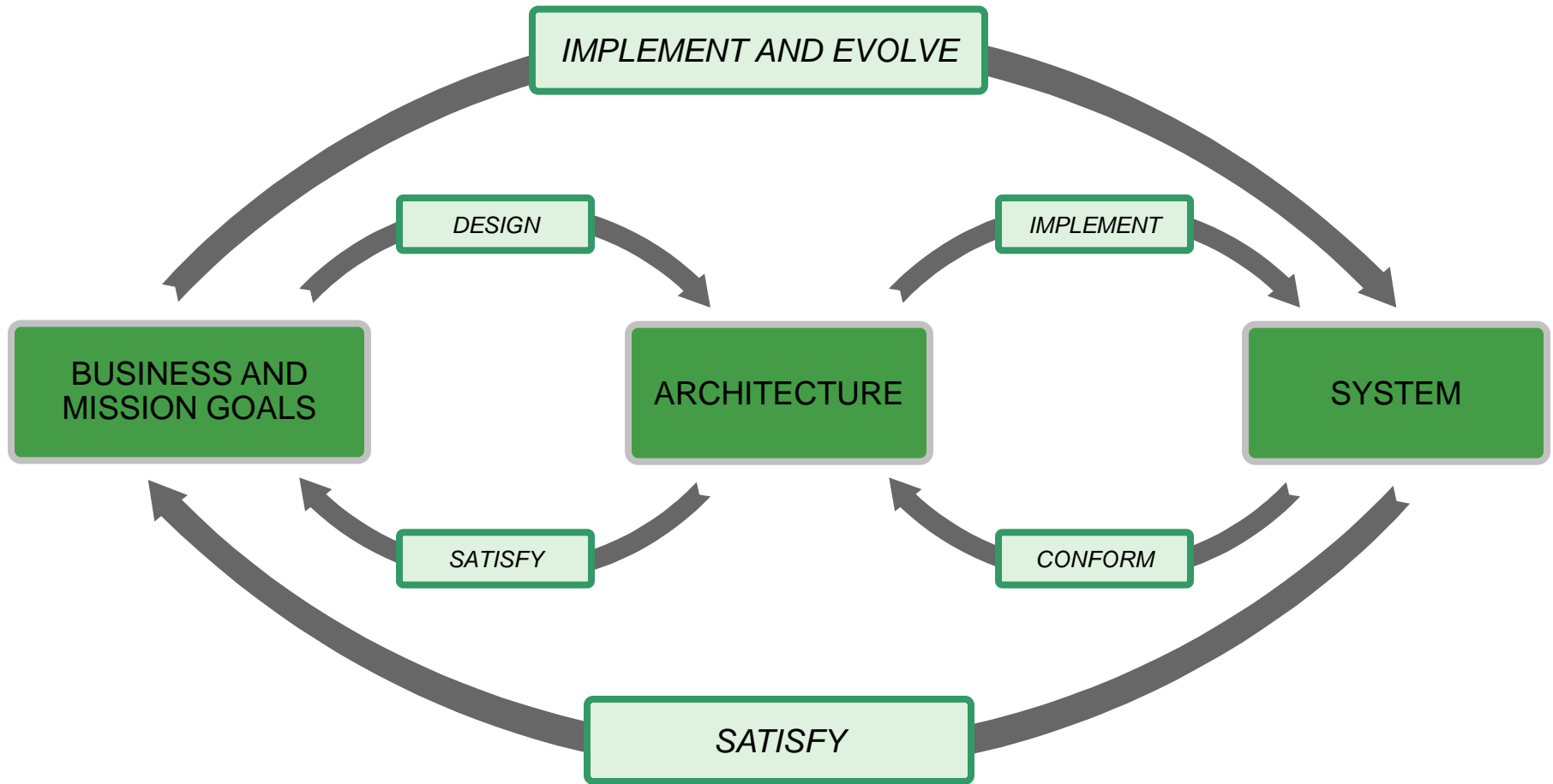
- How do we efficiently translate business goals into quality attribute requirements?
- How do we ensure that these quality attribute requirements are reflected in the tradeoffs and decisions that shaped the architecture?

Agenda

- Review of the SEI perspective on architecture-centric engineering
- Scaling from software context to systems of systems and enterprise architectures
- An approach to developing quality attribute requirements for enterprise architectures
- An approach to first-pass evaluation of enterprise architectures
- Tying together EA and system/software analysis and evaluation



Architecture-centric Engineering – Software and Systems



Principles of Architecture-Centric Engineering

Every system has an architecture, regardless of scale.

Architecture is the appropriate abstraction for reasoning about business or mission goal satisfaction.

Quality attributes have a dominant influence on a system's architecture.

Value derived from business and mission goals governs quality attribute tradeoffs.

Well-founded cost-effective measurements and analyses are the bases for acquiring confidence about system properties.

Architectural prescriptions must be demonstrably satisfied by the implementation.

Today's architectural decisions must appropriately reflect the drivers of system change.



Quality Attributes Shape Architectures

Quality attributes are properties of work products or goods by which stakeholders judge their quality.

Some examples of quality attributes by which stakeholders judge the quality of software systems are

- performance
- security
- modifiability
- reliability
- usability
- calibrateability
- availability
- social-ability
- throughput
- configurability
- subsetability
- reusability



What about Functional Requirements?

Functional requirements define the work that the system is intended to do

- Functionality often has associated quality attribute requirements (e.g., a function is required to have a certain level of availability, reliability, and performance).
- We can achieve functional requirements and yet fail to meet their associated quality attribute requirements.
- Functionality can be achieved using many different architectures.
- Achieving quality attribute requirements can only be achieved through judicious choice of architectures.



Quality Attribute Scenarios for Systems and Software

“When you say the system should have ‘good performance’, what do you mean? Could you give me an example?”

One way to describe quality attribute concerns is to use quality attribute scenarios to clearly characterize them.

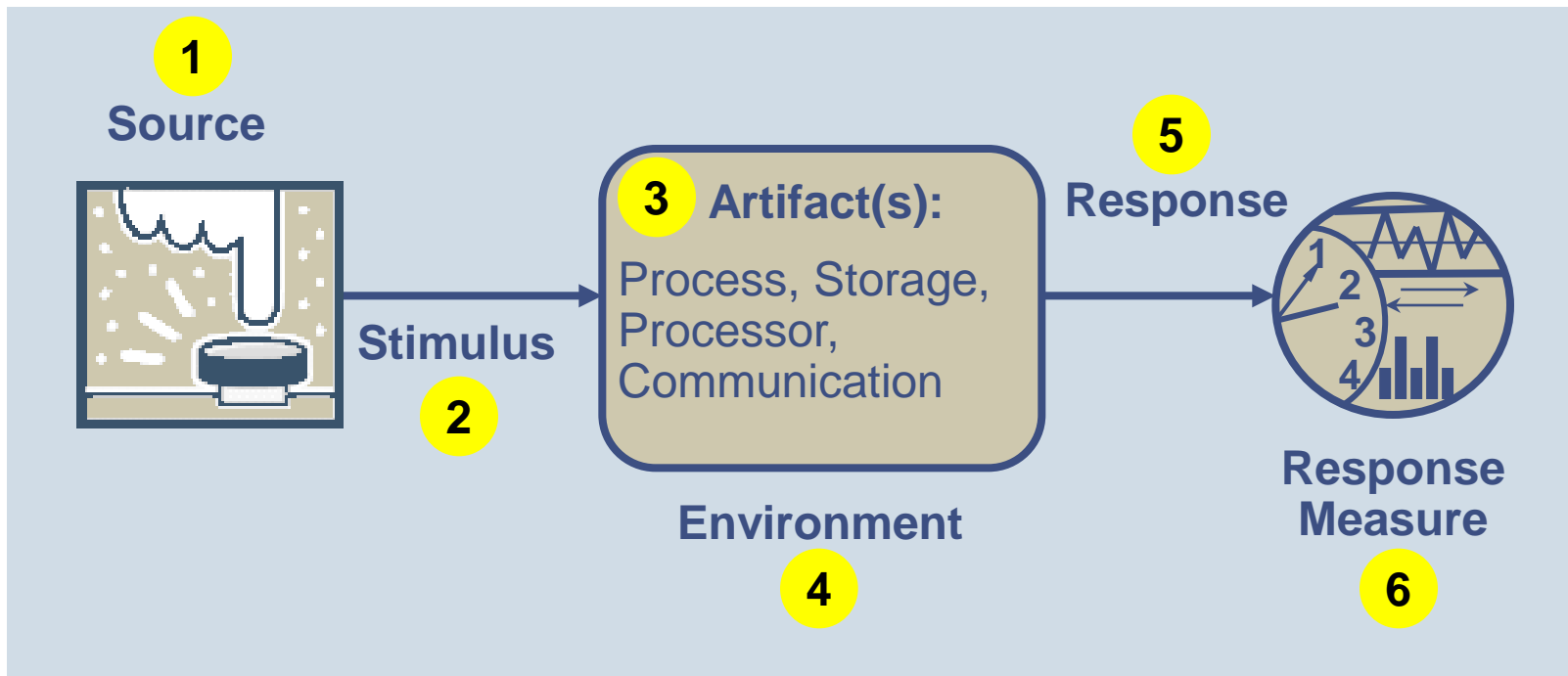
A quality attribute scenario is a short description of how a system is required to respond to some stimulus.

Quality attribute scenarios complement use cases and user stories

- Use cases and user stories focus on functionality – what must the system do
- Scenarios focus on how the system must deliver the functionality
- Stimulus for a scenario can come from the use case or user story



Quality Attribute Scenarios for Systems and Software



System of Systems

What distinguishes a a system of systems from other complex assemblages of large-scale components?

- The constituent components are systems *can* operate independently to fulfill customer or operator purposes on their own (“operational independence”)
- The component systems actually *do* operate independently (“managerial independence”)

We see this frequently in an enterprise IT context – systems serve multiple purposes with different stakeholders and drivers

Maier, M. “Architecting principles for systems-of-systems”,
Systems Engineering, **1998**, Vol. 1(4), pp. 267-284



Taking an SoS perspective provides insights

Maier identifies four design principles that can help us succeed in this challenging environment

- Stable Intermediate Forms – “Complex systems will develop and evolve within an overall architecture much more rapidly if there are stable intermediate forms than if there are not.”
- Policy Triage – “The triage: Let the dying die. Ignore those who will recover on their own. And treat only those who would die without help.”
- Leverage at the interfaces – “The greatest leverage in system architecting is at the interfaces. The greatest dangers are also at the interfaces.”
- Ensuring Cooperation – “If a system requires voluntary collaboration, the mechanism and incentives for that collaboration must be designed in.”

Maier, M. “Architecting principles for systems-of-systems”, *Systems Engineering*, **1998**, Vol. 1(4), pp. 267-284



Enterprise Architecture (EA)

“Enterprise Architecture is the organizing logic for business processes and IT infrastructure reflecting the integration and standardization requirements of the firm’s operating model...The IT unit typically addresses four levels of architecture below the enterprise architecture:

- business process architecture...
- data or information architecture...
- applications architecture...and
- technology architecture...

The term ***enterprise architecture*** can be confusing because the IT unit in some companies refers to one of these architectures—or the set of all four architectures—as the enterprise architecture.”

Ross, J. W., Weill, P., & Robertson, D. C., *Enterprise Architecture as Strategy*, Harvard Business School Press, 2006. <http://www.architectureasstrategy.com/book/eas/about.htm>

See Klein and Gagliardi, *A Workshop on Analysis and Evaluation of Enterprise Architectures*, Technical Report CMU/SEI-2010-023, November 2010 for other definitions.



An enterprise architecture includes a system of systems architecture

Parts of an enterprise architecture

- Information Architecture – Repositories and schemas
- Applications Architecture – Automate business process activities, use Information Architecture Repositories, execute on Technology Architecture platforms
- Technology Architecture – Platforms and networks
- Business Process Architecture – Defines the functions, goals, and constraints for the other three architectures.

Many-to-many relationships between elements of each architecture creates a system of systems context

We find this a useful perspective to take for analysis and evaluation.

Klein and Gagliardi, *A Workshop on Analysis and Evaluation of Enterprise Architectures*, Technical Report CMU/SEI-2010-023, November 2010.



Taking an SoS perspective on enterprise architectures

(Almost) never a greenfield project – need to deal with “as-is”, “to-be”, and the path from here to there

- Remember the principle of “stable intermediate states”

Socio-technical system

- Humans in the loop
 - Indeterminate response characteristics
 - Different failure and recovery modes
- Allocation tradeoffs – human or machine?
- Runtime and development time systems (Mirroring/Conway’s Law)

Complex environments

- Many and diverse stakeholder groups
- A single constituent system can play many roles
- Direct and indirect relationships among elements

EA can be as much about policy as it is about structure

- Remember “Triage” and “Ensuring Cooperation”



An Analysis Tool for Enterprise Architectures – Business Threads

A **Business Thread** describes an end-to-end flow through the enterprise, usually encompassing multiple business processes

Examples:

- “Quote to Cash”
- Shop to Order to Fulfillment to Return
- Symptom/Problem to Primary Care to Hospital Admission to Discharge to Rehab

Inspirations for the concept include Mission Threads (DoD) and “Green Threads” (IBM Rational)

A business thread typically involves

- Multiple actors
- Multiple applications
- Data in multiple repositories
- Often cross organizational boundaries, may include elements “external” to the enterprise



Categories of Business Threads

Core Business Threads

- Trace through the core business processes of the enterprise
- Examples on previous slide are “core business” threads
- Splice together “traditional” business processes

Operations Threads

- Trace through support operations processes
- Examples include deployment, migration, “go-live”, day-to-day management, training, and disaster recovery
- Look for “business processes” in ITIL or Data Center Operations manuals

Development Threads

- Trace through development, test, and integration
- Use development plans and operating procedures as a starting point



Business Thread Types

“As-Is” – these threads reflect as-is capabilities that must be maintained as the EA changes, for example during integration of an acquired company.

“To-Be” – these threads reflect well-defined future capabilities that must be supported in a new or evolved architecture.

“What-If” – these threads are analogous to “Growth Scenarios” in the QAW, exploring opportunities and testing the limits of the EA.



Polling Question #1

We deal with quality attribute requirements...

- a) Systematically - we collect quality attribute requirements for all areas
- b) Partially - we look at some qualities like performance and security, for some capabilities
- c) Barely - we focus mostly on functional requirements



Business Thread Workshop (BTW) – Developing EA Quality Attribute Requirements

The BTW *augments EA business threads* with quality attribute considerations that shape the EA architecture and *identifies EA architectural challenges*, as early in the EA development cycle as possible

The business thread augmentation is performed by key EA stakeholders using a structured and facilitated process (extension of the QAW)

The augmented threads and discovered challenges are used to develop the enterprise architecture, and then reused to evaluate the EA

There will usually be a series of BTWs, depending on scope, scale, and schedule considerations



BTW Inputs

EA Business Drivers Presentation

- Includes quality attributes

EA Architecture Plans Presentation

Business Contexts and Business Threads



Quality Attribute Augmentation Process

For each business thread:

- Elicit any over-arching quality attribute considerations, e.g. security, interoperability, etc.
- Capture any over-arching assumptions, engineering issues, challenges, etc.
- For each step in the business thread:
 - For each quality attribute, elicit quality attribute considerations. Record any engineering issues, assumptions, challenges, etc.
 - Capture any capability or other issues that arise

Stakeholder inputs are key



BTW Outputs

Augmented Business Threads

- Over-arching quality attribute augmentations for the business thread
- Capability and business goal augmentations for the business thread
- Quality attribute augmentations for each event in the business thread
- Identified additional use cases (with context) and business threads

Challenges

- Architectural, capability, and business challenges derived from the business thread augmentations.
- Any candidate legacy system architecture that may require architecture evaluation.



Polling Question #2

We evaluate our architecture...

- a) Systematically - we do it for every project
- b) Partially - we evaluate selected projects
(based on cost, perceived risk, ...)
- c) Barely - we don't really evaluate our architecture
before construction



Perspectives for EA Evaluation

Enterprise architecture is a process

- Evaluate the quality of the process and adherence to the process with methods like CMMI®

The enterprise architecture process is carried out by individuals and teams working within an organization

- Evaluate the competence of the people, teams, and organization using a method like the SEI Architecture Competence Assessment

Business processes are a first-order element in enterprise architecture.

- Use an Organizational Coordination Theory perspective to evaluate alignment between the business processes and organizational structures

“Architecture is Architecture is Architecture” (John Zachman)

- Extend ATAM method to Enterprise Architectures



Enterprise Architecture Evaluation

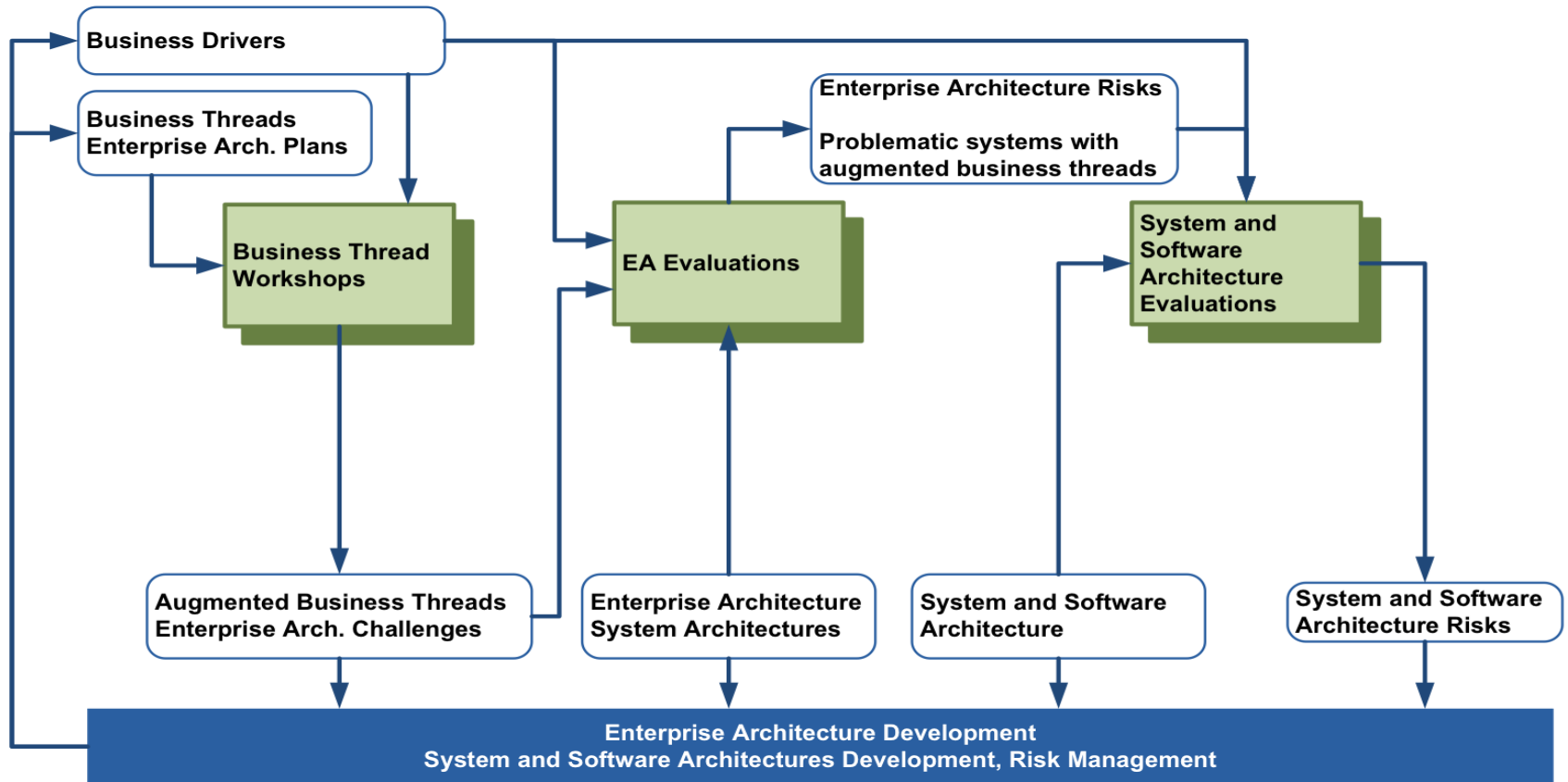
Treat EA as a System of Systems – perform first pass identification of architectural risks and quality attribute inconsistencies across the constituent systems.

Similar approach to ATAM, applied at the EA level

- Instead of scenarios, use augmented mission threads from the BTW
- Pre-select the augmented mission threads for evaluations
- Schedule a series of EA architecture evaluations (1-2 days each)
- Assemble a trained EA evaluation team
- Gather EA and system architecture documentation
- Bring together stakeholders (including EA, system and software architects)
- Architects walk through the architecture, one augmented mission thread at a time, with evaluation team probing for risks, non-risks, etc.
- Risks are rolled up into risk themes
- Identify problematic areas for more focused system/software architecture evaluations, if necessary



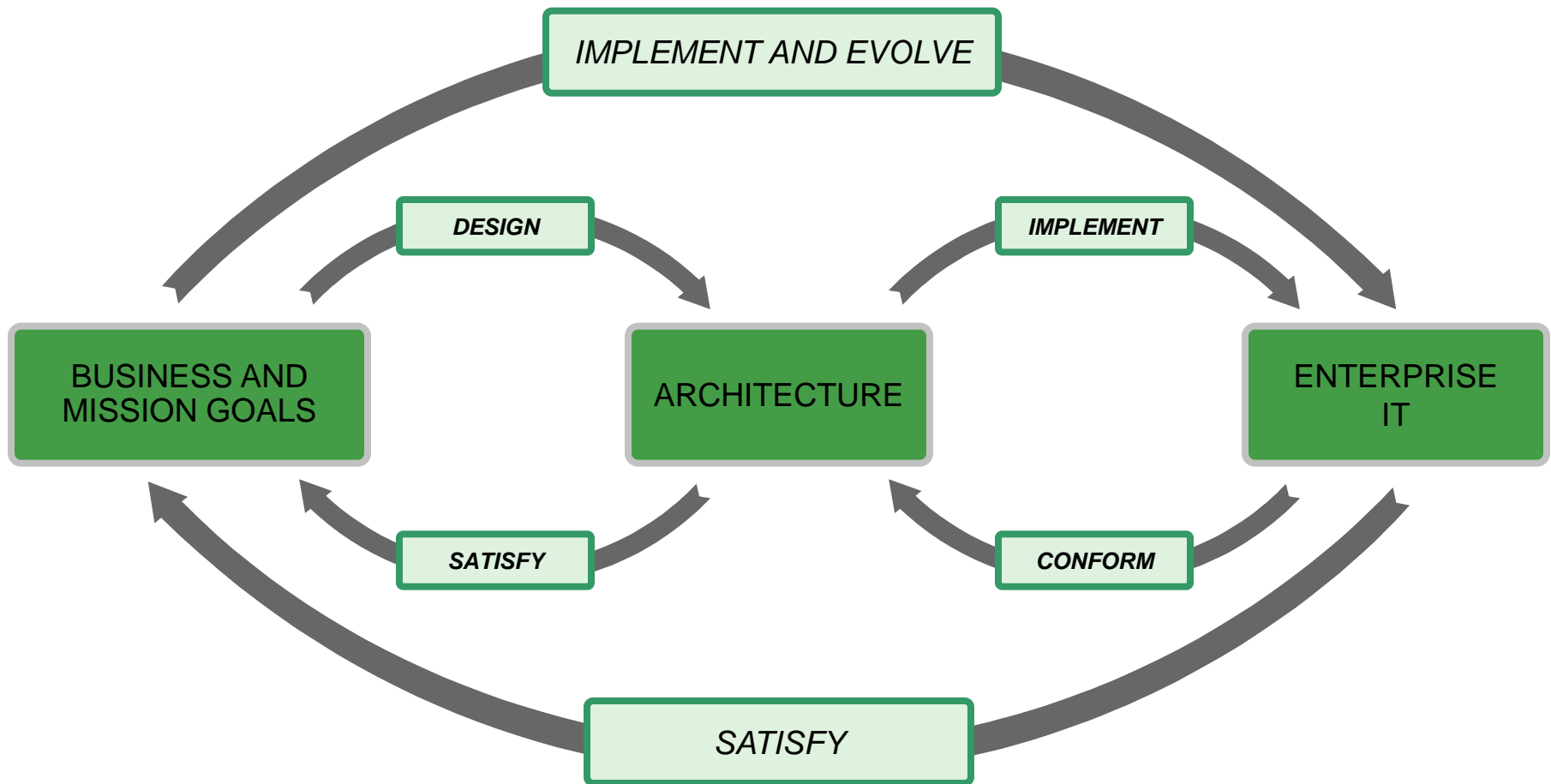
Enterprise Architecture – Uniform Approach



Gagliardi, M., Wood, W. G., Klein, J., & Morley, J. (2009, March/April). A Uniform Approach for System of Systems Architecture Evaluation. *CrossTalk*, 12-15.



Architecture-centric Engineering – Enterprise IT Architecture



Acknowledgements

The work presented here was developed by the SEI Architecture-Centric Engineering Initiative, part of the Research, Technology, and System Solutions (RTSS) Program.

<http://www.sei.cmu.edu/architecture/people/>

The Business Thread Workshop and Enterprise Architecture Evaluation builds on the work of **Mike Gagliardi** and **Bill Wood**

SEI is a Federally-funded R&D Center, sponsored by the US Department of Defense.



NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.



Q&A

SATURN 2012



Software Engineering Institute

Carnegie Mellon

As projects continue to grow in scale and complexity, effective collaboration across geographical, cultural, and technical boundaries is increasingly prevalent and essential to system success. SATURN 2012 will explore the theme of “Architecture: Catalyst for Collaboration.”

St. Petersburg, Florida May 7-11
SATURN Conference 2012



Software Engineering Institute

Carnegie Mellon

Architecting Software the SEI Way

Twitter [#SEIArchitecture](#)

© 2012 Carnegie Mellon University