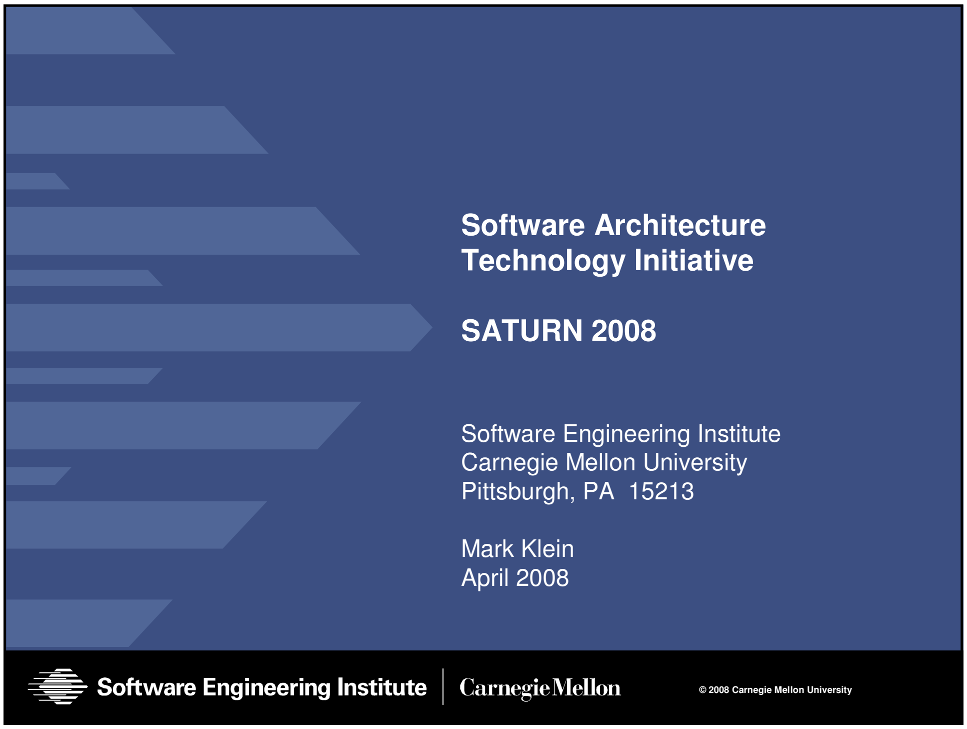
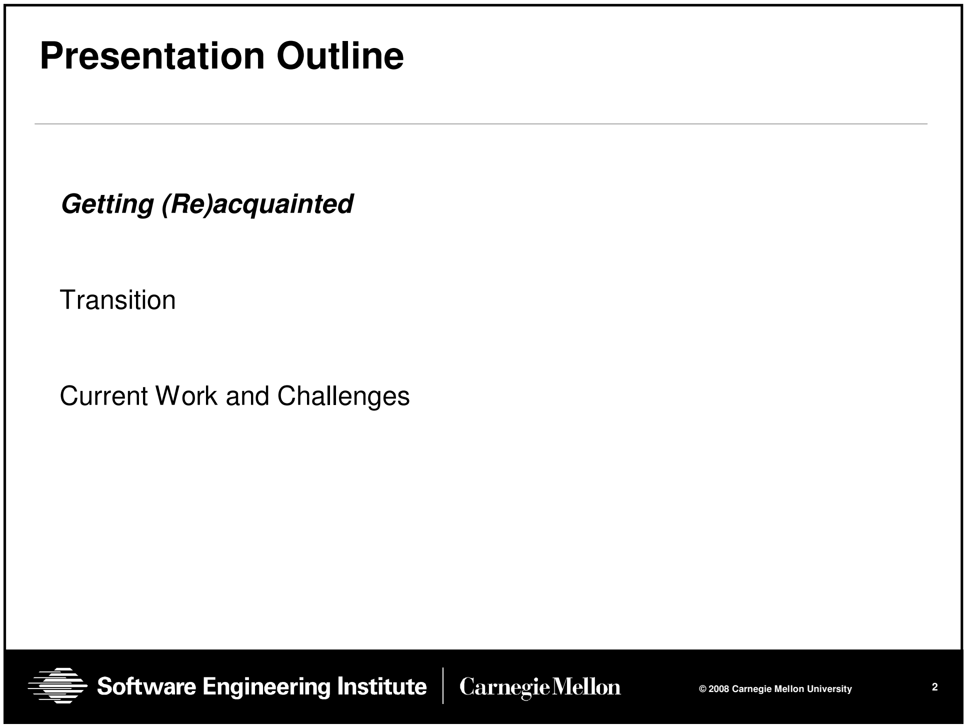


Software Architecture Technology Initiative

SATURN 2008

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Mark Klein
April 2008

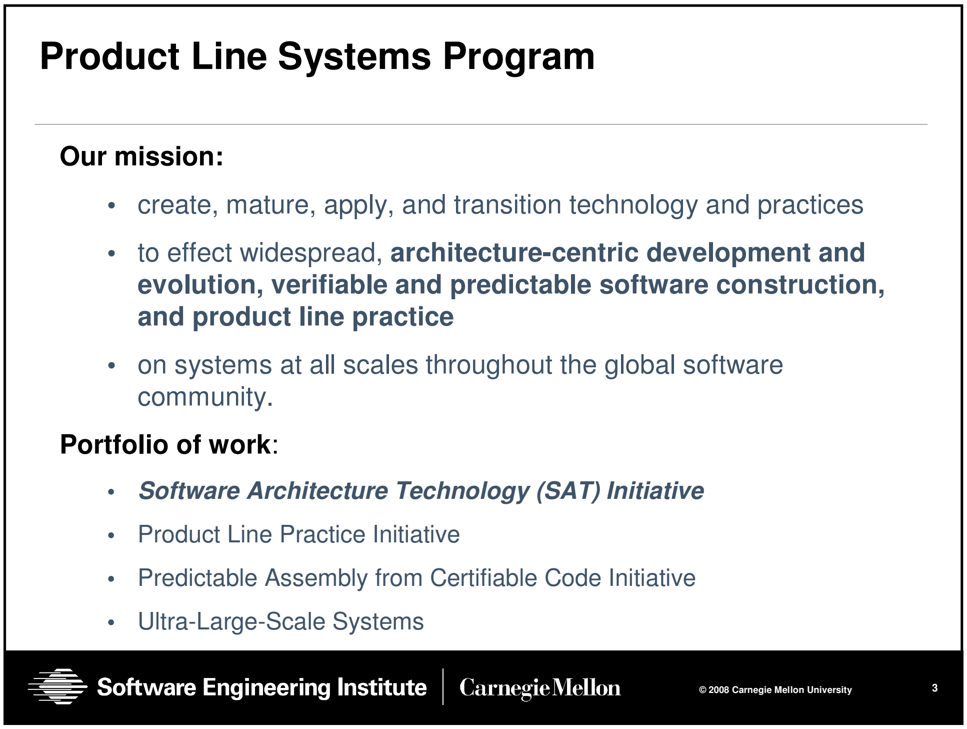


Presentation Outline

Getting (Re)acquainted

Transition

Current Work and Challenges



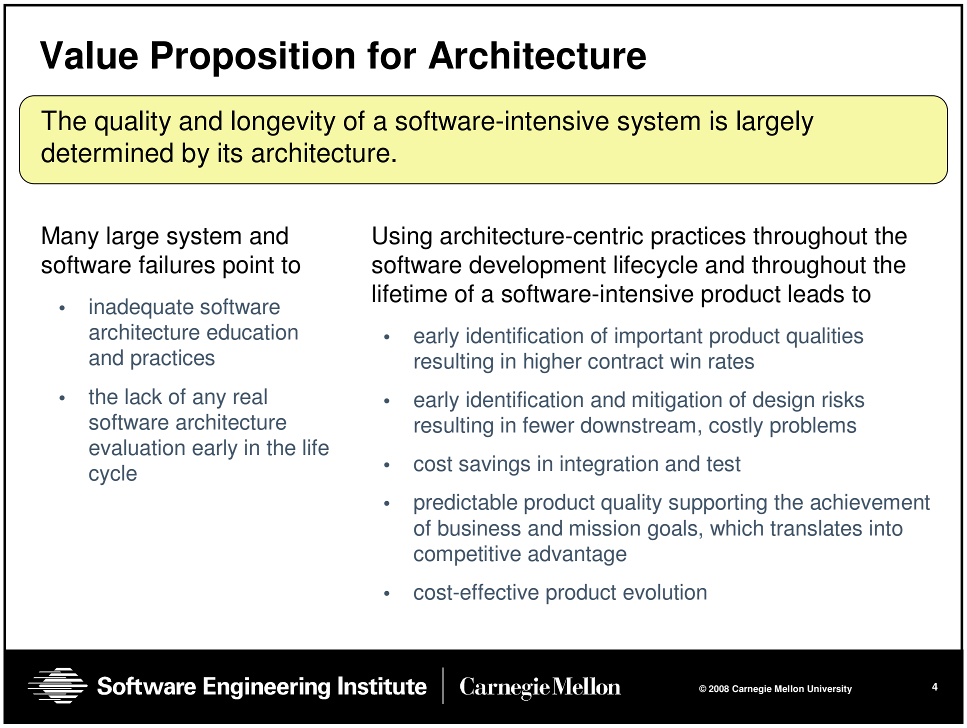
Product Line Systems Program

Our mission:

- create, mature, apply, and transition technology and practices
- to effect widespread, **architecture-centric development and evolution, verifiable and predictable software construction, and product line practice**
- on systems at all scales throughout the global software community.

Portfolio of work:

- ***Software Architecture Technology (SAT) Initiative***
- Product Line Practice Initiative
- Predictable Assembly from Certifiable Code Initiative
- Ultra-Large-Scale Systems



Value Proposition for Architecture

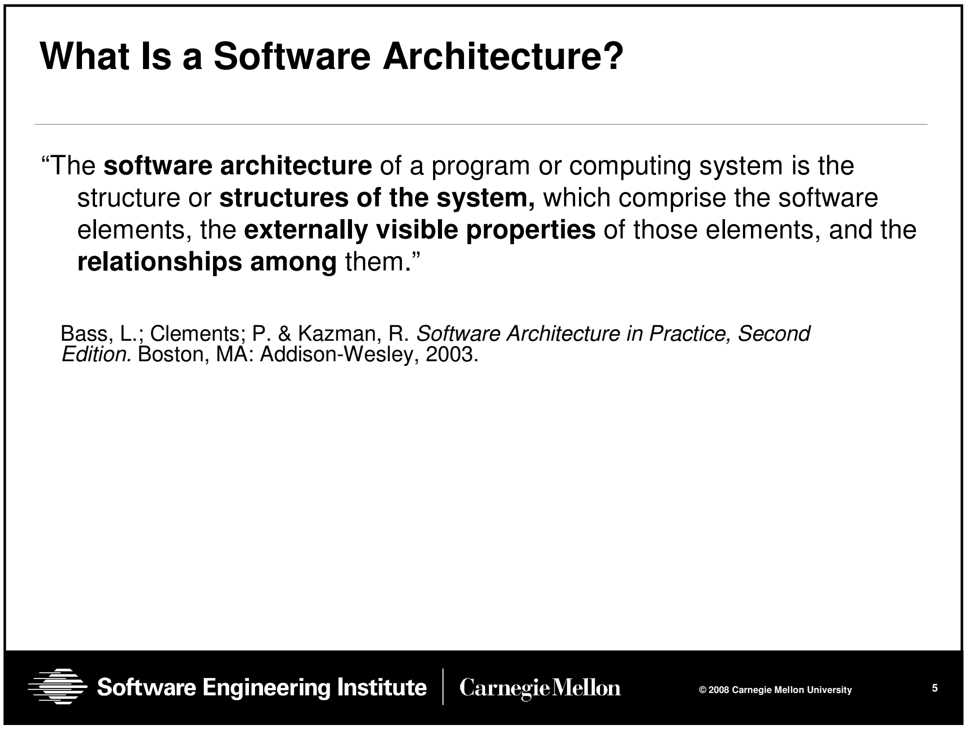
The quality and longevity of a software-intensive system is largely determined by its architecture.

Many large system and software failures point to

- inadequate software architecture education and practices
- the lack of any real software architecture evaluation early in the life cycle

Using architecture-centric practices throughout the software development lifecycle and throughout the lifetime of a software-intensive product leads to

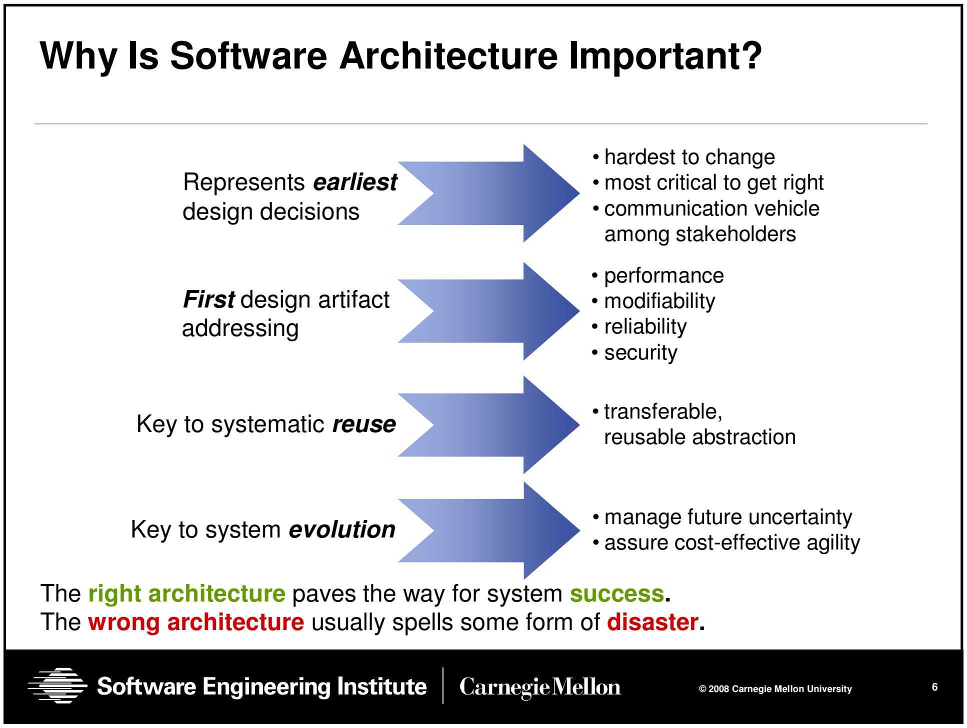
- early identification of important product qualities resulting in higher contract win rates
- early identification and mitigation of design risks resulting in fewer downstream, costly problems
- cost savings in integration and test
- predictable product quality supporting the achievement of business and mission goals, which translates into competitive advantage
- cost-effective product evolution



What Is a Software Architecture?

"The **software architecture** of a program or computing system is the structure or **structures of the system,** which comprise the software elements, the **externally visible properties** of those elements, and the **relationships among** them."

Bass, L.; Clements; P. & Kazman, R. *Software Architecture in Practice, Second Edition.* Boston, MA: Addison-Wesley, 2003.



Why Is Software Architecture Important?

Represents ***earliest*** design decisions

- hardest to change
- most critical to get right
- communication vehicle among stakeholders

First design artifact addressing

- performance
- modifiability
- reliability
- security

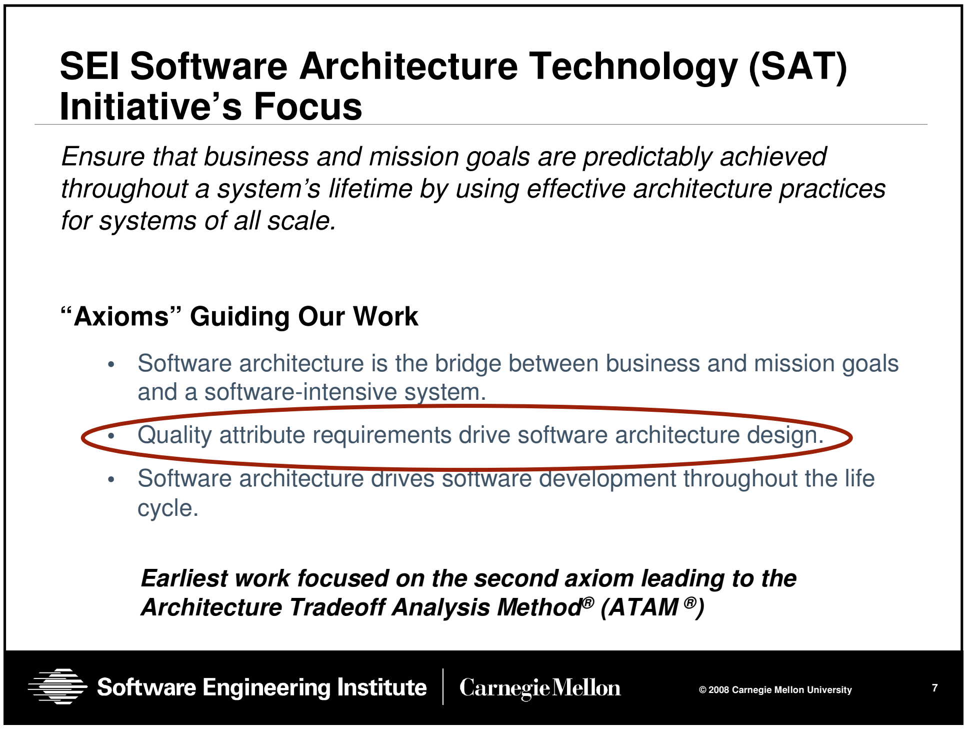
Key to systematic ***reuse***

- transferable, reusable abstraction

Key to system ***evolution***

- manage future uncertainty
- assure cost-effective agility

The **right architecture** paves the way for system **success.**
The **wrong architecture** usually spells some form of **disaster.**



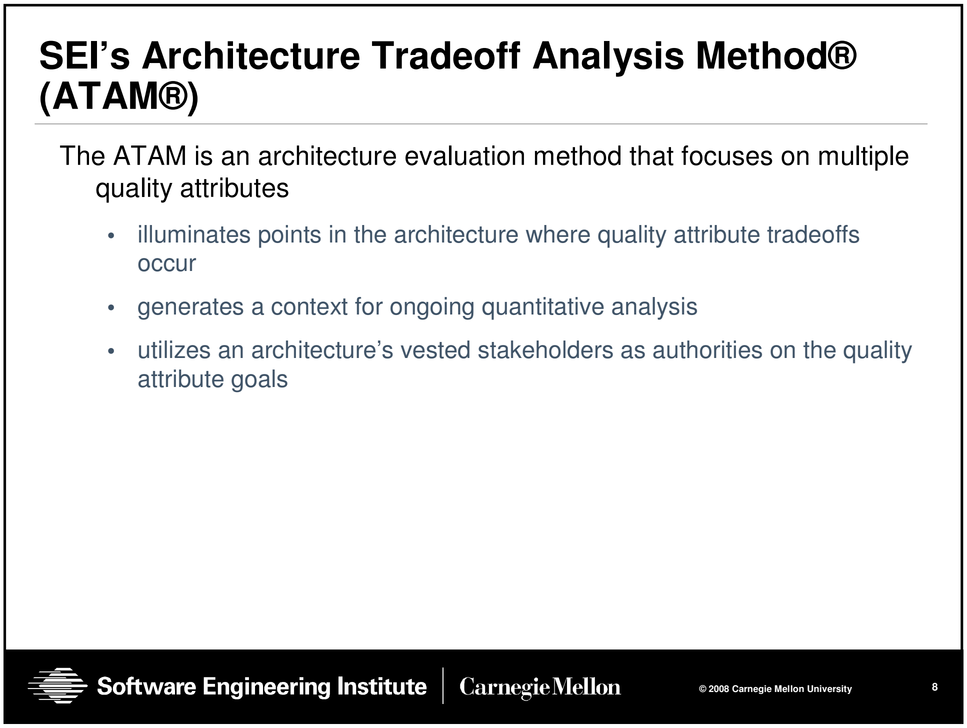
SEI Software Architecture Technology (SAT) Initiative's Focus

Ensure that business and mission goals are predictably achieved throughout a system's lifetime by using effective architecture practices for systems of all scale.

"Axioms" Guiding Our Work

- Software architecture is the bridge between business and mission goals and a software-intensive system.
- Quality attribute requirements drive software architecture design.
- Software architecture drives software development throughout the life cycle.

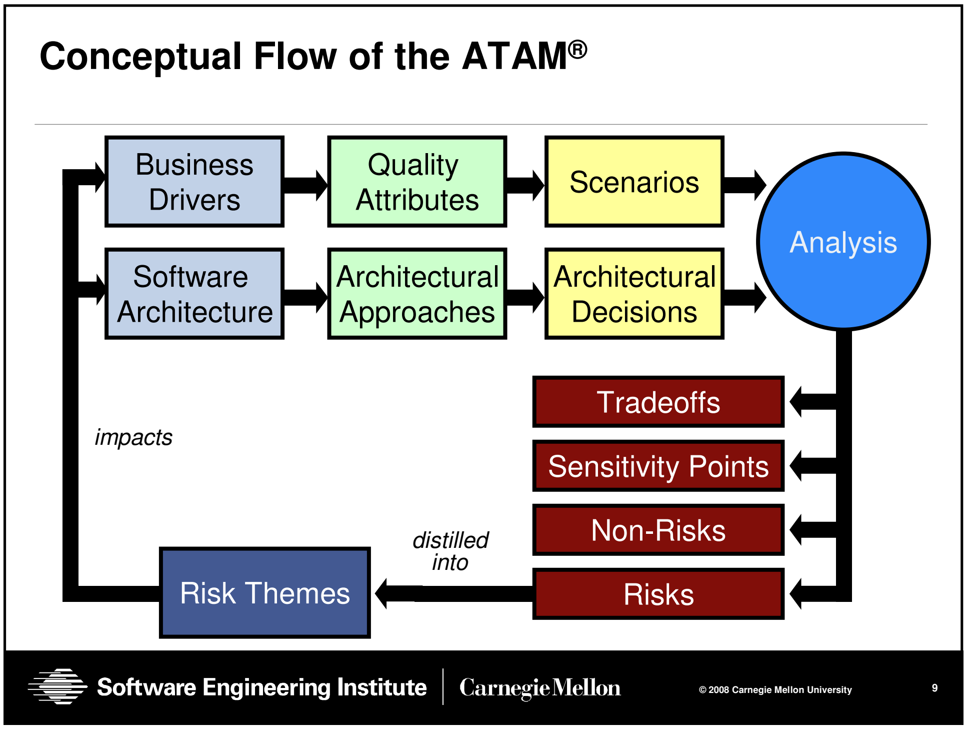
Earliest work focused on the second axiom leading to the Architecture Tradeoff Analysis Method® (ATAM ®)



SEI's Architecture Tradeoff Analysis Method® (ATAM®)

The ATAM is an architecture evaluation method that focuses on multiple quality attributes

- illuminates points in the architecture where quality attribute tradeoffs occur
- generates a context for ongoing quantitative analysis
- utilizes an architecture's vested stakeholders as authorities on the quality attribute goals



Conceptual Flow of the ATAM®

Business Drivers

Quality Attributes

Scenarios

Software Architecture

Architectural Approaches

Architectural Decisions

Analysis

Tradeoffs

Sensitivity Points

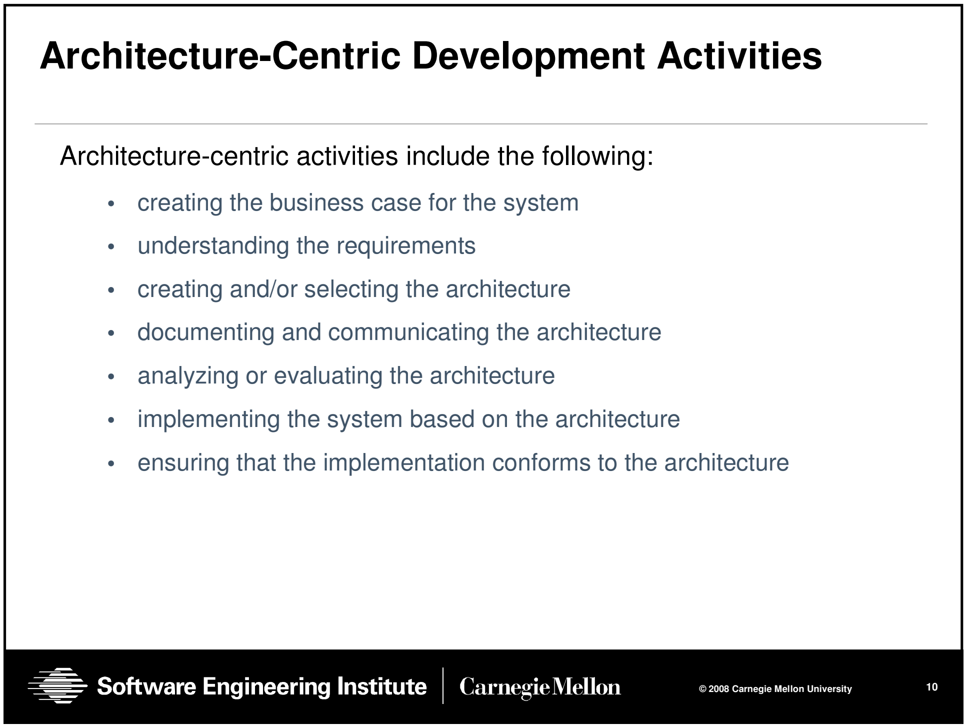
Non-Risks

Risks

distilled into

Risk Themes

impacts



Architecture-Centric Development 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

ATAM® Led to the Development of Other Methods and Techniques

What if the quality requirements are not well-understood?

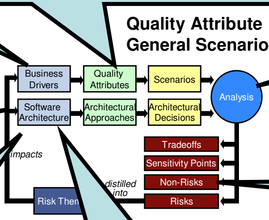
Quality Attribute Workshop (QAW)

Our scenarios tend to be incomplete or ambiguous.

Quality Attribute General Scenarios

What are some of the most important questions to ask?

Quality Attribute Tactics



What if there's no architecture?

Attribute Driven Design (ADD)

Which risks should I work on first?

Cost Benefit Analysis Method (CBAM)

What if I don't know my system's architecture?

Architecture Reconstruction using ARMIN

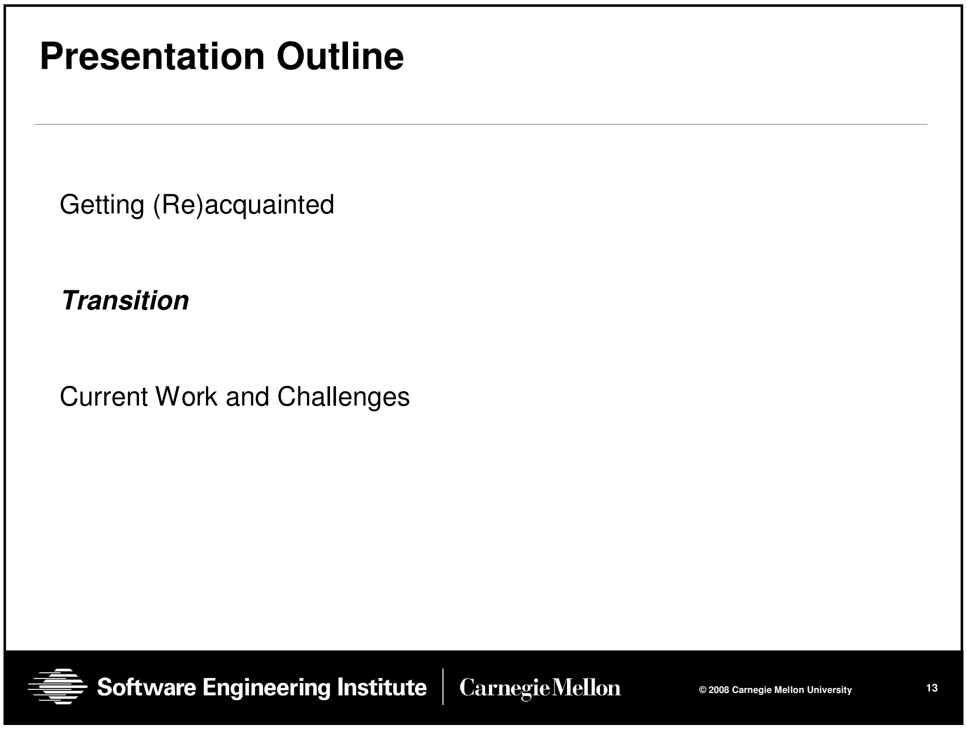
What information should be included in my architecture documentation?

Views and Beyond Approach (VaB)

Characteristics of SEI Methods

QAW
ADD
Views and Beyond
ATAM
CBAM
ARMIN

- are explicitly focused on quality attributes
- directly link to business and mission goals
- explicitly involve system stakeholders
- are grounded in state-of-the-art quality attribute models and reasoning frameworks
- are documented for practitioner consumption
- are applicable to DoD challenges and DoD systems

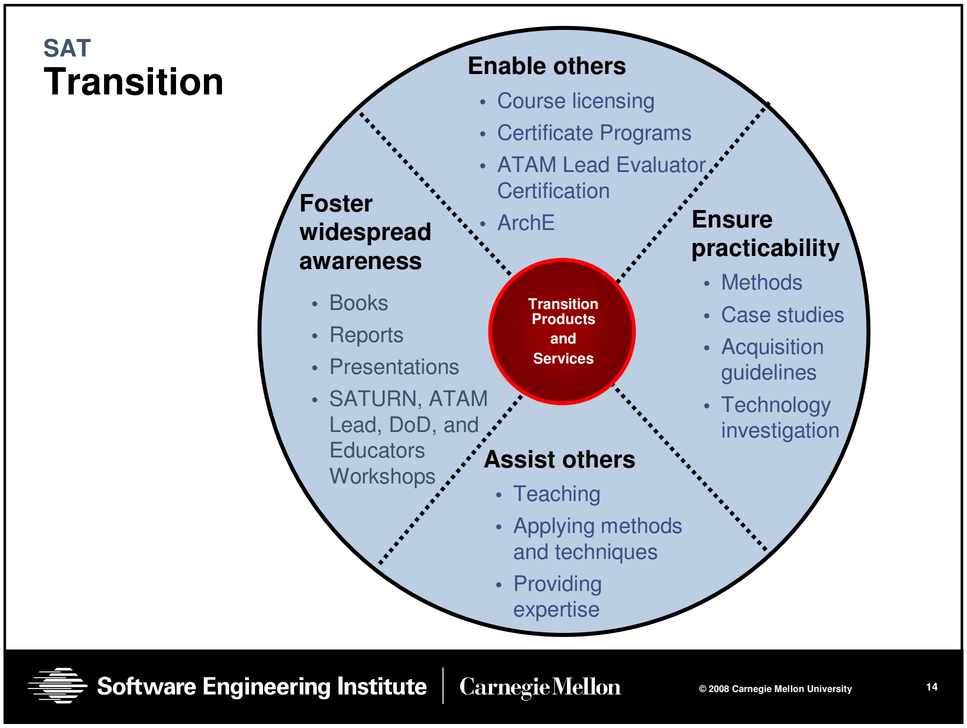


Presentation Outline

Getting (Re)acquainted

Transition

Current Work and Challenges



SAT

Transition

Enable others

- Course licensing
- Certificate Programs
- ATAM Lead Evaluator Certification
- ArchE

Foster widespread awareness

- Books
- Reports
- Presentations
- SATURN, ATAM Lead, DoD, and Educators Workshops

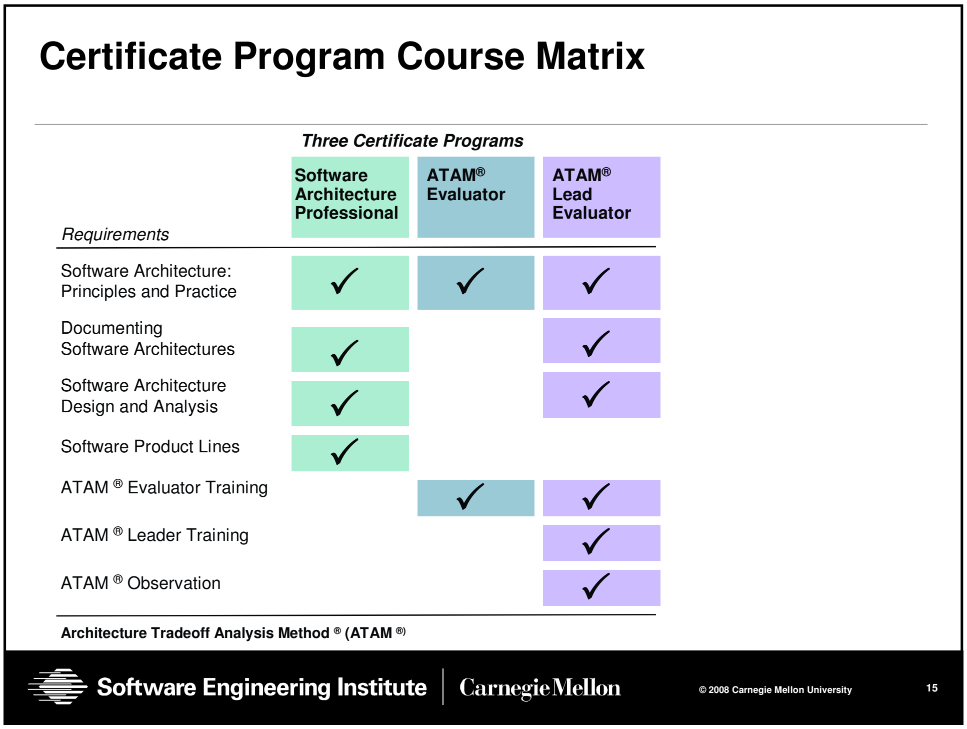
Transition Products and Services

Ensure practicability

- Methods
- Case studies
- Acquisition guidelines
- Technology investigation

Assist others

- Teaching
- Applying methods and techniques
- Providing expertise

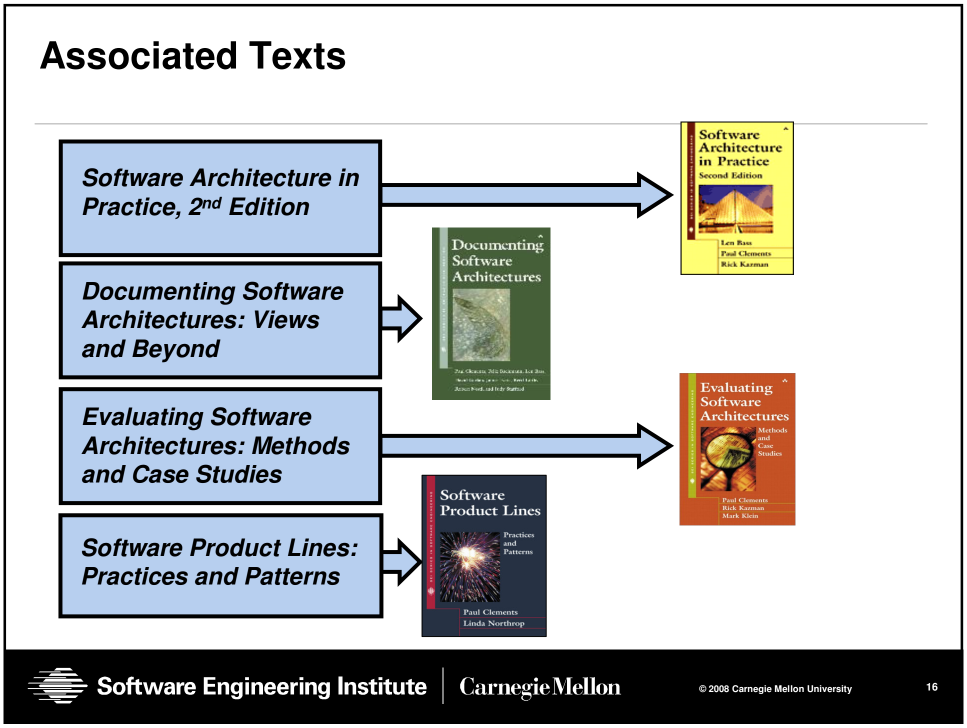


Certificate Program Course Matrix

Three Certificate Programs

Requirements	Software Architecture Professional	ATAM® Evaluator	ATAM® Lead Evaluator
Software Architecture: Principles and Practice	✓	✓	✓
Documenting Software Architectures	✓		✓
Software Architecture Design and Analysis	✓		✓
Software Product Lines	✓		
ATAM ® Evaluator Training		✓	✓
ATAM ® Leader Training			✓
ATAM ® Observation			✓

Architecture Tradeoff Analysis Method ® (ATAM ®)



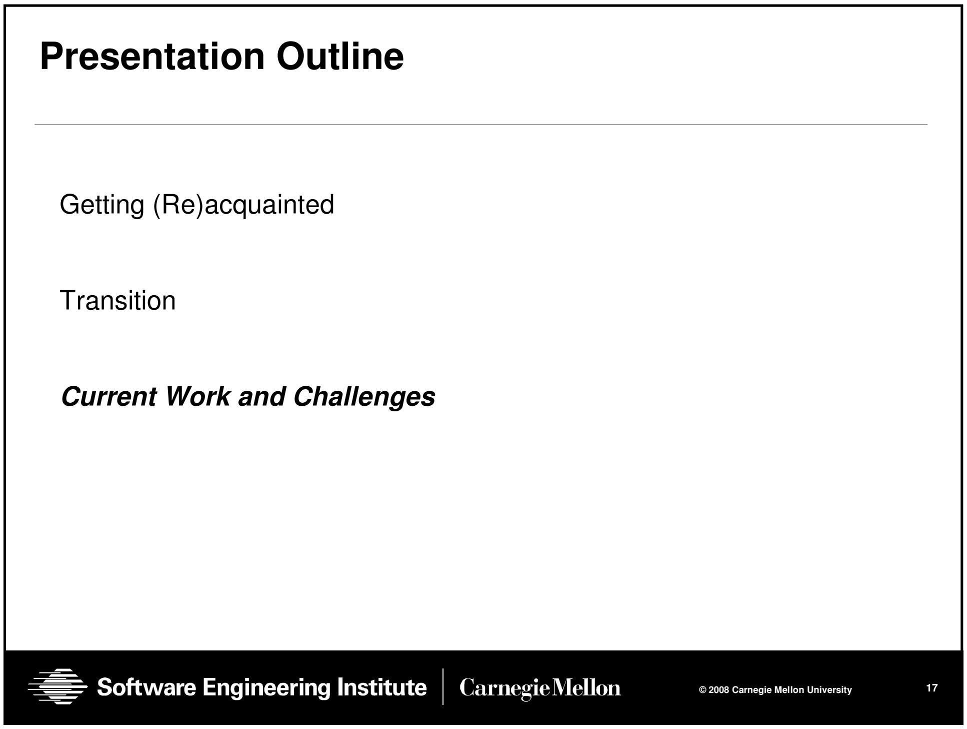
Associated Texts

Software Architecture in Practice, 2nd Edition

Documenting Software Architectures: Views and Beyond

Evaluating Software Architectures: Methods and Case Studies

Software Product Lines: Practices and Patterns

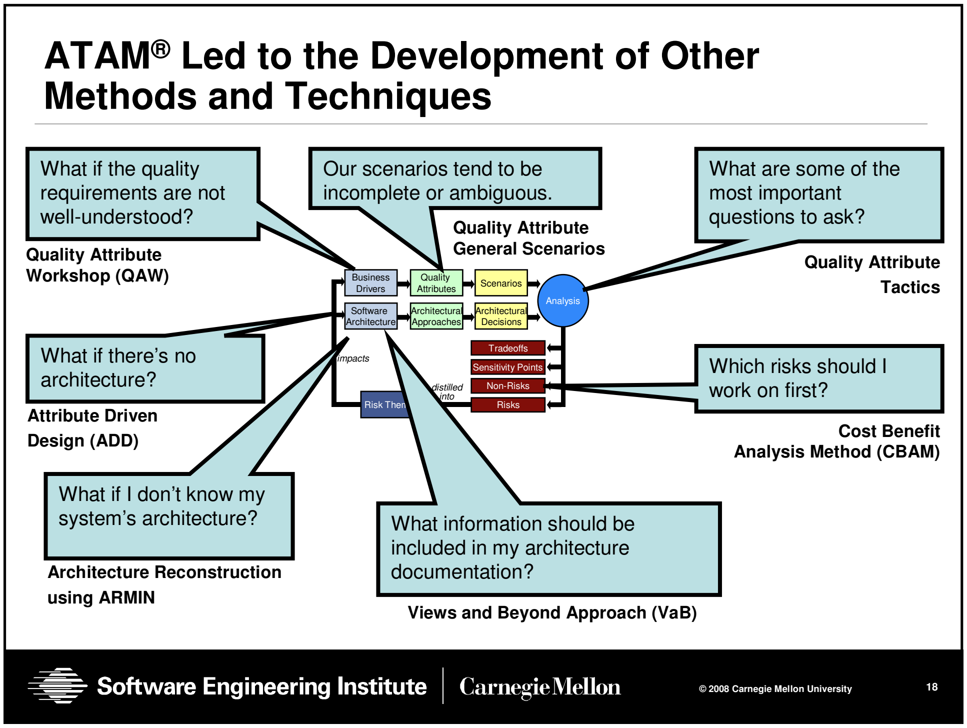


Presentation Outline

Getting (Re)acquainted

Transition

Current Work and Challenges



ATAM® Led to the Development of Other Methods and Techniques

What if the quality requirements are not well-understood?

Quality Attribute Workshop (QAW)

Our scenarios tend to be incomplete or ambiguous.

Quality Attribute General Scenarios

What are some of the most important questions to ask?

Quality Attribute Tactics

What if there's no architecture?

Attribute Driven Design (ADD)

Which risks should I work on first?

Cost Benefit Analysis Method (CBAM)

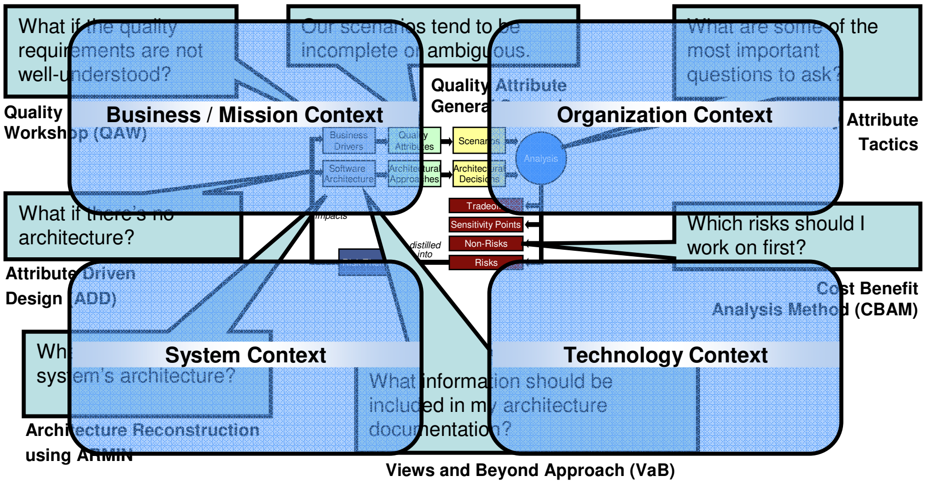
What if I don't know my system's architecture?

Architecture Reconstruction using ARMIN

What information should be included in my architecture documentation?

Views and Beyond Approach (VaB)

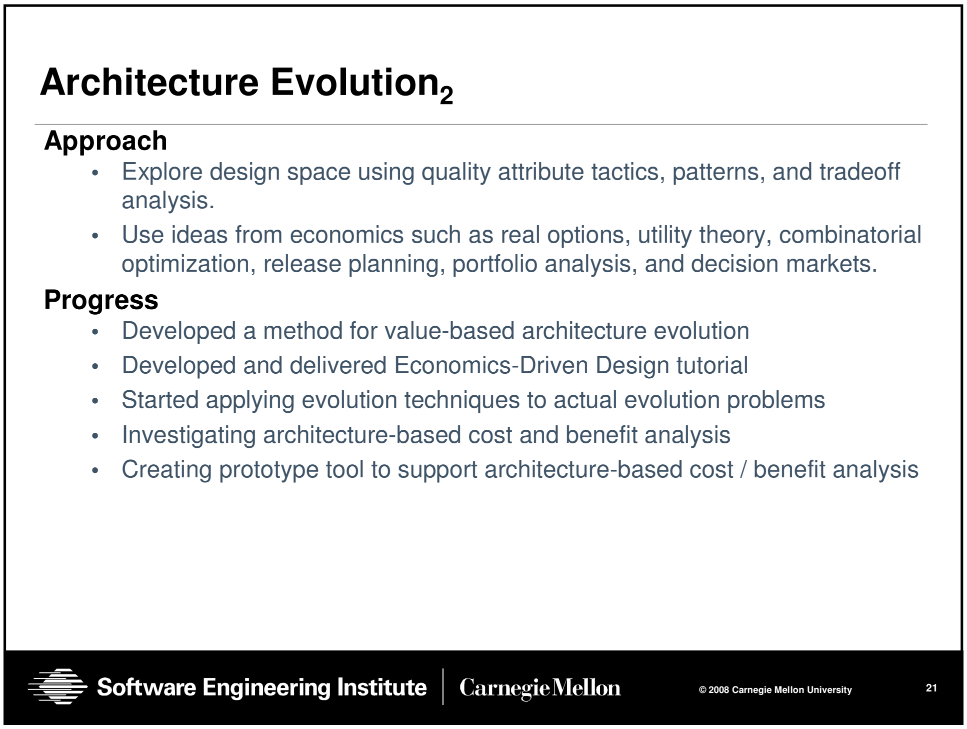
ATAM® Led to the Development of Other Methods and Techniques


What if the quality requirements are not well-understood?
Quality Workshop (QAW)
Our scenarios tend to be incomplete or ambiguous.
Quality Attribute General
What are some of the most important questions to ask?
Attribute Tactics
Business / Mission Context
Organization Context
Business Drivers
Quality Attributes
Scenarios
Software Architecture
Architectural Approaches
Architectural Decisions
Analysis
Tradeoffs
Sensitivity Points
Non-Risks
Risks
impacts
distilled into
What if there's no architecture?
Which risks should I work on first?
Attribute Driven Design (ADD)
Cost Benefit Analysis Method (CBAM)
Wh system's architecture?
System Context
Technology Context
What information should be included in my architecture documentation?
Architecture Reconstruction using ARMIN
Views and Beyond Approach (VaB)

Architecture Evolution$_1$

Problem

- *Systems evolve* to satisfy mission and business goals that change over time.
- Systems must evolve without compromising quality while being constrained by time and resource constraints.
- A sound practicable approach for architecture-based system evolution is needed. Approach should:
  - enable value-based architectural design and analysis
  - allow for tradeoffs between near- and long-term goals
  - foster communication between management and architects



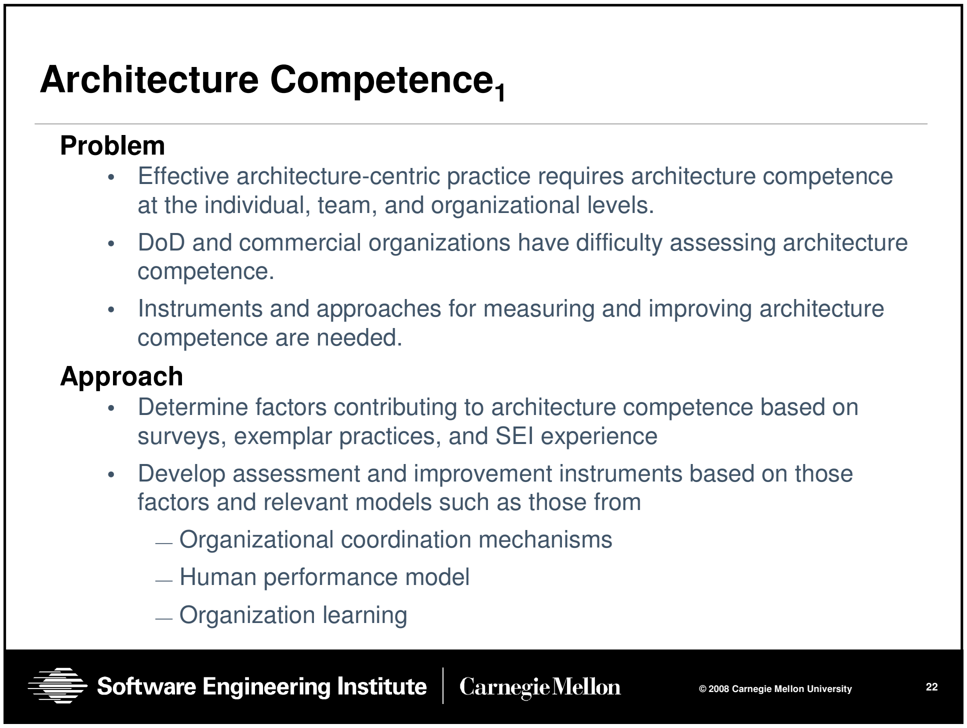
Architecture Evolution$_2$

Approach

- Explore design space using quality attribute tactics, patterns, and tradeoff analysis.
- Use ideas from economics such as real options, utility theory, combinatorial optimization, release planning, portfolio analysis, and decision markets.

Progress

- Developed a method for value-based architecture evolution
- Developed and delivered Economics-Driven Design tutorial
- Started applying evolution techniques to actual evolution problems
- Investigating architecture-based cost and benefit analysis
- Creating prototype tool to support architecture-based cost / benefit analysis



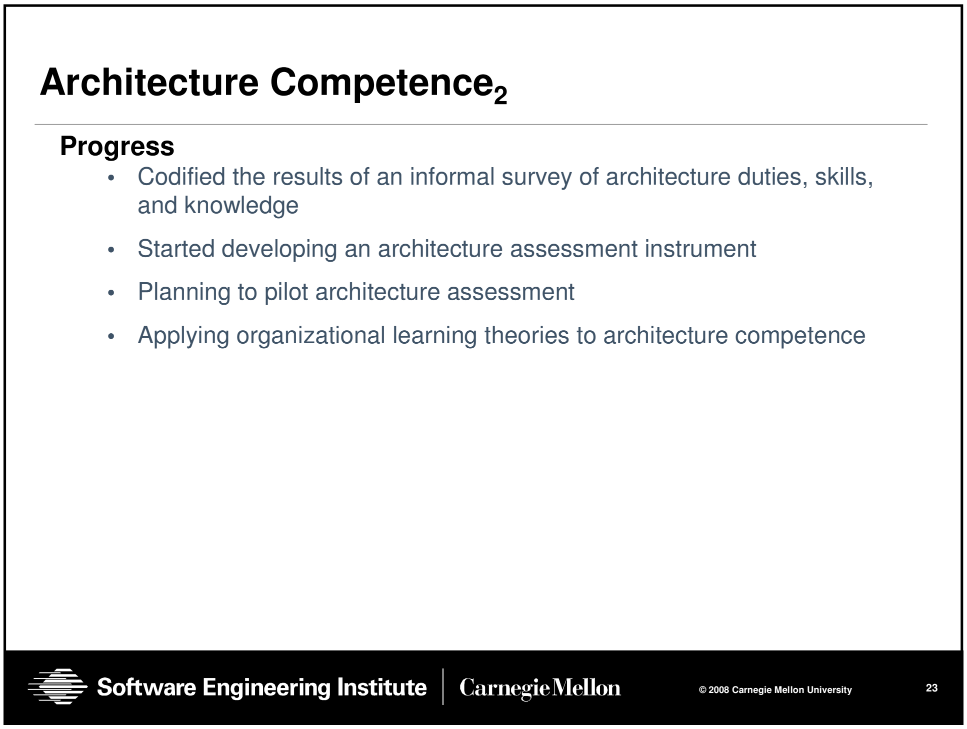
Architecture Competence$_1$

Problem

- Effective architecture-centric practice requires architecture competence at the individual, team, and organizational levels.
- DoD and commercial organizations have difficulty assessing architecture competence.
- Instruments and approaches for measuring and improving architecture competence are needed.

Approach

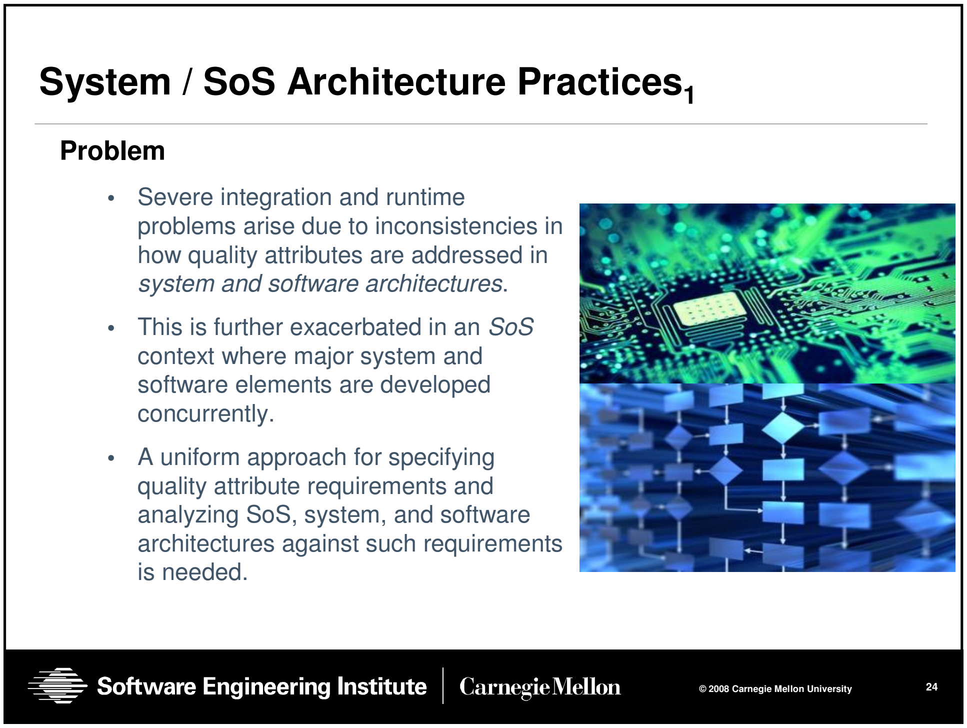
- Determine factors contributing to architecture competence based on surveys, exemplar practices, and SEI experience
- Develop assessment and improvement instruments based on those factors and relevant models such as those from
  - Organizational coordination mechanisms
  - Human performance model
  - Organization learning



Architecture Competence$_2$

Progress

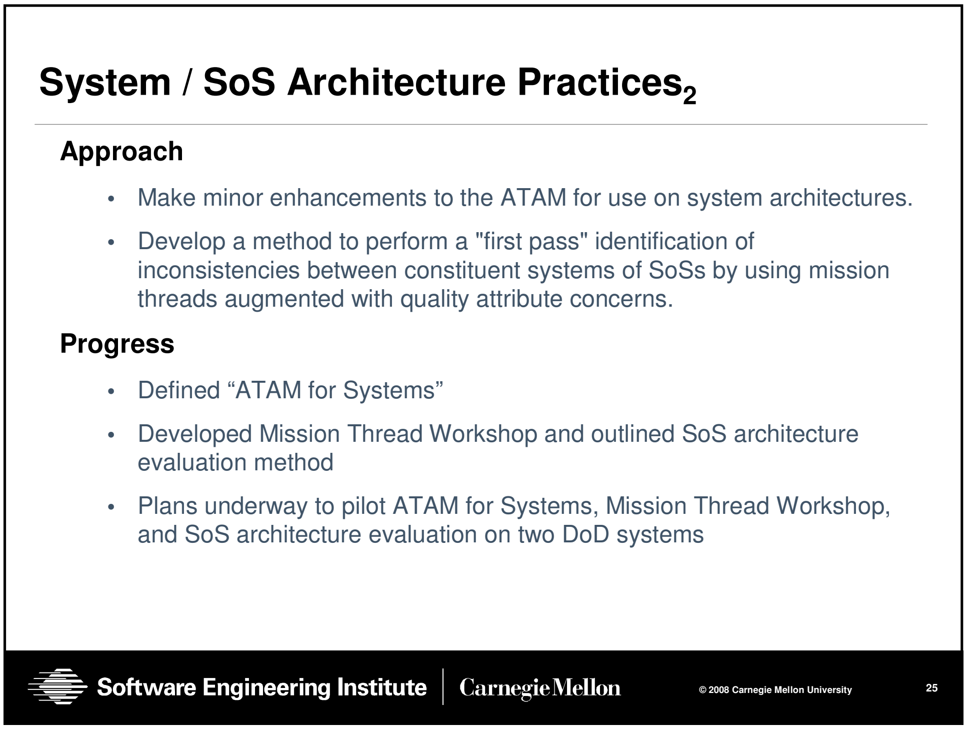
- Codified the results of an informal survey of architecture duties, skills, and knowledge
- Started developing an architecture assessment instrument
- Planning to pilot architecture assessment
- Applying organizational learning theories to architecture competence



System / SoS Architecture Practices$_1$

Problem

- Severe integration and runtime problems arise due to inconsistencies in how quality attributes are addressed in *system and software architectures*.
- This is further exacerbated in an *SoS* context where major system and software elements are developed concurrently.
- A uniform approach for specifying quality attribute requirements and analyzing SoS, system, and software architectures against such requirements is needed.



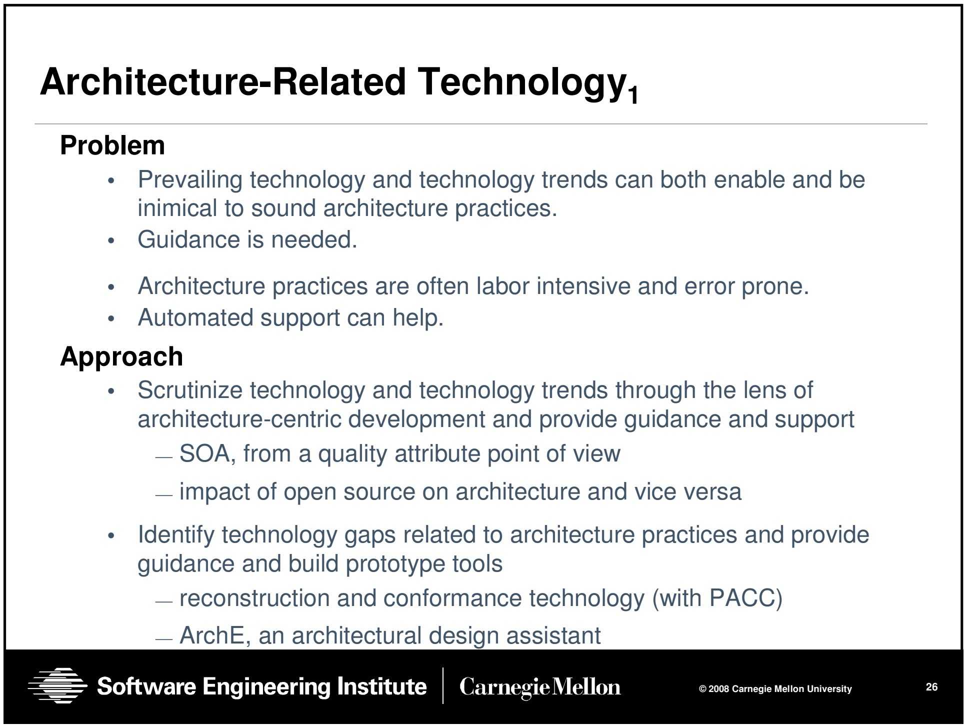
System / SoS Architecture Practices$_2$

Approach

- Make minor enhancements to the ATAM for use on system architectures.
- Develop a method to perform a "first pass" identification of inconsistencies between constituent systems of SoSs by using mission threads augmented with quality attribute concerns.

Progress

- Defined “ATAM for Systems”
- Developed Mission Thread Workshop and outlined SoS architecture evaluation method
- Plans underway to pilot ATAM for Systems, Mission Thread Workshop, and SoS architecture evaluation on two DoD systems



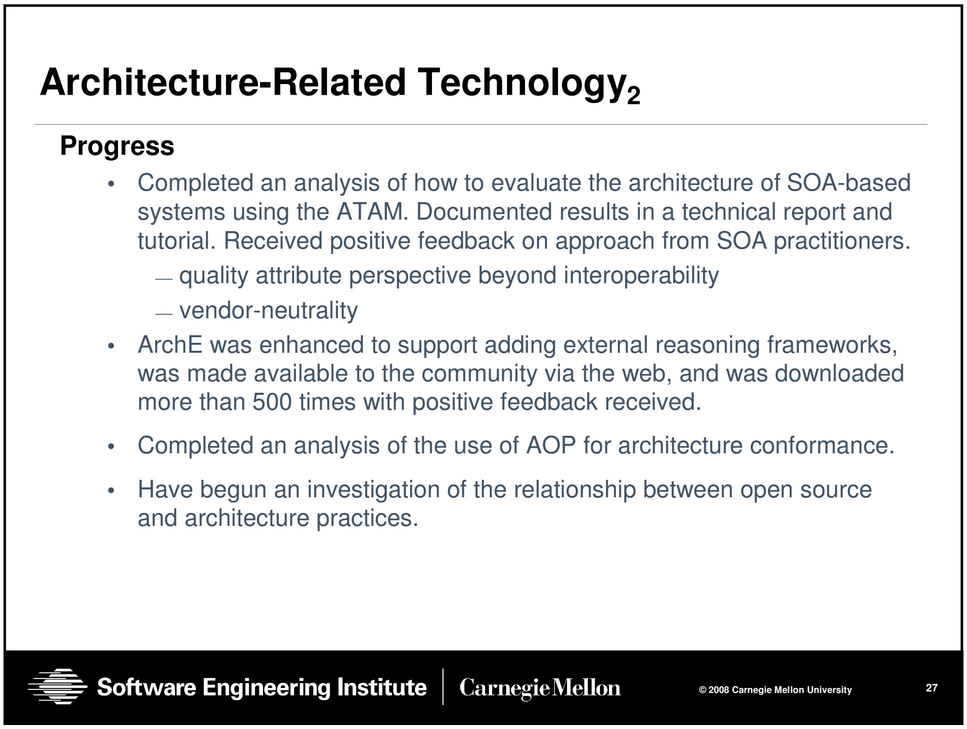
Architecture-Related Technology$_1$

Problem

- Prevailing technology and technology trends can both enable and be inimical to sound architecture practices.
- Guidance is needed.
- Architecture practices are often labor intensive and error prone.
- Automated support can help.

Approach

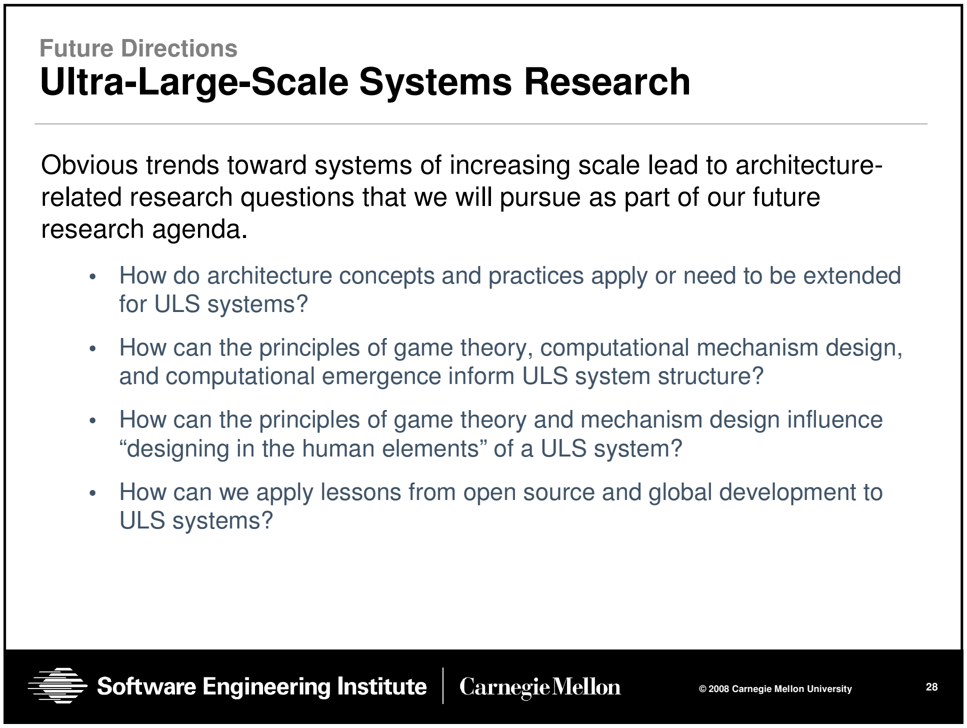
- Scrutinize technology and technology trends through the lens of architecture-centric development and provide guidance and support
  - SOA, from a quality attribute point of view
  - impact of open source on architecture and vice versa
- Identify technology gaps related to architecture practices and provide guidance and build prototype tools
  - reconstruction and conformance technology (with PACC)
  - ArchE, an architectural design assistant



Architecture-Related Technology$_2$

Progress

- Completed an analysis of how to evaluate the architecture of SOA-based systems using the ATAM. Documented results in a technical report and tutorial. Received positive feedback on approach from SOA practitioners.
  - quality attribute perspective beyond interoperability
  - vendor-neutrality
- ArchE was enhanced to support adding external reasoning frameworks, was made available to the community via the web, and was downloaded more than 500 times with positive feedback received.
- Completed an analysis of the use of AOP for architecture conformance.
- Have begun an investigation of the relationship between open source and architecture practices.

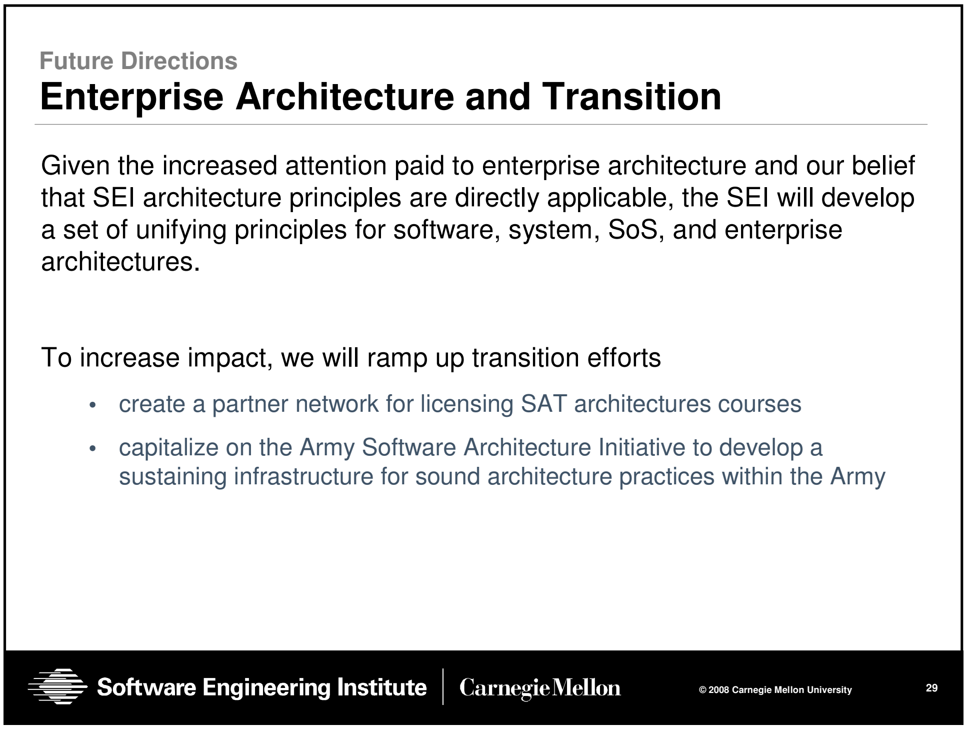


Future Directions

Ultra-Large-Scale Systems Research

Obvious trends toward systems of increasing scale lead to architecture-related research questions that we will pursue as part of our future research agenda.

- How do architecture concepts and practices apply or need to be extended for ULS systems?
- How can the principles of game theory, computational mechanism design, and computational emergence inform ULS system structure?
- How can the principles of game theory and mechanism design influence “designing in the human elements” of a ULS system?
- How can we apply lessons from open source and global development to ULS systems?



Future Directions

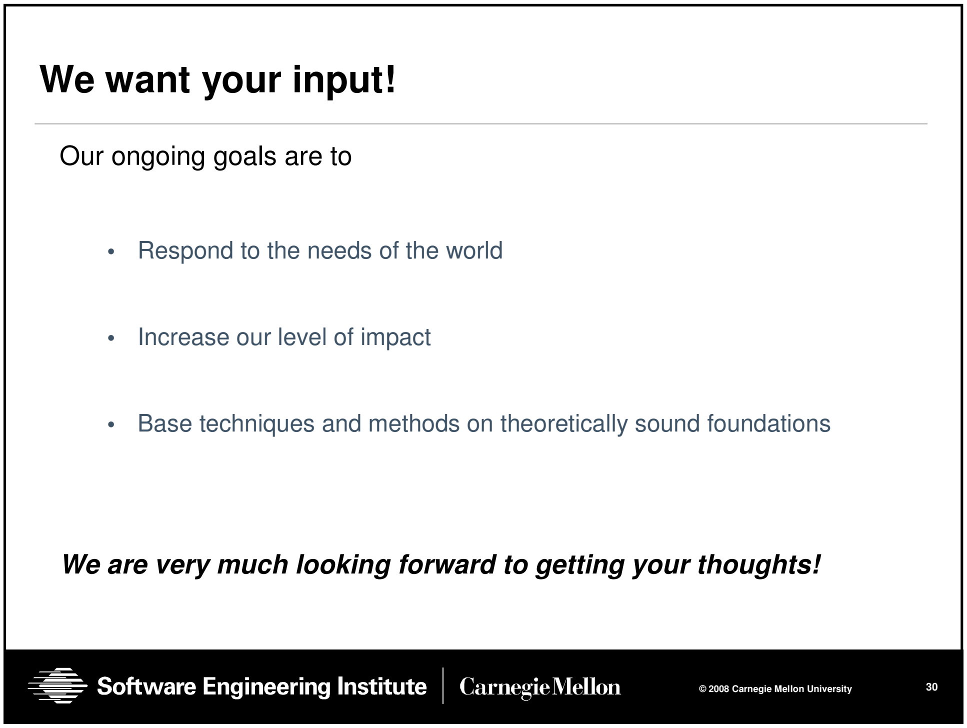
Enterprise Architecture and Transition

Given the increased attention paid to enterprise architecture and our belief that SEI architecture principles are directly applicable, the SEI will develop a set of unifying principles for software, system, SoS, and enterprise architectures.

To increase impact, we will ramp up transition efforts

- create a partner network for licensing SAT architectures courses
- capitalize on the Army Software Architecture Initiative to develop a sustaining infrastructure for sound architecture practices within the Army

Software Engineering Institute | Carnegie Mellon © 2008 Carnegie Mellon University 29



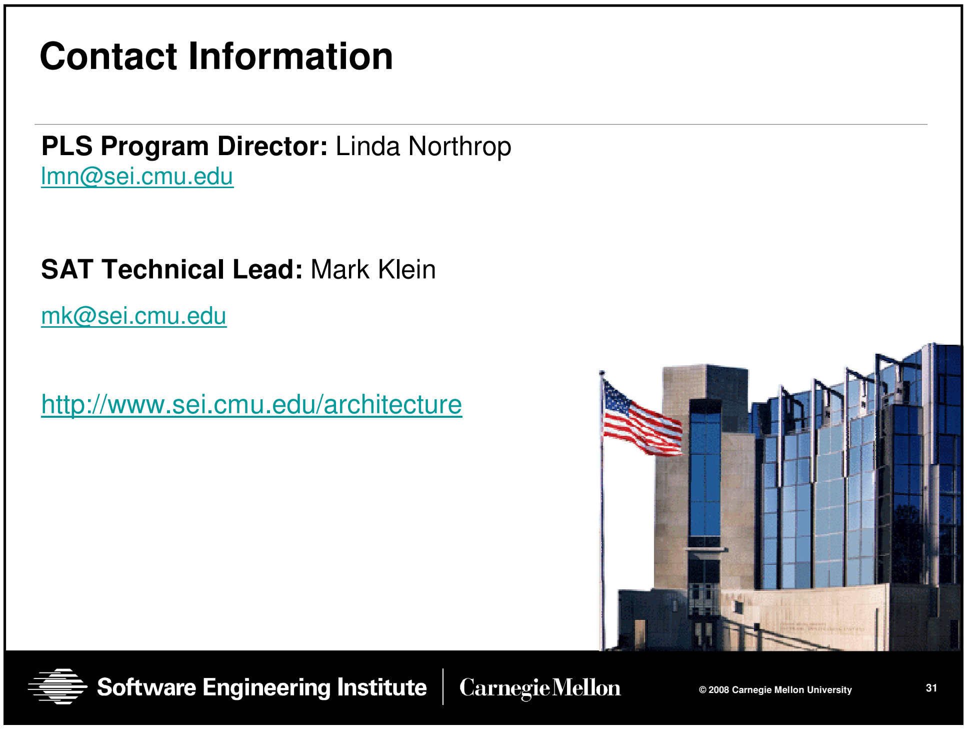
We want your input!

Our ongoing goals are to

- Respond to the needs of the world
- Increase our level of impact
- Base techniques and methods on theoretically sound foundations

We are very much looking forward to getting your thoughts!

Software Engineering Institute | Carnegie Mellon © 2008 Carnegie Mellon University 30



Contact Information

PLS Program Director: Linda Northrop
lmn@sei.cmu.edu

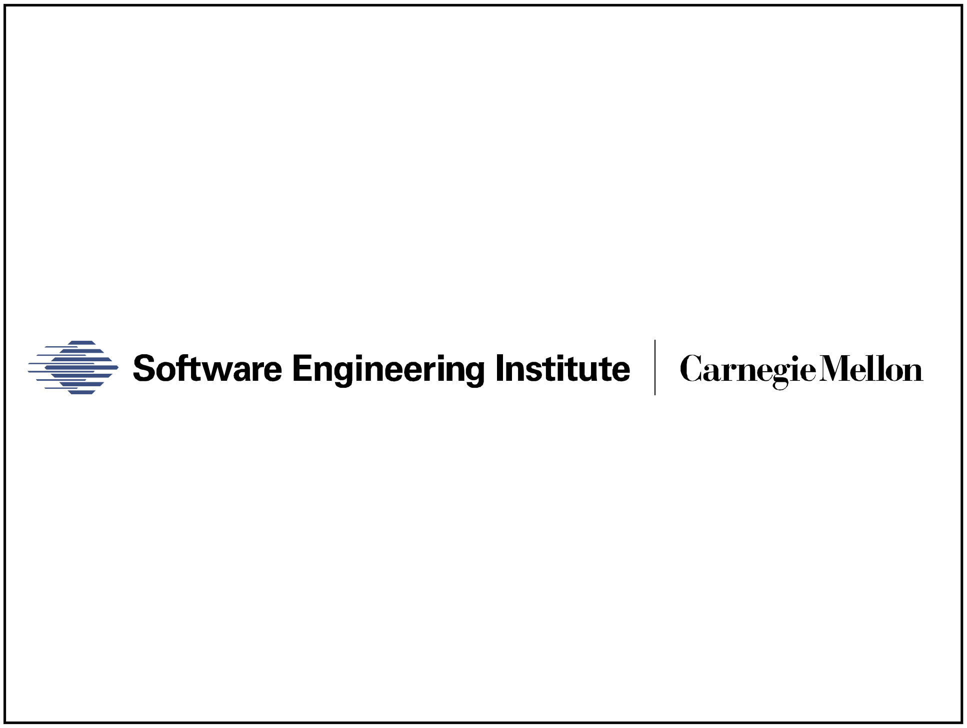
SAT Technical Lead: Mark Klein

mk@sei.cmu.edu

http://www.sei.cmu.edu/architecture

Software Engineering Institute | Carnegie Mellon

31



Software Engineering Institute | Carnegie Mellon