REQUIREMENTS NEGOTIATION PRACTICES: A SYSTEMATIC REVIEW

RESEARCHER:

SAJID RIAZ

REG. # 182-FBAS/MSSE/F07

SUPERVISER:

DR. NAVEED IKRAM

ASSOCIATE PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF BASIC AND APPLIED SCIENCES

INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD

(2011)

1- System programs

2- Operating system

REQUIREMENTS NEGOTIATION PRACTICES: A SYSTEMATIC REVIEW

BY

SAJID RIAZ

REG. # 182-FBAS/MSSE/F07

A Thesis

Submitted in Partial Fulfillment of the Requirements for the Award of Degree of

Master of Science (MS)

in Software Engineering

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF BASIC AND APPLIED SCIENCES

INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD

(September 2011)

International Islamic University Islamabad
Faculty of Basic and Applied Sciences
Department of Software Engineering

Dated..17-10-11..........................

FINAL APPROVAL

It is certified that we have read the thesis submitted by **Mr. Sajid Riaz** of **Reg. no. 182-FBAS/MSSE/F07**. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by International Islamic University Islamabad (IIUI) for the degree of Master of Science in Software Engineering.

VIVA VOCE COMMITTEE

EXTERNAL EXAMINER:

Dr. Rashid Ahmad

Associate Professor

Department of Computer Engineering

NUST, College of Electrical and Mechanical Engineering (E&ME)

Rawalpindi

INTERNAL EXAMINER:

Mr. M. Usman

Assistant Professor

Department of Software Engineering, FBAS, IIUI

SUPERVISOR:

Dr. Naveed Ikram

Associate Professor

Faculty of Computing

Riphah International University

Islamabad

DECLARATION

I hereby declare that my whole thesis or part of it is not plagiarized from any other source. This is purely my own work. Such type of work has not been published yet by any other individual. This work was not used for any other degree in any university or institute. I wrote this thesis with my personal efforts. If it is found plagiarized from any other source, I shall be responsible for the consequences. Where necessary, I used references and quoted the text of other authors.

SAJID RIAZ

182-FBAS/MSSE/F07

ACKNOWLEDGEMENT

There are always unseen hands behind the success of every person. Similar is the case with me. I would like to thank for blessings of Almighty Allah who gave me courage to complete this work. I also thank my friends and fellows who assisted me during the completion of this work. Then I am grateful to my supervisor Dr. Naveed Ikram who assisted me in my work as much as possible. His kind, valuable and timely feedback made it possible for me to do such a work. Without his assistance I might not be able to complete this work in such a manner. He also encouraged me time to time and such encouragement raised my morale and I kept on doing it with patience and consistency. This is all due to my supervisor who directed me so well that I completed my work easily.

LIST OF ACRONYMS

Abbreviation	Title
SLR	Systematic Literature Review
RE	Requirements Engineering
EBSE	Evidence-Based Software Engineering
EBM	Evidence-Based Medicine
SRN	Software Requirements Negotiation
SDLC	Software Development Life Cycle
PICOC	Population, Intervention, Comparison, Outcome, and Context
QAC	Quality Assessment Criteria
QARCC	Quality Attribute Risk and Conflict Consultant
S&C's	Simplifiers and Complicators
ARENA-M	Anytime, anyplace REquirements Negotiation Aids-Mobile
CBSP	Component-Bus-System-Property
ATAM	Architecture Tradeoff Analysis Method
ICRAD	Integrated Conflict Resolution and Architectural Design
CBAM	Cost Benefit Analysis Method
MBASE	Model-Based Architecture in Software Engineering
NLP	Natural Language Processing
IBIS	Issue Based Information Systems
SQFD	Software Quality Function Deployment
FTR	Formal Technical Review

ABSTRACT

Context: Software requirements negotiation is a valuable area of interest in the field of software engineering. There is a lack of cumulative empirical knowledge of best requirements negotiation practices in the field of requirements engineering.

Objective: The objective of this study is to identify evidence-based requirements negotiation practices, their strengths and limitations. The review finds out and classifies the types of requirements conflicts which are addressed by these practices.

Methods: A systematic literature review has been carried out to answer the review questions. Initially data was retrieved from electronic databases using a query. Then study selection criteria were applied for filtering papers. The quality assessment criteria were applied on the previously selected 79 studies. In the result of quality assessment we obtained only 18 empirical studies. The data was extracted from these 18 studies after reading full text of all articles.

Results: The results showed that winwin theory acts like a backbone in the field of requirements negotiation. Most of the practices have been originated from Theory-W. There are very rare negotiation practices which are not winwin based. The evidence collected in the study showed that winwin based practices were only 7% used in all type of industries and the most widely used negotiation style in the industries was accommodating (87%).

Conclusion: It was observed that collaborative behavior was not possible all the times especially in threatening issues. There is no approach in the field of software engineering which can guide what to do if winwin does not work. The research projects in industrial setting are less as compared to academia therefore need for conducting more studies in industrial setting.

Keywords: Systematic review, systematic literature review, software requirements negotiation and software requirements negotiation practices.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1. Introduction

This chapter introduces the thesis message to the audience. It provides a clear description of problem statement, research question, scope of research work and contextual background information. The chapter will also give a brief detail of research methods used to answer the research question. In the end, the outline of chapters will provide a quick glance for the thesis audience.

1.1. Background

Conflicts play a significant role in the field of software engineering although people avoid them and the existing practices handle them inadequately (Grünbacher & Seyff, 2005). Conflicts are inevitable and hard to manage in software projects when the number of requirements and stakeholders increase in size. The problem arises when we need to satisfy all project stakeholders because their concerns vary from one another (B. Boehm, Port, & Al-Said, 2000). For instance, the interests of users, developers, testers, maintainers and customers are entirely different (Grünbacher & Seyff, 2005). The user needs more features and high response time whereas customer requires minimizing the budget and schedule while the developer forces for increase in budget and schedule. All these situations cause serious conflicts which need to be managed systematically. The evidence shows that conflict is massive in the field of software engineering (Curtis, Krasner, & Iscoe, 1988). Therefore, it is essential to handle these conflicts to ensure the success of software projects.

The primary goal of requirements negotiation is identification and resolution of requirements conflicts among project stakeholders. It has been realized as a success-critical factor in various disciplines like management, social sciences, politics, etc (Easterbrook, 1994). The requirements negotiation is a significant activity carried out during requirements analysis phase of requirements engineering (RE). RE is a field of software engineering which is responsible for acquiring negotiated requirements from project stakeholders. There are several requirements negotiation tools, techniques and methodologies which are used for handling requirements conflict. Some of these methods are also available for commercial purpose (Grünbacher & Seyff, 2005). The negotiation practices have been widely studied by the researchers and practitioners these days (Grünbacher & Seyff, 2005) therefore sufficient evidence is available and accessible in this area of interest.

1.2. Problem Identification and Motivation

There is a lack of cumulative empirical knowledge of best practices of requirements negotiation. Such type of work does not exist in the literature which can increase the knowledge of researchers and guide practitioners to choose a requirements negotiation practice in a particular conflict under certain conditions in software projects.

1.3. Aim of Research

The aim of research is to find out kinds of conflicts which are addressed in the literature. Which requirements negotiation practices resolve these conflicts? In the end, we will find pros and cons of these practices reported in the literature. No new practice will be created.

1.4. Scope of Research

Only conflicts of requirements engineering phase of software development life cycle will be part of this research work. The conflicts of other phases of SDLC will not be included. The negotiation in general is also not a part of scope of this research. Therefore, negotiation will be studied only in the context of software requirements. We will include only those requirements negotiation practices which are evidence-based. We will collect only few types of evidence like field/case studies, experiments and industrial/experience reports. The other types of evidences will not be included like expert opinion, survey, etc. The requirements negotiation practice means any framework, methodology, tool, technique, process or model available in the literature will be included.

1.5. Research Question

1. What requirements negotiation practices are used to resolve requirements conflicts in software projects?

   1.1. What kinds of conflict exist in literature that is resolved by requirements negotiation in software projects?

   1.2. What are the reported strengths of existing requirements negotiation practices?

1.3. What are the reported weaknesses of existing requirements negotiation practices?

1.6. Research Methods

We need at least two research methods to answer this research question because some parts of question are evidence-based while only one sub question i.e. 1.1 is not evidence–based. To answer evidence-based questions, we will use evidence-based method while other question will be answered through literature review. The evidence-based questions are similar to the question type addressed by Kitchenham and Charters in his guidelines (B. Kitchenham & Charters, 2007). Therefore, *systematic literature review* is the most suitable method for addressing evidence-based research questions in the field of software engineering.

The other research methods like case study, experiment, simulation, surveys, ethnography, action research, benchmarking are not appropriate. All these methods are not applicable to do such kind of research because these methods are not evidence-based. We need a method that should be evidence-based in nature. Therefore, systematic review is an appropriate research method for this kind of research work.

1.7. Chapters Outline

This section will provide a brief detail of next chapters included in the thesis. This section will help the readers to understand the body of the thesis in brief.

Chapter 1: Introduction

This chapter introduces the thesis message to the audience. It provides a clear description of problem statement, research question, aim of research, scope of research work and contextual background information. The chapter also gives a brief detail of research methods used to answer the research question.

Chapter 2: Requirements Negotiation and Requirements Conflicts

The purpose of this chapter is to introduce the importance of requirements negotiation in the field of requirements engineering (RE). It will also provide the concept of requirements conflicts and then classification of conflicts will be explained to support the answer of research question which has been answered through literature review.

Chapter 3: Systematic Review Process

This chapter explains the concept of evidence based software engineering (EBSE). This chapter also describes process of systematic review with its three phases and respective activities of these phases. This chapter focuses the first phase of systematic review i.e. planning the review. The major contribution of this chapter is identification of the need for systematic review and development of a review protocol. All the contents of review protocol are explained individually. Threats to the validity of review protocol design are also explained.

Chapter 4: Conducting the Review

This chapter focuses on the execution of review protocol design which is the second phase of SLR. This chapter explains the results of all steps included in the execution of SLR which are identification of research, study selection, study quality assessment, data extraction and data synthesis. In the end, the results of the systematic review are compiled.

Chapter 5: Conclusion

This chapter explains *what the thesis said.* The chapter concludes the whole research work and the major findings of the systematic review are discussed. The strengths and weaknesses of the research work are also reported. Then future research directions are proposed for future work.

CHAPTER 2

REQUIREMENTS NEGOTIATION AND REQUIREMENTS CONFLICTS

2. Requirements Negotiation and Requirements Conflicts

The purpose of this chapter is to introduce the role of requirements negotiation in requirements conflicts handling. The chapter will also categorize different types of conflicts available in the literature.

2.1. Requirements Negotiation

Requirements negotiation is an integral part of Requirements Engineering. The main aim of requirements negotiation is to increase customer satisfaction and product quality as much as possible. The evidence shows that most of the software projects fail due to ill defined and poorly negotiated requirements (H. P. In, Olson, & Rodgers, 2002).

In 1995 Standish group chaos report; the three major reasons for challenging projects were lack of including *all* stakeholders, incomplete and volatile requirements and specifications (*The Standish group CHAOS report*, 1995). All these factors show that the major concern of software development is how to deal with the issues of requirements or requirements conflicts. Requirements negotiation is the most success-critical factor for software projects. The outcome of the negotiation enhances the system's value extensively (B. Boehm & Egyed, 1998). Tom De Marco (DeMarco, 1996) said, *"How requirements were negotiated is far more important than how the requirements were specified"*.

Thomas (K W Thomas, 1976) proposed few conflict handling strategies based on the conflict behavior model in organizational psychology. He said that stakeholders have two concerns: one is to satisfy their own needs and second is to satisfy the needs of others. The five dominant strategies are the following:

- *Competing (forcing)* shows *win-lose* situation in which one stakeholder wins while the other loses.
- *Accommodating (smoothing)* shows *lose-win* situation in which one stakeholder sacrifices for the interest of other.
- *Collaborating (problem-solving)* shows *win-win* situation in which all the stakeholders are satisfied.
- *Avoiding (withdrawing from)* shows *lose-lose* situation in which nobody wins.
- *Compromising (sharing)* shows a relationship in which both the parties look for a middle ground.

The discussion shows that requirements conflicts are inevitable and it is necessary to handle them effectively. Resolving these conflicts can surely increase the chances of software project success considerably. Software Requirements Negotiation (SRN) plays a significant role to solve these problems. Requirements negotiation encourages the inclusion of all stakeholders. In the result of requirements negotiation we get mutually satisfactory requirements which address the concerns of all stakeholders. Requirements conflict handling is the essence of requirements negotiation (Grünbacher & Seyff, 2005). SRN addresses most of the requirements issues and resolve them through its several tradeoff practices. Requirements negotiation practices are widely used techniques, frameworks, tools, processes, models and methods or methodologies. Requirements negotiation can be performed in different ways. There are several practices reported in the literature which are available for executing the negotiation process. The main objective of the underlying study is to identify requirements negotiation practices in the literature which are empirically evaluated with their strengths and limitations. We will explore the degree of requirements conflict handling of these practices as well.

2.2. Software Requirements Conflicts

Software requirements conflicts are inevitable in the field of software development. They possess a significant importance in the field of requirements engineering. Often requirements conflicts are avoided in software engineering. There is a little evidence that describes handling these conflicts effectively (Easterbrook, 1994).

There is no particular definition of requirements conflict which creates a kind of confusion as nobody knows what exactly conflict between requirements means? Most requirements negotiation techniques handle binary conflicts i.e. conflict between two requirements, but it is problematic because there can be conflict among three or more requirements (van Lamsweerde, Darimont, & Letier, 1998).

Easterbrook (Easterbrook, 1994) defined conflict from interference point of view:

"Conflict can be a disagreement among problem owners."

Another definition of requirements conflict is that "*requirements conflict is a disagreement among requirements which adversely affects the outcome*" (Bowen, et al., 1989). Requirements conflict may also arise "*when multiple stakeholders bring about an inconsistency i.e. a situation in which two parts of a requirements specification show mismatching behavior*" (Robinson & Volkov, 1999).

2.3. Classification of Software Requirements Conflicts

R.Q.1.1. What kinds of conflict exist in literature that is resolved by requirements negotiation in software projects?

There is a little work done on the classification of requirements conflicts. There is no proper and standard classification of these conflicts that exists in the literature. Several authors (Herrmann, Paech, & Plaza, 2006; Kassab, Constantinides, & Ormandjieva, 2005; Mussbacher, Whittle, & Amyot, 2009) proposed different types of conflicts and most of these conflicts overlap one another. Therefore, there is a need for classification of these conflicts and there is a great need for such kinds of work in future. We will identify and then classify only those requirements conflicts which will support the underlying research work rather than identifying and categorizing all kinds of requirements conflicts because this is not the objective of this thesis. Some of the significant conflict types are as follows:

2.3.1. Requirements Contradiction

Requirements contradict (Herrmann, et al., 2006) with each other when more than one requirement refers to the same requirements concept but require conflicting values. For example R1: A report Y needs to represent all patients' address while R2: Report Y should contain only Name of the patient and Postal code. Now these are contradicting requirements demanding for opposing values.

2.3.2. Requirements Terminology Conflicts

Terminology conflicts (Rashid, et al., 2007) occur when different terminologies do not match with concepts. For example, a terminology may have different meanings in various domains.

2.3.3. Quality Attributes Conflicts

Quality attributes conflicts are (Al-Naeem, Gorton, Babar, Rabhi, & Benatallah, 2005; Chitchyan, Sampaio, Rashid, & Rayson, 2006; Kim, Park, Sugumaran, & Yang, 2007; Liu, 2009; Ramanna, Skowron, & Peters, 2007; Sadana & Liu, 2007) mostly potential conflicts which exist between the quality requirements of software system. For example, the conflict may exist between configurability and performance of the system. Similarly, security and performance may also conflict with each other.

2.3.4. Viewpoint Conflicts

Viewpoint conflicts (Herrmann, et al., 2006; Nhlabatsi, Laney, & Nuseibeh, 2008) occur when stakeholders possess conflicting positions/perspectives/concerns for particular requirements. The stakeholders may conflict (Chitchyan, et al., 2006; Kassab, et al., 2005) with each other for the implementation of a set of requirements. For example, highly experienced personnel humiliate less experienced people.

2.3.5. Resource Conflicts

In business requirements resource conflicts (Herrmann, et al., 2006; Mussbacher, et al., 2009) arise when multiple users read and alter the same resource simultaneously. When more than one users access the same information, then alteration should not be done by multiple users simultaneously during transactions; f*or example,* use of a limited telephone service by multiple functions.

2.3.6. Feasibility Conflicts

Sometimes, requirements do not interfere in the requirements phase but it has been observed that such non-interfering requirements are hard to realize in the architectural design. Therefore, such kinds of conflicts require architectural knowledge for resolving them. These conflicts are *feasibility conflicts* (Herrmann, et al., 2006).

CHAPTER 3

SYSTEMATIC REVIEW PROCESS

3. Systematic Review Process

This chapter introduces the relationship of systematic reviews and Evidence-Based Software Engineering. Then a roadmap for executing the systematic review is presented. This chapter also explains the activities of planning stage of systematic review.

3.1. Evidence Based Software Engineering

Software engineers make decisions for the adoption of some new methodology or technology. They need sufficient empirical evidence for adapting a specific technology so that they can take maximum advantage of that technology. They face problems in such a decision making due to the availability of little empirical evidence of that particular method or technique in the literature. This shows that the research which is carried out, does not meet the challenges of industry. Therefore, we need such kind of methods which can close this gap between research and practice. To avoid this problem, *Evidence-Based Software Engineering* (EBSE) was proposed. The purpose of EBSE is to improve decision making in software development by integrating the best empirical evidence currently available in the literature in the relevant area of interest. Such kind of knowledge can be used by the practitioners to adapt immature technologies with more confidence. The discipline of EBSE was developed in ICSE'04 by B.A. Kitchenham. EBSE is based on the *Evidence-Based Medicine* (EBM) which focuses on the integration of practical evidence in clinical research. The significance of EBSE is obvious today as software-intensive systems are used everywhere in our daily life; for example in cars, stereos, microwave, electric trains, London ambulance service etc. Therefore, EBSE also improves dependability and acceptance of software systems. Evidence-based method is also used by some other disciplines like education, social policy and psychiatry (Dyba, Kitchenham, & Jorgensen, 2005; B. A. Kitchenham, Dyba, & Jorgensen, 2004).

3.2. The Systematic Review Process

There is a little work done in software engineering for developing guidelines to conduct a systematic review. There are no approved standards for performing systematic reviews (B. A. Kitchenham, et al., 2004). Only a few SLR guidelines depend upon the practical evidence. In this regard, B.A. Kitchenham and J. Biolchini formulated guidelines for performing systematic reviews in software engineering.

Both kinds of guidelines are followed by the researchers. We will use the guidelines presented by Kitchenham and Charters (B. Kitchenham & Charters, 2007). According to these guidelines, the systematic review process consists of three main phases which are then further refined into small steps. The systematic review process phases and activities which are suitable for our review are listed below (B. Kitchenham & Charters, 2007):

Phase 1: Planning the Review

The activities related to *planning the review* are:

- Identifying the need for a review
- Specifying the research question
- Developing a review protocol
- Evaluating the review protocol

Phase 2: Conducting the Review

The activities relevant to *conducting the review* are:

- Identification of research
- Selection of primary studies
- Study quality assessment
- Data extraction and monitoring
- Data synthesis

Phase 3: Reporting the Review

The activities of *reporting the review* are:

- Specifying dissemination mechanisms
- Formatting the main report
- Evaluating the report

All the activities are mandatory except the two stages i.e. evaluating the review protocol and evaluating the report is optional and they are totally dependent on the review team. Apparently the activities of SLR are sequential but in reality several activities are iterative in nature and some of them may be refined later on (B. Kitchenham & Charters, 2007).

3.3. Planning the Review

The significant activities included in the planning stage are explained in the coming sections.

3.3.1. Identification of the Need for a Systematic Review

The need for systematic review is a significant activity of planning stage. In this activity, we analyzed whether there was a need for systematic review or not? We identified whether sufficient evidence would be available in the literature or not. We also identified whether any systematic review exists on requirements negotiation in the literature prior to our work or not. For this purpose, *review mapping* technique was used to identify the existing literature (B. Kitchenham & Charters, 2007).

To answer all these questions, we identified that sufficient *primary studies* were available in the field of software requirements negotiation. Secondly, we also did review mapping for determining the existence of such a kind of research work in the area of requirements negotiation. Fortunately, no systematic review was found in the area of requirements negotiation. On the basis of this information, we decided that there was a need for a systematic review in requirements negotiation.

3.3.2. Developing a Review Protocol

The review protocol describes the methods which will be used to execute a systematic review. The review protocol helps to avoid researcher biasness. For instance, in the absence of a protocol it may be possible that study selection or data analysis is driven by the researcher's own expectations (B. Kitchenham & Charters, 2007).

The review protocol addresses the review questions and predefined search strategy including sources of search and search terms. It also describes the criteria for assessing the quality of candidate studies. It includes the data extraction forms and data synthesis methods as well. This whole information helps the other researchers to replicate and validate the results of the study (Khan, 2006).

This section will introduce the necessary elements of the real review protocol (Appendix A) which was developed in the planning phase of systematic review. The contents of our review protocol were as under:

3.3.2.1. Background

Background provides the rationale behind the systematic review. We would like to present information like why did we need it. Some information related to prior work in our research proposal is also included. The detail can be seen in (Appendix A).

3.3.2.2. Review Questions

The review question is the vital part of a systematic review. This section describes the main focus of the systematic review. The review question was designed on the basis of guidelines as prescribed by Kitchenham and Charters (B. Kitchenham & Charters, 2007). The question structure was formulated on the basis of PICOC i.e. Population, Intervention, Comparison, Outcome and Context. In this SLR we identified the most commonly used software requirements negotiation practices with their benefits and limitations. The detail can be seen in (Appendix A).

3.3.2.3. Identification of Research

The search strategy was designed on the basis of question structure (PICOC) and study design. The review emphasized on the requirements negotiation practices with their benefits and limitations in the field of requirements engineering. The detail search strategy can be seen in (Appendix A).

The *search terms* had been derived from the review questions considering PICOC so that we could retrieve maximum accessible literature from the most popular relevant electronic databases. Boolean operator like 'AND' or 'OR' were used for joining multiple strings. The search terms were designed using guidelines of Kitchenham and Charters (B. Kitchenham & Charters, 2007).

Search sources had been selected on the basis of quality, relevance and reliability of the evidence. We selected only high quality databases so that the review should be based on the best available and accessible evidence to ensure the reliability of review results.

3.3.2.4. Study Selection Criteria and Procedures

Study selection criterion was designed to select *primary* studies for the execution of systematic review. Study selection criterion was based on the review question and question structure i.e. PICOC and study design. We developed study inclusion and exclusion criteria considering review questions so that only those studies could be selected which could provide evidence about requirements negotiation practices.

3.3.2.5. Study Quality Assessment Checklists and Procedures

The quality assessment criteria (QAC) were designed to ensure the high quality of primary studies. A checklist (Appendix A) was formulated for assessing the quality of the selected studies. The checklist had two steps. In first step, general questions were asked and if these questions were answered positively, then we would proceed for the next step and ask more questions otherwise study was rejected. The main emphasis of checklist was on identifying empirical evidence.

3.3.2.6. Data Extraction Strategy

The data extraction form (Appendix A) was developed (to avoid bias) for extracting data from these finally selected studies. All the studies were read thoroughly and then evidence was collected from each article. The form focused on the information like strength of research method, study settings, validity of claims, etc. The data extraction form was filled for all the selected studies and this data became a basis for data synthesis and analysis.

3.3.2.7. Data Synthesis

The data synthesis integrated and summarized the data extracted in the previous step. The qualitative data synthesis techniques were used for collating information as the primary studies were heterogeneous. We developed data synthesis forms (Appendix A) for summarizing the data. One of the data synthesis form listed the most widely used requirements negotiation practices. Second form collected benefits of each practice from different articles and provided a summary of all benefits. Third form collected limitations of practices reported in the articles. This information provided a strong basis for analyzing the given data and drawing conclusions.

3.3.2.8. Dissemination Strategy

The final report was written in the *Thesis format* which was a partial requirement of International Islamic University, Islamabad. This report was viewed by both internal

SIG committee as well as external review committee. The detail can be seen in (Appendix A).

3.3.2.9. *Project Time Table*

The project time table described the timeline of the research project with a date of completion. The detail of activities and their deadlines can be seen in (Appendix A).

3.3.3. Evaluating the Review Protocol

The review protocol was developed using the guidelines of Kitchenham and Charters (B. Kitchenham & Charters, 2007). The protocol was reviewed by the thesis supervisor who is a member of review committee of IEEE Computer Society. The protocol was also verified through pilot study to ensure whether it was executable or not. The results obtained through pilot study were satisfactory and they showed that the protocol could be implemented and could give good results. Then the protocol was reviewed by Dr. Mahmood Niazi who is Professor at Keele University in UK. The protocol had been modified according to the feedback given by these reviewers which also ensured the reliability of the review protocol.

3.3.4. Threats to the Validity of Review Protocol

The review protocol was developed by a single person. Therefore, there can be a chance of bias. No doubt the review protocol had been reviewed by the independent experts but even then, the review protocol was not published and opened for public use which also increased threat to its validity. The qualitative data synthesis approach was used rather that quantitative methods because it was difficult to use both the techniques in combination.

CHAPTER 4

CONDUCTING THE REVIEW

4. Conducting the Review

This chapter focuses on the execution of systematic review. The execution is the significant and major phase of systematic review. The execution phase starts after the verification and acceptance of review protocol. The execution will follow the whole plan as it was designed in the review protocol.

4.1. Identification of Research

The purpose of this stage of systematic literature review is to identify and collect primary studies for executing SLR. This stage is the most complex and trickiest stage which needs great expertise and consultation. This activity is more time consuming and challenging than other activities because we have to develop a query which can extract the whole relevant data available in the electronic databases. We also need resources to get studies from both published and unpublished resources (Khan, 2006).

4.1.1. Generating a Search Strategy

The *search strategy* was developed to include maximum available literature in the relevant field. The search strategy should be unbiased. The author developed a search strategy using the guidelines of Kitchenham and Charters (B. Kitchenham & Charters, 2007) after consulting with the supervisor and it included search terms and search sources for collecting data.

4.1.1.1. Source Selection

The search sources were selected on the basis of availability of the relevant data. The most popular electronic databases were included to extract the best available literature in the field of software engineering. Initially Google scholar and cite-seer were also included but later on, they were removed from the list after consulting with the supervisor. The reason for elimination was that same studies were available on the other selected databases as well. Finally selected search sources were IEEE, Springer Verlag, ACM, EI Compendex and Science Direct (Appendix A).

The reference lists of selected primary studies were also reviewed and few more studies were found and included later on after consulting with the supervisor. The author also looked for conferences and journals which were not included in the selected databases but no further evidence was found. Some experts were also contacted for getting some studies which were not accessible. A research register maintained by A. Davis was also searched.

4.1.1.2. Search Terms

The search strategy was developed on the basis of review question. The search terms (Appendix A) were derived from the main words of the question like requirements negotiation practices. This stage was iterative in nature and queries were refined many times. Then they were applied iteratively on all selected databases. After several weeks the final query was developed which provided more satisfactory results and it was then selected for the review protocol.

Table 1: Search Strings for Each Electronic Database

Database	**Search String**
IEEE	(“abstract”:”Requirements negotiation” OR “conflict resolution” OR ”conflict handling” OR ”requirements reconciliation”) **AND** (“abstract”:”technique” OR “process” OR “model” OR “method” OR “framework” OR “approach” OR “tool”)
EICOMPENDEX	((“Requirements negotiation” OR “conflict resolution” OR “conflict handling” OR “requirements reconciliation”) WN **AB) AND** ((”technique” OR “process” OR “model” OR “method” OR “framework” OR “approach” OR “tool”) WN **AB)**
SCIENCE DIRECT	Abstract("Requirements negotiation" OR "conflict resolution" OR "conflict handling" OR "requirements reconciliation") **AND** Abstract(”technique” OR “process” OR “model” OR “method” OR “framework” OR “approach” OR “tool”)
SPRINGER-1.1	su:(“Requirements negotiation” OR “conflict resolution” OR “conflict handling” OR “requirements reconciliation”) **AND** su:(“technique” OR “process”)
SPRINGER-1.2	su:(“Requirements negotiation” OR “conflict resolution” OR “conflict handling” OR “requirements reconciliation”) **AND** su:(“model” OR “method”)
SPRINGER-1.3	su:(“Requirements negotiation” OR “conflict resolution” OR “conflict handling” OR “requirements reconciliation”) **AND** su:(“approach” OR “framework”)
SPRINGER-1.4	su:(“Requirements negotiation” OR “conflict resolution” OR “conflict handling” OR “requirements reconciliation”) **AND** su:(“tool”)
ACM	(Abstract:“Requirements negotiation” OR Abstract:“conflict resolution” OR Abstract:”conflict handling” OR Abstract:“requirements reconciliation”) AND (Abstract:”technique” OR Abstract:“process” OR Abstract:”model” OR Abstract:“method” OR Abstract:“approach” OR Abstract:“framework” OR Abstract:“tool”)

Then a separate search query, as shown in table 1, was formulated for each electronic database to get maximum available empirical evidence from 1968 to 2010. The initial time (1968) was chosen because the field of software engineering was developed in

1968. The search strategy date was also recorded so that when the query was rerun then new studies could be identified. The search strategies were applied rigorously on each database with an interval of few days. Even then same results were obtained and no new study was obtained.

4.1.1.3. Publication Bias

The *grey literature* (unpublished studies) was not searched because printed journals were not available therefore only data of electronic databases was included. This can also be a threat to the validity of the review protocol. It might be possible that all studies would not be included but this threat has been minimized through rigorous search of electronic databases. The review protocol was also externally reviewed by a professor of Keele University in UK which also reduced the threat to the validity of systematic review.

4.1.1.4. Bibliography Management and Document Retrieval

A bibliographic tool i.e. *Endnote* had been used which helped to manage the references in an elegant manner. It had been observed that performing SLR with the help of automated tool was more efficient and easy to manage. The tool reduced a lot of time and effort. The tool was very flexible in nature and it helped the author to make notes about articles which made it very convenient to handle references. The studies were retrieved from the databases and inserted in the program which had been managed automatically. The tool also made the review more reliable because manual work could be error prone (Khan, 2006).

4.1.1.5. Documenting the Search

The electronic databases were searched on the basis of both *abstract* and *full text*. It had been observed that *full text* results as shown in the table were more relevant and realistic. Therefore, *full text* results were selected for the execution of the review. The search results shown in table 3 were finalized on *16-Jun-2008*. The search process was also documented in the review protocol (Appendix A). The search documentation for each database was also stored as shown in table 2.

Table 2: Search Documentation for Each Database

Data source	**Documentation**
Digital library	Name of database: **IEEEXPLORE** Search strategy for the database: • Only English papers • Search criteria is *abstract* only • A search query is executed Date of search: 02-06-10 Years covered by search: 1968-2010
Digital library	Name of database: **EICOMPENDEX** Search strategy for the database: • Only English papers • Search criteria is *abstract* only • A search query is executed Date of search: 02-06-10 Years covered by search: 1968-2010
Digital library	Name of database: **ACM** Search strategy for the database: • Only English papers • Search criteria is *abstract* only • ACM is limited to only ten terms, so main query is divided into four separate sub-queries. Date of search: 02-06-10 Years covered by search: 1968-2010
Digital library	Name of database: **SPRINGERLINK** Search strategy for the database: • Only English papers • Search criteria is *abstract* only • SPRINGERLINK is limited to only ten terms, so main query is divided into four sub-queries. Date of search: 02-06-10 Years covered by search: 1968-2010
Digital library	Name of database: **SCIENCE DIRECT** Search strategy for the database: • Only English papers • Search criteria is *abstract* only • Papers of only subject "Computer Science" Date of search: 02-06-10 Years covered by search: 1968-2010

Table 3: Integrated Search Results

Basis of selection	No. of Papers found in each database					
	IEEE	ACM	EICOMPENDEX	SPRINGER VERLAG	SCIENCE DIRECT	Total
Abstract	550	132	952	638	123	2395
Full text	283	57	50	517	41	948

4.2. Study Selection

The studies which had been extracted in the previous step were now tested for their relevance and then filtered. Only selected studies had been used for SLR.

4.2.1. Study Selection Criteria

The study selection criteria were designed to filter the evidence from the available literature which was relevant to the review question. For this purpose, we had been designed a study inclusion and exclusion criteria (Appendix A) in the review protocol to avoid researcher's bias. The inclusion and exclusion criteria were also discussed with the thesis supervisor and it was finalized with the mutual agreement. Later on, it was also observed by the external reviewer. The criteria were also piloted on few studies and it had been observed that relevant studies were included and irrelevant studies were excluded which also confirmed the validity of criteria.

The study selection criteria were formulated on the basis of research question which had been answered in the systematic review. We included only those studies which fully passed the inclusion criteria and if the study did not fulfill the selection criteria, then it was excluded.

4.2.2. Reliability of Inclusion Decisions

The studies inclusion decision was made by a single researcher which could be biased. To avoid this problem, the studies included after applying inclusion criteria were discussed with the thesis supervisor. A rationale for the excluded studies was maintained and discussed with the thesis supervisor.

4.2.3. Study Selection Process

The study selection process was applied on all studies. In the first round, duplicate studies were identified by using Endnote- a bibliographic tool. It had been observed that more than half of the studies i.e. 494/948 were duplicate studies and they were excluded as shown in table 4.

Table 4: Study Selection Results

Study inclusion/exclusion criterion	Query results	Studies left
Total studies found after searching electronic databases	948	948
Duplicate studies excluded	494	454
Irrelevant studies excluded after reading titles and abstracts	314	140
Simple negotiation studies excluded after reading titles, abstracts, introduction and conclusion	51	89
Studies excluded after consultation with the supervisor due to irrelevance or duplication with different titles	10	79

In the second round, abstracts and titles of 454 papers were searched for the relevance and it was found that 314/454 studies were irrelevant and belonged to other domains. In the third round, the studies of simple negotiation relevant to the area of social sciences were excluded after reading titles, abstracts, introduction and conclusion which were 51/140. Now 89 studies were left which were studied in detail and then discussed with the supervisor. As a result, 10 more studies were excluded as they were irrelevant. Now we finally selected 79 studies for the next step of execution.

4.3. Study Quality Assessment

The study quality assessment was a significant activity of the systematic review to filter out primary studies for SLR. The resulting primary studies provided the evidence of SLR. The quality assessment criterion was detailed in nature while the study selection criterion was an abstract form of assessing quality of studies.

4.3.1. The Hierarchy of Evidence

In our systematic review we had selected experiments, case studies, field studies and industrial experience reports. The evidence based on the experiments and case studies was more credible. Then field studies and experience reports were selected because they were very strong form of evidence. The expert opinion was not included as it was a weak form of evidence and there was sufficient evidence available from other methods therefore it was rejected. This also increased the reliability of the systematic review.

4.3.2. Developing Quality Instrument

To measure the quality of a study, we developed a quality instrument. The study quality assessment criteria (QAC) checklist (Appendix A) was designed to determine the quality of each study. The checklist was designed during the development of review protocol (Appendix A). To avoid the bias, the checklist was designed with the consultation of thesis supervisor and some other experts. The author requested some sample checklists from evidence based software engineering (EBSE) digital group members and they sent him few papers of B.A. Kitchenham. The author also took help from the latest checklists designed by B.A. Kitchenham in her systematic reviews. The checklist was finalized after consulting with the advisor and other domain experts.

4.3.3. Using the Quality Instrument

QAC consisted of nine different questions out of which 3 questions are general and the other 6 are detailed questions focusing on the study design of each study. Each question had three possible answers like Yes/No/Partial with values 1, 0 and 0.5 respectively. The minimum *quality threshold* for each study was 1.5. The reason for setting this threshold was that the answer of the three questions can be 0.5. If the answer of any of first three questions will be *NO* then the study will be rejected and quality assessment process will stop. On the other hand, if the answer of all first three questions will be *Yes/Partial* then quality assessment process will continue to score other questions. Therefore, the appropriate threshold for the studies was 1.5. The detailed view can be seen in (Appendix C). The minimum quality threshold determined:

- Did the study provide empirical evidence or not?
- Did the study resolve any requirements conflict or not?
- Did it present any requirements negotiation practice or not?

If a study passed this initial criterion then it was tested for the next detailed criteria. Otherwise, the study was rejected at this point and not further tested. The primary studies were finalized and extracted after consulting with the thesis supervisor.

4.4. Data Extraction

The data extraction stage of systematic review was responsible for collecting and accumulating the information from the selected primary studies which could properly answer the review questions.

4.4.1. Data Extraction Form

The data extraction forms were designed during developing the review protocol (Appendix A) to avoid researcher's bias. The data extraction form was designed in such a way so that the form could extract all necessary information which was intended for the review question. The form includes both generic and specific questions. The generic questions were like date of extraction, title of study, year of publication etc. The specific questions were relevant to the topic of interest i.e. requirements negotiation practices. The specific questions focused on the detailed description of the findings of the studies and identification of requirements negotiation practices. A pilot study was also performed after developing a review protocol to determine the reliability of the data extraction form. After analyzing the pilot results, the author did few changes in the form which were finalized after discussing with the thesis supervisor. The data extraction form was also checked by an external domain expert.

4.4.2. Data Extraction Procedure

It was recommended that the data should be extracted by the two independent researchers but in this SLR there was a single person (thesis author) who extracted data from the primary studies. This could be a threat to the validity of this systematic review. However, the data extracted from the studies was consulted with the thesis supervisor to minimize the threat to the validity of SLR. The thesis supervisor performed data extraction on a randomized sample of primary studies and evaluated the results which also reduced the threat to the validity of SLR. The disagreements and uncertainties among the results of some primary studies were discussed and resolved after consulting with the thesis supervisor.

4.4.3. Multiple Publications of Same Studies

No doubt the duplicate studies were eliminated during study selection stage of SLR but still there were duplicate studies with different titles and publishers. This problem had been resolved after consulting with the thesis supervisor. We included all those articles referring to the same study but in a single data extraction form and their findings were merged with the initial study with a separate distinction. There were few studies which were absolutely duplicated and published in some other conference then such kind of studies were rejected at this stage.

4.5. Data Synthesis

Data synthesis was responsible for recapitulating the whole information extracted from the primary studies. There were two ways to synthesize data i.e. qualitative/descriptive synthesis and quantitative synthesis. In the underlying SLR descriptive synthesis was used rather than quantitative. As the results were heterogeneous and inconsistent, therefore using both the techniques in combination was difficult. Therefore, descriptive synthesis was more suitable for such kind of results.

The data extracted from the primary studies had been organized in tabular form to answer the review questions. The results obtained from the primary studies were listed below:

RQ.1 What requirements negotiation practices are used to resolve requirements conflicts in software projects?

Answer: The list of requirements negotiation practices can be seen in table 5. The requirements negotiation practices were arranged on the basis of practice ids like P1, P2, etc rather than study ids which are in the form of numerals. Only 18 primary studies were selected for gathering the empirical evidence whereas 12 widely used requirements negotiation practices had been identified in these studies i.e. P1 to P12 as shown in table 5. Each practice resolved one or more kinds of requirements conflicts which will be described later on.

Table 5: List of Requirements Negotiation Practices, Corresponding to R.Q.1

Study id	Quality score	Database name	Year of publication	Practice id	List of practices
9	7.5	IEEE	1999	P1	Simplifiers and Complicators Approach
10	8.0	IEEE	1995	P2	Winwin Spiral Process Model
72	8.0	IEEE	1998		
77	8.0	J.C. Baltzer AG, Science Publishers	1998		
10	8.0	IEEE	1995	P3	Winwin System
25	8.5	IEEE	1998		
39	7.5	Elsevier	2007		
77	8.0	J.C. Baltzer AG, Science Publishers	1998		
20	8.5	IEEE	2008	P4	Wikiwinwin Tool
68	8.0	IEEE	2010		
69	8.0	IEEE	2009		
32	7.5	IEEE Computer Society	2000	P5	Easywinwin
34	8.5	IEEE	2001	P6	CBSP Approach
43	8.5	IJSEKE	2006	P7	ICRAD
54	8.5	IEEE	2007	P8	Hybrid Method
56	7.0	Springer-Verlag	2009	P9	FTR Tool
58	9.0	Springer-Verlag	2005	P10	MEG Groupware
62	8.0	IEEE Computer Society	2005	P11	ARENA-M
73	9.0	IEEE	2001	P12	QARCC
76	8.5	IEEE Software	1996		

RQ.1.2 What are the reported strengths of existing requirements negotiation practices?

Answer: The strengths of existing requirements negotiation practices were described in the table 6. It had been observed that few studies mentioned the advantages of practices clearly whereas some studies had been poorly reported and they didn't mention or explain the benefits of practices. In case of multiple same studies, we had merged the benefits under one heading of practice. Therefore, somewhere you might see results presented from more than one studies.

Table 6: Benefits of Practices, Corresponding to R.Q. 1.2

Practice ID	Benefit ID	Benefits of Practices	Study IDs that cover this benefit	Total no. of study IDs
P1		**Simplifiers and complicators approach:**	9	1
	B1.1	Decreases the problems of unrealistic expectation and win conditions.	9	1
	B1.2	Far less software projects reported the feasibility problems of LCO milestones.	9	1
	B1.3	Can be implemented successfully in other areas as well.	9	1
	B1.4	Resolves "Two Cultures" problem significantly.	9	1
P2		**Winwin Spiral Process:**	10,72	2
	-------	***The benefits of the process reported in the first study are the following:***	--------	-------
	B2.1	It supports involvement of stakeholders in early stages of software life cycle.	10	1
	B2.2	It eliminates *sequential* involvement of multiple stakeholders.	10	1
	B2.3	It supports model-based negotiation and *renegotiation.*	10	1
	B2.4	It manages requirements updates by supporting the facility of renegotiation.	10	1
	-------	***The benefits of the process reported in the second study are the following:***	-------	--------
	-------	*The results obtained in the first year projects are:*	-------	-------
	B2.5	It is highly risk-driven approach which enables to maintain the project control.	72	1
	B2.6	The model is flexible and deals with project risks and uncertainties efficiently.	72	1
	B2.7	The model is highly formalized and emphasizes on the achievement of three main artifacts i.e. LCO (Life Cycle Objective), LCA (Life Cycle Architecture) and IOC (Initial Operational Capability).	72	1
	B2.8	It improves the trust among stakeholders which encourages strong cooperation among them.	72	1

	B2.9	Executing the prototype and negotiation in parallel is more beneficial rather than executing them in a sequence.	72	1
	B2.10	Theory W and Winwin helps to broaden the vision of participants.	72	1
	B2.11	It improved the learning of stakeholders.	72	1
	B2.12	The model can be tailored easily with respect to the project scope.	72	1
	B2.13	The winwin system encourages smooth transformation of agreed win conditions into specification.	72	1
	--------	*The results obtained in the second year projects are:*	---------	---------
	B2.14	The quantity of documentation is decreased.	72	1
	B2.15	There is an improvement in managing the complexity of software architecture.	72	1
	B2.16	It is observed that productivity and quality of the products increased.	72	1
	B2.17	Winwin spiral model is a good practical approach.	72	1
P3		**Winwin system:**	25,39,77	3
	-------	***The benefits of winwin system reported in the first and second study are the following:***	--------	-------
	B3.1	Provides a strong support for collaboration and Winwin negotiation.	25,39	2
	B3.2	The users are satisfied with the tool support.	25,77	2
	B3.3	The tool emphasizes on major issues.	25,39,77	3
	B3.4	It helps to mitigate the friction and creates a balance between loud and quite users.	25,39,77	3
	B3.5	It supports distributed collaboration as well.	25,39,77	3
	B3.6	The quality of requirements was improved.	25,77	1
	B3.7	It improves the learning of stakeholders in requirements process.	25	1
	B3.8	It supports the requirements evolution.	25	1
	-------	***The benefits of the tool reported in the third study are the following:***	--------	---------
	B3.9	Creates a strong cooperative and mutually understanding environment.	77	1
	B3.10	The customer satisfaction was increased.	77	1
	B3.11	It improves the trust and confidence among stakeholders.	77	1

	B3.12	The stakeholders' vision became wide enough due to Theory W and winwin.	77	1
P4		**WikiWinWin:**	20,68,69	3
	--------	***The benefits of the tool reported in the first study are listed below:***	--------	-------
	B4.1	It can effectively support the WinWin requirements negotiation approach.	20	1
	B4.2	It provides an easy and simple way to share vision and knowledge.	20	1
	B4.3	It provides a strong support for flexible way of requirements updating and maintaining the history of changes.	20	1
	B4.4	The tool may be enhanced easily to include other project dimensions.	20	1
	B4.5	It provides boundary objects which improves the learning and cooperation of stakeholders.	20,68	2
	------	***The benefits of the tool reported in the second study are the following:***	-------	--------
	B4.6	It encourages participating *all* stakeholders.	68	1
	B4.7	It helps to elicit customer and user needs.	68	1
	B4.8	It increases the quality of the requirements negotiation process.	68	1
	B4.9	It helps to prioritize requirements.	68	1
	B4.10	It reduces manual work.	68	1
	B4.11	It improves traceability of decisions by maintaining historical record.	68	1
	B4.12	It helps to elicit both minor and critical requirements.	68	1
	B4.13	It is more cost-effective tool.	68	1
	-------	***The benefits of the tool reported in the third study are the following:***	------	--------
	B4.14	The tool was accessible without any difficulty and time limitation on the web.	68,69	2
	B4.15	The tool is more robust and reliable.	68,69	2
	B4.16	There is a better management of artifacts using wikiwinwin.	69	1
	B4.17	The vocabulary is maintained which is beneficial for the users.	68,69	2
	B4.18	The tool notifies automatically that is considered useful.	69	1
	B4.19	The taxonomy of requirements reconciliation can be useful.	69	1

P5		**Easy winwin:**	32,39,54	3
	------	***The benefits of Easywinwin reported in the first study are listed below:***	-------	--------
	B5.1	It is easy to use and "light weight" negotiation approach.	32,39,54	3
	B5.2	Increase in number of stakeholders involved.	32,54	2
	B5.3	Stakeholders' guidance was improved.	32	1
	B5.4	Better analysis of negotiation outcome and integration of groupware with Rational Rose CASE tool.	32	1
	B5.5	It deals with various negotiation scenarios.	32	1
	B5.6	It improves in prioritizing requirements and identifying more issues.	32	1
	B5.7	It is less time consuming.	32	1
	B5.8	Supports early identification and mitigation of risks.	32	1
	B5.9	It generates more artifacts than winwin system due to increase in stakeholder involvement.	32	1
	B5.10	It is a more practical approach.	32	1
	-----	***The benefits of Easywinwin reported in the second study are listed below:***	---------	------
	B5.11	It provides a broad view and helps to create a shared vision among stakeholders.	39,54	2
	B5.12	It supports in learning other techniques as well.	39	1
	B5.13	Creates distinct patterns for group communication.	39	1
	B5.14	Easy winwin in collaboration with RESCUE process provides good results.	39	1
	------	***The benefits of Easywinwin reported in the third study are listed below:***	--------	--------
	B5.15	Better understanding of problem.	54	1
	B5.16	It helps to know what other stakeholders need.	54	1
	B5.17	It helps to elicit tacit knowledge.	54	1
	B5.18	It can be utilized at any stage in the software life cycle.	54	1
	B5.19	It can be used concurrently when building initial prototype.	54	1
	B5.20	It helps in planning the requirements for delivering the system.	54	1
	B5.21	It helps in extending COTS products.	54	1

	B5.22	Develops high level requirements representations.	54	1
	B5.23	Maintains the decision rationale to avoid wrong decision making.	54	1
	B5.24	Improves learning of stakeholder desires.	54	1
	B5.25	It handles a wide range of requirements.	54	1
	B5.26	It provides a sharp and effective mean to convert informal decisions into formal decisions.	54	1
P6		**CBSP Approach:**	34	1
	B6.1	It resolves the problem of capturing requirements informally and specifying in the formal language in the architecture.	34	1
	B6.2	It supports for mapping non-functional requirements into architectural models.	34	1
	B6.3	It provides a better way to address the problem of requirements evolution.	34	1
	B6.4	It does not assure the completeness of requirements but it supports complex relationships between requirements and architecture. It simplifies iterations between them.	34	1
	B6.5	It handles the problem of large scale project requirements.	34	1
	B6.6	It supports the involvement of diverse stakeholders with different backgrounds.	34	1
P7		**ICRAD:**	43	1
	B7.1	The process resolves conflicts of both *requirements space* and *solution space*. It resolves feasibility and contradicting requirements conflicts in *solution space*. It resolves inconsistent requirements conflicts in *requirements space*.	43	1
	B7.2	It stores the negotiation rationale.	43	1
	B7.3	It allows comparison of different options.	43	1
	B7.4	It helps in improving the solutions.	43	1
	B7.5	It helps in decreasing the complexity and managing the requirements easily.	43	1
P8		**Hybrid method:**	54	1
	B8.1	It improves RE decision making during the process of requirements negotiation.	54	1
	B8.2	It increases the quality of negotiation artifacts	54	1

	B8.3	It helps to discover more issues in the requirements document.	54	1
	B8.4	It improves the traceability of the requirements.	54	1
	B8.5	It makes easy to develop formal requirements specification.	54	1
P9		**FTR tool:**	56	1
	B9.1	It highly supports and promotes negotiation-collaboration process better as compared to the typical method of FTR.	56	1
	B9.2	It speeds up the process of FTR as the reviewers rejected the functionality where there was no argument.	56	1
	B9.3	It motivates the argumentation and collaboration with an increase in no. of negotiated outcome.	56	1
	B9.4	The tool is easy to learn and use.	56	1
	B9.5	There is a little training required for using the tool effectively.	56	1
	B9.6	Rules and regulations are clearly defined.	56	1
	B9.7	The tool supports value-neutral knowledge management.	56	1
	B9.8	The tool maintains the history of arguments.	56	1
	B9.9	Supports both synchronous and asynchronous way of communication among stakeholders but face-to-face meeting is preferred to explore more options.	56	1
	B9.10	It provides a better way to register and manage arguments.	56	1
	B9.11	It is easy to update positions towards the agreements.	56	1
	B9.12	It avoids unnecessary iterations of the process without the user's desire.	56	1
P10		**MEG groupware:**	58	1
	B10.1	The tool helps to reconcile and find positions of stakeholders easily and quickly.	58	1
	B10.2	The tool supports very efficient way to reach the consensus.	58	1
	B10.3	The tool resolves both high and low level conflicts as it supports both NSS and GSS.	58	1
	B10.4	It increases the quality of SQFD.	58	1
	B10.5	It encourages adapting integrative strategy (win-win) rather than distributive (win-lose).	58	1

	B10.6	It avoids win-lose and lose-lose situations.	58	1
	B10.7	It is a good approach for reaching to an agreement.	58	1
	B10.8	It also encourages argumentation.	58	1
P11		**ARENA-M:**	62	1
	B11.1	It highly supports context-aware requirements negotiation.	62	1
	B11.2	It is equally effective like ARENA II.	62	1
	B11.3	Experienced users used the mobile devices effectively.	62	1
	B11.4	It is a user friendly groupware.	62	1
	B11.5	It can be used any where at any time.	62	1
	B11.6	It prevents to hinder stakeholders' participation.	62	1
P12		**QARCC:**	73,76	2
	B12.1	***The benefits of QARCC reported in the first study are as under:***		1
	B12.2	QARCC and S-COST identified a great number of quality requirements conflicts and alternatives than manual approach like winwin.	73,76	2
	B12.3	It is an economical approach which helps in identifying conflicts by using a predefined checklist.	73,76	2
	B12.4	The tool is beneficial for both trained and untrained users.	73	1
	B12.5	Such an approach elaborate the conflicts and alternatives very well which are identified by manual approaches.	73	1
	------	***The benefits of QARCC reported in the second study are as under:***	---------	--------
	B12.6	The semi automated approach provides a better way to in addressing quality issues and creates equilibrium among human expertise.	76	1
	B12.7	It is not a domain specific approach but can be customized for any domain.	76	1

RQ.1.3 What are the reported weaknesses of existing requirements negotiation practices?

Answer: The weaknesses of existing requirements negotiation practices were listed in the table 7. A shortcoming had been observed in several studies that the most of the studies did not report the limitations of study at all which was very surprising. That is why; the limitations of few studies were missing because they were not reported in the studies. This fact created problems while collecting the weaknesses of the studies.

Table 7: Weaknesses of Practices, Corresponding to R.Q. 1.3

Practice ID	Weakness ID	Weaknesses of Practices	Study IDs that cover this weakness	Total no. of study IDs
P2		**Winwin spiral process:**	10,72	2
	W2.1	Winwin spiral process is unable to assess major disagreements between requirements.	10	1
P3		**Winwin system:**	25,39,72,77	3
	W3.1	It needs more training and preparation before negotiation.	25,72,77	3
	W3.2	The extensive winwin overhead consumes more time to negotiate.	25	1
	W3.3	The frequency of errors reduces negotiability.	25,77	2
	W3.4	Performing negotiation and prototyping separately is problematic.	25,77	2
	W3.5	There should be a direct communication with the system users.	25,77	2
	W3.6	Some more features should be introduced to facilitate winwin like email and video conferencing.	25,77	2
	W3.7	To reach winwin equilibrium is more time consuming.	25	1
	W3.8	UNIX environment has its own limitations for using the tool.	25	1
	W3.9	There should be an easy way to discuss issues.	25	1
	W3.10	There is a problem of interpretation of objective artifacts.	25	1
	-------	***The limitations of winwin system found in another study are listed below:***	--------	
	W3.11	Some students criticized that these are formal tools therefore the students avoid using them while others said that these tools helped them positively.	39	1
P4		**WikiWinWin:**	20,68,69	3
	W4.1	The tool is not fully automated still some tasks need automation.	20,69	2
	W4.2	It consumes plenty of time in training of users.	20	1
	W4.3	It needs improvement in topic editing during synchronous meetings.	20	1
	-------	***The limitations found in the second study are listed below:***	-------	
	W4.4	The tool is not user friendly.	68,69	2

	W4.5	Only trained users can produce good results.	68,69	2
	W4.6	It does not support individual use very well.	68	2
	------	**The limitations found in the third study are listed below:**	---------	
	W4.7	Users found problems in the email settings for instance unable to receive password through mailbox.	69	1
	W4.8	It cannot give summary of significant issues and requirements.	69	1
P6		**CBSP approach:**	34	1
	W6.1	· The CBSP approach needs massive human interaction.	34	1
	W6.2	· The approach does not guarantee the conversion of CBSP artifacts into architecture.	34	1
P9		**FTR tool:**	56	1
	W9.1	The increase in no. of results can be due to the diversity of team members rather than only FTR tool support.	56	1
	W9.2	The user can enter vague arguments which can be meaningless.	56	1
P10		**MEG groupware:**	58	
	W10.1	The groupware is not user friendly.	58	1
P11		**ARENA-M:**	62	1
	W11.1	Less experienced users felt difficulty while inputting data through mobile devices.	62	1
P12		**QARCC:**	73	1
	W12.1	Only few issues could not be identified by the semi- automated approaches i.e. QARCC.	73,76	2
	W12.2	QARCC cannot estimate the importance or value of a conflict.	76	1

4.6. Results

This section will provide the findings of systematic review in detail.

4.6.1. Evolution of Requirements Negotiation Practices

Winwin spiral model (B. Boehm, Bose, Horowitz, & Lee, 1995) was based on the Theory-W and spiral model. The original spiral model (B. Boehm & Kitapci, 2006) was cyclic in nature with activities including elaborating system objectives, limitations and options at process and product level, analyzing the options and resolving major risks associated with them, elaborating and defining the product and process and in the end modifying plan for the next cycle. In this model, negotiation activities derived from Theory-W i.e. determining the key stakeholders of the system, identifying their win conditions and then negotiating those win conditions were embedded for conflict resolution which lacked in the earlier approaches (B. Boehm & Kitapci, 2006).

Winwin system (B. Boehm, et al., 1995) is a tool support for Winwin spiral model and its artifacts are based upon the Winwin negotiation model. Winwin system is a manual support for winwin spiral model and it provides a flexible way to negotiate requirements whereas past tools were not linked with the requirements engineering approaches. Winwin system based reconciliation depends upon three main concepts i.e. win conditions, Conflicts/Risk/Uncertainty specification (CRU's) and Points of Agreements (POA's). Multiple stakeholders use winwin support system for entering their own win conditions and knowing others' win conditions. These win conditions are then stored in a database and managed by a system engineering organization. The major conflicts or risks are identified by the tool support for preparing a list of CRU's. CRU helps to negotiate among stakeholders. Winwin tool also helps to negotiate with the stakeholders for conflict resolution. When all conflicts resolved then POA's are documented and then closed. The closed POA's documents can be used for building system requirements specifications. This tool, unlike past tools, did not emphasize on user requirements; rather emphasized on the elicitation of requirements from stakeholders directly (Lee, 1996). The evidence showed that this model attempted to resolve *viewpoint conflicts*, *feasibility conflicts* and *terminology conflicts* but paid less attention on resolving *quality attributes conflicts* (B. Boehm & Egyed, 1998; B. Boehm & In, 1996; Hoh, Boehm, Rodger, & Deutsch, 2001). The literature says that winwin spiral model and winwin system can resolve both quality attribute conflicts i.e. *cost conflicts* and *viewpoint conflicts* (B. Boehm & Hoh, 1996).

QARCC- Quality Attribute Risk And Conflict Consultant (B. Boehm & In, 1996) is a knowledge-based tool for identifying and resolving particularly quality requirements conflicts. QARCC was developed on the basis of Winwin system. QARCC (Hoh, et al., 2001) uses 'attributes' part of domain taxonomy for determining quality requirements conflicts. When stakeholders input win conditions then they determine which domain taxonomy elements are appropriate. QARCC then uses its database to identify potential conflicts among quality requirements. It has been observed both in literature and evidence that the tool works well when the no. of requirements increases in size and becomes difficult to handle manually (B. Boehm & Hoh, 1996; B. Boehm & In, 1996). The literature and evidence showed that QARCC worked well enough for identifying and resolving high and low level *quality attribute conflicts* (B. Boehm & Hoh, 1996; B. Boehm & In, 1996). The knowledge-base improved the process of conflict identification and conflict handling. QARCC was unable to determine the value of requirements conflicts. It did not provide a predefined list to identify requirements conflicts as well. It also consumed time to determine insignificant conflicts which was also a demerit of tool (H. In, Boehm, Rodgers, & Deutsch, 2001).

Simplifier and complicators (S&C's) approach (B. Boehm, Abi-Antoun, Port, Kwan, & Lynch, 1999) resolved two-culture problem between users and developers. S&C's approach used simplifiers and complicators lists which showed that how applications could be made easy or difficult to implement from both developer and client perspectives. Both kinds of lists helped to understand the expectations of one another. The evidence showed that it resolved *viewpoint conflicts* and *feasibility conflicts*. Apparently the approach did not belong to winwin theory but actually the approach used win-win behavior to reduce requirements conflicts among simplifiers and complicators.

Winwin system was enhanced into *Easywinwin* (Grunbacher, 2000) because winwin system provided inadequate tool support for winwin negotiation model. The activities of Easywinwin process (B. Boehm & Kitapci, 2006) include reviewing and expanding negotiation topics, brainstorming the interests of stakeholders, converging on win conditions, capturing a glossary of terms, prioritizing win conditions, surfacing issues and constraints, identifying issues and options and then negotiating agreements. The major results of the negotiation process are: well organized negotiation topics in the domain taxonomy, a glossary of difficult or ambiguous terms, the agreements and major risks need to be handled. The evidence and literature showed that Easywinwin tool increased only tool facilitation and it is well suited for face-to-face communication but weakly supports distributed negotiation (Seyff, Hoyer, Kroiher, & Grunbacher, 2005). The evidence only reported that the tool resolves requirements conflicts but it has not been clearly reported what kinds of conflicts are resolved by this tool. The critical review of empirical study and literature showed that it emphasizes on the resolution of *viewpoint conflicts, terminology conflicts and requirements contradiction* (Grunbacher, 2000; Grünbacher, Köszegi, & Biffl, 2006).

ARENA tool was developed for supporting negotiation in distributed environment but it was a desktop application. *ARENA-M- Anytime, anyplace Requirements Negotiation Aids-Mobile* (Seyff, et al., 2005) was a technique addressing the issue of mobile negotiation i.e. *anytime anyplace.* It was based on the Easywinwin and ARENA-II system. ARENA-II is a web application based on ASP.NET. It allows multiple stakeholders to elicit and reconcile requirements in the geographically distributed environment. It supports asynchronous mode of communication. It supports all Easywinwin negotiation activities and provides stakeholders guidance while elaborating agreements. The facilitator is still required during the use of this tool. It could be used only for desktop computers. The availability of stakeholders was an issue. That's why mobile ARENA-M tool was developed which used mobile browser. ARENA-M comprised all the features of ARENA-II with some extensions. This tool best suited for mobile users and it resolved availability issue of stakeholders as well. The evidence and literature showed that the main emphasis of tool was usability issue. The tool only enhanced the communication mode support i.e. asynchronous mode. It was not clearly mentioned in the evidence and literature that which types of conflicts were handled by the tool.

Easywinwin was evolved into *Wikiwinwin* (Da, Di, Koolmanojwong, Winsor Brown, & Boehm, 2008) tool which used Wikiwinwin requirements negotiation process model. Easywinwin did not support adequately to cope with evolving nature of agreements and win conditions. Wikiwinwin process first of all establishes negotiation context for winwin through stakeholder identification and discussing issues with the stakeholders. The second major activity is to negotiate Win conditions, Issues, Options and Agreements with the help of brainstorming, converging and surveying of win conditions. The negotiation results are continuously refined with the passage of time as the project proceeds. These negotiated results are used to create the SSRD- System and Software Requirements Description. Wikiwinwin improved shared vision among stakeholders and information management. The evidence showed that Wikiwinwin only enhanced the tool facilitation and it partially resolved the *resource conflicts.*

The *CBSP- Component-Bus-System-Property approach* (Grunbacher, Egyed, & Medvidovic, 2001) was originated from Winwin approach and ATAM (Architecture Tradeoff Analysis Method) approach. The CBSP approach was developed to represent a method for tradeoff between software requirements and software architecture. The CBSP process starts with the documentation of software requirements then it filters next level requirements, in this step only most critical requirements are considered. Now CBSP taxonomy is applied on those selected requirements. In the next step the requirements are classified and analyzed by the experts architecturally using CBSP taxonomy. The conflicts among requirements from architectural point of view are resolved after consulting with the team members. The requirements are then accepted or rejected on the basis of degree of voting for requirements. If majority of team members agree that a requirement is important from architectural point of view then it is included otherwise excluded. The overlapping requirements are refined or rephrased to avoid ambiguities. In the end, the refined requirements are transformed into CBSP artifacts when no conflict is left. Now a draft-architecture can be built on the basis of these CBSP artifacts. The evidence showed that earlier approaches did not consider architectural knowledge for conflict handling. The CBSP approach attempted to resolve *viewpoint conflicts* and *terminology conflicts.*

ICRAD- Integrated Conflict Resolution and Architectural Design process (Herrmann, et al., 2006) was extracted from ATAM, CBAM, CBSP and Winwin negotiation model. The pre-existing architecture-centric approaches found in the literature did not support conflict resolution adequately. ICRAD process helps to choose right architectural design for implementing a set of requirements. The inputs to the process are software requirements while the output generated by the process is negotiated win conditions with the additional knowledge of their feasibility, cost and benefit, and architectural style. The process is started with the identification of most critical requirements during requirements specification step. Then requirements inter-dependencies or conflicts are identified in the form of requirements bundles. Some requirements conflicts are also solved in this step. In the next step, the identification of logical components reduces the requirements complexity. Now a high-level architecture is designed with alternative solutions. The negotiation for selecting an architectural style is carried out. Then it is checked whether the selected architectural design resolves requirements conflicts and produces the desired functionality or not. When high-level design is finalized then low-level design will be designed. The empirical evidence showed that this approach enhanced the degree of formality during requirements transformation. The approach mainly emphasized to resolve *feasibility conflicts* adequately but the study claims that it can also resolve *requirements contradiction* and *terminology conflicts* which might appear during requirements transformation.

Hybrid method (Kitapci & Boehm, 2007) framework was developed from MBASE (Model-Based Architecture in Software Engineering) and six other approaches i.e. mixed initiated template, NLP, Keyword analysis, inspection, walkthrough and Winwin negotiation approach. Hybrid method transforms informal requirements into formal ones. It automatically uses well managed Easywinwin artifacts in a draft requirements template considering the predefined criteria for the properties of template. It uses NLP, Keyword analysis and inspection methods for analyzing, controlling and increasing the quality of requirements specifications. The feedback process uses the data collected from NLP, keyword analysis and inspection which helps to increase process improvement and gather defect information. It allows strong support for decisions on the quality of intermediate outcomes with the aid of quality checks. This helps stakeholders to remove defects and improve their decisions based on the requirements. The empirical evidence showed that the method only improved the degree of formality during requirements transformation. It had also been reported in the study that the method attempted to resolve *viewpoint conflicts*. The study claims that this method can also resolve *terminology conflicts* during requirements transformation.

MEG (Ramires, Antunes, & Respicio, 2005) tool was based on IBIS- Issue Based Information Systems, Winwin approach and SQFD - Software Quality Function Deployment approach. MEG just provided a tool support for SQFD process. It integrated collaboration, negotiation and argumentation based negotiation. SQFD approach was not winwin-based but later on winwin theory was used in MEG tool because at that time winwin theory was well known. MEG tool uses SQFD matrix which has cells that correlate user requirements with the technical specifications. The tool also uses integrative style of negotiation by voting method. The votes are considered on the basis of integrative and distributive attitudes during the use of system. After an agreement the tool provides a fair solution. MEG tool does not compel to accept the agreement. MEG provides features of both generic GSS (Group Support Systems)-based tools and specific NSS (Negotiation Support System)-based tools. MEG uses IBIS to manage arguments for the requirements. The evidence showed that MEG tool attempted to resolve mainly *viewpoint conflicts*. The study also claims that the tool can resolve *quality attribute conflicts* as well.

FTR- Formal Technical Review (Linhares, Borges, & Antunes, 2009) tool used FTR process and negotiation-collaboration process. FTR tool was not based on Winwin because FTR process was developed in 1976. The negotiation-collaboration process is applied on FTR process with the help of this tool. FTR tool was needed because the current tools did not provide negotiation-collaboration equilibrium. FTR tool solves this problem by the combination of models of collaboration, argumentation and negotiation. This tool is used in the field of software engineering during Functional Specification phase. This tool supports both FTR process and negotiation-collaboration process. FTR is an activity of software quality assurance process and used during verification and validation of software. The input of the negotiation-collaboration process is a problem statement. After getting the input the process executes its three phases. In the first phase, several proposals are elicited from the stakeholders. In the second phase, a consensus is built for each proposal and reasons both for being *in the favor of or against* the proposal have been recorded as arguments. In the third phase, a decision has been taken after analyzing the feasibility of solutions and risks associated with them. The evidence showed that FTR tool mainly emphasized to resolve *viewpoint conflicts* and *terminology conflicts*.

The following figure 1 shows the quick view of the relationship between requirements negotiation practices and type of conflicts addressed by these practices.

Figure 1: Conflict Resolution of Practices, Corresponding to R.Q. 1

Requirements negotiation practice	**Requirements contradiction**	**Terminology conflicts**	**Quality attribute conflicts**	**Viewpoint conflicts**	**Resource conflicts**	**Feasibility conflicts**
Simplifiers and Complicators approach	○	○	○	●	○	●
Winwin spiral process model	○	●	◑	●	○	●
Wikiwinwin approach	○	○	○	○	●	○
Winwin system	○	○	●	●	○	○
QARCC	○	○	●	○	○	○
Easywinwin methodology	●	●	◑	●	○	◑
ARENA-M	◑	◑	◑	◑	◑	◑
The CBSP approach	○	●	○	●	○	○
ICRAD	◑	◑	○	○	○	●
Hybrid method approach	○	◑	○	●	○	○
FTR tool	○	◑	○	●	○	○
MEG tool	○	○	◑	●	○	○

Scale:

Yes = ●
No = ○
Not reported = ◑

The following figure 2 and 3 shows the origin of each requirements negotiation practice and how a practice was enhanced from the previous practice. Figure 2 focuses on the winwin based requirements negotiation practices while figure 3 specifically deals with the approaches which are not based on the winwin behavior.

Figure 2: Relationship of Theory-W with Winwin Approaches

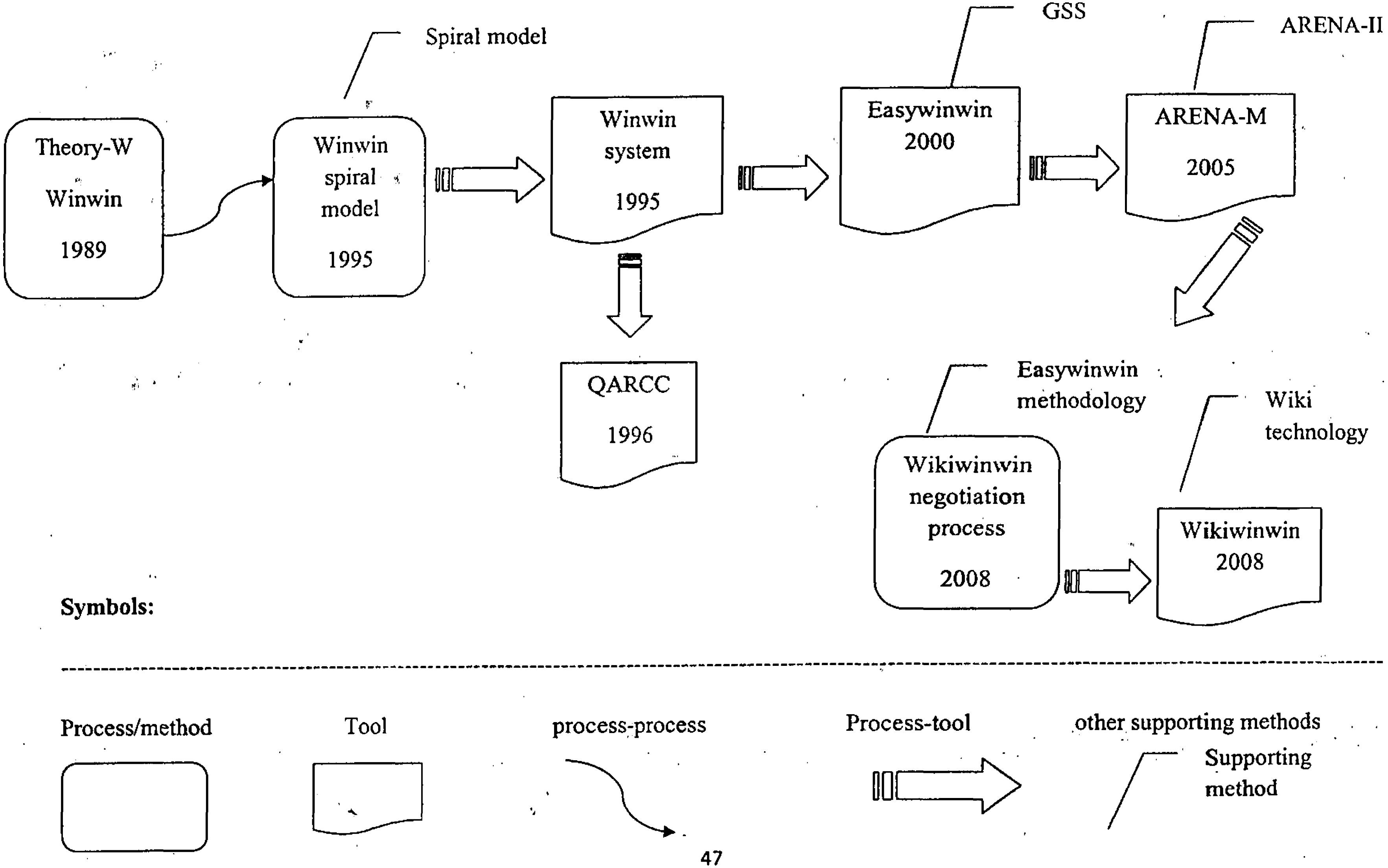
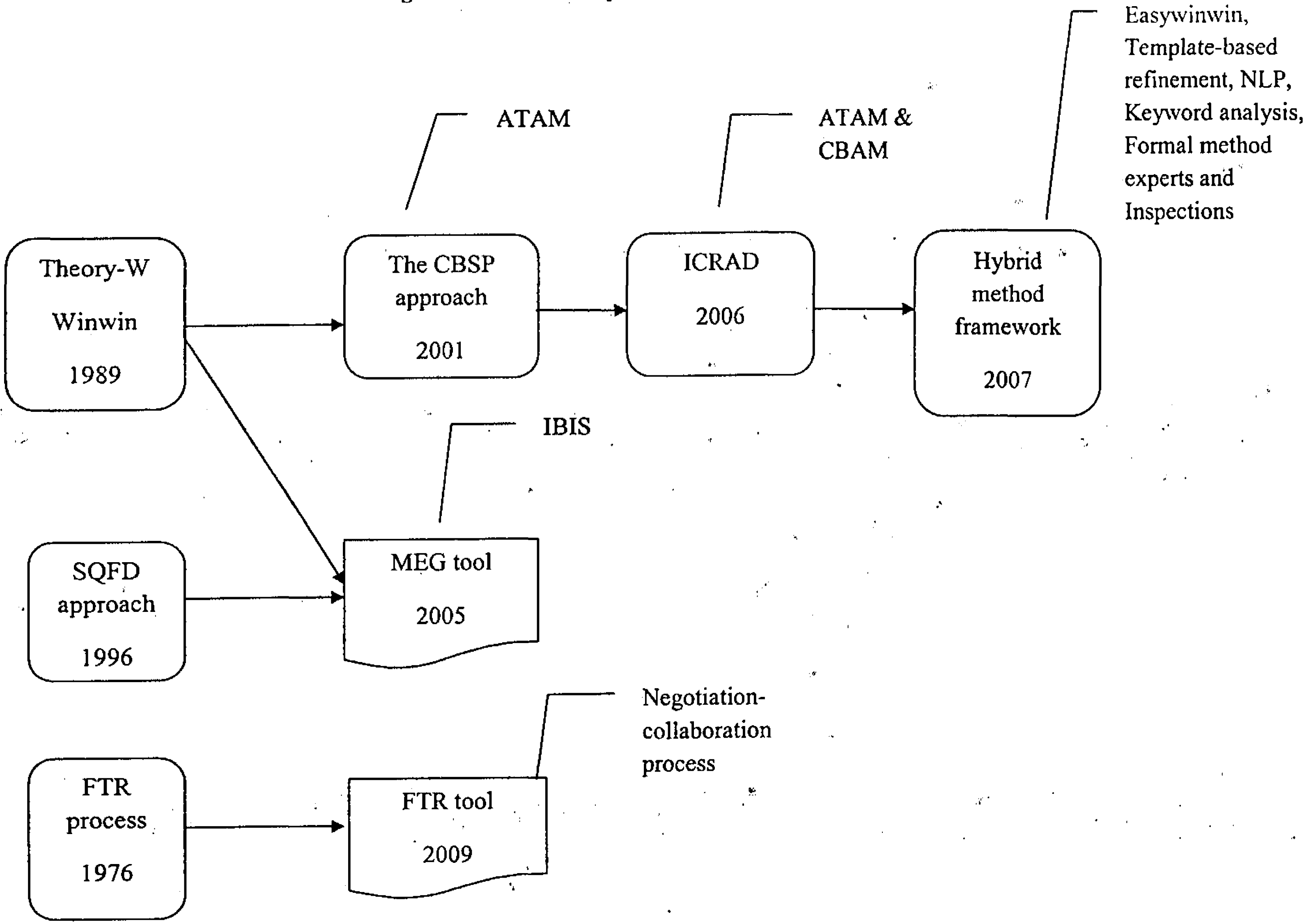


Figure 3: Relationship of Theory-W with Other Approaches



4.6.2. Frequency of Approaches

It has been observed that most of the approaches are based on the Theory-W Winwin approach. Table 8 shows that winwin based approaches are 11/12 i.e. 91.7% and the other approaches found having different origin are 1/12 i.e. 8.3%.

Table 8: Frequency of Approaches on the Basis of Their Origin

Origin-based classification	No. of approaches	Percentage
Winwin-based approaches	11/12	91.7%
Other approaches	1/12	8.3%

4.6.3. Significance of Theory-W

It is obvious from the results that winwin negotiation theory is the base of various modern methodologies. A lot of work has been done in the field of requirements negotiation due to the existence of winwin theory. The techniques and tools developed on the basis of Winwin are more accurate and effective than other approaches. Winwin theory addresses human behavior in a distinct manner. Winwin theory (Kenneth W. Thomas & Kilmann, 2010) is used when both the parties highly trust each other and are willing to resolve the conflicts, when you want to learn the objectives of other stakeholders or when you want mutual gain. Winwin (Kenneth W. Thomas & Kilmann, 2010) outcome is not possible to implement when there are highly threatening issues where stakeholders are non-cooperative or when others' concerns are not incorporated in the agreements. It is also not feasible when winwin behavior is overused like spending more time in conflict resolution and it is possible that some stakeholders may get more benefit due to high trust than the other stakeholders which may result in disliking of winwin negotiation. Collaborative process is very time consuming and needs a lot of energy to reach an agreement. It had been empirically evaluated (Kenneth W. Thomas & Kilmann, 2010) that only 7% organizations used winwin negotiating style whereas accommodating style was highly used i.e. 87%. The other style like competing was used 79%, compromising was used 58% and avoiding was used 22%.

As it is obvious that it is not possible to implement winwin theory in *all* the conflicting situations effectively. There are few methods available in the literature which can be used in such kind of conflicts when winwin behavior is not possible.

Sulha (Gellman & Vuinovich, 2008) is a reconciliation process used by Arabs which is based on the Islamic concept. In this method a third party called *mediator* is involved who is responsible for the whole settlement among the key stakeholders with honor and dignity. The end results (like level of satisfaction) of this process are more convincing than winwin theory. There is another method called *third party consultation* (Fisher, 1972) developed by Fisher which can also be used in a situation when both parties do not want to leave their positions. This method also focuses on the presence of a mediator which is responsible for the tradeoff of substantial issues between the conflicting parties. The effectiveness of the method depends on the third party's impartial and unbiased behavior.

4.6.4. Implementation of Requirements Negotiation Approaches

Requirements negotiation approaches have been implemented both in academia and industry. Table 9 shows that the frequency of using requirements negotiation tools and methodologies is variant. The trend in the table shows that mostly these tools are used for academia and there is a little use of tools in the industry.

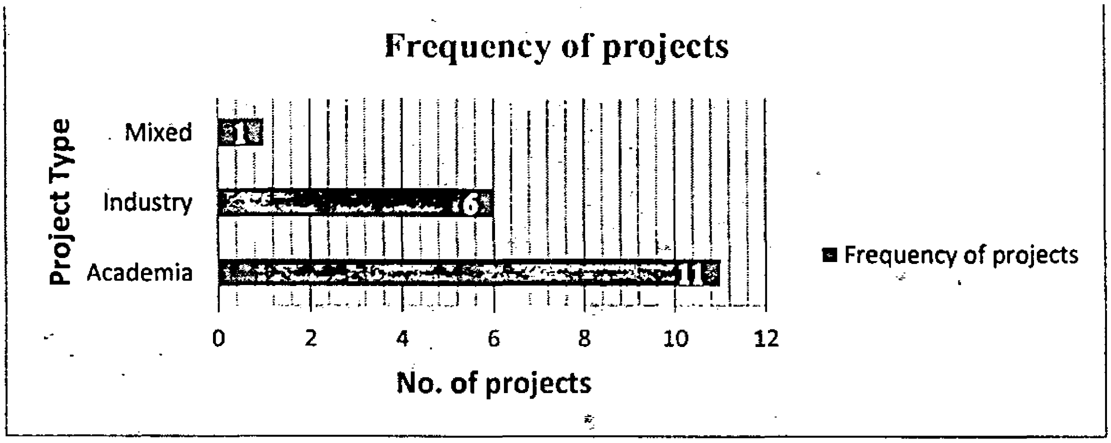
Table 9: Practical Implementation of Practices, Corresponding R.Q. 1.2 & 1.3

Study id	Name of practice	Frequency	Study type	Project type	Subjects	Year
25	*Winwin system*	4	Experiment	University library	Students and clients	1998
77			Experience report	University education	Students and teachers	1998
73			Case study	University library	Students	2001
39			Experience report	University education	Students and teachers	2006
76	*QARCC tool*	2	Experiment	Industry	Not reported	1996
73			Case study	University library	Students and clients	2001
32	*Easywinwin methodology*	3	Case study	University library and bookstores	Students and clients	2000
39			Experience report	University education	Students and teachers	2006
54			Case study	University e-services	Students and real clients	2007
20	*Wikiwinwin*	3	Case study	University library	Students and clients	2008
69			Case study	Real-client university projects	Students, clients & users	2009
68			Case study	Real-client university projects	Practitioners	2010
62	*ARENA-M tool*	1	Experiment	Mixed	Students and practitioners	2005
56	*FTR tool*	1	Industrial experiment	Telecommunication industry	Employees	2009
58	*MEG tool*	1	Industrial experiment	National pension system in a Government agency	Practitioners	2005
9	*Simplifiers and complicators approach*	1	Experiment	20 digital library projects of university	Students and clients	1999
10	*Winwin spiral model*	3	Experiment	Satellite ground station	Not reported	1995
72			Case study	University library	Students and clients	1998
77			Experience report	University education	Students and teachers	1998
34	*The CBSP approach*	1	Case study	Cargo router system	Real clients	2001
43	*ICRAD*	1	Case study	Health industry project	Practitioner and domain expert	2006
54	*Hybrid method*	1	Case study	University e-services	Students and real clients	2007

1.1.1. Frequency of Projects in Organizations

It was observed from the results that out of total 18 projects, 11/18 projects were performed in academia, 6/18 projects were conducted in the industrial organizations and 1/18 were mixed projects. From figure 4, it is obvious that industry projects were 33.33%, academic projects were 61.11% and 5.55% were mixed nature projects. Figure 4 shows that more studies had been conducted in the academia rather than industrial settings.

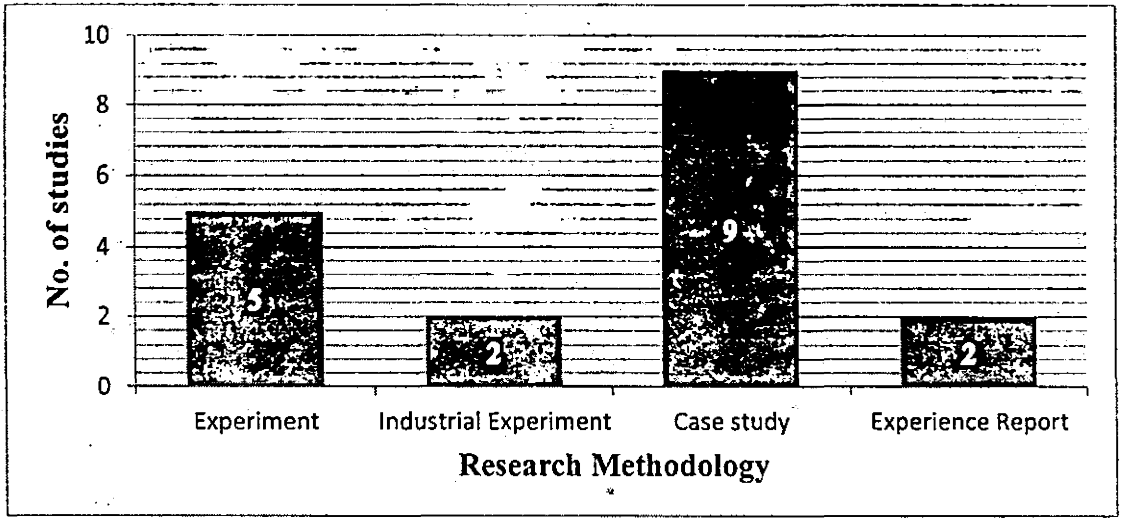
Figure 1: Frequency of Projects, Corresponding to R.Q. 1.2 & R.Q. 1.3



1.1.2. Frequency of Used Research Methodologies

The primary studies selected for systematic review used four types of research methodologies which were experiment, industrial experiment, case study and experience report. Figure 5 shows that 27.77% studies i.e. 5/18 used experiments, 50% studies i.e. 9/18 used case studies, 11.11% studies i.e. 2/18 used experience reports and again 11.11% studies i.e. 2/18 used industrial experiments. It was concluded that research community focused more on experiments and case studies rather than experience reports.

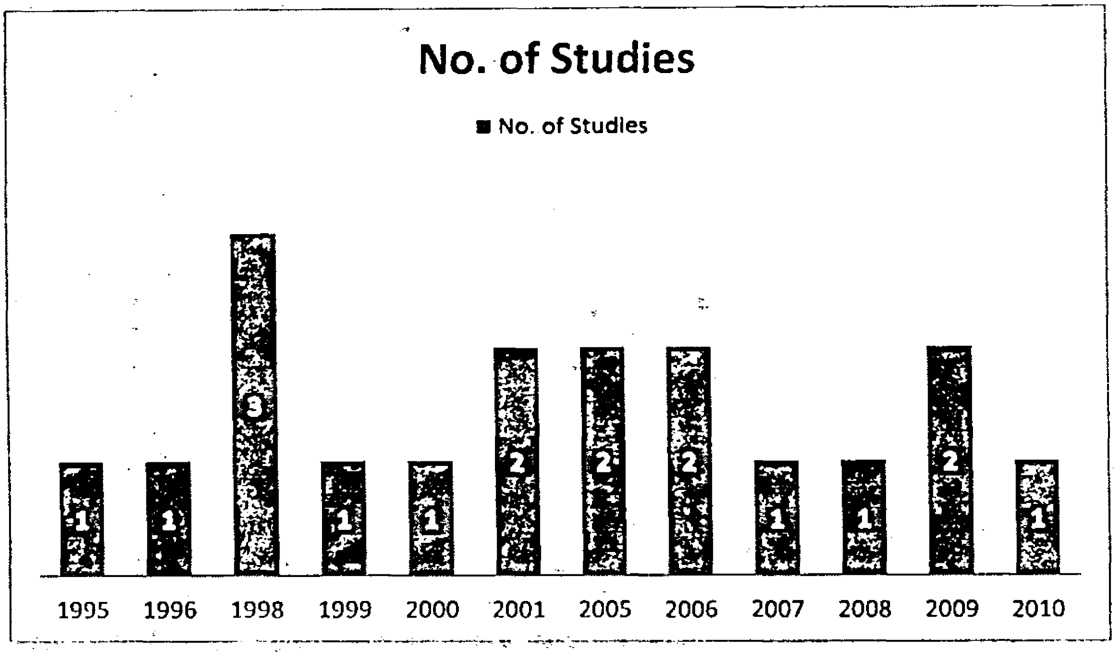
Figure 2: Frequency of Used Methodologies, Corresponding to R.Q. 1.2 & 1.3



1.1.3. No. of Empirical Studies per Year

The following figure 6 shows that no. of empirical studies published since 1995 to 2010. There were total 18 empirical studies. The maximum no. of studies published are 3/18 in 1998 whereas the minimum no. of studies published are 1/18 which is found in many years i.e. 1995, 1996, 1999, 2000, 2007, 2008 and 2010. The medium no. studies (2/18) have been found in the year 2001, 2005, 2006 and 2009. It shows us that the no. of empirical studies published per year in the field of requirements negotiation is very low. There is a great need to do more empirical work in this area because the importance of negotiation outcome is significant in the success of software projects.

Figure 3: Number of Empirical Studies per Year, Corresponding to R.Q. 1.3



1.1.4. Evaluation of Requirements Negotiation Practices

The requirements negotiation practices evolved with the passage of time and their functionality was either enhanced or entirely a new approach arose. Figure 7 shows the issues which were covered/not covered by the requirements negotiation practices. The scale used in the figure 7 is *yes, no* and *not reported.*

The figure 7 shows that Easywinwin is the best requirements negotiation practice which provides adequate tool facilitation. Easywinwin provided great support for the users and reduced communication barrier effectively. Winwin spiral model is also a good quality model in conflict resolution which laid foundation for the development of high quality tools. Wikiwinwin is also a better practice and it also encouraged users to adopt it due to its simplicity. Wikiwinwin is the latest requirements negotiation practice. The architecture centric approaches were also very effective like hybrid method is a high quality practice. The difference between the quality scores of hybrid method, ICRAD and CBSP approach is very close to one another. Therefore, they are also encouraged to apply in the real environment. The low grade practices identified are simplifiers and complicators approach, Meg tool and FTR tool.

Figure 4: Issues Covered by Practices, Corresponding to R.Q. 1.2 & 1.3

Issues addressed by practices	S & C's approach	Winwin spiral model	Winwin system	QARCC	Easywinwin	ARENA-M	Wikiwinwin	CBSP approach	ICRAD	Hybrid method	MEG tool	FTR tool
Support winwin Theory W	●	●	●	●	●	●	●	●	●	●	●	○
Tool support	◑	●	●	●	●	●	●	●	◑	●	●	●
Conflict identification	●	●	●	●	●	◑	●	●	●	●	●	●
Conflict resolution	●	●	●	●	●	◑	●	●	●	●	●	●
Improves stakeholders involvement	○	●	●	●	●	●	●	◑	◑	●	●	◑
Managing volatile requirements	◑	●	●	◑	●	◑	●	●	●	●	◑	◑
Collaboration and cooperation	◑	●	●	◑	●	◑	●	◑	◑	●	●	●
Customer satisfaction	◑	●	●	◑	●	●	◑	◑	◑	◑	◑	◑
Less time consuming	◑	●	◑	◑	●	●	●	◑	◑	◑	●	●
Face to face communication	◑	◑	◑	◑	●	○	◑	●	◑	●	○	●
Mobile communication support	○	○	○	○	◑	●	○	○	○	○	○	○
Distributed negotiation	◑	◑	●	◑	●	○	◑	○	◑	◑	●	●
Cost-effetiveness	●	●	◑	●	◑	●	●	●	●	◑	◑	◑
Knowledge-base support	○	○	○	●	○	○	○	○	○	○	○	○
Requirements complexity and interdepency	◑	●	○	◑	◑	◑	◑	●	●	◑	◑	◑
Minimizing complexity of tool	○	○	◑	◑	●	○	●	○	○	○	◑	○
Group facilitation support	○	◑	◑	◑	●	◑	●	○	◑	●	◑	◑
Requirements transformation	○	●	○	○	●	◑	◑	●	●	●	○	○
Context-aware negotiation	○	◑	○	○	◑	●	○	○	○	○	○	○
Architectural knowledge required	○	○	○	○	○	○	○	●	●	●	○	○
Direct user interaction	◑	●	●	◑	●	●	●	●	◑	●	○	◑
Stakeholder guidance and facilitation	○	◑	◑	◑	●	◑	●	●	◑	●	◑	○
Total score	*9*	*16*	*13*	*13*	*18*	*13*	*16*	*14*	*13*	*16*	*12*	*11*

Scale: **Yes:** ● No: ○ Not reported: ◑

1.1.5. Threats to the Validity

The results of a systematic review should be unbiased. There are different kinds of validity threats which can be used to describe the chance of bias in the systematic review. The most important kinds are: internal validity and external validity.

Internal validity: The underlying research work has been performed by a single author therefore there can be chances of personal bias. To mitigate this problem, the thesis supervisor counter checked all the important activities of systematic review. There was a slight change between review protocol and execution. Some of the tables were useless and therefore they were not included in the execution of systematic review. This might also cause some variation.

External validity: The underlying systematic review has strong external validity. The author reviewed sufficient evidence available in the literature to support the results but still there are issues which can produce bias. For example, the search date, search sources and search terms may be limited. The grey literature was not accessible; therefore it was avoided. The conflict types addressed by the practices are also limited. The review protocol was piloted for its validity and it was also critically reviewed by the supervisor and an external authority.

CHAPTER 5

CONCLUSION

5. Conclusion

This chapter includes the discussion on the principal findings of the systematic review. The chapter also presents recommendations and future work for both practitioners and researchers.

5.1. Discussion

In this study we analyzed only evidence-based requirements negotiation practices and other practices were excluded from the study. The most distinguished search resources: IEEE, Springer Verlag, ACM, Science Direct and EI-compendex databases were chosen for ensuring the strength of evidence so that solid and concrete arguments could be derived. Validity threats were also addressed adequately.

The deep analysis of the results revealed that most of the requirements negotiation practices (91.7%) were based on the Theory-W (B. W. Boehm & Ross, 1989). There was little (8.3%) work done on the other approaches like FTR (Linhares, et al., 2009) and MEG (Ramires, et al., 2005) tools. There was little guidance and very rare practices available in the literature which addressed the issue of what to do if winwin situation does not work? No doubt the literature said that convert lose-lose, lose-win and win-lose situations into winwin situation but it had been observed that winwin relationship was not possible all the times especially when both the parties did not want to leave their positions. Winwin approach was only applicable when both parties were willing to cooperate for mutual gain. There was little work found in this regard; therefore it needed special attention of the researchers for future directions. To address this issue of winwin theory, few models like *Sulha* (Gellman & Vuinovich, 2008) *and Third party consultation* (Fisher, 1972) were used in the literature. Both these models were used when winwin situation did not work. The evidence collected from the study (Kenneth W. Thomas & Kilmann, 2010) showed that only 7% of all kinds of industries encouraged implementing collaborative behavior which was very low. The basic theme of collaborating is to satisfy everyone and accommodate the concerns of other stakeholders as well. It had been observed that highly used negotiating style was accommodating (87%). The basic theme of accommodating style is to *"kill your enemies with kindness"* (Kenneth W. Thomas & Kilmann, 2010). The other negotiating styles like competing (79%), compromising (58%) and avoiding (22%) were used by all type of industries. The main reason for the decline of winwin was that it was very time consuming to reach an agreement. Due to high trust, one stakeholder might get more benefit than the other and it was not used in the threatening issues.

The evidence collected in this study showed that winwin-based approaches were studied more than other approaches in software engineering. Another important issue was that only one school of thought did a lot of work on winwin based negotiation approaches and they did not try to use other theories which could give even better results. Therefore, there is a need for using theories other than Theory-W because these approaches could not convince industry to use them in their software projects. The most widely accepted and studied requirements negotiation practice was *Easywinwin* which resolved issues adequately than other approaches. There were other approaches like Wikiwinwin, hybrid method and winwin spiral model which resolved issues highly but less than Easywinwin. Some practices were used moderately as they resolved issues moderately like winwin system, QARCC, ARENA-M, ICRAD and CBSP. Some practices were used very rarely as they resolved issues inadequately like simplifiers and complicators approach, FTR tool and MEG tool.

The evidence collected in this study showed that requirements negotiation practices can be divided into three different varieties. In the beginning, there were purely *winwin-based practices* and then the dimension of practices changed to *architecture-centric approaches* which included the dimension of software architecture for conflict resolution. In the third variation, there were *validation-based practices* which added the dimension of software quality. It had been observed that most of the times the tool facilitation increased in the next coming tool. In some cases, few concepts were combined to provide a new tool but the theory behind it was not changed. For instance, when Easywinwin (Grunbacher, 2000) was enhanced into Wikiwinwin (Da, et al., 2008) then the major change was only tool facilitation.

The empirical evidence showed that the requirements negotiation practices addressed only some conflict types adequately but others inadequately. It has been realized that *viewpoint conflicts* were addressed by the most requirements negotiation practices i.e. 44.4% whereas requirements contradiction and resource conflicts were addressed very rarely i.e. only 5.6%. The other conflict types were addressed moderately like terminology conflicts (16.7%), quality attribute conflicts (11.1%) and feasibility conflicts (16.7%).

It had been realized that number of empirical studies published per year were very rare. The quality of reporting empirical studies was also poor. It was observed during literature survey that there was no specific classification of requirements conflicts that exist in the literature. Several authors worked in this regard but they used their own terminologies and when we compared those concepts then only terminologies were different but concepts were same. Therefore, there is a lack of standardized classification of requirements conflicts in the literature. This aspect needs special attention of researchers.

5.2. Recommendations

This section will describe the advice for both practitioners and researchers. In this section, future direction will be highlighted which needs due attention.

5.2.1. Implications for Practitioners

The main purpose of this systematic review is to identify requirements negotiation practices which help in conflict resolution. We also uncovered the strengths and weaknesses of requirements negotiation practices. We analyzed during this study that *Easywinwin* (Grunbacher, 2000) is the most widely accepted requirements negotiation practice because it helps in resolving diversity of issues more than other approaches. This approach is applicable when both the parties are willing to participate. An adequate research work has been carried out on Easywinwin and people used it in different ways and it also produced better results. No doubt Easywinwin is the most promising requirements negotiation practice but we couldn't find enough evidence for its usage in industries. Therefore, practitioners are encouraged to use it.

The most commonly addressed requirements conflict type is *viewpoint conflicts* and less attention has been paid to the resource conflicts and requirements contradiction conflicts. There are three types of requirements negotiation practices which are *winwin-based*, *architecture-centric* and *validation-based* practices. Among these, winwin based practices are more popular and widely studied whereas architecture-centric are moderately good while validation-based practices are rarely used. The main focus of most of the studies was increasing tool facilitation or extending the previous approach. No radical change had been observed.

It had been observed that the number of studies carried out in the industrial settings was comparatively less as compared to academia. If both academia and industry would coordinate strongly then better approaches with better results could be achieved and the gap between industry and academia could be reduced.

The evidence collected in this study showed that industrial experience reports available in this area of interest were very rare (11%). So there is a need for more industrial reports by the practitioners in the field of requirements negotiation.

5.2.2. Future Work

There are various future research directions for the research community. The evidence showed that winwin based requirements negotiation practices are widely used whereas little work has been done on the other approaches which are not winwin based; for example FTR and MEG tool. The following possible research areas need further exploration:

- The main problem observed in the literature was that only one school of thought (Boehm and his coworkers) emphasized on the winwin based approaches. They used only one theory i.e. Theory-W. We analyzed that there were other theories as well in disciplines other than software engineering and if they were used, then better approaches could be developed.
- It was not possible that always winwin situation would exist. Therefore, in such a situation there was no empirical work found in the field of software engineering. To address this issue, some evidence was found in social sciences.
- There was no standard classification of requirements conflicts found in the field of software engineering which opened a gateway for future work.
- There was little empirical work done in the field of requirements negotiation. Therefore there was a need of more empirical work in this area of interest. It had been observed that number of non-empirical studies was far more than empirical studies. The quality of reporting empirical studies was also poor. There is a need for a standardized way of reporting empirical studies.

- It had been recognized that most of the practices focused on the *viewpoint conflicts* and little attention given to the other conflict types like *quality attributes conflict* and *resource conflicts*. Therefore there was a need for developing such type of requirements negotiation practices which could address other conflict types as well.

REFERENCES

Al-Naeem, Tariq, Gorton, Ian, Babar, Muhammed Ali, Rabhi, Fethi, & Benatallah, Boualem (2005). *A quality-driven systematic approach for architecting distributed software applications*, St. Louis, MO, United states.

Boehm, B., & Egyed, A. (1998). *WinWin Requirements Negotiation Processes: A Multi-Project Analysis.* Paper presented at the Proceedings of the 5th internationalconference on software processes.

Boehm, B., & Hoh, In (1996, 21-23 Aug 1996). *Software Cost Option Strategy Tool (S-COST).* Paper presented at the Computer Software and Applications Conference, 1996. COMPSAC '96., Proceedings of 20th International.

Boehm, B., & In, H. (1996). Identifying quality-requirement conflicts. *Software, IEEE, 13*(2), 25-35.

Boehm, B., Port, D., & Al-Said, M. (2000). Avoiding the software model-clash spiderweb. *Computer, 33*(11), 120-122.

Boehm, B. W., & Ross, R. (1989). Theory-W software project management principles and examples. *Software Engineering, IEEE Transactions on, 15*(7), 902-916.

Boehm, Barry, Abi-Antoun, Marwan, Port, Dan, Kwan, Julie, & Lynch, Anne (1999). Requirements engineering, expectations management, and the two cultures. *Proceedings of the IEEE International Conference on Requirements Engineering*, 14-22.

Boehm, Barry, Bose, Prasanta, Horowitz, Ellis, & Lee, Ming June (1995). *Software requirements negotiation and renegotiation aids: a theory-W based spiral approach*, Seattle, WA, USA.

Boehm, Barry, & Kitapci, Hasan (2006). The WinWin Approach: Using a Requirements Negotiation Tool for Rationale Capture and Use *Rationale Management in Software Engineering* (pp. 173-190).

Bowen, T. F., Dworack, F. S., Chow, C. H., Griffeth, N., Herman, G. E., & Lin, Y. J. (1989, 3-6 Jul 1989). *The feature interaction problem in telecommunications systems.* Paper presented at the Software Engineering for Telecommunication Switching Systems, 1989. SETSS 89., Seventh International Conference on.

Chitchyan, Ruzanna, Sampaio, Americo, Rashid, Awais, & Rayson, Paul (2006). *A tool suite for aspect-oriented requirements engineering*, Shanghai, China.

Curtis, Bill, Krasner, Herb, & Iscoe, Neil (1988). A field study of the software design process for large systems. *Commun. ACM, 31*(11), 1268-1287.

Da, Yang, Di, Wu, Koolmanojwong, S., Winsor Brown, A., & Boehm, B. W. (2008). WikiWinWin: A Wiki Based System for Collaborative Requirements Negotiation. *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pp.24.

DeMarco, Tom (1996). *The role of software development methodologies: past, present, and future.* Paper presented at the Proceedings of the 18th international conference on Software engineering.

Dyba, T., Kitchenham, B. A., & Jorgensen, M. (2005). Evidence-based software engineering for practitioners. *Software, IEEE, 22*(1), 58-65.

Easterbrook, Steve (1994). Resolving requirements conflicts with computer-supported negotiation *Requirements Engineering* (pp. 41-65): Academic Press Professional, Inc.

Fisher, Ronald J. (1972). Third Party Consultation: A Method for the Study and Resolution of Conflict. *The Journal of Conflict Resolution, 16*(1), 67-94.

Gellman, Mneesha, & Vuinovich, Mandi (2008). From Sulha to Salaam: Connecting local knowledge with international negotiations for lasting peace in Palestine/Israel. *Conflict Resolution Quarterly, 26*(2), 127-148.

Grunbacher, P. (2000, 2000). *Collaborative requirements negotiation with EasyWinWin.* Paper presented at the Database and Expert Systems Applications, 2000. Proceedings. 11th International Workshop on.

Grunbacher, P., Egyed, A., & Medvidovic, N. (2001). *Reconciling software requirements and architectures: The CBSP approach*, Toronto, Ont, Canada.

Grünbacher, Paul, Köszegi, Sabine, & Biffl, Stefan (2006). Stakeholder Value Proposition Elicitation and Reconciliation *Value-Based Software Engineering* (pp. 133-154).

Grünbacher, Paul, & Seyff, Norbert (2005). Requirements Negotiation *Engineering and Managing Software Requirements* (pp. 143-162).

Herrmann, Andrea, Paech, Barbara, & Plaza, Damian (2006). ICRAD: An integrated process for the solution of requirements conflicts and architectural design. *International Journal of Software Engineering and Knowledge Engineering, 16*(6), 917-950.

Hoh, In, Boehm, B., Rodger, T., & Deutsch, M. (2001, 12-19 May 2001). *Applying WinWin to quality requirements: a case study.* Paper presented at the Software Engineering, 2001. ICSE 2001. Proceedings of the 23rd International Conference on.

In, Hoh, Boehm, Barry, Rodgers, Thomas, & Deutsch, Michael (2001). *Applying WinWin to quality requirements: a case study.* Paper presented at the Proceedings of the 23rd International Conference on Software Engineering.

In, Hoh Peter, Olson, David, & Rodgers, Tom (2002). *Multi-criteria preference analysis for systematic requirements negotiation*, Oxford, United kingdom.

Kassab, Mohamad, Constantinides, Constantinos, & Ormandjieva, Olga (2005). *Specifying and separating concerns from requirements to design: A case study*, Novosibirsk, Russia.

Khan, Kashif Ahmed (2006). *A systematic review of software requirements prioritization.* Unpublished Master Thesis, Blekinge Institute of Technology, Sweden.

Kim, Minseong, Park, Sooyong, Sugumaran, Vijayan, & Yang, Hwasil (2007). Managing requirements conflicts in software product lines: A goal and scenario based approach. *Data and Knowledge Engineering, 61*(3), 417-432.

Kitapci, Hasan, & Boehm, Barry W. (2007). *Formalizing informal stakeholder decisions - A hybrid method approach*, Big Island, HI, United states.

Kitchenham, Barbara A., Dyba, Tore, & Jorgensen, Magne (2004). *Evidence-Based Software Engineering.* Paper presented at the Proceedings of the 26th International Conference on Software Engineering.

Kitchenham, Barbara, & Charters, S (2007). *Guidelines for performing systematic literature reviews in software engineering* (EBSE Technical Report No. EBSE-2007-01).

Lee, M. J. (1996). *Foundations of winwin requirements negotiation system.* University of Southern California, Los Angeles.

Linhares, Giovana, Borges, Marcos, & Antunes, Pedro (2009). Negotiation-Collaboration in Formal Technical Reviews. *Groupware: Design, Implementation, and Use*, 344-356.

Liu, Chi-Lun (2009). *Ontology-based requirements conflicts analysis in activity diagrams*, Seoul, Korea, Republic of.

Mussbacher, Gunter, Whittle, Jon, & Amyot, Daniel (2009). *Semantic-based interaction detection in aspect-oriented scenarios*, Atlanta, GA, United states.

Nhlabatsi, Armstrong, Laney, Robin, & Nuseibeh, Bashar (2008). Feature interaction: The security threat from within software systems. *Progress in Informatics*(5), 75-90.

Ramanna, Sheela, Skowron, Andrzej, & Peters, James F. (2007). *Approximation space-based socio-technical conflict model*, Toronto, Canada.

Ramires, Joao, Antunes, Pedro, & Respicio, Ana (2005). *Software requirements negotiation using the software quality function deployment*, Porto de Galinhas, Brazil.

Rashid, Awais, Aksit, Mehmet, Brito, Isabel, Vieira, Filipe, Moreira, Ana, & Ribeiro, Rita (2007). Handling Conflicts in Aspectual Requirements Compositions *Transactions on Aspect-Oriented Software Development III* (Vol. 4620, pp. 144-166): Springer Berlin / Heidelberg.

Robinson, W N, & Volkov, V (1999). *Requirements Conflict Restructuring*. Atlanta: Georgia State University.

Sadana, Vishal, & Liu, Xiaoqing Frank (2007). *Analysis of conflicts among non-functional requirements using integrated analysis of functional and non-functional requirements*, Beijing, China.

Seyff, Norbert, Hoyer, Christoph, Kroiher, Erich, & Grunbacher, Paul (2005). *Enhancing GSS-based requirements negotiation with distributed and mobile tools*, Linkoeping, Sweden.

The Standish group CHAOS report (1995).

Thomas, K W (1976). Conflict and conflict management *Handbook of Industrial and Organizational Psychology* (Vol. 2, pp. 89-935).

Thomas, Kenneth W., & Kilmann, Ralph H. (2010). *Thomas-Kilmann Conflict Mode Instrument* (Report): Consulting Psychologists Press (CPP).

van Lamsweerde, A., Darimont, R., & Letier, E. (1998). Managing conflicts in goal-driven requirements engineering. *Software Engineering, IEEE Transactions on, 24*(11), 908-926.

APPENDIX A

THE REVIEW PROTOCOL

1. Background

Software requirements are win conditions which are finalized after the process of negotiation (Boehm, Bose, Horowitz, & Ming-June, 1994). Requirements negotiation is an important activity of requirement elicitation and requirements prioritization. Software requirements or win conditions are prioritized on the basis of attributes like "*critical, important, desirable and optional*" (Boehm, et al., 1994). Requirements negotiation is a collaborative process which involves *all* stakeholders. Requirements negotiation is an important activity to build stable requirements specification. It also helps to renegotiate as requirements are volatile in nature. Effective requirements negotiation can prevent many projects from failure as poor requirements negotiation is also a cause of project failure (In & Roy, 2001).

The basic reason for requirements negotiation is the conflict of interests among stakeholders. So conflict is the actual motive of requirements negotiation. Conflict is critical in the field of software engineering whereas it is not handled adequately or sometimes avoided by the current requirements negotiation methods. Conflict is unavoidable in software projects as stakeholders pursue conflicting win conditions (Grünbacher & Seyff, 2005).

The sources of requirements conflicts are quality attributes, functional requirements, cost/schedule overrun, major risks, crucial needs, software constraints, volatile requirements, model clashes and using more than one models (Boehm, Port, & Al-Said, 2000).

The main aim of requirements negotiation is identification and resolution of requirements conflicts. The outcome of negotiation is to understand the rationale behind disagreements among stakeholders. These conflicts of interests determine high risks which need to be handled in the software project (Grünbacher & Seyff, 2005).

Requirements negotiation is an umbrella activity which starts in the early stages and continues throughout the process of software development (Boehm & Egyed, 1998).

The empirical and non-empirical evidence (Grünbacher & Seyff, 2005) reported several advantages of requirements negotiation e.g. to understand constraints of software projects, handling volatile requirements, increasing team knowledge, eliciting tacit knowledge, exploring better alternatives and to manage software complexity.

In short, the evidence shows that requirements negotiation is a very important activity of requirements engineering. Without negotiation the software cannot achieve customer needs and desirable user satisfaction. The software development is a highly risk-oriented and uncertain field where customer and user can't explain their needs. Therefore, requirements are largely volatile in nature. To deal with such kind of problems, software requirements negotiation is a better solution to minimize or eliminate these problems. Therefore, it is important to note that without negotiation, software success will not be sure. Without negotiation, the resulting product will have poor customer and user satisfaction, lack of desirable quality; cost and schedule overrun which are the main reasons for the failure of any software project. Without negotiation, we can develop only that software which exhibit stable requirements and we know that requirements are rarely stable in software development. So there is a need of software requirements negotiation in almost all software projects because nearly all software requirements are volatile in nature.

The research work carried out in software engineering is still considered to be immature (Shaw, 2001) and unscientific (Fenton, Pfleeger, & Glass, 1994) as the evaluative work is very rare in this field (Tichy, Lukowicz, Prechelt, & Heinz, 1995). Only a little work exists on systematic reviews in requirements engineering area. After a pre-review mapping it has been observed that there is no evidence of systematic review on requirements negotiation. Therefore, a need for review exists in this field of study.

2. Review Questions

1. What requirements negotiation practices are used to resolve requirements conflicts in software projects?
2. What are the reported strengths and weaknesses of existing requirements negotiation practices?
   - a) What are the reported strengths of existing requirements negotiation practices?
   - b) What are the reported weaknesses of existing requirements negotiation practices?

2.1. *Question Structure*

Petticrew and Roberts proposed a new method (using PICOC- Population, Intervention, Comparison, Outcome and Context) to form questions of systematic review (Beelmann, Petticrew, & Roberts, 2006).

Table 10: PICOCS of Review Questions

PICOCS	R.Q. 1	R.Q. 2a	R.Q. 2b
Population	Software projects		
Intervention	Requirements negotiation practices		
Comparison	-----------	--------------	------------
Outcome	Conflict resolution	Strengths of requirements negotiation practices	Weaknesses of requirements negotiation practices
Context	Academia, Industry, practitioners		
Study design	Primary studies		

3. Identification of Research

Identification of research includes the following activities:

- Source selection
- Search terms
- Documenting the search

3.1 *Source selection:*

The following electronic sources will be searched for the collection of evidence in literature because relevant papers exist in these databases and these are the major databases of software engineering (Barbara Kitchenham & Charters, 2007).

- IEEE
- ACM Digital library
- ScienceDirect (www.sciencedirect.com)
- SpringerLink
- EIcompendex (www.engineeringvillage2.org/Controller/Servlet/AthensService)

Other sources of evidence include the following material (Barbara Kitchenham & Charters, 2007):

- Reference lists from selected studies

- Conferences and journals of requirements engineering which are not included in the mentioned electronic databases.
- Domain experts and researchers working in the area will be contacted and asked them if they know of any unpublished work
- Research registers (e.g. Requirements Engineering bibliography by Alan Davis)

3.2 Search Terms

The search terms will be applied on the title and abstracts of papers. The following strategy is used to create search strings (B. Kitchenham, Mendes, & Travassos, 2006):

a) Identify major terms from the review question using PICO.

b) Use alternative terms, spellings or similar words of major terms.

c) Analyze the keywords mentioned in the relevant research papers.

d) Combine alternative spellings, terms or synonyms using Boolean OR operator.

e) Combine major strings using Boolean AND operator.

Results of a):

Table 11: Major Terms from PICO

PICO	*Major Terms*
Population:	Software projects
Intervention:	Requirements negotiation practices
Comparison	--------------- -----------------
Outcome:	Conflict resolution

Result of b): Synonyms of major terms and alternate spellings

Note: Bold and italicized word(s) extracted after performing part c.

Table 12: Synonyms of Major Terms & Alternate Spellings

Major term	*Synonyms*
Requirements negotiation	Requirements negotiation, ***conflict resolution***, conflict handling, requirements reconciliation
Practices	Technique, process, model, method, framework, approach, tool

Results of c): Keywords we already have in relevant papers:

- **P. Grunbacher & N. Seyff (2005)** Keywords: Negotiation process, ***conflict resolution***, collaboration, negotiation tools, stakeholder win-win (Grünbacher & Seyff, 2005).

Results of d) Combine alternative spellings, terms or synonyms using Boolean OR operator.

1 . Requirements negotiation OR conflict resolution OR conflict handling OR requirements reconciliation

2 . Technique OR process OR model OR method OR framework OR approach OR tool

f) **Result of e)** Combine major strings using Boolean AND operator.

Separate search strings have been designed for each data source to extract better results. The following is a general query.

General Query:

(Requirements negotiation OR conflict resolution OR conflict handling OR requirements reconciliation) **AND** (technique OR process OR model OR method OR framework OR approach OR tool)

3.3 Final Changes after Pilot Study Results

Earlier there were more major terms like *"software projects"* and *"benefits and limitations"*. These terms were deleted because the query became more complex and it was difficult to extract records using this query. When these terms were used in broad search, even then irrelevant results were obtained. Therefore, these terms were dropped. Keywords of many relevant papers were excluded as those keywords were irrelevant and unnecessary.

General query and database specific queries were designed and refined for each database. The most effective results were obtained after executing query including search terms i.e. *"requirements negotiation practices"* therefore this query was selected for final search. Initially, query was executed on Title and Abstract and then on Full text to maximize the search results.

As there were found numerous irrelevant papers when full text was selected, therefore it was finally decided to limit the search for *Abstracts* only. After data extraction from all the databases (*2395* papers were found), it has been seen that the terms *conflict resolution and conflict handling* were problematic. Due to these two terms numerous irrelevant papers i.e. *2243/2395* were entirely irrelevant. So the query was then slightly modified and now the terms *"conflict resolution"* and *"conflict handling"* are changed to ***"requirements conflict resolution"*** and ***"requirements conflict handling"***.

After applying these two terms in the query on the basis of TITLE and ABSTRACT, we found *151* papers and most of these papers were relevant. For more regressive search the second query was also applied to FULL TEXT. Now we found **948** papers from all the databases. These results covered more relevant papers than query applied to TITLE and ABSTRACT. So finally this query was used on the basis of FULL TEXT for data extraction.

The range of publication years covered in the systematic review is from 1968 to 2010. The reason for this constraint is because the concept of software engineering was proposed in 1968 at a conference held to discuss what was then called the "SOFTWARE CRISIS".

3.4 Documenting the Search

The documentation of search process is required to ensure the transparency of systematic review process so that it can replicable later on by the other authors (Barbara Kitchenham & Charters, 2007). The search process of this review will be documented in the following way:

Table 13: Search Process Documentation

Data source	**Documentation**
Digital library	Database name: IEEE Search strategy for the database: • Only English papers • Search criteria is metadata only Search date: 02-06-10 Years covered: 1968-2010
Conference proceedings	Proceedings' title Conference name (if different) Translation of title (if necessary) Name of journal (if published in journal)
Other sources	Searched/contacted date URL Any particular search condition

The individual paper record will be stored in an electronic database created in Microsoft Access.

Table 14: Individual Study Form

Study ID	
Paper Title	
Authors	
Proceeding	
Volume	
Issue	
Pages	
Relevance (title/abstract based relevance)	
Publication year	
Search Date	
Search Resource	
String id	

For recording results for individual search strings in each data source, another form will be used shown in Table 15.

Table 15: Individual String Form for Each Data Source

String Id:	
Search string:	
Data Source:	
Number of References:	
Author (who executes):	
Search Date:	

For recording the integrated results of all data sources, we developed a form that will maintain the count of total studies found in each data sources.

Table 16: Integrated Search Results

Basis of selection	No. of Papers found in each database					
	IEEE	ACM	EICOMPENDEX	SPRINGERLINK	SCIENCE DIRECT	Total
Abstract						
Full text						

4. Study Selection Criteria and Procedures

The following study selection criteria and procedures will be adopted.

4.1. Study Selection Process

The search strings will be used which are already identified. Primarily titles and abstracts of papers will be studied and then relevant articles will be identified. If the author will be doubted after reading title and abstracts then introduction and conclusion will also be studied.

If still article does not match the criteria then it will be excluded. The relevant/irrelevant studies will be selected/rejected and then consulted with the supervisor. A list of rejected papers will be maintained with reasons for rejecting the papers.

4.2. Inclusion Criteria

Articles will be included on the basis of reading their titles and abstracts. If the author will be in doubt then introduction and conclusion will also be studied **AND** only articles relevant to software requirements negotiation will be included **AND** only primary studies written in English language will be considered. **AND** only case/field studies, experiments and experience/industrial reports will be included for gathering evidence from literature.

4.3. Exclusion Criteria

The research thesis performed at various universities on the same topic and textbooks will be excluded **OR** if an article does not provide evidence, it will be excluded **OR** articles of *simple negotiation* will be excluded **OR** papers which are based on the expert opinion will be excluded **OR** Papers written other than English language will be excluded **OR** if a paper is published in several conferences or journals, the most complete version, on the basis of studies discussed in the article, will be included.

5. Study Quality Assessment Checklists and Procedures

A study quality assessment criterion is a detailed criterion for filtering the research papers further more. Each paper will be assessed after full reading. Here we have been designed separate quality assessment checklists for case/field studies, experience/industry reports and experiments.

Table 17: Quality Assessment Criteria (QAC) Checklist (B. Kitchenham, et al., 2006)

ID	QUALITY ASSESSMENT CRITERIA	FEEDBACK	SCORE
Common criteria for all papers			
1	Is the context of study clearly defined?	Y/N/P	
2	Did the study review the related work for the problem?	Y/N/P	
3	Are the findings systematically reported and sufficient evidence reported to justify the relationship between evidence and conclusion?	Y/N/P	
Criteria for case/field studies and experiments			
4	Are the objectives of study clearly defined?	Y/N/P	
5	Is the design of study clearly defined?	Y/N/P	
6	Is it clear which requirements negotiation practice is used in the study?	Y/N/P	
7	Is it clear which requirements conflict is resolved by the requirements negotiation practice?	Y/N/P	
8	Does the study clearly state the benefits & drawbacks of the requirements negotiation practice?	Y/N/P	
9	Does the study clearly define the nature of software projects?	Y/N/P	

The scale for evaluating the quality of all these questions will be:

- YES =1
- PARTIAL= 0.5
- NO =0

Table 18: Quality Assessment Form for an Individual Database

Conductor's name:	
Data source:	
Maximum quality score:	9.0
Study Id	**Quality Score**

5.1. *Study Quality Assessment Procedure*

When the studies pass the inclusion criteria, then a detailed quality assessment will be done on those selected studies. Now candidate primary studies will be obtained after applying a detailed quality assessment. Full reading of selected candidate primary studies will be required. After full reading of an article, the data will be extracted.

6. Data Extraction and Synthesis

The aim of data extraction is to extract necessary information from primary studies using a data extraction form. To avoid bias the data extraction forms will be piloted when the review protocol will be completed (Barbara Kitchenham & Charters, 2007).

6.1. *Required Data*

The information required from each primary study which is selected after applying quality criteria and inclusion criteria, has been mentioned in the data extraction forms (Barbara Kitchenham & Charters, 2007). The data will be extracted by one researcher and checked by another research or supervisor.

Table 19: Data Items Extracted from All Papers

Data item	Value
Name of data extractor	
Application domain(s)	
Study design	
Concepts covered by primary studies	
No. of relevant articles in all the databases	
No. of articles extracted from other sources	
Total no. of relevant articles	
No. of duplicate studies	
No. of studies left after duplicate studies	
No. of articles relevant to negotiation	
No. of articles left after applying study inclusion/exclusion criteria	
No. of articles left after applying quality assessment criteria	

Table 20: Data Items Extracted for Each Article (B. Kitchenham, et al., 2006)

Data item	Value	Additional notes
Title of article		
Year of publication		
Database name		
Type of study		Case study/field study/ experiment/ industry experience report
Aim of paper		
Is the research question mentioned clearly?		Yes/No
What is the issue?		

Which conflict is resolved?		Conflict reported and focused
Any other conflicts reported to be resolved.		
Major contribution		Significance/importance of study
Are the hypothesis clearly stated?		Yes/no/partially
Are the hypothesis supported well?		Yes/no/partially
Which requirements negotiation practice reported in the paper?		Practice means Technique/ method/ tool/ methodology/ framework
If more techniques reported then which is more appropriate?		
Names of other practices reported		
Are the claims valid?		Yes/no/partially
Are the assumptions valid?		Yes/no/partially
Is the evidence valid?		Yes/no/partially
Is the practice compared with other practices?		
What measures were used for comparing results?		Criteria for comparison
How the results are represented?		Graphical/tabular/descriptive
Are the results valid?		
Is the research question answered properly? if not then what is missing?		
Does the practice resolve conflict? If yes to what extent?		
Benefits of practice reported		
Limitations of practice reported		
Any other information not mentioned		
Future work		

6.2. *Data Extraction Process*

The data extraction form will be filled by the data extractor and then it is checked by the data checker. The data checker will be the supervisor or another researcher. The data checker will ensure the accuracy of the data extracted from the studies. If any disagreement exists then the form will be revised. Separate forms will be maintained for each study selected after study selection and quality assessment criteria.

6.3. *Data Synthesis*

The aim of data synthesis is to integrate and summarize the data collected from the studies. Data synthesis forms will be used to provide such kind of information. Table 21 provides a list of requirements negotiation practices reported in different papers:

Table 21: Data Synthesis Form Corresponding To R.Q.1

Study IDs	Quality score	Database name	Year of publication	Practice ID	List of practices
S1	1	IEEE	2007	P1	Winwin
------	-----	-------	--------	----------	-------

Table 22 provides cumulative information about the total no. of studies which reported each requirements negotiation practice. This table corresponds to R.Q.1.

Table 22: Counting Study Ids Covering Each Practice

Practice IDs	Study IDs that cover this practice	Total no. of study IDs
P1	S1,S2	2
--------	-----------	----------
	Grand Total	2

Table 23 provides cumulative information about the total number of studies which reported several benefits of requirements negotiation practices. This table corresponds to R.Q.2a.

Table 23: Counting Study Ids Covering Strengths of Practices

Practice ID	Benefit ID	Benefits of Practice	Study IDs that cover this benefit	Total no. of study IDs
P3	B1	Resolves personal conflicts	S1,S3,S5	3
-------	---------	--------	---------	--------
			Grand Total	3

Table 24 provides a cumulative knowledge about the total no. of studies which reported shortcomings of different requirements negotiation practices. This table corresponds to R.Q.2b.

Table 24: Counting Study Ids Covering Weaknesses of Practices

Practice ID	Limitation ID	Limitations of Practices	Study IDs that cover this limitation	Total no. of study IDs
P3	L1	Resolves personal conflicts	S1,S3,35	3
-------	---------	--------	---------	--------
			Grand Total	3

7. Data Analysis

Summary of each study's information will be represented in the tabular form in a particular order (the most recent studies will come first). The number of studies will also be counted.

Table 25: Layout of Answer of RQ.1

Practice id	*Name of practice*	*References*	*Conflict(s) addressed*

Table 26: Layout of Answer RQ.2a

Practice id	*Strengths of practice*	*References*

Table 27: Layout of Answer RQ. 2b

Practice id	*Weaknesses of practice*	*References*

The tables 25-27 will be reviewed to answer the research questions. We identify existing requirements negotiation practices and their pros and cons.

Question.1 What requirements negotiation practices are used to resolve requirements conflicts in software projects?

- This question will be answered by identifying a list of existing requirements negotiation practices found in the papers.
- The tables 19 and 20 in the data extraction will answer this question.

Question.2 What are the reported strengths and weaknesses of existing requirements negotiation practices?

- This question will be automatically answered by answering Question 2a and 2b.

Question.2a What are the reported strengths of existing requirements negotiation practices?

- List of strengths of each requirements negotiation practice will be reported while answering this question.
- This information will be provided by the table 23.

Question.2b What are the reported weaknesses of existing requirements negotiation practices?

- List of weaknesses of each requirements negotiation technique will be reported in the answer of this question.
- This information will be provided by table 24.

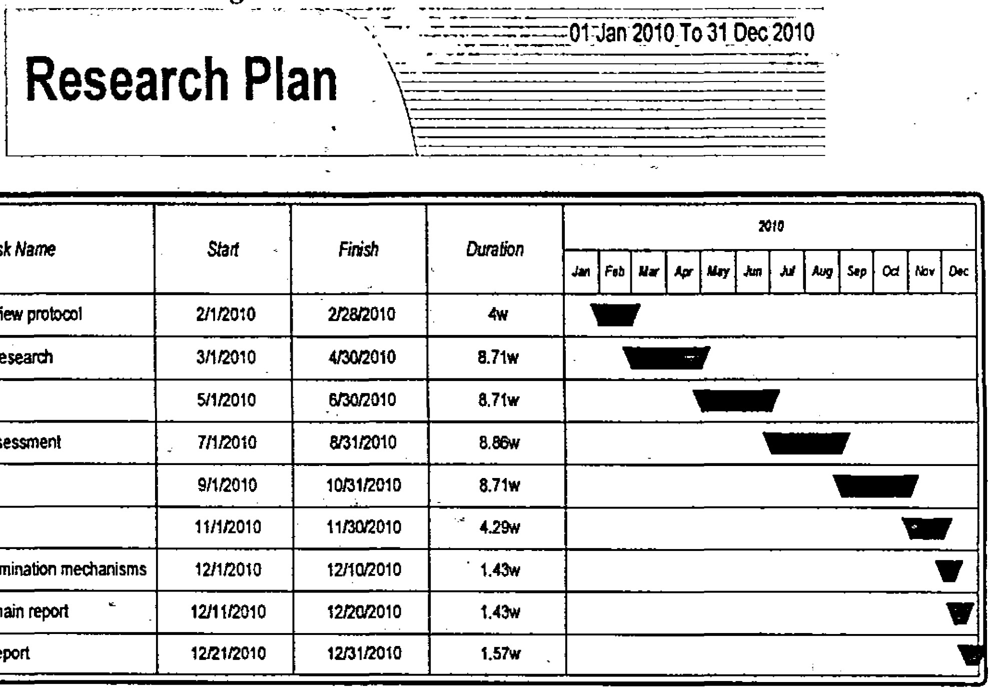
8. Dissemination Strategy

The results of systematic review will be convincing for both software engineering group and researchers. The results of systematic review will be written in a *"Technical Report"* format. The report will be reviewed by the internal SIG committee of

International Islamic University, Islamabad and external committee as well. A short version of report will also be sent to practitioners for their comments. We also try to publish it in the journal and conferences.

9. Project Time Table

Figure 8: Thesis Timeline



Research Plan

01 Jan 2010 To 31 Dec 2010

ID	Task Name	Start	Finish	Duration	2010 Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	Evaluation of review protocol	2/1/2010	2/28/2010	4w		■										
2	Identification of research	3/1/2010	4/30/2010	8.71w			■	■								
3	Study selection	5/1/2010	6/30/2010	8.71w					■	■						
4	Study quality assessment	7/1/2010	8/31/2010	8.86w							■	■				
5	Data extraction	9/1/2010	10/31/2010	8.71w									■	■		
6	Data synthesis	11/1/2010	11/30/2010	4.29w											■	
7	Specifying dissemination mechanisms	12/1/2010	12/10/2010	1.43w												■
8	Formatting the main report	12/11/2010	12/20/2010	1.43w												■
9	Evaluating the report	12/21/2010	12/31/2010	1.57w												■

REFERENCES

Beelmann, Andreas, Petticrew, M., & Roberts, H. (2006). Systematic reviews in the social sciences. A practical guide. *European Psychologist, 11*(3), 244-245.

Boehm, B., Bose, P., Horowitz, E., & Ming-June, Lee (1994, 18-22 Apr 1994). *Software requirements as negotiated win conditions.* Paper presented at the Requirements Engineering, 1994., Proceedings of the First International Conference on.

Boehm, B., & Egyed, A. (1998). WinWin Requirements Negotiation Processes: A Multi-Project Analysis. [Conference Proceeding].

Boehm, B., Port, D., & Al-Said, M. (2000). Avoiding the software model-clash spiderweb. *Computer, 33*(11), 120-122.

Fenton, N., Pfleeger, S. L., & Glass, R. L. (1994). Science and substance: a challenge to software engineers. *Software, IEEE, 11*(4), 86-95.

Grünbacher, Paul, & Seyff, Norbert (2005). Requirements Negotiation *Engineering and Managing Software Requirements* (pp. 143-162).

In, H., & Roy, S. (2001). *Visualization issues for software requirements negotiation*, Chicago, IL, United states.

Kitchenham, B., Mendes, E., & Travassos, G. (2006). Protocol for systematic review of within-and cross-company estimation models. *14*. Retrieved from http://www.dur.ac.uk/ebse

Kitchenham, Barbara, & Charters, S (2007). *Guidelines for performing systematic literature reviews in software engineering* (EBSE Technical Report No. EBSE-2007-01).

Shaw, Mary (2001). *The coming-of-age of software architecture research*. Paper presented at the Proceedings of the 23rd International Conference on Software Engineering.

Tichy, Walter F., Lukowicz, Paul, Prechelt, Lutz, & Heinz, Ernst A. (1995). Experimental evaluation in computer science: A quantitative study. *Journal of Systems and Software, 28*(1), 9-18.

APPENDIX B

INCLUDED PRIMARY STUDIES FOR SYSTEMATIC REVIEW

Included Primary Studies for Systematic Review

Boehm, Barry, Abi-Antoun, Marwan, Port, Dan, Kwan, Julie, & Lynch, Anne (1999). Requirements engineering, expectations management, and the two cultures. *Proceedings of the IEEE International Conference on Requirements Engineering*, 14-22.

Boehm, Barry, Bose, Prasanta, Horowitz, Ellis, & Lee, Ming June (1995). *Software requirements negotiation and renegotiation aids: a theory-W based spiral approach*, Seattle, WA, USA.

Boehm, B., & Egyed, A. (1998). WinWin Requirements Negotiation Processes: A Multi-Project Analysis. [Conference Proceeding].

Boehm, B., Egyed, A., Kwan, J., Port, D., Shah, A., & Madachy, R. (1998). Using the WinWin spiral model: a case study. *Computer, 31*(7), 33-44.

Boehm, Barry, Egyed, Alexander, Port, Dan, Shah, Archita, Kwan, Julie, & Madachy, Ray (1999). A stakeholder winwin approach to software engineering education. *Ann. Softw. Eng., 6*(1-4), 295-321.

Boehm, B., & In, H. (1996). Identifying quality-requirement conflicts. *Software, IEEE, 13*(2), 25-35.

Da, Yang, Di, Wu, Koolmanojwong, S., Winsor Brown, A., & Boehm, B. W. (2008). WikiWinWin: A Wiki Based System for Collaborative Requirements Negotiation. *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pp.24.

Grunbacher, P. (2000, 2000). *Collaborative requirements negotiation with EasyWinWin*. Paper presented at the Database and Expert Systems Applications, 2000. Proceedings. 11th International Workshop on.

Grunbacher, P., Egyed, A., & Medvidovic, N. (2001). *Reconciling software requirements and architectures: The CBSP approach*, Toronto, Ont, Canada.

Grünbacher, Paul, Seyff, Norbert, Briggs, Robert O., In, Hoh Peter, Kitapci, Hasan, & Port, Daniel (2007). Making every student a winner: The WinWin approach in software engineering education. *Journal of Systems and Software, 80*(8), 1191-1200.

Herrmann, Andrea, Paech, Barbara, & Plaza, Damian (2006). ICRAD: An integrated process for the solution of requirements conflicts and architectural design.

International Journal of Software Engineering and Knowledge Engineering, 16(6), 917-950.

Hoh, In, Boehm, B., Rodger, T., & Deutsch, M. (2001, 12-19 May 2001). *Applying WinWin to quality requirements: a case study.* Paper presented at the Software Engineering, 2001. ICSE 2001. Proceedings of the 23rd International Conference on.

Kitapci, Hasan, & Boehm, Barry W. (2007). *Formalizing informal stakeholder decisions - A hybrid method approach*, Big Island, HI, United states.

Linhares, Giovana, Borges, Marcos, & Antunes, Pedro (2009). *Negotiation-Collaboration in Formal Technical Reviews. Groupware: Design, Implementation, and Use*, 344-356.

Ramires, Joao, Antunes, Pedro, & Respicio, Ana (2005). *Software requirements negotiation using the software quality function deployment*, Porto de Galinhas, Brazil.

Seyff, Norbert, Hoyer, Christoph, Kroiher, Erich, & Grunbacher, Paul (2005). *Enhancing GSS-based requirements negotiation with distributed and mobile tools*, Linkoeping, Sweden.

Wu, Di, Yang, Da, & Boehm, Barry (2010, 5-8 Jan. 2010). *Finding Success in Rapid Collaborative Requirements Negotiation Using Wiki and Shaper.* Paper presented at the System Sciences (HICSS), 2010 43rd Hawaii International Conference on.

Wu, Di, Yang, Da, Koolmanojwong, Supannika, & Boehm, Barry W. (2009). *Experimental evaluation of wiki technology and the shaper role in rapid interdisciplinary requirements negotiation*, Waikoloa, HI, United states.

APPENDIX C
QUALITY ASSESSMENT CRITERIA RESULTS

Table 28: Quality Assessment Criteria Results for Each Study

QAC#	Quality Assessment Criteria	Study Ids																	
		9	10	20	25	32	34	39	43	54	56	58	62	68	69	72	73	76	77
1	Is the context of study clearly defined?	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	Did the study review the related work for the problem?	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	Are the findings systematically reported and sufficient evidence reported to justify the relationship between evidence and conclusion?	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	Are the objectives of study clearly defined?	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	Is the design of study clearly defined?	0	0	1	1	1	0.5	0	1	1	0	1	1	1	1	1	1	0.5	1
6	Is it clear which requirements negotiation practice is to be used in the study?	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	Is it clear which requirements conflicts is resolved by the requirements negotiation practice?	1	1	0.5	0.5	0	1	0.5	1	1	0	1	0	0	0	0	1	1	0
8	Does the study clearly state the benefits & drawbacks of the requirements negotiation practice?	0.5	1	1	1	0.5	1	1	0.5	0.5	1	1	1	1	1	1	1	1	1
9	Does the study clearly define the nature of software projects?	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Total		7.5	8.0	8.5	8.5	7.5	8.5	7.5	8.5	8.5	7.0	9.0	8.0	8.0	8.0	8.0	9.0	8.5	8.0

Scale: Yes=1, No=0 and Partial=0.5

Threshold = 1.5