For many organizations, the achievement of business and mission goals is frustrated by software system problems such as:

- poor quality—seen in, for example, systems crashing or behaving unpredictably, or users judging that a system is unusable; and
- slow time to market—seen in missed integration deadlines due to difficulty in identifying root causes of problems.

In the Architecture-Centric Engineering (ACE) Initiative, we aim to improve product development and quality by using architecture to gain early confidence in achieving system-related business and mission goals.

**Our Work**

We develop and use techniques for analyzing quality attribute behavior based on a system’s architecture. Our current research focuses on quality attribute foundations in cyber-physical systems, architecture-centric practices for large-scale systems, and architecture principles for ultra-large-scale (ULS) systems.

The ACE approach results in:

- early identification and mitigation of design risks—yielding fewer downstream, costly problems and cost savings in integration and test
- predictable system quality—creating competitive advantage
- flexibility—enabling cost-effective system evolution

**How We Can Help**

The ACE team can help organizations and individuals through:

- designing and analyzing software, system, and system-of-systems (SoS) architectures relative to business and mission goals
- developing precise models and quantitative analyses of systems for mission-critical quality attributes such as safety, performance, and reliability
- evaluating architecture evolution technical plans
- reconstructing a system’s architecture from its implementation
- assessing and improving organizational architecture competence
- coaching to help introduce architecture-centric development to a software organization

**Our Methods**

- Architecture Tradeoff Analysis Method® (ATAM®)—for evaluating software architectures relative to quality attribute goals
- System and Software ATAM and SoS Architecture Evaluation—for evaluating system and SoS architectures, respectively, to ensure early identification of quality attribute inconsistencies and architectural risks
- Architecture Analysis and Design Language (AADL)—for specifying analyzable architecture models and enabling early discovery of integration problems
- Architecture Improvement Workshop (AIW)—for improving software architectures relative to quality attribute goals
- Architecture Evolution Workshop (AEW)—for evaluating architecture evolution plans relative to evolution-specific quality attribute goals
- Attribute-Driven Design (ADD) Method—for designing the software architecture of a software-intensive system
- Quality Attribute Workshop (QAW)—for identifying a system’s architecture-critical quality attributes
- Mission Thread Workshop (MTW)—for eliciting an SoS’s mission threads and augmenting them with architecture-critical quality attributes
- Architecture Documentation, Views and Beyond (V&B) Approach—for
guidance on documenting a software architecture
• Active Reviews for Intermediate Designs (ARID)—for ensuring quality detailed designs in software
• Cost Benefit Analysis Method (CBAM)—for determining the costs and benefits of architectural decisions

Our Courses
• Courses in software architecture essentials, documenting software architecture, software architecture design and analysis, ATAM Evaluator and Leader roles, and modeling system architectures
• Software Architecture Professional and ATAM Evaluator certificates
• SEI certification program for ATAM Leaders

Research Collaboration Opportunities
ACE researchers are interested in teaming with in areas such as
• developing techniques for agile architecting to balance rapid capability delivery with preparation for future capability delivery
• investigating the use of economic analyses for architecture evolution
• developing new metrics and approaches for using architecture knowledge to assure the reliability of software-reliant systems
• investigating techniques for using architecture knowledge to reduce system testing
• developing resource allocation strategies and analyses for predicting real-time performance
• developing analytic techniques for making system-wide tradeoffs between conflicting quality attributes
• identifying SoS architecture patterns and associated quality attribute analyses
• developing protocols and analyses for ensuring predictable quality of service in wireless devices
• investigating the use of scalable static analysis in the migration of systems to multi-core platforms
• investigating the use of market-based mechanisms for resource allocation

Relationship to Other SEI Work
The ACE Initiative resides within the Research, Technology, and System Solutions (RTSS) Program, which is part of the Carnegie Mellon® Software Engineering Institute (SEI). The RTSS team focuses on the structure and behavior of software-reliant systems—and the intimate relationship between structure and manifested quality attributes—to enable assured and flexible system capabilities at all scales.

Related Website
www.sei.cmu.edu/architecture/

For More Information
SEI Customer Relations
Phone: 412-268-5800
FAX: 412-268-6257
Email: info@sei.cmu.edu
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15313-2512

About the SEI: The SEI is a federally funded research and development center that provides the technical leadership to advance the practice of software engineering, so that software-reliant systems can be acquired and sustained with predictable and improved cost, schedule, and quality.