Fall 2010

Contribution-based priority assessment in a web-based intelligent argumentation network for collaborative software development

Maithili Satyavolu

CONTRIBUTION-BASED PRIORITY ASSESSMENT IN A WEB-BASED INTELLIGENT ARGUMENTATION NETWORK FOR COLLABORATIVE SOFTWARE DEVELOPMENT

by

MAITHILI SATYAVOLU

A THESIS

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN COMPUTER SCIENCE

2010

Approved by:

Dr. Frank Liu, Advisor
Dr. Ming C. Leu
Dr. Maggie Cheng

ABSTRACT

Decision making is an important aspect of the collaborative software development process which usually involves complex process of conflict resolution. Stakeholders approach decision making process from multiple perspectives and their priorities play a vital role in it. The priority assessment methods used in the argumentation process so far are usually static. Priorities remain constant throughout the decision making process. In order to make the collaborative system more closely replicate real-world scenarios, this work incorporates dynamic priority assessment into a web-based collaborative system which is based on intelligent computational argumentation. It evaluates priorities dynamically for each cycle of decision process based on contribution of individual participant. The contribution is assessed based on the impact of each participant's arguments on a winning design alternative. More successful participants have higher priorities in argumentations during collaboration. An empirical case study is conducted to evaluate effectiveness of dynamic priority assessment in improving quality of the argumentation based decision making.

ACKNOWLEDGMENTS

I am indebted to many people and wish to thank all who have helped me throughout this thesis project. First, I thank my advisor, Dr. Frank Liu, who has encouraged and challenged me throughout the research. His professional suggestions and help allowed me to do this research more efficiently and easily.

I would also like to thank Dr. Ming Leu and Dr. Maggie Cheng for serving on my thesis committee and taking time to review this work. I am grateful as well to the members of my research team, Ravi Arvapally and Rubal Wanchoo, for their timely support and help.

I would also like to acknowledge the generous funding provided by the Intelligent Systems Center (ISC) at Missouri University of Science and Technology.

I am most grateful to my family and my friends, Vikram Annambhotla, Giraj Kandukuri, Shrikant Jarugumilli, Ekta Khudkhudia, and Kirthana Akunuri for their help and support.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

INTRODUCTION

A software product design and development process becomes complex, involving designers and customers located at different sites. The need for reducing the cost and time of design and development has resulted in development of collaborative decision support systems to achieve effective collaboration among participants in a software development process. These are interactive computer-based systems that enhance the effectiveness of the decision making process by facilitating communication among the stakeholders.

In order to make collaboration more efficient, some mechanism is necessary to resolve conflicts and provide consensus. Computer-supported collaborative argumentation (CSCA) is a kind of CDSS developed to facilitate decision making through argumentation. A number of CSCAs have been developed that rely on argumentation theories.

In previous research issue of conflict resolution has been addressed by developing a web-based argumentation system for collaborative software development decision making [1]. The system has further been enhanced to detect self-conflicting arguments [2] and to reassess argument strengths based on evidences. Currently we improve its priority assessment method, enhancing the quality of decision making by dynamically prioritizing participants based on their contribution to winning design alternative in an argumentation based collaborative software development process.

The priority assigned to each participant is a basic factor in objective decision making in an argumentation system. Priorities are usually assigned based on roles of the participants in a software development process. These priority assignments usually remain constant throughout decision process in a collaborative software development process. However, in real applications, it is essential to update the priorities dynamically.

This thesis discusses dynamic priority assessment in a web-based intelligent argumentation collaborative system [1], and [2] to make a collaborative decision making in a collaborative software development process more objective and analogous to the real-world scenario of argumentation. The priorities of participants are assessed based on their contributions to decision making process. The greater a participant's contribution to a successful decision, the higher his or her priority. Similarly, the less significant the contribution, the lower is the participant's priority. Thus, the priority of a participant is determined dynamically based on individual's contribution made to a winning design alternative in a collaborative software development process.

RELATED WORK

1.1. COMPUTATIONAL ARGUMENTATION SYSTEM

Argumentation systems are built based on a classical model of argumentation developed by Philosopher Toulmin [3]. An earlier method called graphical IBIS (gIBIS) represented design dialog as a graph [4]. Although capable of representing issues, positions, and arguments, gIBIS failed to support representation of goals or requirements and outcomes. REpresentation and MAintenance of Process knowledge, or REMAP [5], extended gIBIS by permitting the representation of goals, decisions, and design artifacts.

Sillince [6] offered an alternative to these systems, proposing a more general argumentation model. His is a logic model in which dialogs are represented as recursive graphs and the rules of both rhetoric and logic are used to manage the dialog and determine when it has reached closure. Potts and Burns [7] outlined a generic model for representing design deliberation and the relationship between deliberation and the generation of method-specific artifacts. It differs from the system proposed here in its lack of decision making capabilities. HERMES [8] is a system that not only captures the informal organizational memory embodied in decision making settings, but also helps users during the decision making process. This system however uses a weighting factor that is ineffective because it does not relate to the position entered.

Pike et al.[9] developed a scalable reasoning system, that represents various reasoning artifacts such as arguments and evidence. It is similar to the present work in that it represents the relationship between two artifacts as support or refute. However, the system does not include any inference engine for decision-making.

2.2. PRIORITY ASSESSMENT

Requirements prioritization is an essential task when working in a collaborative environment. Since the requirements come from multiple stakeholders with different interests, prioritizing the requirements is a challenging task. To develop a successful software project, effective negotiation among the stakeholders is essential.

Many prioritization techniques have been proposed based on analytical and mathematical approaches. The analytic hierarchy process (AHP) proposed by Saaty [10] uses an exhaustive pair-wise evaluation in a hierarchy. This approach has proved complicated, time-consuming, and impractical for large projects. Several researchers have proposed techniques to overcome the computation exploitation [11]. None of these techniques capture the co-relation among the requirements for prioritization among different stakeholders.

In 2006, Liu et al. [12] proposed a correlation-based priority assessment framework (CBPA). It prioritizes software process requirements gathered from multiple stakeholders by incorporating relationships among requirements. However, this framework does not address the negative correlations. A subsequent project [13] developed web based collaborative system to incorporate the priority of participants into an intelligent collaborative system based on computational argumentation.

The priority assessment techniques used in all the above methods are static, that is, the priority value of a participant does not change no matter how valuable or insignificant an individual's contribution to the decision making process. Even if a participant with a high priority based on his or her position in an organization makes a poor judgment or offers a poor argument, that individual may retain a high priority for subsequent decisions.

WEB-BASED INTELLIGENT ARGUMENTATION SYSTEM FOR DECISION MAKING IN COLLABORATIVE SOFTWARE DEVELOPMENT

A computational argumentation model for decision making in collaborative software development decision making has been developed to promote consensus among stakeholders and identify the most favorable development alternative. The system is based on client-server architecture. On the client side, it provides a user interface for argumentation-based conflict resolution, a whiteboard for sharing designs, and chat rooms for real-time information exchange. On the server side, it manages client communication and an argumentation network. Figure 3.1 shows the graphical user interface of the system.

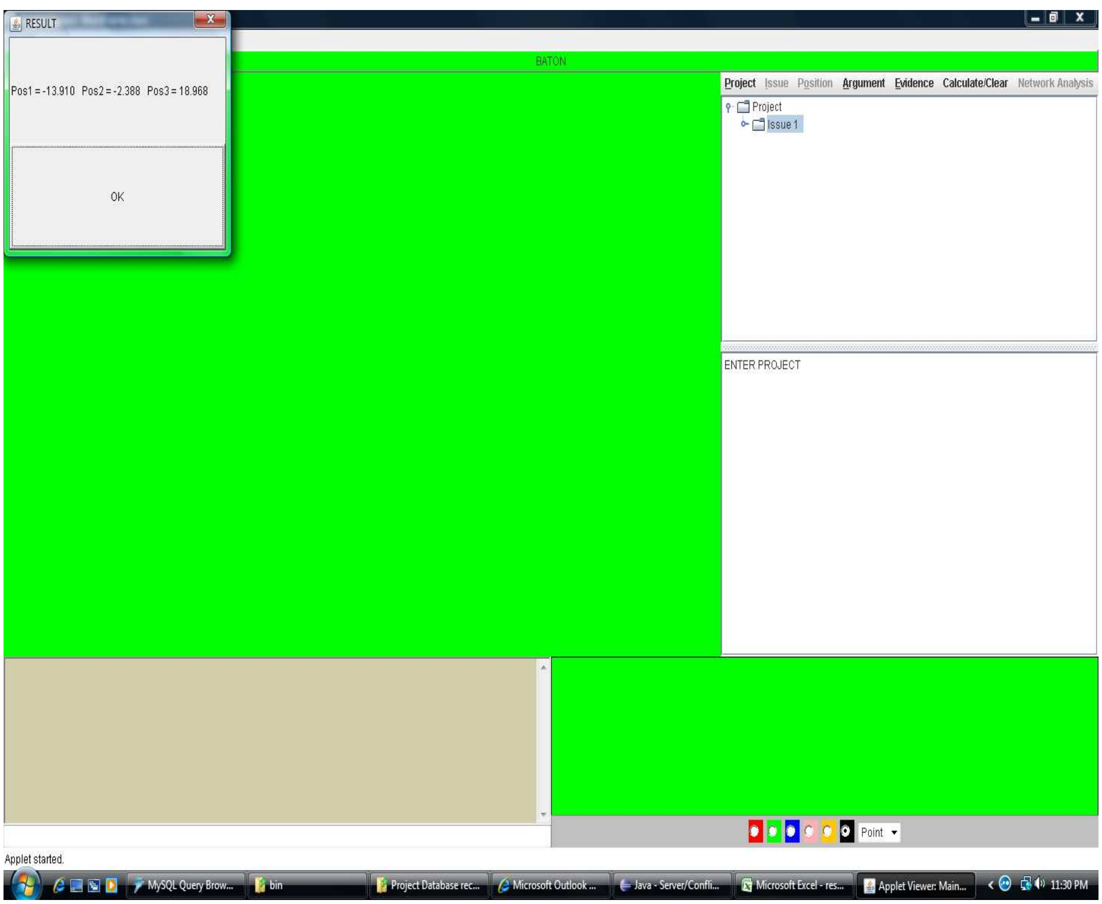


Figure 3.1 Graphical User Interface

The argumentation theory used in this system is that proposed by Toulmin [2], and the arguments are arranged in a hierarchical structure called the argumentation network. At the top of the hierarchical structure is the design issue at hand. For any design issue, there are multiple alternative solutions, or positions, that make up the second level in the argumentation network. An argumentation dialog for a development issue in an argumentation model can be captured as a weighted argumentation tree, as shown in Figure 3.2.

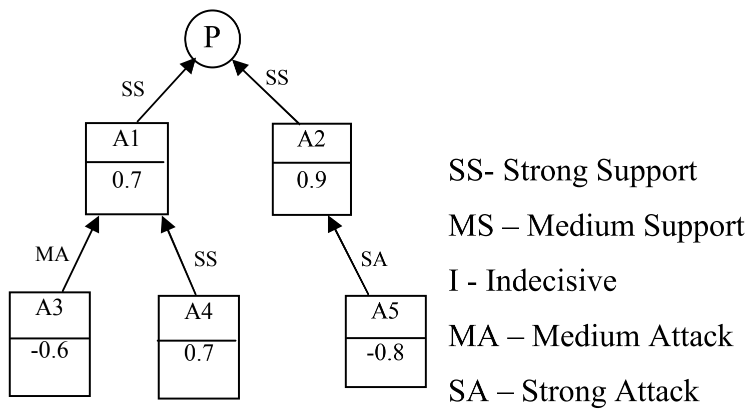


Figure 3.2. Argumentation Tree

The arguments below the positions in the argumentation tree represent opposing viewpoints - those that support another argument or position, and those that attack another argument or a position. The nodes in the argumentation tree in Figure 3.2 are design alternatives and arguments. The node denoted by a circle is a position, that is, a design alternative, and the nodes denoted by rectangles are arguments. Arrows represent a relationship (attack or support) from an originating argument node to a terminating argument or position node. The weight assigned to an argument is the argument strength. It is the measure of an argument's degree of attack or support of either a position or another argument. The weight value is a real number between -1 and 1. A positive number denotes support, and a negative number denotes attack; zero denotes indecision.

In addition to a hierarchical organization of the argumentation elements, this system also provides a means to evaluate quantitatively the favorability of the various design alternatives available. This evaluation is accomplished by assigning a numerical value to each argument and inputting these values to a fuzzy inference engine. A fuzzy inference engine was developed using the fuzzy association memory for argumentation reduction. It permits assessment of the quantitative impact of indirect arguments on a software development alternative. The favorability of a software development alternative is computed by a weighted summation of the strengths of arguments attached directly to it. The position with the maximum favorability factor is the best option.

CONTRIBUTION-BASED PRIORITY ASSESSMENT

The contribution of a participant is evaluated based on an assessment of the impact of his or her arguments on the winning design alternative. In argumentation, participants represent their support for or opposition to a particular position as weights, associating their arguments with the position. A participant's contribution reflects the amount of support or opposition for a winning alternative. It is a real number between -1 and 1. A weight closer to 1 represents a greater positive contribution made to a successful decision. A weight closer to -1 represents a greater negative contribution made to a successful decision.

4.1. EVALUATION OF CONTRIBUTION OF PARTICIPANTS

The contribution of a participant is evaluated after the argumentation reduction phase [15] during which arguments move up in the argumentation network. All the arguments are directly attached to positions on a single level. The process of evaluating a participant's contribution is as follows:

1) Identify and group all the arguments from each participant.

Let, $P_1, P_2, \ldots, P_n$ be stakeholders participating in the argumentation process, where n is the total number of stakeholders. Assume that a participant P_i ($1 \leq i \leq n$) has M_i arguments in an argumentation network. The arguments are grouped for participant P_i.

2) Calculate the weighted summation of strengths of each participant.

Let $S_{i,j}$ ($1 \leq i \leq n$, $0 \leq j \leq M_i$), represent the strength of argument $A_{i,j}$ posted by participant P_i. Assume that WS_i is the weighted summation of the argument strengths of participant P_i. The weighted sum of the arguments $A_{i,j}$ is then calculated as:

$$WS_i = \sum_{j=0}^{M_i} S_{i,j} \qquad 1 \leq i \leq n. \qquad (1)$$

Similarly, the weighted summation of the argument strengths $WS_{1,}$ $WS_{2,\ldots,}WS_{n.}$ is calculated for all the other participants.

3) Classify participants as positive or negative contributors based on their weighted summation values.

i) Positive Contribution: Let K be the number of positive contributors, i.e., the participants whose weighted summation values are greater than zero, where $1 \leq K \leq n$. For the purpose of priority assessment, the positive contribution values of individual participants should lie between 0 and 1.0. Since the weighted summation values calculated in equation (1) for a positive contributor may exceed this range, each positive contributor's contribution is normalized as follows:

$$C_T = WS_T / HP \,. \tag{2}$$

where $1 \leq T \leq K$ and HP is the highest positive contribution value among all participants. Substituting equation (1) into equation (2) gives the positive contribution made by the participant:

$$C_T = \sum_{j=0}^{M_T} S_{T,j} \; / \; \max\left(\sum_{j=0}^{M_T} S_{T,j} \,. \right) \tag{3}$$

ii) Negative Contribution: Let L be the number of negative contributors, i.e., the participants whose weighted summation values are less than zero, where $1 \leq L \leq n$. Dynamic priority reassessment based on contributions assumes that the contribution values lie between the range -1.0 and 0. Since the weighted summation values calculated in the equation (1) for a negative contributor may exceed this range, each negative contributor's contribution is normalized as follows:

$$C_G = WS_G / HN \qquad . \tag{4}$$

where $1 \leq G \leq L$, and HN is the highest negative contribution value among all participants. Substituting equation (1) into equation (4) gives the negative contribution made by the participant:

$$C_G = \sum_{j=0}^{M_G} S_{G,j} \;/\; \max\left(\sum_{j=0}^{M_G} S_{G,j} \right) \quad (5)$$

In this procedure, equations (3) and (5) represent the degree of the positive or negative contribution made by each participant towards the winning design alternative.

4.2. REASSESSMENT OF PRIORITY-BASED CONTRIBUTION USING FUZZY LOGIC

After evaluating the contribution of each participant, the priorities of the participants can be reassessed based on the following heuristic rules:

General Priority Reassessment Heuristic Rule 1: The greater a participant's contribution to a successful decision, the higher that participant's priority.

General Priority Reassessment Heuristic Rule 2: The less significant a participant's contribution to a successful decision, the lower that participant's priority.

Participants are classified as positive or negative contributors. The positive contributors are rewarded, and the negative contributors are penalized. Therefore, there are two association matrices for adjusting the priority of participants based on their contributions. One is for a priority increase based on a positive contribution; and other is for a priority decrease based on negative contributions. The general heuristic rules are extended to nine fuzzy inference rules in each of the two fuzzy association matrices. The linguistic labels used for the strength of priorities and contributions are high (H), medium (M), and low (L). The inputs to the fuzzy association matrix are the contribution (horizontal) and priority (vertical) values of the participants. Figure 4.1 shows a fuzzy association matrix for priority increase based on positive contributions. The final priority values are calculated as:

$$\text{Final Priority } (P_i) = \text{Initial Priority } (P_i) + (\text{Output}/10)*\text{Initial Priority } (P_i). \quad (6)$$

The value of the output of this equation is calculated using the fuzzy association matrix shown in Figure 4.1. Figure 4.2 shows the fuzzy association matrix for a priority decrease based on a negative contribution. The final priority values are calculated as:

Final Priority (P_1) = Initial Priority (P_1) - (Output/10)*Initial Priority (P_1). (7)

The output value of this equation is calculated using the fuzzy association matrix shown in Figure 4.3.

P \ PC	H	M	L
H	H	M	L
M	H	M	L
L	H	M	L

P-Priority
PC- Positive Contribution
H-High
M- Medium

Figure 4.1. Fuzzy Association Matrix for Priority Assessment of Positive Contribution

P \ NC	H	M	L
H	H	M	L
M	H	M	L
L	H	M	L

P-Priority
NC- Negative Contribution
H-High
M- Medium

Figure 4.2. Fuzzy Association Matrix for Priority Assessment of Negative Contribution

This fuzzy inference engine incorporates contribution to revise the priority of a participant. Fuzzy membership functions are used to characterize linguistic labels quantitatively. Based on previous research, the fuzzy membership function chosen for the priority is the piecewise linear trapezoidal function. The three fuzzy sets are low, medium, and high, and the membership functions for priority are shown in Figure 4.3. The fuzzy membership function chosen to represent contribution is also the piecewise linear trapezoidal function. The three fuzzy sets are low, medium, and high, and the membership functions for contribution are shown in Figure 4.4.

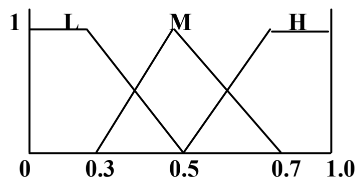


Figure 4.3. Fuzzy Membership Functions for Priority

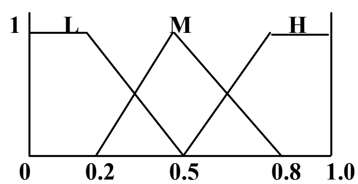


Figure 4.4. Fuzzy Membership Functions for Contribution

4.3. EXAMPLE

A specific argumentation network involving three design issues provides an example of dynamic priority reassessment. The argumentation tree shown in Figures 4.5, 4.6, and 4.7 represents these design issues. Each issue involves five participants who have posted their arguments either in favor of or against a position.

The priorities are reassessed based on the contributions made by the participants to discussion the design issues. Each contribution is evaluated with the help of the final weights representing the strengths of the arguments after the argumentation reduction process. The output of this process is a one-level argumentation tree that contains the arguments posted by each participant and the strengths of those arguments.

The weighted summation of all the arguments for each participant is then calculated. The normalized contribution of each positive contributor is the ratio of the individual's contribution to the highest positive contribution value among all the positive contributors. The normalized contribution of each negative contributor is the ratio of the individual contribution to the highest negative contribution value among all the negative contributors.

The contributions of the participants towards the winning design alternative are calculated. The notation P_i denotes the i^{th} participant. In this example, the impact of a contribution on priority reassessment is shown using participant P_3.

i) Design Issue 1: Initial priority values are assigned heuristically to the participants based on their roles. Table 4.1 shows the argumentation scores calculated for the two positions under design issue 1. Position 1 is identified as the winning design alternative. Therefore, the contribution of participants to Position 1 is calculated.

Table 4.1 Favorability Factors for Issue 1

Position 1	Position 2
0.677	0.100

Contribution to Position 1: Only four participants offer an argument about this position. The weighted summation of the argument strengths is calculated for each participant using equation (1).

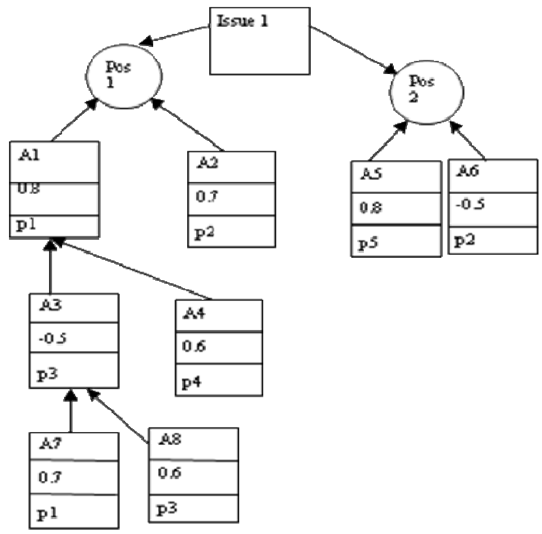


Figure 4.5. Argumentation Tree for Issue 1

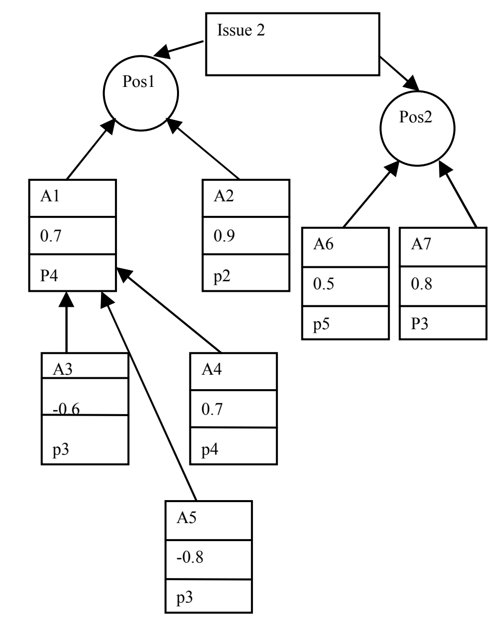


Figure 4.6. Argumentation Tree for Issue 2

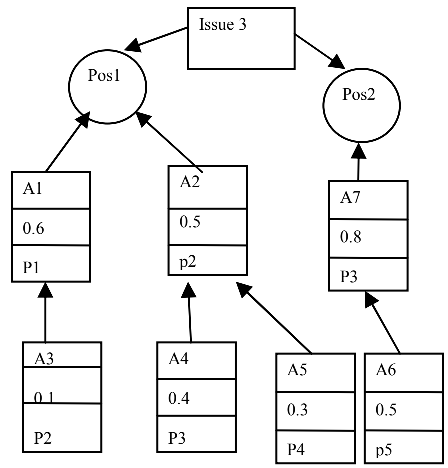


Figure 4.7. Argumentation Tree for Issue 3

For participant P_3, the weighted summation value is -0.5. The normalized positive contribution (C_T) and the negative contribution (C_G) for each participant is calculated based on equations (3) and (5), respectively. For participant P_3, the negative contribution is equal to -1.0; which shows that participant P_3 has contributed negatively to design issue 1. That participant's priority is calculated based on the fuzzy rules in the association matrix shown in Figure 4.2. The inputs are the fuzzy membership values of the contribution and the initial priority. The output is the percentage decrease in the priority. The fuzzy membership values for the priority and contribution of P_3 are calculated using Figures 4.3 and 4.4, respectively; as shown in Table 4.2.

Table 4.2 Fuzzy Membership Values for Priority and Contribution

Priority	Contribution
P_H (0.4) = 0	C_H (-1.0) = 1.0
P_M (0.4) = 0.5	C_M (-1.0) = 0
P_L (0.4) = 0.5	C_L (-1.0) = 0

The output is calculated based on the fuzzy membership values shown in Table 4.2.

Output = w1 * P(M,H)+w2 * P(L,H) / (w1+w2).

where, P(M,H) is the priority value when the contribution is high and the priority is medium, and P(L,H) is the priority value when the contribution is high and the priority is low:

w1= min [P_M (0.4), C_H(-1.0)] = 0.5 and w2= min [P_L (0.4), C_H(-1.0)] = 0.5.

Output = (0.9*0.5+0.9*0.5)/(0.5+0.5) = 0.9.

The modified priority of participant P_3 is calculated using equation (7) as 0.364. Similarly, the modified priority values for all the other participants are calculated as shown in Table 4.3.

Table 4.3. Reassessed Priority Values for Issue 1

Participants	Contribution	Initial Priority	Modified Priority
P1	0.892	0.4	0.436
P2	1.0	0.6	0.654
P3	-1.0	0.4	0.364
P4	0.892	0.7	0.763
P5	0	0.5	0.5

Participants P_1, P_2, and P_4 have made positive contributions in selecting the winning position, and the participant P_3 has made a negative contribution. This fact is reflected in their updated priority values.

ii) Design Issue 2: The choice of the winning design alternative depends on the reassessed priority values, i.e., updated priority values obtained as an output from the first design issue. With these priority values, the argumentation scores are calculated for the two positions under design issue 2, as shown in Table 4.4. Position 1 is identified as the winning design alternative. Therefore, the contribution of participants to Position 1 is calculated.

Table 4.4. Favorability Factors for Issue 2

Position 1	Position 2
1.615	0.490

Contribution towards Position 1: Only three participants offer an argument about this position. The weighted summation of argument strengths is calculated for each participant using equation (1). For example, the weighted summation for participant 3 is -0.79. The normalized positive contribution and negative contribution for each participant are calculated based on equations (3) and (5), respectively. For participant 3 the negative contribution is -1.0. The modified priority of participant 3 is calculated using equation (7) as 0.332. Similarly, the priority values are calculated for all the other participants as shown in Table 4.5.

Table 4.5. Reassessed Priority Values for Issue 2

Participants	Contribution	Initial Priority	Modified Priority
P1	0	0.436	0.436
P2	1.0	0.654	0.72
P3	-1.0	0.374	0.332
P4	0.77	0.763	0.83
P5	0	0.5	0.5

Participant P_3 has negatively contributed to the winning alternative, i.e., Position 1. This is reflected as a decrease in this individual's priority value. Participants P_2 and P_4 made more contributions, therefore their priority value rises.

iii) Design Issue 3: The third design issue demonstrates the importance of participant priority in decision making. Participants P_1, P_2, and P_4 select a design alternative based on their credit values with the help of dynamically changing priorities. In design issue 3, Position 1 is the winning design alternative, as indicated by the argumentation scores shown in Table 4.6.

Table 4.6. Favorability Factors for Issue 3

Position 1	Position 2
0.941	0.540

Contribution towards Position 1: The contribution of participants to Position 1 is calculated. Only four participants offer an argument about this position. The weighted summation of the argument strengths is calculated for each participant using equation (1). That for participant 3 is 0.5. The positive contribution and the negative contribution for each participant are calculated based on equations (2) and (4), respectively. The negative contribution of participant 3 is 0.83.The modified priority of participant 3 is calculated using equation (6) as 0.36. Participant P_3 has contributed positively to the winning alternative, i.e., Position 1, which is reflected as an increase in that participant's reassessed priority value. Similarly, the modified priority values are calculated for all the other participants, as shown in Table 4.7.

Table 4.7. Reassessed Priority Values for Issue 3

Participants	Contribution	Initial Priority	Modified Priority
P1	1.0	0.436	0.48
P2	0.83	0.72	0.78
P3	0.83	0.34	0.36
P4	0.5	0.83	0.9
P5	0	0.5	0.5

EMPIRICAL ANALYSIS OF THE CONTRIBUTION-BASED DYNAMIC PRIORITY ASSESSMENT FOR AN INTELLIGENT ARGUMENTATION SYSTEM

5.1. EMPIRICAL STUDY

Three related case studies were conducted to evaluate the effectiveness of the dynamic priority assessment in an intelligent collaborative software development decision making system. The objective of these case studies was to show the improvement in the quality of the decision making due to the dynamically changed priorities of the stakeholders. These case studies also demonstrate the effectiveness of the argumentation system in the decision making process.

1.1.1. Framework. Adoption of software metrics has become a key area of concern in the software industry. The extent to which software metrics are applied depends on a number of factors, including the size of an organization. The three case studies developed here are concerned with the adoption of software metrics to manage projects and improve software quality. They use an argumentation system to determine scale of software metrics used to manage the quality of the projects.

Software organizations can be classified as small, medium, or large. The organizations involved in these case studies are classified based on the number of employees. A small-scale software company has fewer than 100 employees, a medium-scale software company has 100 to 500 employees, and a large-scale software company has more than 500 employees.

An argumentation network was developed to capture the development rationale of stakeholders outside the software development company. This group consisted of 25 students with software engineering backgrounds who are well aware of the software development phases and have a preliminary knowledge of the software metrics program. An environment was simulated in which the stakeholders used the argumentation system collaboratively to make the decision. All the stakeholders began with the same initial priority; based on their contributions to the winning alternative, their priorities were dynamically calculated. At the end of each argumentation phase, the most favorable option was identified with the help of the intelligent argumentation system, and the decision made was consistent with the stakeholder’s expectations.

To evaluate the performance of the argumentation system and to understand its effect on the decision making process, the project conducted two surveys, one before the argumentation process and one after. The main objective of this study was to understand how the argumentation process would influence any changes in decisions from survey 1 to survey 2. The following analysis addresses three phases of the study: survey 1, the argumentation process, and survey 2 as shown in Figure 5.1.

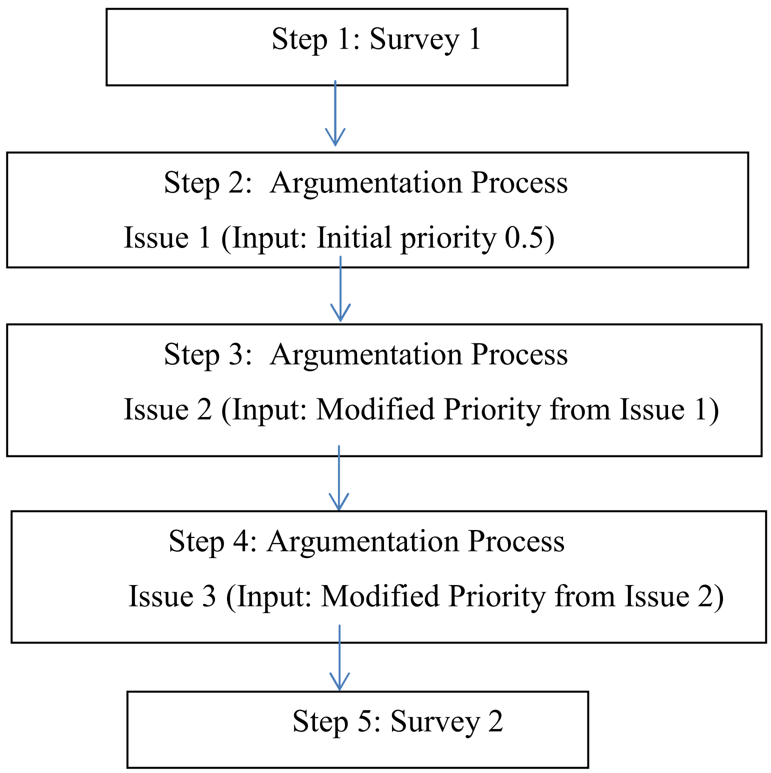


Figure 5.1.Steps Involved in Empirical Study

1.1.2. Issues. The goal was to determine the scale of a software metrics program for managing quality, improving productivity, and reducing the cost of projects in small, medium, and large organizations. Three alternatives were considered:

1) No Metrics Program. Organizations do not want to adopt a software metrics program.

2) Lightweight Metrics Program. Fewer than 35% of the artifacts are measured using a metrics program.
3) Comprehensive Metrics Program. From 35% to 60% of the artifacts are measured by using a metrics program.

The participants debated on the pros and cons of each alternative based on criteria taken from a survey on software metrics adoption for various organizational levels. The criteria for determining the scale of software metrics program were cost of implementation, product quality, product development, effectiveness of software project management, project schedule, project planning, competitiveness: competing in a large market scenario, and customer satisfaction based on quality assurance standards.

i) Issue 1: An argumentation environment was simulated in which the participants entered their arguments for or against each alternative in the web-based collaborative software system to decide what type of software metrics program should be adopted for a large organization. After one week of argumentation, the winning design alternative was calculated. The priorities of the stakeholders were reassessed based on their contribution to the winning alternative.

ii) Issue 2: With their priorities having been adjusted based on their contributions, to the first issue, the same participants then debated the scale of software metrics program to be used to manage projects and improve software quality in a medium-scale organization. Over the course of a week, the participants posted their arguments in favor of or against the same three alternatives. The stakeholders were given one week's time to complete the argumentation process. At the end of the week, the winning alternative was calculated, and the priorities of the stakeholders were reassessed again based on their contributions to the winning alternative.

iii) Issue 3: Using the same procedure, the participants debated the third issue, the scale of the software metrics programs to be adopted by a small organization. This process provided a practical demonstration of participants priority adjustments on the decision making process. The better a participant's judgment, the higher is his or her priority in subsequent debates.

1.1.3. Background of Hypothetical Organizations. The case study assumed three hypothetical organizations for which the scale of the software metrics was to be determined. Detailed information about each of these organizations was provided to the stakeholders to help them develop an argumentation network. This information includes the company's profile, its organizational structure, its revenue, the scope of its projects, its product development features, team specifications, and the problems of small, medium, and large organizations.

i) Large Organization:

- Profile: A company develops mission-critical software; therefore software assurance is a major concern. It requires a software program to verify and validate techniques of risk mitigation. Under its sponsorship, numerous experiments have been designed and executed to study the flight dynamics applications. A software engineering laboratory (SEL) database was established to support the research on the measurement and evaluation of the software development process. The major functions of the SEL database include the collection of detailed software engineering data describing all facets of the development process, and the archiving of this data for future use. To this end, the SEL has created and maintained an online database for the storage and retrieval of software engineering data.
- Organizational Structure and Revenue: The organization has 180,000 employees spread out over 30 branches. The company has a budget of $18.7 billion, out of which about $2.2 billion is invested in software development. The number of employees engaged in the SELD is between 100 and 200. The number of projects is 15 per year. The development defect rate is 40%. The cost due to failure has been reduced by 12%, and the reuse of improvised software has increased to 8% over the last ten years.
- Types of Projects: About 52% of the organization's projects are real-time software projects divided into sequential phases. These projects have clear objectives and solutions. The emphasis is on planning, scheduling, target dates, budgets, and implementation of an entire system at one time. Tight control is maintained over the life of the project through the use of extensive written

documentation, as well as through formal reviews and approval/signoff by the user. Information technology management occurs at the beginning of each new phase to ensure the quality, reliability, and maintainability of the software.

Approximately 20% of the projects are maintenance projects that conserve resources and in which the progress of system development is measurable. In such cases, project requirements are stable throughout the system development life cycle. About 18% of the projects are large, expensive, and complicated. For such projects, there is no pressure for immediate implementation. The project requirements can be stated unambiguously and comprehensively. The requirements for 10% of the projects evolve continuously. For such projects, small-scale mock-ups of the system are developed following an iterative modification process until the prototype evolves to meet the users' requirements.

- Team Specifications: Experienced, flexible team members are needed from multiple disciplines. The user community is knowledgeable about the business and its application. Strict requirements exist for formal approval at designated milestones. Team members and the project manager are experienced. The team composition is stable. Stakeholders can be given concrete evidence of project status throughout the project life cycle
- Problems: The problems involved include delays of approximately 500 man hours in scheduled time estimation due to an inflexible, slow, costly, and cumbersome process demanded by a rigid structure and tight controls. Such delays occur in 2% of projects for which early identification and specification of requirements is not possible because users are initially unable to define their needs clearly. In 5% of projects for which there is no information about similar past projects, requirements are inconsistent, system components are missing, and unexpected development are discovered during design and coding. Overall project failures due to a lack of quality metrics reduce revenues by about 15%.

ii) Medium Organization:

- Profile: Company develops, markets, and sells a suite of software products for deploying Web applications. The company's Enterprise Web Suite combines portal, content management, collaboration, integration, and search technologies. It also offers development and administration tools that are used to assemble, customize, deploy, and manage applications for users inside and outside the enterprise.

- Organizational Structure and Revenue: This is a public company with 201 to 500 employees. Its net income is approximately 500 million. About 35.5% of the revenue is used for quality assurance. The number of projects is around 20 per year. The system is combination of hierarchical and flat. Small teams of developers, testers, and analysts work on the projects.
- Product Development Features: This organization aims to produce high quality systems quickly, primarily through the use of iterative prototyping (at any stage of development), active user involvement, and computerized development tools. These tools may include graphical user interface (GUI) builders, computer aided software engineering (CASE) tools, database management systems (DBMS), fourth-generation programming languages, code generators, and object-oriented techniques. The inherent project risk is reduced by breaking a project into smaller segments and providing more ease-of-change during the development process. The emphasis is on fulfilling the business need; technological or engineering excellence is less important.
- Project Types: In 70% of the projects, the project control is satisfied by prioritizing development and defining delivery deadlines. In 45% of the cases, if the project begins to slip, the emphasis is on reducing requirements to meet the deadline. There must be a significant savings of time, money, and human effort. Only about 10% of the employees can be trained to do quality assurance.

For 42% of projects the business objectives are well defined and narrow. These are the projects which are now in the maintenance stage. Data for such projects already exists (completely or in part), and the project largely comprises analysis or reporting of the data. Technical architecture is clearly defined. Key technical components are in place and tested. Technical requirements (e.g., response times, throughput, database size, etc.) are reasonable and well within the capabilities of the technology in use.

About 28% of projects every year introduce many new technologies, or involve a technical architecture that is unclear, and much of the technology will be used for the first time within the project. For such projects, requirements cannot be defined, accurately ahead of time. About 58% of the projects have small scope or short duration (e.g., 6 man years of development effort). The project scope is focused. Application is highly interactive, has a clearly defined user group, and is not computationally complex. The functionality of the system is clearly visible at the user interface.

- Customer Satisfaction: Users are intensely involved in system design through workshops. Rapid changes in the system design must be made based on user requirements. Users are understood to gain a sense of ownership of a system, whereas developers are understood to gain more satisfaction from producing successful systems quickly.
- Team Composition: Team members are skilled both socially and in terms of business. The team composition is stable. Project control is effective. Developers are skilled in the use of advanced tools. Users possess detailed knowledge of the application area. Senior management is committed to ensuring end-user involvement. The development team is empowered to make design decisions on a day-to-day basis without the need for consultation with their superiors, and decisions can be made by a small number of people who are available and preferably located at the same site.
- Problems: Module reuse and scalability may be difficult. Problems most often involve inconsistencies and misalignment due to missing information and incomplete documentation. Some problems may be pushed aside to demonstrate early success. Greater speed and lower costs sometimes lead to lower overall system quality. Projects may end up with more requirements than needed. The overall loss due to these issues is approximately 2.5% of the revenue.

iii) Small Organization:

- Profile: The company is develops innovative class of storage that makes time a dimension of storage. It is working on data revival and data protection. Its revenue is $17.4 million, and it has 20 employees. The continuous data protection scheme used is continuously changing and not predetermined. The company spends around $7 million for risk management and quality assurance. It has used its application in both the health and software industries.
- Product Development Features: The organization's focus is on risk assessment and minimizing project risk. It accomplishes these goals by breaking a project into smaller segments, ensuring ease-of-change during the development process, and providing the opportunity to evaluate risks and consider project continuation throughout the life cycle. Deadlines are firm. Approximately 65% of projects have a low risk of falling to meet user requirements but a high risk of missing budget or schedule targets. Each project is highly customized and thus is quite complex, limiting reusability. A skilled and experienced project manager is required. There are no established controls for moving from one cycle to another cycle. Without controls, each cycle may generate more work for the next cycle.

  Projects involve real-time or safety-critical systems in which risk avoidance is a high priority. Minimizing resource consumption is not a necessity. Strong approval and documentation control are needed. Project might benefit from a mix of other development methodologies. A high degree of accuracy is essential. Implementation has priority over functionality, which can be added in later versions.
- Problems: In about 38% of software deliverables, especially the large ones, the effort required at the beginning of the software development life cycle is difficult to assess. There is little emphasis on necessary designing and documentation. The project can easily get taken off track if the customer representative is not clear about the final outcome desired. In approximately 2% of cases the development cycle continues with no clear termination condition, creating the risk that the project will meet neither budget nor schedule.

- Team Composition: Projects have an adaptive team that is able to respond to changing requirements. The ideal development team size utilizing will comprise of 5 to 9 people. The project manager is highly skilled and experienced. Team members are from various departments depending on the deliverable required from the project. The team is responsible, mutually dependent and self-organizing. Thus, effective project communications are extremely important.

  One team member is assigned to elicit customer input, thus freeing the rest of the team to focus on the development process. Face-to-face communication and continuous inputs from customer representatives leave no space for guesswork. Documentation is crisp and to the point, thus saving time. Only senior programmers are capable of making development decisions; therefore the organization has no place for new programmers, unless they are paired with experienced individuals.

1.1.4. Survey 1. The first survey asked a focus group of 24 students to select the scale of a software metrics program in a large, medium, and small organization. The focus group was provided all the background information about the three hypothetical organizations introduced in here. Respondents had three days to form an opinion and prepare a written explanation of the reasoning behind their choice. The level of support for each alternative is shown in Figures 5.2, 5.3, and 5.4.

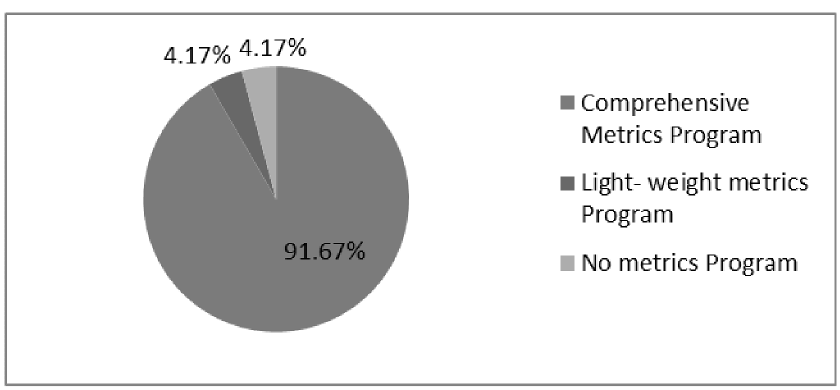


Figure 5.2. Scale of Software Metrics for Large Organization for Survey1

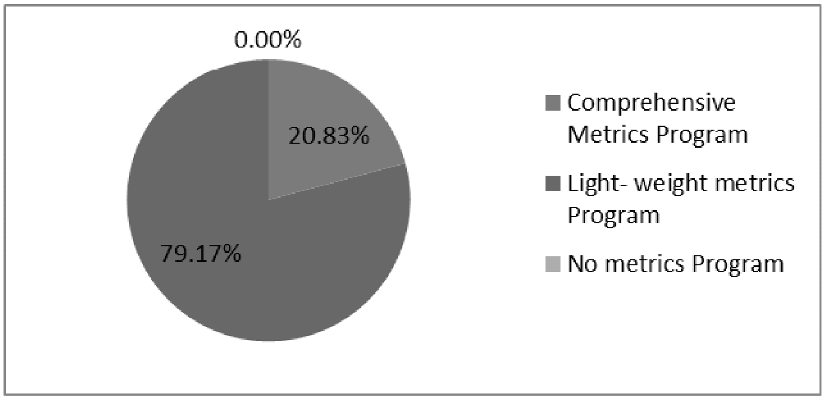


Figure 5.3. Scale of Software Metrics for Medium Organization for Survey1

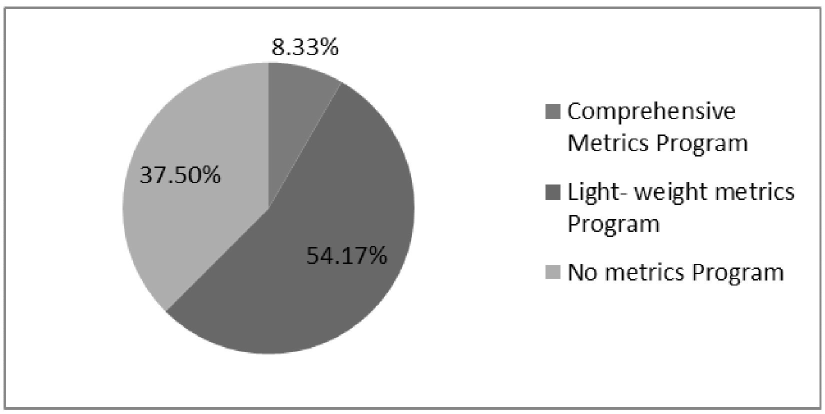


Figure 5.4. Scale of Software Metrics for Small Organization for Survey1

The respondents explained the criteria on which they based their choices. Figure 5.5 shows these criteria and indicates the percentage of respondents who relied on each.

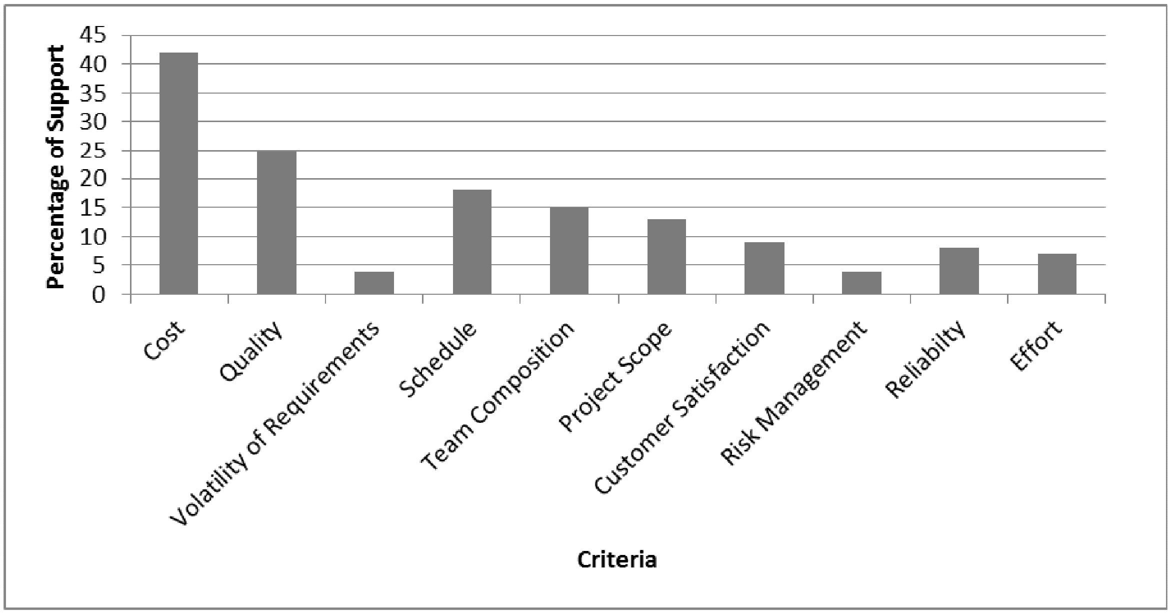


Figure 5.5. Percentage of Support per Criterion for Survey1

1.1.5. Argumentation Network. The group was introduced to an intelligent argumentation system and made familiar with its function. After responding to survey 1, the group was asked to give their feedback using the argumentation system. One week of argumentation time permitted for each issue, and the priorities of the stakeholders were evaluated based on the results. The priorities were updated after each issue and were used as input priorities for the next issue. In the argumentation system, the chosen alternative is called as the winning alternative, and the amount of support given to each alternative is known as its favorability factor. The favorability factors for the three issues are as shown in Table 5.1

Table 5.1. Favorability Factors for Issues 1, 2, and 3

Alternatives	Large	Medium	Small
No Metrics Program	-13.910	-17.103	-15.554
Light weight Metrics Program	-2.398	32.968	24.413
Comprehensive Metrics Program	18.868	-12.378	-2.656

A total of 563 arguments were posted for the three issues, and each argument posted by a stakeholder was based on certain criteria. The graph as shown in Figure 5.6 shows the criteria. It is similar to that developed in the survey 1; however, the respondents generated several new criteria during argumentation that has not been mentioned in response to the survey. In addition, the level of support for criteria changed from the survey to the argumentation process.

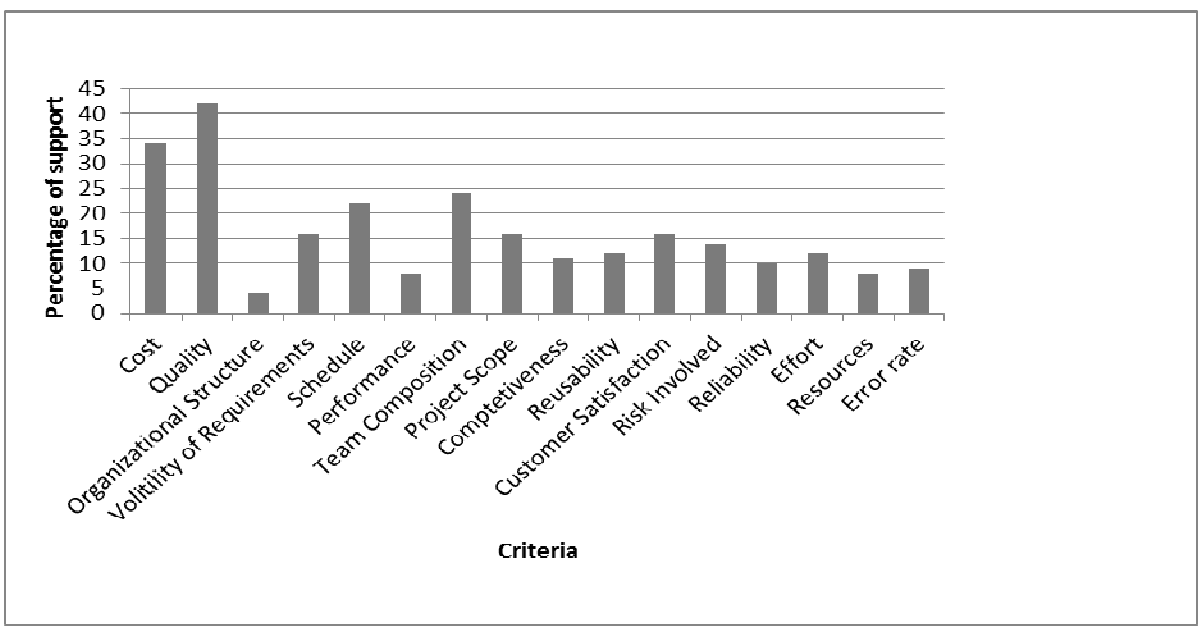


Figure 5.6. Percentage of Support per Criterion for Argumentation Network

1.1.6. Survey 2. After three weeks of argumentation, the respondents were asked to complete survey 2. The second survey posed the same questions asked on survey1. In addition, however it included a section in which the stakeholders were asked to describe how the argumentation process affected their decision making. Respondents were asked to select the scale of software metrics to be used for each of the three hypothetical organizations. This survey was conducted to study the impact of argumentation on the thought process of the stakeholders. The results obtained are shown in Figures 5.7, 5.8, and 5.9.

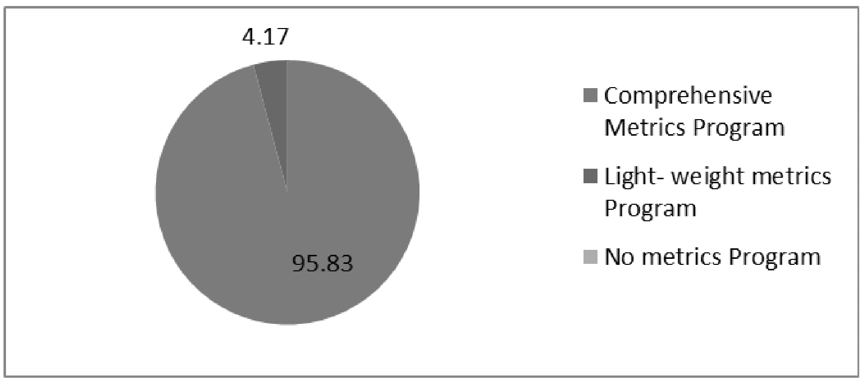


Figure 5.7. Scale of Software Metrics for Large Organization for Survey2

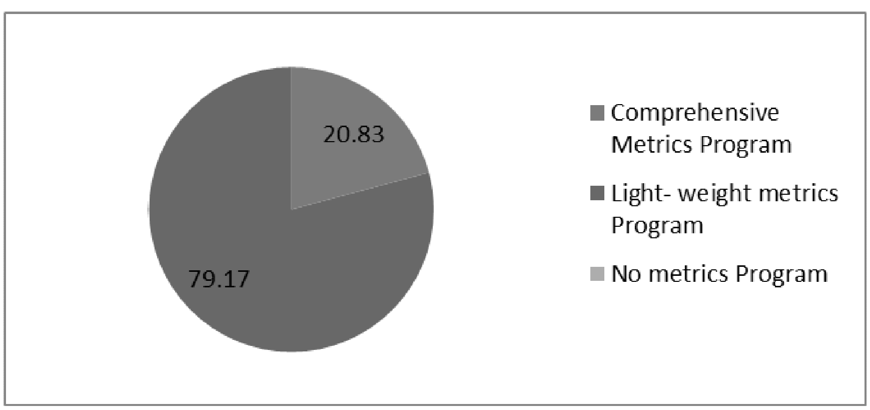


Figure 5.8. Scale of Software Metrics for Medium Organization for Survey2

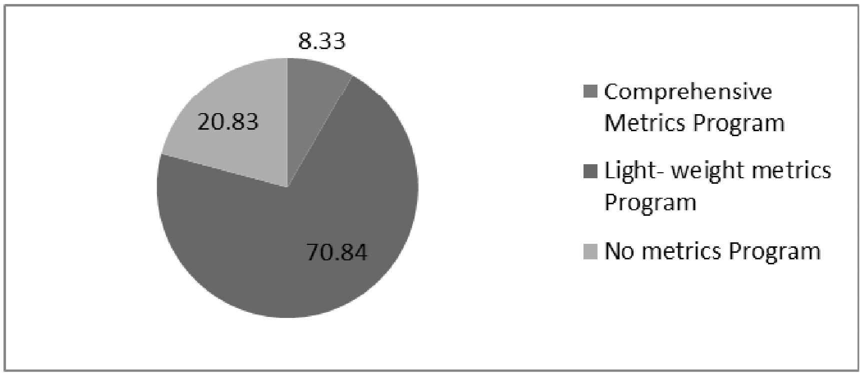


Figure 5.9. Scale of Software Metrics for Small Organization for Survey2

These figures demonstrate that the criteria on which the respondents based their decision changed significantly after argumentation. Figure 5.10 shows the criteria graph developed after survey 2.

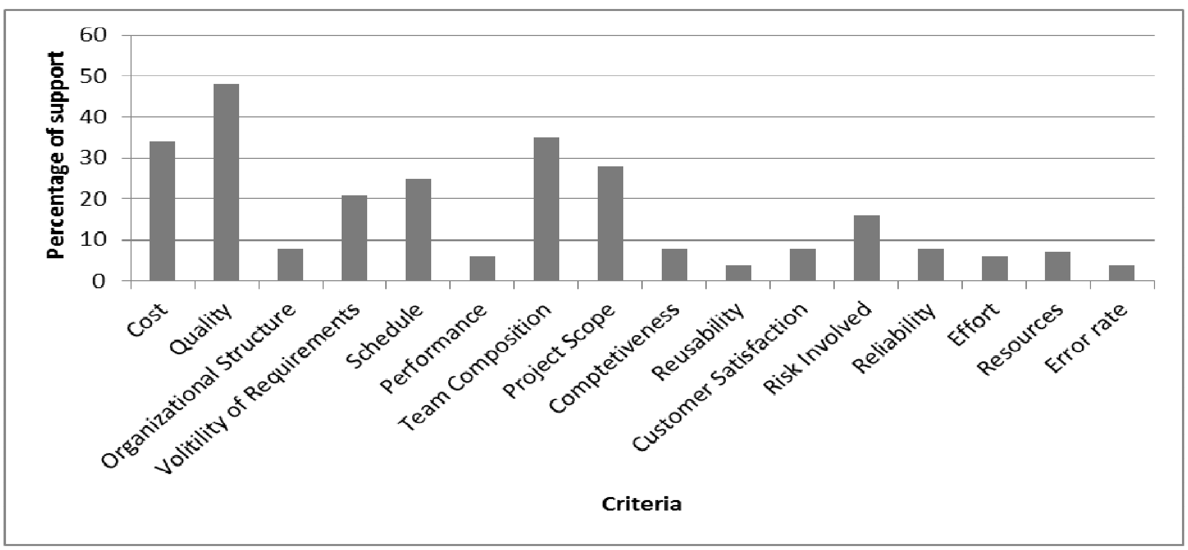


Figure 5.10. Percentage of Support per Criterion for Survey2

5.2. EMPIRICAL ANALYSIS

This study was carried out over a period of four weeks. It investigated two important issues: the impact of argumentation on the decision making process and the impact of dynamic priority assessment on the decision making process.

5.2.1. Impact of Argumentation on Decision Making Process. In order to understand the extent to which argumentation improves decision making, this work conducted two surveys. Survey 1 reflects the individual decisions of each stakeholder without consideration of the perspectives of other group members. Survey 2 reflects the impact of argumentation on decision making. This impact is apparent from respondents' evaluation of the effects of argumentation during the feedback after survey 2.

The stakeholders observed two major effects of argumentation on the decision making process. First, they noted that argumentation clarified the issues and improved their confidence. Second, they indicated that argumentation affected their opinions on the issues. Survey 1 showed that each stakeholder had an opinion, but the criteria on which they based their opinion were limited because they viewed the issues from a narrow perspective.

After the second survey, the stakeholders indicated that they were more comfortable in making decisions because argumentation had given them more evidence for their decisions and exposed them to new criteria as well. Figure 5.11 compares the criteria on which respondents based their decisions during survey1, argumentation, and survey2.

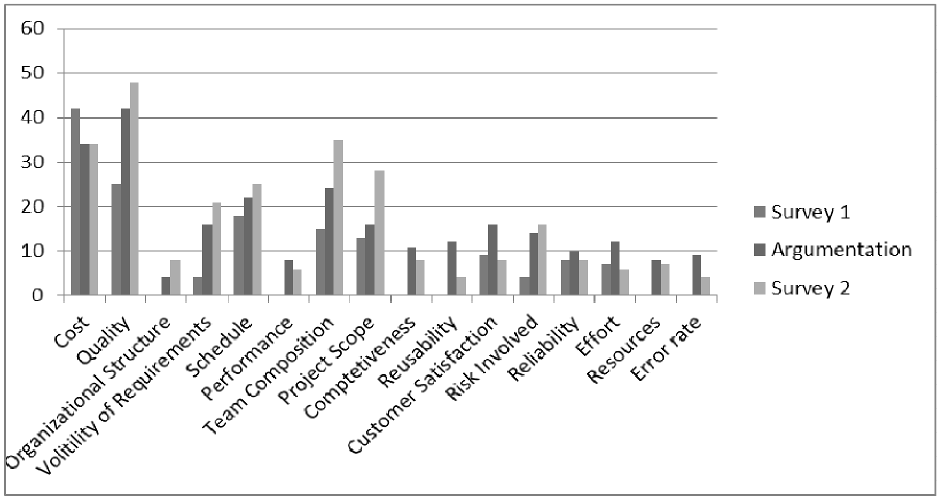


Figure 5.11. Percentage of Support per Criterion Comparison Graph

Figure 5.11 indicates that several criteria were noted during argumentation and survey 2 that were not mentioned in survey1: organizational structure, performance, competitiveness, reusability, resources, and error rate 1. Cost was the only criterion that was more significant in Survey 1 but after the argumentation the other criteria were also discussed in depth. The significance of all other criteria grew after argumentation and survey 2. These observations demonstrate that the argumentation network had an effect on the thought process of the stakeholders.

To quantify the impact of argumentation on the decision making process, this work measured the percentage shift in the favorability factor from survey 1 to survey 2 as shown in Table 5.2.

Table 5.2. Percentage Shift towards Winning Alternatives from Survey 1 to Survey 2

Issue	Percentage shift towards the winning alternative
Scale of metrics for Large Organization	4.17%
Scale of metrics for Medium Organization	None
Scale of metrics for Small Organization	16.67%

Table 5.2 shows that there was only a small shift on the first issue but a considerable shift on the third issue. The second issue showed no change of opinion; however, the feedback from many respondents mentioned that argumentation provided strong confirmation of their opinion.

This study used various quality metrics to evaluate the effectiveness of the argumentation process. These quality metrics define the relevance of argumentation to the issues, they are discussed in detail below.

i) Relevance refers to the quality of the arguments posted by the stakeholders: Some arguments posted may be irrelevant, or redundant, or of poor quality. In such cases, the decision may not be the right one. This case study, classified the arguments as excellent, good, or average based on their relevance to the current issue, as shown in Figure 5.12. This classification determines whether argumentation has achieved good results.

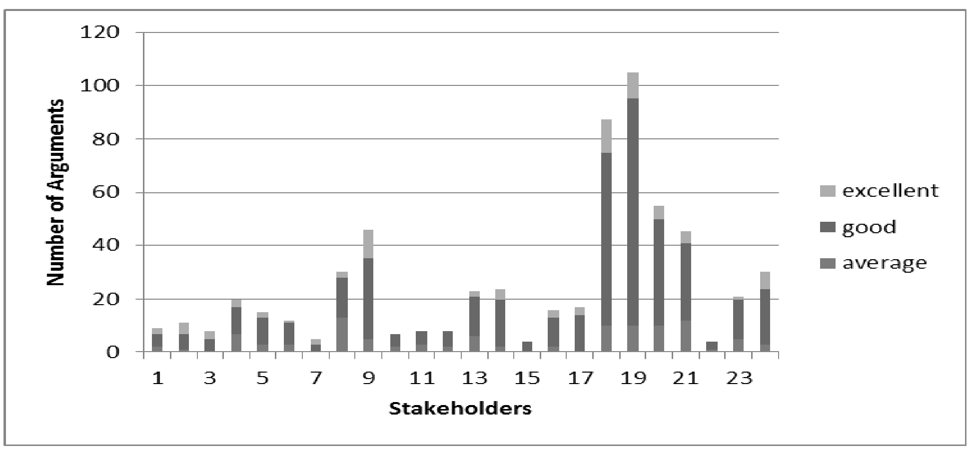


Figure 5.12. Stakeholder's Argument Relevancy Count

Figure 5.12 demonstrates that the majority of the arguments were good and relevant. Some were excellent in terms of relevance and clarity. The average arguments were relevant, but they offered no valid justification.

ii) Support/Attack: This metric calculates the percentage of support for and attack of a particular alternative. It is calculated in terms of the number of arguments as shown in Figures 5.13, 5,14 and 5.15 and it is relevant to the favorability factors being computed.

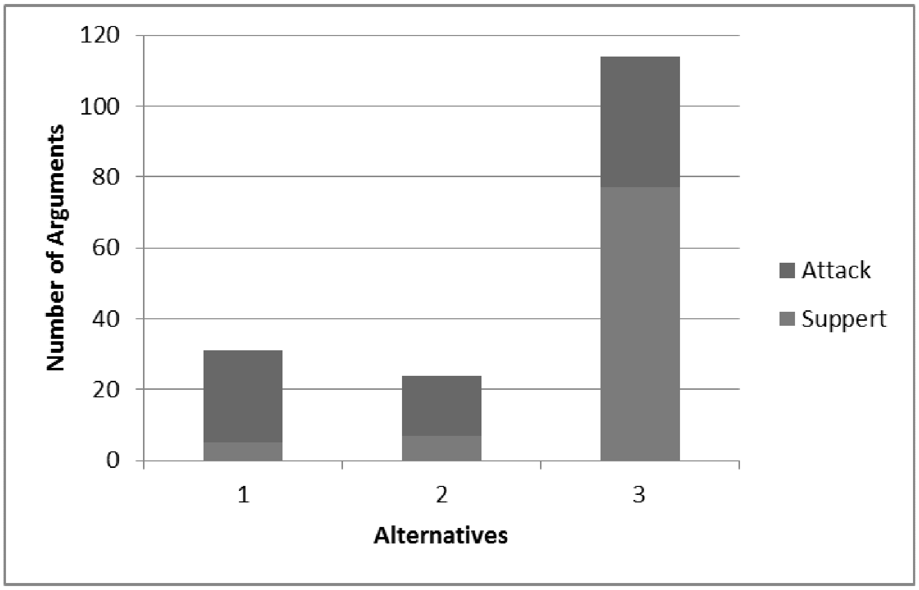


Figure 5.13. Percentage of Support and Attack for Alternatives of Issue 1

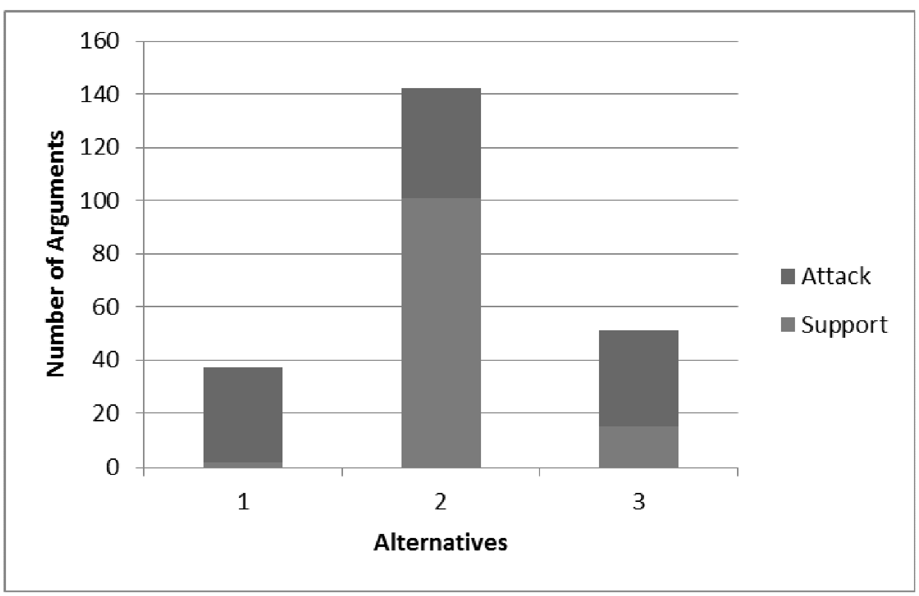


Figure 5.14. Percentage of Support and Attack for Alternatives of Issue 2

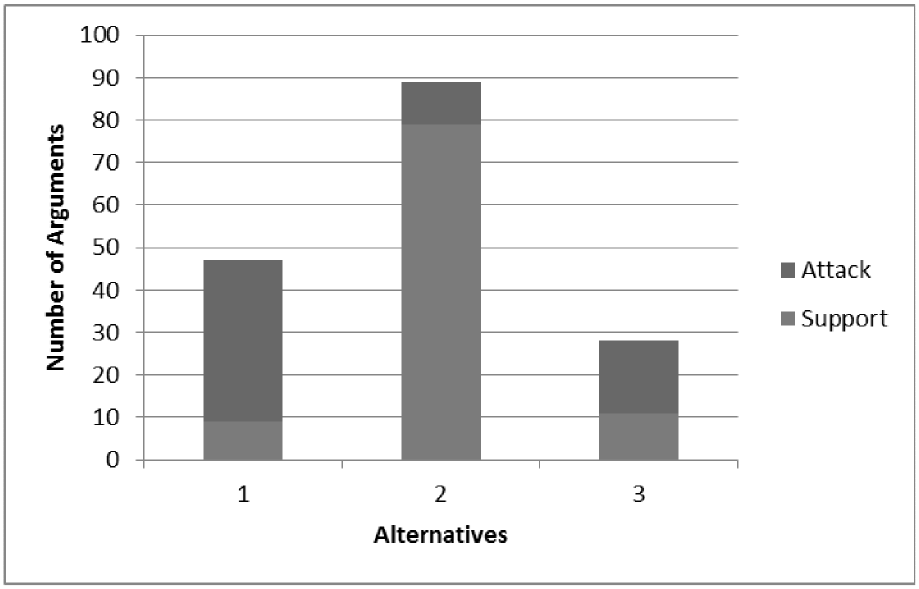


Figure 5.15. Percentage of Support and Attack for Alternatives of Issue 3

The values shown in the graph in Figures 5.13,5,14, and 5.15 were used to calculate the percentage of support for each alternative. These calculated values show the relevance of the argumentation with respect to the results obtained from Survey 2, as shown in Table 5.3.

Table 5.3. Percentage of Support for Alternatives within Argumentation Network

Alternatives	Large Organization	Medium Organization	Small Organization
No metrics program	5%	1%	9%
Light Weight Metrics Program	7%	86%	80%
Comprehensive Metrics Program	88%	13%	12%

iii) Depth of argumentation tree: Another metric to that can be used to evaluate the role of effective argumentation is the level up to which the arguments have been posted, as shown in Table 5.4. This level indicates the amount of the active participation by the stakeholders and how closely each stakeholder follows the argumentation network itself. This calculation revealed significant patterns in the nature of the arguments posted at various depths. Although most of the arguments were posted to convey the favorability of stakeholders towards a particular alternative, some were also posted as part of a self-correction process. These arguments minimized the effect of a particular argument that had provided an incorrect weight. Here, weight signifies either a higher or a lower weight values than the required one.

Table 5.4. Depth of Argumentation Tree

% of arguments Issue 1	% of arguments Issue 2	% of arguments Issue 3	Depth of Tree
48.8	56.8	72.8	Level 1
32.2	27.2	22.2	Level 2
13.2	11.2	4.2	Level 3
4.4	2.9	0.8	Level 4
1.4	1.3	-	Level 5
-	0.6	-	Level 6

5.2.2. Impact of Dynamic Priority Assessment on Decision Making Process. Dynamic priority assessment was incorporated in the argumentation process to improve the quality of the decision making. All stakeholders began with an initial priority of 0.5. After each issue their priorities were calculated. The new priorities were based on the stakeholder’s contribution to the winning alternative. Thus, stakeholders who have a record of making good decisions have a higher priority.

This work used two metrics to evaluate this process; sensitivity analysis and the impact of arguments on priorities.

i) Sensitivity Analysis: The stakeholders were initially assigned the same priority values, giving them equal stature in the decision making process. Sensitivity analysis of stakeholder determines how the favorability of positions changes as participant priorities change. Two scenarios were used, as shown in Tables 5.5 and 5.6. The percentage change in the favorability factors was also calculated to demonstrate the qualitative impact of the dynamic priority assessment scheme.

Scenario 1

The argumentation network was used to resolve three issues in the case study. In the first scenario, the same priority, i.e. 0.5, was given to each stakeholder for each issue, and this priority value was held as constant for all three issues.

Table 5.5. Favorability Factors without Dynamic Priority Assessment

Alternatives	Large	Medium	Small
No Metrics Program	-13.910	-16.653	-15.160
Light weight Metrics Program	-2.398	31.827	22.386
Comprehensive Metrics Program	18.868	-11.975	-2.407

The favorability factors for each of the three alternatives were computed using the intelligent argumentation system. Alternative 3 was most favored for Issue 1, and alternative 2 was most favored for Issues 2, and 3, as shown in Table 5.5.

Scenario 2

In the second scenario, the priority values were reassessed after each issue was decided. The resulting priority values were given as input to the next issue. The priority values were calculated based on the contribution made by each participant. The favorability factors of each alternative for each issue were computed using the intelligent argumentation system. Alternative 3 was most favored for Issue 1, and Alternative 2 was most favored for Issues 2, and 3, as shown in Table 5.6. Figure 5.16 shows the change in the priority of the participants after each issue.

Table 5.6. Favorability Factors with Dynamic Priority Assessment

Alternatives	Large	Medium	Small
No Metrics Program	-13.910	-17.103	-15.554
Light weight Metrics Program	-2.398	32.968	24.413
Comprehensive Metrics Program	18.868	-12.378	-2.656

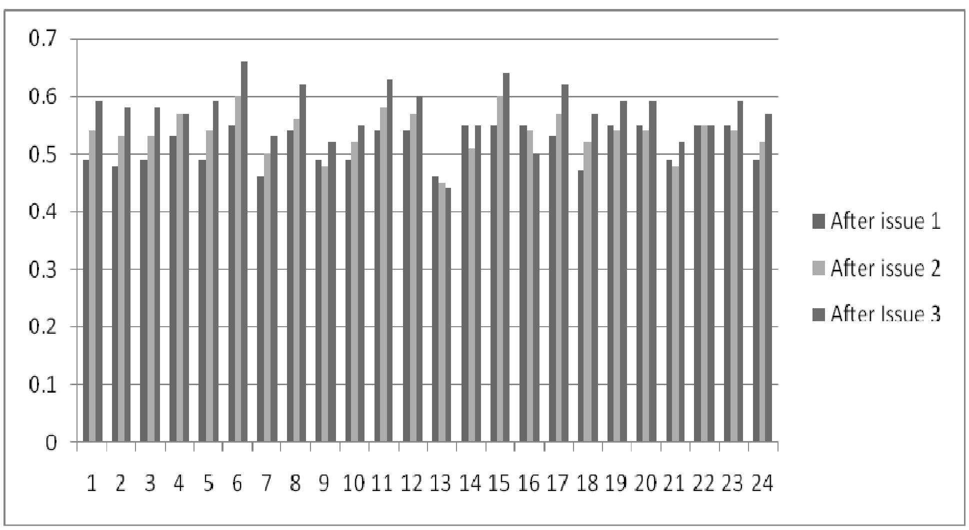


Figure 5.16. Reassessed Priority Values of Stakeholders for Issues 1, 2, and 3

The two scenarios demonstrate that changes in the priorities of participants affected the favorability of the alternatives. In other words, the results were sensitive to priority assessment. To determine whether priorities improve the quality of decision making, the percentage change in the favorability factors of each alternative was calculated for each issue, as shown in Table 5.7.

Table 5.7. Percentage change in Favorability Factors of Winning Alternatives

Alternatives	Medium	Small
Light weight Metrics Program	3.5%	7%

Table 5.7 shows the percentage increase in support for each alternative. Therefore, the modified priorities impact the quality of the decision made. The normalized values of the favorability factors for the two scenarios are shown in Table 5.8 and Table 5.9, respectively.

Table 5.8 Normalized Favorability Factors without Dynamic Priority Assessment

Alternatives	Medium	Small
No Metrics Program	0	0
Light weight Metrics Program	48.48	37.546
Comprehensive Metrics Program	4.678	12.753

The percentage of support for the winning alternative for issue 3 is 74.5% for Scenario1 and 75.6% for the Scenario 2. Table 5.10 shows that the normalized percentage support for the winning alternative of Issue 3 is 83%.This shows that results of the Survey 2 are more consistent with the results of the argumentation when dynamic priority reassessment was used. Therefore dynamic priority reassessment helps in obtaining relatively more accurate results.

Table 5.9 Normalized Favorability Factors with Dynamic Priority Assessment

Alternatives	Medium	Small
No Metrics Program	0	0
Light weight Metrics Program	50.671	39.967
Comprehensive Metrics Program	4.75	12.898

Table 5.10 Normalized Percentage Support for Alternatives for Survey2

Alternatives	Medium	Small
No Metrics Program	0	17%
Light weight Metrics Program	79.17%	83%
Comprehensive Metrics Program	20.83%	0

ii)Impact of Arguments on the Priorities: To analyze the function of the priority assessment scheme, a metric was developed to determine the impact of contributions on the priority values. For each stakeholder, the number of positive and negative arguments posted for the winning alternative was calculated. These are the positive and negative contributions shown in Figures 5.17, 5.18, and 5.19. The results were compared with the change in priority values.

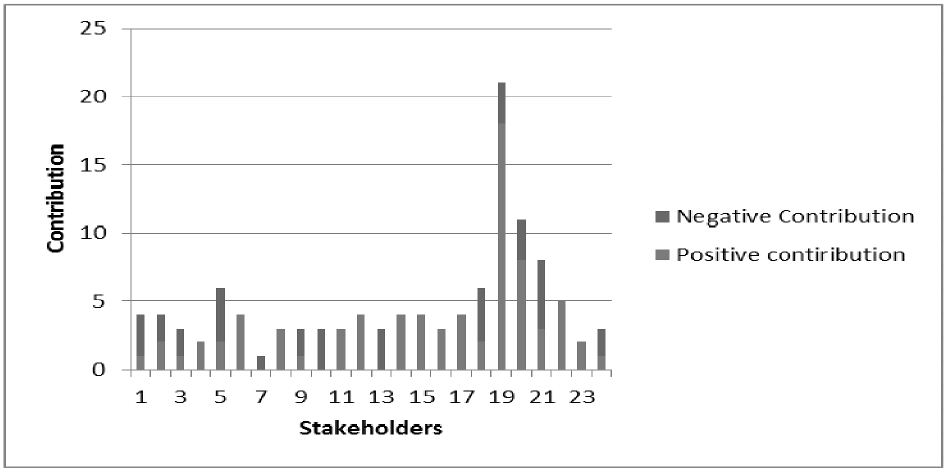
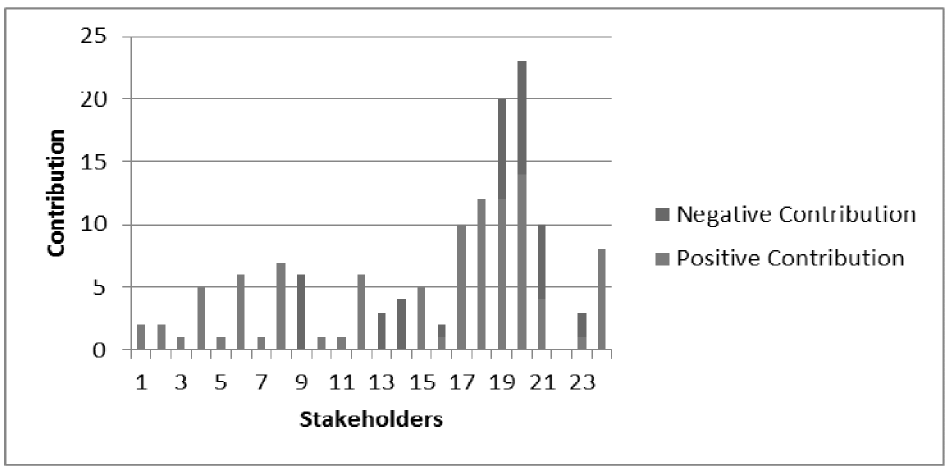


Figure 5.17. Percentage of Contributions to Winning Alternative for Issue 1



\Figure 5.18. Percentage of Contributions to Winning Alternative for Issue 2

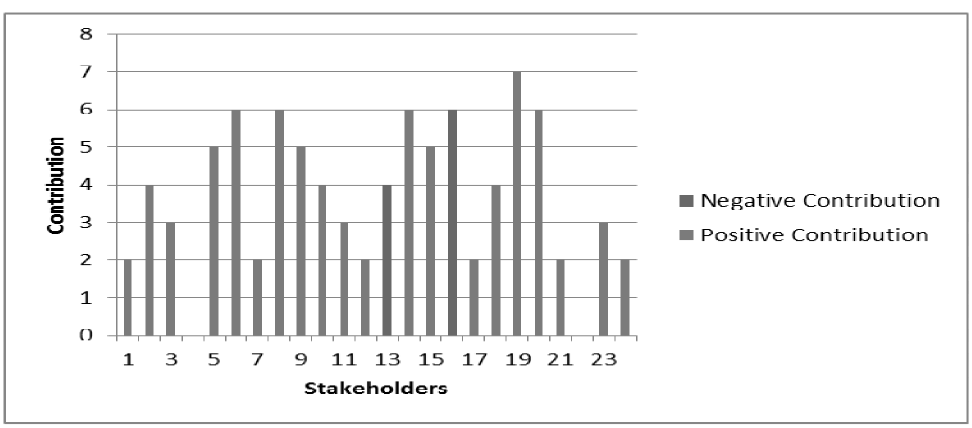


Figure 5.19. Percentage of Contributions to Winning Alternative for Issue 3

Comparison of the values in Table 5.6 with the graphs reveals a trend in the priority change based on the contributions of stakeholders. Figure 5.19 shows that only two stakeholders contributed negatively to the winning alternative; therefore their final priority values decreased.

The priority of the stakeholders decreases if they make a negative contribution towards the winning design alternative. This has made the stakeholders to be extremely careful while choosing the design alternative. This can be understood by the decrease in the negative contribution which is observed after each issue.

The number of people who have been contributing negatively to the winning design alternative has decreased after each case study. In the first case study around 13 stakeholders out of 24 made negative contribution as shown in Figure 5.17. In the second case study it reduced to 8 people who contributed negatively towards the winning alternative as shown in Figure 5.18. In the third case study there were only 2 people who contributed negatively towards the winning alternative as shown in Figure 5.19.

This improvement in the design decision shows that the dynamic priority assessment is helping the stakeholders to reconsider their options and to make a better quality decision. This is an improvement over the static method because when the priorities change based on the contribution of the stakeholders, the stakeholders are more careful while contributing their opinion as the decreased priorities effects their reputation.

CONCLUSION AND FUTURE WORK

The main contribution of this thesis is the development of a dynamic contribution-based priority assessment for a web-based argumentation based collaborative system. The incorporation of participant's contributions to decisions in priority assessment grants successful contributors a higher priority in an argumentation process for collaborative software development engineering design. The empirical study shows that an argumentation process and the quality of decisions are improved as a result of use of dynamic priority assessment.

In the future, the proposed argumentation method and priority assessment tools will be evaluated using a range of real-world applications. The current dynamic priority assessment scheme involves only the contribution to the winning alternative, this scheme can be extended to incorporate contributions to other alternatives as well.

BIBLIOGRAPHY

[1] Liu, X. F., Zheng, M., Leu, M. C., Khudkhudia, V. Wen, L. and Sajja,V., "An Intelligent Computational Argumentation System for Supporting Collaborative Software Development Decision Making," accepted for publication as a chapter of a book: Artificial Intelligence Application for Improved Software Engineering Development, IGI, 2008.

[2] Liu, X. F., Raorane, S., Zheng, M., and Leu, M. C., "An Internet Based Intelligent Argumentation System for Collaborative Engineering Design," Proceedings of 2006 IEEE International Symposium on Collaborative Technologies and Systems (CTS 2006), Las Vegas, Nevada, May 14-17, 2006, pp. 318-325.

[3] Toulmin, S. E. "The Uses of Argument", Cambridge, UK, University Press,1958.

[4] Conklin, J. and Begeman, M., "gIBIS: A Hypertext Tool for Exploratory PoliDiscussion," Transactions on Office Information Systems, 1988.

[5] Ramesh, B. and Dhar, V., "Supporting Systems Development by Capturing Deliberations during Requirements Engineering," IEEE Transactions on Software Engineering, 1992.

[6] Sillence, J., "Intelligent Argumentation Systems: Requirements, Models, Research Agenda, and Applications," in Allen Kent (Ed.), Encyclopedia of Library and Information Science, 1992.

[7] Potts, C., and Burns, G., "Recording the Reasons for Design Decisions," Proceedings of the 10th international conference on Software engineering, Singapore, 1988.

[8] Karacapilidis, N., and Papadias D., "HERMES: Supporting Argumentative Discourse in Multi-Agent Decision Making," Proceedings of 15th National Conference on Artificial Intelligence (AAAI-98), Madison, Wisconsin, AAAI/MIT Press, 1998.

[9] Pike, W. A., May, R., Baddeley B., Riensche R., Bruce J., and Younkin K., "Scalable Visual Reasoning: Supporting Collaboration through Distributed Analysis," Proceedings of 2007 IEEE International Symposium on Collaborative Technologies and Systems (CTS 2007) Orlando, Florida, May 21-25, 2007.

[10] Saaty, T. L., "Fundamentals of Decision Making and Priority Theory with the Analytical Hierarchy Process," RWS Publications, Pittsburgh. PA, 1994.

[11] Carmone, F. J., Kara, A., and Zanakis, S. H., “A Monte Carlo investigation of Incomplete Pairwise Comparison Matrices in AHP,” European Journal of Operational Research, 1997.

[12] Liu, X. F., S., Sun, Y., Veera, C. S., Kyoya, V., and Noguchi, K., “Priority Assessment of Software Process Requirements from Multiple Perspectives,“ Journal of Software and Systems, 2006.

[13] Liu, X. F., Zheng, M., and Leu, M. C., “Management of an Intelligent Argumentation Network for a Web-Based Collaborative Engineering Design Environment” Proceedings of 2007 IEEE International Symposium on Collaborative Technologies and System (CTS 2007) Orlando, Florida, May 21-25, 2007.

[14] Liu, X. F., Khudkhudia, E., and Leu, M. C., “Incorporation of Evidences into an Intelligent Computational Argumentation Network for a Web-Based Collaborative Engineering Design System” Proceedings of 2008 IEEE International Symposium on Collaborative Technologies and System (CTS 2008) Orlando, Florida, May 19-23, 2008.

[15] Liu, X. F., Satyavolu, M., and Leu, M. C., “Contribution based Priority Assessment in a Web-based Intelligent Argumentation Network for Collaborative Software Development” Proceedings of 2009 IEEE International Symposium on Collaborative Technologies and System (CTS 2000) Baltimore, May 18-22, 2008.

VITA

Maithili Satyavolu was born on July 8, 1986 in Visakhapatnam, India. She received her B.E. degree in Computer Science and Engineering from Anil Neerukonda Institute of Science and Technology in 2008. She has been a graduate student in the Computer Science Department at Missouri University of Science and Technology since August 2008 and worked as a Graduate Research Assistant under Dr. Frank Liu from August 2008 to May 2009. She will receive her Masters in Computer Science at Missouri University of Science and Technology (formerly University of Missouri - Rolla) in December 2010.