

Categorizing Business Goals for Software Architectures

Rick Kazman
Len Bass

December 2005

TECHNICAL REPORT
CMU/SEI-2005-TR-021
ESC-TR-2005-021



CarnegieMellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Categorizing Business Goals for Software Architectures

CMU/SEI-2005-TR-021
ESC-TR-2005-021

Rick Kazman
Len Bass

December 2005

Software Architecture Technology Initiative

Unlimited distribution subject to the copyright.

This report was prepared for the

SEI Administrative Agent
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

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.

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 2006 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 FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

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

Table of Contents

Abstract	v
1 Introduction.....	1
1.1 Purpose of this Report.....	1
1.2 Method Used in this Study.....	1
1.3 Outline of this Report.....	3
2 Business Goal Categories	5
2.1 Reduce Total Cost of Ownership	5
2.2 Improve Capability/Quality of System.....	6
2.3 Improve Market Position.....	7
2.4 Support Improved Business Processes.....	7
2.5 Improve Confidence in and Perception of the System	7
3 Using the Categorization	9
3.1 Reduce Cost of Development	10
3.2 Reduce Cost of Deployment and Operations	10
3.3 Reduce Cost of Maintenance	11
3.4 Reduce Cost of Retirement/Moving to a New System	11
3.5 Improve Capability/Quality of System.....	11
3.6 Improve Market Position.....	12
3.7 Support Improved Business Processes.....	12
3.8 Improve Confidence in and Perception of the System	12
4 Summary	13
Appendix Elicited Business Goals.....	15
References.....	27

List of Figures

Figure 1: An Example ATAM Utility Tree 2

Figure 2: Business Goal Categories from the Affinity Diagram 5

Figure 3: The Extended Architecture Business Cycle 14

Abstract

Business goals are the foundation on which software systems are justified, analyzed, and built. Software systems are constructed to realize business or mission goals. Software architecture is the bridge between the business goals and the realized system. Those claims about business goals underlie many methods for designing and analyzing software architectures. However, precisely eliciting and characterizing business goals has always been problematic. Business goals come in many forms and at many levels of abstraction, and the stakeholders of the system are usually not accustomed to making goals explicit.

This report provides a categorization of possible business goals, so that stakeholders can have guidance in the goals' creation, expression, and documentation. The categorization was derived by mining a set of 190 distinct business goals elicited in 25 Architecture Tradeoff Analysis Method[®] (ATAM[®]) evaluations and then by performing an affinity diagram process to group the business goals into categories. For each goal, example scenarios are provided to illustrate how the goal might impact a system. Finally, this report shows how the architecture business cycle (ABC) may be extended by the business goal categorization.

1 Introduction

Two of the premises of the work of the Carnegie Mellon[®] Software Engineering Institute (SEI) in software architecture are that software systems are constructed to support some business or mission goals and that software architecture is the bridge between the business goals and the system. The argument for the first of these claims is that an organization or individual investing time and money in the construction of a system would not make this investment without having some goals in mind. The argument for the second claim is that software architecture is the artifact that allows for intellectual control of large systems; hence, it is a bridge between the abstractions of the business and the concreteness of the system being constructed.

The SEI has embodied these premises in the methods developed for

- generating architectural requirements (the SEI Quality Attribute Workshop [QAW] [Barbacci 03])
- designing architectures (the SEI Attribute-Driven Design [ADD] method [Bass 03])
- evaluating architectures and choosing among potential modifications (the SEI Architecture Trade-off Analysis Method[®] [ATAM[®]] [Kazman 99] and the SEI Cost Benefit Analysis Method [CBAM] [Kazman 01])

In each of these methods, a business goal elicitation step is used to understand the context and the “win” conditions for the system.

1.1 Purpose of this Report

The elicitation of business goals has often been problematic. Business goals (and the associated business context) come in many forms and at many levels of abstraction. Often the stakeholders of the system are not accustomed to making these goals explicit. Frequently these goals must be elaborated on to make them useful and understandable by a wider group, including analysts and external reviewers. Our goal in this report is to provide a categorization of possible business goals for software-intensive systems, so that individuals using our methods can have some guidance in the elicitation, expression, and documentation of business goals.

1.2 Method Used in this Study

A *utility tree* is constructed during an ATAM evaluation to elicit and prioritize the quality attribute goals of concern to a project. An example utility tree is shown in Figure 1. In that example, the top

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

level of the tree is the total utility of the system, and the second level is a list of quality attributes (with subdivisions) that are important to the project. The leaves of the tree (not shown in Figure 1) consist of quality attribute scenarios that are then candidates for evaluation.

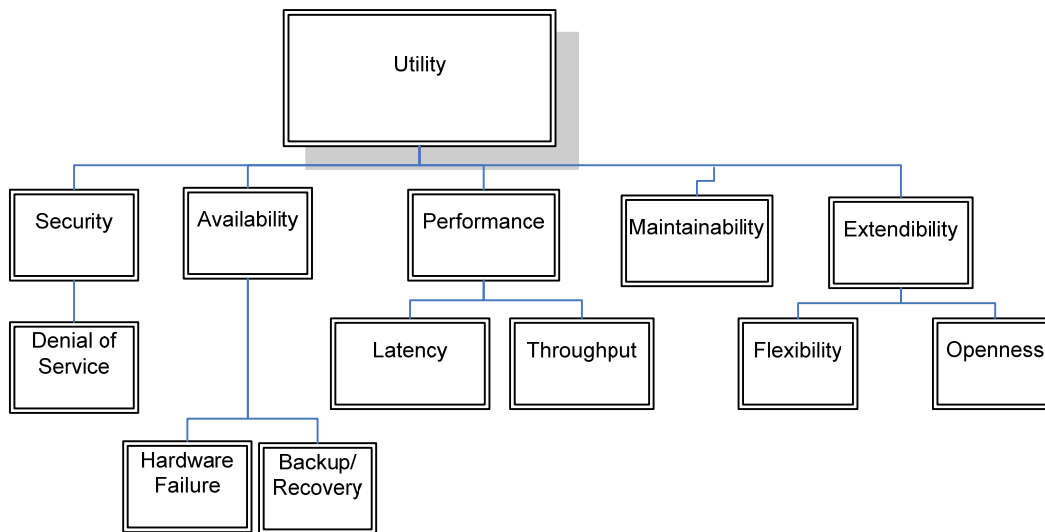


Figure 1: An Example ATAM Utility Tree

In this report, we modify the construction of the utility tree so that business goals appear at the upper levels and quality attribute scenarios derive directly from business goals—without the necessity of determining those business goals during an ATAM evaluation.

We derived our categorization of business goals by developing an affinity diagram [Beyer 98, Tague 04] from the 190 distinct business goals collected during 25 ATAM evaluations performed by the SEI between 1998 and 2005. Eighteen of those ATAM evaluations were performed on government systems, and seven were performed on commercial systems. The data that we present in this report has been “sanitized” to remove any system- or customer-specific information.

The *affinity diagram* was originally developed by Jiro Kawakita, an anthropologist, to discover meaningful groups of ideas from a raw list. Kawakita’s idea is to examine the list and let groupings emerge naturally, using the right side of the brain, rather than following a preordained categorization. An affinity diagram allows for categories that are not mutually exclusive. The steps of the affinity diagram process are as follows:

1. *Assemble the team.* Generating an affinity diagram is typically a team activity, relying on multiple viewpoints and ideas.
2. *Write individual statements on note cards.* The individual statements to be clustered are written down on note cards or self-adhering notes and given unique ID numbers. These statements may come from interviews, documents, surveys, brainstorming, or any other source.
3. *Group the statements.* The team then attempts to group the individual statements. There is no right or wrong in this activity, and the groupings should not proceed from any predetermined categorization. The categories should emerge from the statements and the ideas of the team.

Statements are allowed to be put into multiple groups (in which case the statement is written on multiple note cards).

4. *Name each group.* Once all statements have been allocated and the initial groups have been made, it is time to give each cluster a name. This name should be representative of the ideas that the group of statements have in common.
5. *Cluster the groups.* Typically there will be a large number of groups. These groups can themselves be grouped (i.e., groups will have a natural affinity with other groups), and the resultant higher level grouping should also be given a name.

For our set of 190 business goals, we completed Steps 1 and 2. Then, we iterated through Steps 3, 4, and 5 three times to assure that the groupings were stable and meaningful.

It should be noted that what we finally derived is a categorization not a taxonomy. It is permissible and even likely that the business goals of a particular organization could be placed into multiple categories. The important thing is that the business goals do, in fact, have at least one category in which they can appear. In this aspect, we followed the spirit of the organizational structure for quality attribute scenarios in which it is possible for a particular concrete scenario to be an instance of several different general scenarios, possibly deriving from different quality attributes [Bass 03]. This emphasis on categorization, rather than on taxonomy, has the virtue of freeing an organization trying to determine its own business goals from arguing over the appropriate placement for a business goal that could potentially belong to several categories.

1.3 Outline of this Report

In Section 2 of this report, we present our categories of business goals. (In Appendix A, we present the set of sanitized business goals that we used to derive the categories and the categories to which those goals were assigned.) In Section 3, we discuss how the categories can be translated into quality attribute requirements. In Section 4, we summarize the report and postulate on the use of the categorization to extend the architecture business cycle (ABC).

2 Business Goal Categories

As shown in Figure 2, five categories emerged from the affinity diagram process: (1) reduce total cost of ownership, (2) improve capability/quality of system, (3) improve market position, (4) support improved business processes, and (5) improve confidence in and perception of the system. We will discuss each of these categories in turn.

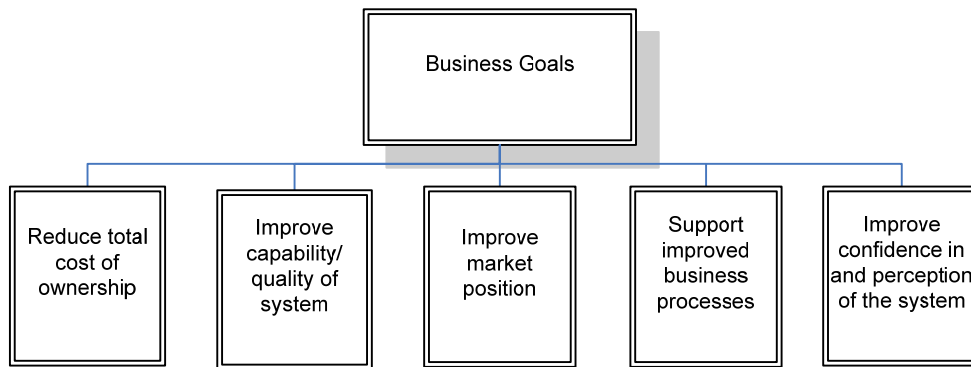


Figure 2: Business Goal Categories from the Affinity Diagram

2.1 Reduce Total Cost of Ownership

Not surprisingly, cost reduction is a business goal frequently mentioned in the ATAM evaluations. In some cases, cost reduction was mentioned as a general goal. In other cases, cost reduction was identified in a specific portion of the life cycle. We named our category *Reduce total cost of ownership* and subdivided it into the following groups: reduce cost of development, reduce cost of deployment and operations, reduce cost of maintenance, and reduce cost of retirement/moving to a new system. The kinds of business goals that compose these groups are as follows:

- reduce cost of development
 - manage flexibility
 - distributed development
 - portability
 - open systems/standards
 - testability
 - product lines
 - integrability
 - interoperability
- reduce cost of deployment and operations
 - ease of installation

- ease of repair
- reduce cost of maintenance
 - flexibility/configurability
- reduce cost of retirement/moving to a new system
 - retiring systems
 - smooth transition to follow-on systems
 - replace legacy systems

Note that many of these business goals are quality attributes [Bass 03]. All quality attributes are potentially business goals, but not all business goals are quality attributes.

2.2 Improve Capability/Quality of System

Another frequently mentioned group of business goals refers to the improvement of a system capability or quality compared to prior versions of the same system or contrasted with the system(s) being replaced. Sometimes the business goals only specified requirements on the current system without reference to prior systems. We grouped all of those business goals into this category. The underlying groups of goals in this category are as follows:

- performance
- reliability/availability
- product lines
- ease of use
- security
- safety
- scalability/extendibility
- functionality
- system constraints
- internationalization

2.3 Improve Market Position

Some business goals that emerged from the commercial ATAM evaluations have to do with market position or timing. The groups of goals underlying this category are as follows:

- expand or retain market share
- maintain or improve reputation
- enter new markets
- reduce time to market

2.4 Support Improved Business Processes

Another significant category of business goals is concerned with improving the internal business processes and the structure of the organization. Here, the underlying groups of goals are as follows:

- support distributed development
- maintain jobs of workforce on legacy systems

2.5 Improve Confidence in and Perception of the System

The final category includes goals intended to enhance the reputation of the developing organization. There was one group of goals in this category: maintain/expand reputation.

3 Using the Categorization

Now that we have created this categorization, what can we use it for? The motivation for this “archaeological dig” into the sets of business goals from past ATAM evaluations was that organizations have typically had trouble creating the top level of a utility tree and a business goals presentation. Furthermore, the business goals presentations that they did create were of widely varying quality. Frequently, organizations simply recycled an existing presentation. This presentation was typically not well tuned to the needs of the ATAM and hence produced poor results.

Hence, we see two significant uses of the business goals categorization that have ramifications for software architecture analysis.

1. *to aid in eliciting and structuring business goals.* The set of business goals presented in Appendix A and the categories that we have created via the affinity diagram process provide a starting point for an organization in thinking about its business and goals. In addition, an organization can consult this categorization to increase its confidence that it has created a complete, exhaustive set of business goals.
2. *to provide a link between utility and the specific quality attribute scenarios in the utility tree creation process.* The utility tree is a useful top-down elicitation and structuring device that has been successfully employed in ATAM evaluations for many years. Typically, however, stakeholders have trouble starting the process. Having example business goals to show stakeholders, along with sample scenarios that are derived from these goals, will facilitate the initiation of the process.

The first use of the categorization is reasonably straightforward. We will focus on the second use. In the balance of this section, we give examples of quality attribute scenarios that might be generated as a result of each business goal we identified. From the perspective of the ATAM evaluation, the point of business goals is twofold—to lead to the quality attribute scenarios and to express risk themes in terms of threats to the business goals. In discussing business goals in this section, we refer to the goals themselves not to the categories of goals.¹

¹ Our titles for Sections 3.1 through 3.4 are taken from the groups of goals in the category named *Reduce total cost of ownership* (described in Section 2.1). For Sections 3.5 through 3.8, our titles are taken from the category names.

3.1 Reduce Cost of Development

We took the more detailed breakdown of the *Reduce cost of development* goal (see Section 2.1) and generated sample scenarios, as shown below:

- *Business goal:* manage flexibility
Quality attribute scenario: An additional parameter is added to the system. The value of this parameter is checked for consistency and possible incorrect values within 10 minutes.
- *Business goal:* distributed development
Quality attribute scenario: Component A is being developed at Location X, and Component B is being developed at Location Y. A missed deadline is determined to involve both Components A and B. The cause of the missed deadline is determined, and a solution is proposed within two person-days.
- *Business goal:* portability
Quality attribute scenario: System A is moved from Platform B to Platform C without loss of functionality within two person-weeks.
- *Business goal:* open systems/standards
Quality attribute scenario: A component that adheres to Standard X is integrated into the system within two person-days.
- *Business goal:* testability
Quality attribute scenario: A modification to a feature is tested completely within two person-days.
- *Business goal:* product lines
Quality attribute scenario: A new product is produced. This product should reuse more than 80% of the core assets and should take no more than eight person-years to complete.
- *Business goal:* integrability
Quality attribute scenario: The integration of Subsystems A, B, and C will be completed within two person-months.
- *Business goal:* interoperability
Quality attribute scenario: When deployed, System A will be able to interoperate with existing Systems B and C using Protocol D without further modification.

3.2 Reduce Cost of Deployment and Operations

Business goal: ease of installation and ease of repair

Quality attribute scenario: A new release of System A can be deployed and installed on the desktops of all users within two days.

Quality attribute scenario: The number of operators required to operate System A will be half of that required to operate existing System B.

3.3 Reduce Cost of Maintenance

Business goal: flexibility/configurability

Quality attribute scenario: A new feature can be added to System A within two person-days.

3.4 Reduce Cost of Retirement/Moving to a New System

Business goals: retiring systems, smooth transition to follow-on systems, and replace legacy systems

Quality attribute scenario: At the end of its useful life, System A can be retired and its functions can be transferred to a new system within two person-days, exclusive of hardware modifications.

3.5 Improve Capability/Quality of System

- *Business goal:* performance
Quality attribute scenario: User enters a map request over the Internet during normal operation and the complete map is sent to the user within two seconds.
- *Business goal:* reliability/availability
Quality attribute scenario: When a sensor stops responding, the system detects the faulty sensor, sets its status to *unavailable*, and reports the status to an operator within 100 milliseconds.
- *Business goal:* product lines
Quality attribute scenario: A new product is produced. This product should reuse more than 80% of the core assets and should take no more than eight person-years to complete.
- *Business goal:* ease of use
Quality attribute scenario: With no training, a novice user can create a simple drawing using the drawing tool.
- *Business goal:* security
Quality attribute scenario: When an end user tries to access an unauthorized Web page by navigating to it directly without signing in, access is denied, and the attempt is logged.
- *Business goal:* safety
Quality attribute scenario: A pilot attempts to land an aircraft without lowering the landing gear. The system warns the pilot and takes corrective action.
- *Business goal:* scalability/extendibility
Quality attribute scenario: Two new data servers can be added to the system in two person-days, with no system downtime.
- *Business goal:* functionality
Quality attribute scenario: Add functionality to a Web site, such as the ability for end users to calculate their own amortization schedule.

- *Business goal:* system constraints
Quality attribute scenario: The system will be implemented using PHP 5, MySQL 4.1, and Apache 2.0.
- *Business goal:* internationalization
Quality attribute scenario: The system will be ported to a new language with one person-day of development effort, assuming that the text-string files have already been translated.

3.6 Improve Market Position

- *Business goal:* expand or retain market share
Quality attribute scenario: System A can be migrated to Platform B within one person-week.
- *Business goal:* maintain or improve reputation
Quality attribute scenario: System A provides 20% better response time on Use Case X than its competitors with 10% fewer errors.
- *Business goal:* enter new markets
Quality attribute scenario: System A will enable Organization B to sell products in New Market C.
- *Business goal:* reduce time to market
Quality attribute scenario: System A will be completed within two calendar years.

3.7 Support Improved Business Processes

- *Business goal:* distributed development
Quality attribute scenario: System A will be developed jointly by teams in Locations B, C, and D within 20 person-years and within six months.
- *Business goal:* maintain jobs of workforce on legacy systems
Quality attribute scenario: System A will be maintained by existing teams in Locations A and B for the next two years.

3.8 Improve Confidence in and Perception of the System

Business goal: maintain or improve reputation

Quality attribute scenario: System A provides 20% better response time on Use Case X than its competitors with 10% fewer errors.

4 Summary

In this report, we have categorized the 190 business goals that we have observed in 25 different ATAM evaluations. The categories were developed through the use of the affinity diagram method and included

- reduce total cost of ownership
- increase capability/quality of system
- improve market position
- support improved business processes
- improve confidence in and perception of the system

While the affinity diagram process is not exact, the business goal categories derived are intuitively appealing. They represent the core sets of business issues that affect the vast majority of systems. For each business goal categorized, we have provided sample scenarios. (Further, in Appendix A, we have provided a sanitized, representative list of business goals drawn from nearly a decade of experience with ATAM evaluations. We removed some goals from the list in order to protect the proprietary interests of the customer organizations.)

In this report, too, we discussed how the categories that we derived, along with the sample scenarios generated from the categorization, can assist in an architectural analysis in two distinct ways:

1. enabling organizations to make more focused and complete presentations of their business goals
2. simplifying the generation of the utility tree through the insertion of business goals at the higher levels of the tree and the provision of sample scenarios for each business goal

In addition, we can speculate about a third use for the categorization: as a means of organizing and elaborating the ABC [Bass 03]. Originally, the ABC was envisioned as a means to depict the influences on a software architect and to show how architectures can eventually influence the very things that originally shaped them. Architects were viewed, in the original ABC, as being influenced by stakeholders, by the developing organization, by the technical environment of the day, and by their organization. Similarly, the resulting architecture affected all of these influences. These feed-forward and feedback influences form a *cycle*.

In light of the categorization presented here, we can now postulate a far more detailed ABC, one that includes a greater variety of business goal considerations, as shown in Figure 3.

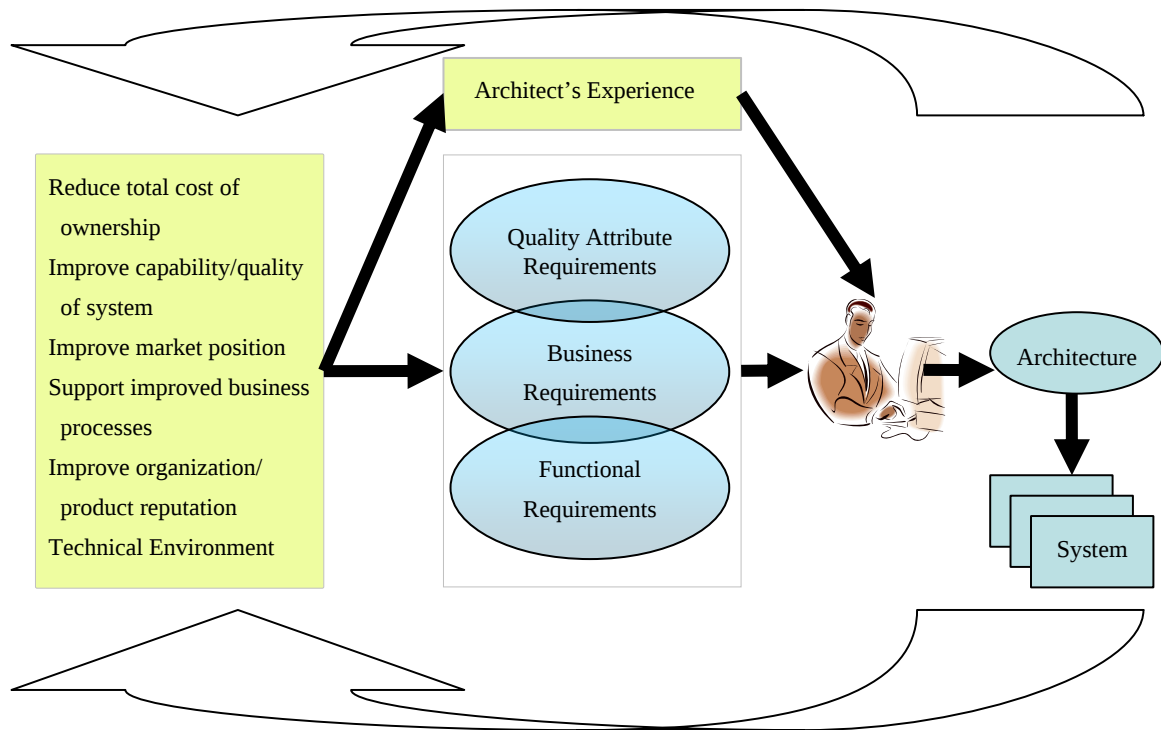


Figure 3: The Extended Architecture Business Cycle

Appendix Elicited Business Goals

In this appendix we present the categories labeled in Section 2 along with the (sanitized) business goals that we used to derive them. We also provide a justification of the grouping that we performed by presenting what we saw as the common theme.

Retiring Systems

In this category, we placed business goals that refer to the replacement or retirement of systems:

- Support System X to System Y transition.
- Improve Obsolescence Management.
- Replace physical hardware every 26 weeks.

Expand or Retain Market Share

In this category, we placed business goals that refer to market share:

- Business Unit A needs to expand market leadership, while Business Unit B needs to maintain market leadership and its role as a leading system supplier.
- Expand the role of the company as a global supplier.
- Open new markets for Company A.

Develop a Product Line of Software

In this category, we placed business goals that refer to the desire to build multiple systems of a similar character:

- Create a hardware and software platform that can be reused to a very high degree. Only features specific to a new product would need to be implemented. All existing functionality can be used without any change.
- Provide a complete product range: low, mid, and high.
- Promote a flexible service concept for applications (i.e., enabling third-party development and/or integration using an approachable, modular platform).
- System X is the first instance in Product Line L. The product line includes multinational customers, so *separability* of design components is required by export. The product line needs to lower the cost and reduce the time to market for new customers.
- Provide scalability between platforms.

- The system has a few key constraints. One is that demonstrable portions of the system are required during the architecture definition.
- There is a need to serve several manufacturers and several lines of each manufacturer with the same set of software. There is the need to use the same components for different systems and also the need to support both low-cost and high-end systems.
- Have the same interface and/or architecture for Business Unit A and Business Unit B products. Currently their systems have different architectures and different interfaces. Company A would like to migrate to a common architecture, core functionality, and interface for Business Unit A and Business Unit B systems.
- Scale systems down for both low-cost and high-end markets.

Manage Flexibility

Organizations whose systems are designed for flexibility frequently have many parameters to control the behaviors of those systems and the products in a product line. The business goals in this category include

- Provide the ability to support offshore production and/or calibration: Company A would like to produce and calibrate its systems in a variety of countries.
- Reduce the calibration effort for Product Line 1 and Product Line 2 systems. Currently, calibrating systems is complex and problematic and requires special expertise. Consequently, it is costly.
- Enable extendibility. Organization A must be able to support “what if” analysis by accommodating new models, capabilities, and algorithms—in addition to supporting the increased fidelity required by users.

Distributed Development

These business goals are concerned with the design used to develop systems when developers are distributed globally:

- Support offshore production and/or calibration. Company A would like to produce and calibrate its systems in a variety of countries.
- Utilize offshore development capacities to ensure on-time delivery and use of the best expertise per task.

Portability

These business goals are concerned with developed systems being able to run on a variety of platforms:

- System A must also be flexible with respect to its host hardware environment.
- Operating system independence is an issue. Current customer requires OS 1, whereas other manufacturers may require other operating systems.

- The system must be modifiable to migrate to other platforms and allow for upgrades as devices change or new devices are made.
- Sell software as a product. Currently, Company A delivers hardware and software bundled as a system. In most cases, the software is highly customized. Thus the software will have to run on multiple platforms.

Enhance Reputation and Credibility

These business goals are concerned with the reputation of the developing organization or the adoption of the approach used:

- The team is attempting to demonstrate the “goodness” of its approach to other business units.
- Be the best in class. The products of Company A are currently the best in class in terms of functionality, and the company would like to continue to be the best in class while expanding to a broader range of markets.
- Produce high-quality systems. Quality here is measured in terms of system returns per line of code in the industry. Need to maintain reputation as the best in class.
- Users of System A must be able to trust the system for use in integration, test, and analysis roles.

Performance

These business goals are concerned with the runtime performance of the developed system:

- Support the ability to scale systems down for low-cost market as well as up. Generally, Company A is able to scale systems up but finds it difficult to scale them down to meet low-cost demands.
- Use the same components for different platforms and to support both low-cost and high-end models.
- Provide the ability to scale between platforms.
- The system has to handle soft real-time performance requirements for a large number of objects. Performance growth is a concern, as higher capacity external networks become available.
- Provide predictable operation in terms of performance and resource usage.
- Have a smaller memory footprint.
- Performance margin is the root of all increased capability; without it System A has no ability to grow. However, customers want System A to be as fast as possible.
- Connect to more outside partners in support of the integration roles, allow more hardware, and provide better performance.
- The system has to handle both hard and soft real-time performance requirements. Performance growth is a concern, as higher capacity external networks become available.
- Improve processor throughput.

- Increase bus bandwidth for primary system and backplane busses.
- Beat the competition (i.e., be more reliable and have higher performance while offering more features than the competition).
- Performance is an important consideration beyond having simply to meet hard deadlines. For example, additional performance considerations include a tradeoff between bandwidth consumption and local processing, and flexibility versus predictability.

Use of Open Systems and Standards

These business goals pertain to the use of open systems and standards in an effort to reduce the costs and encourage the use of commercial off-the-shelf (COTS) systems:

- The definition of standard interfaces gives vendors a target and permits competition among vendors in developing products for the same subdomain.
- The plentiful use of COTS will lower both initial and life-cycle costs.

Flexibility/Configurability

These business goals pertain to the requirement that the systems under development operate in a variety of contexts:

- The software has to support different hardware configurations where some functions are either located in a separate device or integrated into the head unit. This configurability enables more flexible pricing for different market segments.
- This system is expected to have a 40-year life. All systems will not have the same configuration; they may even change from run to run. Flexibility is *key* to the system.
- The ability to support a wide range of users and roles is critical with the new user communities that System A will have to support. Flexibility has to be maintained at the model and architectural levels to support multiple users, locations, and so forth. System A must also be flexible with respect to its host hardware environment.
- Reduce the calibration effort for Product Line 1 and Product Line 2 systems. Currently calibrating systems is complex and problematic, requires special expertise, and consequently is costly.
- The system is expected to have a 30-year life. The system must be expandable to allow the addition of new input devices and new or additional computing resources as these technologies evolve. The system must be reconfigurable, so that the same system can be used for many different purposes.

Reliability/Availability

These business goals pertain to the requirements for reliability and availability that the system must have:

- There is a need to accommodate faults and a high volume of changes due to equipment failure and operational considerations, both within a platform and across the network.
- Eliminate customer returns.
- Beat the competition (i.e., be more reliable and have higher performance while offering more features than the competition).
- Make all data holdings available all the time to every user.
- Ensure data integrity for an archive.
- There is a need to provide better test coverage. System A now has to be more reliable for multiple concurrent customers and for a growing number of customers overall. To support this condition, there is a need for better test tools and an ability for users to score a given test as a success or failure.
- The system must have an operational availability of 95% (and a maintenance ratio that does not exceed 0.05 maintenance man-hours/operating hour).
- Company A must offer survivability and recoverability.
- Company A must improve the quality and reliability of products.
- Company A needs to ensure safety and reliability to limit liability as much as possible.
- System A must be at least as reliable as the attached devices and must not diminish the reliability associated with using those devices. The failure of any device cannot compromise the rest of the system, and devices must still be able to work when System A fails. The system must be available 24x7, since users are spread all over the globe and can be working in any time zone. The availability of System A will be from 90% to 99.9999% depending on the product instance.

Testability

These business goals pertain to the ease and effectiveness of testing the system:

- Users of System A must be able to trust the system for use in integration, test, and analysis roles.
- There is a need to provide better test coverage. System A now has to be more reliable for multiple concurrent customers and for a growing number of customers overall. To support this condition, there is a need for better test tools and a capability for users to score a given test as a success or failure.

Creation of New Markets

This business goal pertains to enabling an organization to move into new markets: sell software as a product. Currently, Company A delivers hardware and software bundled as a system. In most cases, the software is highly customized. Thus, the software will have to run on multiple platforms.

Reduce Costs

These business goals pertain to the reduction of costs associated with the system. All types of costs are included in this category. As a result, this category includes all of the goals in other cost-related categories, in addition to other goals:

- Use COTS components whenever they are reliable enough.
- Deliver the first product using System A for the price of “X” dollars apiece.
- Use off-the-shelf software components when possible.
- Adopt a “buy rather than build” approach to software.
- System A will lower the cost and cycle time for system integrators.
- System A’s goal is to enable system builders to integrate systems on time and within cost—while meeting performance needs. The number one objective is to drive down the cost of systems.
- Automate operations to minimize operational costs.
- Improve Obsolescence Management.
- Support technology refresh.
- Reduce and eventually eliminate the need for “virtuosos” to support the setup and operation of the system.
- System A must support two kinds of debugging: (1) scenario debugging and (2) software debugging. The system must be responsive to “non-virtuoso” customers. It must take less time to implement fixes and to add new features for analysis, test, and integration customers. Maintenance must be easy and quick.
- The cost of ongoing software maintenance is minimized. Latent defects are corrected without major effort.
- The system must have an operational availability of 95% and a maintenance ratio that does not exceed 0.05 maintenance man-hours/operating hour.
- Lower maintenance and support costs. There are over 10,000 systems fielded and several software baselines. System A will be in the field for the foreseeable future.
- Reduce lifecycle costs.
- Reduce manning.
- Minimize sustainment costs.

- Minimize systems acquisition.
- Reduce the cost to develop new products.
- The architecture should help reduce the current product cost of Product Line 1 and Product Line 2.

Integrability

These business goals pertain to the ease of integration of various pieces of the system during construction:

- Support “plug and play” and “smart” devices.
- System builders should be able to use Product A to quickly integrate the subdomains. Under current practices, integration costs are very high. Point solutions are too expensive and not timely.
- System A’s goal is to enable system builders to integrate systems on time and within cost—while meeting performance needs. The number one objective is to drive down the cost of systems.
- Support a flexible service concept for applications: enabling third-party development and/or integration using an approachable and modular platform.
- Support externally developed algorithms/applications.
- Support system software development attributes: modifiability, quick and cost-effective changes to the system, reduced integration time, ability to effectively add new functionality, and limited impact by external changes.
- The definition of standard interfaces gives vendors a target and permits competition among vendors in developing products for the same subdomain.

Functionality

These business goals pertain to the value delivered to customers as a result of the existence of the system:

- Support a diverse set of customers.
- Support the step-by-step buildup of system competence.
- Beat the competition (i.e., be more reliable and have higher performance while offering more features than the competition).
- Provide one-stop shopping (i.e., location transparency of system components and data).
- Support operational capabilities, improvements, and future growth.
- The database and applications developed by Program A will enable the development of subsequent modernized systems.

- Program A will enable employees to post transactions and update account data from their desks. Updates will be immediately available to anyone who accesses data and will provide a complete, timely, and accurate account of the customer's information.
- The project will replace the master files and related processing with new technology, new applications, and new relational data stores.
- The system must be capable of complexity management—including planning, fusing of information, and running estimates.
- Software evolution and modifiability must be supported in a seamless manner. The software must be able to accommodate changes in doctrine, performance requirements, modernization, and hardware and software insertion. Changes need to be inserted in one place and propagated throughout the system.
- Support the deployment of applications for mission planning or execution monitoring.
- Support modernization with minimal impact to readiness.
- Maintain flexible relationships with customers. Company A currently has very flexible relationships with its customers and would like to maintain these relationships.
- Support functional upgrades.

Ease of Installation

These business goals pertain to the cost of installing copies of a system:

- Usability requires self-configuration, self-healing, remote troubleshooting, and repair. The user needs to use the software with a minimal amount of on-site field software support and a minimal logistics footprint.
- Support deployability.
- Support usability in terms of deployment, configuration, and operation. System A needs to be able to support multiple end users currently. To these ends, System A needs to be “self-sufficient.” That means reducing and eventually eliminating the need for “virtuosos” to support the setup and operation of the system.

Interoperability

These business goals refer to the ability of the system to exchange data with a variety of other systems and devices:

- Connecting different devices, such as radios or CD players, must be done using the bus standard. Use of this standard helps manufacturers achieve their business goals by enabling them to use different suppliers.
- Contribute to achieving system interoperability.
- The system will need to interoperate with many other systems. It will need to communicate on several existing and future networks. Additionally, the system needs to support network-centric

operations, which increases the speed and quantity of data/information exchanged with other systems.

- System A will need to comply with a large variety of standards and protocols.
- Ensure that the new systems work well with other programs.
- System A must be able to interoperate with other such systems. The standards in this domain are evolving, and conformance with the standards is not sufficient to ensure interoperability.
- Ensure communications/interoperability.
- The definition of the standard interfaces gives vendors a target and permits competition among vendors in developing products for the same subdomain.

Ease of Repair

These business goals pertain to the cost of detecting or repairing failures in the system after deployment:

- Ensure supportability/ease of upgrade.
- System A must support two kinds of debugging: (1) scenario debugging and (2) software debugging. The system must be responsive to “non-virtuoso” customers. It must take less time to implement fixes and to add new features for analysis, test, and integration customers. Maintenance must be easy and quick.
- Usability requires self-configuration, self-healing, remote troubleshooting, and repair. The user needs to use the software with a minimal amount of on-site field software support and a minimal logistics footprint.

Time to Market

These business goals pertain to the time taken to implement the system:

- Limit the time frame to implement the system to a two-year period.
- Reduce the average time required to deliver products to market.
- Deliver the first product using the platform for the price of \$240–\$360 each by September 2006.
- The system will emphasize reuse in order to meet schedule constraints. Reuse will include the use of COTS components, as well as a large number of artifacts derived from existing systems—including design, code, and test cases.

Ease of Use

These business goals refer to the ease with which an end user can operate the system:

- Predictable user experience impacts the “sellability” of system.

- The training capability must be embedded within the software to enable on-demand training and simulated training.
- Provide better support for usability by operators.
- The system should be usable by an average human. There is no support, no help desk, and no training provided for the system. The system should be localized to meet different language/regional specific requirements and support differences in culture. This requirement applies to different user interfaces for different manufacturers.
- The system must be a common portal for all users. Based on roles, they will have access to different interfaces. Given the large number of users and their remote locations, it is important that the user interface be intuitive and easy to learn.
- Avoid operator overload.

System Constraints

These business goals pertain to the constraints introduced by the physical environment in which the system must live:

- The system has a few key constraints. One is that demonstrable portions of the system are required during the architecture definition. Another is that the hardware must fit (dimensions, power, and weight) on the vehicle.
- The system must be transportable worldwide by air, sea, highway, and rail modes.

Security

These business goals refer to security requirements on the system:

- System A must support authentication for both remote and local users for access to various capabilities. System A must support confidentiality as the system might have multiple users with different levels of permission simultaneously using the system.
- Physical security is not an issue, since the system is protected by means of a door lock. What is of concern is the protection of personal data especially against viewing by service technicians and protection against illegal manipulation of the system. In the event of an accident, product liability is a concern, and illegally manipulating the system may be a cause of an accident.

Safety

These business goals pertain to the issue of safety:

- There will be a formal hazard identification program that will require a risk assessment for each identified hazard. Development of mitigation strategies will be the primary responsibility of all affected teams, and the software process requirements associated with each risk level will need to be identified.

- There should be almost no defects in the system. The system is for the commercial market, which means that the buyer must be able to trust the entire product. Errors from the system yield serious damage to the image of the original equipment manufacturer. System A is for the high-end market, and a “quality image” is even more important in this segment. There is a safety issue in that a defect in the system may distract the user.

Legacy Systems

These business goals pertain to the replacement of existing systems:

- Provide a new single system that combines the distinct legacy financial systems.
- System A addresses the needs of its two major businesses: (1) Business X, consisting of 11 different acquired businesses and (2) Business Y services, supporting 14 different subscribers. System A will replace the multiple existing legacy systems, which are old (one is more than 25 years old), based on aging technology (e.g., COBOL and IBM assembler), difficult to maintain, and unresponsive to the current and projected business needs of the division.

Internationalization

These business goals pertain to the use of the system in multiple countries with diverse languages.

- Support the goal of opening new markets for the division.
- Support multiple languages and currencies.
- Support the ability to deal with diverse cultural and regional differences.
- Expand the role of Company A as a global supplier.
- Provide support for diverse customers.

Set Standards

These business goals pertain to the desire for organizations to set standards:

- Set the standard for function, quality, and architecture: Company A would like to build a software architecture that becomes a *de facto* standard for all systems in its market.
- The definition of the standard interfaces gives vendors a target and permits competition among vendors in developing products for the same subdomain.

Maintain Jobs of Workforce

The following business goal pertains to the commitment of an organization to maintain all of the jobs of its workforce as a new system replaces a legacy system: support an aim of retraining existing employees and a commitment to the employees of Company A of no jobs lost.

References

- [Barbacci 03]** Barbacci, M.; Ellison, R.; Lattanze, A.; Stafford, J.; Weinstock, C.; & Wood, W. *Quality Attribute Workshops (QAWs), Third Edition* (CMU/SEI-2003-TR-016, ADA418428). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003.
<http://www.sei.cmu.edu/publications/documents/03.reports/03tr016.html>.
- [Bass 03]** Bass, L.; Clements, P.; & Kazman, R. *Software Architecture in Practice, Second Edition*. Boston, MA: Addison-Wesley, 2003 (ISBN 0-321-15495-9).
- [Beyer 98]** Beyer, H. & Holtzblatt, K. *Contextual Design: Defining Customer-Centered Systems*. New York, NY: ACM Press, 1998.
- [Kazman 99]** Kazman, R.; Barbacci, M.; Klein, M.; Carriere, S.; & Woods, S. "Experience with Performing Architecture Tradeoff Analysis," 54–63. *Proceedings of the 21st International Conference on Software Engineering (ICSE 21)*. Los Angeles, CA, May 16–22, 1999. New York, NY: ACM Press, 1999.
- [Kazman 01]** Kazman, R.; Asundi, J.; & Klein, M. "Quantifying the Costs and Benefits of Architectural Decisions," 297–306. *Proceedings of the 23rd International Conference on Software Engineering (ICSE 23)*. Toronto, Canada, May 12–19, 2001. Piscataway, NJ: IEEE Computer Society Press, 2001 (ISBN 0-7695-1050-7).
- [Tague 04]** Tague, N. R. *The Quality Toolbox, Second Edition*. Milwaukee, WI: ASQ Quality Press, 2004 (ISBN 0-87389-639-4).

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
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 2005		3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE Categorizing Business Goals for Software Architectures			5. FUNDING NUMBERS FA8721-05-C-0003	
6. AUTHOR(S) Rick Kazman & Len Bass				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213			8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2005-TR-021	
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 ESC-TR-2005-021	
11. SUPPLEMENTARY NOTES				
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS			12B DISTRIBUTION CODE	
13. ABSTRACT (MAXIMUM 200 WORDS) Business goals are the foundation on which software systems are justified, analyzed, and built. Software systems are constructed to realize business or mission goals. Software architecture is the bridge between the business goals and the realized system. Those claims about business goals underlie many methods for designing and analyzing software architectures. However, precisely eliciting and characterizing business goals has always been problematic. Business goals come in many forms and at many levels of abstraction, and the stakeholders of the system are usually not accustomed to making goals explicit. This report provides a categorization of possible business goals, so that stakeholders can have guidance in the goals' creation, expression, and documentation. The categorization was derived by mining a set of 190 distinct business goals elicited in 25 Architecture Tradeoff Analysis Method® (ATAM®) evaluations and then by performing an affinity diagram process to group the business goals into categories. For each goal, example scenarios are provided to illustrate how the goal might impact a system. Finally, this report shows how the architecture business cycle (ABC) may be extended by the business goal categorization.				
14. SUBJECT TERMS affinity diagram, business goal, quality attribute, quality attribute scenario, quality attribute scenarios, architecture analysis, ATAM, Architecture Tradeoff Analysis Method			15. NUMBER OF PAGES 38	
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	