

# **Integrating the Architecture Tradeoff Analysis Method (ATAM) with the Cost Benefit Analysis Method (CBAM)**

Robert L. Nord  
Mario R. Barbacci  
Paul Clements  
Rick Kazman  
Mark Klein  
Liam O'Brien  
James E. Tomayko

*December 2003*

**Architecture Tradeoff Analysis Initiative**

**Technical Note**  
CMU/SEI-2003-TN-038

Unlimited distribution subject to the copyright.

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.

Copyright 2004 Carnegie Mellon University.

#### NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL 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 report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number F19628-00-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

<b>Abstract.....</b>	<b>vii</b>
<b>1 Introduction .....</b>	<b>1</b>
<b>2 An Approach to Integrating the ATAM and CBAM.....</b>	<b>3</b>
<b>3 Enhancing the ATAM .....</b>	<b>5</b>
<b>4 Enhancing the CBAM .....</b>	<b>13</b>
<b>5 Reflections on Integrating the ATAM and CBAM.....</b>	<b>18</b>
<b>6 Future Work .....</b>	<b>20</b>
<b>7 Summary.....</b>	<b>21</b>
<b>References .....</b>	<b>23</b>



---

## List of Figures

Figure 1: Architecture-Centric Development Life-Cycle Activities .....	2
Figure 2: ATAM Inputs, Outputs, and Participants .....	5
Figure 3: Example Template for the Business Case Presentation [Clements 02].....	7
Figure 4: Example Template for the Architecture Presentation [Bass 03].....	8
Figure 5: Example Template for a View Packet [Clements 03].....	9
Figure 6: Example Template for Architectural Approach Analysis .....	10
Figure 7: CBAM Inputs, Outputs, and Participants .....	13



---

## List of Tables

Table 1:	Enhancing the ATAM .....	6
Table 2:	Enhancing the CBAM .....	14



---

## Abstract

The Architecture Tradeoff Analysis Initiative at the Carnegie Mellon<sup>®</sup> Software Engineering Institute (SEI) has developed a number of architecture-centric methods currently in use including the SEI<sup>SM</sup> Architecture Tradeoff Analysis Method (ATAM), the SEI Quality Attribute Workshop (QAW), the SEI Cost Benefit Analysis Method (CBAM), SEI Active Reviews for Intermediate Designs (ARID), and the SEI Attribute-Driven Design (ADD) method. Building on our success in developing and piloting a collection of software architecture methods, we're now focusing on integrating them, and building the bridges between them and the processes and architecture efforts outside the SEI, all the while continuing to refine existing methods and models.

This technical note reports on a proposal to integrate the SEI ATAM and SEI CBAM. The ATAM provides software architects with a framework for understanding the technical tradeoffs and risks they face as they make design decisions, but it does not provide any guidance for understanding economic tradeoffs. The CBAM helps software architects consider the return on investment of any architectural decision and provides guidance on the economic tradeoffs involved. The CBAM takes the architectural decision analysis done during the ATAM and helps make it part of a strategic roadmap for software design and evolution by associating priorities, costs, and benefits with architectural decisions.

---

<sup>®</sup> Carnegie Mellon is registered in the U.S. Patent and Trademark Office.  
<sup>SM</sup> SEI is a service mark of Carnegie Mellon University.



---

# 1 Introduction

The Architecture Tradeoff Analysis Initiative at the Carnegie Mellon<sup>®</sup> Software Engineering Institute (SEI) has developed a number of architecture-centric methods currently in use, of which the SEI<sup>SM</sup> Architecture Tradeoff Analysis Method (ATAM) was the first. The SEI ATAM helps a system's stakeholder community understand the consequences of architectural decisions with respect to the system's quality attribute requirements and business goals [Bass 03, Kazman 00]. The ATAM is a method that helps stakeholders ask the right questions to discover potentially problematic architectural decisions. Discovered risks can then be made the focus of mitigation activities.

As we gained experience from conducting ATAM evaluations, we developed methods to extend earlier into the software development life cycle. The SEI Quality Attribute Workshop (QAW) provides a method for eliciting quality attribute requirements [Barbacci 03]. The SEI Attribute-Driven Design (ADD) method provides an approach to defining a software architecture by basing the design process on the system's quality attribute requirements [Bachmann 00].

We also developed complementary evaluation methods. SEI Active Reviews for Intermediate Designs (ARID) [Clements 02, Clements 00] are based on the ATAM and active design reviews [Parnas 01]. The review concentrates on whether the design being proposed is suitable from the point of view of other parts of the architecture that must use it. The SEI Cost Benefit Analysis Method (CBAM) is a method for architecture-based economic analysis of software-intensive systems [Bass 03, Kazman 02]. It can be used to help the system's stakeholders choose architectural alternatives for enhancing the system, during design or maintenance phases of the software development life cycle.

Although these methods share a common heritage, set of concepts, and activities, they have not been integrated explicitly with each other or integrated into an organization's architecture-based software development life cycle. Integration of this kind is essential if organizations are to reap the total possible benefits of adopting an architecture-centric approach to software development.

We have already begun moving toward these goals. In a previous report, we showed how the architecture-centric methods can influence a wide variety of activities throughout the software development life cycle [Kazman 03]. We also showed how to distribute the activities of the methods across a generic software development life cycle. Some

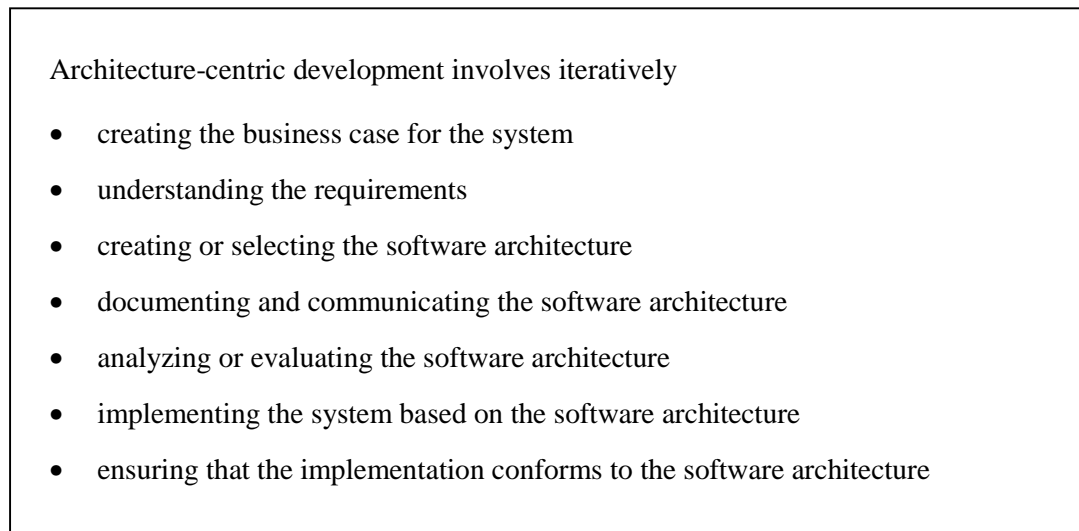
---

<sup>®</sup> Carnegie Mellon is registered in the U.S. Patent and Trademark Office.

<sup>SM</sup> SEI is a service mark of Carnegie Mellon University.

organizations have applied more than one method throughout the life cycle, but we have not documented the benefits of this integration explicitly until now. For example, in our interactions with the National Aeronautics and Space Administration's (NASA's) Earth Observing System Data Information System (EOSDIS), we integrated the ATAM and CBAM activities. We documented the results of the ATAM evaluation [Clements 02] and the CBAM evaluation [Bass 03, Kazman 02] separately.

The purpose of this report is to concentrate on the ATAM and CBAM to see how they can be enhanced and specifically integrated with each other within a software development life-cycle process to assist organizations in methodically designing, evaluating, and evolving complex software-intensive systems by making a series of business-critical architecture design decisions. Such an architecture-centric development life cycle would include activities as shown in Figure 1. The integrated ATAM/CBAM combination evaluates the software architecture and prepares for the next design iteration by considering the business goals and requirements.



*Figure 1: Architecture-Centric Development Life-Cycle Activities*

In Section 2 of this report, we propose an approach for integrating the ATAM and CBAM. In Section 3, we suggest a way of enhancing the activities of the ATAM evaluation, given that a CBAM evaluation will be subsequently performed. In Section 4, we discuss enhancing the activities of the CBAM evaluation, with the assumption that an ATAM evaluation was performed previously. In Section 5, we reflect on the wider integration issues, and in Section 6, we note opportunities for further work.

---

## 2 An Approach to Integrating the ATAM and CBAM

This report is intended to help an architect or project manager answer the question often asked at the completion of an ATAM evaluation: “Now that we’ve discovered potentially problematic architectural decisions, what do we do next?” It also helps answer related questions such as

- How do we make these discovered risks the focus of mitigation activities such as further design and analysis?
- How do we find architectural solutions to these risks more systematically?
- What are the costs and benefits of these envisioned solutions?

The ATAM provides software architects with a framework to help them understand the technical tradeoffs they face as they make design decisions, but it does not provide any guidance for understanding economic tradeoffs. The CBAM helps software architects consider the return on investment (ROI) of any architectural decision and provides guidance on the economic and project tradeoffs involved. In this report, we concentrate on enhancing an ATAM evaluation for use in conjunction with a CBAM evaluation. To this end, we draw on concepts and activities from other techniques as necessary, such as the ADD method (e.g., to help develop architectural solutions) and the “views and beyond” approach to documenting software architectures [Clements 03] (e.g., to enhance the flow of information between the ATAM and CBAM evaluations). Note that other types of risks uncovered by an ATAM evaluation and mitigation activities are not addressed in this report. For example, when requirements are not well understood, a QAW can be used to draw them out.

At the conclusion of an ATAM evaluation, the evaluation team reports the risks that were uncovered to the architecture stakeholders and the project decision makers (those empowered to speak for the development project or given the authority to mandate changes to it). The project decision makers use this information to focus on project and architectural alternatives that will help to mitigate the identified risks. However, the decision makers will need to associate costs and benefits with each alternative to decide how to spend their budget on improving the system. The CBAM is designed to complement the ATAM by providing a means of determining those costs and benefits. Together, the two methods build a value chain [Porter 85] linking business goals to quality attributes to scenarios to architectural decisions. They quantify and make explicit the effect of business goals on architectural decisions.

Knowing that we will be performing an ATAM evaluation followed immediately by a CBAM evaluation provides an opportunity for enhancing and optimizing the methods to get better results than if the two were performed serially and independently. Knowing that the CBAM

evaluation is to follow, for example, could direct the evaluation team to refine the scenarios in specialized ways during the ATAM evaluation. Refining scenarios in this context will enhance the result and optimize the time needed from the stakeholders, since the scenarios will be gathered from the stakeholders only once for the combined activities.

To better understand the points of integration between the two methods, we deconstructed the methods into activities and observed where they fit into the software development life cycle [Kazman 03].

Then, we decided to package these activities into a combined method that would provide all the benefits gained from using the ATAM and CBAM separately, and hopefully more, from the synergistic use of the two together. At the same time, our goal was to make conducting the combined approach cheaper than conducting the methods separately.

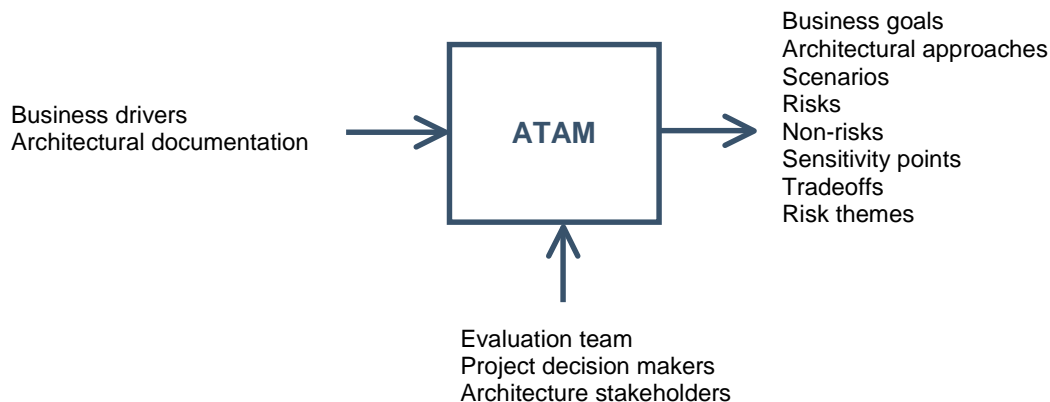
A spectrum of integration possibilities exists for integrating the methods. At one end of it, lies “fine-grained integration”—building a new method from scratch using the core concepts and activities. At the other end, lies “course-grained integration”—treating the existing methods as black boxes or groupings of prepackaged activities, and using them together in a software development life cycle.

We propose an approach closer to the coarse-grained end of the spectrum. We used the packaging of the activities into existing methods such as the ATAM and CBAM as a guide and made enhancements to the methods’ steps, while remaining true to the spirit of the original approaches. We believe that this is a natural first step, because it enables us to leverage our experience with the ATAM and the lessons we learned while transitioning the method to organizations. Refinements over the years have been based on feedback from using the method in practice, and through this iteration, we believe that the activities are now packaged in a useful and transitionable way. Rather than start from scratch, it makes sense to build on this packaging of activities. Our experience with NASA’s EOSDIS also showed that this type of method integration is useful to the customer. We will look at other integration possibilities in the future.

---

### 3 Enhancing the ATAM

Recall that the ATAM helps a system's stakeholder community understand the consequences of architectural decisions with respect to the system's quality attribute requirements and business goals. Figure 2 provides a summary of the ATAM's inputs, outputs, and participants. This figure is based on functional modeling notation where inputs flow in from the left, outputs flow out to the right, and the participants of the method are noted below [IEEE 98]. The first column of Table 1 provides a summary of the steps of an ATAM evaluation. More details about the ATAM are described by Clements, Kazman, Klein, and Bass [Clements 02, Bass 03].



*Figure 2: ATAM Inputs, Outputs, and Participants*

Assume that an organization is currently architecting a new system or evolving a system it has already fielded. The software architecture of the system is documented and supported by an architect or an architecture team. The organization needs help to understand the technical tradeoffs already made as they make further design decisions. Because these tradeoffs involve costs and benefits, as well as technical issues, we operate under the premise that the ATAM evaluation will be followed by a CBAM evaluation.

Table 1 shows the enhancements to the steps for an envisioned ATAM evaluation applied in this situation. Terms used in the table are described in the text that follows it. The reader may want to first skim the table to get an idea of the approach before reading the text and then return to the table at the end for a summary of the approach.

Our goal is to build on the success of the ATAM, and thus preserve the steps and objectives of the method, while at the same time enhancing its steps to optimize the total integration effort.

Table 1: Enhancing the ATAM

ATAM Steps	Enhancements
Step 1 – Present the ATAM	Add information such as multiple response measures for scenarios and tactics used to achieve a desired quality attribute response.
Step 2 – Present business drivers	Elicit additional information on business goals: rationale, barriers, enablers, and schedule and cost constraints.
Step 3 – Present architecture	Elicit additional information on the architecture, for example, architectural elements, relations, and their properties, and design rationale.
Step 4 – Identify architectural approaches	Cross reference approaches listed with supplementary architectural information.
Step 5 – Generate quality attribute utility tree	No enhancements are needed for this step.
Step 6 – Analyze architectural approaches	Supplement scenario refinement by eliciting worst-case and best-case response measures. Systematically elicit alternative architectural approaches for dealing with scenarios. Explicitly document suggested alternative architectural approaches for dealing with scenarios.
Step 7 – Brainstorm and prioritize scenarios	No enhancements are needed for this step.
Step 8 – Analyze architectural approaches	Supplement scenario refinement by eliciting worst-case and best-case response measures. Refine the top one-third but analyze a handful of the highest priority scenarios. Systematically elicit alternative architectural approaches for dealing with scenarios. Explicitly document suggested alternative architectural approaches for dealing with scenarios.
Step 9 – Present results	Include information about alternative architectural approaches generated during the evaluation.

All nine steps are described below, including a brief summary of the activities they involve (based on the work of Bass, Clements, and Kazman [Bass 03]) and further enhancements to them.

*Step 1 - Present the ATAM.* The evaluation leader presents the ATAM to the assembled project representatives. Using a standard presentation, the evaluation leader describes the steps and outputs of the evaluation.

The ATAM presentation is supplemented with information about the concepts and activities from the CBAM and ADD method that will be used during the ATAM evaluation, for example, information on multiple response measures for scenarios and architectural tactics. Architectural tactics are architectural decisions that are used to achieve a desired quality attribute response [Bachmann 02]. The use of activities from the ADD method is motivated by the need for filling the gap between the risks that the ATAM produces and the architectural solutions that the CBAM requires. Neither method offers guidance on how to develop such solutions.

*Step 2 - Present business drivers.* A project decision maker—usually the project manager or system’s customer—presents a system overview from a business perspective. The presenter is asked to describe the following: the system’s most important functions; any relevant technical, managerial, economic, or political constraints; the business goals and context as they relate to the development project; the major stakeholders; and the architectural drivers (that is, the major quality attribute goals that shape the architecture). A template, such as the one in Figure 3, can help the project manager prepare the presentation. The output from this step is an articulation of the business goals.

<b>Business Drivers Presentation (~ 12 slides; 45 minutes)</b>
Description of the business environment, history, market differentiators, driving requirements, stakeholders, current need, and how the proposed system will meet those needs/requirements
Description of business constraints (e.g., time to market, customer demands, standards, costs)
Description of the technical constraints (e.g., commercial off-the-shelf [COTS] products, interoperation with other systems, required hardware or software platform, reuse of legacy code)
Quality attribute requirements and the business needs from which they are derived
Glossary

*Figure 3: Example Template for the Business Case Presentation [Clements 02]*

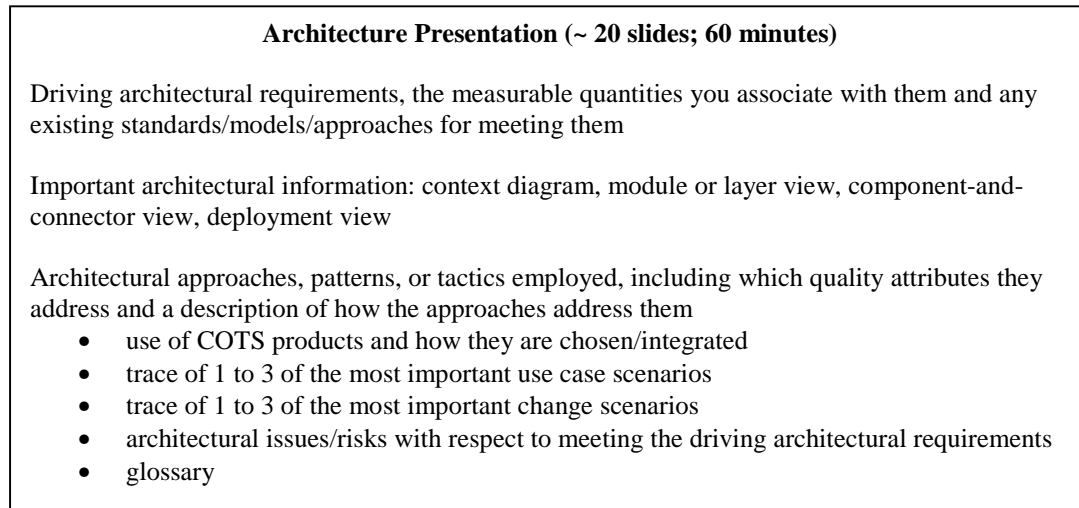
The CBAM requires additional business information to be elicited during this step, including

- the rationale for each business goal
- the dependencies among and priorities of the business goals
- barriers to adopting the business goals and strategies for overcoming them
- enablers of adopting the business goals and strategies for exploiting them
- schedule and cost constraints from the business side; for example, time to market as a business goal
- the relationship between business goals and quality goals

The more information you know about business goals, qualities, and requirements, the easier it is to focus on generating scenarios and determine their priorities in subsequent steps. Knowing the rationale will help quantify the benefit or utility of the business goals later on in the evaluation. Barriers and enablers provide a basis for understanding project risks that introduce uncertainty in the cost and benefit information that is collected during the CBAM evaluation. Cost and schedule constraints are necessary for computing the ROI.

*Step 3 - Present architecture and Step 4 - Identify architectural approaches.* The lead architect or architecture team makes a presentation describing the architecture at an appropriate level of detail. A template, such as the one in Figure 4, can help the architect

prepare for this presentation. The evaluation team identifies and catalogs the architectural patterns and approaches that are observed.



*Figure 4: Example Template for the Architecture Presentation [Bass 03]*

In the ATAM analysis steps, the architecture is used during the scenario walkthrough to identify risks. This qualitative analysis is one determining factor of the appropriate level of detail for the architecture presentation. During the CBAM evaluation, a more comprehensive analysis is needed to estimate or compute response measures that determine how well the system prescribed by the architectural design decisions performs with respect to the chosen scenarios. New architectural approaches are also elicited during the CBAM evaluation. A more in-depth understanding of the architecture may be required in these cases to determine response measures and to suggest and understand modifications to the architecture, as opposed to understanding the existing design.

Information about the views can be recorded in a view documentation package consisting of a set of view packets (such as the one in Figure 5) that are related by sibling and parent/child relationships. The information that is needed for an ATAM evaluation includes the primary presentation and context diagram, called out in the template as Sections 1 and 3. More information about what the template refers to as the element catalog and architecture background is needed for the subsequent CBAM evaluation, see the work of Clements and associates for more details [Clements 03].

### Template for a View Packet

- Section 1. Primary presentation of the view packet
- Section 2. Element catalog
  - Section 2.A Elements and their properties
  - Section 2.B Relations and their properties
  - Section 2.C Element interfaces
  - Section 2.D Element behavior
- Section 3. Context diagram
- Section 4. Variability guide
- Section 5. Architecture background
  - Section 5.A Design rationale
  - Section 5.B Analysis results
  - Section 5.C Assumptions
- Section 6. Other information
- Section 7. Related view packets

*Figure 5: Example Template for a View Packet [Clements 03]*

*Step 5 - Generate quality attribute tree.* The evaluation team works with the project decision makers to identify, prioritize, and refine the system's most important quality attribute requirements, expressed as scenarios. Scenarios are used to represent stakeholders' interests and to understand quality attribute requirements. Scenarios should cover a range of anticipated uses of (use case scenarios), anticipated changes to (growth scenarios), and unanticipated stresses to (exploratory scenarios) the system.

No enhancements are needed to this step. The desire to explore and evaluate new or alternative solutions during the CBAM evaluation that will follow places no additional constraints on the elicitation of scenarios. Changes that need to be addressed in the CBAM evaluation could arise from risks found in any scenario, or from high-priority growth scenarios.

*Step 6 - Analyze architectural approaches.* The evaluation team examines the highest ranked scenarios, one at a time. First, the chosen scenario is refined. Then, the quality attribute and business goals it supports are recorded, as well as the three parts of the scenario—stimulus, environment, and response. Next, the architect explains how the architecture supports each scenario and answers any questions the team has. During this process, the architectural decisions made to support this scenario are recorded along with any risks, non-risks, sensitivity points, and tradeoffs. This information can be recorded in a template such as the one shown in Figure 6.

ATAM Architectural Approach Analysis		
Scenario		
Business Goal(s)		
Attribute		
Attribute Concern		
Scenario Refinement	Stimulus	
	Environment	
	Response	
Architectural Decisions and Reasoning		
Risks		
Sensitivities		
Tradeoffs		
Non-risks		
Other Issues		

*Figure 6: Example Template for Architectural Approach Analysis*

The ATAM has only one notion of response measure whereas the CBAM has five: (1) worst case, (2) best case, (3) current case, (4) desired case, and (5) expected case. A response measure in an ATAM evaluation is comparable to a desired-case response measure in a CBAM evaluation. In addition to the desired-case response measure recorded during this step, information about worst-case and best-case response measures needed by the CBAM is elicited during this step.

As the architecture is evaluated, changes to the design are often considered. These can be local changes, that is, changes the architect anticipated and the architecture localizes. They might be changes that occur inside a component and therefore don't affect the architecture. Or, they might be changes that are compatible with the architectural approach, such as adding another client to an architecture that supports multiple clients and servers. Sometimes a solution required to fulfill a scenario is one that has not been anticipated and therefore requires a major restructuring of the design. These changes are architectural ones, that is, those that involve modifying the gross system topology, communication, and coordination mechanisms.

Eliciting these alternative architectural solutions is not the focus of an ATAM evaluation, but sometimes they emerge in discussion during the analysis. Analyzing a scenario may require a "what if" analysis that considers changes to the design that are not categorized as local or architectural in nature until later. If solutions do emerge, they can be recorded for use in a

subsequent CBAM evaluation that evaluates the risks uncovered during the ATAM evaluation and seeks solutions for mitigating them.

When the ATAM is used in conjunction with the CBAM, we seek to systematize the development of alternative architectural approaches that has previously been done in an ad hoc way. The CBAM depends on these architectural alternatives but doesn't offer any guidance for developing them. Concepts from the ADD method [Bachmann 00] and the work of Bachmann, Bass, and Klein in architectural tactics [Bachmann 02] can help generate and explicitly document these architectural alternatives.

The ADD method is an approach to defining a software architecture by basing the design process on the quality attribute requirements of the system. The ADD approach follows a recursive decomposition process where, at each stage in the decomposition, architectural tactics and patterns are chosen to satisfy a set of quality scenarios. Each tactic helps achieve a particular quality attribute—or more precisely, an architectural tactic is a means of satisfying a quality attribute response measure (such as average latency or mean time to failure) by manipulating some aspect of a quality attribute model (such as performance-queuing models or reliability Markov models) through architectural design decisions. The patterns that implement tactics impact other quality attributes.

*Step 7 - Brainstorm and prioritize scenarios.* The evaluation team asks the stakeholders to brainstorm scenarios that are operationally meaningful with respect to the stakeholders' individual roles.

No enhancements are needed to this step. The reasoning is the same as that for Step 5, but the number of scenarios carried forward differs. In an ATAM evaluation, the evaluation leader looks for a sharp drop-off in the number of votes and carries forward the handful of top scenarios; the CBAM calls for the top one-third to be carried forward. So, for example, if we have 60 scenarios, the top 5 might be selected in the ATAM evaluation, whereas the top 20 would be selected in the CBAM evaluation.

The ATAM and CBAM evaluations share the same stakeholders, but the CBAM analyzes a larger set of scenarios. This is true because the ATAM focuses on scenarios that represent what the architecture is supposed to do, whereas the CBAM focuses on scenarios that represent possible changes to the architecture. If the ATAM analysis part of this step determines that the architecture is risk free with respect to certain scenarios, they are removed from the list of scenarios that the CBAM evaluation considers. Note that a growth scenario represents a change in the requirements that may or may not require a change to the architecture. Changes to the architecture that need to be addressed in the CBAM evaluation could arise from risks found in any scenario.

*Step 8 - Analyze architectural approaches.* The activities are similar to those in Step 6. The ATAM evaluation team analyzes the top handful of scenarios.

While the stakeholders are convened, the desired-case, worst-case, and best-case responses are elicited for the top one-third of the scenarios in anticipation of their use in a CBAM evaluation. All other refinement and analysis activities are restricted to the top handful of scenarios normally considered in an ATAM evaluation.

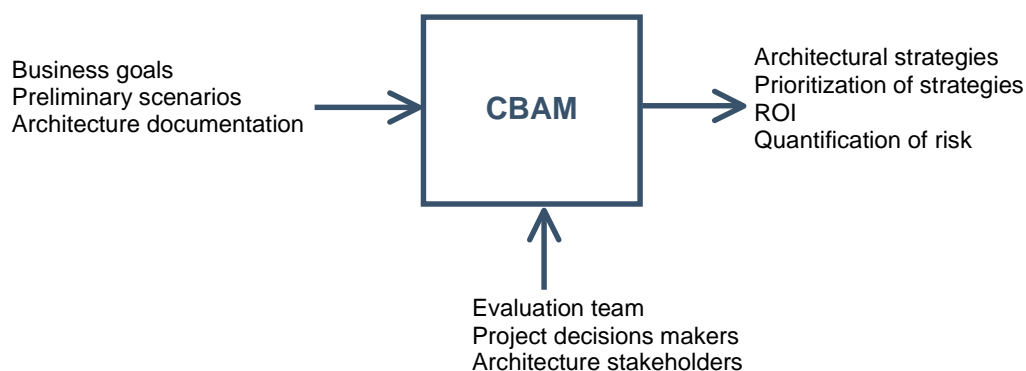
*Step 9 - Present results.* The evaluation leader recapitulates the steps of the ATAM and all the information collected in the steps. The evaluation team adds value by grouping the risks into risk themes, based on some common underlying concern.

The final presentation is supplemented with information regarding any alternative architectural approaches generated during the analysis, grouped into themes, and cross-referenced with associated risks.

---

## 4 Enhancing the CBAM

Recall that the CBAM helps the system's stakeholders choose architectural alternatives for enhancing the system. Figure 7 provides a summary of the inputs, outputs, and participants of the CBAM; the first column of Table 2 summarizes the steps of the CBAM. Bass, Clements, and Kazman describe the CBAM in detail [Bass 03].



*Figure 7: CBAM Inputs, Outputs, and Participants*

Again, assume that an organization is currently architecting a new system or evolving a system it has already fielded. The software architecture of the system has been documented and supported by an architect or an architecture team. An enhanced ATAM evaluation has been held, resulting in a set of risks that need to be addressed with architectural approaches. The organization needs help assessing the benefits of proposed architectural approaches with respect to their ROI. The organization has cost models for software development.

The architect wants to understand the economic tradeoffs in the architectural design decisions. The architecture needs to be evaluated with respect to the requirements, strategies for meeting them need to be elicited and evaluated on a technical basis, and the ROI must be computed so that the organization can decide how to spend its money wisely on system improvements. In this context, our enhanced CBAM will help evaluate the costs and benefits of the envisioned solutions.

Table 2 shows the steps for an envisioned enhanced CBAM evaluation applied in this situation. Following the table, the steps are described in more detail. Our goal is to preserve the objectives of the CBAM, while enhancing its steps to optimize the total integration effort.

Table 2: *Enhancing the CBAM*

<b>CBAM Steps</b>	<b>Enhancements</b>
Step 1 – Collate scenarios	Completed during the ATAM evaluation
Step 2 – Refine scenarios	Partially completed during the ATAM evaluation when the desired-case, best-case, and worst-case response measures for the scenarios were elicited from the stakeholders. The architect completes this step by determining the current-case response measures for the scenarios carried forward.
Step 3 – Prioritize scenarios	Enhanced business goals and traceability from scenarios to quality attributes to business goals gathered during the ATAM evaluation help guide this step.
Step 4 – Assign intra-scenario utility	Enhanced business goals gathered during the ATAM evaluation help guide this step.
Step 5 – Develop architectural strategies and determine quality attribute response levels	Some architectural approaches may have been identified during the ATAM evaluation. New ones can be added prior to or during this step, using activities from the ADD method. When the expected-case response measure is being determined, the ATAM analysis template might be used to record risks.
Step 6 – Determine the utility of the expected quality attribute response levels by interpolation	No enhancements are needed for this step.
Step 7 – Calculate the total benefit obtained from an architectural strategy	Risks from the ATAM evaluation provide input into variations in benefits.
Step 8 – Choose architectural strategies based on the ROI	Enhanced business goals about cost and schedule constraints gathered during the ATAM evaluation help guide this step.  Risks from the ATAM evaluation provide input into variations in costs.
Step 9 – Confirm results with intuition	The association of business goals to quality attributes to scenarios to architectural decisions helps confirm that the selected architectural decisions best meet the business goals.

In the description of the CBAM that follows, the steps of the method are first summarized briefly, based on how Bass, Clements, and Kazman describe the method [Bass03]. Then, enhancements to the method are described.

*Step 1 – Collate scenarios.* The stand-alone CBAM collates scenarios previously elicited and gives the stakeholders the opportunity to contribute new ones. The scenarios are prioritized based on satisfying business goals and the top one-third are chosen for further study.

This step is no longer necessary in this enhanced version of the CBAM, since the ATAM scenarios provide the desirable change scenarios necessary to begin the CBAM analysis. They have been prioritized and the top one-third identified. This step could be invoked if a

new set of stakeholders emerges or becomes available in the interim between the conclusion of the ATAM evaluation and the beginning of the CBAM evaluation.

*Step 2 – Refine scenarios.* Elicit the response measures for the scenarios from Step 1.

This step was partially completed during the ATAM evaluation when the worst-case, desired-case, and best-case response measures were elicited from the stakeholders. The architect completes this step by determining the current-case response measures for the scenarios carried forward.

Note that the software architecture may or may not be needed in this step, depending on the type of quality attribute being analyzed and whether the system has been implemented. For example, the architecture is not needed when the current-case response is determined by instrumenting the system for runtime properties. But, the architecture is needed when the current-case response is estimated by using scenario walkthroughs to assess the development-time properties such as modifiability. While a scenario walkthrough might look similar to the analysis done during an ATAM evaluation, the CBAM analysis is more rigorous. The ATAM analysis is qualitative and computes a rough order of magnitude and/or identifies sensitivity points as a means towards the end goal of identifying risks. Often, the system is not implemented, and more precise analysis is not available or even beneficial given the cost.

This step is where additional information about the architecture that comes from Steps 3 and 4 of the ATAM can be beneficial in performing a more comprehensive analysis. Also, for those scenarios that were analyzed previously during the ATAM evaluation, the architect can make use of analysis information about sensitivity points when determining current-case response measures. The documentation of sensitivity points provides traceability of scenarios to architectural decisions in the existing architecture that can facilitate the computation of the current-case response. The documentation is also useful when assessing the impact of side effects and understanding tradeoffs in later steps.

*Step 3 – Prioritize scenarios.* Prioritize the scenarios based on the desired response and carry forward the top half of them.

The enhanced business goals gathered during the ATAM evaluation help guide this step. The list of quality attributes that concern the stakeholders can be readily generated from the traceability of scenarios to qualities from the utility tree generated in the ATAM evaluation.

*Step 4 – Assign intra-scenario utility.* Determine the utility for each response measure. A utility of 100 for the best case tells us that we completely achieved the business goals; a utility of 0 for the worst case tells us that we failed to achieve any utility. The values assigned for the current and desired cases are relative to the business goals.

The outputs from the ATAM evaluation give us traceability from scenarios to quality attributes to business goals. The enhanced information about business goals (such as rationale) gathered during the ATAM evaluation help guide the assignment of utility.

*Step 5 – Develop architectural strategies and determine quality attribute response levels.*

Develop architectural approaches (or capture those already developed) that address the chosen scenarios and determine the expected-case quality attribute response measure for the relevant architectural approaches with respect to each chosen scenario.

Some of these approaches emerged during the ATAM evaluation and, if they made the cut for Step 6 or 8 of the ATAM, the results from that analysis can help facilitate the estimation of the expected-case response measure as it did for the current-case response measure in Step 2 of the CBAM. The documentation of sensitivity points provides traceability of scenarios to architectural decisions in the existing architecture that can facilitate the estimation of the expected-case response.

If additional architectural approaches are called for, the ADD method can be used as it was in the ATAM evaluation to identify candidate architectural decisions. That method offers steps to selecting architectural tactics that can be used as part of a larger architectural approach in the fulfillment of a quality attribute requirement. These tactics may also be used to validate the chosen approaches.

An architectural approach is developed with a scenario or group of scenarios in mind. Whenever a new approach is introduced, it needs to be checked for side effects (i.e., unintended effects) on existing scenarios. The sensitivity points and tradeoffs identified during the ATAM analysis can help provide traceability from the portion of the architecture affected by the posited change and the affected scenarios.

Analysis of these architectural approaches, corresponding roughly to the architectural approaches identified in the ATAM evaluation, can benefit from the ATAM's analysis of risks. As the expected case is determined, the ATAM analysis template can be employed to capture risks and sensitivity points. These risks provide input into computing variations in the costs and benefits during Steps 7 and 8 of the CBAM.

*Step 6 – Determine the utility of the expected quality attribute response levels by interpolation.* The elicited utility values from Step 4 form a utility curve with the x-axis representing the quality attribute response and the y-axis representing the utility. Using this utility curve, calculate, via interpolation, the utility of the expected quality attribute response measure for the architectural approach.

No enhancements are needed for this step.

*Step 7 – Calculate the total benefit obtained from an architectural strategy.* Subtract the utility value of the current-case response from the expected response and normalize it using the votes elicited in Step 3. Sum the benefit due to a particular approach across all scenarios and relevant quality attributes.

Risks provide input to adjusting the values of the benefits and costs (determined in the next step). A range of values is computed for the cost and benefit with minimum and maximum values computed as the end points of the confidence levels. Many of these risks will be the ones identified using the ATAM analysis template during the ATAM evaluation or during Step 5 of the CBAM. Here, we use them to quantify a range around the quality attribute response measures and utility levels. Additional properties about the risks are needed to use them in the CBAM evaluation—notably probability, impact, and mitigation strategy.

*Step 8 – Choose architectural strategies based on the ROI.* Determine the cost and schedule implications of each architectural approach. Calculate the ROI value for each approach as a ratio of benefit to cost. Rank the architectural approaches according to the ROI value, and choose the top ones until the budget or schedule is exhausted.

Risks provide input for adjusting the values of the costs and project management and schedule implications. The enhanced business goals about cost and schedule constraints gathered during the ATAM evaluation help guide this step.

Architectural approaches are chosen to address one or more of the scenarios under consideration. These approaches can have side effects on quality attributes or scenarios not previously considered.

*Step 9 – Confirm results with intuition.* For the chosen architectural approaches, consider whether they seem to align with the organization's business goals. If not, consider issues that may have been overlooked while doing this analysis.

The association of business goals to quality attributes to scenarios to architectural decisions helps confirm that the selected architectural decisions best meet the business goals.

---

## 5 Reflections on Integrating the ATAM and CBAM

To integrate the ATAM and CBAM, we used our understanding of their artifacts and activities to widen the interface between the two methods. This allows us to change when activities are performed or artifacts are refined to the most optimal times, enhancing both methods in the process. Some activities occurred earlier; for example, Step 1 and part of Step 2 of the CBAM became part of Steps 7 and 8 of the ATAM. Other activities were used later; for example, activities and the associated analysis template from Steps 6 and 8 of the ATAM became part of Step 5 of the CBAM. And we considered the context in which the activities were performed. Knowing that a CBAM evaluation is coming next and that the effect of business goals on architectural decisions needs to be made explicit provides a further goal for directing the elicitation of ATAM artifacts such as business goals and architectural documentation. It also provides direction for analysis results and obtaining priorities, benefits, and utility.

This integrated approach offers the following benefits:

- It provides both technical and economic analysis of a software architecture. At the conclusion of the ATAM evaluation, further analysis is needed to consider costs and benefits. Risks need to be quantified in terms of probability, impact, and mitigating strategies. A CBAM evaluation provides a natural follow-on to an ATAM evaluation for those wanting an integrated approach.
- It optimizes the process. The results from the ATAM evaluation can be fed into the CBAM evaluation and don't have to be duplicated. Time is saved in the CBAM evaluation by performing some activities during the ATAM evaluation. For example, Step 1 and part of Step 2 of the CBAM can be eliminated, which also means that the stakeholders are convened less often. Part of Step 5 may have been completed if the ATAM evaluation produces architectural approaches for the scenarios. The enhanced business goals and traceability to quality attributes to scenarios to architectural decisions from the ATAM evaluation help guide and optimize several steps in the CBAM evaluation. Risks from the ATAM evaluation provide input into variations in costs and benefits. It is expected that the two days required to perform a basic stand-alone CBAM evaluation can be reduced to a day in this integrated context. Note that the time required to identify architectural approaches and their associated costs is variable and factored out. The CBAM offers guidance in computing benefits but assumes that techniques and knowledge exist to elicit architectural approaches and their associated costs, and that they can be obtained expeditiously.

The resources for conducting the ATAM evaluation remain essentially the same. Additional effort is expended to supplement presentations and elicit information about

quality attributes and the software architecture, to document the architectural approaches that emerge during analysis, and to refine scenarios. Note that documenting architectural approaches does not introduce an additional design activity in the ATAM evaluation; its purpose is to explicitly document design decisions that are discussed informally during the analysis steps. Most of the extra time needed is for refining the top one-third of the scenarios, which will add an extra hour or two. We still complete the ATAM evaluation in its allotted time of 1 – 1 ½ days for Phase 1 and 2 – 2 ½ days for Phase 2.

- It improves the quality of the results. The CBAM's initial steps take place in the context of the ATAM Phase 2 during which the business goals and architecture are reviewed. Presenting the business goals is helpful prior to and during scenario brainstorming and prioritization, especially as we quantify how those goals affect the architectural decisions.

Using activities from the ADD method can help provide guidance in eliciting the architectural approaches posited during the evaluation more systematically. The integrated approach more clearly differentiates the analysis of existing architecture versus the development and analysis of new architectural approaches. It makes explicit the documentation needed to record new architectural decisions.

Using activities and artifacts from the ATAM evaluation in the CBAM evaluation improves the quality of the computed results. Using the analysis template during Step 5 of the CBAM evaluation provides information about risks that can be used to compute more accurate costs and benefits.

---

## 6 Future Work

Further work remains to be done to verify the benefits of our proposed approach through pilot projects with customers. The goal of such projects is to provide tailored architecture methods as a means of helping customers add architecture-based and quality-attribute-based thinking to their planning and development efforts.

Further work remains in improving techniques such as capturing and organizing business goals; associating costs with architectural decisions; integrating concepts into the software development life cycle; and identifying further points of integration.

The ATAM and CBAM use different approaches to elicit scenario response measures. The ATAM has one voting scheme where stakeholders vote on scenarios based on a single response measure. The CBAM uses a two-part voting scheme. The first part is similar to how, in the ATAM, stakeholders vote based on the desired response value and how well it satisfies the business goals of the system. In the second part, stakeholders vote on scenarios again based on their knowledge of the spread between desired- and current-case response measures. The CBAM's approach is more thorough and, hence, more costly. Should this approach apply to all scenarios or just those that "make the cut"? Perhaps the CBAM's approach would help make the cut in a more intelligent way. We won't know for sure without further thought and experience.

Combining the two analysis methods and adding concepts from the "views and beyond" approach to documenting software architecture and the ADD method force us to rethink the staging of the combined methods. For example, often, scenarios are proposed for which no existing architectural approach suffices. In that case, the architect needs to spend time (off-line) to design an approach. This new approach would ideally be analyzed with ATAM-like techniques. Also, side effects may need to be considered as a consequence of these new architectural approaches, and the scenarios involving those side effects may not be the top-rated ones. So, the steps start to spread out, and to be applied incrementally and iteratively during the software development life cycle.

The CBAM provides an opportunity (and the need) for cost modeling. Right now, we have no specific advice to give on this point. We should look at tailoring the constructive cost model COCOMO II or a similar technique so that we can advise projects that do this cost modeling.

---

## 7 Summary

With the advent of cost-effective, repeatable, architecture evaluation methods such as the ATAM, architecture evaluation is becoming a standard part of architecture-centric development. Being part of the life cycle of such a development provides an opportunity to integrate the ATAM with other methods, such as the CBAM, that add value to the ATAM's outputs. Quality attributes and architecture documentation based on views play a unifying role in describing the activities and artifacts of the architecture-centric methods and help us in this integration effort. However, additional work is required to integrate the methods with each other and into an organization's software development life cycle.

This technical note reports on a proposal to integrate the ATAM and the CBAM. The CBAM takes the architectural decision analysis done during the ATAM and helps make it part of a strategic roadmap for software design and evolution by associating priorities, costs, and benefits with architectural decisions.

The integration of different methods and techniques with each other or with other development life cycles is possible and will be addressed in future reports. Investigating such potential combinations is part of a larger effort to understand how to

- integrate the approaches
- package the methods' activities into the software development life cycles of typical organizations
- understand the appropriate fit with other architectural processes and technologies
- connect with other business, management and/or acquisition processes that can help enforce software architecture practices throughout the life cycle



---

## References

*URLs valid as of the publication date of this document*

- [Bachmann 00]** Bachmann, F.; Bass, L.; Chastek, G.; Donohoe, P.; & Peruzzi, F. *The Architecture-Based Design Method* (CMU/SEI-2000-TR-001, ADA375851). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. <<http://www.sei.cmu.edu/publications/documents/00.reports/00tr001.html>>.
- [Bachmann 02]** Bachmann, F.; Bass, L.; & Klein, M. *Illuminating the Fundamental Contributors to Software Architecture Quality* (CMU/SEI-2002-TR-025, ADA407778). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. <<http://www.sei.cmu.edu/publications/documents/02.reports/02tr025.html>>.
- [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). 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, 2nd edition*. Boston, MA: Addison-Wesley, 2003.
- [Clements 00]** Clements, P. *Active Reviews for Intermediate Designs* (CMU/SEI-2000-TN-009, ADA383775). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. <<http://www.sei.cmu.edu/publications/documents/00.reports/00tn009.html>>.
- [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 98]** Institute of Electrical and Electronics Engineers. *IEEE Standard for Functional Modeling Language* (IEEE Std 1320.1-1998). New York, NY: IEEE Computer Society, 1998 (ISBN 0-738-10340-3).
- [Kazman 00]** Kazman, R.; Klein, M.; & Clements, P. *ATAM: Method for Architecture Evaluation* (CMU/SEI-2000-TR-004, ADA382629). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. <<http://www.sei.cmu.edu/publications/documents/00.reports/00tr004.html>>.
- [Kazman 02]** Kazman, R.; Asundi, J.; & Klein, M. *Making Architecture Design Decisions: An Economic Approach* (CMU/SEI-2002-TR-035, ADA408740). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. <<http://www.sei.cmu.edu/publications/documents/02.reports/02tr035.html>>.
- [Kazman 03]** Kazman, R.; Nord, R. L.; & Klein, M. *A Life-Cycle View of Architecture Analysis and Design Methods* (CMU/SEI-2003-TN-026). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. <<http://www.sei.cmu.edu/publications/documents/03.reports/03tn026.html>>.
- [Parnas 01]** Parnas, D. & Weiss, D. "Active Design Reviews," Ch. 17, 337-351. *Software Fundamentals: Collected Papers by David L. Parnas*. Boston, MA: Addison-Wesley, 2001.
- [Porter 85]** Porter, M. *Competitive Advantage: Creating and Sustaining Superior Performance*. New York, NY: Free Press, 1985.

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
Public reporting burden for this 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, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE December 2003	3. REPORT TYPE AND DATES COVERED Final		
4. TITLE AND SUBTITLE Integrating the Architecture Tradeoff Analysis Method (ATAM) with the Cost Benefit Analysis Method (CBAM)		5. FUNDING NUMBERS F19628-00-C-0003		
6. AUTHOR(S) Robert L. Nord, Mario R. Barbacci, Paul Clements, Rick Kazman, Mark Klein, Liam O'Brien, James E. Tomayko				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2003-TN-038		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116		10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS		12B DISTRIBUTION CODE		
13. ABSTRACT (MAXIMUM 200 WORDS)  The Architecture Tradeoff Analysis Initiative at the Carnegie Mellon® Software Engineering Institute (SEI) has developed a number of architecture-centric methods currently in use including the SEI <sup>SM</sup> Architecture Tradeoff Analysis Method (ATAM), the SEI Quality Attribute Workshop (QAW), the SEI Cost Benefit Analysis Method (CBAM), SEI Active Reviews for Intermediate Designs (ARID), and the SEI Attribute-Driven Design (ADD) method. Building on our success in developing and piloting a collection of software architecture methods, we're now focusing on integrating them, and building the bridges between them and the processes and architecture efforts outside the SEI, all the while continuing to refine existing methods and models.  This technical note reports on a proposal to integrate the SEI ATAM and SEI CBAM. The ATAM provides software architects with a framework for understanding the technical tradeoffs and risks they face as they make design decisions, but it does not provide any guidance for understanding economic tradeoffs. The CBAM helps software architects consider the return on investment of any architectural decision and provides guidance on the economic tradeoffs involved. The CBAM takes the architectural decision analysis done during the ATAM and helps make it part of a strategic roadmap for software design and evolution by associating priorities, costs, and benefits with architectural decisions.				
14. SUBJECT TERMS software architecture, architecture analysis, architecture evaluation, economic modeling, architecture-centric, software development life cycle		15. NUMBER OF PAGES 34		
16. PRICE CODE				
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	