Proceedings of the First Software Architecture Technology User Network (SATURN) Workshop

Robert L. Nord
Len Bass
Paul Clements
Linda Northrop
James E. Tomayko

September 2005

Software Architecture Technology Initiative

Technical Note
CMU/SEI-2005-TN-037

This work is sponsored by the U.S. Department of Defense.

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

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.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Contents

1 Introduction 1

2 List of Participants 3

3 Presentations 6

3.1 Keynote Speakers 6

3.1.1 Making the Role Your Own: Notes on Process Adoption at Pitney Bowes 6

3.1.2 Architecture Reviews @ Bosch 7

3.1.3 Leading the World to a Software-Enriched Society 7

3.2 Presentations 7

3.2.1 The ABACUS Architectural Approach to Computer-Based Systems and Enterprise Evolution 7

3.2.2 Architecture Design Expert 8

3.2.3 Are All Quality Goals Created Equal? 8

3.2.4 ATAM Experiences 8

3.2.5 An Experience Report on Using UML 2.0 to Document Software Architectures 8

3.2.6 Implementing the SEI's Software Architecture Technology: Principles and Variations 9

3.2.7 Integrating Software Architecture Evaluation in a DoD System Acquisition 9

3.2.8 Product Line Engineering for Global Development 9

3.2.9 Quality-Attribute-Driven Software Architecture Reconstruction 10

4 Working Sessions 11

4.1 Gaps in the SEI Architecture-Centric Methods 12

4.1.1 Tailoring the ATAM 12

4.1.2 Identifying Prerequisite Knowledge for Performing an Architecture Evaluation Using the ATAM 12

4.1.3 Standardizing the ATAM 13

4.1.4 Applying the ATAM to Large Systems, Enterprise Architectures, and System Architectures 13

4.2 Measurement 14

4.2.1 What to Measure 14

4.2.2 Why We Measure 14
4.2.3 For Whom We Measure 14
4.2.4 When We Measure 15
4.2.5 How to Measure 15
4.2.6 Benefits of Architecture 15
4.2.7 Benefits of Architecture Activities 16
4.3 Architecture and Process 17
4.3.1 Models 18
4.3.2 Architect's Role and Authority 19
4.3.3 Adoption 19
4.3.4 Standards 19

5 Closing Session 20

6 Future of SATURN 22

References 23

List of Tables

Table 1: Distribution of SATURN Participants 3

Table 2: Software Architecture Benefits and Stakeholders..... 16

Abstract

The first Carnegie Mellon® Software Engineering Institute (SEI) Software Architecture Technology User Network (SATURN) Workshop was held on April 6-7, 2005, at the SEI in Pittsburgh, Pennsylvania. Software systems engineers, architects, technical managers, and product managers exchanged best practices and lessons learned in applying SEI software architecture technology in an architecture-driven development or acquisition project. In the closing session, workshop participants noted the following highlights: peer collaboration, shared understanding, SEI technical staff presence, developing metrics that measure benefits, exploring case studies that highlight how to apply architecture-centric methods, learning what's new in software architecture, learning about the acquisition support available for software architecture, and agreeing that software architecture technology has reached the early majority.

This report describes the workshop format, discussion, and results, as well as plans for future SATURN workshops.

1 Introduction

The first Carnegie Mellon® Software Engineering Institute (SEI) Software Architecture Technology User Network (SATURN) Workshop was held on April 6-7, 2005, at the SEI in Pittsburgh, Pennsylvania.

The goal of the workshop series is to bring together software systems engineers, architects, technical managers, and product managers to share experiences using SEI software architecture technology. Participants discuss ideas, issues, and needs related to software architecture practices and develop a network of individuals who are interested in using and improving those practices. SEI architecture-centric methods include the Quality Attribute Workshop (QAW) [Barbacci 03], Attribute-Driven Design (ADD) [Bass 03], Active Reviews for Intermediate Designs (ARID) [Clements 02], the Architecture Tradeoff Analysis Method® (ATAM®) [Clements 02], the Cost Benefit Analysis Method (CBAM) [Bass 03], the Views and Beyond (V&B) approach to documentation [Clements 03], and Quality-Attribute-Driven Software Architecture Reconstruction (QADSAR) and the Architecture Reconstruction and Mining (ARMIN) tool [Kazman 02]. These methods are based on a core set of attribute models, reasoning frameworks, and architectural tactics.

Participants discussed the challenges they face in meeting quality attribute requirements, predicting quality attribute behavior, and making practical and informed tradeoffs about quality attributes early in the software development life cycle. The SATURN workshop provided a unique opportunity for participants to learn from each other about how to make progress towards the common goal of using effective software architecture practices across the life cycle to ensure predictable product qualities, costs, and schedules. It also provided an opportunity to give feedback to the SEI about promising future directions in software architecture technology and practices.

During this first SATURN workshop, 50 participants exchanged best practices and lessons learned in applying SEI software architecture technology in an architecture-driven development or acquisition project. The workshop consisted of 12 talks (including keynotes), 2 working sessions (covering 3 topics and running in parallel), a reception, and opening and closing sessions.

® Carnegie Mellon, Architecture Tradeoff Analysis Method, and ATAM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Linda Northrop, director of the SEI Product Line Systems Program, gave the kickoff and closing presentations. Keynote speakers included

- Nanette Brown, director, Applied Architecture and Quality Assurance, Pitney Bowes, Inc.
- Stefan Ferber, director, C-SEPG, Robert Bosch GmbH
- Paul Nielsen, director and chief executive officer (CEO), SEI

This report describes the workshop format, discussion, and results, as well as plans for future SATURN workshops. It is organized as follows:

- Section 2 lists workshop participants.
- Section 3 provides an overview of the presentations.
- Section 4 includes discussion from the working sessions: 1) Gaps in the SEI Architecture-Centric Methods, 2) Measurement, and 3) Architecture and Process.
- Section 5 includes the closing session discussion that focused on themes emerging from the workshop, workshop highlights, and what to do next.
- Section 6 describes the future of SATURN workshops.

2 List of Participants

The 50 SATURN workshop participants were from the sectors shown in Table 1.

Table 1: Distribution of SATURN Participants

Sector	**Number**
Department of Defense (DoD)	4
DoD contractor	4
U.S. commercial	13
International commercial	5
Academia	7
Federally funded research and development centers (FFRDCs) other than the SEI	1
SEI (members of the Product Line Systems Program)	10
SEI (members outside of the Product Line Systems Program)	5
Other	1

Participants included

- Alberto Avritzer, member of the technical staff (MTS)/architect, Siemens Corporate Research
- Felix Bachmann, senior member of the technical staff (SMTS), SEI
- Len Bass, SMTS, SEI
- Matt Bass, MTS, Siemens, and SEI resident affiliate
- John Bergey, SMTS, SEI
- Kevin Bodie, engineering manager, Architecture and Technology, Pitney Bowes, Inc.
- Nanette Brown, director, Applied Architecture and Quality Assurance, Pitney Bowes, Inc.
- Lisa Brownsword, SMTS, SEI
- Dennis Buchovecky, vice president, Product Development Systems & Solutions, Inc.

- Michael Burke, chief software architect, Visteon Corporation
- Angela Llamas Butler
- Michael Clark, senior systems engineer, Carnegie Mellon University, Harris
- Paul Clements, SMTS, SEI
- Larry Cramer, DaimlerChrysler Corporation
- Arthur Culbertson, principal architect, Lockheed Martin Information Technology
- Stefan Ferber, director, C-SEPG, Robert Bosch GmbH
- Mark Hodge, chief engineer, Raytheon Integrated Combat Systems
- Xiaohong Jin, research engineer, Corporate Research, ABB AB
- Rick Kazman, visiting scientist, SEI
- Madhu Keshavamurthy, Ansys, Inc.
- Mark Klein, SMTS, SEI
- Constantin Kostenko, Booz Allen Hamilton
- Edward Lee, computer engineer, U.S. Army Armament Research, Development, and Engineering Center (ARDEC)
- Grace Lewis, SMTS, SEI
- Jim Linnehan, Secretary of the Army, Acquisition, Logistics, and Technology (SAALT), Headquarters, Department of the Army (HQDA)
- Adimoorthy Mahadevan, consultant, Knowledge Inherited Systems LLC
- David Mason, software engineer, CECOM/PM WIN-T
- Cameron Maxwell, Institute for Information and Communication Technologies, University of Technology Sydney
- Paulo Merson, MTS, SEI
- J. Darby Mitchell, software engineer, Massachusetts Institute of Technology (MIT) Lincoln Laboratory
- Tim Morrow, MTS, SEI
- Edward Neubecker, IBM-Rational
- Paul Nielsen, director and CEO, SEI
- Robert Nord, SMTS, SEI
- Linda Northrop, director, Product Line Systems Program, SEI
- Liam O'Brien, SMTS, SEI
- Tim O'Neill, deputy director, Architecture-Based Engineering Research Program, Institute for Information and Communication Technologies, University of Technology Sydney
- Artem Parakhine, Institute for Information and Communication Technologies, University of Technology Sydney

- Daniel J. Paulish, program manager, Siemens Corporate Research
- Bertrand Salle, director of programs, SESAM-Vitale
- Nigel Sheridan-Smith, PhD research student, Institute for Information and Communication Technologies, University of Technology Sydney
- Jeannine Siviy, technical advisor to the director and CEO, SEI
- Bob Smith, MIT
- Ben K. Steele, consultant, PDSS, Inc.
- Eric Stephens, solution architect, Excellus BlueCross BlueShield
- John Steven, technical director, Office of the Chief Technical Officer (CTO), Cigital, Inc.
- Sujata Telang, lecturer, Institute for Software Research International (ISRI), Carnegie Mellon University
- James E. Tomayko, visiting scientist, Carnegie Mellon University
- George Vinansky, physical scientist, ARDEC
- Yue Zhao, consulting scientist, Carnegie Mellon University

3 Presentations

Linda Northrop, director of the SEI Product Line Systems Program, gave the kickoff presentation: *SATURN—SEI Software Architecture Technology User Network.* The SEI has been working in the area of software architecture for over two decades, and its impact is far ranging:

- Individuals from more than 180 different companies have taken courses in the SEI Software Architecture Curriculum.
- The U.S. Army has launched a software architecture technology initiative based on SEI software architecture technology.
- Companies like Raytheon, Boeing, and Robert Bosch GmbH use the ATAM to evaluate software architectures.
- The military and its contractors use the ATAM to uncover risks in critical systems (e.g., Wargame 2000, Missile Defense Wargame and Analysis Resource [MDWAR], next generation destroyer [DDX], Force XXI Battle Command Brigade and Below [FBCB2], and Future Combat Systems [FCS]).

Today, organizations of all sizes are recognizing the importance of software architecture. Books, courses, certificate programs, conferences, and workshops on software architecture abound. New technologies (model-driven architecture [MDA], service-oriented architecture [SOA], aspects) change the incidentals, but the fundamentals of software architecture and quality attributes endure. The SEI considered it time to create a forum for practitioners to meet, exchange best practices, and describe their experiences with software architecture technology.

The following sections include descriptions of the three keynote speakers' remarks and abstracts of the nine presentations. Slides for the presentations are available at http://www.sei.cmu.edu/architecture/saturn/2005_abstracts_presentations.html.

3.1 Keynote Speakers

3.1.1 Making the Role Your Own: Notes on Process Adoption at Pitney Bowes

Nanette Brown, director, Applied Architecture and Quality Assurance, Pitney Bowes, Inc.

Brown described how her company recently adopted a quality attribute approach to add more rigor and focus to its architecture review process.

3.1.2 Architecture Reviews @ Bosch

Stefan Ferber, director, C-SEPG, Robert Bosch GmbH

Ferber shared how using the ATAM for five years to review important software and system architectures has raised stakeholders' awareness of architectural decisions, tradeoffs, and risks.

3.1.3 Leading the World to a Software-Enriched Society

Paul Nielsen, director and CEO, SEI

Nielsen spoke about the value of software architecture and the SEI vision of leading the world to a software-enriched society. For more than 20 years, the SEI has had the national mandate to advance the state of the practice of software engineering and to serve as a national resource in software engineering and technology. The SEI strategy is to openly engage a broad-based community with a focus on improving the effects of software in the world.

3.2 Presentations

3.2.1 The ABACUS Architectural Approach to Computer-Based Systems and Enterprise Evolution

Tim O'Neill, deputy director, Architecture-Based Engineering Research Program, Institute for Information and Communication Technologies, University of Technology Sydney

The enterprise computer-based systems used by today's organizations can be extremely complex. Not only do they consist of countless hardware and software products from many varied sources, but they often span continents, piggybacking on public networks. These systems are essential for undertaking business and general operations in the modern environment, but the ability of organizations to control their evolution is questionable.

The emerging practice of enterprise architecture seeks to control that complexity through the use of a holistic and top-down perspective. However, the tool sets already in use are very much bottom up by nature. To overcome the limitations of current enterprise architecture practices, use of the Architecture-Based Analysis of Complex Systems (ABACUS) methodology and tool set is proposed.

Using ABACUS to analyze software and enterprise systems, architects can guide the design and evolution of architectures based on quantifiable nonfunctional requirements. Furthermore, hierarchical 3-D visualization provides a meaningful and intuitive means for conceiving and communicating complex architectures.

3.2.2 Architecture Design Expert

Felix Bachmann, SMTS, and Mark Klein, SMTS, Product Line Systems Program, SEI

The goal of the architecture design expert (ArchE) tool is to help an architect make architectural decisions that support the system's quality attribute requirements. This presentation described the architecture design process and demonstrated, using the ArchE tool, how implemented quality attribute knowledge can help in designing software systems.

3.2.3 Are All Quality Goals Created Equal?

John Steven, technical director, Office of the CTO, Cigital, Inc.

Process-savvy organizations highlight nonfunctional attributes of software and have provided process tools to help organizations consider software beyond its features. But are all these nonfunctional (or quality) goals created equal?

When considering each quality goal, different activities are optimal. Performance takes a different knowledge set than maintainability. Tools that help with code vulnerabilities operate very differently than tools that probe code for performance bottlenecks.

When can practitioners think about quality goals as a whole, and at what level of process does each quality goal demand its own attention? When will development benefit from generic methods such as the QAW, ARID, and the ATAM, and when will they need specific activities (such as Rational Unified Process's [RUP's] Comprehensive, Lightweight Application Security Process [CLASP]) and knowledge?

In this presentation, the speaker addressed these questions using his own experiences.

3.2.4 ATAM Experiences

Xiaohong Jin, research engineer, Corporate Research, ABB AB

Corporate Research at ABB AB uses the ATAM to analyze the architectures of ABB's applications and systems, which improves the company's software practices and provides decision-making support for its stakeholders.

3.2.5 An Experience Report on Using UML 2.0 to Document Software Architectures

Arthur Culbertson, principal architect, Lockheed Martin Information Technology

The success of UML 1.x as a notation supporting a broad range of software modeling requirements has led to the language's emergence as the standard medium of communication for the software engineering community. However, UML 1.x does not provide constructs well suited to documenting software architectures, and attempts to adapt UML 1.x semantics to

support software architecture concepts have yielded mixed results. UML 2.0, which was recently adopted as a standard by the Object Management Group (OMG), provides a number of new and modified constructs that address several key deficiencies of UML 1.x related to software architecture. Despite these enhancements, the size and complexity of the UML 2.0 specification—combined with the lack of experience-based guidance—presents a serious challenge to practitioners who want to adopt UML 2.0 as a comprehensive notation for documenting software architectures.

3.2.6 Implementing the SEI's Software Architecture Technology: Principles and Variations

Bertrand Salle, director of programs, SESAM-Vitale

This presentation, which was intended to be interactive, examined the attempts (and successes) of a consultant, architect, and manager at deploying the SEI's software architecture technology with a particular focus on the QAW, ADD, the V&B approach, and the ATAM.

The following questions were addressed:

- How does the adoption process apply for the consultant, the architect, and the manager?
- What does the SEI's software architecture technology look like after the teachers and consultants are gone?
- What things worked the best?
- What other things were adapted by the organizations?

3.2.7 Integrating Software Architecture Evaluation in a DoD System Acquisition

John Bergey, SMTS, Product Line Systems Program, SEI and Tim Morrow, MTS, Acquisition Support Program, SEI

This presentation described how the Common Link Integration Processing (CLIP) program applies software architecture technology to an acquisition program. CLIP's background, quality attributes, QAW, stage in the DoD 5000 Acquisition Framework, and lessons learned were addressed. CLIP is a joint U.S. Air Force, Marine, and Navy effort that focuses on standard handling of tactical data links.

3.2.8 Product Line Engineering for Global Development

Daniel J. Paulish, program manager, Siemens Corporate Research

This presentation described how product line engineering practices are being used in Siemens to better plan and manage global development projects. Software products are growing in complexity, and the development organizations that implement new features are growing in

size. A summary was provided of an approach that decomposes large-scale requirements into a well-structured set of software components that can be developed in parallel among globally distributed development teams. The approach applies best practices of software requirements engineering including business object modeling coupled with product line architecture design. Agile development processes are exploited so that a collection of development teams for small, distributed application components are controlled by a central organization. Applying modern industrial practices to requirements, design, and organizational patterns should yield substantial time-to-market and productivity improvements.

3.2.9 Quality-Attribute-Driven Software Architecture Reconstruction

Liam O'Brien, SMTS, Product Line Systems Program, SEI

Architecture reconstruction is the process by which architectural views of an implemented system are obtained from existing artifacts. This presentation describes research on architecture reconstruction that is driven by quality attribute analysis. The analysis typically occurs when existing systems hit their architectural boundaries caused by product growth or expansion scenarios. The information gathered during architecture reconstruction has to satisfy the information needs for these scenarios in order to provide reasoning during decision-making processes. The work is crucial for organizations that have to make architectural decisions about existing systems or want to lower the adoption barriers for product lines by investigating the reuse of existing assets.

4 Working Sessions

Three working sessions were scheduled to provide further discussion of topics related to software architecture. Participants proposed the following working session topics during a brainstorming and consolidation session, and items 2, 3, and 4 were selected for the working sessions.

1. how to build a software architecture community (i.e., how to condition an organization to accept architectural practices)
2. gaps in the SEI architecture-centric methods
   - tailoring methods
   - standardizing the ATAM, the value of having a standard, and how to get one
   - applicability to other architectures (e.g., enterprise architectures)
   - prerequisite (domain) knowledge for the ATAM
3. measurement (i.e., what to measure)
4. architecture and process
   - the relationship of architecture and process
   - how to build systems (how the organization is structured to meet quality attribute requirements)
5. how quality attribute scenarios are being used in architecture reconstruction
6. architecture visualization

4.1 Gaps in the SEI Architecture-Centric Methods

During this working session, the following topics were discussed:

- tailoring the ATAM
- identifying prerequisite knowledge for performing an architecture evaluation using the ATAM
- standardizing the ATAM
- applying the ATAM to large systems, enterprise architectures, and system architectures

4.1.1 Tailoring the ATAM

Participants were satisfied with the ATAM's steps but made the following suggestions for improving the process of capturing the results:

- Include system measures (e.g., size, number of distinct locations where the system was being developed, number of developers) in the statistics.
- Include a follow-up step that questions developers about their response to identified risks. For example, during a phone conference three to four weeks after the conclusion of an architectural review, developers at AT&T report actions taken to address the serious risks that were identified.

One participant commented that nothing should be done to change the ATAM from a collaborative exercise between the evaluators and the project team into an adversarial exercise.

4.1.2 Identifying Prerequisite Knowledge for Performing an Architecture Evaluation Using the ATAM

There was general agreement that the evaluation team should have expertise in the

- ATAM
- quality attributes of importance to the system being reviewed
- domain(s) addressed by the system

There was also general agreement that not every member of the evaluation team needed to be expert in all of these areas. Different teams would be constituted from among the available personnel so that all of the prerequisite knowledge is represented.

Gaining the prerequisite knowledge about the ATAM can be done in a variety of ways:

- Take courses that are part of the SEI's Software Architecture Curriculum and Certificate Programs (for more information, visit http://www.sei.cmu.edu/architecture/arch_curriculum.html).
- Purchase the ATAM evaluation book [Clements 02].

- Participate in ATAMs led by trained evaluators.

The entry method becomes a cost/benefit question. Taking the courses and receiving training reduces the learning time and the probability of errors, but cost may be an issue. Different members of the group who had performed ATAM evaluations had arrived there via different routes.[1]

4.1.3 Standardizing the ATAM

Participants felt that if the ATAM was officially recognized outside of the SEI, it would be easier for them to convince their organizations to perform ATAM exercises. One possibility is for the ATAM to become an official (e.g., Institute of Electrical and Electronics Engineers, Inc. [IEEE] or International Standards Organization [ISO]) standard.

Another method was for the DoD to require ATAM evaluations for its large contracts. The Army is currently putting language in its requests for proposals (RFPs) for Acquisition Category (ACAT) I and II programs to require ATAM evaluations.

Although participants felt that having an official standard would be beneficial, no one was willing to participate in standardization efforts.

4.1.4 Applying the ATAM to Large Systems, Enterprise Architectures, and System Architectures

Applying the ATAM to large systems, enterprise architectures, and system architectures introduces problems of scale and scope.

Problems of scale occur in any large system with multiple subsystems regardless of whether the ATAM is applied to a software, system, or enterprise architecture. Evaluating individual subsystems—as well as interactions between subsystems—in a single ATAM exercise is problematic in three ways: (1) there are too many issues for the evaluators, (2) the stakeholders are different, and (3) the important quality attributes may be different. It is too difficult to get to enough depth to evaluate the architectural decisions. One participant in the breakout group said, “The larger the system, the more the evaluation focuses on management rather than on the technical issues.”

Problems of scope are introduced in a system architectural context. Although the ATAM is agnostic regarding the quality attributes it evaluates, attributes such as power consumption, physical footprint, and physical environment are important when examining system architectures. The types of expertise needed in the evaluation team grows far beyond software and quality attributes.

[1] SEI-authorized evaluations using the ATAM must be performed by individuals who have received the SEI ATAM Evaluator Certificate and must be led by an SEI-certified ATAM Lead Evaluator.

Participants recommended that scope problems be dealt with in Phase 0 of the ATAM evaluation. One way the scope of an evaluation can be captured is in the operational concept. If such a concept was defined for the system under consideration, the ATAM evaluation can focus strictly on the concept's scope rather than delving into subsystems or focusing on the high-level context within which the scope exists.

Participants also raised the possibility of a series of ATAM exercises for very large systems. For example, one exercise could evaluate individual subsystems, and another could evaluate interactions between subsystems.

4.2 Measurement

This working session discussed measuring the costs and benefits of architecture and architecture-centric practices. Because there is a large body of work on measuring cost, participants concentrated on benefits.

First, they established the groundwork for the discussion, covering what to measure and why, who measures and who consumes, and when to measure and how.

4.2.1 What to Measure

We can measure products and artifacts, or we can measure the architecture process. Architecture-centric measures related to products and artifacts start with the quality attributes of the deployed system that the architecture is intended to imbue (e.g., performance, security, and availability). Some quality attributes are direct manifestations of business goals, such as the time to market, return on investment (ROI), total cost of ownership, amount of reuse, and success of product migration/evolution. Process-related measures include time and productivity measures, such as the amount of rework or the cost of particular activities (e.g., evaluation).

4.2.2 Why We Measure

We measure to track progress, to see if we are meeting goals, and to predict the future based on trends. In an atmosphere of architecture evangelism, we also measure to make a case for architecture-centric development and practices. We measure to find a "poster child" activity that will make a positive case for architecture-based development, establishing a baseline for best practices and collecting lessons learned.

4.2.3 For Whom We Measure

Consumers of measures include leadership and management, financing authorities, sales and marketing staff, and engineering departments.

4.2.4 When We Measure

The real question is “How early can we measure?” The implicit assumption is that earlier measures (with an awareness of the associated problems and trends) are more desirable than later measures. Because participants were unable to identify any definitive milestones that would work in all situations on all projects, this discussion was abbreviated and inconclusive. However, participants agreed that management should be attuned to opportunities to take meaningful measurements throughout the project schedule.

4.2.5 How to Measure

Participants quickly concluded that how (and what and when) we measure should be tailored to produce results that are useful to the stakeholders who need them. Also, because measurement can be a costly activity, only those measures necessary should be collected.

4.2.6 Benefits of Architecture

Participants brainstormed the benefits of software architecture that we can be revealed through measurement. Table 2 lists benefits and includes the stakeholders who would reap them.

Table 2: Software Architecture Benefits and Stakeholders

Benefit	**Stakeholder**
Interoperability	Marketing staff
Productivity	Management
Reduced number of angry voicemails from CEO	CTO
Reduced number of calls at service center	Product manager
New mission capability in a short time	All stakeholders
Revenue (allows price target to be met)	CEO Marketing staff
Product/feature diversity	CTO Marketing staff
Enabled strategic direction (e.g., globalize)	CEO
Increased organizational knowledge	Chief operating officer (COO) Management
Product stability with respect to new technology	Product manager
Total cost of ownership versus time to market	Marketing staff
Long-term productivity and efficiency	CEO COO CTO

4.2.7 Benefits of Architecture Activities

Finally, participants discussed how to measure the benefits imbued by specific architecture-related activities.

For an ATAM-based architecture evaluation, they realized that two kinds of risks are uncovered: (1) risks that were previously known and (2) risks that were previously unknown. However, even if a risk is known, it would not necessarily be acted on. One important benefit of an ATAM evaluation is that management is made aware of risks that may have been known only by the technical staff, and management can then allocate resources and adjust the schedule to address those risks.

The participants reasoned that this expected benefit of an ATAM exercise can be expressed as follows:

SUM[i=1,n] [(cost of risk-i) x (probability of risk-i)] -
(cost of performing the evaluation)

where the risks are those uncovered during the evaluation that would not otherwise have been acted on. This benefit is minimal because it does not address intangible benefits such as increased stakeholder communication and improved documentation.

They then turned their attention to the benefit of architecture documentation. After brainstorming a list of reasons why documentation should be beneficial, the participants crafted the following expression to quantify this benefit:[2]

$\sum$ [i=1, n] {Δ(cost of some activity)i} - (cost of documentation)

Activities include coding, analysis, project management planning, maintenance, making changes, performing downstream design, testing, and so on. The delta (Δ) refers to the cost of performing that activity without documentation minus the cost of performing that same activity with documentation. Presumably, the cost of carrying out these activities with documentation will be lower. (For activities that are not affected by having documentation, the delta is zero; they do not contribute to the benefit equation.)

Participants suspected that this expression was easy to generalize to quantify the benefit of any architecture-related activity. They hypothesized that the benefit of an activity is the resulting cost reduction in subsequent activities minus the cost of the activity in the first place.

4.3 Architecture and Process

This working session discussed the relationship between architecture and process in the context of how to build systems and how the organization as a whole meets quality attribute requirements. Some initial questions were raised to establish the groundwork for the discussion:

- What processes are relevant?
- Is there a reference framework that people use?
- On which processes does the organizational context exist?
- How do we cope with processes that don't have an architecture focus?
- Are quality requirements met in the architecture, the organization, or both? How are the architecture and the organization aligned?

[2] The group originally added a term to this expression to count the *avoided cost of poor quality* to the benefit of documentation. However, subsequent consideration suggested that this avoided cost is captured in the reduced cost of subsequent activities.

Discussion then progressed through topics listed in Section 4.3.1 through 4.3.4.

4.3.1 Models

Participants made the following observations about models for architecture and process:

- Few organizations use a predefined framework. They are building their own.
- For the sake of argument, couldn't something like the Zachman framework be used as the common language or model for architecture and process? It is the most comprehensive framework, and it could be used to set the roadmap for an organization.
- Many forces impact the architecture, and some are not quantifiable (e.g., none of the frameworks have a cell that includes politics and money).
- Cultural differences between the architecture group and the process group are one of the biggest problems—often there is a lack of communication between the various groups.
- There is no official link between Capability Maturity Model® Integration (CMMI®) and architecture. The people involved are in different parts of the organization.
- Bosch's experience included a separate rollout of product line and process improvement, but there was some cross-fertilization between the two groups when architecture staff members were included in the process group and vice versa.
- The terms *software*, *system*, and *enterprise* are interrelated and will be part of any discussion on the relationship of architecture and process.
- Participants would like to see the SEI take the initiative and merge process and architecture. Industry would be willing to pick it up from there.
- Someone suggested starting with tailoring a lightweight ATAM evaluation in a small-to-medium-enterprise (SME) process.
- CMMI is more descriptive (what to do), and architecture is more prescriptive (how to do it).
- The *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems* (IEEE Standard 1471-2000) [IEEE 00] tries to link process and architecture but has not been widely adopted.

® Capability Maturity Model and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

4.3.2 Architect's Role and Authority

Participants identified the following issues regarding the architect's authority:

- The DoD has the policy of procedure. Its desire is to make systems engineering (which includes software architecture review) as important as cost and schedule. Doing so would give system/software architects more influence on process.
- Even when policies are in place, projects are schedule driven. This creates tension between the person driving the schedule and the architect.
- Architecture is part of quality and should be part of process. Currently, there is a disconnect.

4.3.3 Adoption

Participants discussed topics related to organizations adopting architecture-centric practices:

- There needs to be a simple assurance scheme that includes
  - stakeholder buy-in
  - technical feasibility
  - test cases to validate the approach
- As with physical exercise, people deny the need for adopting architecture-centric practices or desire an easy solution, and they must be educated.
- One adoption barrier is the perception that architecture-centric practices significantly inflate the time required for good system engineering practices. Someone commented that even if this perception is correct, rework is costlier.
- Adoption occurs in phases, moving from awareness to commitment to architecture improvement (e.g., documentation, evaluation, and so on).
- There needs to be a champion with some clout within the organization.

4.3.4 Standards

Finally, the following points were made regarding standards and architecture:

- Industry depends on research organizations. There is an opportunity for research organizations to take a leadership role in establishing standards.
- Establishing a standard could be stating what needs to be done without necessarily saying how to accommodate different domains with different drivers (government model).

5 Closing Session

In the closing session, Linda Northrop led participants in a discussion on emergent themes, workshop highlights, and the future of SATURN.

Participants agreed on the following themes:

- Quality attributes are key.
- Sharing tips and lessons learned applying SEI software architecture technology will advance and refine architecture-centric practices.
- There is a need for software architecture advocacy.
- There is a need for quantifying the benefits of software architecture practices.
- Architecture is about thinking and communicating.
- The role of the architect is changing.

Participants noted the following highlights:

- peer collaboration
- shared understanding
- SEI technical staff presence
- developing metrics that measure benefits
- exploring case studies that highlight how to apply architecture-centric methods
- learning what's new in software architecture
- learning about the acquisition support available for software architecture
- agreeing that software architecture technology has reached the early majority

Participants requested that the SEI do the following:

- Publish
  - the most frequent risks, non-risks, quality attributes
  - industry trends
  - checklists for domains to aid facilitators
- Develop an education program for executives.
- Be more aggressive about public relations (PR) and outreach (i.e., road show in hub cities).

- Partner with SATURN participants to write articles and make conference presentations (target the Systems & Software Technology Conference [SSTC]; Object-Oriented Programming, Systems, Language, and Applications [OOPSLA]; Software Development East and West [SD East, SD West]; UML World; Working IEEE/IFIP Conference on Software Architecture [WICSA]; EclipseCon).
- Target product developers and managers.
- Write an article for the Harvard Business Review on software architecture.
- Rebrand the ATAM and other software architecture technology (the SEI is too closely associated with process and that deters adoption).
- Offer a service to include external participants in SEI ATAM evaluations.
- Set up an SEI software architecture technology partner network.
- Influence the Software Engineering Body of Knowledge (SWEBOK) project to include software architecture technology.
- Get involved in EclipseCon and develop plug-ins for architecture support.
- Influence tool vendors.
- Have another SATURN workshop.

The SEI will take these requests under advisement but will probably not address all of them (particularly the ATAM rebranding).

Participants indicated that they could perform the following activities:

- Organize a SATURN workshop in Europe (E-SATURN).
- Organize local meetings (rings) dedicated to software architecture.
- Set up blogs.
- Set up virtual organizations.
- Create Webcasts.

The SEI hopes that participants will actively engage their colleagues in these and other software architecture outreach activities.

6 Future of SATURN

The SEI intends to make SATURN an annual event. Based on the positive feedback from the first SATURN workshop, the SEI is planning a second SATURN workshop in Pittsburgh in the spring of 2006. One participant is creating a Washington D.C. SATURN Ring with meetings beginning in September 2005. The SEI pledged to help kick off local SATURN meetings.

For more information, visit http://www.sei.cmu.edu/architecture/saturn.

References

URLs are valid as of the publication date of this document.

[Barbacci 03] Barbacci, M. R.; Ellison, R.; Lattanze, A. J.; Stafford, J. A.; Weinstock, C. B.; & Wood, W. G. *Quality Attribute Workshops (QAWs), Third Edition* (CMU/SEI-2003-TR-016, ADA418428). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. http://www.sei.cmu.edu/publications/documents/03.reports/03tr016.html

[Bass 03] Bass, L.; Clements, P.; & Kazman, R. *Software Architecture in Practice, Second Edition*. Boston, MA: Addison-Wesley, 2003.

[Clements 02] Clements, P.; Kazman, R.; & Klein, M. *Evaluating Software Architectures: Methods and Case Studies.* Boston, MA: Addison-Wesley, 2002.

[Clements 03] Clements, P.; Bachmann, F.; Bass, L.; Garlan, D.; Ivers, J.; Little, R.; Nord, R.; & Stafford, J. *Documenting Software Architectures: Views and Beyond*. Boston, MA: Addison-Wesley, 2003.

[IEEE 00] Institute of Electrical and Electronics Engineers. *Recommended Practice for Architectural Description of Software-Intensive Systems* (IEEE Standard 1471-2000). New York, NY: Institute of Electrical and Electronics Engineers, 2000.

[Kazman 02] Kazman, R.; O'Brien, L.; & Verhoef, C. *Architecture Reconstruction Guidelines, Third Edition* (CMU/SEI-2002-TR-034, ADA421612). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. http://www.sei.cmu.edu/publications/documents/02.reports/02tr034.html

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	September 2005	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Proceedings of the First Software Architecture Technology User Network (SATURN) Workshop	FA8721-05-C-0003

6. AUTHOR(S)

Robert L. Nord, Len Bass, Paul Clements, Linda Northrop, James E. Tomayko

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2005-TN-037

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

The first Carnegie Mellon® Software Engineering Institute (SEI) Software Architecture Technology User Network (SATURN) Workshop was held on April 6-7, 2005, at the SEI in Pittsburgh, Pennsylvania. Software systems engineers, architects, technical managers, and product managers exchanged best practices and lessons learned in applying SEI software architecture technology in an architecture-driven development or acquisition project. In the closing session, workshop participants noted the following highlights: peer collaboration, shared understanding, SEI technical staff presence, developing metrics that measure benefits, exploring case studies that highlight how to apply architecture-centric methods, learning what's new in software architecture, learning about the acquisition support available for software architecture, and agreeing that software architecture technology has reached the early majority.

This report describes the workshop format, discussion, and results, as well as plans for future SATURN workshops.

14. SUBJECT TERMS	15. NUMBER OF PAGES
software architecture, architecture-centric methods, architectural best practices, quality attributes	32

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL