Carnegie Mellon
Software Engineering Institute

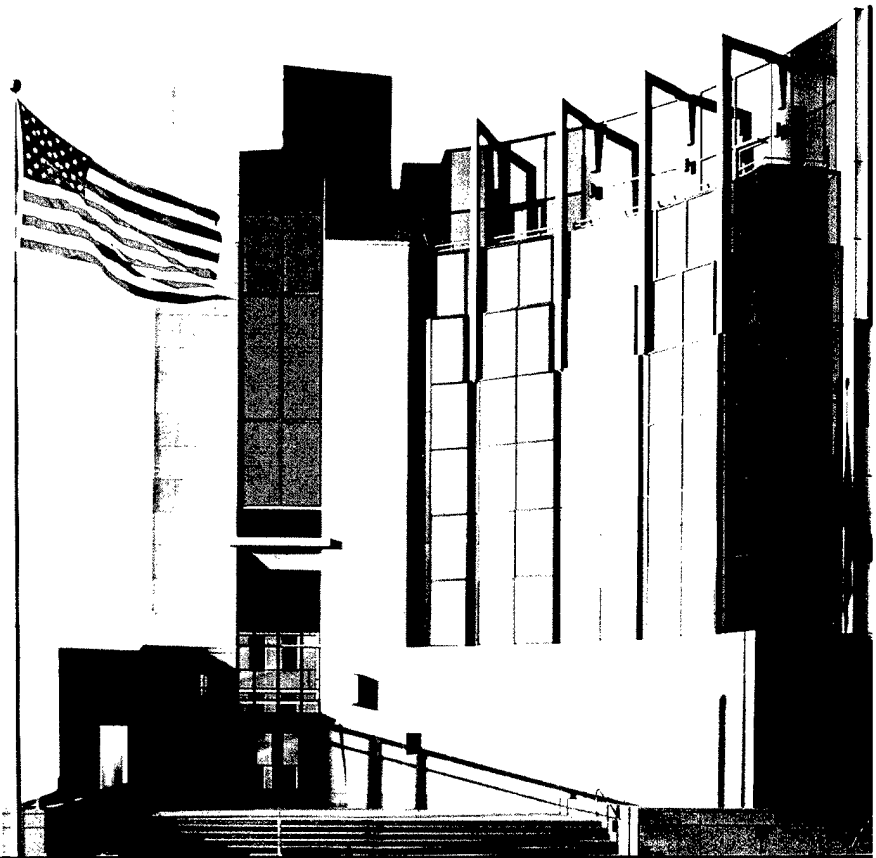
Applicability of General Scenarios to the Architecture Tradeoff Analysis Method[SM]

Len Bass
Mark Klein
Gabriel Moreno*

October 2001

* Gabriel Moreno is sponsored by YPF Foundation and the Fulbright program of the U.S. Department of State.

TECHNICAL REPORT
CMU/SEI-2001-TR-014
ESC-TR-2001-014



Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment, or administration of its programs or activities on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state, or local laws or executive orders. However, in the judgment of the Carnegie Mellon Human Relations Commission, the Department of Defense policy of "Don't ask, don't tell, don't pursue" excludes openly gay, lesbian and bisexual students from receiving ROTC scholarships or serving in the military. Nevertheless, all ROTC classes at Carnegie Mellon University are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2056.

Obtain general information about Carnegie Mellon University by calling (412) 268-2000.

Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

Applicability of General Scenarios to the Architecture Tradeoff Analysis Method[SM]

CMU/SEI-2001-TR-014
ESC-TR-2001-014

Len Bass
Mark Klein
Gabriel Moreno, YPF Foundation

October 2001

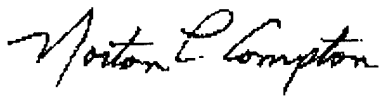
Architecture Tradeoff Analysis Initiative

This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER



Norton L. Compton, Lt Col., USAF
SEI Joint Program Office

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 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).

printed 10/12/01 8:43 AM v2.0 / bw4le

Table of Contents

List of Figures

List of Tables

Abstract

The SEI has been developing a list of scenarios to characterize quality attributes. The SEI has also been conducting Architecture Tradeoff Analysis Method[SM] (ATAM[SM]) evaluations. One output of an ATAM evaluation is a collection of scenarios that relate to quality attribute requirements for the specific system being evaluated. In this report, we compare the scenarios elicited from five ATAM evaluations with the scenarios used to characterize the quality attributes. This effort was designed to validate the coverage of the existing set of general scenarios and to analyze trends in the risks uncovered in ATAM reports.

[SM] Architecture Tradeoff Analysis Method and ATAM are service marks of Carnegie Mellon University.

1 Introduction

The Architecture Tradeoff Analysis Method[SM] (ATAM[SM]) evaluates software architecture in light of quality attributes [Kazman 00]. The goal is to understand the tradeoffs between attributes and uncover risks that may prevent the architecture from achieving its quality goals. Twenty pilot ATAM evaluations have been done, contributing to its maturation. In a separate effort, we are characterizing quality attributes in terms of general scenarios [Bass 00].

This study was intended to gauge the progress of both efforts by

- validating the coverage of the existing collection of general scenarios. We provide an overview of ATAM scenarios and their mapping to general scenarios in Section 2. We provide the details of the coverage in Appendix A.
- identifying patterns in the scenarios generated during ATAM evaluations. We discuss patterns of the scenarios in Section 2.3.
- identifying patterns in the risks documented in the ATAM reports. We discuss risk patterns in Section 3.
- testing the hypothesis that, during brainstorming sessions, scenarios are generated in order of importance. We discuss the results of scenario generation sequence in Section 4.

This study is based on five ATAM evaluations. These evaluations provided a sample of actual scenarios. They also served as an early test bench for analyzing the applicability of general scenarios to the ATAM. We have not verified our results by comparing them with scenarios generated in the remaining 15 ATAM evaluations.

[SM] Architecture Tradeoff Analysis Method and ATAM are service marks of Carnegie Mellon University.

2 Scenarios

In addition to satisfying the functional requirements of a system, software developers have to address its quality attributes such as performance, modifiability, availability, and usability. [Barbacci 95]. The ability of a system to meet these quality attributes is largely determined by its architecture; therefore, it is very important to understand the relationship between software architecture and quality attributes [Bass 98].

We use the concept of a *general scenario* to describe what achieving a quality attribute goal means [Bass 00]. General scenarios describe how the architecture should respond to a certain stimulus. The following is an example of an availability general scenario:

> A failure occurs and the system notifies the user; the system may continue to perform in a degraded manner.

One step of an ATAM evaluation is generating scenarios for specific quality attributes in the system under evaluation. These specific scenarios should be instances of the general scenarios we have enumerated. Therefore, it is important that the collection of general scenarios cover the specific scenarios that we developed.

In this study, we used the list of general scenarios current as of February 2001. In Appendix B, we present both the current list of general scenarios and the list as of February. To determine list coverage, we analyzed "real world" scenarios elicited during five ATAM evaluations and presented in the final ATAM reports. We chose these reports because of their availability and because they represented a variety of domains, including large financial systems, driver information systems, engine controllers, large scientific data set management, and battlefield management.

Next, we mapped the scenarios found in the five ATAM reports to the general scenarios in our list. We performed additional scenario analyses and analyzed the risks captured in the ATAM evaluations. (Appendix A presents the scenarios and their dispositions, and provides the raw data that is discussed in the remainder of this report.)

The five ATAM reports contained a total of 170 specific scenarios. When we attempted to match each particular scenario with its general counterpart, we found that four outcomes were possible:

1. There was a general scenario that covered the ATAM scenario.
2. A general scenario partially covered the ATAM scenario; slightly modifying the wording of the general scenario would completely cover the ATAM scenario.
3. The ATAM scenario was meaningful, but no general scenario covered the ATAM scenario.
4. The ATAM scenario was not meaningful.

Of the 170 specific scenarios, 125 scenarios could be mapped directly onto corresponding general scenarios. Eleven additional scenarios could be mapped by slightly modifying the general scenario. Thirteen specific scenarios could not be categorized with the existing set of general scenarios. Twenty-one of the specific scenarios were malformed to the extent that they lacked the information needed to utilize them in this study (see Section 2.3.3). The collection of general scenarios (either as written or with slight modifications) covered 91% of the 149 specific scenarios that could be examined.

Because ATAM reports are confidential and contain proprietary information, the identity of organizations or software systems cannot be disclosed. Hence, throughout the remainder of this report, the ATAM reports are referred to as reports A, B, C, D, and E. The identities contained in the reports have been disguised according to the following pattern:

- *System A*: the software being evaluated in report A
- *Subsystem A-B*: subsystem B of the software being evaluated in report A
- *Company A*: the organization developing software A
- *COTS-A-B*: commercial off-the-shelf (COTS) component B used in system A
- *Plant A-B*: plant B of organization A

The remainder of Section 2 details the results of the study. It also provides successful instances of general scenarios, suggests improvements to the overall collection of general scenarios, and illustrates problems with some of the scenarios elicited in the ATAM evaluations.

2.1 Overall Summary of Scenarios

Table 1 gives the overall summary of scenarios. For each ATAM report, the table gives the number of scenarios considered, the number successfully mapped to general scenarios, the number not able to be mapped, and the number of scenarios that were unusable.

Table 1: Summary of Scenarios from Various ATAM Evaluations

Report	A	B	C	D	E
Elicited scenarios	24	35	15	70	26
Mapped to Modifiability general scenarios	4	20	5	28	17
Mapped to Perform-ance general scenarios	7	1	3	7	0
Mapped to Usability general scenarios	3	2	3	20	0
Mapped to Availability general scenarios	6	5	1	2	0
Mapped to Security general scenarios	0	0	0	2	0
Not currently categorized	4	1	0	3	5
Malformed	0	6	3	8	4

2.2 Successful Instantiation of General Scenarios

This section presents instances of the ATAM scenarios and how they are mapped to the general scenarios. Only one example for each quality attribute is presented here. The remainder of the ATAM scenarios together with their mapping can be found in Appendix A. The total list of general scenarios is presented in Appendix B. For each example, we first show the unedited ATAM scenario and then present the corresponding general scenario.

An examination of Table 1 shows that 91% of ATAM scenarios are instances of the general scenarios current as of February 2001. This demonstrates that the coverage of our general

scenarios is quite good. On the other hand, a general scenario represents an abstraction of a variety of specific scenarios. By performing an abstraction, some details are lost. Although this abstraction is necessary to cover all the possible scenarios, it has yet to be determined whether this is the right level of abstraction. During an ATAM evaluation, the general scenarios must be mapped into ATAM scenarios, and we do not know yet if the general scenarios are too abstract to map easily into specific scenarios.

2.2.1 Availability

Instance

121. Fixed scene orders for electronic FTP push to a site whose FTP server is down, system suspends within 10 minutes of first failed request and all resources are available while requests are suspended. Distribution to others is not impacted.

General scenario

A failure occurs and the system notifies the user; the system may continue to perform in a degraded manner.

2.2.2 Modifiability

Instance

18. New reporting requirement arrives that requires modification to metadata. Affects Subsystem D-A to Subsystem D-B translation.

General scenario

A request arrives to change the functionality of the system. The change can be to add new functionality, to modify existing functionality, or to delete functionality.

2.2.3 Performance

Instance

83. Turn the car right and the map display should turn so that the current heading of the car is at the top of the map.

General scenario

An event is initiated with resource demands specified and the event must be completed within a time interval.

2.2.4 Security

Instance

135. Unauthorized intrusion is resisted.

General scenario

The system discloses information only to authorized people.

2.2.5 Usability

Instance

114. A domain knowledgeable user can reach proficiency for core functions in a week.

General scenario

System designers should strive to make upgrades and transitions occur as smoothly as possible. Response measures include number of errors made by a new user familiar with prior releases or other members of the product line.

2.3 Scenarios with Problems

We found several problems while trying to map particular scenarios to general scenarios. In some cases, there was no general scenario to address one of the elicited scenarios. This suggested adding a new general scenario to the collection. In other situations, some minor changes to existing general scenarios were suggested to make them a better fit. Nevertheless, most of the problems found were not due to the general scenarios but to the elicited scenarios. (Section 2.3.3 analyzes those malformed scenarios in detail.)

2.3.1 Scenarios that Suggest the Need for New General Scenarios

The following subsections present instances that could not be mapped to the existing general scenarios. There were 13 such scenarios out of the 170 scenarios in the five ATAM reports. They fit into several categories:

Deployment

143. Want to distribute set of changes to a set of clients consistently (forms and configurations).

This scenario describes the deployment of a modified component. The current modifiability general scenario does not address deployment.

Locating Errors

72. Company D's client encounters a search response problem, finds/fixes problem, issues new release.

This scenario is not only a deployment scenario, as above, but also a scenario about the amount of effort required to find or fix an error.

Cost/Effort Estimation

145. Injection Prototype: Deliver a prototype and a cost estimation for a new injection timing for a new engine. There is no experience yet at Company E. Make a feasibility study to acquire a new customer project in two months.

These scenarios involve estimating development cost.

Personalization

146. Single driver personalization: Support calibration by the driver/end user to adjust the engine for more power or fuel efficiency. Personalize the car for one driver.

147. Personalize the car for more than one driver.

141. Same information presented to user, but different presentation (location, fonts, sizes, colors, etc.)

These scenarios are usability scenarios dealing with personalization. There is no general scenario that applies.

Safety

148. Safety of customer-delivered software: Ensure the safety of the system with customer-supplied object code to control the throttle control.

This scenario is about safety. There are no general scenarios about safety.

2.3.2 Scenarios that Suggest Modifications to Existing General Scenarios

In the following examples, we suggest minor modifications to general scenarios so that ATAM scenarios can be mapped to them. There were 11 ATAM scenarios that could be mapped with these minor modifications.

Usability

118. Form-to-form response (a user finishing one form to using the next one) in a single MOU environment and the system responds to the movement within one second.

This scenario can be characterized as a usability scenario similar to the *pace tolerance* scenario. However, the pace tolerance general scenario only accounts for a system that makes the user go faster than he/she can. In this instance, the system must not slow the user down. Therefore, the existing pace tolerance general scenario should be modified to address this situation. This suggested modification is shown in italics:

> Pace tolerance. A system might not accommodate a user's pace in performing an operation. This perceived systemic "impatience" may make the user feel hurried or frustrated. For example, ATMs often beep incessantly when a user "fails" to insert a payment quickly enough. *On the other hand, the system should not prevent the users from performing an operation as fast as they can.* Systems should account for human needs and capabilities when pacing the stages in an interaction. Systems should also allow users to adjust this pace as needed. Response measures include user satisfaction measurement to determine whether the system imposes an uncomfortable pace on the user.

Modifiability

9. What happens if the customer requests a reduced functionality configuration that includes communications, VMF, subset of the protocols, subset of the missions, etc. Full ballistics kernel. Subset of screens, subset of training?

This scenario can be considered an instance of the following modifiability scenario (again with the suggested changes in italics):

> A request arrives to change the functionality of the system. The change can be to add, modify, delete, *or to vary* existing functionality.

2.3.3 Malformed Scenarios

To be well formed, scenarios must clearly state the stimulus, the environmental conditions, and the measurable or observable response to the stimulus [Kazman 00]. Scenarios that do not satisfy this requirement are malformed. Now that we have introduced this distinction, we must note that several scenarios used as examples in Section 2.1 are in fact malformed scenarios. Even so, they have been used as examples because the desired response was evident from either the scenario itself or its context in the ATAM evaluation report.

The presence of malformed scenarios is not alarming. Scenarios are generated during a brainstorming session, and malformed scenarios can usually be understood from the context without rewording them so that they become well formed. If ATAM participants choose the scenarios for analysis, chances are, they will refine them so at least the stimulus is more precise. Still, increasing the precision of the scenarios will increase the likelihood that all stakeholders will understand them in the same fashion during prioritization.

Anti-Scenarios

Among the malformed scenarios, there were several instances of "anti-scenarios." In these situations, the stakeholders worded the scenario in a way such that the response is not what is really expected from the system. In general, a well-formed scenario has the following structure:

The system receives a stimulus and *some desirable response is observed.*

stimulus **response**

Consider the following scenario extracted from one of the reports:

> 91. Accounting has requirement to close financial statements within two business days, and system cannot respond.

If we analyze this scenario following the previous pattern, we might understand that the expected response to "Accounting has requirement to close financial statements within two business days" is that "the system cannot respond." Obviously, this is not what the stakeholders want. A person may propose a scenario like this for two reasons:

1. He/she thinks that there is a possibility that this scenario will take place, and wants the architect to show how the system could be modified to overcome the problem.
2. He/she wants to be sure that this scenario will never take place, and wants to see how the architect has addressed the situation.

In the first case, the stakeholder wants to resolve "accounting has to close financial statements within two business days, and system cannot respond." In this instance, "system cannot respond" is part of the stimuli and the response is missing. The solution is to convert the malformed scenario into a modifiability scenario in which the response part is how the system can be modified to solve the problem. For example, the repaired scenario could be the following:

> Accounting has requirement to close financial statements within two business days, and system cannot respond. *The system can be modified to improve its performance so that it can meet the deadline.*

In the second case, the stakeholder wants the system to avoid the scenario. This type of "anti-scenario" can be easily converted into a well-formed scenario by logically negating the response. The equivalent well-formed scenario in this case is the following:

> Accounting has requirement to close financial statements within two business days, and system *can* respond.

In general, a scenario of the form

<stimulus> not <response>

should be converted into one of these two forms:

1. a modifiability scenario where the original <stimulus> not <response> is the stimulus and a new response is identified: <<stimulus> not <response>> <new response>
2. a straightforward scenario of the form: <stimulus> <*not* response>

ATAM evaluators should be aware of anti-scenarios to convert them to a straightforward form during the course of an evaluation. In the example cited, it is obvious that the stakeholders do not want a system that cannot respond. Nevertheless, there may be other cases in which the intention of the scenario is not so obvious. Consider this scenario that a stakeholder might propose in the context of a house with a security system that controls its doors and windows [Bachmann 00]:

> An intrusion is detected, and the system cannot lock the doors.

This anti-scenario may be interpreted in different ways. The stakeholder that proposed the scenario may be looking for some solution to the fact that "An intrusion is detected, and the system cannot lock the doors." Therefore, the response is missing. The rephrased scenario could be:

> An intrusion is detected, and the system cannot lock the doors. The system activates the electromagnetic fence so that the intruder cannot escape.

Alternatively, a stakeholder may think that it should never be the case that the system cannot lock the doors upon an intrusion. The anti-scenario can be rephrased to reflect that:

> An intrusion is detected and the system *can* lock the doors.

A third stakeholder may take the original anti-scenario as a well-formed one, and think that in the event of an intrusion, the house shall not lock the doors because that would prevent the inhabitants of the house from running away from the armed intruders.

This example stresses the importance of converting anti-scenarios into well-formed scenarios before the stakeholders vote on them.

Questions Instead of Responses

Out of 170 scenarios found in the reports, 12% were in the form of a question. These can be divided into three categories according to the intent of the question. The three categories are (1) missing stimulus, (2) missing response, or (3) exploratory.

1. Missing stimulus:

   > What happens when a backup takes over for FDC?

In this scenario, the stakeholder wants to know how a desirable response is realized by the architecture. In this example, "a backup takes over for FDC" is the expected response and the stimulus is missing.

2. Missing response:

    97. What happens if the database engine crashes?

    In this case, "the database engine crashes" is the stimulus but the response is missing.

3. Exploratory:

    63. Company D's client purchases company three times its size. What are the implications including database partitioning?

    55. The current single processor system changes to a two-processor system for performance reasons. How can we handle the load balance between the two processors? Is dynamic scheduling needed?

3 Generalization Of Risks

We now move from analyzing the scenarios within the ATAM reports to analyzing risks found. One of the main purposes of an ATAM evaluation is to identify and document risks that may jeopardize achieving a quality goal [Kazman 00]. In this study, we went a step further and tried to find common risk trends across all the ATAM reports analyzed. This was beneficial for two reasons: First, the results can help ATAM evaluators identify risks that are likely to be found in other projects. Second, in other work, we are attempting to couple general scenarios with a collection of *attribute primitives* [Bass 00]. Examining risks in the light of these attribute primitives will help identify the contents necessary in an attribute primitive write-up.

The following categories of risks were identified across the reports:

- risks due to unknowns
- risks due to side effects of architectural decisions
- risks due to improper architectural decisions
- risks due to interaction with other organizations

The following subsections describe and provide examples of these categories.

3.1 Risks Due to Unknowns

Many of the risks identified by the ATAM evaluations were a consequence of something being unknown about the software, the hardware, or some commercial off-the-shelf (COTS) component. We found several causes for those items being unknown: (a) something was not specified in the system requirements, (b) the hardware was not available, or (c) the development team had not yet addressed the issue when the ATAM evaluation was performed.

Examples of risks due to unknowns follow:

Report B

In addition to the risks already identified, it should be noted that Windows NT presents an unknown with respect to performance and reliability. In the System B context, the performance of NT is not likely to be an issue, but reliability will. The mechanism for data distribution is not currently fully defined.

Report C

Necessary hardware has not yet been delivered and there are no reasonable estimates about its performance. This introduces a risk that the system will be unable to meet its performance requirements.

Since the hardware for the map display has yet to be determined, it is not known whether a graphic accelerator is needed or whether the hardware is adequate to draw the display in real time. This is a performance risk.

Insufficient thought has been given to the runtime detection of errors. This is a reliability risk.

Some investigation (prototyping or modeling) needs to be done for the startup routine. This is a performance risk.

The Blue Tooth specification has not been examined in depth. It is not clear how to connect the system to laptops, hand-held devices, etc. The use of Blue Tooth as another front end type will simplify the creation of applications. This is a modifiability risk.

There is a strong dependence on the certification process for security (e.g., to catch possible Trojan Horses) and no thought has yet been given to the certification process. This is a reliability risk.

Report D

Final performance testing is not yet completed.

Report E

Don't know if architecture layers are separated at the right levels. This may not support markets.

It's not clear if the layers in the architecture are partitioned correctly.

3.2 Risks Due to Side Effects of Architectural Decisions

In the ATAM reports analyzed, appropriate architectural decisions sometimes had side effects that subsequently introduced new risks. In the following examples, the side effects of architectural decisions were not properly considered.

137. Deploy 5B version of the Subsystem A-A and update engineering science data types (ESDT) and latitude-longitude box (LLBox) support into A5 baseline in less than eight hours with no impact on other subsystems or search, browse, and order availability.

Architectural decision 1: Backwards compatibility of interface

The risk identified by the ATAM was

> Not using infrastructure capability to "sign" an interface (that is, can ensure syntactic, but not semantic compatibility). Consequence: Interface may be syntactically compatible but semantically incompatible, and system won't catch this. Could result in incorrect results or failure.

The following scenario and the architectural decision show another example of a risk due to a side effect:

70. Five times improvement for search response times.

Architectural decision 1: Subsystem A-A as primary client

The risk identified by the ATAM was

> Different formats add data models for Subsystems A-A and Subsystems A-B. Consequence: Require translation between the two, which results in delayed response to user.

In this case, a side effect of the decision to use the EDG as a primary client was a performance cost of translation.

3.3 Risks Due to Ignoring Architectural Solutions to Attribute Requirements

While analyzing the ATAM reports, we came across several situations in which making appropriate architectural decisions could have avoided risks.

> 129. Search, browse, and order submission unavailable no more that one hour/week.

Architectural decision 1: Single online copy of DB

This decision implies not using replication. The following risks are a consequence of not having used this primitive:

> Single copy of DB online. Consequence: All activity on DB is halted for DB consistency check and upgrade. Can back up DB without halting activity. Consequence: Online DB performance is decreased (>50%) and backup takes longer.

If replication had been used, the backups and consistency checks could have been performed on one of the replicas without affecting access to the others.

> 58. A CD player not in use is removed from the system. Delete all traces of the CD.

The following are two of the risks identified by ATAM for this scenario:

> The configuration manager keeps track of which components are used and the dependencies on them, but there is no mechanism to track multiple usages of a single component. This introduces a reliability risk of incorrect behavior when a component is removed.

Multiple applications using the same data item (such as a CD profile item) may cause a problem when de-installing software since there currently is no tracking of use of data items with the configuration manager. This is a reliability risk (if the data item is mistakenly deleted) or a performance risk (if the data item is not deleted).

In this case, several users may require the same data, and the system does not track multiple uses. An architectural decision that enabled the data producer to remain ignorant of data consumers but record usage counts would have prevented this risk.

3.4 Risks Due to Interaction with Other Organizations

Several risks found in this study stem from organizations that provide system components. Two variants are described as follows:

The first case involves external organizations providing COTS components. The system developer has no control over the architectural and business decisions of COTS providers. For instance, a provider suddenly might change the product interface, making it difficult to upgrade. A provider may go out of business or simply discontinue support for the product in question. The following are examples of these kinds of risks:

COTS D. Subsystem D-A

Continued support of Subsystem D-A, and especially its COTS D-B component, by the vendor is far from certain. COTS D-B provides the performance gains necessary to allow using Subsytem D-A at all, but it is a one-of-a-kind solution not well understood by either Company D or its customer's personnel. (This is not surprising, since it depends heavily on the implementation of Subsytem D-A, and possibly the C++ run-time environment.) Continued reliance on it represents an ongoing risk to the long-term viability of System D, unless something dramatic changes in the vendor's marketplace to alleviate the concern about support.

Reliance on old version of COTS D-C product

The System D development effort is currently locked into an old version of the COTS D-C product, which in turn is locking the project into an old version of the Oracle database and (presumably) an old version of the COTS D-D product. While this will not prevent successful deployment of System D configurations, it does represent a long-term risk to the project that will

become more serious as System D's life expectancy grows.

The second case involves supplier organizations. In this instance, the supplier organization is developing some component for the system. However, the organization developing the main system does not control the software development process of the supplier organization. Therefore, the quality of those components cannot be ensured and may affect the safety and reliability of the entire system. The following are risks that arose from such a situation:

The architecture contains no explicit mechanism to isolate errors. This means that integration of COTS introduces a reliability risk.

No rules exist for defensive programming of the Company C components that must interact with the third-party components. Techniques such as pre-conditions, assertions, and contracts can be used to help isolate errors and keep them from propagating. This is a reliability risk.

There is a strong dependence on the certification process for security (e.g., to catch possible Trojan Horses), and no thought has yet been given to the certification process. This is a reliability risk.

Bad pointers in customer-supplied software will corrupt Company E's software.

4 Analysis of the Priorities of Brainstormed Scenarios

The second phase of an ATAM evaluation involves brainstorming and prioritizing scenarios [Kazman 00]. In this phase, a facilitator helps stakeholders generate scenarios that reflect their interests and concerns. The top scenarios are analyzed in subsequent steps of the ATAM evaluation. Because the brainstorming effort usually produces more scenarios than can be analyzed during the time available, the stakeholders cast votes for the scenarios they prefer. This, in effect, ranks scenarios by importance. The most important scenarios are then analyzed.

We suspected that the scenarios developed at the beginning of the brainstorming process would be the most important. Therefore, they would get the highest vote. To verify this hypothesis, we compared the order in which scenarios were brainstormed to their priority in the voting.

Figures 1–4 depict the results of these analyses for the ATAM evaluations. They show the *priority* of each scenario in the vertical axis, with higher priority meaning having received more votes. The horizontal axis represents the order in which the scenarios were brainstormed.

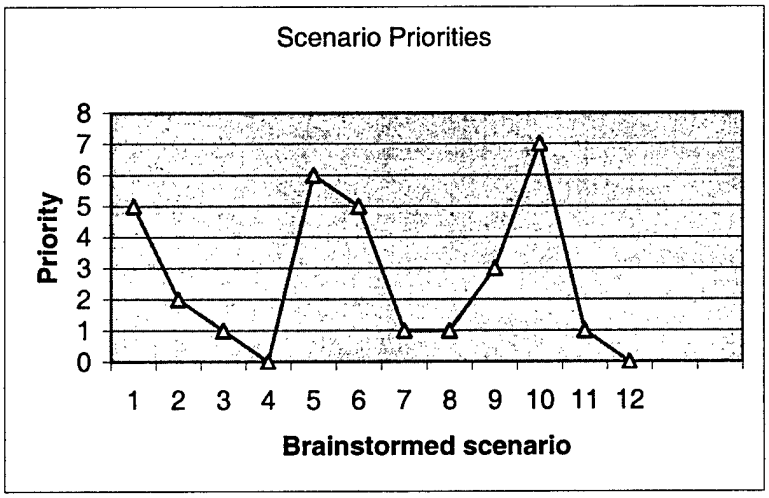


Figure 1: Priority of Scenarios in Report A

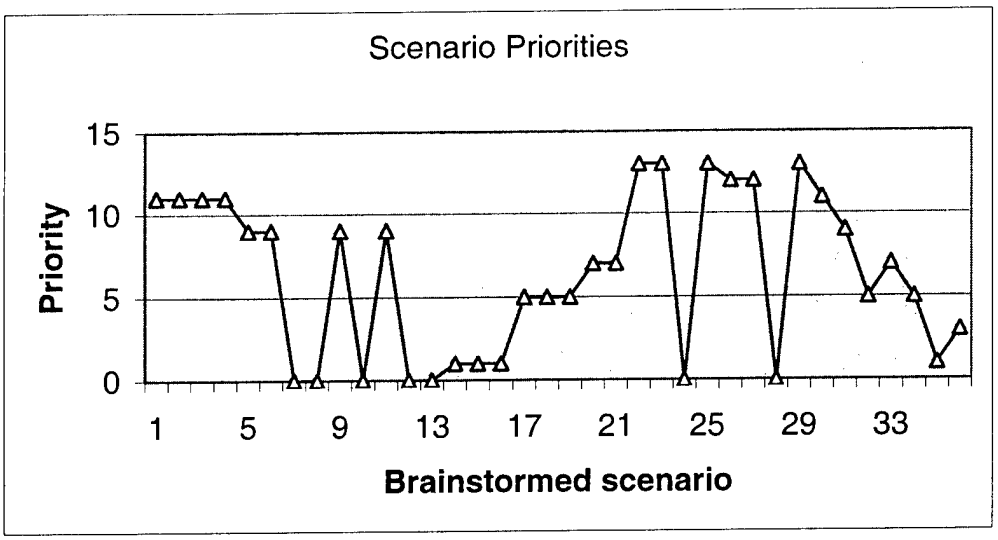


Figure 2: Priority of Scenarios in Report B

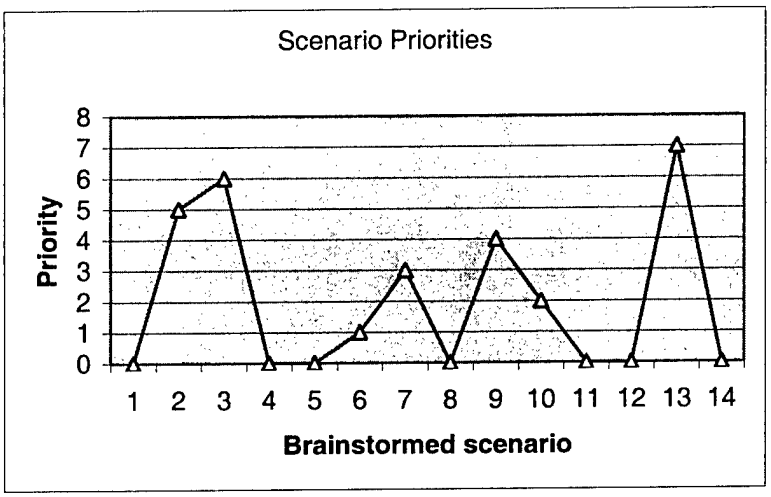


Figure 3: Priority of Scenarios in Report C

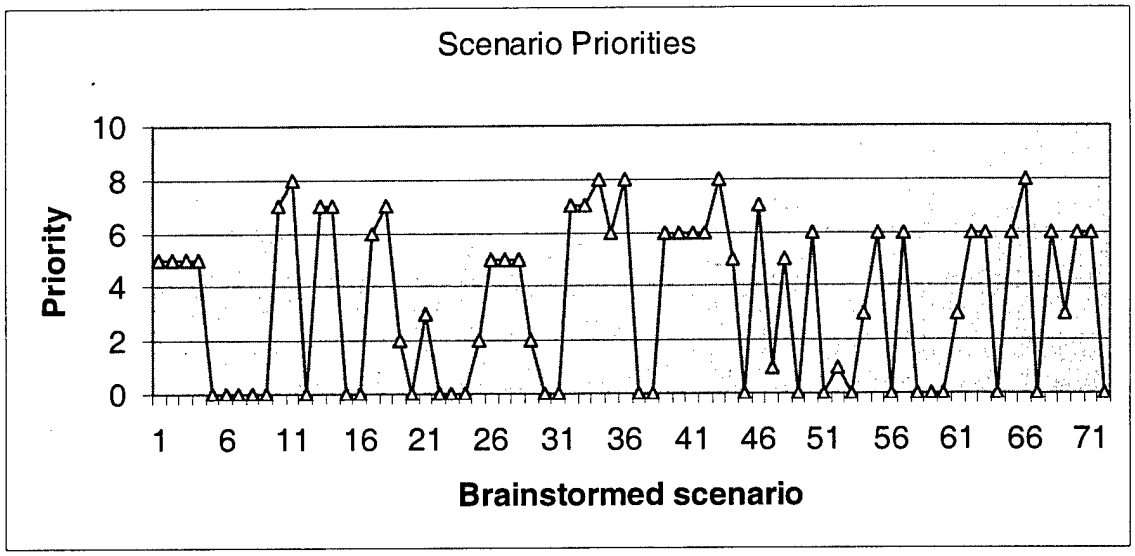


Figure 4: Priority of Scenarios in Report D

Figures 2 and 4 correspond to two ATAM evaluations in which comparable scenarios were represented by a new scenario. The voting process was done after the consolidation step. Due to the objective of this analysis, the scenarios shown in the horizontal axis represent the original ones to preserve their order in the brainstorming effort. During the consolidation process, some scenarios were used to generate not one but two new scenarios. In these cases, the scenario with the higher priority was assigned to the original one.

The charts show that important scenarios emerged throughout the brainstorming process. No statistical analysis was necessary because there obviously was no correlation between brainstorming order and the priority assigned. Therefore, our hypothesis that the most important scenarios emerge first was rejected.

5 Conclusions

The reports from ATAM evaluations can be a valuable tool in bringing quality attribute research closer to practice. These reports provided a large set of elicited scenarios and associated risks.

By mapping specific scenarios to general scenarios, we have shown that a collection of general scenarios can abstract situations of concern in software projects. Furthermore, using scenarios from ATAM evaluations as a sample population, we showed that the set of general scenarios in place when this study was begun has a very good coverage. At the same time, we identified new general scenarios to be added, as well as modifications that should be made to existing general scenarios.

It has yet to be determined whether the present collection of general scenarios provides the right level of abstraction. Every general scenario is an abstraction that, by definition, implies losing some details. The current level of abstraction may hide important details. This could be the subject of future research.

General scenarios could be used in the process of an ATAM evaluation as a guideline or checklist to create the utility tree. This does not mean that every system should have an instance of every general scenario. Rather, it does imply that the list of general scenarios can help identify instances that manifest certain quality attributes. In that way, it would be less likely that some scenario is neglected. Also, while eliciting scenarios, ATAM evaluators should be aware of malformed scenarios so they can avoid ambiguity problems.

References

[Bachmann 00] Bachmann, F.; Bass, L.; & Klein, M. H. *An Application of the Architecture-Based Design Method to the Electronic House* (CMU/SEI-2000-SR-009, ADA383836). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. WWW: <http://www.sei.cmu.edu/publications/documents/00.reports/00sr009.html> (2000).

[Barbacci 95] Barbacci, M.; Klein, M. H.; Longstaff, T. H.; & Weinstock, C. B. *Quality Attributes* (CMU/SEI-95-TR-021, ADA307888). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. Available WWW: <http://www.sei.cmu.edu/publications/documents/95.reports/95.tr.021.html> (1995).

[Bass 98] Bass, L.; Clements, P.; & Kazman, R. *Software Architecture in Practice.* Reading, MA: Addison-Wesley, Inc., 1998.

[Bass 00] Bass, L.; Klein, M. H.; & Bachmann, F. *Quality Attribute Design Primitives* (CMU/SEI-2000-TN-017). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. Available WWW: <http://www.sei.cmu.edu/publications/documents/00.reports/00tn017.html> (2000).

[Kazman 00] Kazman, R.; Klein, M. H.; & Clements, P. *ATAM: Method for Architecture Evaluation* (CMU/SEI-2000-TR-004, ADA382629). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. Available WWW: <http://www.sei.cmu.edu/publications/documents/00.reports/00tr004.html> (2000).

Appendix A: All ATAM Scenarios

This section presents the mapping resulting from the ATAM scenarios from the five ATAM reports to the general scenarios. There are several cases in this mapping:

1. The ATAM scenario maps directly onto only one of the general scenarios. In this case, we enumerate the ATAM scenario under the general scenario.
2. The ATAM scenario maps directly onto more than one of the general scenarios. In this case, we present the scenario as enumerated under one of the general scenarios and ignore it in the other one.
3. The ATAM scenario maps onto one of the general scenarios but the mapping requires a minor modification to the general scenario. In this case, we enumerate the ATAM scenario under the general scenario. We also indicate that it required a modification by appending (MOD) to the ATAM scenario.
4. The ATAM scenario does not map to any of the existing general scenarios. In this case, we enumerate the ATAM scenarios under a category, "Not Covered by General Scenarios."
5. The ATAM scenario is confusing. We enumerate these scenarios under a category called "Malformed" and do no further analysis of them.

We number the scenarios in this appendix from all of the ATAM reports that we examined sequentially from 1-170. Their order here does not reflect the order that they appeared during the ATAM evaluation.

Modifiability: 1

A request arrives to change the functionality of the system. The change can be to add new functionality, to modify existing functionality, or to delete particular functionality.

Report B (11 scenarios)

1. Additional data is requested to be presented to user.
2. User requests additional screens.
3. User requests different process (change of dialog, functional changes, etc.) or additional functionality.

4. Two fire orders, second is higher than the first. I cancel the first and respond to the second. Suppose doctrine changes. Example: We want to use this on a different weapon that has a different doctrine.

5. Map data format changes.

6. Automate the pointing of the weapon.

7. Automate the firing of the weapon.

8. User requests the simulation of the target location device (for training.) But current implementation allows dynamic simulation.

9. What happens if the customer requests a reduced functionality configuration that includes communications, VMF, subset of the protocols, subset of the missions, etc. Full ballistics kernel. Subset of screens, subset of training? (MOD)

10. An enumerated data type in a message is increased (e.g., new ammunition type or new fuse type, new characteristics for a mortar).

11. Message format changes from VMF to JVMF.

Report C (2 scenarios)

12. Customer wants different systems with different capabilities but using the same software.

13. Connect laptop to the system to send mail through the email features.

Report D (16 scenarios)

14. New type of fee is added to System D.

15. Business decides to increase late charge fee. Change made within two days.

16. Create a product that supports leasing.

17. Add a product that handles cash-on-demand.

18. New reporting requirement arrives that requires modification to metadata. Affects Subsystem D-A to Subsystem D-B translation.

19. Company D's client is told to participate in the ERCM program.

20. Adding portfolio manager role and supporting functionality.

21. Sell System D to insurance and have it support their business.

22. User requests new field for asynchronous query.

23. Company D's client needs to centralize approval decision process across multiple affiliates, and associated business process is re-engineered.

24. Manager wants report on historical delinquency rates for people who drive blue Chevies.

25. What if a proposed law change is applied to an account?

26. Affiliate re-defines business day and month.

27. Introduce new workflow process.

28. Report needs to be generated using info from two affiliates using different configurations.

29. End user wants to change output from paper to online viewing.

Report E (4 scenarios)

30. New Emission Laws: Cut the emissions by half in January 2002.

31. New Fuel Type: Use hydrogen by January 2002.

32. Simpler Engine Models: Replace the engine models in the software with simple heuristics for the low cost market.

33. Flight Recorder: A flight recorder is demanded by one customer in the diagnosis. Other customer needs diagnoses without the flight recorder. How can this be realized in one platform?

Modifiability: 2

The platform is changed. The system must be modified to continue to provide current functionality. There may be a change in hardware including input and output hardware, operating system, or COTS middleware.

Report A (1 scenario)

34. **Requirement:** Reduce time to upgrade OS, DB, Inf, AMASS COTS by 50% or within six months of release, whichever is sooner.

    Scenario: Upgrade from IRIX 6.2 to IRIX 6.5 and replace some hardware in one day. Upgrade Sybase in one day. Upgrade DCE in one day.

Report B (7 scenarios)

35. An additional device is added to the net. For instance a new target location device that sends back very accurate GPS information.

36. An existing device adds a new message type: same messages but with additional fields that we are currently not set up to handle.

37. Install or upgrade a new ballistics kernel.

38. User requests updates to existing kernel.

39. Modem baud rate is increased. Throughput of tactical network increases by a factor of four. Same data, but higher frequency of data per second.

40. Operating system changes to Solaris.

41. Can a new schedule be accommodated by the current operating system?

Report C (1 scenario)

42. Customer wants a driver information system with a different communication bus.

Report D (6 scenarios)

43. Oracle releases new version, and it is successfully hot-swapped.

44. Decide to support web-based interface.

45. Company D's client changes POS device vendors.

46. Need to change NT to another operating system.

47. Oracle is replaced by Informix.

48. COTS-D-A cannot support required version of Oracle.

Report E (7 scenarios)

49. Sensor change.

50. Knock control: The current ASIC and its software is replaced by a DSP and new software. This will influence the communication timing but the scheduling strategy shall be kept. There is no experience with DSP today at Company E.

51. High speed bus for actuators: Instead of wiring injectors and ignition, a serial time slot-driven high-speed bus is used.

52. Reduce memory: During development of an engine control, the customer demands to reduce costs by downsizing the flash-ROM from 600Kbyte to 512Kbyte on chip.

53. Continuous actuator: Changing two-point (on/off) actuators to continuous actuators within one month (e.g., for the EGR or purge control valve).

54. Absence of dual ported RAM: For a new system generation, the dual ported RAM mechanism is not available any more. Change the memory access for calibration data area.

55. Two processor system: The current single processor system changes to a two processor system due to performance reasons. How to handle the load balance between the two processors? Is dynamic scheduling needed?

Modifiability: 3

The operating environment is changed. For example, the system now has to work with systems previously not considered. It may have to operate in a disconnected mode. It may have to dynamically discover what devices are available. Or it may have to react to changes in the number or characteristics of users.

Report B (2 scenarios)

56. Change the number of weapons to be handled from 18 to 30.

57. Change the number of simultaneous targets from six to 12.

Report C (2 scenarios)

58. A CD player not in use is removed from the system. Delete all traces of the CD.

59. Integrate air conditioner into the system.

Report D (5 scenarios)

60. Build a system that has no central site connections but that has basic core functionality.

61. Company D's client divests a business unit. (MOD)

62. Consolidate two business units. (MOD)

63. Company D's client purchases company three times its size. What are the implications including partitioning of database? (MOD)

64. New subscriber wants to use current equipment instead of standard configuration.

Report E (4 scenarios)

65. V10 engine: How to support a 10 cylinder engine.

66. V8 with two systems

67. Multiple engine types in one car: hybrid engine

68. New technology: combustion pressure sensor: Add a combustion pressure sensor, remove many other sensors, and change the physical models. Deliver the first prototype in one year and support an SOP in four years.

Modifiability: 4

A request arrives to improve a particular quality attribute such as reliability, performance, or usability. The system should be modified to achieve better usability, for instance.

Report A (3 scenarios)

69. System initialization (Subsystem A-A and Subsystem A-B). Then individual computers initialize. Do you care about how long it takes to start up? Currently 10 minutes. Suppose the deadline for start-up is reduced from 10 minutes to five or three?

70. **Requirement:** 5X improvement for search response times

    Scenario: Search with 100 hits under normal ops, result in 30 seconds.

71. **Requirement:** Reduce regression testing from five days to one day.

    Scenario: Regression test Subsystem A-A deployment from M1 in one day.

Report D (1 scenario)

72. Company D's client encounters a search response problem, finds/fixes problem, issues new release.

Report E (2 scenarios)

73. Calibration: Reduce the calibration time by 50% for a customer.

74. Degraded start-up mode: During start-up not all devices, like some memory areas, are available due to low voltage. How can an incremental software start-up be implemented? (Today's solution is a "hot fix.")

Performance

Initiated event with resource demands must be completed within a time interval.

Report A (7 scenarios)

75. **Requirement:** System able to re-prioritize 1000 orders in 20 minutes by user class, data types, media type, destination, or user.

    Scenario: Backlog management: After 24 hours of down time, operations re-prioritizes backlogged workload in 30 minutes to ensure tasks are worked off in priority order and that normal operations continue to be supported with no degraded throughput following resumption of normal ops.

76. **Requirement:** System able to re-prioritize daily production in 20 minutes based on datatype and temporal coverage.

    Scenario: Backlog management

77. **Requirement:** Able to service 1000 concurrent requests through V0 Gateway or Sub-system A-A without operations intervention.

    Scenario: Sub-system A-B down for 24 hours, recovers and requests two days of data; work off in priority order. Receive 100 concurrent search requests, don't reject high priority requests and work off without overloading system as capacity permits.

78. **Requirement:** System can support 50 sites.

    Scenario: Cross-site order tracking across 50 sites, status in two minutes for a five-site order. Cross-site user registration in 24 hours across 50 sites.

79. **Requirement:** System can support ingest from 100 data sources.

    Scenario: Receive ingest requests from 100 sites, work off in priority order, and manage throughput to requirements.

80. **Requirement:** System can support electronic distribution to 2,000 sites.

    Scenario: Subscription fires for 2,000 users to send one GB of data to each, system works in priority order.

81. **Requirement:** System able to scale to 10X requirements for ingest, distribution, and processing without software changes.

    Scenario: 10,000 DPR/Day: An additional 16,000 DPRs will be planned and executed each day as part of normal operations with no additional staff or hardware.

Report B (1 scenario)

82. Suppose COTS-B-A software is too slow?

Report C (3 scenarios)

83. Turn the car right and the map display should turn so that the current heading of the car is at the top of the map.

84. Adjust audio volume and ensure immediate feedback.

85. Start the car and have the system active in five to 10 seconds.

Report D (7 scenarios)

86. A user initiates "update customer account" transaction under twice the current peak load, and the transaction completes within a second.

87. A user initiates "calculate terms" transaction under twice the current peak load, and the response is that the transaction completes within five seconds.

88. The 90th percentile of throughput testing trials (at 150 transactions per second with specified mix) will meet performance requirements.

89. Rule fires and data access are too slow.

90. Customer posts payment at a busy time and response is slow (in a testing environment).

91. Accounting has requirement to close financial statements within two business days, and system cannot respond.

92. Online calculation designed to run in batch mode has to process 60M accounts in four hours. (Evaluate impact to OLTP.)

Usability: 1

Systems should provide a batch or macro capability to allow users to aggregate commands. One response measure is the number of actions required to execute a series of commands once a batch or macro command has been specified. Another response measure is the difficulty of specifying the batch or macro command.

Report A (2 scenarios)

93. **Requirement:** Operator able to specify re-prioritization for 1000 orders in 10 minutes by user class, data types, media type, destination, or user.

    Scenario: Backlog management

94. **Requirement:** Operator able to specify re-prioritization of daily production in 10 minutes based on datatype and temporal coverage.

    Scenario: Backlog management

Report D (1 scenario)

95. One affiliate sells a large portfolio to another business unit.

Usability: 2

Users should have the means to reduce the amount of work lost in system failures. A response measure is the amount of time lost because of system failures.

Report B (2 scenarios)

96. What happens when the backup database is corrupted?

97. What happens if the database engine crashes?

Report D (12 scenarios)

98. In response to a complaint, Company D's client discovers it has been incorrectly collecting late charges for six months.

99. Defect corrupts data, not detected until next reporting cycle. (MOD)

100. Error in replication process causes OLTP database to be out of sync with DSR. (MOD)

101. Error in system causes all payments to accounts in Iowa to be un-postable. (MOD)

102. Transaction log audit trail fails on Subsytem D-A for three days. (How to recover?) (MOD)

103. Main communication to branches from info hub goes down.

104. Fire in data center forces movement of information hub to new location.

105. Recover from external security compromise ASAP.

106. Discover a configuration in branch server in all Western states invalid, causing problems in credit application processing.

107. Software distribution fails in the middle of the process.

108. GL/JE mapping in Subsystem D-A is incorrect and posted to ledger incorrectly.

109. Nightly rule versioning fails.

Usability: 3

Help procedures should be context dependent and sufficiently verbose to assist users in solving problems. Response measures include number of user problems solved using help facilities.

Report D (1 scenario)

110. A user in a particular context asks for help, and the help reflects the appropriate context.

Usability: 4

Systems should be easily configurable for deployment in multiple cultures. Response measures include the amount of human effort necessary to adapt the system to either a new language and culture or to invoke the system configured for a supported language and culture.

Report C (1 scenario)

111. One passenger is German and one is English; each should have an interface in the native language.

Report D (2 scenarios)

112. Spanish-speaking person uses system in familiar (nomenclature) Spanish.

113. Decide to support German.

Usability: 5

System designers should make upgrades and transitions occur as smoothly as possible. Response measures include number of errors made by a new user familiar with prior releases or by other members of the product line.

Report D (2 scenarios)

114. A domain-knowledgeable user can reach proficiency for core functions within a week.

115. A current System D-knowledgeable branch manager becomes proficient in Subsystem A-A system in less than a week.

Usability: 6

System designers should account for human needs and capabilities when deciding what aspects of system state to display and how to present them. Response measures include user satisfaction measurement to determine whether system state is displayed appropriately.

Report A (1 scenario)

116. **Requirement:** 75% of problems can be diagnosed by low level staff; 20% of problems can be diagnosed by

mid-level development staff; no more than 5% require senior staff.

Scenario: Server restart after fault takes more than 10 minutes (too long); operator detects in one minute. Granule insertion fails due to bad g-polygon in metadata; operator detects in 15 minutes. DDIST warm re-start fails because of bad request in DB; operator detects in 15 minutes.

Report C (1 scenario)

117. Perform remote diagnosis.

Usability: 7

Systems should account for human needs and capabilities when pacing the stages in an interaction. Systems should also allow users to adjust this pace as needed. Response measures include user satisfaction measurement to determine whether the system imposes an uncomfortable pace on the user.

Report D (2 scenarios)

118. Form-to-form response (a user finishing one form to using the next one) in a single MOU environment, and the system responds to the movement within one second. (MOD)

119. A user initiates loan application, and the user should be able to interact with the customer without the system impeding the user's performance. (MOD)

Usability: 8

Systems should provide a novice (verbose) interface to offer guidance to users operating in unfamiliar contexts. Response measures include error rates for users operating in an unfamiliar context.

Report D (1 scenario)

120. User in one business unit needs to perform actions on behalf of other business unit.

Availability: 1

A failure occurs and the system notifies the user; the system may continue to perform in a degraded manner.

Report A (5 scenarios)

121. **Requirement:** No system resources are held by data inputs or outputs that are failed or suspended for more than 10 minutes.

     Scenario: Fixed scene orders for electronic FTP push to a site whose FTP server is down, system suspends within 10 minutes of first failed request, and all resources are available while requests are suspended. Distribution to others is not affected.

122. **Requirement:** Data errors with one part of request (input/output) should not prevent fulfillment of other parts.

     Scenario: Order for 100 granules, three on off-line tape/drive, system suspends these requests in 10 minutes of first failure and operator is able to resume remainder.

123. **Requirement:** Operator should be able to identify problematic requests in 15 minutes and prevent similar requests (based on common characteristics) from entering the system until the problem is resolved.

     Scenario: See R1 and add user with failed site continues to send new order every 10 minutes, system queues requests. Distribution to others is not affected.

124. **Requirement:** No requests should be lost as a result of system overloads or failures.

     Scenario: Subsystem A-A must be cold started due to HW problem, pending orders identified and re-started in five minutes.

125. **Requirement:** All request responses are correct.

     Scenario: User orders five restricted granules (to which they do not have access); user gets notification of failed request in one hour.

Report C (1 scenario)

126. Detect software errors existing in third party or COTS software integrated into the System C.

Report D (2 scenarios)

127. Branch database server fails to boot.

128. Remittance center submits same batch of payments twice and activity occurs after second submission.

Availability: 2

A failure occurs and the system can interrupt services for a short time period. This interruption is not measured against system availability unless it exceeds a well-defined interval.

Report A (1 scenario)

129. Search, browse, and order submission unavailable no more that one hour/week.

Report B (1 scenario)

130. Will the system be able to convert from being a client to being a server within an acceptable amount of time? Ten minutes is an upper bound because it is the start-up requirement.

Availability: 3

A failure occurs in a component in a critical system, and the system continues to supply its services without interruption.

Report B (4 scenarios)

131. FDC gets taken out. Noticed and a new FDC is assigned via a priority.

132. How is battlefield geometry kept consistent between guns/FDC so that if the FDC is taken out the backup can take over? What happens when the rate of update of the backup approaches continuous?

133. What happens when a backup takes over for FDC?

134. What happens when a backup is taken out as it is being converted to an FDC, or just before conversion to FDC? Would battlefield geometry be lost?

Security

The system discloses information only to authorized people.

Report D (2 scenarios)

135. Unauthorized intrusion is resisted.

136. Previously public data is made private, and access is adjusted accordingly. (MOD)

Not Currently Categorized

Report A (4 scenarios)

137. **Requirement:** Changes to one subsystem require no changes to other subsystems.

     Scenario: Deploy 5B version of the Subsystem A-A and update engineering science data types (ESDT) and latitude-longitude box (LLBox) support into A5 baseline in less than eight hours with no impact on other subsystems or search, browse, and order availability.

     .

138. **Requirement:** Independently roll back subsystem deployments.

     Scenario: Perform rollback of SDSRV from M1.

139. Translate customers' requirements to software modules in one month (instead of three months today) with higher reuse and fewer errors.

140. **Requirement:** Able to update a PGE in Ops mode in less than 10 minutes.

     Scenario: PGE Updates: After SSI&T has approved a new PGE release, operations commissions the new PGE in operations mode in less than 10 minutes as part of normal operations. (See assumptions.)

Report B (1 scenario)

141. Same information is presented to user, but different presentation (location, fonts, sizes, colors, etc.).

Report D (3 scenarios)

142. Need to support multiple versions of System D at same affiliate simultaneously.

143. Want to distribute set of changes to a set of clients consistently (forms and configurations).

144. NT administrator has slowly transferred small amounts into various accounts. How to discover and determine extent?

Report E (5 scenarios)

145. Injection prototype: Deliver a prototype and a cost estimation for a new injection timing for a new engine. There is no experience yet at Company E. Make a feasibility study to acquire a new customer project in two months.

146. Single driver personalization: Support calibration by the driver/end user to adjust the engine to more power or more fuel efficiency. Personalize the car for one driver.

147. Personalize the car for more than one driver.

148. Safety of customer delivered software: Ensure the safety of the system with customer supplied object code to control the throttle control.

149. Changing integration responsibility: Company E's Calibration center in Asia is not deeply familiar with the motronic software, hence wants to change small portions of the software themselves for customers in Asia. This will change the delivery chain. Currently, only Plant-E-A does the integration.

Malformed

These scenarios are missing vital elements and we were unable to categorize them.

Report B (6 scenarios)

150. NT schedule is unpredictable.

151. Does COTS OS affect performance characteristics (without worrying about any particular mechanism)?

152. What happens when the backup FDC can't be used as a weapon because of too much update data swamping it?

153. What happens if the database can't keep up with the amount of data coming in and change requests?

154. The COTS-B-B software crashes.

155. Suppose you have a psychopath at the controls?

Report C (3 scenarios)

156. Purchase item over the Internet.

157. Information from the car is sent to a service station remotely.

158. Show rear camera image on screen.

Report D (8 scenarios)

159. Sell components of System D.

160. Data in info hub is replicated to branch office, and performance is degraded.

161. Attribute logging table grows very quickly (e.g., performance, archiving).

162. Batch processes are initiated based on time and events.

163. Need to refresh TLI on client side.

164. Task log grows very quickly.

165. Immediately reassign roles for employee branch to branch.

166. Phone company changes area code.

Report E (4 scenarios)

167. Incompatible scheduling: Customer supplies software that does not fit the task container model.

168. Critical software delivered by customer: A customer supplies software that does the ignition timing. How does Company E ensures that it works properly (e.g., does not run the engine backwards)? How does Company E debug and schedule the software?

169. Avoid low cost system with high features: A customer demands a low cost system with high end features (e.g., a variable valve control for the same customer's high end engine). How to tell the customer what is possible and feasible with current architecture and restrictions due to costs.

170. Business model change: Company E delivers only system and basic software. Application software, integration, and calibration is done by a customer. The Application Programmer Interface (API) is defined (a) by Company E, (b) by this customer, (c) differently by different customers.

Appendix B: General Scenarios

This section provides our list of general scenarios. There are actually three lists. Those portions of the scenarios that were in the list as of February 2001 are presented in normal font. Those portions that reflect modifications are indicated in italics, and those portions that were added to reflect omissions in the list are indicated in bold.

Each set of general scenarios is represented by a table. The table has the following entries: source of stimulus, stimulus, response, and possible response measures. The first two of these entries represent the stimulus and the last two represent the response. An actual general scenario is generated by specifying one entry from each row of the table and putting the result into acceptable English. That is, specify the source of the stimulus, the stimulus, the response, and how that response is measured.

Availability

Portion of Scenario	Possible Values
Source of Stimulus	internal to the system external to the system
Stimulus	unexpected event non-occurrance of an expected event
Response	The system should detect the event and then do one or more of the following: • Record it. • Notify appropriate parties including the user and other systems. • Turn off sources of events that cause failure according to defined rules. • Be unavailable for a pre-specified period where the period depends on the criticality of the system. • Continue to operate in a normal or degraded mode.
Response Measures	time period when the system must be available availability time time period in which the system can be in degraded mode repair time

Modifiability

Scenario portion	Possible Values
Source of stimulus	**end user** developer system administrator
Stimulus	wishes to add/delete/modify/*vary* functionality quality attribute *capacity*
Response	Locates places in the architecture to be modified Makes modification **without affecting other functionality** Tests modification. **Deploys modification.**
Response measures	difficulty in terms of time cost/effort in terms of number of components affected effort money

Performance

Scenario portion	Possible Values
Source of stimulus	event arrives from one of a number of independent sources
Stimulus	periodic stimuli sporadic stimuli stochastic stimuli
Response	Processes stimuli. Changes level of service.
Response measures	latency deadline throughput jitter miss rate data loss

Security

Scenario portion	Possible Values
Source of stimulus	individual or service that is correctly identified identified incorrectly of unknown identity who is highly motivated not highly motivated with access to limited resources vast resources
Stimulus	tries to display information change/delete information access system services *reduce availability to system services*
Response	Authenticates the user. Hides the identity of the user. Blocks access to data and/or services. Allows access to data and/or services. Grants or withdraws permission to access data and/or services. Records access/modifications or attempts to access/modify data/services by identity. Stores data in an unreadable format. Recognizes an unexplainable high demand for services and informs a user or another system and restricts availability of services.
Response measures	time/effort/resources required to circumvent • security measures with – probability of success – probability of detecting attack – probability of identifying individual responsible for attack or access/modification of data and/or services • percentage of services still available under denial of services attack • time/effort to restore data/services • extent to which data/services were damaged and/or legitimate access denied

Usability

Scenario portion	Possible Values
Source of stimulus	end user
Stimulus	wants to learn system features use a system efficiently minimize the impact of errors adapt the system feel comfortable
Response	The system provides one or more of the following responses to support: • "learn system features" – Help system is sensitive to context. – Interface is familiar to a user. – Interface that is usable in an unfamiliar context. • "use a system efficiently" – Aggregate data and/or commands. – Reuse already entered data. – Support efficient navigation within a screen. – Provide distinct views with consistent operations. – Provide comprehensive searching . – Allow multiple simultaneous activities. • "minimize the impact of errors" – Undo. – Cancel. – Recover from system failure. – Recognize user error. – Retrieve forgotten password. – Verify system resources. • 'adapt the system' – Provide **customizability.** – Provide internationalization. • "feel comfortable" – Display system task/state/duration/security level. – Work at the user's pace—neither too fast *nor too slow.*
Response measures	task time number of errors number of problems solved user satisfaction gain of user knowledge ratio of successful support requests to total requests *amount of time/data lost*

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	October 2001	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Applicability of General Scenarios to the Architecture Tradeoff Analysis Method	F19628-00-C-0003

6. AUTHOR(S)

Len Bass, Mark Klein, Gabriel Moreno

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2001-TR-014

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	ESC-TR-2001-014

11. SUPPLEMENTARY NOTES

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

13. ABSTRACT (MAXIMUM 200 WORDS)

The SEI has been developing a list of scenarios to characterize quality attributes. The SEI has also been conducting Architecture Trade Off Analysis Method[SM] (ATAM[SM]) evaluations. One output of an ATAM evaluation is a collection of scenarios that relate to quality attribute requirements for the specific system being evaluated. In this report, we compare the scenarios elicited from five ATAM evaluations with the scenarios used to characterize the quality attributes. This effort was designed to validate the coverage of the existing set of general scenarios and to analyze trends in the risks uncovered in ATAM reports.

SM Architecture Tradeoff Analysis Method and ATAM are service marks of Carnegie Mellon University.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Architecture Tradeoff Analysis Method, ATAM, architecture evaluations, quality attributes, general scenarios, specific scenarios, quality attribute primitives	65

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