a|EA JOURNAL Article

System Architecture Concerns: A Stakeholders' Perspective

By Rashmi Jain, Anithashree Chandrasekaran, and George Elias

Abstract

This article is based on a research on identifying different stakeholders' concerns for system architecture and design. It explores if the different stakeholders' need for system architecture information is related to their concerns. It addresses two important research questions on system architecture descriptions, namely, 1) Do all stakeholders have different needs for information on system architecture concerns? and 2) How much of similarities and differences exist between the stakeholders' need for such information. The authors analyze if the system architecture information needs of the stakeholders' can be addressed by providing different views to different stakeholders. Based on the findings of their research, the authors propose a two-view architecture framework, Summary view and In-Depth view, which can help development projects that are required to generate stakeholder specific architecture views. The findings of the study suggest that the information needs of different groups of stakeholders for system architecture are driven by their own need to get their tasks of system realization completed.

Keywords

Stakeholders, System Architecture, System Architecture Concerns, Architecture Views, System Design

INTRODUCTION

This article is based on our research on identifying different stakeholders' concerns for system architecture and design. The research was also designed to determine if the priorities of different stakeholders are different for information on system architecture and design. The findings of our study suggest that the concerns of different groups of stakeholders for system architecture and design are driven by their own need of what they require to get their task completed. Based on the findings the authors propose a multiple-view framework of system architecture and design that will represent the different stakeholders' concerns on architectural issues.

System architecting is an essential task of systems engineering that lays the foundation for the tasks in the later phases of the systems engineering process. System architecture is a key factor in a system's success or failure, as well as a key cost driver. Thus, identifying means to better understand issues around system architecting will lead to improvements throughout the system lifecycle.

Lately, an emphasis on documentation of system-level architectures has been gaining attention (Bass et al. 1998). A good architecture description helps in the successful deployment of any system. Architectures are developed in the early stages of system development, and knowing as much as possible in the early stages of design would enable effective decision-making and allocation of resources, thus reducing risk. Most likely, an architecture would be frozen during the first phase of the project where much of the project cost is already committed. Creating architecture artifacts and communicating it to all stakeholders before the resources are committed would be a cost saving initiative. Research indicates (Stevens Institute of Technology 2006; Defense Acquisition Guidebook 2006) that over half of the costs are committed at the end of the conceptual & preliminary design phase. Classically, the preliminary design phases end in a preliminary design review, which includes a review of the system architecture's functional baseline and physical allocation (Stevens Institute of Technology 2006). By enabling more effective communications, an architect can prevent misunderstanding and better address the concerns of stakeholders thereby reducing the cost of erroneous decisions based on incomplete information. Better communications are an enabling factor in stakeholders' ability to make critical decisions about architecture. In addition to possibly reducing cycle time, enabling communication between stakeholders may also help to increase the quality of the system by better addressing stakeholder requirements.

Architecture, Architectures Description, Architecture Views

System architecture, a fundamental concept of systems engineering, seems to have as many definitions as there are system engineers. Architecture and the understanding of architecture are fundamental to system engineering. The term system architecture has been variously defined in existing literature. Table A lists some of the prominent definitions.

TERM	DEFINITION
System Architecture	The selection of the types of system elements, their main characteristics, and their arrangements (INCOSE 2004).
Architecture	Logical construct for defining and controlling the interfaces and the integration of all the components of the system (Zachman 1987) .
Architecture	The organizational structure of a system or component, their relationships, and the principles and guidelines governing their design and evolution over time (IEEE 610.12-1990).
Architecture	The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution (Bass et al. 1998).

Table A. Architecture Definitions

The importance of consistent definitions of system and system architecture is as important as the discipline itself. Zachman defines architecture as a "logical construct for defining and controlling the interfaces and the integration of all the components of the system (Zachman 1987) ." This accurately defines architecture in a way that separates the architecture construct from other constructs in systems engineering. Some other definitions of architecture often overlap with the definition of other construct such as architecture description and system.

By accepting architecture as a logical construct we allow one to differentiate architecture from architecture descriptions. Architecture descriptions are "those products of system development that capture architecture information (Bass et al. 1998)." Clearly making the distinction between architecture descriptions and architecture allows for one to address the scope of each of these separately. Whereas architecture descriptions are a form of articulation or communication, architecture is the subject of such communications. Figure 1 depicts this relationship from problem discovery to system instantiation.

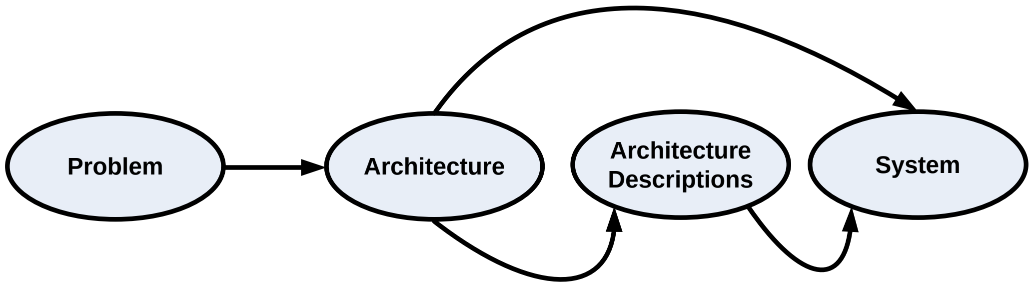

Figure 1. Architectures versus Architecture Descriptions

In Figure 1 we see how architecture may not necessarily be expressed through an architecture description(s). This happens when the system is simple in nature and its architecture can be fully understood, without expression or articulation, and created through the efforts of a single individual. So this may also include mental models in individual minds. All other architectures would tend to be described through different forms of architecture descriptions. It is also possible that an architecture may not be able to be described or instantiated in a system. This may happen when the architecture is not feasible or is not selected as the optimal solution. Nevertheless, architectures can always be expressed through an architecture description. The benefits of using Architecture Descriptions have been discussed in (Bass et al. 1998; Bachmann et al. 2003). Some important benefits of architecture descriptions are analysis of alternative architectures, business planning for transition from a legacy architecture to a new architecture, communications among organizations involved in the development, production, fielding, operation, maintenance of a system, and between acquirers and developers as a part of contract negotiations, and planning and budget support.

Architecture descriptions contain one or more (architecture) views (Bass et al. 1998). Views are “a representation of a whole system from the perspective of a related set of concerns” (Bass et al. 1998). A view is a representation or description of the entire system from a single perspective (Bass et al. 1998). A view refers to a particular architecture of a system (an individual system, a product line, a system-of-systems, etc…). Each view provides a different abstraction of the underlying implementation detail. Therefore, each view is a subset of the detail that exists in the implementation (Baragry and Reed 2001). Architectural views reflect a set of specific interests that concern a given group of stakeholders applied to a specific representation of architecture (Kruchten 1995). A viewpoint is basically a model or set of rules for analyzing, building and interpreting views. A viewpoint perspective can be instantiated through one or more views. These perspectives and concerns are related to stakeholders. Some concerns are expressed as architecture ‘ilities’ or the properties that the architecture or system - past, present, or future – contains/should contain. It is recommended that stakeholder concerns be used to develop system requirements. Inevitably, in complex systems with multiple stakeholders these concerns are often in conflict that can be resolved (Boehm and In 1996). Perspectives or viewpoints are the type of information that the stakeholder is interested in knowing. The viewpoint determines how the view communicates the architecture to the stakeholder (Bass et al. 1998). There is a relationship between stakeholders, their roles, and concerns (Boehm and In 1996) that directly influence the stakeholder’s unique viewpoint (Alexander 2005).

Architecture views are representations of the overall architecture that are meaningful to one or more stakeholders in the system. The architect chooses and develops a set of views that will enable the architecture to be communicated to, and understood by, all the stakeholders, and enable them to verify that the system will address their concerns. An architecture is usually represented by means of one or more architecture models that together provide a coherent description of the system's architecture (TOGAF 2007).

The intent of usage of architecture views is to abstract away some information from the system and yet provide enough information to be a basis for analysis, decision making, and hence risk reduction. The architecture views understand and exploit on the concept that the systems can comprise of more than one structure, and that no one structure holds the irrefutable claim to be the system architecture. Views are comprised of various types of (architecture) artifacts including models, architecture documents, diagrams, and even verbal communications. Some of the frequently used views are discussed in Table B.

Views of Architecture	Reference
Module viewtype, Component-and-connector (C&C) viewtype, Allocation viewtype.	(Bachmann et al. 2003)
“4+1 View” of architecture Logical view, Process view, Physical view, Development view, These four views are depicted with a fifth view that illustrates them with a few use-cases or scenarios.	(Kruchten 1995)
Tu and Godfrey added one more view to Krutchen’s 4+1 view called Build-Time view.	(Tu and Godfrey 2001)
Module structure, Conceptual or logical structure, Process structure or co-ordination structure, Physical structure, Uses structure, Calls structure, Data flow, Control flow, Class structure.	(Bass et al. 1998)]
All view, Operational view, Systems view, Technical view	(Carlomusto 2005)
Conceptual architecture view, Module architecture view, Code architecture view, Execution architecture view.	(Hofmeister 2000)
Business architecture views, Data architecture views, Applications architecture views, Technology architecture views.	(Bass et al. 1998)

Table B. Architecture Views

Views within an architecture description need to be consistent with one another in order to convey the architecture information accurately. Inconsistencies within architecture descriptions should always be avoided. But if they exist they are required to be documented (Bass et al. 1998). A good architecture description also addresses all stakeholder concerns and perspectives through at least one view 17 (Bass et al., 1998). This requires careful attention to all the stakeholders of architecture, and investigating who these stakeholders are and what they will require of the architecture descriptions. Stakeholders always include the users and developers, but often include financers, maintainers, investors, etc. Using a broad-view of stakeholders and paying attention to their needs will insure that architecture is addressed in a comprehensive manner.

ROLE OF STAKEHOLDERS

Determining the relevance of different stakeholders for any given system is one of the first steps in the systems engineering process. How one determines the relevant stakeholders and how the requirements are interpreted shape the design and ultimately the system. Stakeholders can be in the form of an enterprise, organization, or an individual having an interest or a stake in the outcome of the engineering of a system (ANSI/EIA 1999). Each system has a unique set of stakeholders driving the design and purpose of each system. A search through the literature provides numerous lists of recommended stakeholders or minimum sets of stakeholders (Bass et al. 1998; Clements 2000; Kazman 2000; SEI 2001; Bachmann 2001; IEEE 2004; ANSI/EIA 1999).

For the purposes of this article, we are not interested in determining the specific stakeholders of a particular system, but the stakeholder groups related to (or associated with) system architecture descriptions. Stakeholder groups are an acceptable alternative (IEEE 2004) to defining specific groups of stakeholders representing a set of architecture descriptions and are appropriate for the purpose of our study. In determining the stakeholder groups, it is necessary to create a list that comprehensively covers the majority of stakeholder groups. Understanding these intended audiences for architecture descriptions is very important (SEI 2001). An architecture description that is comprised of multiple architecture views enables communication to the various stakeholders (TOGAF 2007). Identifying these stakeholders and their unique relationship to the system will help determine how to document and communicate architecture.

Even though there are many lists of stakeholders in the literature, often these lists have certain common stakeholders. IEEE 1471-2000 lists Users, Acquirers, Developers, and Maintainers as a minimum set of stakeholders for any system architecture (Bass et al. 1998). This list of stakeholders is not complete, and is applicable as a minimum set for only software intensive systems (ANSI/EIA 1999). The most frequent software-system stakeholders are: users, customers, developers, maintainers, interfacers, and the general public, while there are other less common stakeholders: product-line managers, testers, and subcontractors (Boehm and In 1996). More complex systems can have many more types of stakeholders in comparison to less complex systems (Boehm and In 1996). Other kinds of systems such as a house, have a different set of stakeholders for its architecture documentation. These are Planner, Owner, Designer, Builder, and Subcontractor (Zachman 1992). Given these two examples, it would seem impossible to generate a comprehensive list of stakeholders that would apply to all kinds of systems.

Our goal is to create a list of stakeholder groups that contains the majority of stakeholders who would be using the architecture description communicated through the different artifacts. We included stakeholders that are classified as primary, secondary, and tertiary in their system interaction (Alexander 2005). Different stakeholder groups have unique information needs for architecture description because of how they relate to the system (Zachman 1987). Included in these stakeholder groups are surrogate stakeholders (Alexander 2005) who by nature would have overlapping concerns and information needs. Each stakeholder group for the purposes of this study contains similar groups of stakeholders that relate to the system in a very similar manner, thus having communication needs for system architecture information that is similar in nature. Table C lists the stakeholder groups for this study
.

Stakeholder Group	**Definition**
End-User	This stakeholder is the primary operator of the deployed system.
Designer	Personnel involved in the process of defining, selecting, and describing solutions (conceptual, preliminary, or detailed) of the system, system elements or system end-items to requirements in terms of products and processes.
Builder/Developer	This stakeholder uses the architecture and requirements to create the system or a portion of the system.
Buyer/Acquirer/Financer	This stakeholder manages the acquisition process and financing.
Tester/Integrator	This stakeholder is primarily responsible for verifying, validating and/or integrating the system or system components.
Team Lead/Manager	The stakeholder responsible for directing and coordinating resources (labor, materials, etc) throughout the life of a project to achieve predetermined objectives such as scope, quality, time, cost, and stakeholder satisfaction.

Table C. Stakeholder Groups

Architectural Views,
Addressing the Concerns

Architectural views allow separation of concerns. Views are at the foundation of architectural documentation and are a mechanism introduced to handle complexity in the architecture by improving separation of concerns (Bouck'e and Holvoet 2005). Each view emphasizes certain facets of the solution as clear and concise as possible while deemphasizing and ignoring other facets. A concern is an area of interest or focus in a system. Architectural concerns are defined as those concerns that significantly influence the architecture (Bouck´e and Holvoet, 2005). Concerns are the key interests that are crucially important to the stakeholders in the system, and determine the acceptability of the system. Concerns may pertain to any aspect of the system's functioning, development, or operation, including considerations such as performance, reliability, security, distribution, and evolvability (TOGAF 2007). Dealing with concerns having a high impact on the gross-level structures and quality attributes has to be done during architectural design, leading to an architecture-centric approach (Bouck'e and Holvoet 2005). Since there are several sets of architectural concerns available to an architect, the architect will end up using a limited number of the sets (the headlines amongst the architectural concerns) to structure the architectural models and documentation. This set of high priority architectural concerns, called architectural drivers have a broad influence on the architecture, drive the architect while defining its architecture and make up large chunks of the architectural documentation. (Bouck'e and Holvoet, 2005).

A given system architecture has a finite number of concerns addressed. Each stakeholder of the architecture will be interested only in a subset of the addressed concerns. Architecture views help stakeholders to look at that subset of interest as opposed to the finite universal set of concerns. The authors building upon the literature on architecture and the subject matter experts' opinions believe that a subset of architectural concerns can be mapped to each group of stakeholders for any given system. These subsets will not be mutually exclusive of each other. This mapping will further help in developing views that address only the given subset of concerns and hence provide a stakeholder group specific views. The scope of this research is to develop such a mapping of some common architecture concerns to some common stakeholder groups and propose a framework that can used by projects to provide the relevant architecture information to the different stakeholders that will address their concerns. An example of specific views corresponding to different group of stakeholders from prior work is shown in are Figure 2. A mapping of these views to the schema of the well-known Zachman Framework is also illustrated in (TOGAF 2007). Zachman was one of the pioneers of mapping architecture representations to stakeholders based on each stakeholders' needs. He highlights the fact that each architecture representation differs from the other by level of detail. He states that the level of detail is an independent variable, varying within any one architecture representations. Each representation is the reflection of the stakeholders' perspective. Since there are different perspectives for each architecture, there are different representations for architecture. Also, he introduces the idea of different types of descriptions for the same object. So he states that the same product can be described, for different purposes, in different ways, resulting in different types of descriptions. The combination of the two ideas suggests that for every different type of description, there are different perspectives for each of the different participants. Architecture representations are additive and complementary. Improving professional communications and developing improved approaches to produce each of the architecture representations are some benefits of the framework.

Literature on architecture discuss about various types of concerns. Some of these credible literatures are referenced throughout this article. For this research an exhaustive list of architecture concerns were consolidated based on these literatures. The authors then chose 28 architecture concerns that commonly interest the above discussed stakeholder groups. These architecture concerns can also be referred as architecture drivers in most cases and are given higher priority by architects. The decision of choosing these 28 concerns as commonly addressed were based on the literature review, brainstorming sessions, and understanding and experiences of the authors on the topic. These 28 concerns and their descriptions are shown in Table D.

Survey on Architectural Concerns by Stakeholder Groups

The authors conducted a survey to determine different stakeholder concerns about system architecture. We asked respondents to describe a system of interest and identify their stakeholder relationship to their described system. Stakeholders were then asked to rate their information need of each concern based on the following scale:

Level 1: No information is required about the concern. (As an example, no information is required about the performance requirements of the system).

Level 2: Top level or abbreviated information is required about the concern. (Need to know if any performance requirements have been identified).

Level 3: Some detailed information is required about the concern. (Need information on all performance requirements that were identified).

Level 4: Significant amount of detailed information is required about the concern. (Need information on how the performance requirements are addressed in the architecture).

Level 5: Complete and detailed information is required about the concern. (Need information on how the performance requirements are addressed in the architecture in terms of decomposition, modularity, interdependence etc. and how they impact other aspects of architecture and architectural decisions)

<table>
<tr><td colspan="4">To address the concerns of the following stakeholders...</td></tr>
<tr><td>Users, Planners, Business Management</td><td>Database Designers and Administrators, System Engineers</td><td>System and Software Engineers</td><td>Acquirers, Operators, Administrators, & Managers</td></tr>
<tr><td colspan="4">... the following views may be developed</td></tr>
<tr><td>Business Architecture Views</td><td>Data Architecture Views</td><td>Applications Architecture Views</td><td>Technology Architecture Views</td></tr>
<tr><td>Business Function View</td><td rowspan="5">Data Entity View</td><td rowspan="5">Software Engineering View</td><td rowspan="4">Networked Computing/ Hardware View</td></tr>
<tr><td>Business Services View</td></tr>
<tr><td>Business Process View</td></tr>
<tr><td>Business Information View</td></tr>
<tr><td>Business Locations View</td><td rowspan="3">Communications Engineering View</td></tr>
<tr><td>Business Logistics View</td><td rowspan="4">Data Flow View (Organization Data Use)</td><td rowspan="4">Applications Interoperability View</td></tr>
<tr><td>People View (Organization Chart)</td></tr>
<tr><td>Workflow View</td><td rowspan="2">Processing View</td></tr>
<tr><td>Usability View</td></tr>
<tr><td>Business Strategy and Goals View</td><td rowspan="5">Logical Data View</td><td rowspan="5">Software Distribution View</td><td rowspan="3">Cost View</td></tr>
<tr><td>Business Objectives View</td></tr>
<tr><td>Business Rules View</td></tr>
<tr><td>Business Events View</td><td rowspan="2">Standards View</td></tr>
<tr><td>Business Performance View</td></tr>
<tr><td></td><td colspan="3">System Engineering View</td></tr>
<tr><td colspan="4">Enterprise Security View</td></tr>
<tr><td colspan="4">Enterprise Manageability View</td></tr>
<tr><td colspan="4">Enterprise Quality of Service View</td></tr>
<tr><td colspan="4">Enterprise Mobility View</td></tr>
</table>

Table D. Example Taxonomy of Architecture Views (TOGAF 2007)

RESEARCH DEMOGRAPHICS

The survey was distributed to different groups of stakeholders of system architecture. In order to facilitate their familiarizing and relating with the concepts of system architecture the respondents were asked to describe in a few lines about the system of interest that they are going to respond to the survey about. This also gives the context of the system architecture to both the participants and the researchers. Describing the system helps the participants in identifying which of the given group of stakeholders they belong to.

The surveys were either emailed or personally handed as hard-copies to the participants. The participants were not given a time limit to fill in the survey. The completed surveys were returned to the research team by the respondents through emails, faxes, and in person. The survey had demographic questions in the beginning followed by an open ended question to describe a system that they are a stakeholder for. This question was asked to provide context to the survey and to limit the respondent from alternating between systems or the stakeholder roles. Then the respondent was asked to map his or her stakeholder role to one of the listed 6 stakeholder groups. Once the respondents identify the system and their role as a stakeholder they were then asked to rate the above discussed 28 concerns based on their information need.

There were 14 different organizations that participated in the survey from different industry domains. These were defense, pharmaceutical, aerospace, academics, finance, software, and electronics. The 23 respondents who provided completed surveys ranged from 3 to 44 years of professional experience. The stakeholder groups, Builder/Developer, Buyer /Acquirer/Financer, Designer, End User, Team Lead/Manager, and Tester/Integrator, were represented by 4, 3, 3, 5, 5, and 3 respondents respectively. The diversity in demographics of survey participants helped us control the impact of industry or experience bias to some extent. The varying levels of system complexity associated with the different industries represented in this research helps address the diversity of the participants' information need for the architecture concerns.

LIMITATIONS OF THE RESEARCH METHODOLOGY

There are some severe and some moderate limitations to the methodology adopted for this research, which may be threats to its validity. The most significant limitation is the size of the sample. Our findings are based on responses from 23 respondents responding on 28 different architecture concerns on a system that they were involved with as a stakeholder as defined in this article. The size of the sample may be constraining in representing the views of the larger population. However, relevant to the topic of our research we were more focused on selecting the 'right' people who can be truly representative of the stakeholder group in terms of their experience and knowledge. The research did not control for the experience, education, type of system, type of industry domain, etc. which may have contributed as confounding variables.

ARCHITECTURE INFORMATION NEEDS AMONG ALL STAKEHOLDERS

Stakeholder Groups with the Most and Least Information Needs

Overall, most stakeholders (77%) responded positively to all concerns requiring at least a level 2 information need or higher. Only the Builder / Developer stakeholder group responded with a low need (level 1) for information on Service Level Agreements (SLA) and ergonomics.

The two stakeholder groups with the highest information needs are the Tester/ Integrator and the Designer with an average information need of 3.96 and 3.83 respectively. This contrasts with the other two groups of stakeholders with the lowest information needs which are the Buyer/Acquirer/Financer and the End-User with an average information need of 3.27 and 3.25 respectively. We interpret this to mean that on average stakeholders with the highest information needs require at least a significant amount of information (Level 4) across most concerns and stakeholders with the lowest average information need only require some information (Level 3). In an architecture description this can be the difference between just listing details and giving detailed information. For example, a list of PC interfaces might be: USB, Bluetooth, SVGA, etc; a list of the details for these interfaces might describe the protocols or characteristics of the physical layers.

This finding makes good common sense because the End-User certainly does not need the same detailed information as the Tester/Integrator.

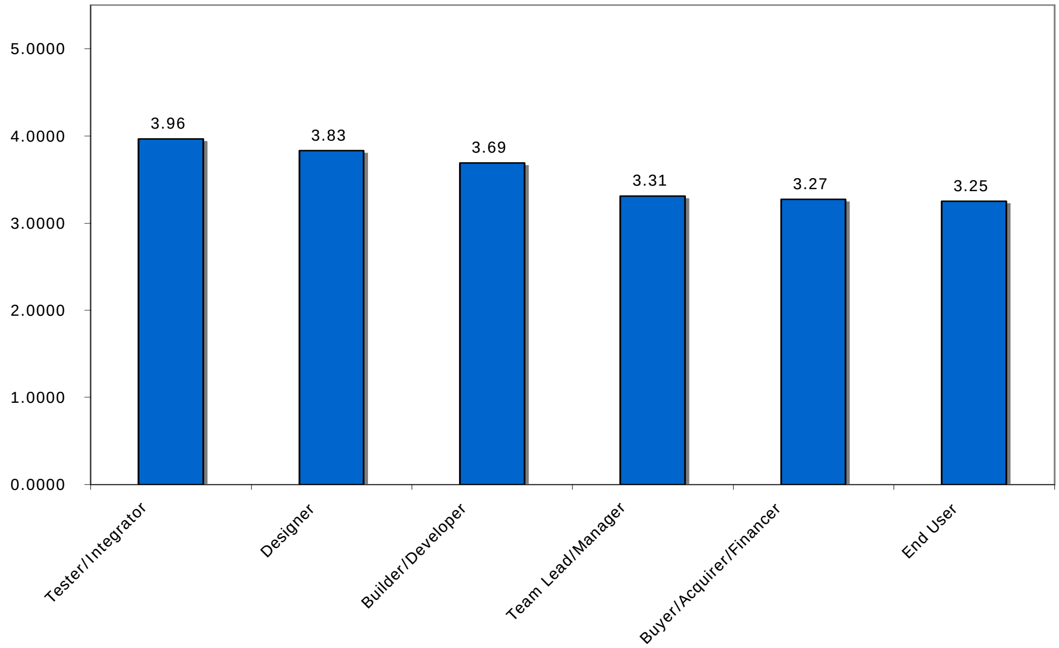

Table E. Mean Level of Information Need by Stakeholders

Architecture Concerns Requiring Highest and Lowest Levels of Information

Respondents rated each concern based on their information need. Overall, stakeholders said that they need complete (Level 5) or significant (Level 4) information about the system appropriateness with 75% (18 out of 23 respondents) of responses at these levels. Other concerns with high information needs among stakeholders are: integration, purpose, scope, functional capabilities, external interfaces, etc.

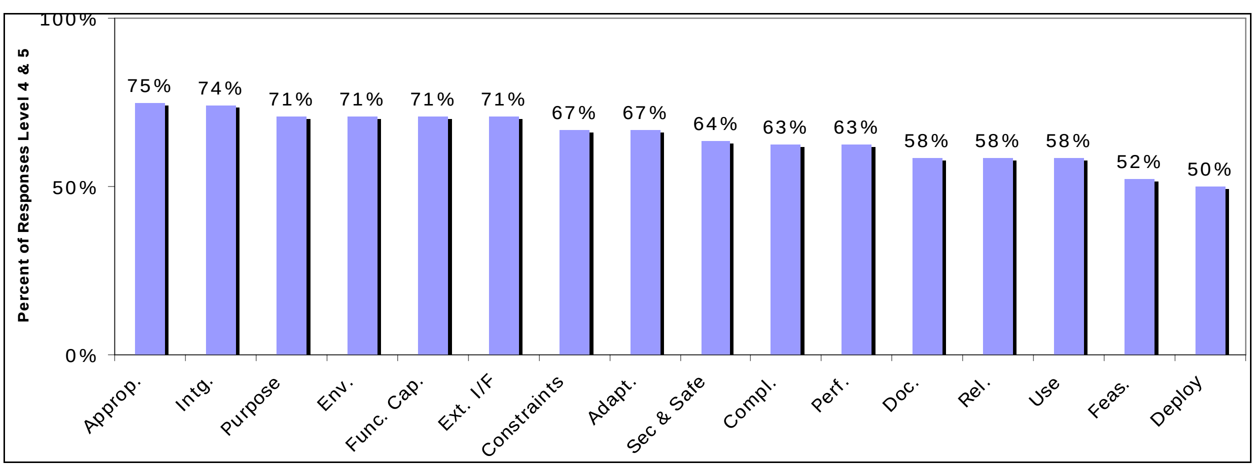

Table F. Highest Information Needs Among Stakeholders

Similarly, certain concerns were rated with a low information need. Overall, stakeholders said that they need top level or no information about the system retirement with 64% of responses at level 1 or 2. Retirement was by far rated lower than any other concern with the next lowest being 22% away (9 out of 23 respondents). This may have to do with attitudes towards planning for later phases in the lifecycle.

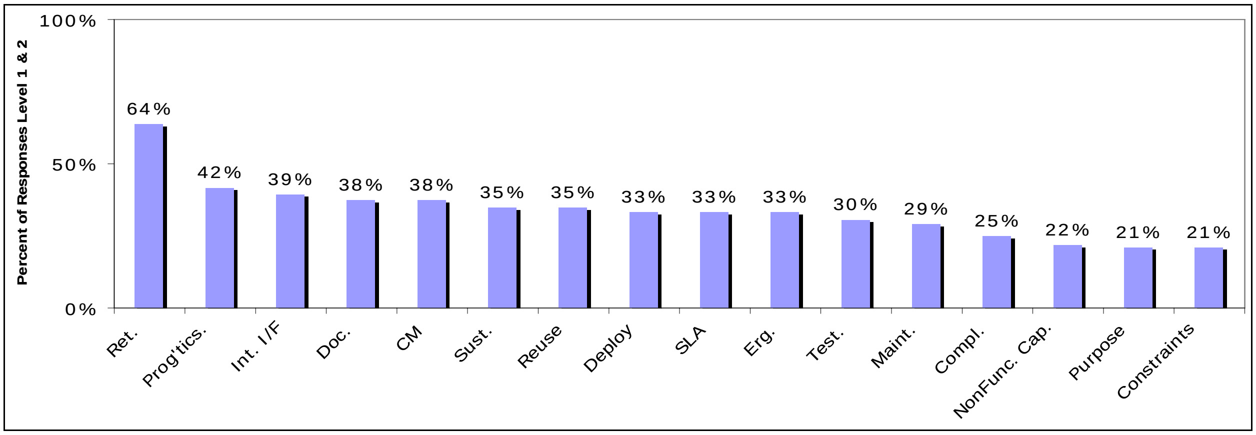

Table G. Lowest Information Needs Among Stakeholders

The next series of pareto charts describe the information needs in *descending order* by stakeholder category. We plotted the average stakeholder response within these pareto charts in order to display and understand the uniqueness of the individual stakeholder category's information need. We found that the stakeholder groups did not disagree on all points. System appropriateness was rated high by most stakeholders. Stakeholders needing information on system appropriateness show that they are all concerned with the system's ability to accomplish it's mission. Even though one may argue that not all stakeholders need to know this, there may be a human response where a stakeholder needs to feel that the system is good. This can be related to system acceptance and how it is easier for one to work with a system once they understand it's purpose and how the system fulfills it's mission. Most of the top 20% of the overall concerns can be explained in this way, and are common concerns among most stakeholders. Conversely, the bottom 20% of concerns seems to be related to only a limited number of stakeholders. For instance, most stakeholders reported a relatively low information need on service level agreements, reuse, programmatics, sustainment, configuration management, ergonomics, reuse, and retirement.

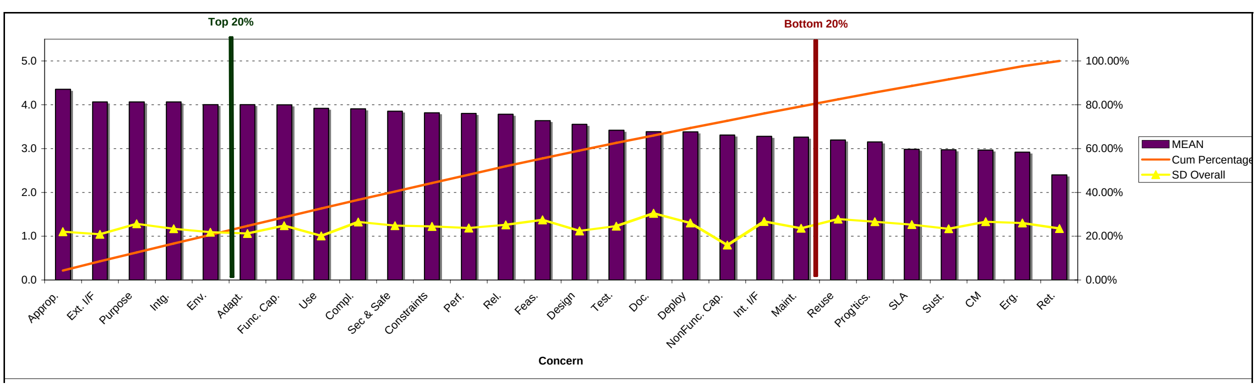

Table H. Overall Stakeholder Information Need

By far, stakeholders *disagreed the most* on the need for information on system design documentation. Again, these responses make sense because stakeholders would have various needs for the design documentation. For some stakeholders the design documentation is necessary for them to fulfill their role, so it would make sense for them to have a high information need for these.

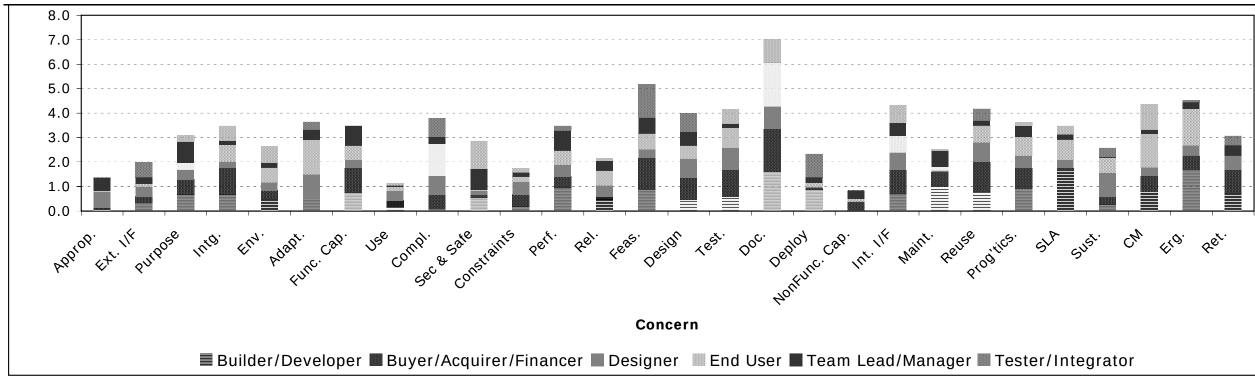

Table I. Average Stakeholder Response Difference From Overall Average

For the Builder/Developer, one of the most notable findings is the low information need on programmatics. Programmatics refer to issues related to project schedule, cost, risks, etc. Having a low information need for programmatics may explain the typical cost and schedule over-runs when building and developing systems that we often come across. In addition to their low information need on programmatics, the Builder/Developer group of stakeholders differed most from the average stakeholder. The data shows that they were more polarized in their information needs than the average. This likely has to do with the nature of their relationship to the system, and that building a system right requires significant amount of information about certain aspects of the system. A reaction to this high information need may cause this stakeholder to reduce information in areas they do not want to be concerned with.

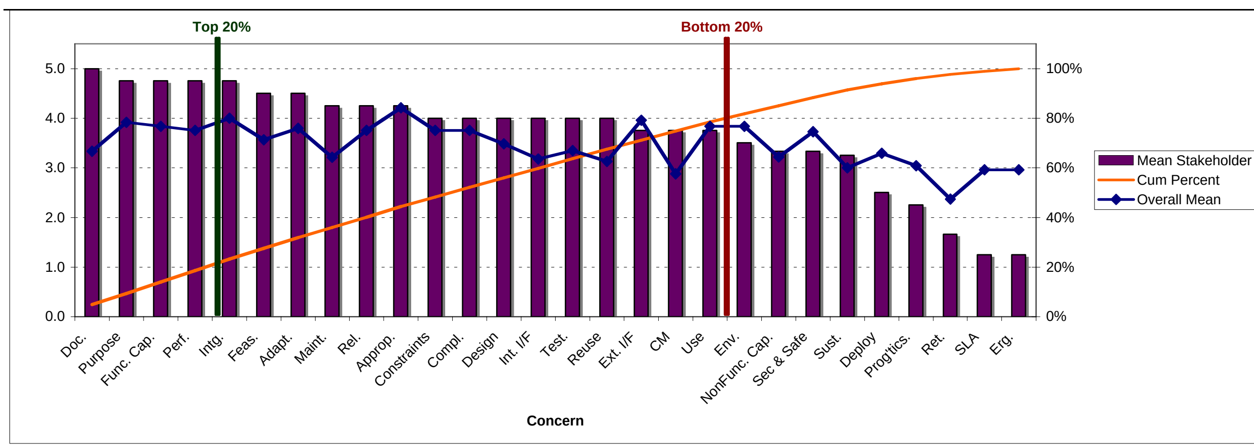

Table J. Builder/Developer Pareto Chart

The Buyer/Acquirer/Financer group demonstrated a higher information need for some concerns that may be termed as technical in nature such as external interfaces, adaptability, security and safety. By traditional norms of the role of Buyer/Acquirer one would have expected programmatics to be important to this group. Programmatics are the management aspects of the project. These aspects consider the cost, schedule, and risk of building, fielding, and supporting the solution. An interest and need for information on these technical aspects that impact the deployability of the system or restrain it may be pointing towards a new and evolving role of this stakeholder group which not only puts accountability on them for programmatics but also the successful implementation of the system.

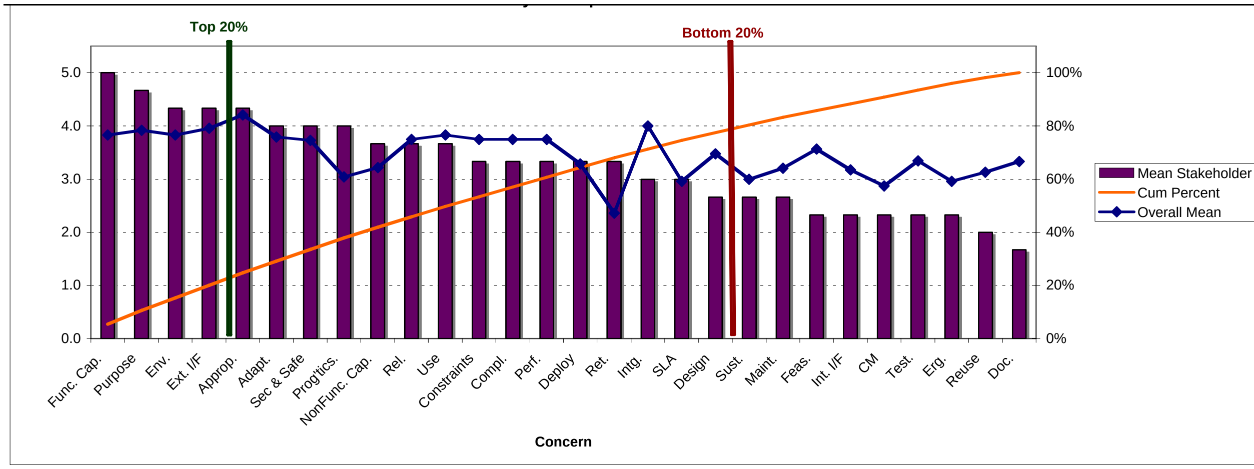

Table K. Buyer/Acquirer/Financer Pareto Chart

The Designer is largely involved in the beginning phases of system development. This stakeholder would naturally be interested the system compliances, constraints, environment, and scalability as indicated by the responses. The responses on the top level seem to follow this pattern where the Designer had high information needs for the concerns associated with initial system design questions.

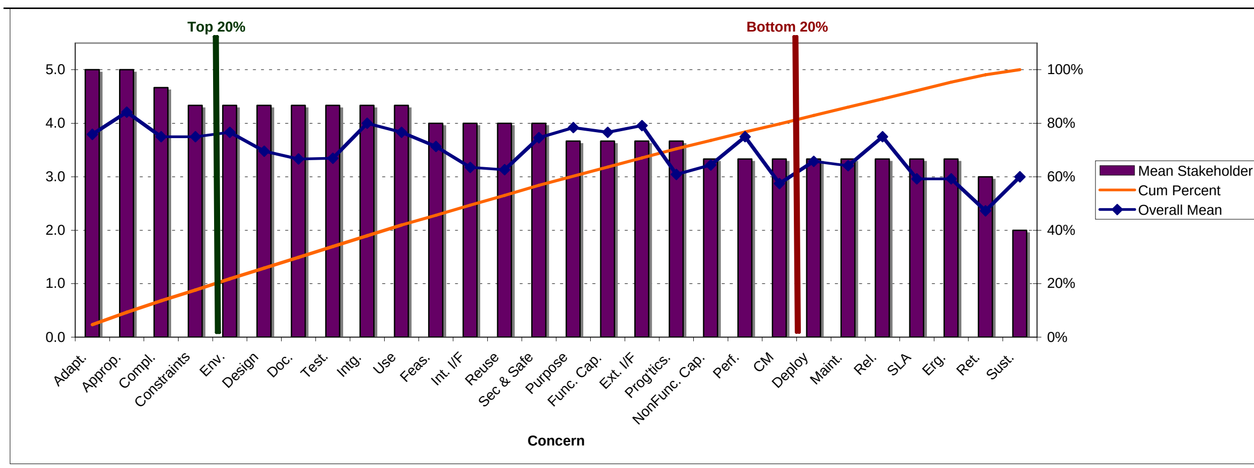

Table L. Designer Pareto Chart

The End-Users group reported high information needs on concerns that would be associated with stakeholders involved in the design. This may reflect the End-Users' desire to know everything about the system even-though they may not directly use this information. For the End-User, knowing this information may help their comfort level with the system, and providing this information may help in system acceptance.

On the other hand, the authors would have expected for the End-Users to have a high information need on SLAs. In reviewing the responses, it was noticed that overall there is a high level of disagreement among End-Users on the level of information they need on concerns which is reflected in a high standard deviation of 1.2 in responses. A similar level of difference in responses was noted for the Team Lead/Manager.

We believe that the significant level of differences in these two stakeholder groups is likely a result of either or a combination of two factors. One, the type of system largely affects the stakeholder response for information. In other words, the information needs for some concerns are primarily dependent on the system rather than the stakeholder group. The other reason for such a wide variety of responses is the individual's idea of what the different levels of information may mean. For instance, most End-Users of cars may be surprised if they received detailed and complete performance information on their automobiles. For example, this type of detailed and complete information may include information on now the different components of the exterior of the car contribute to the overall drag coefficient. This information is rarely needed by the End-User and most likely would be unintelligible to most.

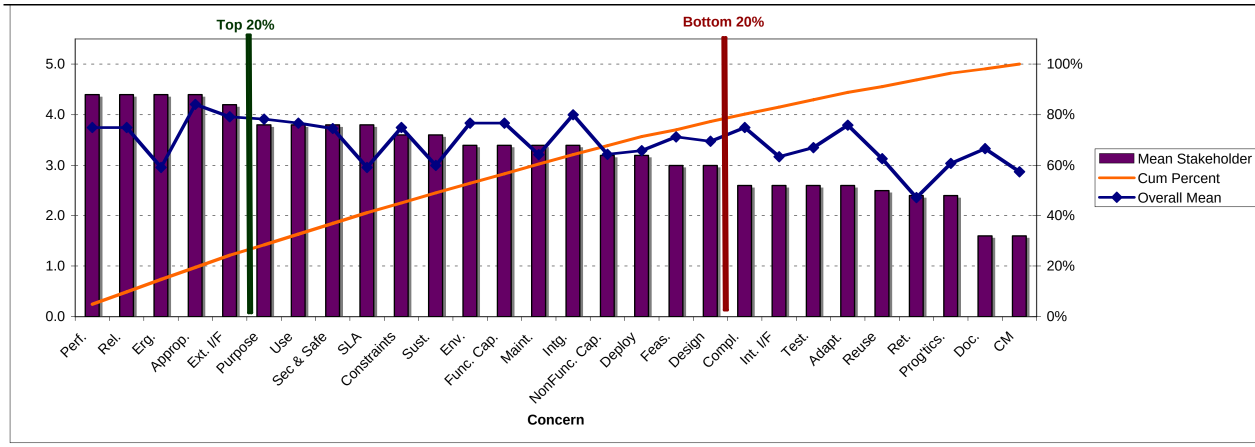

Table M. End-User Pareto Chart

One of the most interesting findings for the Team Leader / Manager group is that they reported lower than expected information need on programmatics. One would expect for programmatics to be in the top twenty percent of Manager's concerns; however the responses were only in the top fifty percent. They reported high information need on integration, compliance and constraints implying that these Team Leaders / Managers have a technical role and are concerned about more than just programmatics.

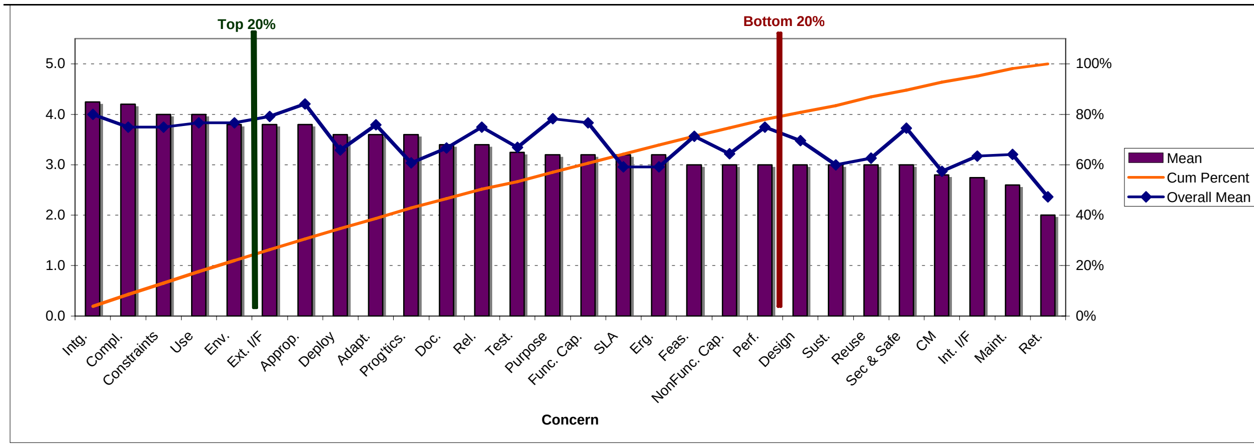

Table N. Team Lead/Manager Pareto Chart

For the Tester/Integrator, one of the most interesting finding is their much above average need for information on system feasibility. In fact, they are more concerned about feasibility than any other stakeholder. This may be due to how we defined feasibility: "The ability to determine an adequate qualification method for demonstration that the system satisfies the requirement." The term 'qualification" (to mean testing) may have prompted this response from the Testers.

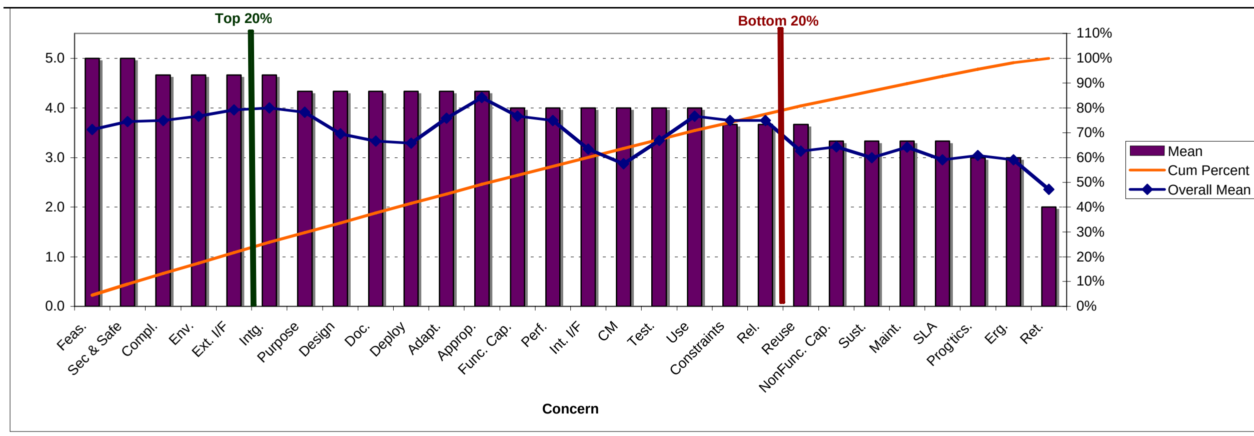

Table O. Tester/Integrator Pareto Chart

Overall Similarity Among the Stakeholder Groups

The observations on information need for each stakeholder group shows that there are some groups of stakeholders with very similar information need and some others with high variability between them. The similarity in the need for information on architectural concerns can be attributed to the role of each stakeholder from system conception to operation, the degree of interactions of each stakeholder with the system, and the direct and indirect benefit/consequence to each stakeholder as a result of system development and its operation. To understand the trend (similarity vs. variability) of information need correlation analysis was performed on the information need of each stakeholder group across 28 different concerns. The results of the correlation study are shown in

	Builder/ Developer	Buyer/ Acquirer/ Financer	Designer	End-User	Team Lead/ Manager	Tester/ Integrator	Overall
Builder/Developer	1.000						
Buyer/Acquirer/Financer	.065	1.000					
Designer	.444*	.128	1.000				
End-User	-.077	.461*	-.149	1.000			
Team Lead/Manager	.229	.324	.562**	.231	1.000		
Tester/Integrator	.606**	.157	.571**	.061	.502**	1.000	

*Correlation is significant at the 0.05 level. **Correlation is significant at the 0.01 level. N= 28.

Table P. Correlations among the stakeholder groups on their required information level about the given set of architecture concerns

The six groups of stakeholders aligned themselves in 3 pairs based on the correlation results in Table P. The 3 pairs are 1) Builder/Developer and Tester/Integrator, 2) Designer and Team Lead/Manager, and 3) Buyer/Acquirer/Financer and End-User. In the Designer and Team Lead/Manager pair, Designer is closely correlated with the Builder/Developer and Tester/Integrator pair than the Buyer/Acquirer/Financer and End-User whereas Team Lead/Manager has relatively high correlation with the later than the former. This pairing of stakeholder groups is in line with the commonalities in the roles of each of these system stakeholders.

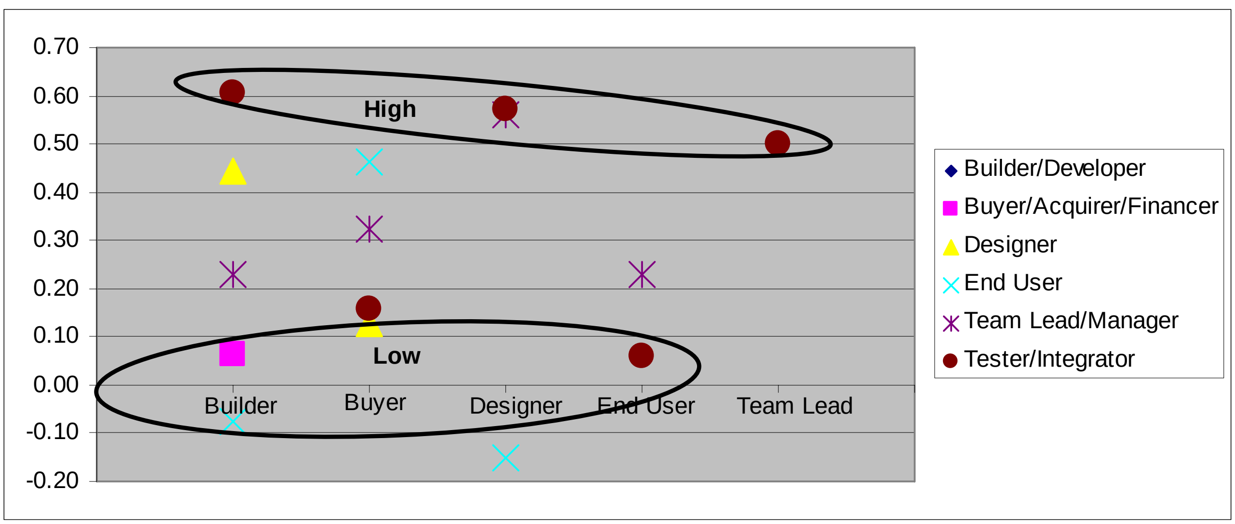

Table Q. High and low correlations among the stakeholder groups on their required information level about the given set of architecture concerns

The high (> 0.5) and low (<0.15) correlations among the stakeholder groups are shown in Table Q. End-User group has relatively low correlation with other stakeholder groups with the exception of Buyer/Acquirer/financier group. This group indeed has a negative (low) correlation with the Builder/Developer and Designer.

Stakeholder Groups With Very Similar Needs for Information

The two pairs of stakeholder groups that are highly correlated are: 1) Builder/Developer and Tester/Integrator, and 2) Designer and Tester/Integrator. The trends of these 3 stakeholder groups in the required level of information across all 28 architecture concerns are shown in

. When the trends for each pair are observed independently they show high degree of similarity (correlation). Each of the highly correlated pair is discussed here briefly to highlight the likeness and the variance.

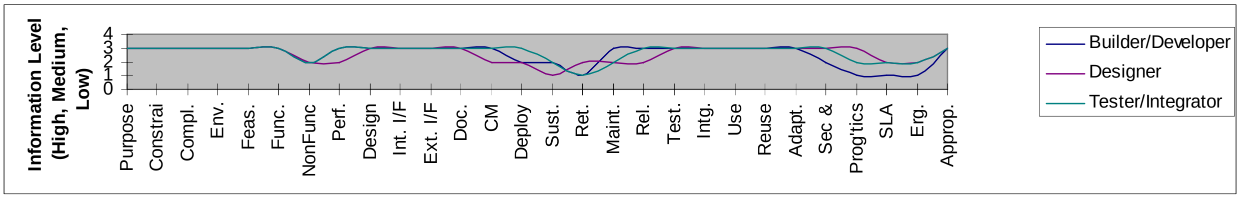

Table R. Trends of Stakeholder groups in their required information level across the architecture concerns

Table R shows how the two pairs, Designer and Tester/Integrator, and Builder/Developer and Tester/Integrator are highly correlated. The higher degree of similarity and less variances in the required level of information on all 28 architecture concerns supports the finding that these two pairs are correlated in their need for information. *Designer and Tester/Integrator* have similar need for information on about half of the 28 concerns and vary by only one level in the remaining half of the concerns. Though *Builder/Developer and Tester/Integrator* have the high level of similarities they do vary by more than one level (by two levels) on the required amount of information on 3 concerns namely, *security and safety, SLA, and ergonomics*. On these 3 concerns the Tester/Integrator group requires more information than the Builder/Developer. These variances across these 3 concerns are as expected. As a Builder or Developer the need is to build a system with the given specification. These specifications incorporate these concerns with more emphasis on security and safety than the other two. Hence top level information would be sufficient for Builders/Developers. But as a Tester/Integrator the need is to test against the requirements for these 3 concerns. Hence at least some information (at least the requirements) more than just the top level information is required on these 3 concerns with more emphasis or details on security and safety than the other two.

Stakeholder Groups with High Variability in Their Need for Information

The three pairs of stakeholder groups that have low correlations as shown in
 are namely, 1) Tester/Integrator and End-User, 2) Buyer/Acquirer/Financer and Builder/Developer, and 3) Builder/Developer and End-User. The analysis of the mean level of information need by stakeholder group also highlights the variances across the high level of information need by the Tester/Integrator and Designer, and the low level of information need by End-User and Buyer/Acquirer on most of the architecture concerns. The trends of these 4 stakeholder groups in the required level of information across all 28 architecture concerns are shown in
. When the trends for each pair are observed independently they show high degree of variance (correlation). Each of the poorly correlated pair is discussed here briefly to highlight the variance and the likeness. The high degree of variance among these stakeholder groups in their need for information across the 28 architecture concerns is highlighted by the data provided in Table U.

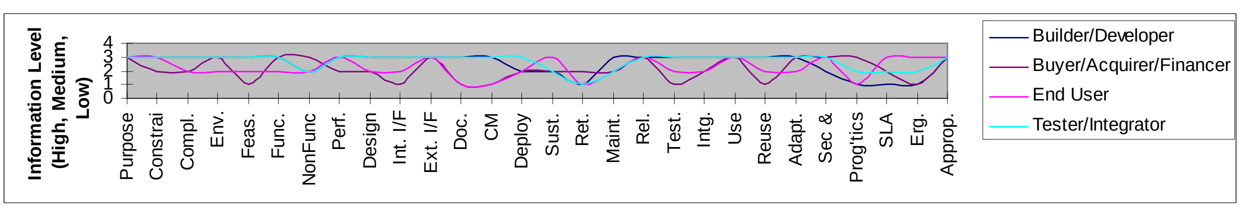

Table S. Trends of Stakeholder groups in their required information level across the architecture concerns

Tester/Integrator and End-User
These two groups differ highly in the required level of information about the following 5 concerns, namely, c*ompliance, scope, design documentation, configuration management, and integratability*. The Tester/Integrator is more interested about these 5 concerns than the End-User. These variances across these 5 concerns are as expected because the criticality of these 5 concerns on the effectiveness of testing and integration is very high requiring detailed information. But as an End-User the interest lies in just the information on what requirements are related to compliance, scope, and integratability and the knowledge of existence of design documentation and configuration management.

Buyer/Acquirer/Financer and Builder/Developer
The two groups differ significantly (3 degrees or level of information) in their need for information on *system feasibility and design documentation*. The Buyer/Acquirer/Financer is less interested in system feasibility and design documentation than the Builder/Developer. The two groups also differ highly (two degrees) in the required level of information about the following 7 concerns, namely, *internal interface, configuration management, testability, reusability, programmatics, SLA and integratability*. The Buyer/Acquirer/Financer is less interested in these concerns except SLA and programmatics than the Builder. This variance is as expected because as a Buyer/Acquirer/Financer the interest in programmatics and SLA is higher due the accountability associated with these two concerns. The Builder/Developer pareto analysis in the earlier section also discusses the reasons on the low level of information need on programmatics.

Builder/Developer and End-User
The two groups differ significantly (3 degrees) in the need for information on *SLA, ergonomics, and design documentation*. The End-User is more interested in SLA and ergonomics, and less interested in design documentation than the Builder. This variance is as expected because an End-User has high interest in ergonomics and SLA. An End-User requires understanding of all the services that will be received while the system is being developed and during its operational lifecycle. The End-User will also be more interested in understanding the human interfaces and aesthetics of the system as these will impact his/her effectiveness.

The two groups also differ highly (two degrees) in the required level of information about the following 3 concerns, namely, *feasibility, configuration management, and integratability*. The Builder/Developer is more interested in these concerns than the End-User. These variances across the 3 concerns are expected because the criticality of these 3 concerns on the development success is very high. But as an End-User the interest lies in just the information on what requirements are related to feasibility, and integratability and the knowledge of existence of configuration management.

PROPOSING MULTIPLE VIEWS: CATEGORIZING STAKEHOLDERS' CONCERNS

On analyzing the responses from the different groups of stakeholders we were able to identify the architecture concerns that were specific to each of these groups. These have been discussed in the earlier sections of this article. Based on our analysis of how the different groups of stakeholders differed on their concerns and need for information on those concerns we extracted some patterns and commonalities among some of these stakeholder groups. These patterns helped us create a framework that would propose the relevant concerns and related need for information on these concerns for each group of stakeholders.

Our first step for a proposed framework was to categorize all architecture concerns into four "Cs" groups, namely, context, constraints, contractuals, and characteristics. Context related concerns are those that would be related to the system as a whole and its suitability within its environment. These are those related to system purpose, its scope, design (including internal interfaces), and its appropriateness to its mission. Constraints are those concerns without addressing which the system will cease to exist. So these are in the nature of regulations and other constraining system capabilities that have to be addressed in order for the system to operate. These are: compliance requirements, external interface design issues, reliability, integratability, usability, scalability, security and safety etc. Contractual concerns are as the name suggests those that are related to terms agreed and binding in the different types of contracts and MoUs. They are also constraining in nature for the system design and as a result there may be some overlap between contractuals and constraints related concerns. It will depend on the project management to determine the category of a certain system design concern as to whether it is a constraint and contractually binding requirement. These may be concerns related to CM, deployment, sustainment, retirement, maintainability, reusability, cost, schedule, risks, or those that are contained in the SLAs. Characteristics of the system or system features including both the functional and non-functional features of the system form the basis of this category of concerns. These may be those constraining the system functionality and capability in terms of technology, technical feasibility, functional capability, non-functional capability, performance, design documentation, testability, and ergonomics. A mapping of the architecture concerns to these four categories is shown in Table V.

In our next step we analyzed the distribution of the stakeholders concerns based on the four Cs in terms of 'complete' (level 5), 'significant' (level 4), 'some' (level 3), and 'no' and 'abbreviated' (level 1 and 2) need for information on architecture. These information levels were then converted into percentage of complete information. For example level 5 translates to 100% complete information about the concern is required by the stakeholder. Similarly, level 4 translates to 70%, level 3 to 40%, and level 1 and 2 to 10% of the complete information about the concern. Given this scale and the mapping of concerns to the 4 Cs, the percentage of complete information required by the stakeholder groups on each 4 Cs was derived. In other words the priority of need for information for each stakeholder group on each of the 4 Cs was derived. The alignment of concerns and their priority for different stakeholder groups is illustrated in the figure below. The 4 axis in the figure represent the percentage of complete information required by the stakeholder for each of the 4 Cs. The data table for the 4Cs radar chart is shown in Figure 2.

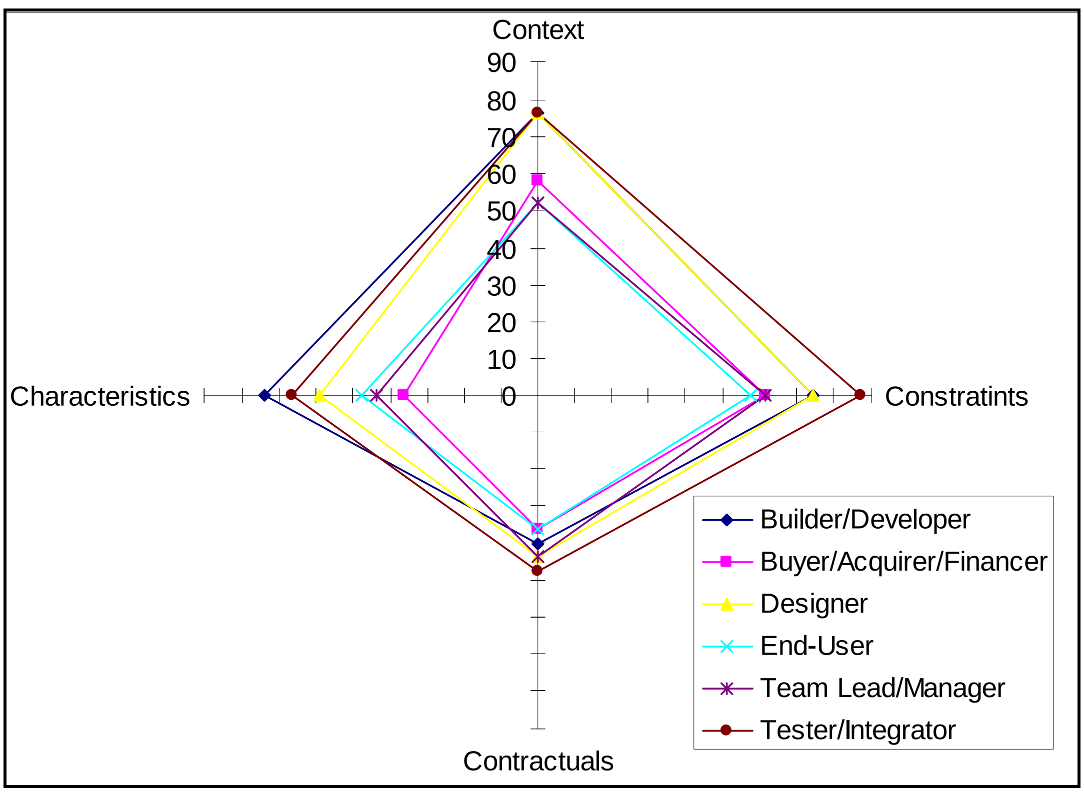

Figure 2. 4Cs Radar Chart for Stakeholder Groups

Figure 2 clearly shows an alignment of the architecture concerns on the four Cs along two distinct classes of stakeholders in terms of the inner circle (pink, sky blue, and purple) and the outer circle (yellow, burgundy, and blue). These two classes are: 1) Builder/Developer, Designer, and Tester/Integrator, and 2) Buyer/Acquirer/Financer, End-User, and Team Lead/Manager. These stakeholder groups under each class have similar priorities on architecture concerns when viewed from the four Cs perspectives and therefore any architecture views that are intended to help these stakeholder groups should focus on elaborating on those aspects of system architecture that address these concerns.

Generally, the results indicate that the stakeholder class, Builder/Developer, Designer, and Tester/Integrator, have a need for more information on all the 4 Cs compared to the other class, Buyer/Acquirer/Financer, End-User, Team Lead/Manager (except for a few exceptions where any one of the stakeholders group in the latter class seem to be having similar level of need as one of the stakeholders group in the former class. For example Team Lead/Manager stakeholder group's need for system contractuals is the same as that of the Designer stakeholder group). In other words this class needs higher level of information regarding the system architecture concerns. The need for higher level of information is directly related to the fact that these concerns help them in achieving their system realization tasks.

This also explains why Tester/Integrator requires higher percentage of complete information about system constraints and system contractuals than Builder/Developer or Designer. The Testers/Integrators have to verify and validate compliance with all system constraints and system contractuals and thus require more complete information on these concerns in order to perform their system realization tasks.

Builder/Developer requires higher percentage of complete information about system characteristics than Tester/Integrator or Designer. The Builders/Developers require more complete information on this concern in order to perform their system realization tasks of building or developing the system exhibiting the required characteristics such as behavior and functionality. Context of the system is equally important for all the three stakeholders, Builder/Developer, Designer, and Tester/Integrator, to perform their system realization tasks and thus requiring the same and yet higher percentage of complete information.

The stakeholder class, Buyer/Acquirer/Financer, End-User, and Team Lead/Manager, have a need for less information on all the 4C's compared to the other class, Builder/Developer, Designer, and Tester/Integrator. In other words, this group requires a lower level of information across all architecture concerns.

In comparing these two classes, there is a distinct difference in the tasks that these classes perform in relation to the system. Class 1 actively develops artifacts and products that are directly related to the system, and class 2 tends to develop artifacts and products that are not directly related to the system. This inherently defines that class 2 requires less information that is directly related to the system. The stakeholders in both classes are most concerned with information that enables them to perform their job. For the class 2 stakeholders, the tasks they perform tend not to require the same level of information that the class 1 stakeholders require. This does not imply that the information that the class 2 stakeholders require is not important, but that the information on these concerns should be at a higher level of abstraction or summary level in comparison to the class 1 stakeholder concerns who seem to be requiring this information at a more detailed level.

Our analysis of stakeholders concerns and corresponding architecture views is similar to the work published by Bachman et al. and other authors discussed in earlier sections (Bachmann et al. 2003). The uniqueness of our analysis comes from our focus on understanding the stakeholders' architecture concerns and their need for information arising out of such concerns. All prior research on this topic focuses on the different views of architecture required by the different stakeholders, for example, Bachman et al. focuses on need for different architecture tools and artifacts by the different groups of stakeholders (Bachmann et al. 2003). The article in particular discusses three different view types and stakeholder-specific details in each view type. Our research identifies the architecture concerns that create the need for information (stakeholder-specific details) by different stakeholders. These concerns and corresponding need for architecture information is specific for a specific group of stakeholders.

Our findings on priority of stakeholders' need for system architecture information confirm with this earlier research results (Bachmann et al. 2003). Bachman et al categorizes the information level provided by the different views as 'd' (detailed), 's' (some details), 'o' (overview information, and 'x' (anything). Translating these to comparable levels with our average scores in Figure 1 we find Bachman et al views provide highest level information to the architect, next is the builder/developer, followed by tester/integrator, Team lead/Manager, and the last is the end-user we assigned a 3 for 'd', 2 for 's', 1 for 'o' and 0 for 'x' to arrive at Bachman's comparable averages). Our results are very similar except our Tester/Integrator has the highest need for information, followed by the designer/architect, and then the builder/developer.

The 'Two-View' Framework

The Two View Framework proposed in this section describes two classes of architecture descriptions which differ significantly in the depth and breadth of information they provide to the stakeholder. The first class of view, In-Depth View, in this framework provides significant and complete information across the majority concerns and summary or partial information across the remaining concerns. This view addresses the information needs of the Builder/Developer, Designer, and Tester/Integrator.

In contrast, the second view, Summary View provides top level and summary or partial information across the majority concerns while providing significant and complete information on select concerns. This view addresses the information needs of the Buyer/Acquirer/Financer, End-User, and Team Lead/Manager.

As discussed in the earlier sections and illustrated in Figure 2 there is distinct difference in the architecture concerns and need for information on these concerns between two classes of stakeholders comprised of the six stakeholder groups. These two distinct classes are:

- Buyer/Acquirer/Financer, End-User, and Team Lead/Manager
- Builder/Developer, Designer, and Tester/Integrator

We can further analyze these commonalities and differences in the following Venn diagrams illustrated in Figure 3.

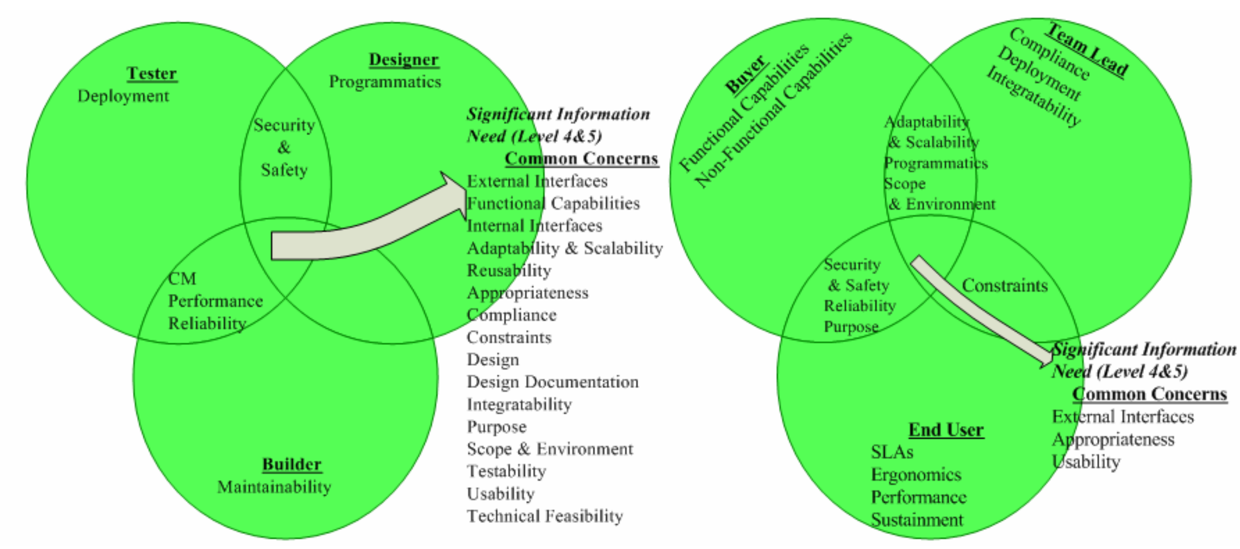

Figure 2. In-Depth View

Figure 3 describes the architecture concerns that are common to each of the two classes of stakeholders. As is evident the first class of 'Builder/Developer, Designer, and Tester/Integrator' has many similar concerns compared to the second class. The most notable aspect of the above diagrams is that the first class of stakeholders has a 'significant' level (level 4&5) of need for architecture information on majority of the concerns. Whereas, the second class of stakeholders has a 'significant' level (level 4&5) of need for architecture information on very few of the concerns. This finding is of tremendous help for projects in identifying the stakeholders who have to be provided more detailed level of architecture information for successful system realization. Projects can create views of architecture supported with the right tools and artifacts that would align with and satisfy the specific information needs of these different groups of stakeholders. For example, a Tester would need more architecture information for deployment, security and safety (that he/she will share with the Designers), CM, performance and reliability (that he/she will share with the Builder/Developer), and all the other common concerns shared between the three groups as shown above. A Tester who gets this kind of detailed information would be able to plan, design, execute, and manage system testing better than if he would not have such information or have such information partially (levels 3,2&1).

The diagrams below illustrate the same analysis of differences between the two classes of stakeholders for concerns for which their need for information is limited to 'some' (level 3), and 'abbreviated' (level 2) and 'no' (level 1). An interesting aspect that diagram on the right below brings out is the concept of a 'surrogate stakeholder'. For example, we have the Team Lead owning Security, Safety, SLAs, Purpose, and Reliability. Does that make the Team Lead a surrogate stakeholder – one acting on behalf of all operational stakeholders as well as the customer? These additional diagrams would help projects if they have the time, and resources to go beyond focusing on only the important architecture information needs to also provide brief or high-level information on the concerns as shown in Figure 4 below:

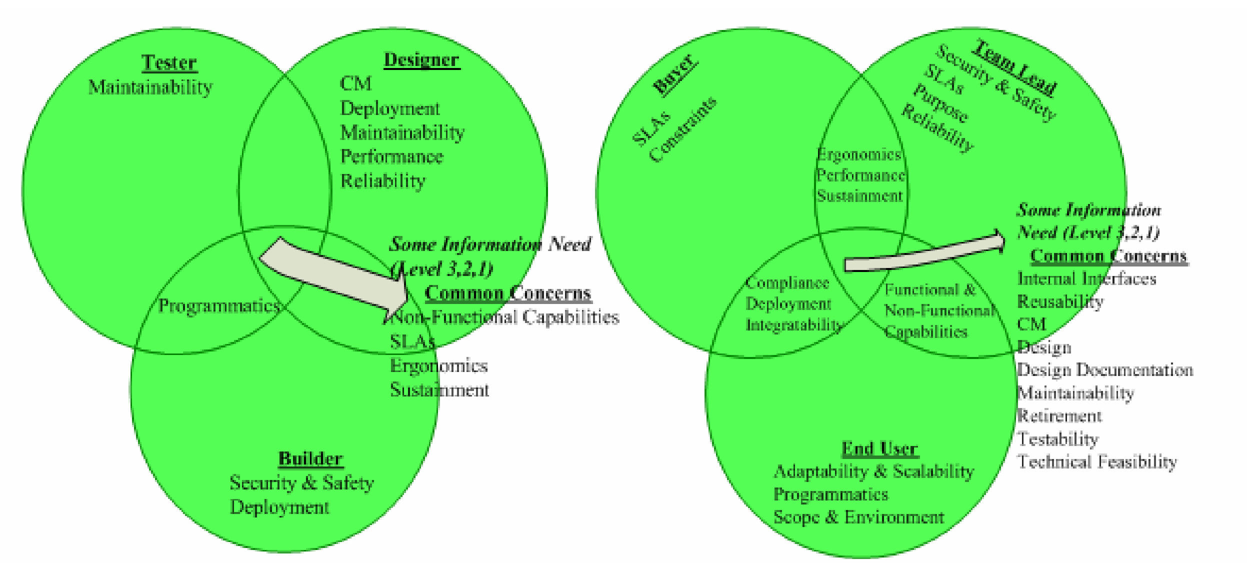

Figure 4. Summary View

We propose this 'Two-View' framework of architecture descriptions that would address the specific concerns for each of the six groups of stakeholders as appropriate. Projects could use these views to create the relevant architecture documentation, tools, and other artifacts to provide the level of information required by each group of stakeholder that would help them in successful system realization. Supporting information on the aforementioned observations is provided in the following Tables (T-W).

Table T. Common Stakeholder Concerns for System Architecture

System Purpose	**The need that the system is intended to satisfy.**
System Constraints (Technology, Standards, and Regulations)	Limitations or implied requirements that define the scope/boundary conditions. These requirements constrain the design solution or implementation of the system and are not changeable (IEEE 1998; INCOSE 2004).
System Compliance (Technology, Standards, and Regulations)	The property where the system is implemented such that it follows established guidelines, standards, specifications, and regulatory/legal requirements.
System Scope and Environment	Information on the scope of the system and its interactions with other systems, environment and its stakeholders. The system environment may include natural or induced environment conditions, anticipated system environment interfaces, and user interactions within which the system is expected to be operated (Kossiakoff 2003).
Technical Feasibility of the System	The ability to determine an adequate qualification method for demonstration that the system satisfies the requirement (SEI 1993).
Functional System Capabilities	The set of functions that is required to produce a particular output or meet a specific objective. This comprises of several set of functions that must be performed to achieve a specific objective. A function is a process that takes inputs in and transforms these inputs into outputs.
Non-Functional System Capabilities	The operational, technical and compliance design constraints such as reliability, maintainability, standards, regulations, etc… associated with the core functional capabilities.
System Performance	A quantitative measure characterizing a physical or functional attribute relating to the execution of a mission or function. (McCabe and Polen 2004) It is the degree to which a system or component can, within a given set of constraints, accomplish its designated functions, such as speed, accuracy, memory usage, etc. (Kossiakoff 2003).
System Design (Decomposition, Modular, Robust, Open, etc)	The description of the design and architecture, and addresses the concerns of design such as decomposition, modularity, robustness, reusability, etc…
Internal System Interfaces	The plane or place at which independent components or subsystems meet and act or communicate with each other (Jain 2007). Internal interface connects a subsystem component to another. The interface of a subsystem contains both a logical element and a physical element (or link) that are responsible for carrying items from one component or subsystem to another (Buede 2000)
External System Interfaces	The plane or place at which independent systems meet and act or communicate with each other (Jain 2007). External interface connects a system to another system.
System Design Documentation	The recording, maintaining, and accessing (as per the agreed upon guidelines) of information on design details in terms of requirements, architecture, functionality and system specifications.
System Configuration Management	The discipline of applying technical and administrative direction and surveillance to: (1) Identify and document the functional physical characteristics of a configuration item, (2) Control changes to those characteristics, and (3) Record and report change processing and implementation status, (4) Audit and review through validation the completeness of a product and maintaining consistency among the components (IEEE 1987).
System Deployment	The ease with which a system can be transported, received, processed, assembled, emplaced, installed, tested, checked-out, trained, housed, and/or stored, as necessary, to field the system into a state of full operational capability, as well as any removal or cleanup of elements of earlier system versions previously deployed (McCabe and Polen 2004).
System Sustainment	The extent to which the system can maintain itself in an operational state sufficient to accomplish its purpose without outside help (McCabe and Polen 2004).
System Retirement	The permanent removal of a system or component from its operational environment, and/or removal of support from an operational system or component (IEEE 610.12-1990).
System Maintainability	The ease with which a system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment (IEEE 1990).
System Reliability	1. The duration or probability of failure-free performance under stated conditions. (MILSTD- 1388-1A, Para 20.) (United States Department of Defense 2003)] 2. The probability that an item can perform its intended function for a specified interval under stated conditions. (For non-redundant items, this is equivalent to definition (1). For redundant items, this is equivalent to mission reliability. (MILSTD- 1388-1A, Para 20.) (United States Department of Defense 2003).
System Testability	The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met (IEEE 1990).
System Integratability	The ability that the hardware, software, and human system components will interact to achieve the system purpose and/or satisfy the customer's need (Jain 2007).
System Usability	The ease with which a user can learn to operate, prepares inputs for, and interprets outputs of a system or component (IEEE 1990).

System Purpose	The need that the system is intended to satisfy.
System and Sub- Systems Reusability	The degree to which a module, component, system or other work product can be used more than once (IEEE 1990).
System Adaptability and Scalability	The ease with which the system can grow to accommodate increased performance (e.g., higher transaction rates, more customers, etc.), expanded functionality (e.g., additional pricing methods) or scaled back to cost effectively support reduced levels of performance or functionality (SEI 2007).
Security and Safety of the System	Security: The property of a system such that reliance can be placed in the absence of accidents. (Barbacci et al. 1995) Safety: Protection from unauthorized use of resources (Barbacci et al. 1995).
System Programmatics (Cost, Schedule, Risk, ROI, TCO, etc)	The management aspects of the project. These aspects consider the cost, schedule, and risk of building, fielding, and supporting the solution to include the cost, schedule, and risk for changing the necessary business processes (SEI 2002).
Service Level Agreements on the System	Contractual agreements from the vendors or suppliers describing the service level parameters such as maintenance, upgrades, support, guarantees, warrantees, and other contractual terms.
System Ergonomics (including aesthetics)	The comfort and appeal of a system design to the human interface for usage and operation.
System Appropriateness (in fulfilling it's mission)	The ability of a system to full fill its purpose within its scope and environment

Table U. Stakeholder Correlations: Similarity and Variance

Stakeholder Group	**Correlation**	**Similarity/Variance in the need for information across the 28 architecture concerns between the groups.**	**Differ by two degree in the level of information across the 28 architecture concerns between the groups. Example: group 1 needs level 5 (complete) information on testability whereas group 2 needs only level 3 (some) information.**	**Difference when compared across high (level 5 & 4), medium (level 3), and low (level 1 and 2) level of required information across the 28 concerns.**
Builder/Developer and Tester/Integrator	High	50% Similarity	11%	21% (Out of which 50% differ in high to medium, and 50% differ in medium to low)
Designer and Tester/Integrator	High	50% Similarity	0%	21% (Out of which 66% differ in high to medium, and 34% differ in medium to low)
Tester/Integrator and End-User	Low	68% Variance	21%	64% (Out of which 83% differ in high to medium, 6% differ in medium to low, and significant 12% differ in high to low)
Buyer/Acquirer/Financer and Builder/Developer	Low	68% Variance	32%. Also 7% variance by three degrees.	61% (Out of which 47% differ in high to medium, 12% differ in medium to low, and significant 41% differ in high to low)
Builder/Developer and End-User	Low	68% Variance	32%. Also 11% variance by three degrees.	64% (Out of which 78% differ in high to medium, and significant 22% differ in high to low)

Table V. Architecture Concerns Under the 4Cs

No.	Architecture Concerns	Context	Characteristics	Constraints	Contractuals
1	System Purpose	X			
2	System Constraints (Technology, Standards, and Regulations)		X		
3	System Compliance (Technology, Standards, and Regulations)			X	
4	System Scope and Environment	X			
5	Technical Feasibility of the System		X		
6	Functional System Capabilities		X		
7	Non-Functional System Capabilities		X		
8	System Performance		X		
9	System Design (Decomposition, Modular, Robust, Open, etc)	X			
10	Internal System Interfaces	X			
11	External System Interfaces			X	
12	System Design Documentation		X		
13	System Configuration Management				X
14	System Deployment				X
15	System Sustainment				X
16	System Retirement				X
17	System Maintainability				X
18	System Reliability			X	
19	System Testability		X		
20	System Integratability			X	
21	System Usability			X	
22	System and Sub- Systems Reusability				X
23	System Adaptability and Scalability			X	
24	Security and Safety of the System			X	
25	System Programmatics (Cost, Schedule, Risk, ROI, TCO, etc)				X
26	Service Level Agreements on the System				X
27	System Ergonomics (including aesthetics)		X		
28	System Appropriateness (in fulfilling itÕs mission)	X			
	TOTALS:	5	8	7	8

Table W. Percentage of Information Required by Stakeholder Groups Across the 4Cs

Stakeholder Group	Context	Constraints	Contractuals	Characteristics
Builder/ Developer	76	74.28571	40	73.75
Buyer/ Acquirer/ Financer	58	61.42857	36.25	36.25
Designer	76	74.28571	43.75	58.75
End-User	52	57.14286	36.25	47.5
Team Lead/ Manager	52	61.42857	43.75	43.75
Tester/ Integrator	76	87.14286	47.5	66.25

AUTHOR BIOGRAPHIES

Dr. Rashmi Jain is an Associate Professor of Systems Engineering at Stevens Institute of Technology. Dr. Jain has over 15 years of experience of working on Information Technology (IT) systems. Prior to joining Stevens she was with Accenture (formerly known as Andersen Consulting). Over the course of her career she has been involved in leading the implementation of large and complex systems engineering and integration projects.

She has done invited lectures internationally at Keio University, and Shibaura Institute of Technology, Japan, Overseas Chinese Institute of Technology (OCIT), Taiwan, Indian Institute of Technology – Delhi etc. She is a visiting professor for System Architecture and

Integration at Keio University. Her teaching and research interests include systems integration, systems architecture and design, business process reengineering, and rapid systems engineering. Dr. Jain has authored several papers on these topics. Her teaching and research interests include systems integration, systems architecture and design, and rapid systems engineering. She holds Ph.D. and M.S. degrees in Technology Management from Stevens Institute of Technology.. She can be contacted at Rashmi.Jain@stevens.edu.

Anithashree Chandrasekaran is a Doctoral Candidate in the School of Systems and Enterprise at Stevens Institute of Technology. Her research interests include Rapid Systems Development and its processes, Process reengineering, Risk Management and Modeling, System Integration, System Design and Architecture. She obtained her B.E. in Electrical and Electronics Engineering from P.S.G. College of Technology, India. She obtained her M.S. in Systems Engineering from Stevens Institute of Technology. Anithashree currently serves as the president of Stevens INCOSE student chapter and can be reached at Anithashree.Chandrasekaran@stevens.edu.

George M. Elias is a Systems Engineering Doctoral Candidate at Stevens Institute of Technology. Mr. Elias is a Systems Engineer at ITT Electronic Systems in Clifton, New Jersey. Mr. Elias has an Undergraduate degree in Information Systems from Rutgers and New Jersey Institute of Technology and a Masters in Computer Science from Stevens Institute of Technology. Mr. Elias has research interests in Systems Architecture and Systems Engineering Attributes. He can be contacted at gelias@stevens.edu.

REFERENCES

Alexander, I. (2005). A Taxonomy of Stakeholders: Human Roles in System, Development, International Journal of Technology and Human Interaction, Vol. 1, Issue 1, pp 23 – 59.

ANSI/EIA (1999). Standard 632, Process for Engineering a System,

Bachmann, F., Bass, Ll., Clements, P., Garlan, D., Ivers, J., Little, R., Nord, R., and Stafford, J., (2001). Documenting Software Architectures: Organization of Documentation Package, August 2001.

Bachmann, F., Bass, L., Clements, P., Garlan, D., Ivers, J., Little, R., Nord, R., and Stafford, J. (2003).., Documenting Software Architectures: Views and Beyond, Addison Wesley Professional., 2003.

Baragry, J. and Reed, K. (2001). Why we need a different view of software architecture, Proceedings of the Working IEEE./IFIP Conference on Software Architecture, IEEE Computer Society, 2001.

Barbacci, M., Klien, M., Longstaff, T., and Weinstock, C. (1995), Quality Attributes - Technical Report CMU/SEI-95-TR-021 ESC-TR-95-021, Carnegie Mellon Software Engineering Institute, 1995.

Bass, L., P. Clements, and R. Kazman (1998), Software Architecture in Practice, Addison-Wesley. Professional, 1998

Boehm B. ., and H. In (1996). In H., Identifying Quality-Requirement Conflicts, IEEE Software.

Bouck´e, N. and Holvoet, T. (2005), Dealing with concerns ask for an architecture-centric approach. In European Interactive Workshop.

Buede, D. (2000). The Engineering Design of Systems. Wiley Series in Systems Engineering.

Carlomusto, M., Giammarco, K., and Lock, J. (2005). Development and analysis of integrated C4ISR architectures, U.S. Army Research.

Clements, P. (2006(2000), Active Reviews for Intermediate Designs, Paul C. Clements, August 2000.

Defense Acquisition Guidebook, Defense Acquisition University.

Defense Acquisition Guidebook (2006), Defense Acquisition University,

Hofmeister, C., Nord, R., and Soni, D. (2000)., Applied Software Architecture, AddisonWesley..

Institute of Electrical and Electronics Engineers - IEEE (1987). Standard 1042, Guide to Software Configuration Management.

IEEE (1990). IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries.

IEEE (1990). Standard 610.12-1990: IEEE Standard Glossary of Software Engineering Terminology, 1990.

IEEE (1998). Standard 1220-1998: IEEE Standard for Application and Management of the Systems Engineering Process.

IEEE (2000). Standard 1471, IEEE Recommended Practices for Architectural Description of Software-Intensive Systems.

IEEE (2004). Standard 15288, Systems Engineering—System Life Cycle Processes, Adoption of IS0 15288.

INCOSE (International Conference on Systems Engineering), Systems Engineering Handbook, A "What To" Guide for all SE Practitioners, Version 2a, 2004.

Jain, R. (2007). SYS 605 – System Integration, Course Notes.

Kazman, R., Klein, M., and P. Clements (2000). ATAM:Method for Architecture Evaluation, August 2000.

Kossiakoff, A. and Sweet, W. (2003). Systems Engineering Principles and Practices, Wiley &. Sons, NY.

Kruchten, P. (1995). Architectural Blueprints - The "4+1" View Model of Software Architecture, IEEE Software.

McCabe, R. and M. Polen (2004). Evaluating Architectures With System Attributes, Software Productivity Consortium, March 2004.

Software Engineering Institute – SEI (1993). Taxonomy-Based Risk Identification- Technical Report CMU/SEI-93-TR-6 ESC-TR-93-183, Carr, M., Konda, S., Monarch, I., Ulrich, F.C., Walker, C., Carnegie Mellon, June 1993.

SEI (2001). Workshop on Software Architecture Representation, 16-17 January 2001,

SEI (2002). Evolutionary Process for Integrating COTS-Based Systems: A Overview, Carnegie Mellon, July 2002.

SEI (2007). Glossary, http://www.sei.cmu.edu/str/indexes/glossary/

Stevens Institute of Technology, SYS 625 – Fundamentals of Systems Engineering, Course Notes, 2006.

The Open Group Architecture Framework - TOGAF™ Version 8.1.1 (2007). Enterprise Edition, Open Group, 2007.

Tu, Q. and W. Godfrey (2001). The build-time software architecture view, IEEE.

United States Department of Defense (2003). Military Standard (MIL-STD) 1388-1A, MIL-HDBK-502 Logistic Support Analysis, 1983, updated 28 June 2003.

Zachman J. (1987), A Framework for Information Systems Architecture, IBM Systems Journal, Volume 26, Number. 3, 1987.

Zachman, J. and J. Sowa (1992). Extending and formalizing the framework for information systems architecture, IBM Systems Journal Volume 31, Number 3, 1992.