

**IMPACT OF TEAM SKILLS ON SOFTWARE QUALITY: A  
COMPARATIVE STUDY OF SOFTWARE DEVELOPMENT  
TEAMS IN INDIA AND UNITED STATES**

A THESIS

*Submitted by*

**VIJI VINOD**

*for the award of the degree*

*Of*

**DOCTOR OF PHILOSOPHY  
IN  
COMPUTER APPLICATION**



**DEPARTMENT OF COMPUTER APPLICATION  
Dr. M.G. R. EDUCATIONAL AND  
RESEARCH INSTITUTE UNIVERSITY  
(Declared U/S 3 of the UGC Act, 1956)  
CHENNAI 600095**

**FEBRUARY 2010**



## **DECLARATION**

I declare that the thesis entitled ‘Impact of Team skills on software Quality: A comparative study of Software development Teams in India and United States ‘submitted by me for the award of the degree of Doctor of Philosophy under the guidance of Dr. Sunil Sahadev, Lecturer, Sheffield University, United Kingdom and have not formed the basis earlier for acquiring any degree, diploma, associateship, fellowship in this university or similar institution of higher learning.

**Viji Vinod**

## ABSTRACT

It is well known that software products typically fail to achieve their expected quality. While there are many studies on determining the quality of a software product in the software engineering literature on process, design complexity, software quality models, the quality of a software product is basically determined, by the people who are populated in the software team that developed the software (Richard 2007; Justin 2006; Evans 2004; Krishnan 1998; Boehm 2007). The skills of the software team determine the quality of the software product (Krishnan 1998). The technical skills of the software team and the project management skills of the software team determines the quality of the software developed (Judith 2007; Tschang 2001).

In the past, software development was mostly done in – house; mainly in the developed countries like United States and now, it has been extended to various locations. The problem of 2000's has been globalization of software development in different locations (Nicholson 2004; Herbsleb 2001). Many of the software projects are outsourced from the developed countries like the United States America who are already quite well advanced in the software development industry to the developing countries in the Far East like Singapore and India.

However, the team skills (in the context of software development) available in these developing countries and the main disparities between the team skills available in the developing and developed countries have not yet received sufficient research attention. Hence there is a need to study the impact of critical team skills on quality of the software product and assess its variation across different countries. Specifically, the objective of this study is to explore the critical skills on quality of software product and how these team skills impact software quality in the United States of America and India. During the course of this study we surveyed software development teams in India and in the United States and collected data about the impact of critical team based skills on software quality. The data collected from the two countries was compared using analysis of variance methodology to establish the differences between the two countries.

## ACKNOWLEDGEMENT

I sincerely submit my respectful regards to our **Chancellor Mr. A. C. Shanmugam**, Dr. M. G. R. Educational & Research Institute University, Chennai-95, with whose blessing this work has been completed successfully.

I am equally indebted to our **Pro-Chancellor A.C.S. Arun Kumar**, Dr. M. G. R. Educational & Research Institute University, Chennai- 95 for his generous patronage throughout the work.

I convey my special thanks to our **Vice President – Administration Dr. R. M. Vasagam**, for his valuable guidance.

I express my gratitude to our **Vice President – Academic Dr. P.T. Manoharan**, for his valuable advice.

I express my profound sense of gratitude with pleasure and pride to our **Vice Chancellor Dr. K. Padmanabhan**, for his inspiring and excellent guidance.

I convey my special thanks to our **Registrar Dr. S. Dinakaran**, Dr. M G R University, for their valuable guidance.

I am grateful to **Controller of Examination Dr. P. Sivakesan** and **Dean – E & T Dr. N. Senthil Velan**, for their valuable guidance.

I would like to express my heartfelt gratitude to our **Dean – Research Dr. P. Aravindan**, for his valuable suggestions throughout the study.

I deem it a great pleasure to place on record my deep sense of gratitude and indebtedness to my **Research Guide Dr. Sunil Sahadev, Lecturer, Shieffield University, United Kingdom** who has been a source of inspiration to me all through my work and continuous supervision at every stage of this work.

I sincerely thank **Dr. S.P Rajagopalan, Dean – Computer Applications**, for his expert comments in this study.

My sincere thanks to **Dr. Bhuvaneshwari** , **HOD - Department of computer Applications** and **Prof. F. W. Jesudas**, **HOD – Department of Information technology**, Dr. M G R University, for his invaluable help throughout this study.

I extend my gratitude to **Dr. R.M Suresh**, **Head of the department, Information Technology**, **R.M.K Engineering College**, Chennai for his guidance through out the study.

# TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

	CHAPTERS	PAGE NO
<b>1</b>	<b>INTRODUCTION</b>	<b>01</b>
<b>1.1</b>	<b>Academic Importance of the Topic Area</b>	<b>02</b>
	1.1.1 Globalization of software development	<b>05</b>
<b>1.2</b>	<b>Managerial Importance of the Topic Area</b>	<b>06</b>
<b>1.3</b>	<b>The Research context</b>	<b>07</b>
<b>1.4</b>	<b>Objective of the study</b>	<b>08</b>
	1.4.1 General research problem	<b>09</b>
	1.4.2 Research questions to be addressed	<b>09</b>
<b>1.5</b>	<b>Definitions</b>	<b>10</b>
<b>1.6</b>	<b>Conclusions and outline of the Thesis</b>	<b>11</b>
<b>2</b>	<b>LITERATURE REVIEW - Software Quality Determining factors</b>	<b>13</b>
	2.1 Introduction	<b>13</b>
	2.2 Software engineering and achieving software quality	<b>15</b>
	2.2.1 Software Development in Organizations: The need of high quality	<b>16</b>
	2.2.2 Definitions of Software Quality	<b>17</b>

<b>2.2.3</b>	<b>Models and frameworks for assessing quality.</b>	<b>20</b>
<b>2.2.3.1</b>	<b>Existing models for determining software quality</b>	<b>21</b>
<b>2.2.3.2</b>	<b>Existing frameworks for quality assessment.</b>	<b>27</b>
<b>2.2.3.3</b>	<b>Critique of existing quality models</b>	<b>31</b>
<b>2.2.3.4</b>	<b>Models and frameworks which incorporated software team in assessing quality.</b>	<b>32</b>
<b>2.3</b>	<b>Team skills determining software quality – The present study.</b>	<b>35</b>
<b>2.3.1</b>	<b>Technical skills for product development</b>	<b>40</b>
<b>2.3.1.1</b>	<b>Programming skill</b>	<b>41</b>
<b>2.3.1.2</b>	<b>Language skill</b>	<b>42</b>
<b>2.3.1.3</b>	<b>Hardware skill</b>	<b>42</b>
<b>2.3.1.4</b>	<b>Application Domain skill</b>	<b>43</b>
<b>2.3.2</b>	<b>Project management skills</b>	<b>44</b>
<b>2.3.2.1</b>	<b>Leadership</b>	<b>45</b>
<b>2.3.2.2</b>	<b>Communication</b>	<b>45</b>
<b>2.3.2.3</b>	<b>Motivation</b>	<b>46</b>
<b>2.3.3</b>	<b>Team Size</b>	<b>46</b>
<b>2.4</b>	<b>Summary</b>	<b>47</b>



<b>3</b>	<b>LITERATURE REVIEW – Globalization of Organizations and its impact on software development</b>	<b>48</b>
3.1	Introduction	48
3.2	Effects of Globalization	51
3.3	Outcomes of Globalization: Disparity on team skills in India and United States software development projects.	55
3.3.1	Differences in the type of projects developed in India and United States.	58
3.3.2	Migration of Experienced professionals to US	61
3.3.3	Cultural differences in India and United States	63
3.4	Summary Literature (Chapters 2 and 3)	74
<b>4</b>	<b>PROPOSITION DEVELOPMENT</b>	<b>77</b>
4.1	Introduction	77
4.2	Refinement of conceptual model incorporating multiple skills for team personnel.	78
4.3	Proposed relationships/disparity between the determinants of software quality in United States and India	80
4.3.1	Skills of the team personnel on software team’s quality differ in India and United States	81
4.3.1.1	Technical experiences of team members on software quality varies in India and US	82
4.3.1.2	Project management skills of team members on software quality varies in India and US	84
4.4	Summary	87

<b>5</b>	<b>RESEARCH METHODOLOGY</b>	<b>89</b>
5.1	Introduction	89
5.2	Research design	89
5.3	Data Collection	90
5.4	Main survey	91
5.5	Scale Development of construct Measures	93
5.6	Summary	95
<b>6</b>	<b>DATA ANALYSIS AND RESULTS</b>	<b>96</b>
6.1	Introduction	96
6.2	Sample Profile	96
6.2.1	Final Profile	96
6.2.2	Descriptive statistics	96
6.3	Construct validation	101
6.3.1	Dimensionality of constructs	101
6.4	Validity	103
6.5	Factor Analysis	104
6.6	Reliability Analysis	105
6.7	Analysis of variance	106
6.8	Summary	110
<b>7</b>	<b>DISCUSSION AND CONCLUSIONS</b>	<b>112</b>
7.1	Introduction	112
7.2	Primary research purpose	112
7.3	Discussion of Research Results	112
7.4	Managerial Implications	116
7.5	Summary	117
7.6	Contributions of the study	119
7.7	Conclusion	120
7.8	Limitations and directions for future research.	121

**APPENDICES**

**REFERENCES**

## LIST OF TABLES

- Table 2 – 1** Garvin’s views of quality
- Table 2 – 2** Existing models for determining software quality
- Table 2 – 3** Comparing quality models Boehm and McCall’s.
- Table 2 – 4** Comparison of existing software process frameworks
- Table 2 – 5** Overview of frameworks developed for a team’s quality assessment.
- Table 2 - 6** Team skills determining Quality
- Table 3 - 1** Employment in manufacturing sector
- Table 3 - 2** Number of US jobs moving offshore
- Table 3 - 3** Composition of Indian software development and services
- Table 3 - 4** Software Industry Growth
- Table 3 - 5** Indian software firms: descriptions for migration
- Table 3 – 6** Strodthbeck (1961) theories based on the ‘patterns of behavior
- Table 3 – 7** Cultural dimensions in India and United States
- Table 3 – 8** Laurent division of cultural dimensions in to societies: US verses India
- Table 3 – 9** Leadership behaviours and their acceptability from 60 countries
- Table 3 – 10** Dimensions of multicultural environment in different countries
- Table 3 – 11** Summary of skills require in India and United States
- Table 5 - 1** Research Design
- Table 5 - 2** Measures Tested in Pilot Studies
- Table 6 -1** Descriptive statistics for the sample in United States
- Table 6 – 2** Descriptive statistics for the Indian sample
- Table 6 - 3** Summary statistics of these variables in the United States and India
- Table 6 - 4** Exploratory Factor Analysis –summary
- Table 6 - 5** Factor analysis and scale reliabilities
- Table 6 - 6** ANOVA of the relative effect of team factors on software quality in terms of India and United States
- Table 6 - 7** Summary statistics of these variables.

## **LIST OF FIGURES**

- Figure 1 - 1 Initial conceptual model**
- Figure 2 - 1 Organization of the Literature review**
- Figure 2 – 2 Software engineering development strategy**
- Figure 2 – 3 McCall’s triangle of quality**
- Figure 2 – 4 Boehm’s software quality characteristics**
- Figure 2 – 5 Dormeys quality model.**
- Figure 2 - 6 ISO 9126 Quality Model**
- Figure 2 – 7 Maturity levels of CMM**
- Figure 3 – 1 Organization of Literature review**
- Figure 3 – 2 Effects of Globalization**
- Figure 3 – 3 Dimensions of Multicultural environment**
- Figure 4–1 Conceptual model – proposition development**

# CHAPTER 1

## INTRODUCTION

The major goal of software organizations during the last decade has been achievement of a high level of software quality especially in global software development environments. Both the academicians and practitioners in the software engineering field have contributed to the literature on this subject (for example; Pressmen 2005; Kan 2002; Evans 2004). Academicians have in particular actively researched the areas of software quality (for example; Galin 2004; Herbsleb 2001; Hyatt 1996).

The software engineering literature readily acknowledges the importance of people skills in the attainment of software quality in globalized software development environments (Carmel 2001; Castels 1994). Many studies agree that there is a significant relationship between software team skills and software quality (Krishnan 1998; Kekre 1995).

*.....report provided empirical evidence that development team skill has a significant effect on the quality of the software product. (Justin 2006: Page 5).*

However, there has been a shortage of work that specifically addresses the variations of team skills in the countries United States and India (DCosta 2002).

Accordingly, the main objective of this thesis has been to empirically examine and extend our knowledge of the impact of team skills on software quality. Specifically, the aim is to consider the skills of the team members that constitute the software development team. The thesis aim to examine the totality of the skills of team members that ultimately determines the software team's skill as a whole. Krishnan (1998) examines the impact of team factors in software development, such as the domain and language experience of the team members and the personnel capability of the team, on the costs and quality of the software products. Hence, a key objective of this research is to empirically test the effect of these team constructs on software quality in conjunction with other constructs identified in the literature that may influence software quality. Timgoles (2008) discussed on the critical skill sets for

quality software in the developing countries like Europe and United States who outsource the software projects and in developing countries like Singapore and India where the projects are outsourced. Thus the research also identifies the disparity among these team skills in software development projects in the United States and India.

Specifically, the independent variables of interest in this research study include the technical experience of the team as language experience, programming experience, application experience, hardware experience and project management skills as motivation, leadership and communication (see Figure 1-1, Initial conceptual model). The research context is set within the software developers in the United States and Indian project teams.

The aim of this chapter is first, to establish both the academic and managerial importance of the topic area. Second, the context in which the study will be undertaken is discussed. Third, the study objectives and key research questions are stated. Fourth, the definitions of the constructs under investigation are presented and finally, the remainder of this thesis will be outlined.

### **1.1 Academic Importance of the Topic Area**

It is well acknowledged that due to the unique characteristics of software development environments, software teams play a key role in shaping the expectations of the software product, managing and controlling software specifications, and ultimately in shaping the conformance to design requirements (Pressman 2005, Crosby 1979, Stevens 1994). In essence they represent a significant component in exploring the internal and external characteristics of the software product (Stevens 1994).

While software engineering has traditionally been concerned with making promises and building expectations of software quality, organizational behavior has often had the task of delivering these promises. More and more there is a realization that the two disciplines need to be integrated for optimal software quality to eventuate (for example, Pressman 2005). The organizational behavior literature has generally paid little attention to the quality of the product, where as software engineering literature

has largely ignored the team factors that affect quality (Justin 2006, Barry 2007, Diaz 1997, Haley 1996).

In recognition, this software engineering thesis focus on the software team as well as the skills essential for software developers in a software organization and attempts to delineate the team factors influencing the software quality. It thereby adds to the body of literature that discusses the importance and management of the software project teams in India and United States.

Pressman (2005) considers that in software quality management there were four factors for discourse: people, problem, and process.... *manager who forgets that software engineering work is a human endeavor will never have success in project management (Pressman 2005: 58)*. People /software developers constitute a software team. It is the area that this thesis addresses, using a testing/development approach (Krishnan 1998), which aims to understand conceptual relationships and derive insights related to the team skills on software quality.

Judith and Kate (2007) perceived that the body of research that addresses these issues had three perspectives: business development skills, product development skills and people management skills of a team that effect software quality. This thesis is concerned with the perspectives, product development skills and people management skills:

Despite the intuitive relationship between development team skill and software product quality in the software development, the body of software engineering research is surprisingly sparse in its coverage of the topic. While very small number of studies in the software engineering literature (Richard 2007; Robert 2006; Justin 2006; Evans, 2004; Krishnan 1998; Boehm 2007) have attempted to capture development team skill as a driving cause of the success of software projects, the primary focus of most software quality determining models is the analysis of design complexity measures as indicators of software quality (Beaver 2005; Marc 2007; Humphrey 2005; Basili 1996; Kan 2002; Kaner 2002; Stephen 2002; Khoshgoftaar 1992; Subramanyam 2003; Futrell 2001). In practice, the software development industry has focused on well defined and followed software processes as the key to

software quality and project success (Michael 2007; Baddoo 2003; Diaz 1997; Haley 1996). While it is difficult to argue that complexity and process are factors in the quality of a developed software product, there is a general failure to address what every software project manager knows all too well: the skills of the software development team is a driving factor in software product quality (Justin 2006).

The team factors such as personnel capability and experience, personal motivation, coordination and communication among team members are believed to be critical for project success in software development (Faraj 2000; Carmel 1998; Curtis 1988). The cultivation of motivated highly skilled software people has been discussed since the 1960's (Lethbridge 2002; Cougar 1980; Demacro 1987; Whitaker 1994). Organizations that achieve high levels of maturity in the people management area have a higher likelihood of implementing effective software engineering (Pressmen 2005). The major categories of potential improvements in quality of software are focused around technology, software development team, process, and product factors (Pressmen 2005). Specifically, the importance of the team factor in software projects has been identified in few studies on software projects (Boehm 1981; Carmel 2001, 1997). The size of the team itself is often argued to be an important factor (Brooks 1993; Carmel 2001). A recent survey of leading software organizations in Europe identified that the quality of people in software teams is one of the most important factors in improving productivity and quality in software projects (Blackburn 1996; Kemerer 1992) reports:

.....”If you were to give an experienced software project manager a proposal and schedule for a software development project, but allow her to dictate one other factor, what would she choose? The answer is most project managers would want to select the staff members to be included in the project team” (Justin 2006: Page 1)

..... “The most important ingredient that was successful on this project was having smart people.... very little else matters in my opinion...The most important thing you do for a project is selecting staff...The success of the software development organization is very much associated with the ability to recruit good people”(Curtis, 1988: Page 59)



James Bach (1995) insists that the quality of the people is the primary driver for software quality, and that too much industry focus has been on the process (Bach 1995). Bach argues that personal performance; the experience and abilities of an individual play an important part in the ultimate quality of their developed software. However, there is little empirical evidence to support this logical assumption. This research is aimed in this direction. The research first attempts to capture these factors determining the team's quality in a considerably larger group of new respondents, software developers, and to analyses the disparity of effectiveness of team skills in the globally located locations US and India.

### **1.1.1 Globalization of software development**

Software development is a globally sourced commodity (Hayes 2002). This has led to the migration of software development and maintenance operations to geographically distributed locations. Catherine (2005) examines that geographically dispersed teams have been prevailing in many service-based organizations and significantly effects team effectiveness (Bitner 1994). Today a large percentage of software outsourcing is now centered in the Far East and Asian countries like India and Malaysia. The number of organizations globalizing their software development continues to rise (McCabe 2006). Ultimately, this means that globally distributed software development will continue to have a significant impact on the software industry and world economy as a whole (Kettinger 1995). Bateman's (1990) study on the US teams in Microsoft and IBM, says, "no matter the size of an enterprise, or whether its goals are market leadership or the pursuit of pure knowledge, the enterprise will come to be defined by the quality and character of its many teams". The Indian firms mainly concentrate on early stages of development namely low-level design, coding in software development. The Indian software industry mainly specializes in the export of low-end software development services, competing primarily on cost and availability of software talent (Arora 1999, 2001; Murali 2002).

However, while many studies on relationships of team skills and software quality (Krishnan 1998) has been referred to by a considerable number of academics over the past decade (Broy 2002, Whitaker 2005), it would appear that there has been no

confirmatory empirical study of the proposed relationships; the product development and project management team skills on software quality in the countries US and India.

The key challenge facing IT software and service providers is identifying critical skill sets (Timgoles 2008). The software service providers place more emphasis on business domain and project management skills than on technical skills. Judith (2007) examined the Indian software industry team skills as: basic technical skills as coding and programming and language skills, soft management skills as leadership, motivation and communication capability (Bitner 1990).

Cost, timelessness and quality are primary drivers that will lead to intense competition for software work over the next few decades. The developed countries like United States and Western Europe have well established software industries. But developing countries in the Far East (e.g. Korea, Singapore) in Asia (India, China) and in eastern Europe, all offer a large pool of highly motivated, competently educated and relatively low cost professional (Pressmen 2005).

Feigenbaum and McCorduck (Feigenbaum 1983) authored an article on the impact of software information services on the United States and the world, state the following: The American computer industry has been innovative, vital, and successful. It creates value by transforming the brainpower of knowledge workers, with little consumption of energy and raw materials.

The intent of this study is to cover the gaps in the existing literature. In doing so, the research model addresses the team variables that affect the software quality in a team on countries US and India. Further, it addresses the product development skill of the software team and the project management skills. The next section gives you the managerial importance of the “team skills on software quality in the countries United States and India.

## **1.2 Managerial Importance of the Topic Area**

Software people are considered the highest cost element of software development organizations as well as the greatest asset. The average company spends about 4 to 5 percent of revenue on Information technology with those that are highly IT dependent

such as financial and telecommunications companies spending more than 10% on it (Robert 2006). In other words IT is now one of the largest corporate expenses. Past research has identified the importance of people who will populate the team and their skill levels (Bill 2001).

Examples of the possible areas of use for the research in this thesis include:

- Selection and recruitment of software employees, since the constructs investigated can be measured by Human Resources divisions among potential personnel. For example, given a better understanding of people factors that impact on the quality of the software product among candidates could be incorporated in to screening procedures.
- Development and enhancement of training program will give the management / software developers, given a better understanding of the effect of team skills on software quality
- Manpower optimization in various department teams of the software organization for the manager.
- Future formula for reengineering of organization's team processes, by better understanding the influence of critical skill sets for a team.

In summary, the skill of software development team is addressed (Tingoles 2008). Similarly a more detailed understanding of influence of team factors on software quality in India and in United States would assist the management in the better develop their organization processes in a global software development environment.

Further, forthcoming section will explain the research context, objective of the study, the research problems and the research questions to be addressed

### **1.3 The Research context**

Since the conceptual model of the thesis is concerned with the team skills associated with the software quality in software development teams, it was necessary to conduct the research among the software development teams that were appropriate to this

characterization. Many research studies in this area considered the significance team skills in determining software quality of which Krishnan's (1998) "role of team factors on software quality and cost" involved domain skills and language skill of a team member that were considered to be higher for a higher quality software product. Higher personality skill of the team member showed a higher quality of the software product. Notably, Krishnan (1998) described the respondents of software developers in packaged software teams, and Arora (2006) extensively discussed the software development market globally located in the United States and service projects outsourced to Far East countries mainly in India.

Therefore, following the direction of Krishnan (1998) and Arora (2006), this research chose the setting of the software project organizations and the population of software developers in the teams to investigate the proposition's raised by the research inquiry. Software developers in India and United States were considered to be most appropriate survey respondents due to the considerable role they occupy.

Over the years software developers have been the subject of various forms of research investigation (Judith 2007, Justin 2006) .The research in this thesis was conducted with a cross-section of software organizations undergoing projects in India and United States both domestic and exports. Due to an historical employment policy, a considerable number of software employees are male, rather than the often-stereotypical female complement. The focus of this study will thus be build on the existing literature related to software development project teams in general, and particular software developers.

#### **1.4 Objective of the study**

The preceding discussion has sought to demonstrate the importance of understanding the nature of the relationships between team factors that influence software quality in software project teams in general. Further, it has been argued that the critical team skills on quality of software product on software quality and how these team skills impact software quality in countries India and United States warrant further investigation, both to academic researchers and practitioners. More specifically the study attempts to address the following research problem and research questions.

### **1.4.1 General research problem**

The goal of this research is to empirically examine and extend our knowledge of the determinants of team factors on software quality in a software development team; specifically, those who work in software organizations where the team status is generally considered significant. Importantly, this research has undertaken the team skills and their impact on quality from the perspective of software in different countries globally located like Europe and United States who are already achieved in the software development industry to the developing countries in the Far East like Singapore and India. A key aspect of the thesis is to find the critical team skills on quality of software product and the extent of the relationship of team's skill in India and United States.

### **1.4.2 Research questions to be addressed**

1. Whether the strength and the direction of the relationship between the domain experience/skills of team on quality of the software product the same in India as well as in U.S. software teams?
2. Whether the strength and the direction of relationship between the language skills of team on quality of software product the same in India and United States?
3. Whether the strength and the direction of relationship between the programming skills of team member on quality of software product the same in Indian and U.S teams?
4. Whether the strength and the direction of relationship between the hardware skills of team member on quality of software product the same in Indian as well as U.S teams?
5. Whether the strength and the direction of relationship between the motivational skills of team member on quality same in Indian and U.S software teams?

6. Whether the direction of relationship between the leadership skills of team member on quality the same in Indian as well as U.S teams?
7. Whether the degree of relationship between the communication skills of team member and quality of software product the same in India as in United States?
8. What is the degree of relationship between the size of a team and quality of software product stronger in India than in U.S teams?

### **1.5 Definitions**

The following definitions have been used for the variables under investigation in the conceptual model. A more detailed explanation of these variables is reported in the literature review Chapters 2 and 3.

Software quality is defined as: “Conformance to requirements or program specification “, the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use; (Nielsen 1993).

**Product development skills are defined as: The technical skills of a team member on software product development.**

Project Management skills are defined as: The managerial skills leadership, motivation and communication of a team member on software development

Team Size is defined as the number of members in the team. Team size and composition affect the team processes and outcomes. The optimal size (and composition) of teams is debated and will vary depending on the task at hand. At least one study of problem solving in groups showed an optimal size of groups at four members Ken (2005). Other works estimate the optimal size between 5-12 members. Less than 5 members results in decreased perspectives and diminished creativity. Membership in excess of 12 results in increased conflict and greater potential of sub-groups forming.

## 1.6 Conclusions and outline of the Thesis

This chapter has sought to establish the academics and managerial importance of the topic area as well as the outline the fundamental research problem and eight specific research questions, which are the focus of the study. Following this chapter, chapters 2 and 3 will provide a review of the literature germane to this research and those empirical and theoretical works relevant to this study. The purpose of these chapters is to provide the background and historical support for this study. Chapter 2 will discuss the body of research that this study is located within and the pertinent literature concerning the variable, software quality team. Further the various team skills that determine the quality of software are discussed.

Chapter 3 will review the relevant literature pertaining to the discussion globalization of Software development and how the teams in Indian

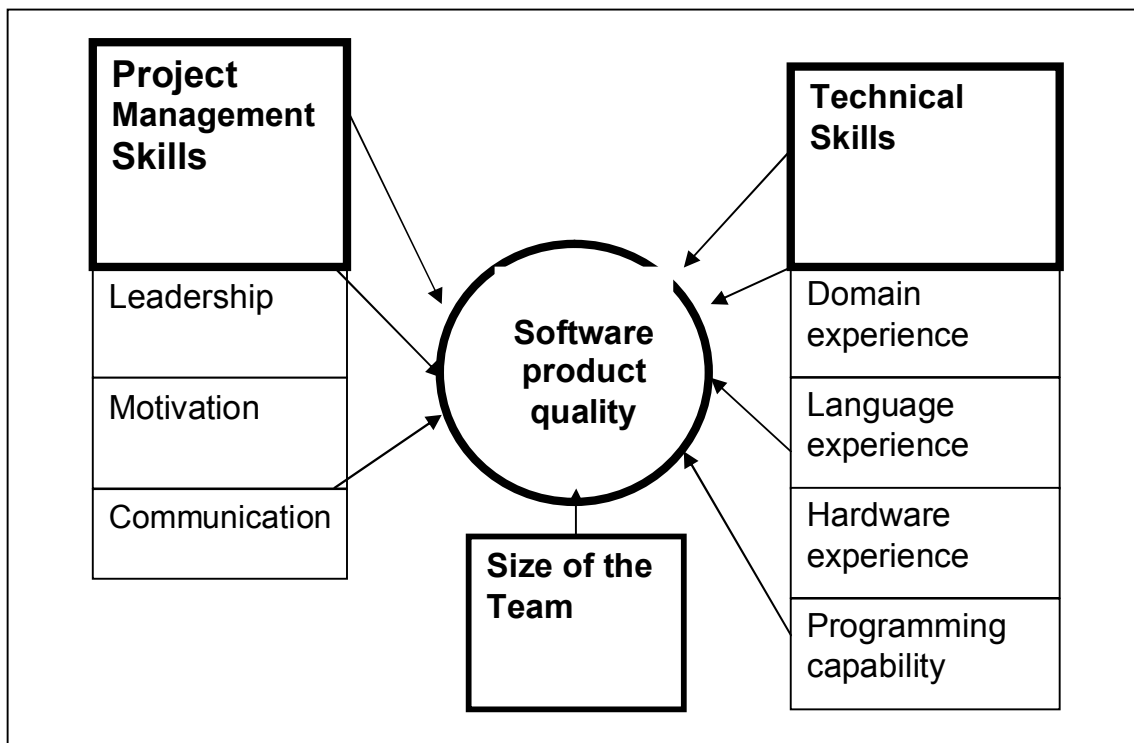
The reasons for varying team skills in the software projects developed are dealt in this chapter.

Following the literature review, Chapter 4 will develop the specific propositions to be tested. Chapter 5 will discuss the methodology to be used, including the research description of the organizations. Also, the research design is outlined, which encompasses data collection, pre testing and scale development and the research instrument. Finally the operational definitions of the constructs are described.

Chapters 6 will then present the research findings. First, the descriptive statistics of the study and details of the construct validation, which includes factor analysis. Further, the report of the results of the proposition testing by way of data analysis using Analysis of variance is discussed in this chapter.

**Chapter 7 presents a summary and discussion of the research results along with the managerial implications of the research. The contributions of the research will be outlined and limitations of the research addressed. Finally, the possible future research directions will be considered.**

**Figure 1 - 1 Initial Conceptual Model**





## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **SOFTWARE QUALITY DETERMINING FACTORS**

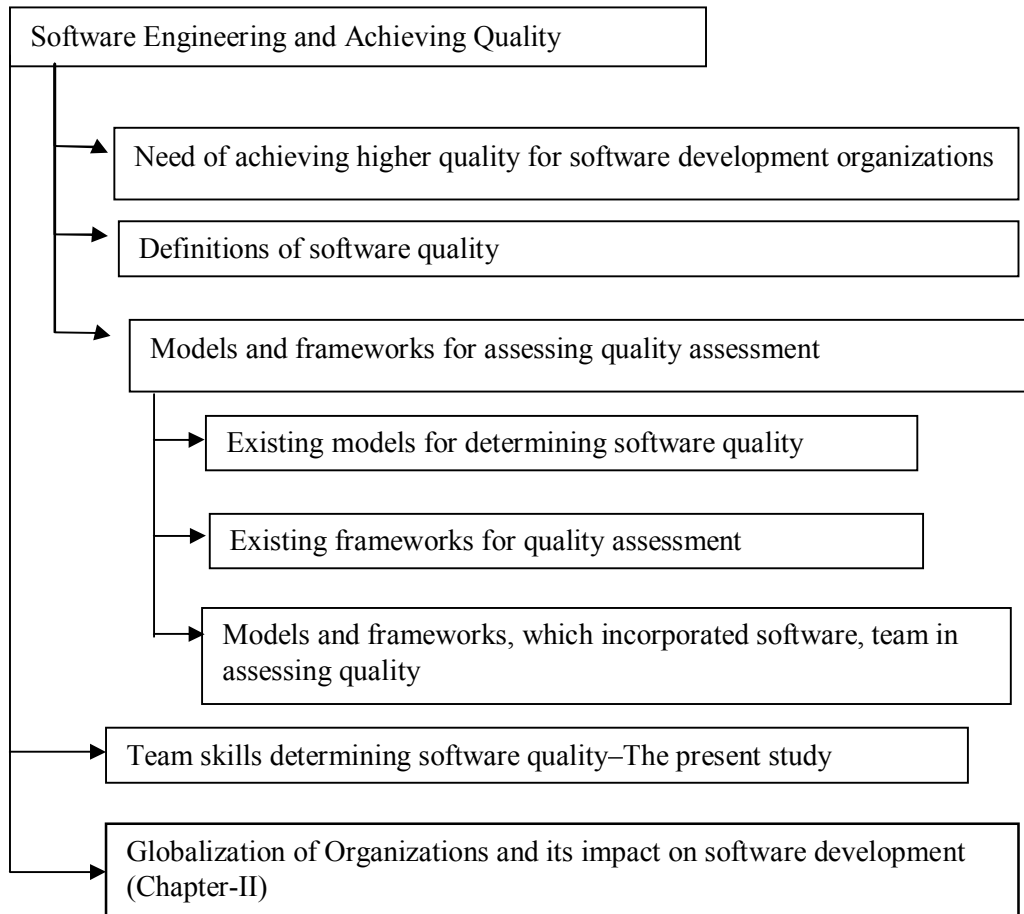
##### **2.1 Introduction**

The overall organization of literature review is as follows: Taking into consideration the two main domains in which the research falls, the entire literature review is divided into two sections: (i) Software quality determining factors and (ii) Globalization of Organizations. Chapter 2 discusses on the first section and Chapter 3 discusses on the second section.

In order to bring order and logic to the review, the first section is subdivided into the following sections:

- a) Need of achieving higher quality for software development organizations
- b) Definitions of software quality
- c) Models and frameworks for assessing quality.
- d) Team skills determining software quality – the present study.

The section c is again subdivided in to a) Existing models and frame works for determining software quality b) Existing frameworks for quality assessment c) Models and frameworks which incorporated software team in assessing quality. The section d is further divided in to a) Technical skills of the software team and b) Project management skills of the software team. The organization of this literature review is shown in Fig.2.1

**Figure 2 - 1 Organization of the Literature review.**

## 2.2 Software engineering and achieving software quality

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software (Alian 2004). It is a field of study dedicated for creating software that is of higher quality, more affordable, maintainable, and quicker to build (Naur 2008).

Software development, a much used and more generic term, does not necessarily subsume the engineering paradigm, but it is also the set of activities that results in software products. Software development may include research, new development, modification, reuse, reengineering, maintenance or any other activities that results in software products (McCarthy 1995). Especially the first phase in the software development may include marketing, research, engineering and general management (Conde 2002). The field's future looks bright according to Money Magazine and Salary.com who rated "software engineering" as the best job in the United States in 2006 (Kalwarski 2006).

The major category of potential improvements in quality of software is focused around technology, process and product factors (Pressmen 2005; Banker 1998). The primary focuses on studies in these areas are the analysis of design complexity measures as indicators of software quality. (Beaver 2005; Marc 2007; Humphrey 2005; Basili 1996; Kan 2002; Kaner 2002; Stephen 2002; Khoshgoftaar 1992; Subramanyam 2003; Futrell 2001). The software development industry has focused on well-defined software processes as the key to software quality and project success (Michael 2007; Baddoo 2003; Diaz 1997; Haley 1996). Bach (1999) insists that too much industry focus has been on the process and very limited studies are on the quality of the people.

The research area of the thesis lies within the field of software development, which highlights the importance of people team skills on quality of the software product. Software engineering also draws on concepts from organizational behavior in order to better understand the engineering strategies that incorporate the utilization of software

team as the quality criteria of software product that can be developed and auctioned (Evans 2004).

The following section discusses on the need of quality software's developed in software development organizations. Further, the section deals with the definitions of quality and the existing models and frameworks developed and used based on these definitions. Ultimately, various team skills determining software quality which is the topic of our thesis is discussed.

### **2.2.1 Software Development in Organizations: The need of high quality**

The cost of failed IT projects in the United States was recently estimated at \$84 billion in just 1 year (Smith 2001). So the software quality matters more now than it ever has, and it matters to you and me because we use that software for more tasks than ever before in Information technology, Information systems for businesses and embedded systems in consumer goods such as phones and of course across the internet (Sol 2001). The risks associated with software failure have increased with the use of software; these include greater exposure of organizations when software fails or is not satisfactory, and greater disappointment or loss for individuals when they are let down by software.

**Meanwhile modern organizations are facing increasing pressures on various fronts. For instance, the importance of time to market, cost reduction, value for money, increased expectation and knowledge of customers, global communication, constant change, and the need to find new markets-becomes pressures on software teams to produce more software, more quickly worth increased expectation of what that software can deliver as benefits (Brooks 1993).**

Software organizations fail to achieve higher quality in software development due to the reason that the software is build by people, and people make mistakes (Evans 2004; Deming 1988). This is true for any human activity, but in software development we have a number of exacerbatng circumstances:

- Software is often developed by teams of people other than those who will use them. In these circumstances any poor skills of people in software teams increases the chances poor quality.
- Software development is challenging and it needs skills that are innovative and changing, and so are working on continuous learning curve. Both the suppliers and users of software rarely have time to consolidate knowledge before they face yet more change.
- Software development organizations are notorious for their failure to align the software they produce with the culture, processes, or objectives of the business for which the software is intended.
- Software development systems are complex, and are becoming increasingly so in themselves and in their intercommunications with other systems.

**Only one of these is a technical problem (last), yet most of the emphases for software quality in organizations are seeking improvement in technical processes. All the other points are to do with the people, their ability to learn from each other and from experience. Hence we need a study in software engineering and development that addresses the issues of software quality through an emphasis on team's skills, their organizations on team's technical skills and project management skills. This thesis is intended to achieve this goal. To develop this study of achieving software quality, the literature review begins with the various definitions of authors on software quality.**

### **2.2.2 Definitions of Software Quality**

**To progress further, it is important to define software quality. However software quality is a multi-faceted construct and we therefore look at the important ways in which software quality has been defined. The software engineering literature has five definitions for quality (Garvin 1984). These definitions can be arranged based on the core-area they address: Typically they can be considered as: The definitions will use one of the forms, explained in Table 2 – 1.**

(1) Product based

(2) Manufacturing based

(3) User based

(4) Value based

(5) Transcendent definitions (Evans 2002; Garvin 1984; Trienekens 1997)

**Table 2 – 1 Garvin’s views of quality (adapted from Garvin 1984)**

Quality view	Quality criteria	Quality definition
Product based	Characteristics of subparts	Quality of delivered product
Manufacturing based	Ability to manufacture a product	Conformance to original requirements
User based	User and purchaser	Fitness to use
Value based	Budget, time and cost	Value for money
Transcendent	Non quantifiable	Customer expectations

**In the product based definition, quality is based on a well-developed set of software quality attributes that must be measured in an objective and quantitative way and derive acceptance criteria to objectively assess the quality of the delivered product. Product view on quality usually specifies that the characteristics of product be defined by the characteristics of its subparts, e.g. size, complexity, and test coverage. Module complexity measures, Design & code measures etc.**

The manufacturing based definition focuses on the manufacture of software products that is their specification and construction. Quality depends on the extent to which requirements have been implemented in conformance with the original requirements. We measure faults and failures in products. Success is measured by the ability to follow a process and deliver products against agreed specifications. Manufacturing view on quality focuses on conformance to specification and the organizations capacity to produce software according to the software process. Here product quality is achieved through process quality. Waste reduction, Zero defect (defect count and fault rates, staff effort rework costs) are concepts usually found within this view.

The user based and value based definition of quality reflects the views of software user and purchaser. These perspectives are about supporting the needs of the organization and its stakeholders within the organizations constraints. Because the

pressures on an organization change over time, what constitutes quality may change over time to match.

The user-based definition says that quality is fitness for use. User view on quality or “fitness for purpose” takes the starting point in software that meets the users’ needs; Reliability (failure rate, MTBF), Performance/Efficiency (time to perform a task), Maintainability and Usability are issues within this view. Software quality should be determined by the users of a product in a specific business situation. Different business characteristics require different types of software products; not only to do different things but also to cater to how different people want to carry out their tasks. This can be subjective and cannot be determined only on the basis of quantitative metrics. It is the user based definition that encourages us to validate as well as to verify the system. Example: fit for purpose.

Value based view on quality measures and produces value for money by balancing requirements, budget and time, cost & price, deliver dates (lead time, calendar time), productivity etc. The value-based definition is focused on things that impact on the running of the business as a whole. Software quality should always be determined by means of a division process on trade offs between time, effort, and cost aspects. This is done by communications with all parties involved –customers, developers and producers.

In transcendental view, where quality is recognized but not defined. The transcendental view is a subjective and non quantifiable of defining software quality. It often results in software that transcends customer expectations. According to the transcendental view quality is synonymous with “innate excellence” (Tuchman 1980). It is both absolute and recognizable a mark of uncompromising standards and high achievement. Nevertheless, proponents of this view claim that quality cannot be defined precisely; rather it is simple, un-analyzable property that we learn to recognize only through experience.

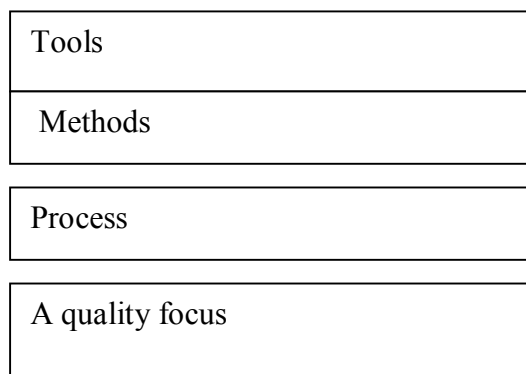
A software product is developed in higher quality, is accomplished with the support of various software development methodologies and process models in-order to attain

higher quality software product (Pralhad 1999). The next section deals with the various methodologies /frameworks and software development process models.

### 2.2.3 Models and frameworks for assessing quality.

A software engineering process model is a strategy or a software engineering paradigm, which helps to solve the actual problems in an industry setting (Humphrey 1989). A software engineer or a team of engineers must incorporate a development strategy that encompasses the process methods and tools layers described in Figure 2 -2.

**Figure 2–2 Software engineering development strategy (adapted from Pressmen 2005).**



The term “software engineering tool” can refer to the software used for the automatic development of systems software, i.e., computer code. The functions include analysis, design, and programming. These automated tools automate methods for designing, documenting, and producing structured computer code in the desired programming language (Telecom Glossary 2000).

Several mmethods and mathematical approaches are used to solve software (and hardware) problems at the requirements, specification and design levels. Examples of formal methods include the B-Method, Petri nets, Automated theorem proving, RAISE and VDM. Various formal specification notations are available, such as the Z notation.



A software process is the roadmap to building high quality software product. Software processes are adapted to meet the needs of the software engineers and managers as they undertake the software development of a software product. It provides a framework for managing activities that can very easily get out of control. Different projects need different software processes. The software engineers work products are produced as consequences of activities defined by the software process. The best indicators of how well a software process has worked are the quality, timeless and long term viability of the resulting software product (Pressmen 2005; Richard 2005). In the context of software engineering, a software quality measures how well software is designed (quality of design), and how well the software conforms to that design (quality of conformance) (Pressmen 2005).

In our study, we need to conceptualize a model, which can achieve high quality of software. In order to achieve this, the next section of the literature review will give an overview of the existing models, which are currently used by the software organizations for achieving higher quality of software product.

### 2.2.3.1 Existing models for determining software quality

One of the purposes of a software process model is to determine software quality. Software process model is defined as a strategy or a software engineering paradigm, which helps to solve the actual problems in an industry setting. We have dealt here with four software process models (Table 2 – 2).

- a) McCall's
- b) Boehm's
- c) Grady Booch's.
- d) Dormey's - Furps

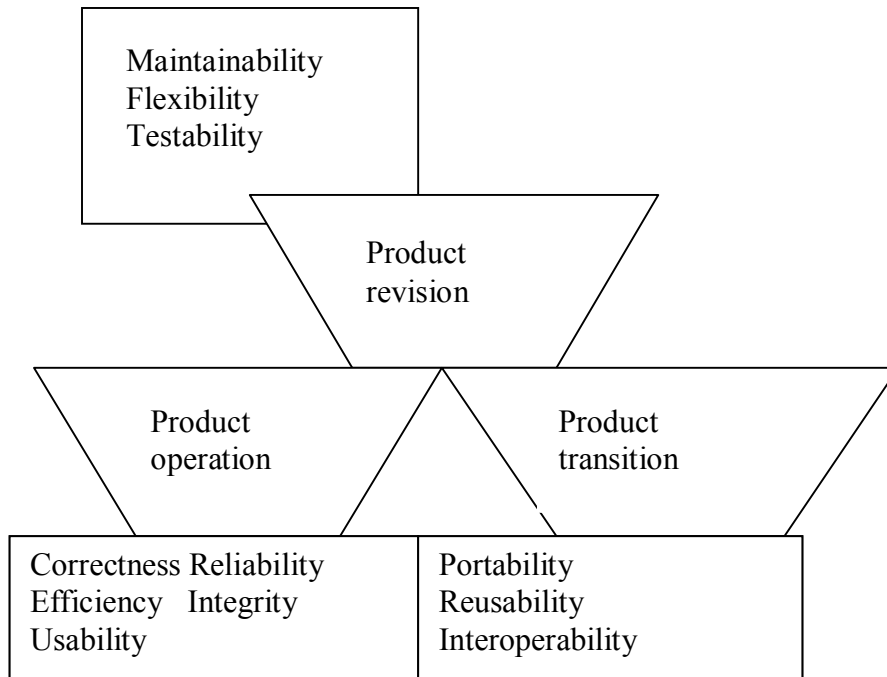
**Table 2 – 2 Existing models for determining software quality**

Quality models	McCall's	Boehm's	Grady Booch	Dormey's – Furps
Year	1977	1977	1992	1996

Quality perspective	Product revision, transition, operation	Higher level, lower level	Human factors, aesthetics, consistency, Reliability	Concrete relationships, setup attributes
Involvement of the variable team's technical skill	No	No	No	No
Involvement of variable -team's project management skill	No	No	Yes	Yes

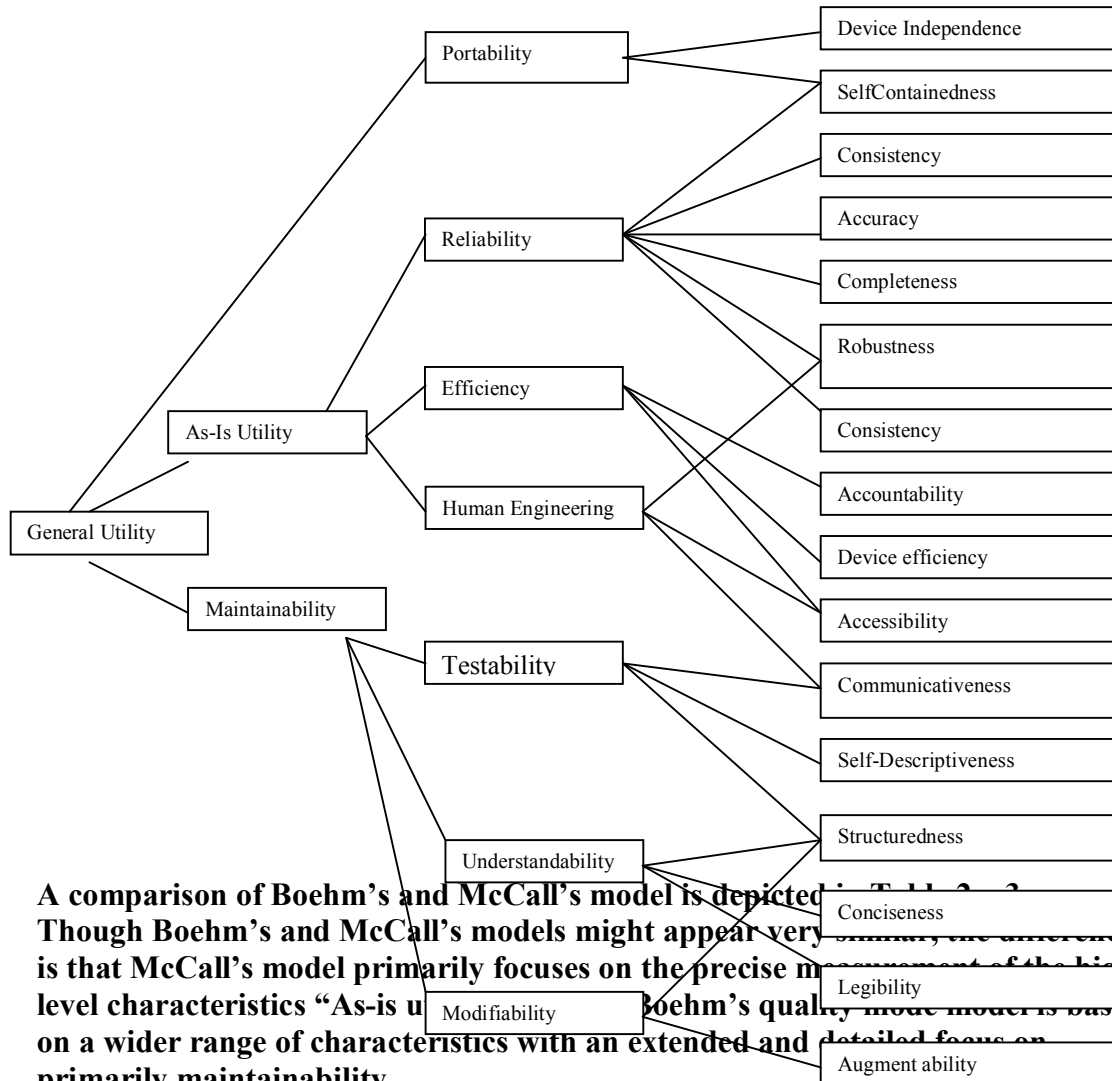
McCall's quality model (1977) has three major perspectives for defining and identifying the quality of a software product: product revision (ability to undergo changes), product transition (adaptability to new environments) and product operations (its operation characteristics). Quality of product operations depends on correctness (the extent to which a program fulfils its specification), reliability (the system's ability not to fail), efficiency (further categorized into execution efficiency and storage efficiency and generally meaning the use of resources, e.g. processor time, storage), integrity (the protection of the program from unauthorized access) and usability (the ease of the software). McCall's Quality Model has a hierarchy of 11 quality factors and related to 23 quality criteria. This is explained in McCall's triangle of quality (Figure 2 – 3).

**Figure 2 – 3 McCall’s triangle of quality (adapted from McCall 1977).**



Boehm (1977) addresses the contemporary shortcomings of models that automatically and quantitatively evaluate the quality of software. In essence his models attempts to qualitatively define software quality by a given set of attributes and metrics. The high-level characteristics represent basic high-level requirements of actual use to which evaluation of software quality could be put – the general utility of software. The lowest level structure of the characteristics hierarchy in Boehm’s model is the primitive characteristics metrics hierarchy. The primitive characteristics provide the foundation for defining qualities metrics – which was one of the goals when Boehm constructed his quality model. Figure 2 – 4 explain Boehm’s model.

**Figure 2 – 4 Boehm’s software quality characteristics (adapted from Boehm 1978)**



A comparison of Boehm’s and McCall’s model is depicted in Table 2-3. Though Boehm’s and McCall’s models might appear very similar, the difference is that McCall’s model primarily focuses on the precise measurement of the high-level characteristics “As-is utility”. Boehm’s quality model is based on a wider range of characteristics with an extended and detailed focus on primarily maintainability.

**Table 2 – 3 Comparing quality models Boehm and McCall’s (adapted from Garvin 1984).**

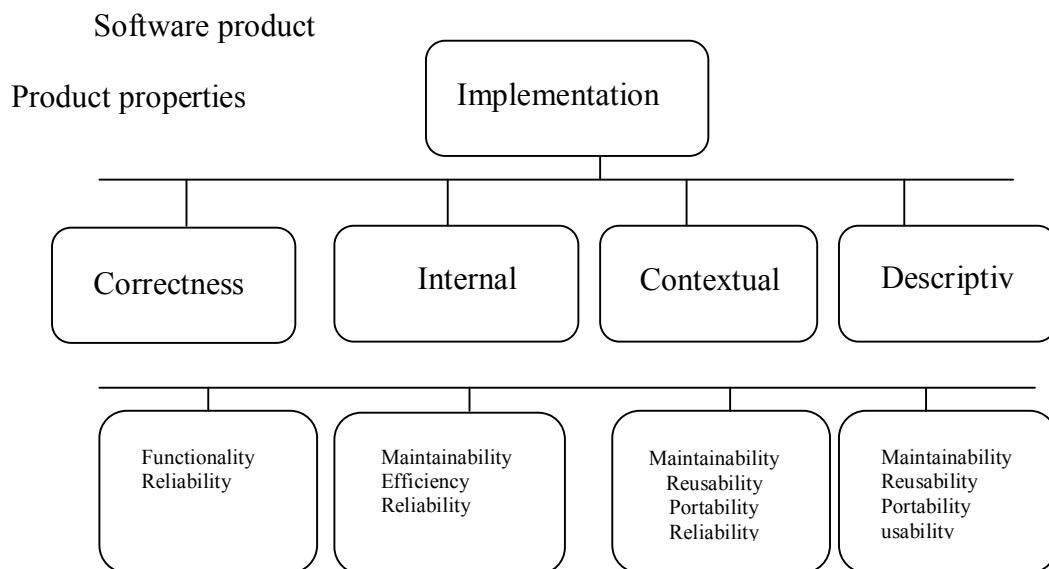
Criteria goals	McCall’s 1977	Boehm 1978	ISO 9126
Correctness	*	*	Maintainability

Reliability	*	*	*
Integrity	*	*	
Usability	*	*	*
Efficiency	*	*	*
Maintainability	*	*	*
Testability	*		Maintainability
Interoperability	*		
Flexibility	*	*	
Reusability	*	*	
Portability	*	*	*
Clarity		*	
Modifiability		*	Maintainability
Documentation		*	*
Resilience	*	*	
Understandability	*	*	
Validity	*	*	
Functionality	*		
Generality	*	*	
Economy	*	*	

**Grady (1992) FURPS model originally presented by Robert Grady defines the quality characteristics Functionality – which may include feature sets, capabilities and security, Usability - which may include human factors, aesthetics, consistency in the user interface, online and context sensitive help, wizards and agents, user documentation, and training materials, Reliability - which may include frequency and severity of failure, recoverability, predictability, accuracy, and mean time between failure (MTBF), Performance - imposes conditions on functional requirements such as speed, efficiency, availability, accuracy, throughput, response time, recovery time, and resource usage, Supportability - which may include testability, extensibility, adaptability, maintainability, compatibility, configurability, serviceability, installability, localizability (internationalization) Functionality, Usability, Reliability, Performance, Supportability, configurability, serviceability, instability, localizability Jacobson(1999) and Krutcehn (2000) extended the FURPS model in to rational software. IBM website (2003) has extended the model again in to Furp 3+.**

An even more recent model similar to the McCall's, Boehm's and the FURPS (+) quality model is the quality model presented by Dromey (1995). Dromey (1996) proposes a product based quality model that recognizes that quality evaluation differs for each product and that a more dynamic idea for modeling the process is needed to be wide enough to apply for different systems. Dromey is focusing on the relationship between the quality attributes and the sub-attributes, as well as attempting to connect software product properties with software quality attributes. The Dromey's quality model is explained in Figure 2 – 5.

**Figure 2 – 5 Dromey's quality model (adapted from Dromey 1996).**



Based on these quality models, various frameworks are developed for assessing quality in software organizations. A framework explains the methodology used for software development. The next section is a review of software process frameworks in the existing literature.

### **2.2.3.2 Existing frameworks for quality assessment.**

A software development framework is used to structure, plan, and control the process of developing information systems. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. One system development methodology is not necessarily suitable for use by all projects. Each of

the available methodologies is best suited to specific kinds of projects, based on various technical, organizational, project and team considerations.

The next section looks at the various process frame works presently being used for software development in organizations to achieve higher quality of resultant software product. The various frameworks are

1. ISO 9001
2. ISO 9126
3. ISO/IEC 15504

The Table 2 - 4 gives a description of these process frameworks.

**Table 2 – 4 Comparison of existing software process frameworks**

<b>Quality frameworks</b>	<b>ISO 9000</b>	<b>ISO 9126</b>	<b>ISO IEC 15504</b>	<b>IEEE</b>	<b>CMMI</b>
Approach	Process oriented	Process oriented	Process oriented	Process oriented	Process
Quality perspective	Design, Document, Implement Support Monitor Control Improve	Functionality	Process reference, model, Qualification, Supplier, process capability	Primary, Supporting, Problem resolution	Optimistic quantitative defined, Managed, Incomplete

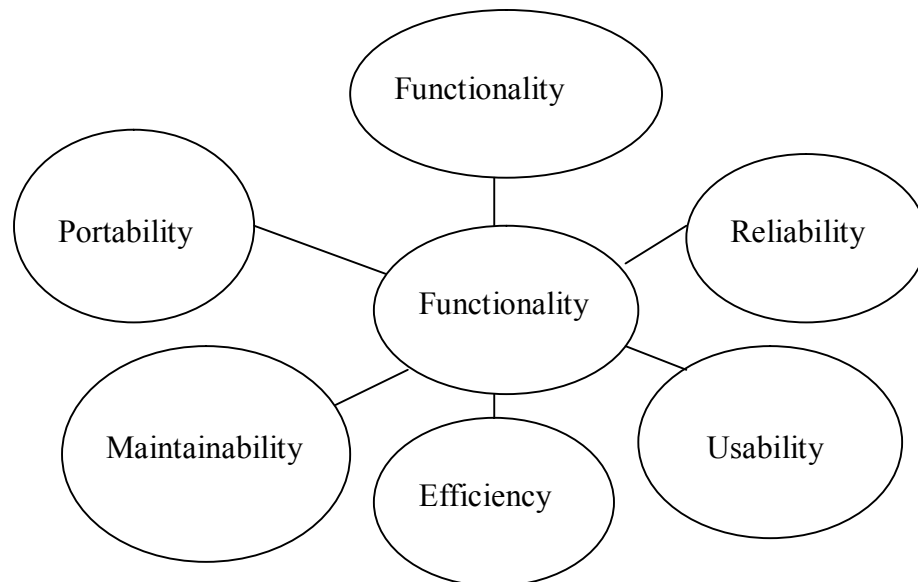
Involvement of the variable team's technical skill	No	No	No	No	No
Involvement of the variable team's project management skill	No	No	No	Yes	Yes

ISO 9001 is an international quality management system standard applicable to organizations within all type of businesses. ISO 9001 internally addresses an organization's processes and methods and externally at managing (controlling, assuring etc.) the quality of delivered products and services. ISO 9001 is a process-oriented approach towards quality management. That is, it proposes designing, documenting, implementing, supporting, monitoring, controlling and improving (more or less) each of the following processes:

Besides the famous ISO 9000, ISO has also release the ISO 9126: Software Product Evaluation: Quality Characteristics and Guidelines for their Use standard (2000). This standard was based on the McCall and Boehm models. Besides being structured in basically the same manner as these models, ISO 9126 also includes functionality as a parameter, as well as identifying both internal and external quality characteristics of software products. ISO 9126 proposes a standard which species six areas of importance, i.e. quality factors, for software evaluation. Figure 2 – 6 explains the ISO 9126 Quality Model.



**Figure 2 - 6 ISO 9126 Quality Model (adapted from ISO 9126 – 1:2001, software engineering product quality 2001)**



The ISO/IEC 15504: Information Technology - Software Process Assessment is a large international standard framework for process assessment that intends to address all processes involved in Software acquisition, Development, Operation, Supply, Maintenance and Support. ISO/IEC 15504 consists of 9 component parts covering concepts, process reference model and improvement guide, assessment model and guides, qualifications of assessors, and guide for determining supplier process capability. Given the structure and contents of the ISO/IEC 15504 documentation it is more closely related to ISO 9000, ISO/IEC 12207 and CMM, rather than the initially discussed quality models (McCall, Boehm and ISO 9126).

IEEE has also released several standards (Humphrey, 2000). To name a few: it is probably the implementation of ISO/IEC 12207: 1995 that most resembles previously discussed models in that it describes the processes for the following life-cycle:  
 Primary Processes: Acquisition, Supply, Development, Operation, and Maintenance,  
 Supporting Processes: Documentation, Configuration Management, Quality Assurance, Verification, Validation, Joint Review, Audit, and Problem Resolution,  
 Organization Processes: Management, Infrastructure, Improvement, and Training

In fact, IEEE/EIA 12207.0-1996 is so similar to the ISO 9000 standard that it could actually be seen as a potential replacement for ISO within software engineering organizations. The IEEE Standard 1061-1998 is another standard that is relevant as the standard provides a methodology for establishing quality requirements and identifying, implementing, analyzing and validating the process and product of software quality metrics.

The Carnegie Mellon Software Engineering Institute (SEI), non-profit group work at getting US software more reliable. Examples of relevant material produced from SEI are the PSP and TSPi. SEI has also produced a number of more extensive Capability Maturity Models that is very similar to ISO 9000 that addresses the topic of software quality (Paulk 1993). They are CMM / SW-CMM, P-CMM and CMMI (Curtis 2001, 2007; CMM website 2004)

CMMI Integrates systems and software disciplines into one process improvement framework and is structured around the following process areas: Process management, Project management Engineering, Support

Similar to the SW-CMM, CMMI discusses the following maturity levels,

Maturity level 5: Optimizing - Focus on process improvement

Maturity level 4: Quantitatively managed - Process measured and controlled.

Maturity level 3: Defined - Process characterized for the organization and is proactive.

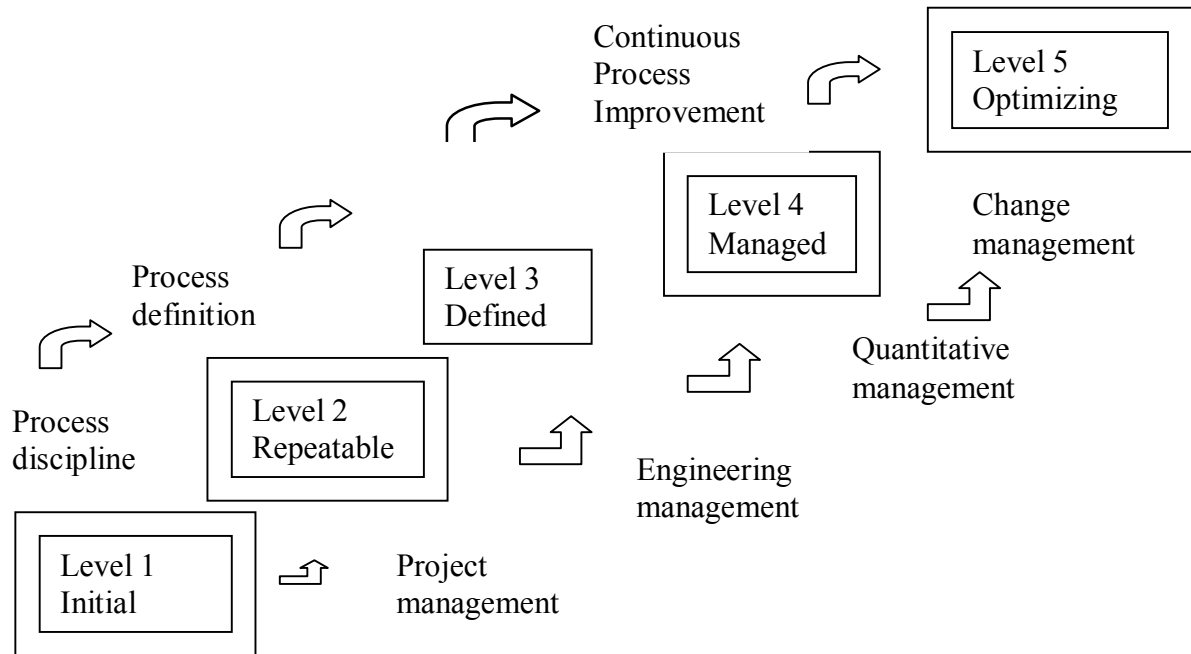
Maturity level 2: Managed - Process characterized for projects and is often reactive.

Maturity level 1: Initial - Process unpredictable, poorly controlled and reactive.

Maturity level 0: Incomplete

The figure 2 – 7 explains the different levels of CMM and the areas of software development it is used.

**Figure 2 – 7 Maturity levels of CMM (adapted from Paulk 1993).**



### 2.2.3.3 Critique of quality models

Most of the quality models presented till now probably could be fitted within the user view, manufacturing view or product view – though this is a futile exercise with little meaning. The models presented herein are focused around either processes or capability level (ISO, CMM etc.) where quality is measured in terms of adherence to the process or capability level, or a set of attributed/metrics used to distinctively assess quality (McCall, Boehm etc.) by making quality a quantifiable concept.

Though having some advantages (in terms of objective measurability), quality models actually reduce the notion of quality to a few relatively simple and static attributes. All these models and frameworks have focus on software process, which have little significance on the technical and personal capabilities of the software people in a team.

The benefit of quality models is that they are simpler to use. The benefit of the quality management philosophies is that they probably more to the point capture the idea of quality. Now we will review the frameworks, which measure the software people skills and are used to improve the organizations quality goals.

#### **2.2.3.4 Models and frameworks which incorporated software team in assessing quality**

A team-based framework is a process framework that is designed for software quality assessment especially in the perspective of a software team. The topic of our study is the impact of team skills assessing quality in software teams so it is necessary to understand the various frameworks existing and used in organizations that has quality criteria's to assess software in terms of a software team. The three main frameworks which incorporated software team in the processes (Table 2 – 5) are:

1. People CMM
2. EFQM
3. TSP

The People CMM (Curtis 2001) provides a roadmap for transforming an organization by steadily improving its workforce practices. As do all Capability Maturity Models, the People CMM consists of five maturity levels, or evolutionary stages, through which an organization's workforce practices and processes evolve. At each maturity level, a new system of practices is added to those implemented at earlier levels. Each overlay of practices raises the level of sophistication through which the organization develops its workforce. Within this environment individuals experience greater opportunity to develop their career potential and are more motivated to align their performance with the objectives of the organization PCMMM

**Table 2 – 5 Overview of frameworks developed for a team's quality assessment**

<b>Quality frameworks</b>	<b>People CMM</b>	<b>EFQM</b>	<b>TSP</b>
Year	2001	2004	2002
Used country	USA	Europe	India and US
Quality criteria	Capability maturity levels	Results Approach Development Assessment Review	Cost and time
Author	SEI	EVANS	HUMPHREY- IEE
Use of variable technical skill of the team	Yes	Yes	Yes
Use of variable Project management skill of the team	Yes	Yes	Yes

Evans (2004) model dealt with the European Foundation for Quality Management (EFQM) Excellence model (6) is a n organizational excellence model using a non prescriptive framework under nine criteria; five of these are 'Enables' for a continuous improvement feedback loop known as RADAR (results, approach, development, assessment and review). It is closely related to Malcolm bridge model. It may be used with organizations of any size and type and is intended for corporations, companies, or non-profit organizations.

Humphrey's (2000) Team Software Process (TSP) provides a defined operational process framework that is designed to help teams of managers and engineers organize and produce large-scale software projects of sizes beyond several thousand lines of code (KLOC). The TSP is intended to improve the levels of quality and productivity

of a team's software development project, in order to help them better meet the cost and schedule commitments of developing a software system. Cost and time are treated as criteria for achieving higher quality.

The existing models for process and people improvement have proved to deliver their promises in large and medium size companies. However, to get the education and training in these models and further to get the certificate is costly both in nominal terms, but also in terms of the effort required, and lost income during the training and introduction time. Certainly, the large companies can take the costs and time needed to systematically implement all the actions needed to gradually climb on the improvement ladder, whether it will be CMM, P-CMM, and PSP for TSP. In small enterprises, however, with 5-15 employees, the situation is different: it may be difficult to motivate the time and costs; the manager may be a member of the development team, thereby having a double role. The organization structures proposed in CMM and also in other improvement programs, are impossible to implement without modification in small companies. For example, senior management, QA, SEPG, PAT, are roles requiring a possession by a person. One and the same person might need to accept the role of SEPG member, QA, and a PAT. Naturally, the basic idea of the roles will then diminish.

A company starting with a couple of software developers may function very well: all the developers have oversight and natural communication. If the company is successful, it will usually grow, with new employees hired. At some point in time, the oversight and natural communication will be lost. The company, especially if successful, will have increasing number of contracts, with the consequence of increasing profit, increasing staff, and also with the existing staff working overtime to rescue projects that are late.

Usually, it is well understood that something must be done in terms of people management, but in a chaotic process, it is difficult to take the time to do the improvements, just to get the current job done. Especially, for the small companies, the improvement efforts should become a way of life and organization culture before the oversight is lost and there is chaos. To this end, quality management support is

needed which will enhance their personal relations and team building capability, before they grow.

Yet another analysis on the existing studies is that they evaluate the team's skill on the design complexity and process complexity of the software developed by the team. That is, they conclude that if the design of the software developed by the team is complex; the quality of the resultant product will be lower and vice versa. Or if the criteria implemented for the software process is inefficient, the software application developed works less productively and the resultant software product will be lower in quality. Whereas the technical and project management capability of a team member is the personal capability or experiences of the team member in evaluating the quality of the software's are not considered.

There have been few studies that have specially addressed the different software quality models, which concentrate on team skills of the software development team and their effect on software quality. The review of existing software development models based on process and team processes has given a motivation for the topic of our thesis, i.e. the importance of team skills in development products quality. Thus it provides justification for the central theme of the thesis. Further, the literature studies on the importance of team skills by various authors are dealt.

### **2.3 Team skills determining software quality – The present study.**

The section above addressed the importance of process and complexity factors in quality of a software product. While it is difficult to argue that complexity and process are factors in the quality of a developed software product, there is a general failure to address what every software project manager knows all too well: the skills of the software development team is a driving factor in software product quality (Leornardi 2007; Krishnan 1998).

In this section we discuss on the different team skills that determine software quality. The quality of the software team is determined by the skills of the team members. The two major skills types are the team's technical skills and project management skills. The forthcoming section introduces the various technical and project management skills that determine the teams quality and thus the quality of the software product

A **team skill** is defined as the cumulative skills of the team members working on a software development team in order to build a higher quality software product (Krishnan 1998). The team member skills in software development is found to be very significant in the stage of design and implementation of the software product (Justin 2006; Samer 2000; Sarkar 2003). There are very few researchers have attempted to capture development team skill as a driving cause of the success of software projects (Boehm 1976, 2007 ; Richard 2007; Robert 2007, 2007). The importance of the team factor in software projects has been identified in several studies of software projects (Boehm 1981; Carmel 2001, 1997). The cultivation of motivated highly skilled software people has been discussed since the 1960's (Cougar 1980; DeMacro 1987; Whitaker 1994).

A recent survey of leading software organizations in Europe identified that the quality of people in software teams is one of the most important factors in improving productivity and quality in software projects (Ann 2006)... Blackburn 1996; Kemerer 1992 reported in this way

*If you were to give an experienced software project manager a proposal and schedule for a software development project, but allow her to dictate one other factor, what would she choose?  
The answer is most project managers would want to select the staff members to be included in the project team*

The following are the studies of the researchers on the area of team skills and quality. Various factors are taken as a measure of quality .One would be cost, other would be the time to completion .The team skills that are researched upon in existing studies are technical experience of the team members and project management skills. The next section gives the studies of other researchers. We present the major studies reviewed in the literature in Table 2- 6





**Table 2 - 6 Team skills determining Quality**

<b>Author</b>	<b>Focus of the study</b>	<b>Main team skills considered in the study</b>
Krishnan (1998)	Role of team factors on cost and quality	Application Domain skill, Programming skill
Justin (2006)	Measurement framework for software quality	Team's technical skills: Programming skill, Language skill, Application /domain skill
Carmel(2001)	Focus on time to completion of software as quality factor	Application domain skill, Programming skill
Bach (1999)	Team member capability on performance ranking	Motivation, Communication
Kartz (2001)	Technical and interpersonal efficiency on quality	Application domain skill, programming skill, Leadership skill, Motivation
Pressmen(2006)	Quality of people	Motivation
Krishnan(1999)	Package software product in system software domain	Application Domain skill
Zachary (1998)	Quality factor as field defects	Language skill, hardware skill, application domain skill
Blackburn(1996)	Focused on small firms	Motivation

Justin (2006) provided empirical evidence that development team skill has a significant effect on the quality of a software product. A measurement framework for assessing software development skill and experience was used to quantify the capabilities of the software development team. These measures were correlated to measures of software product quality taken from the ISO/IEC 9126 standard. As expected, higher proportions of skilled and experienced staff in the requirements, design, and implementation life cycle phases increased the adequacy of the design modules and source code units, and had a similar but more muted effect on the

functional implementation completeness and correctness, and on the volatility of the software specification. The proposed framework for measuring development team skill and experience provided an adequate representation of the distribution of individual skills on a given development team to better understand the strengths and weaknesses of their staff that could prevent the assignment of tasks that are not well suited to less skilled and experienced engineers, and thus prevent the negative impact on product quality.

Carmel's (2001) study focuses on the factors that reduce the development time for completion of software. A statistical model is used to assess these factors. A diskette-based survey was used to collect data from development efforts at a sample of North American packaged software firms. The data are used in a regression model that reveals three factors explaining 38% of variation in time-to-completion. Out of the three factors, team cross-functionality (representation of different functional experiences in the team reduces time-to-completion); team independence (a higher degree of independence from top management turns out to increase time-to-completion); and product complexity (which increases time-to-completion), team factors were found to have substantial relevance.

Krishnan (1998) studied the role of team factors on software cost and quality. As per his study, the domain experience, personal capabilities and size are the team factors that affect software quality. Again in Krishnan's (2000) study the domain experience and personal capabilities showed a positive impact on the quality of the software product. The team sizes were not seen to impact on the software quality.

Bach (1995) examined through his empirical study argues in favor of software heroism. All software managers are faced with three P's when software is to be built: people, problem, and process. His results conclude that out of the 3 p's of project management, more weightage should be given for the people roles. A recent survey of leading software organizations in Europe identified that the quality of people in software teams is one of the most important factors in improving productivity and quality in software projects (Blackburn 1996). Banker (1991) modeled the capability of each team member based on their performance ranking within the organization

Kartz (2001) has identified three essential management skills: (1) technical, (2) human. Technical skills encompass the ability to apply specified knowledge or expertise. The ability to work with, understand and motivate other people, both individually and in groups, describes human skills. Managers that are technically and interpersonally competent can only be high performing.

Pressmen (2005) and James Bach insists that the quality of the people is the primary driver for software quality, and that too much industry focus has been on the process. He argues that personal performance is "guided by higher level process models embedded within experience, education, and insight."

Krishnan's (1999) empirical analysis indicates that experience of the software team in the specific software domain significantly affects product quality. One reason for this finding is that the packaged software products in this study belong to systems software domain (such as compilers and network utilities). In systems software development, the team needs to have at least a few key members who are very experienced in the specific domain since these products may need to be made compatible with other utilities. It may be noted that all the three empirical models in this study identify overall personnel capability of the team as an important variable for project success.

Carmel and Sawyer (1998) and Zachary (1998) confirm the findings that the total experience, the language experience, the hardware experience, and the application experience of the team determine software quality. In addition, measures of programmer, manager and analyst capabilities have also been proposed. The analysis in this study is based on the data collected on 37 package software projects from a leading firm in the packaged software industry. The results indicate that higher levels of personnel capability on the software development team are significantly associated with both improved productivity and quality of software products. Also higher levels of domain experience of the software team are associated with a reduction in the number of field defects in the product.

Blackburn (1996) researched focused on small firms and widened to quality management strategies. Insights in to the project management skills motivation and

influence are constraints of manager quality strategies. Software firms research in small business. The research was done in Kingston university small business research centre. Questionnaires, telephone samples were taken. Quality strategies are related to functionality.

The next section discusses on the skills of the team establishing the software quality, which is taken from the early literature and utilized in the current study. A team requires two types of skills to perform effectively. First it needs people with technical expertise. Second, it needs people with higher project management skills (Stevens 1994). The next two sections deal with 1) The technical skills needed for product development and 2) The project management skills required to manage the project.

### **2.3.1 Technical skills for product development**

A technical skill of a software team is the totality of team member skills in a team (Krishnan 1998). The assessment of the capability of the software development team requires insight into the team's collective experience (Justin 2006; Fahmi 2008; Samer 2000).

Literature shows that there are four major team skills that determining quality. They are

- (a) Programming skill
- (b) Language skill
- (c) Application domain skill
- (d) Hardware skill

**Carmel (1998) and Curtis (1988) study indicates that team factors such as personnel capability and experience, personnel motivation, coordination and communication among team members are believed to be critical for project success in packaged software development**

Kemerer (1993) argues that the quality of the software team (i.e. capability of programmers and analysts) is the second most significant factor in determining either

cost or quality in software products. Kemerer (1992) noted that experienced software project manager consider a proposal and schedule for a software development project as the most important inputs for successful projects, but allow her to dictate one other factor, what would she choose? The answer is most project managers would want to select the staff members to be included in the project team. The importance of the team factor in software projects has been identified in several studies of custom software projects (Boehm, 1981; Brooks, 1993).

Kemerer (1993) findings indicate that the effect of team experience on costs and quality is not linear. However, a serious limitation in these experimental studies is the sample population. That is, the samples for these experimental studies are typically not drawn from professional software projects and the programmers are often students. Thus, these findings need to be validated on professional software projects.

### **2.3.1.1 Programming skill**

Within software engineering, programming (the *implementation*) is regarded as one phase in a software development process (Marciniak 2002). Programming is the process of writing, testing, debugging/troubleshooting, and maintaining the source code of computer programs. Programming skill is the skill / experience of the team member in programming software. The next section gives you the early literature findings on the programming skills of the software developer and its impact on the quality of the software. Reagans (2005) investigated a large number of productivity and quality related studies have attempted to identify the effect of individual differences in software programmers.

Curtis (1981) studied the effect of individual differences between programmers, it was noted that three programmers took 39 minutes to find a planted bug whereas another programmer took only three minutes to debug the same problem, an order of 13:1 difference. The major difference between these programmers was their years of experience in the specific software domain (Kemerer 1993). The study attempts to prove that the programming skill of the software developer /team member has a significant importance in determining the quality of the software product as it is known that the completion time and cost are a major determinant of quality.

### 2.3.1.2 Language skill

In the previous section, we explained the software development as the process of writing, testing, debugging/troubleshooting, and maintaining the source code of computer programs. This source code is written in a programming language. The code may be a modification of an existing source or something completely new. Some languages are very popular for particular kinds of applications (e.g., COBOL is still strong in the corporate data center, often on large mainframes, FORTRAN in engineering applications, scripting languages in web development, and C in embedded applications), while some languages are regularly used to write many different kinds of applications.

The expertise of the team member or a software developer in the programming language syntax is termed as “language skill”. The thesis is dealt on the “impact of team member skills on software development” and thus the language skill is taken as an independent variable in the study. Various studies on language skills are dealt in the next section

Rose (2002) in his experimentation showed significant correlation between the performance of the software developers in software tasks and the domain and language experience of the developers in number of years. However, in the same study, the significance of this correlation was not found for developers with more than three years of experience. Similar experimental studies have been conducted to compare the defects in software developed by individuals with different levels of programming language (Humphrey 1989; Kemerer 1993; Sheppard 1979).

### 2.3.1.3 Hardware skill

Hardware is a term given to the machinery and the various individual pieces of equipment. It refers to the physical devices of a computer system like input, storage, output, control and processing devices are hardware. In order for a computer to produce a useful job, hardware and software must work together. Nothing can be done with the computer hardware of its own, and software cannot be utilized without supporting hardware (Pradeep 2003). The same hardware can be loaded with different software to make a computer perform different type of jobs.

The skill or experience of an individual on the physical devices of a computer system on which the software development activity is performed is termed as “hardware skill”. For a software development team to work effectively in development of software projects, it needs the support of a hardware skilled team member.

Banker (1991) developed an approach on evaluation of managerial impact of each of the factors influencing software productivity (Humphrey 1989). A number of environmental factors affects the labour hours required to complete a project and hence the productivity of the software maintenance effort. These factors include the ability and previous experience of the software personnel (Sackman 1968; Chrysler 1978), hardware and software tools (Thandhani 1984; Lambert 1984; Boehm 1981; Jones 1986) and the attention spent on systems quality (Kriebel 1979; Case 1985).

#### **2.3.1.4 Application Domain skill**

Application domain skills are the skills or experience of a software developer on the particular business domain of the application they are developing on (Krishnan 1998).

Adelson (1985) noted that a designer’s expertise rests on the knowledge and skills, which develop, with experience in a domain. As a result, when a designer is designing an object in an unfamiliar domain he will not have the same knowledge and skills available to him as when he is designing an object in a familiar domain. In this paper we look at the software designer's underlying constellation of knowledge and skills, and at the way in which this constellation is dependent upon experience in a domain. What skills drop out, what skills, or interactions of skills come forward as experience with the domain changes? To answer the above question, a study expert designers in experimentally created design contexts with which they were differentially familiar. In the study, a description the knowledge and skills were found central to each of the above contexts and discuss the functional utility of each. In addition to discussing the knowledge and skills it was observed in expert designers and also compared novice and expert behavior.

Barstow (1985) investigated the extent to which domain knowledge is crucial to an automatic programming system and the interaction between domain knowledge and programming at the current time. His project at Schlumberger-Doll Research has been



investigating this issue in the context of two application domains related to oil well logging. Based on these experiments we have developed a framework for domain-specific automatic programming. Within the framework, programming is modeled in terms of two activities, formalization and implementation, each of which transforms descriptions of the program as it proceeds through intermediate states of development. The activities and transformations may be used to characterize the interaction of programming knowledge and domain knowledge in an automatic programming system.

### **2.3.2 Project management skills**

Project management skills are the skills of the managers in a software team to efficiently manage a project (Ramasubbu 2008). Project management skills are very important in a team to work effectively. In order to achieve higher outcome from a software development, the team has to be motivated and focused toward the goal (Ferratt 2005). It is an ongoing process that requires project management skills and an understanding of many organizational factors. For most software projects teams are not effective, they fail primarily because people, particularly highly talented software developer's lacks project management skills (Hasan 2008).

James (1999) examines the impact of the spectrum of risks on different aspects of system development. The results of a survey of 86 project managers that indicate common aspects of project effectiveness are generally under control, but are most affected by lack of expertise on the project team. Significant relationships also show that lack of clear role definition and conflicts on the team are also elevated risks. Other items are not as critical or limited to a much smaller aspect of effectiveness than overall success. This focusing in on the more important risk aspects will allow for more effective management of the project and a narrowing of techniques to mitigate the significant risks.

The key challenge facing IT software and service providers is identifying critical skill sets (Tingoles 2008). The team factors such as personnel capability and experience, personal motivation, coordination and communication among team members are

believed to be critical for project success in software development (Javed 2004; Faraj 2000; Carmel 1998; Curtis 1988).

Lorange (1986) studied the effective project management skills as: Team building, Communication, Motivation, Leadership. Having a certain level of technical competence is helpful, but managerial and interpersonal skills are the most important for project managers (Hargie 1999). Broad background is more important than expertise in any technical area. Successful project managers are generalists, not technical specialists. They can come from various parts of the organization, not necessarily from the IS organization. This thesis deals with the project management skills Leadership, motivation and communication.

### **2.3.2.1 Leadership**

Leadership is the ability to influence others in order to work towards the achievement of shared goals. It is too well known that leadership is the critical factor determining the success or failure of an organization (Bruce 2007).

Project leadership starts with the selection of the project manager and the key members of his or her management team. Effective leadership is needed to keep the team focused and motivated throughout the project. Judy (2002) through his study has shown that leadership and communications are the project management skills that lead the team towards success.

### **2.3.2.2 Communication**

Many project failures can be traced to a breakdown of communications (Downs 1988). It is the responsibility of the project manager to create an environment for effective communications within the team and manage the communication process with external stakeholders, particularly with the client's organization (Dutoit 1988). Effective project managers keep all involved parties informed. They never surprise the client. They also do not depend on formal reporting structures alone. Body language at a status meeting can often provide more information than a carefully worded written status report. Good communications among project team members

and all affected parties are required. Many project failures can be traced to a breakdown of communications

Javed (2004) and Durani (2004) investigated the factors (related to team communication) that have a significant influence on software quality. For this study, 23 factors that could possibly affect job satisfaction are taken into consideration. These factors were grouped into categories like communication, working environment, and duration of service.

Cohen (2001), Clampitt (1993) and Downs (1988) have investigated the impact of effective communication practices on software quality at the organizational level. Kortner (2002) has shown through various experiments that organizational communication is a strong predictor of team effectiveness (Myers 1982)

### **2.3.2.3 Motivation**

One of the significant factors of determining software quality in the context of teams is motivation of the team members. Team member motivation and its positive impact are dealt in many areas of organization behavior. Motivation is the process that accounts for an individual's intensity, direction and persistence of effort toward attaining a goal (Mithas 2007).

When we think about motivation in teams, the impact of the national culture is striking. For an Indian, work is not just a means of living but it is worship. For a US citizen, work is only a means of living. National culture and social institutions also influence the levels of satisfaction workers expect to receive from an organization and how committed workers are to their organization and its social goals (Elashmavi 2001)

### **2.3.3 Team Size**

Another important team-related factor is the team size. The software engineering literature indicates significant benefits from using small teams. Carmel and Bird (1997) find that the common justification is not because a small team is so advantageous, but that a large team is disadvantageous.

The most effective teams are neither very small (under four or five members) nor very large (over a dozen). Very small teams are likely to lack diversity of views. But when teams have more than about 10 -12 members, it becomes difficult interacting constructively and agreeing, and develops cohesiveness, commitment and high performance (Nieva 1985).

Ioannis (2003) investigated whether those projects of similar size exhibit similar productivity. This is due to the software team size and organization characteristics (e.g. large teams have lower productivity because of the higher interaction needed between the team members

Jefferya (2000) examined team size as a criterion of quality. The variables suggestion for software projects was team size; language type, and development platform that are recommended as important criteria for excellent quality software. As productivity is taken as criteria of quality, a higher team size indicates that more working hours were needed and, therefore, shows a lower productivity and thus quality and vice versa. The relationship between team size and Productivity is not linear.

## **2.4 Summary**

In this chapter we looked at the different definitions of software as well as the major models used in defining and assessing software quality. However as most of these models don't consider the people aspect of software quality, the need for incorporating the team skills in assessing software quality is argued. In summary, the literature study highlights the importance of team factors on determination of software quality. The quality of the software team is determined by the skills of the team members. The two major skills are product technical skills and project management skills. The product technical skills are specified as (1) Programming skills (2) Language skills (3) Hardware skills and (4) Application domain skills. The project management skills are (1) Leadership (2) Motivation (3) Communication. Apart from these, team size is also an important factor. In forthcoming chapter we first assess the impact of globalization on software projects and the software development process. We also frame our hypotheses by considering how globalization could alter the impact of team skills on software quality.

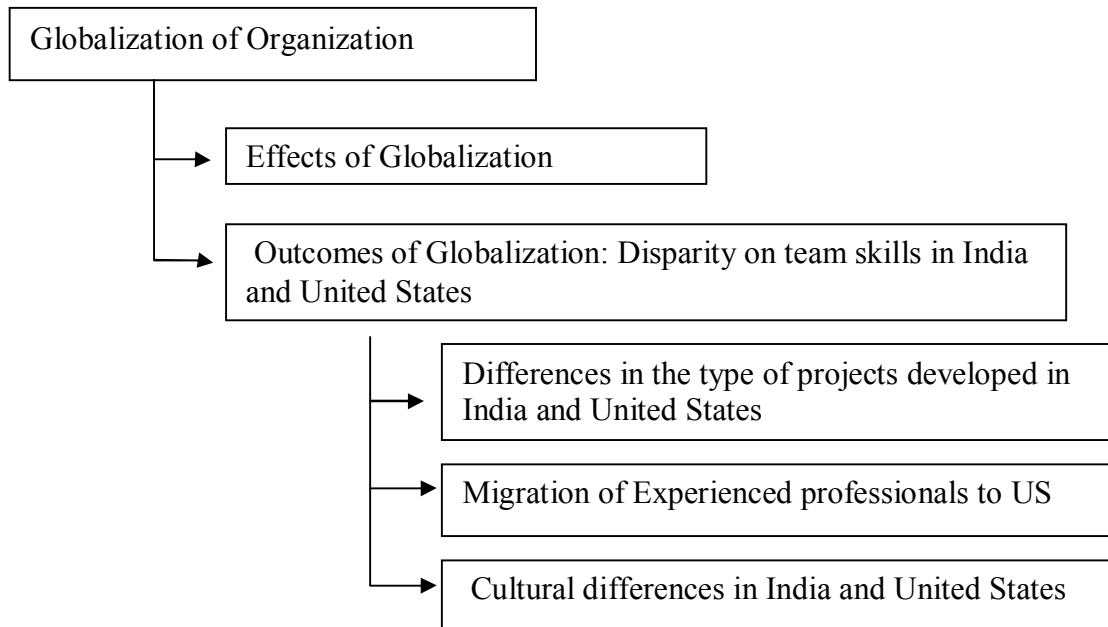
## **CHAPTER 3 - LITERATURE SURVEY**

### **GLOBALIZATION OF ORGANIZATIONS AND ITS IMPACT ON SOFTWARE DEVELOPMENT**

#### **3.1 Introduction**

The previous chapter sought to demarcate the broad domain within which the area of research of this study is located. Further, a review of the pertinent literature was presented that concerns the dependent variable of the conceptual model -- software team quality. This chapter aims to complete the review of the germane literature concerning the antecedents depicted in the conceptual model that are proposed directly or indirectly to influence the software quality and the reasons for disparity of these variables in a globalised scenario by focusing on software development issues in two different countries -- India and the United States of America.

This chapter proceeds as follows. First, it gives a review of literature on globalization and the related effects. Second, it discusses the literature concerning the reasons for disparity in team skills between India and United States as an outcome of globalization. Third, review of technical skills and project management skills and their differences between India and United States are discussed. The organization of this literature review is shown in Fig.3.1

**Figure 3 – 1 Organization of Literature review**

**Globalization** is the process of integration of economies across the world through cross border flow of factors, products and information. Globalization may be considered at two levels (Ducker 1995), at the macro level (i.e. globalization of the world economy) and at the micro level (i.e. globalization of the business and the firm), which are very much interdependent. Globalization of the world economy is achieved, quite obviously, by globalizing the national economies. In our study, our concern is the second level - the corporate level.

At the corporate level globalization in its true sense is a way of corporate life – necessitated, facilitated and nourished by the trans-nationalization of the world economy and developed by corporate strategies (Philip 1994). Globalization is an attitude of mind it is a mindset which views the entire world as a single market so that the corporate strategy is based on the dynamics of the global business environment (Mitchell 2000).

Normally, a firm passes through different stages of development before it becomes a truly global corporation. Typically, a domestic firm starts its international business by exporting. Later it may establish joint ventures or subsidiaries abroad. From an international firm it may develop in to a multinational firm and finally into a global/transnational one (Suvillivan 2000).

Ohmae (1990) identifies five different stages in the development of a firm into a global corporation. The first stage is the arms length service activity of essentially domestic company, which moves into new markets overseas by linking up with local dealers and distributors. In stage two the company takes over these activities on its own. In the next stage, the domestic based company begins to carry out its own manufacturing, marketing and sales in the key foreign markets. In stage four, the company moves to a full insider position in these markets, supported by a complete business system including R and D and engineering. In the fifth stage the company moves towards a genuinely global mode of operation.

Although the globalization trend has been very pronounced in the last two decades or so, it is not a new phenomenon. The period 1870 to 1913 experienced a growing trend towards globalization (Konicki 2002). The new phase of globalization which started around the mid-20<sup>th</sup> century became widespread, more pronounced and overcharging since the late 1980's by gathering more momentum from the political and economic changes that swept across the communist countries, the economic reforms in other countries, the latest multilateral trade agreement which seeks to substantially liberalize international trade and investment, and the technological and communication revolutions (Francis 2009).

There are several similarities and differences between the two phases of globalization. However, the current phase of globalization is characterized by several new features: new markets, new technologies, new actors and rules and norms (Crowston 2006).

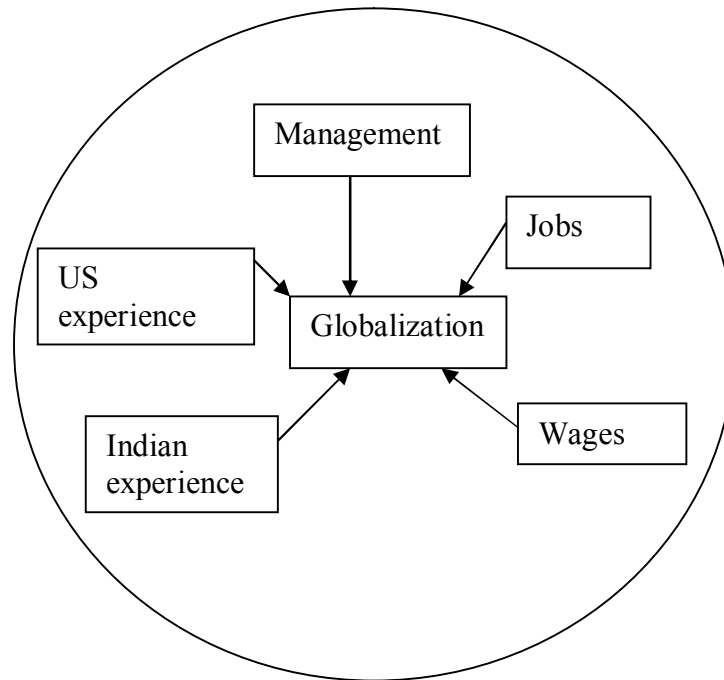
### **3.2 Effects of Globalization**

**The effects of globalization are varied, ranging from cultural degradation to the fear of loss of sovereignty. The next section discusses on the beneficial and harmful effects of globalism in different countries. We focus on the effects of internationalization on management practices, wages, and jobs and lastly on the Indian experience of globalization, providing a multifaceted flavor of how globalization has touched the lives of Indians. See Figure 3 -2.**

Globalization has both beneficial and harmful effects (Elashmawi 2001). It differently affects different countries sectors, industries, firms and sections of people. It is interesting to note that developing countries like USA are angry against globalization because of deindustrialization, job cuts etc.



**Figure 3 – 2 Effects of Globalization (adapted from Elashmawi 2001)**



Although globalization can benefit the developing countries in several ways, unregulated globalization will cause serious problems for developing countries. The almost universal acceptance of the market economy and the globalization driven by private enterprise tend to aggravate, in the absence of proper regulation, most of the harmful effects traditionally attributed to neo-colonialism. Several approaches at the national and international level s are required to mitigate the harmful effects and to reap the benefits of globalization. A significant yet subtle shift has occurred in the area of management practice (Suvillian 2000).

While the domain of management is well studied and documented in the western, developed countries, it is not the same with the countries in Asia. The skills that drive globalization efforts also impact the way people are managed in the corporate (Aswathappa 2007). The older personnel management approach has given way to the “human resource management” approach. The autocratic style that was fed by hierarchical position conscious systems is being swiftly replaced by flat

organizational hierarchies. The individual in a position of power, driving policies and processes have swiftly evolved in to team based collaborative management methods.

Another landmark change in management method is initiated by globalization that has happened in the area of organizational leadership. A new generation of leadership skills, styles and methods have evolved. The straightjacketed approach to certain defined good and bad leadership skills, styles and methods has been replaced with multiple theories supporting a variety of leadership styles. Leadership today is associated with the particular phase in the life cycle of an organization and is considered to be industry specific (Ducker 1995).

Globalization adds to the employment especially in developing countries, as Table 3-1 shows. With the increased free trade, more than 500, 000 jobs moved out of the US by 2005 and the figure will go up to 3300, 000 by 2015 (Aswathappa 2007). See Table 3 – 1.

**Table 3 - 1    Employment in manufacturing sector (adapted from economic and political weekly, January 3 2004, p.106).**

1978 – 79	14,903
1983-84	10,198
1989-90	7,252
1991 – 92	4,234
1996-97	2,881

### **3.2.1    Indian experience of globalization.**

India's economic integration with the rest of the world had been very limited because of the restrictive economic policies followed until 1991. Till then Indian firms confined themselves, by and large to the home market. Foreign investments by Indian firms were found to be very insignificant. With the new economic policy ushered in 1991, there has however been a change. Globalization has in fact become a buzzword

with Indian firms now and many are expanding their overseas business by different strategies (Mitchell 2000).

The Indian business however suffers from a number of disadvantages in respect of globalization of business. Not only that government policy lacks a positive orientation in some respects, but also the government policy and procedures in India are among the most complex, confusing and cumbersome in the world (Suvillivan 2000). Another problem is that the high cost /inadequacy of many vital inputs and other factors such as raw materials and intermediates, power, finance, infrastructural facilities like port etc. tend to reduce the international competitiveness of the Indian business. There are also problems related to technology, small size and lack of experience of firms, poor quality, lack of R&D efforts etc.

Although, India has several handicaps, there are also a number of favorable factors for globalization of Indian business (Ebert 2001). These include the human resources, growing entrepreneurship, growing market, the growing foreign market that is opening up more, niche markets and large number of non-resident Indians who are resourceful (Konicki 2002). If the Indian firms have the facility to obtain the latest technology in the world, to raise finance from the cheapest source and procure materials from the best source in the world, they are on equal footing with the foreign firms in many respects. Further, if the Indian firms can muster some edge over the foreign firms in respect of labour cost, productivity, product quality/features that could be a competitive advantage.

### **3.3 Outcomes of Globalization: Disparity of team skills in India and United States software development projects.**

In this study we focus on an interesting outcome of globalization: the globalization of software development activities and the attendant problems faced by firms especially in the context of managing software development teams. With India emerging as an important center of software development, several firms have located some of their software development activities to India. However the team skills those are easily available in India and in a western country (like the U.S), are quite different. There is a wide range of disparity in team skills found in countries India and United States. Of

course one of the main reasons behind this disparity in skills is due to the past legacy of the nature of globalization of the software development projects (Arora 2006).

Software development is globally outsourced in an extensive manner (Pozzebbon 2005; Gambardella 2004; Herbsleb 2001; Hayes 2002). This has led to the migration of software development and maintenance operations to geographically distributed locations. The number of US jobs moving offshore is depicted in the Table 3-2. From the nineties, the trend is outsourcing the software projects to centers in India. Catherine (2005) explains that geographically dispersed teams have been prevailing in many service based organizations and significantly effects team effectiveness.

Globalization adds to the employment, particularly in developing countries, as Table 3 - 2 shows. With increased free trade, more than five hundred thousand jobs moved out of the US by 2005, and the figure will go up to 330 thousand by 2015. This does not necessary imply that the US economy suffers as a result of this loss of job. Globalization argues that Free trade will result in countries specializing in the production of those goods and services that they can produce most effectively, while importing goods that they cannot produce as efficiently. If, for example, US are outsourcing 14,000 legal jobs, the country is saving enormously on the cost o performing those jobs with the country.

**Table 3 - 2 Number of US jobs moving offshore To Low wage countries such as India, China, Mexico and the Philippines (Source: the Economic Times, February 2, 2003).**

	<b>2005</b>	<b>2010</b>	<b>2015</b>
Life Sciences	3700	14,000	37,000
Legal	14,000	35,000	75,000
Art Design	6,000	14,000	30,000
Business Operation	61,000	162,000	348,000
Computer	109,000	277,000	473,000
Architecture	32,000	83,000	184,000
Sales	29,000	97,000	227,000
Office Support	295,000	791,000	1, 700,000
Total	588,000	1,475,010	3,300,000

The phenomenon of global outsourcing of software has led to severe skill shortages (Watson 1990). This is compounded by the need for a multitude of skills in software development and the business domains. Several authors have looked at these issues in detail. We present some of the important studies that considered issues related to the shortage of software development and allied skills in the context of the global outsourcing of software development.

Judith (2007) studied the key challenges facing IT software and service providers in identifying critical skill sets. The challenge is compounded as a result of the continuing growth of outsourcing and the increasing demand for technology solutions. Through a survey of IT software and service providers, a comparison of skills sought by those firms which are in IT services and software providers were done. Results indicate that, surprisingly, software and services providers place more emphasis on business domain and project management skills than on technical skills.

Tschang's (2001) investigations showed that the skills most in demand in India will be areas as business application of software development, project management and specific technical skills. Judith (2007) examined the Indian software industry skills as: product development skills: basic technical skills such as coding and programming languages, application skill gained by experience and project management skills (Zollo 2002). Every task from software or services to software products involves not only technical conceptualization skills, but also project management skills. Various types of project management skills are needed, particularly for developing and managing software development projects, particularly soft communication ability, interpersonal skills that are leadership, motivation for dealing with people management. The study explored the Indian software industry team skills as: basic technical skills as coding and programming and language skills and project management skills as leadership, motivation and communication capability (Hertel 2003; Gopal 2002; Jones 1986). Similarly (Catherine 2005) found that software development is globally sourced and it affects team effectiveness. (Judith 2007) found that the team member skills that affect quality most in demand in different countries are the technical skills and project management skills of the team member. The

Product technical skills are programming skills and language skills and project management skills are leadership, motivation and communication

Outsourcing of software development encounters various difficulties by cultural and language differences, lack of communication, technical ability and experience (Ebert 2001). One such difficulty is the skills of software developers that vary in different countries at different locations. The next section reveals the various reasons for the disparity in skills of software team development in the countries India and United States that are relevant to this study.

The three major factors that contribute towards the disparity in team skill between India and the United States are:

- (a) Differences in the type of projects developed in India and United States.
- (b) Migration of experienced professional to United States.
- (c) Cultural differences in India and United States

### **3.3.1 Differences in the type of projects developed in India and United States.**

**The section discusses on the disparity in team skills in countries United States and India. The section explains the variables of the study that are the technical skills/experience of the software developers that are (1) Programming skill (2) Language skill (3) Application domain skill and (4) Hardware skill differ in India and United States due to the following grounds:**

- (1) Types of projects developed differ in India and United States.
- (2) Migration of experienced professional to United States.

The first one is discussed in this section 3.3.1 and the second one is discussed in section 3.3.2.

The software development activity comprises of six stages: (1) requirements analysis (2) high level design (3) low level design (4) coding (5) testing and (6) postproduction support (Royce 1970, Raccon 1995). Each stage has feedback loops to the previous stage: meaning that a particular stage does not necessarily lead to a succeeding stage, but could lead back to a preceding stage. The software development industry in the

developing countries like United States mainly concentrated on the first 3 stages of software development and the later stages were done in the developing countries that were outsourced to Asian countries like India (Arora 1999).

While thinking about the Indian software development industry in India where the survey was undergone, it has nearly 65% of the revenues from exports that are predominantly low-end software development services (Shah 1995; Arora 1999; DCosta 1998, 2002). Out of the revenue from exports, 80% are from service oriented projects including custom software development, consultancy and professional services. Table 3 – 3 shows the explanation to this observation.

**Table 3 - 3    Composition of Indian software development and services  
(Nasscom 1999)**

<b>Software activity</b>	<b>Domestic software (%)</b>	<b>Export software (%)</b>
Professional services	52.0	8.4
Custom development services	28.6	31.5
Products and packages	4.1.4	8.8
Training	6.1	1.5
Support and maintenance	3.2	3.0

**Given the tight labor market conditions in the U.S., especially for software development employees, the availability of software development services from India has been of substantial value to many large and medium-sized U.S. firms that have been able to free up their in-house IT staff for more valuable and creative projects (Bassellier 2004; Mithas 2007). The growth of Indian software development industry over the years from 1991 to 2001 is given in the Table 3 – 4.**

**Table 3 – 4 software Industry Growth (Nasscom 2005)**

<b>Countries</b>	<b>Average growth in 1990's</b>	<b>Export % of sales (2002 or latest available year)</b>
Brazil	20	1 – 2
China	30-35	11
India	40	80
Ireland	20	85
Israil	20	70

Samer (2000) describe the Indian software industry value chain in terms of the following activities: (1) data entry, (2) body shopping, (3) offshore development, (4) customized solutions, (5) premium services, (6) niche technologies, and (7) products.

The activities are ranked according to increasing value added as well as increasing risk. While these activities actually describe discrete lines of business, elements of these are present in the classifications of software development activities by Arora (1999) and Heeks (1996).

A third category of work involved companies in the Indian software industry is examined by Arora (1999) in his empirical survey that is performing a mixture of offshore and onsite work. Whereas onsite services dominated in 1991-1992 with about 95 percent of total software exports (with offshore services at 5 percent), by 1999-2000, offshore services in India had increased to 42 percent, and overseas onsite accounted for the residual 58 percent (NASSCOM, 2005).

The domestic as well as export sectors in Indian firms usually provide low level design, coding and some types of testing services (Ashish 2001; DCosta 2002). The software service providers place more emphasis on business domain, technical skills and project management skills (Timgoles 2008).



Indian firms provide essential maintenance and development services, enabling US firms to use their scarce in-house IT staff for higher value added work, such as design and develop new types of applications (Ashish 2001). Managers at most of the U.S. firms agreed that the type of work outsourced was neither technologically very sophisticated nor critical to their business. Requirement analysis and high-level design is typically done either in-house or by U.S. based consultants (Arora 1999, 2001; DCosta 2002).

A number of areas of dissatisfaction that the Indian firms had no domain knowledge and poor management skills were investigated by Arora (1999) and, DCosta (1998). Even a highly rated Indian subsidiary of a leading electronics and communication firm was considered 4-5 years behind the latest communication technologies. Most of the managers believed that those Indian firms could not work on high level specifications or project definition stages of a project, although for the most part, this belief was untested. Designer's expertise rests on knowledge and skills, which develop, with experience in domain (Adelson and Soloway 1985; Boehm 2007)

Tschang's (2001) study conducted in the organizations in Asia found that the product and business development skills are lacking in India. The skills most in demand over the next three years will be in areas such as business applications of software development (23 percent), Web-based applications (34 percent) and Application Service Providers (ASPs); systems skills such as software engineering (18 percent) and project management; and specific technical skills such as Java and OOPS, etc. (NASSCOM, 2000). In addition, advanced technical skills and knowledge appear to be lacking. Business skills were generally lacking, and the absence of experienced people may be due to constant migration. One perspective holds that an engineer could learn business expertise, but not the other way around. The most important factor is that the Indian experience was not based on manufacturing, but, rather, more on "service" and intellectual work. India's capability for manufacturing computer hardware and other electronics has been and still is relatively poor. The Empirical evidence shows these observations on skills:

1. Skills and organizational capabilities played a vital role in the growth of software development industry. In particular technical and project management skills contributed to the development of IT industry.
2. The current skills sets are a limiting factor on further growth, being narrowly focused on technical skills and too limited in broader in middle level experience.
3. Higher project management and technical skills are needed for further development.

### 3.3.2 Migration of Experienced professionals to United States.

Experienced engineers with software development or (IT) training are now in short supply in India (Gibbons 2004, 2006) as well, impeding the ability of Indian software firms to offer high-end services (Arora 1999, 2001; Heeks 1996). The Indian software industry experiences shortage of experienced software professionals. India has an abundant supply of entrepreneurs who recognized and responded to the opportunity that the IT revolution in the West represented. Though, India graduates about 155,000 engineers of various sorts, and another 200,000 diploma holders per year, about 60,000 of these enter the IT sector. Despite the apparently large stock of human capital in India, NASSCOM claims that by year 2010, demand will outstrip supply. This is due to the constant migration of these professionals to the United States (Table 3 – 4). Hence, advanced technical skills and domain skills were lacking and absence of experienced people may be due to constant navigation (Gibbons 2004).

**Table 3 - 5 Indian software firms: descriptions for migration (adapted from Simon Commander, 2004.)**

	Share of workers who left in last year		Share of worker who have worked on site and offsite				Relevant experience
	2002	1999	2002		1999		2002
			Onsite	Offsite	Onsite	Offsite	
Managers	3.9	11.3	20.9	60.7	22.8	56.3	54.3
Conceptualizes	3.5	8.4	25.8	59.3	27.2	58.2	54.1

Developers	10.8	16.2	23.0	45.4	25.6	47.2	44.7
Modifiers	5.6	9.3	33.2	43.7	37.2	39.2	17.4
Supporter	2.6	9.4	19.3	46.8	21.9	46.9	6.4

It is widely believed that the key to the success of the Indian software exports is the supply of trained, low cost software professionals (Arora 2001). The estimated wage costs in India were about 20% to 30% of the corresponding U.S. levels for comparable work. India not only has the largest number of people working in the software industry but also apparently the highest rate of growth of revenue, while having the lowest revenues per employee. Whereas the Irish and Israeli firms appear to earn as much as \$100,000 per employee or more, firms in the Indian software industry earn only about \$15,000-\$20,000. However, revenue per employee is probably a far better indicator of productivity in the software services industry. In this industry, equipment is not very expensive compared to the other service industries in India and firms do not differ very significantly in terms of the capital employed per employee (DCosta 2002). Thus, revenue per employee provides a reasonable motivational measure of productivity.

This section explains the reasons for disparity of team's technical skills in software development teams in countries United States and India. The difference in the technical skills that are programming skill, Language skill, application domain skill and hardware skills in these countries are due two reasons: (1) The type of projects developed in these countries and (2) Migration of skilled professionals to United States. Further, the literature review showed these findings

- (1) The type of projects in developing countries in Asian countries, like in India outsourced is mainly software services, where they need expertise in the low level programming and language skills
- (2) The developing countries like the United States concentrate on the high level design, which involves application domain experiences for the team members.
- (3) Experienced professionals are migrating and more in United States due to the high income as a motivation in these countries.

(4) A team member with excellent hardware skills in the team impact the success of the team's development of software product.

The next section reviews the reasons for the differences in the project management skills in countries United States and India. The reason is the cultural differences in these nations. National culture impacts the work culture in organizations.

### **3.3.3 Cultural differences in India and United States**

This section explains the cultural differences of the employees in the countries United States and India. Organizational culture differences in turn impact the personal capabilities and project management skills of the employees/software developers in organizations (Jorge 2009). The various project management skills concentrated in this study are the (1) Leadership (2) Motivation (3) communication.

Globalization of software development shapes the organization culture by technologies and markets and cultural preferences of a manager/leader or employees (Juran 1988). National culture influences the extent to which leadership; teams and employee activities are socially valued and supported.

In 1870, anthropologist Edward Taylor, defined culture as that complex whole which includes knowledge belief, art, morals, law, custom and other capabilities acquired by a man as a member of society. Hofstede (1991), an expert on cross cultural differences and management, defined culture as, 'the collective programming of the mind which distinguishes the members of one human group from another...

Culture, in this sense, includes systems of values; and values are among the building blocks of culture .Values are basic convictions that people have regarding what is right and wrong, good and bad, important and unimportant. The values are learned from the culture in which individual is reared and they direct the persons behavior.

Our study focuses on cultural differences in the work place of software development organizations, especially in United States and India. Hofstede (1991) in his book 'cultures and organizations: software of the mind focuses on IBM studies involving 1,

16,000 employees in 70 countries and 3 regions, tried to analyze the influence of different national cultures. The findings demonstrate that: work related values are not universal.

Kluckholm and Strodtbeck's (1961) theory based on the 'patterns of behavior' in different cultures is showed in Table 3 - 5. The researchers distinguish and compare cultures on various dimensions and the findings are (1) America is a nation with mixed orientations. They are optimistic about other people's motivations and capacities. Here conflict is not disapproved of and differences in views are encouraged. (2) Indians have an orientation to work in harmony, believe that there should be peace between man and nature. There is a desire to avoid conflicts. They believe that destiny and god control everything. Indians are more philosophical and spend more time in abstract thinking. Life is derived from birth, age, sex, family and social connections more than through one's achievement. Regarding the case of temporal focus of human activity, it is believed in the United States that better fate can be planned and controlled. In India, people base their decisions on lesson learned from the past.

**Table 3 – 6 Kluckholm F and Strodtbeck (1961) theories based on the 'patterns of behavior'**

Sl. No.	Orientation of people	Pattern of behavior	India	US
1	Nature of people	Good, evil, mixed	Mixed	Mixed
2	Relationship to nature	Dominant, harmony	Harmony	Dominant
3	Relationship to individual	Hierarchical, collectivist	Hierarchical	Collectivist
4	Modality of human activity	Doing, being, containing	Being	Doing
5	Temporal focus to human activity	Future, past, present	Future	Past
6	Conception of space	Private, public mixed	Public	Private

Strodthbeck's formulations of cultural differences are regard for rules or relations individualism, emotions, and societies, where as Trompenaar's (1993) work considers attitudes to time, relations to nature as formulations of cultural differences. His study proved that in cultures, which are more particularistic in countries like India, people see relationships as more important than applying rules according to friendship and kinship relations. This has implications for recruitment and promotion policies in organizations in some Asian countries, which are different from practices in countries such as the United States.

A study of the personal value systems of over 2500 managers in India, United States, Australia and Japan supported the assumptions that despite the value differences among managers in these countries and value diversity within each county, there is a common pattern of translation of values into behaviors across countries. While pragmatists have economic and organizational competence orientation, moralists exhibit a humanistic orientation (England 1975). The implications of such values to the management of people in Indian organizations appear be that

- (a) Indian managers are more responsive to human consequences of their actions
- (b) They are more influenced by positions and approaches, which utilize philosophical and moral justifications
- (c) They are more responsive to internal reward and controls
- (d) Change in moralistic managers in India is likely to be slower and more difficult.

Another study undertaken in 1970's (Smith and Thomas 1972) of cross-cultural attitudinal differences between American and Indian managers identified the flowing differences in area of authority and influence:

- a. Indian managers at both middle and senior levels in organizations, profess a belief in group based participative decision making, but have little faith in the capacity of workers for taking initiative and responsibility.

American managers on the other hand place a relatively higher faith in the capacity of individuals to take responsibility and a lower faith in-group oriented decision-making.

- b. In contrast, to American managers, Indian managers favor labor and government interventions in the affairs of the organization.
- c. Middle level managers in India espouse a greater belief in change and are less conservative than their American counterparts at this level.

Richard (2000) in his book cross culture business behavior has put in to workable categories in four cultural categories showed in Table 3 – 6.

**Table 3 – 7 Cultural dimensions in India and United States**

<b>Serial Number</b>	<i>Cultural Dimensions</i>	<b>Team behavior in India</b>	<b>Team behavior United States</b>
1	Dealers/relationship	Dealers	Relationships
2	Formal, informal	Informal	Formal
3	Rigid time, Fluid time	Fluid	Rigid time
4	Expressive, Reserved	Reserved	Expressive

Laurent (1986) divides the employee team in workplace of an organization in to various societies, which are shown in Table 3 – 7. He divided the culture in to four dimensions. Societies are individualist, collective and masculine. In individualist societies tasks are more important than relation, in the collective societies relations more important than task and the masculine societies, values are based on material success.

**Table 3 – 8 Laurent division of cultural dimensions in to societies: US verses India**

<b>Serial Number</b>	<b>Cultural Dimensions</b>	<b>Team behavior in India</b>	<b>Team behavior US</b>
1	Individualist societies, Tasks are more important than relationships	No	Yes
2	Collective societies, relationships are important than task	Yes	No
3	Masculine society, values are based on material success	No	Yes

Allison and Hayes (1988) undertook cross-cultural differences between the United States and Indian managers in two broad learning styles: an analysis orientation and an action orientation. Results show that the US managers were action oriented and the Indian managers were analysis oriented.

Sahay's (2001) study noted that the Indian software analysts were trained in the rigorous International Standards Organization accredited methodology. The effect of globalization involved the disembedding of Western-derived methodology into the Indian context where it was embodied by the Indian developers. Project management expertise is scarce, because the industry is still young in India and large-scale projects where project managers are trained are still relatively rare. Effective project managers should have the following skills: communication, organization, team building and the leadership skills (Mithas 2007).

Sahay & Brian (2001) studied the bi-directional effects of globalization. This was demonstrated in their case study of software outsourcing to India by US firms. In this case, the Indian programmers are seen to bring in structured software development methodologies like the "waterfall methodology". The US approaches to software



development typically do not involve extensive documentation but rely more on discussions and personal face-to face contact.

Walsham (2002) point out that Indian managers and developers tend to be members of different social systems arising from both work- and non-work related systems such as intellectual groups, local community and family. These systems have various rules and resources embedded within them which managers and developers draw upon in their process of creating agency and making action mutually intelligible, which in turn can potentially either reinforce or change social structure. Often these rules and resources drawn upon to form these systems are conflicting, for example the work norm of efficiency would clash with the family norm of helping a relative. There is thus constant tension and contradiction in the creation and articulation of agency.

Norms of hierarchy often seen in Indian family relations and the caste system are structural conditions that can be interpreted as being drawn upon by Indians both implicitly and explicitly in developing agency. The caste system has contributed to value systems relating to status, power and relationships. Partly as a result, social relations are often seen to be hierarchical amongst Indians and people show status consciousness. In India, social relations tend to exist between groups of a particular social standing which tends not to be as rigid in Europe or US.

According to Nicholson and Sahay (2004) the effects of globalization are often discussed in terms of the western culture and management methods have in other cultures. The “traditional skilling” of Indians, reflects the emphasis on discipline in Indian schools with traditional rote learning approaches and mathematical skills forming a large proportion of the curriculum. As a result, generally Indians tend to be mathematically adept and disciplined in their thinking.

Nicholson (2001) noted that hierarchical structuring is so ingrained in India that it is often easier to work in a superior subordinate role than as equals on contractual terms. The Hindu virtues of contentment, absence of desire and stability tend to oppose the dynamic striving for success and unlimited consumption.

Nidumolu (1993) states that, in Indian work relations, the superior is seen to be “kind” and the subordinates “submissive”. This perception of the Indians feeling a desire to please may be related to the need for accommodation but also a sense of duty to the family and one’s superiors. The study also says that a cultural difference in India and United states in the point of education of the skills trained in Indian and US schools differ in organizational skills and management skills.

Elashmavi (2001) findings show that certain traits - both positive and negative are common to all cultures. Other behaviors are culture specific. Positive behaviors such as trust worthy, encouraging, effective communication, and team-building capabilities are universally acceptable. Negative traits that have universal application are egocentricism, non-cooperation and authorization. Table 3 – 8 shows the universally acceptable traits and behaviors that are culture specific.

**Table 3 – 9 Leadership behaviors and their acceptability from 60 countries**

Universally acceptable traits and behaviors		Culture specific behaviors
Positive	Negative	
Trustworthy	Loner	Group orientation
Just	Asocial	Self protectiveness
Honest	Not cooperative	Participative skills
Plans ahead	Non explicit	Humaneness
Encouraging	Egocentric	Autonomy
Positive	Ruthless	Charisma
Good bargaining	Dictatorial	Humility
Dynamic		
Motivator		
Confidence builder		
Dependable		
Intelligent		
Decisive		
Win win problem		
Skilled administrator		
Communicator		

Informed		
Team builder		

The studies so far in the literature discussed on the various cultural differences in countries India and US .The cultural differences in turn affect the values of the employees at the work place and ultimately the behaviors of developers that work in teams in software organizations in these countries. The various cultural dimensions, the way it affects the team effectiveness in software organizations in India and US are also discussed.

The next section reviews the variables in this study (1) Leadership (2) Motivation (3) Communication that are the skills of the software development team which are depicted from the chapter 2 that determine the software quality in organizations.

#### 3.3.3.1 Leadership

Members of different cultures in India and United States would respond to different leadership traits in their intellectual interactions. Narendra (2003) explores the lack of leadership and motivational managerial skills in Indian software industry

Zigurs (2003) showed a significant observation upon the leadership skills of the managers in India. The managers in Indian software development organizations are members of different social systems which are work and non work related members of different social systems such as intellectual groups, local community and family whose rules are conflicting-- clash with the family norm of helping. Their leadership abilities have a clash with the managers in the /United States where the family norms are never indulged in the work culture here.

#### 3.3.3.2 Motivation

Effective and appropriate management involves motivating people to go in to a work usually in medium to large goal directed organizations that often separate from their family, leisure and community life. In the organization motivating people to work hard and employ skills knowledge and attitudes varies from culture to culture (Jackson 2004)

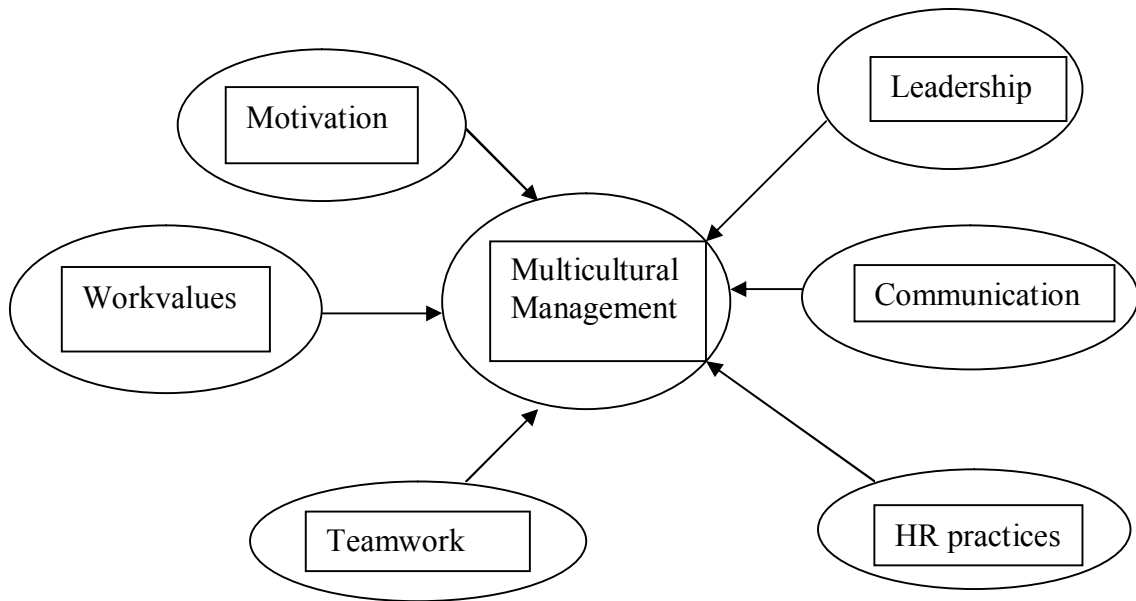
Managers in India were critical of the Indian system of promoting software programmers to managers based on seniority rather than on proven managerial ability which weakened their project management skills (Sahay 2001; Walsham 2002; Tracy 2007; Thomson 2007). In the same study it was noted that the work culture of the US managers differ in their motivational ability.

In the United States, the enjoyment-based intrinsic motivation, namely how creative a person feels when working on the project, is the strongest and most pervasive driver of motivation (Karim 2003). The findings indicated that a majority of software teams in the United States are skilled and experienced professionals working in IT-related jobs, with approximately 70 percent being motivationally satisfied (Lakhani 2006).

#### 3.3.3.3 Communication

Cultural factors have a major impact on the communications skills of the people in India and United States. As the national culture differs in these countries, the communication skills too differ.

The Indians lack the skills to operate in multicultural environments (Gopal 2002). The dimensions of multicultural environment is shown in Figure 3 - 3. The study of dimensions of values in different countries is depicted in Table 3 - 9. There were a number of cultural and political issues that US managers perceive as irritants or barriers in the communication skills of Indians (Arora, 1999; Arunachalam 1999). One such issue is the apparent unwillingness of Indian software professionals to point out potential problems up-front, and in general, an unwillingness to say no for fear of offending the clients (Arora 2001). Another related weakness is the lack of familiarity of many Indian firms and professionals with the work culture and work norms in the West, and especially in the United States. Other cultural differences included the unwillingness of Indians to speak out in meetings, necessitating "offline" conversations to convey the needed information.

**Figure 3 - 3 Dimensions of Multicultural environment**

The success of an organization in terms of its productivity, employee satisfaction and minimal turnover rate depends primarily on the effective communication practices (Downs 1977; Galin 2004). The need for effective coordination, visibility, communication and cooperation are key variables for success (Carmel 2001; Karolak 1999).

**Table 3 – 10 Dimensions of multicultural environment in different countries  
(adapted from Farid Elashmawi, p.144)**

	<b>American</b>	<b>India</b>
Management styles	Leadership, Motivation, Communication	Persuasion, Communication, Functional group activities
Control	Independence Decision making Space, Time	Group harmony, Money
Emotional appeal	Opportunity	Group participation, company success
Recognition	Individual contribution	Group identity, Belonging to group
Material Awards	Annual bonus, Commission, profit sharing	Salary, Social services, Fringe benefits, Promotion through seniority
Threats	Loss of job	Out of family, caste groups
Cultural Values	Competition, Risk taking, Material possession, Freedom	Group harmony, Achievement, Belonging

#### 3.3.4 Team Size

Many studies have shown the importance of team size in determining the quality of the software. Researchers have found that a team size of four is the one which gives maximum productivity and quality. Team sizes that are lesser and greater than four give less efficient.

Team size is also important factor affecting productivity on short, one person projects are likely to be overwhelmed by individual skill differences across project staff members (Curtis 1981; Dickey 1981; DeMacro 1982). The projects of similar size exhibit similar productivity due to team size (Brooks 1993). Larger teams showed lower productivity due to higher interaction in the study. The size of the team itself is often argued to be an important factor (Carmel 2002).

### 3.4 Summary of Literature Review (Chapters 2 and 3)

The preceding chapter (2) and chapter (3) have aimed to review the pertinent literature that supports the relationships proposed in the conceptual model of this research. The tenet of this thesis is that further research that looks at the disparity of impact of team skills on software quality in Indian and United States needs to be undertaken due to:

- A. The critical position of technical and project management skills hold on a team's performance.
- B. The bearing it has on the evaluation of software quality
- C. The limited studies that have been undertaken in this area.

The chapter 2 summarizes the importance of team skills on software quality. With the support of the literature review (Judith 2007), the team skill that impact software quality is divided in to

- (1) The technical skills of the team members
- (2) The project management skills of the team members.
- (3) Team size

**The importance of technical skills in determining quality is dealt in chapter 2. Further, the technical skills are subdivided in to (a) Programming skill (b) Language skill (c) Application domain skill and (d) Hardware skill. The supporting literature discussions on these skills and its importance in determining the quality of the software are dealt (Krishnan 1998; Srikanth 1991). The project management skills of the team are characterized in to (a) Leadership skills (b) Motivation and (c) communication (Judith, 2007). The skills required in India and United States has been summarized in the Table 3 – 10.**

**Table 3 – 11 Summary of skills required in India and United States**

	<b>India</b>	<b>US</b>
<b>Technical experience</b>		
Language	High	Low
Programming	High	Low
Domain	Low	High
Hardware	Low	High
<b>Project management skills</b>		
Leadership	Low	High
Communication	Low	High
Motivation	Low	High
Size	-	-

Software development is globalized in to various locations (Hayes 2002). Globalization of software development effects team effectiveness (Catherine 2005). Globalization has led the disparity in team skills in various countries. The reason for the differences in team skills is due to three reasons.

- (1) Type of projects developed in various locations.
- (2) Migration of experienced professionals to US
- (3) Cultural differences in various nations.

The disparity in team's technical skills is due to the type of projects developed in various nations. This further is extended for the reason of migration of experienced professionals to developing countries like United States. The reason shows the differences in project management skills of the software team member is due to the



cultural differences of the software developers working on the team. These are the conclusions drawn from the literature.

Due to the limited empirical work in this research area, a considerable portion of the literature review has aimed to support the study's propositions by reference to research that has investigated relationships with comparable constructs.

Finally, the objective of this literature review has been to present evidence for further research in line with the relationships depicted by the conceptual model. These relationships and the ensuing propositions are further developed in the next chapter.

Essentially, it is considered that additional research for relationships depicted in the conceptual model is warranted since:

- A. There is essentially no empirical work in the quality literature that investigates the antecedents and consequences of team skills on quality of the software product in software development organizations. This includes the influence of technical and project management skills of software development team.
- B. Existing research in to the relationships that may exist between role of team member and software quality reports equivocal findings.
- C. Limited studies exist within the germane literatures that pertain to research concerning the relationship between team's quality and software quality.
- D. There are no studies that investigate the relationships between the noted variables in one model.
- E. Limited research exists within the literatures that pertain to research concerning the impact of disparity in team skills in countries India and United states.

## CHAPTER 4

### PROPOSITION DEVELOPMENT

#### 4.1 Introduction

The purpose of this chapter is twofold. First, it further develops the initial conceptual model of the thesis. Second, from this refined model a series of research propositions are developed that form the basis of this study.

The need for an examination of the antecedents of software quality in organizations appears necessary for the reason that team behaviors influence quality of the software product in software development organizations. Further, there is limited empirical research that addresses the role of teams in software development and a comparative study of these factors in countries US and India. The variables under investigation in the conceptual model as antecedents of software product quality were selected on the basis of:

1. The existing literature that discusses, by way of conceptual argument or general discussion, various aspects of our model
2. Empirical data that has found comparable variables to hold significant relationships (e.g. Krishnan 1998; Ramasubbu 2007)

Therefore, the primary purpose of this thesis is to empirically test and extend our knowledge of the determinants of software quality and comparing the impact of team skills on software quality for Indian and US teams. A key aspect of this thesis is the extension and empirical testing of the relationships of team factors on software quality proposed in the early literature (e.g. Krishnan 1998), concerning the relationship between software team skills and quality of the software developed. A further goal of the thesis is the development of prepositions as a result of empirical analysis of the disparity of the team skills in the countries India and United States.

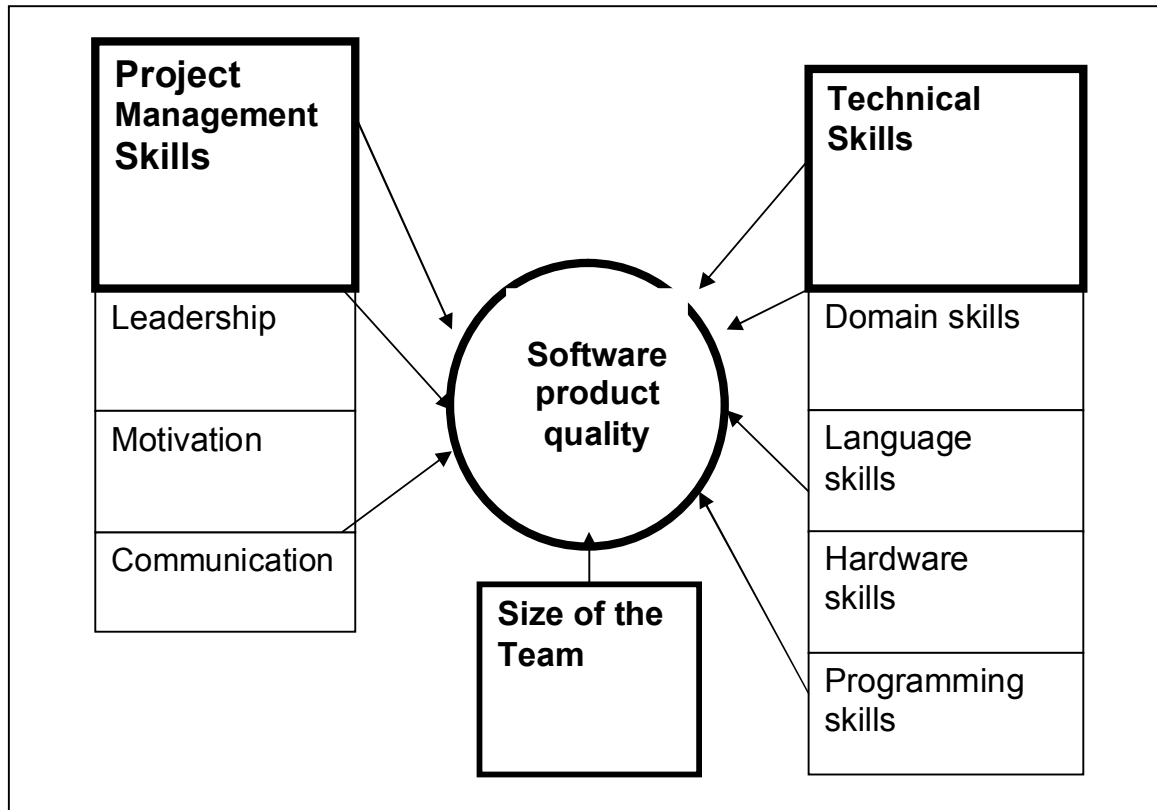
#### **4.2 Refinement of conceptual model incorporating multiple skills for team personnel.**

Refinement of conceptual model incorporating multiple skills of the team personnel constructs as two notions of skills i.e., technical skills and project management skills of the software developers in the software development organizations. While Krishnan did not specifically indicate whether the propositions related to technical or project management skills, his propositions referred to the skills of the team had focus on the quality of the software team. Therefore, it is considered that Krishnan's (1998) propositions generally depicted the relationship between team personnel's technical skills and software product quality. Hence, considering this literature, the propositions are used as the basis for the development of the relationship between the thesis variables, perceived technical skills and together with project management skills of the team personnel.

The higher the team skills of the personnel, the higher would be both the software product quality. i.e.; team's quality leads to increased software product quality (Krishnan 1998; Justin 2006).

The preceding literature review found support for the influence of team skills relative to software development inside the organization. However, few studies discussed the disparity in team skills relative to software development. , save Arora's (2001) characterization of relative skills of the team on software quality .This study attempts to extend the notion of software development team skills by investigating relative team skills from two perspectives. These are identified as perceived technical skills (Skills of the team personnel relative to product development) and perceived project management skills (i.e., skills of team personnel relative to project management). These constructs have replaced the team skill variable in the initial conceptual model and are depicted in figure 4 - 1, conceptual model – proposition development. However, team's software quality is defined as a composite measure in the conceptual model as it is considered that there is more support for relationships between the general software team quality, and software product quality (Krishnan 1998).

**Figure 4–1 Conceptual model – proposition development**



Support for the approach of separating team's quality in to two perspectives can be also found in the literature. Krishnan (2004) argued that the team skills in congruency of the individual should be interpreted across several team skills hierarchies rather than only one. While the ascribed team skills of the personnel incorporates the individual characteristics, the team personnel brings to the software quality encounter, the achieved team skills of the personnel can be seen to incorporate two levels: first the skill or experience attained by the individual for the product development, and second ,the skill or experience of the team member to manage a software project i.e. the project management skill of the team personnel .The former is considered to portray perceived technical skills of the team and latter, perceived project management skills of the team ..

Additionally, the conceptual model describes relationships proposed to exist between other variables that have been noted and supported in the literature review, specifically, variables on team's project management skills - communication, leadership and motivation. Communication has been selected for inclusion in the thesis due to the logical association with the software quality and the literature that supports this relationships (e.g. Javed 2004). The variable leadership and motivation has been selected for inclusion in the thesis due to the literature within the service field of study that highlights the influence leadership and motivation skills has on team's quality (Liteky 2004).

Support for the inclusion of perceived project management skill of the team personnel and teams quality in a model were found in Ramasubbu and Mithas (2007). They considered that the investigation of team's skills on quality required the assessment of a number of other variables among which included the communication skill, leadership and motivational skills of the team personnel. The conceptual model of the thesis addresses these issues as it includes the team personnel skills which reflect communication, leadership and motivation.

Further to the extension of propositions concerning software team skill initially suggested by Krishnan (1998), the conceptual model depicts the formalization of propositions concerning the technical skills of the team and software quality of the team personnel in a globally located environment (Catherine 2005). Finally, as a result of these propositions and the support of existing literature, the model describes disparity in the impact of technical team skills and the software quality of team personnel in United States and India.

#### **4.3 Proposed relationships/disparity between the determinants of software quality in United States and India**

The remainder of this chapter will now present support for the propositions of this thesis. First, support is presented for the propositions concerning the project management skills and technical skills of the team personnel as an antecedent to team's quality and software product quality in United States and India. Second, support is presented for the proposition concerning the disparity of impact of team

skills on software team's quality and software products quality in United States and India. Finally support is presented for the propositions concerning the differences on the effect between software team project management skills, software team's technical skill and software product quality in these countries. (Refer Conceptual Model –Proposition development, for a graphic depiction of the directionality of each of the propositions within the conceptual model.

#### **4.3.1 Skills of the team personnel on software team's quality differ in India and United States**

Krishnan's (1998) primary discussion revolved around the relationship between the technical skills of the team member and quality of the software project. In each of his propositions, team's quality was considered a single construct with technical skills defined as two constructs. However, his characterization of team personnel skills incorporated two notions of team skills, domain skills and personnel capability. While Krishnan did not specifically indicate whether the propositions related to technical or project management skills, it is considered that Krishnan's (1998) propositions generally depicted the relationship between team personnel's technical skills and team's quality as a whole. But there is a gap in the literature extending the findings of Krishnan (1998) concerning the team member's project management skills and its influence on software quality among the various countries which are globally located especially in developed countries US that outsource their software development activities to developing in India. Our thesis is a direction towards this.

Software development is a globally sourced commodity (Pozzebon 2005; Gambardella 2004; Herbsleb 2001; Hayes 2002; Hoffmann 2001; Ebert 2001). Catherine (2005) examines that geographically dispersed teams have been prevailing in many service-based organizations and significantly effects team effectiveness.

Tschang (2001) investigated that the skills most in demand in India will be areas as business application of software development, project management and specific technical skills. Judith (2007) examined the Indian software industry team skills as: basic technical skills as coding and programming and language skills, project management skills as leadership, motivation and communication capability (Hertel

2003; Gopal 2002; Jones 1986). The key challenge facing the software development industry is identifying critical skill sets (Ethiraj 2005; Lee 1995). The next section reveals the various reasons for the disparity in skills of software team development in the countries India and United States

#### ***4.3.1.1 Technical experiences of team members on software quality varies in India and US***

There are a variety of reasons cited in the literature that experiences of team members for achieving software quality in India and the United states differ substantially (Demacro 1999; DCosta 2002; Sinclair 2000). A major reason for the differences in quality of software between the two countries are the types of software projects done in the US and Indian teams. As we discussed, Indian software firms largely provide services such as maintenance, porting & enhancement rather than products neither technologically very sophisticated nor critical to their business. Software development consists largely of low-level design, coding, and maintenance services (Arora 2001; DCosta 2002). As a result, software development teams in India are bound to work on testing of the existing software and to some extent maintenance of old UNIX based software. These software project teams which tasked with maintenance services, the developers need a larger knowledge of programming skill and language syntax. Hence, quality of the software project teams could be achieved only through the team members with good experiences in programming skill and language skill.

Proposition: 1.

The extent to which team member language experience impacts software quality is higher in India than in US.

“This would mean that a highly language experienced developer could enhance the quality of the software to a much greater degree while working in a project in the India, than if the same programmer were to be working in a software project in the United States”

Proposition: 2

The relationship between programming experience of team members and the quality of the software produced is stronger in India compared to that of the United States

The firms in United states do not want to invest in-house capability in areas outside their core-competence (such as developing applications for old computing platforms) and do want to free their in-house IT staff from mundane maintenance tasks for more creative projects (Arora 1999). Further, projects done are mainly design and development of new applications .The US firms use there scarce in-house IT staff for higher value added work, such as design and to develop new types of applications. Requirement analysis and high-level design is typically done either in-house or by US based consultants. Indian software project teams lack business development skills and product skills (Tschang, 2001).

The Indian software industry experiences shortage of experienced software professionals. India has an abundant supply of entrepreneurs who recognized and responded to the opportunity that the IT revolution in the West represented. Though, India graduates about 155,000 engineers of various sorts, and another 200,000 diploma holders per year, about 60,000 of these enter the IT sector. Despite the apparently large stock of human capital in India, NASSCOM claims that by year 2010, demand will outstrip supply. Hence, advanced technical skills and domain skills were lacking and absence of experienced people may be due to constant navigation (Gibbons 2004).

Hence we form the hypothesis

Proposition: 3

H3: The degree to which the domain experience of team member's impact software quality is higher in the United States than in India.

Many lacking entrepreneur ability (Tschang 2001). The Indian software experience does not fit the East Asian model of growth. The most important factor is that the



Indian experience was not based on manufacturing, but rather more on services and intellectual work (Udell 1993). India's capability for manufacturing computer hardware and other electronics has been and still relatively poor (Tschang 2001).

Large teams can be assembled at much shorter notice and projects can be terminated with much greater flexibility. A number of environmental factors affects the labour hours required to complete a project and hence the productivity of the software maintenance effort which includes the hardware tools (Thandhani 1984; Lambert 1984; Boehm 1981; Jones 1986). So forth, a varied deal of hardware skills needed to attain software quality which will fulfill the different project based behaviors in the United States and Indian teams.

Proposition: 4

The relationship between team member hardware experience and quality of the software produced is stronger in the US compared to India

#### ***4.3.1.2 Project management skills of team members on software quality varies in India and US***

Various project management skills of team members like motivation and leadership, communication, coordination is found to be an important factor in assessing software quality (Sackman 1968; Chrysler 1978 and Larman 2003). Judith (2007) defines project management skills as leadership, motivation and communication. Since the software companies face severe market pressure from their competitors, this environment demands entrepreneurial spirits, long work hours and determination from the team members. As a result, software companies often attract some specific personality types that are risk seeking and exhibit individualism, innovation and creativity (Futrell 2001; Carmel 1998; Castels 1994).

##### ***4.3.1.2.1 Leadership skills of the team members on software quality varies in Indian software teams***

Effective project managers should have the following skills: communication, organization, team building and the leadership skills (Zigurs 2003). Edgemon (1995)

explores the lack of leadership and motivational managerial skills in Indian software industry. The Indian project managers were generally lacking the ability to lead a team, and one reason may be due to constant migration. Walsham (2002) point out that Indian managers and developers tend to be members of different social systems arising from both work and non work related members of different social systems such as intellectual groups, local community and family. These systems have various rules and resources embedded within them which managers and developers draw upon in their process of creating agency and potentially either reinforce or change social structure. Often the rules and resources drawn upon to form these systems are conflicting, for example the work norm of efficiency would clash with the family norm of helping.

Western culture often exhibit high intelligence, dominance, self confidence, integrity, energy levels and task relevant knowledge (Charles 2000). Some of the traits thought to be important to leadership are culturally determined like authorization and dominance seem to be more acceptable in the developing countries like India (Walsham 2002; Mithas 2007) than in developed countries like the United States. Western explanations of leadership and motivation (e.g. equity theory, Adams 1965; Weinberg 1986) depend on the assumption that people will act in their own self-interest and there is no superior – subordinate relationship and rather they treat the subordinates as friends. There fore we should expect that members of different cultures in India and United States would respond to different leadership traits in their intellectual interactions.

Proposition: 5

H5: The Extent to which the leadership skill of the team member impacts software quality is higher in India than in US.

#### *4.3.1.2.2 Lack of motivational skills in Indian software teams*

Various reasons behind the lack of project management skills in Indian teams were seen in the literature (Feller 2005; Arora 2001; Shah 1995). Most of the managers were critical of the Indian system of promoting software programmers to managers based on seniority rather than on proven managerial ability, which weakened their

project management skills (Sahay 2001; Walsham 2002; Tracy 2007; Thomson 2007). Indian firms, on the other hand, cited this practice as a way of providing a career path to their professionals and a major part of their attempts to hold down employee attrition (Arora 2001). Additional dissatisfaction is about the low productivity. For instance, managers at the electronics and telecom firms also mentioned that they found that they needed to assign more engineers to the same task in their Indian subsidiary than would be assigned in the U.S. A very large fraction of the managers considered employee attrition a big problem and wanted their Indian suppliers to tackle it quickly. The revenue per employee is probably a far better indicator of productivity in the software services industry (Dyke 1997; DCosta 1998). In this industry, equipment is not very expensive compared to the other service industries in India and firms do not differ very significantly in terms of the capital employed per employee. Thus, revenue per employee provides a reasonable motivational measure of productivity (Green 1989). In the United States, the enjoyment-based intrinsic motivation, namely how creative a person feels when working on the project, is the strongest and most pervasive driver of motivation (Lakhani 2006). The findings indicated that a majority of software teams in the United States are skilled and experienced professionals working in IT-related jobs, with approximately 70 percent being motivationally satisfied (Lakhani 2006)

Proposition: 6

H6: The degree to which the motivational skill of the team member impacts software quality is higher in India than in US.

*4.3.1.2.3 Lack of Communication skills in Indian software teams*

The Indians lack the skills to operate in multicultural environments (Gopal 2002). There were a number of cultural and political issues that US managers perceive as irritants or barriers in the communication skills of Indians (Arora 1999; Tod 1995). One such issue is the apparent unwillingness of Indian software professionals to point out potential problems up-front, and in general, an unwillingness to say no for fear of offending the clients (Arora 2001) Another related weakness is the lack of familiarity of many Indian firms and professionals with the work culture and work norms in the

West, and especially in the United States. Other cultural differences included the unwillingness of Indians to speak out in meetings, necessitating “offline” conversations to convey the needed information.

Proposition: 7

H7: The relationship between communicational skill of the team members and quality of the software produced is stronger in the US compared to India.

#### *4.3.1.3 Team Size*

Another important team related factor is team size. The software engineering literature indicates significant benefits from small teams (Carmel and Bird 1997). Project size is also important in that the factors affecting productivity on short, one person projects are likely to be overwhelmed by individual skill differences across project staff members (Curtis 1981; Dickey 1981; DeMacro 1982). The projects of similar size exhibit similar productivity due to team size (Brooks 1993). Larger teams showed lower productivity due to higher interaction in the study.

Proposition: 8

H8: The relationship between the effect of size of the team and quality of the software produced is stronger in the US compared to India.

## **4.4 Summary**

This chapter has aimed to provide the argument for the development of the conceptual model as depicted in Figure 4 - 1, which proposes a number of interrelated relationships between variables that have as yet not been empirically tested as a whole. In doing so the discussion has sought to demonstrate support for direct and indirect relationships between the antecedents of language skills, programming skill, hardware skill, domain skill, communication skill, motivation and leadership skills, with the dependent variable of software quality and a comparison of these variables in the countries United States and India.

The propositions presented in this chapter will now be empirically tested. The following chapter (5) will outline the questionnaire design and data collection process for the study.

Chapter 6 and 7 will then present the research findings and chapter 8 will offer a discussion of the findings and managerial implications.

## CHAPTER 5

### RESEARCH METHODOLOGY

#### 5.1 Introduction

The previous chapters reviewed the relevant literature and outlined the conceptual model for this study. The research propositions to be tested and their development were so presented. This chapter will now discuss the research methodology applicable to this study. First, the research setting and the choice of data collections organizations as respondents are discussed. Second, the research design is outlined. This encompasses data collection, pre-testing, scale development and the research instrument. Third, the operational constructs of the design are described.

#### 5.2 Research design

Research design provides the answers to research questions while controlling for variance and gives us the framework for our study by suggesting the types of observations to make, how to analyze them and possible conclusions that can be drawn from the analysis (Kerlinger 1973; Belsley 1980). Survey research, specifically a mail questionnaire, was chosen to assess the propositions raised by this study (Berdie 1989). The survey research method is the most commonly used method for assessing employee role perceptions (Jackson 1985). The advantages of the survey research method include normality of surroundings for the sample respondents, thereby limiting responses from the artificial environments, easy replicability to verify results; generalisability of results from the sample to population; and relative accuracy of data, regardless of sampling error (Robson 2002). Disadvantageous of the method include time and cost issues; sources of sampling error and that respondents may answer in terms of preferred rather than actual behaviors (Kerlinger 1973).

This thesis used a three-stage approach in its research design, which included two pilot studies and the main survey (see Table 5.1 – Research design). Following well-accepted practice (Dillman 1978), the research instrument was comprehensively tested prior to its distribution. The pilot studies were undertaken prior to the main

study, both comprising a qualitative and quantitative phase (Holiday 2002). The qualitative component focus groups with the respondents and the qualitative component were an email survey. The focus groups had two purposes (Calder 1977). First, they allowed further item generation over and above the findings from the literature. Second; they allowed a pilot test of potential items in the scale. The focus group respondents were all from the same software development company. The final stage of the research was a questionnaire mailed to a large sample of the population. Demographics and the background information were collected in all mail surveys.

**Table 5 - 1 Research Design**

Stage	Study	Component	Dates
1	Pilot A	Qualitative Quantitative	July 2005 September 2005
2	Pilot B	Qualitative Quantitative	February 2006 May 2006
3	Main survey	Quantitative	July 2006

### 5.3 Data Collection

#### 5.3.1 Pilot A

Pilot A was directed at the scale development of the technical skills of the team members and a pretest of the perceived language, programming, hardware, domain measures (see Table 5.2 – measures tested in pilot studies). Development of the team skill perceptions measure followed Churchill's (1979) recommended procedure. This involved the generation of the items from the existing software quality literature and from focus groups held in the software organizations. A total of 500 self-completed questionnaires were placed in the personnel mailboxes of software developers, using a random skip procedure. It was requested that on completion, the survey should be mailed directly to the researcher in the postage paid envelope, which was supplied with each survey. A final sample size of 202 was used for the analysis. The team

members were predominantly male's .The mean age of software developers was 33 years, within a range of 22 years to 53 years. The mean number of years software developers has been working in software development had been was 7.7 years, within a range of 1 month to 15 years.

**Table 5 - 2 Measures Tested in Pilot Studies**

<b>Pilot study</b>	<b>Constructs pretested</b>
Pilot A	Language experience Programming experience Hardware experience Domain experience
Pilot B	Communication Leadership Motivation

### **5. 3. 2 Pilot B**

Pilot B was directed at several issues. The key issue was a pre-test of the project management skills of the team members in determining software quality (Ramasubbu 2007). This involved the application of Boehm 1987 scale.

The quantitative data was collected by way of a self-completed questionnaire, using a convenience sample of 100 software companies from the software organizations. A university researcher administrated the questionnaire in the team prior to software team members .In total, 15 respondents were removed from the data set as they scored 15 out of more out of a possible 20 on the test. The demographics of the final 85 respondents approximated the company population data.

## **5.4 Main survey**

The questionnaires were distributed to 500 software developers in various countries in India and United States. The questionnaire was sent to each of the software developers in teams as email attachments, using a random skip procedure generated



from the company's human resources database. The survey questionnaire is given in Appendix 3. The list of organizations responded to the survey is given in Appendix 1. Description of respondents participated in the survey is given in Appendix 2.

Following Dillman's (1978) recommendations, a comprehensive promotional approach was used in an endeavor to ensure a high response rate, given the variability of previous survey work. First, each questionnaire that was distributed included a covering letter from the researcher ensuring responses would be kept confidential (see Appendix 5 – covering letter).

Second, approximately two weeks after the survey was initially distributed the first follow - up email was distributed to the same personal mailboxes as had been the surveys, (Appendix 5 - A). This thanked those who had returned questionnaires and reminded those who hadn't that their cooperation would be greatly appreciated. Information was also given for obtaining additional questionnaires in lieu of misplaced ones.

Third, a further two weeks later on, a final reminder was once more posted to the same personal mailboxes, this time in a different form so as to be recognized as a different (see Appendix 5 – B Second follow up email). This thanked the team members for returning the survey and reminded them that they had contributed to a worthwhile fund. It also extended the closing date by two weeks to enable those who had not completed the survey to do so. Information concerning misplaced surveys was again given.

In total, the team members had approximately took six months to complete and return the surveys. An analysis of response bias was not undertaken due to the inability to accurately assess when respondents received their surveys and therefore how long they took to return their surveys. This is due to the fact that it is not possible to ascertain at which point during the six month a period and individual team member may have received the survey.

## **5.5 Scale Development of construct Measures**

### **5.5.1 Application/ Domain experience**

This variable measures the level of experience of the software team members in the specific application domain of the product. The numbers of years of domain experience is drawn from the personnel records. For missing data, the respective product managers provided estimates. The domain experience score for each team was the average of the level of experience of individual team members, measured in number of years.

### **5.5.2 Language experience**

This variable measures the level of experience of the software team in the specific programming language used in developing the product. Since most of the products in this analysis were developed in the same language, this data was relatively easier to collect from the personnel records. The level of experience for each team was measured by using the average of the level of experience of individual team members in years. When full personnel records were not available, inputs from the respective product managers were used to compute the scores.

### **5.5.3 Hardware experience**

This variable measures the level of experience of the software team in the hardware used in developing the product. The construct was operationalized using seven statements that mirror certain context specific issues related to hardware used for software development. The respondents were requested to express their degree of agreement with these statements on a five point scale anchored between strongly agree and strongly disagree. The items used in the scale were developed from the insights gained through pre-study interviews conducted among the software engineers and project leaders in the software organizations. The arithmetic mean of the responses obtained against the items was used for further analysis.

#### **5.5.4 Leadership**

The variable measures the project management skill, leadership on the team productivity. This is a measure of the team factor base on software quality. The team leaders were requested to express their degree of agreement with these statements on a five-point scale anchored between strongly agrees and strongly disagrees. The items used in the scale were developed from the insights gained through pre-study interviews conducted among team leaders and team members. The arithmetic mean of the responses obtained against the items was used for further analysis.

#### **5.5.5 Motivation**

The variable measures the effect of motivational skills of the team leader on quality. This is a measure of the team factor based on software quality. The team leaders were requested to express their degree of agreement with these statements on a five-point scale anchored between strongly agrees and strongly disagrees. The items used in the scale were developed from the insights gained through pre-study interviews conducted among team leaders and team members. The arithmetic mean of the responses obtained against the items was used for further analysis.

#### **5.5.6 Programming capability**

This variable measures the level of programming capability of the software team in developing the product. The measure was computed from the responses obtained from the team lead and software engineers. The respondents were requested to express their degree of agreement with these statements on a five point scale anchored between strongly agree and strongly disagree. The arithmetic mean of the responses obtained against the items was used for further analysis.

Other team related dependent variables such as the effect of communication on quality, team size was computed based on inputs from product manager and members of the software team.

## 5.6 Summary

This chapter has described the research and questionnaire design and data collection for this thesis. Two pilot studies and a main survey were undertaken to affect this research. Respondents in the main survey were randomly selected and assured total anonymity.

Most of the construct measures required initial development or further development of scales for this study to proceed. Specifically, Leadership skill was a completely new scale developed for this study, reported in Section 5.4.4 - development of constructs measures. The further development of the team skills on software's quality was also reported in this section. The two software quality scales, perceived technical skills and perceived project management skills, were also designed specifically for this study and were accordingly pretested in the pilot studies. Validity and reliability was discussed for each measure. The next chapter (6) will now present the preliminary results of the data from the main survey.

## CHAPTER 6

### DATA ANALYSIS AND RESULTS

#### 6.1 Introduction

The previous chapter outlined the research methodology for this study. This comprised a review of the research context and the suitability of the study's respondents, from countries United States and India. An overview of the research design was presented. This included data collection, scale development, the pre-testing of the survey instrument over two pilot studies encompassing qualitative and quantitative stages, a description of the research instrument and the operational definitions of the research variables.

The aim of the current chapter is to present the preliminary analysis of the data and testing of the propositions developed in Chapter 4. This chapter commences with a discussion of the sample, the comparison of the sample data to the population data and the descriptive statistics of the sample data. Following this, the results of construct validation are presented, which include exploratory factor analysis, reliability analysis, and analysis for discriminant. The next analysis addresses the results of the proposition testing by way of ANOVA analysis.

#### 6.2 Sample Profile

##### 6.2.1 Final Profile

From the 1000 surveys that were distributed to software team members in both Indian teams, and US 205 Indian and 158 US were returned. Of the 205 and 158 surveys that were returned, 47 and 33 deleted from the sample since they had missed sections in the questionnaire. This reduced to the final sample to 158 Indian and 125 US samples.

##### 6.2.2 Descriptive statistics

**The discussion in Section 5.6, operational definitions of research variables outlined the rationale for the utilization of the various scales used in this research. Following is a brief overview of the descriptive statistics for the variables investigated in this (see – Table 6-3 – Descriptive statistics).**

**Table 6 -1 Descriptive statistics for the sample in United States**

	<b>Mean</b>	<b>Standard Deviation</b>
<b>Variable name</b>	<b>US</b>	<b>US</b>
Language experience on quality	3.917	1.0048
Hardware experience on quality	3.696	.9773
Application experience on quality	3.728	1.1871
Programming experience on quality	3.920	1.1681
Leadership on quality	4.112	1.1014
Motivation on quality	4.240	.8926
Communication on quality	4.400	.7071
Team size on quality	3.976	1.1177

The above table shows the descriptive statistics for the sample variables in the United States. From Table 6 - 1, the mean value of showed highest for the variable communication, which shows that maximum respondents in the United States chose communication as a team variable that significantly affects the software quality. The variable motivation was the next that the respondents chose a determinant of software quality. Leadership was the sample variable that had the highest respondents in the third place. These results show that there is a major significance for the project management skills in the United States.

Many of the respondents choose team size as a variable that determine software's quality in the project teams of United States. This support the literature that more higher or the more larger the team size have a negative effect on the quality of the software teams.

The next sample variable that showed prominent value for the mean was the programming experience of the software developers. Language skills of the software

developers had almost the same place of respondents' .The skill of the software developers in their application domain too had an outstanding group of respondents. The respondents choose the hardware skill of the US teams as the factor least determine the software quality of the team.

In the case of standard deviation of these variables, application experience had higher respondents showed .The next team factor, which the respondents choose was programming skill of the team members on software development. Thirdly, Team size was the factor of team that respondents agreed to have a highest deviation. The variable Leadership skills of the software project managers had a standard deviation in the fourth place. The opinions of the software developer respondents had the standard deviation 1.0101 for Language skill. The standard deviation of Hardware skill had the next place of respondents. Motivation and communication had the least value of standard deviation as 0.8926 and 0.7071.

**Table 6 – 2 Descriptive statistics for the Indian sample**

<b>Variable name</b>	<b>Mean</b>	<b>Standard deviation</b>
Language experience on quality	3.634	1.0101
Hardware experience on quality	3.189	1.2474
Application experience on quality	3.879	1.0117
Programming experience on quality	3.987	.8886
Leadership on quality	4.278	.7890
Motivation on quality	4.291	.7341
Communication on quality	4.113	.8444
Team size on quality	3.911	1.3132

The above table shows the descriptive statistics for the sample variables in the United States. From Table 6 - 2, the mean value of showed highest for the variable leadership, which shows that maximum respondents in India chose communication as

a team variable that significantly affects the software quality. The variable motivation was the next that the respondents chose a determinant of software quality. Communication was the sample variable that had the highest respondents in the third place. Here too the results showed very similar to the US sample that there is a major significance for the project management skills in Indian sample.

Many of the respondents choose team size as a variable that determine software's quality in the project teams of India. This support the literature that more higher or the larger the team size have a negative effect on the quality of the software teams.

The next sample variable that showed prominent value for the mean was the programming experience of the software developers. This is also similar to the case of US sample. Domain application skills of the software developers had almost the same place of respondent's .The skill of the software developers in their language domain too had an outstanding group of respondents. The respondents choose the hardware skill of the US teams as the factor least determine the software quality of the team.

In the case of standard deviation of these variables, hardware skills had higher respondents showed .The next team factor, which the respondents choose was team size of the software development project team. Thirdly, application skill of the team members was the factor of team that respondents agreed to have a highest deviation. The variable language skills of the software project technical team had a standard deviation in the fourth place. The opinions of the software developer respondents had the standard deviation 0.886 for programming skill. The standard deviation of communication skill had the next place of respondents. Leadership and motivation of the project team members had the least value of standard deviation as 0.7890 and 0.7341. Table 6 - 3 presents summary statistics of these variables in the United States and India.

**Table 6 - 3 Summary statistics of these variables in the United States and India**



Variable name	Mean		Standard deviation	
	India	US	India	US
Language experience on quality	3.634	3.917	1.0101	1.0048
Hardware experience on quality	3.189	3.696	1.2474	.9773
Application experience on quality	3.879	3.728	1.0117	1.1871
Programming experience on quality	3.987	3.920	.8886	1.1681
Leadership on quality	4.278	4.112	.7890	1.1014
Motivation on quality	4.291	4.240	.7341	.8926
Communication on quality	4.113	4.400	.8444	.7071
Team size on quality	3.911	3.976	1.3132	1.1177

The descriptive statistics of each of the dimensions of software quality are outlined in Table 6 - 3. This shows that leadership is undertaken far more frequently than any of the other team behavior dimensions in India and communication is undertaken far more frequently than any of the other team factor dimensions in United States.

The respondents showed a highest mean for language experience in the United States than in India .The results showed 3.917 for US and 3.917 for India. The hardware skills were the variable that the respondents showed higher in the United States than in India. Considering the value of application skills, the Indian sample respondents choose more than the United States team members. This was the same in the case of programming skill, leadership and motivation. All the three variables mean were higher for India and lower for the sample population United States for role of these factors in determining software quality. The variable communication and had a fewer respondents in the Indian sample.

The project management skills of the Indian sample that respondents chose were lower as the mean were lower. This was similar to the case of Team size the respondents showed the impact of team 's size on determining software quality as higher for the US sample and lower in the Indian sample.

### **6.3 Construct validation**

As the constructs in this study, have had minimal if any prior measurement undertaken, the recommendations of Gerbing and Anderson (1988) were followed in the assessment of the unidimensionality of the measurement constructs. Traditionally, the development of measurement scales had relied on one or more of coefficient alpha, item total correlations and exploratory factor analysis.

Therefore, following Gerbing and Anderson (1988) and in preparation for the analysis of the data the testing of the study's propositions, the following procedures have been undertaken. First, the expected dimensionality of the constructs is discussed. Second, exploratory factor analysis is undertaken on each of the constructs to investigate proposed dimensionality. Third, reliability of scales is assessed through Cronbach alpha (Cronbach, 1951). Fourth, discriminant validity is assessed through factor analysis.

#### **6.3.1 Dimensionality of constructs**

Most of the constructs in this study have had minimal research undertaken to determine their dimensionality. While role conflict has previously been extensively researched, the reported dimensionality of the other constructs has been very limited. The expected dimensionality of the measured constructs follows:

##### **6.3.1.1 Software quality**

Krishnan and Justin (2006) found 4 dimensions in their work. However, since Pilot B (Ramasubbu, 2007) reported 8 dimensions in their study of team factors on software quality, an 8-dimension result is expected.

##### **6.3.1.2 Language experience**

DCosta (2002) measured language experience they did not report dimensionality. In the absence of further evidence, a unidimensional construct is expected.

#### 6.3.1.3 Programming experience (4 items)

Programming experience was conceptualized as a distinct construct (). While there has been debate over the dimensionality of the two scales, there appears to be support for their convergent and discriminant validity (Sinclair 2000). Since, multiple dimensions within programming experience are not reported in the literature; a uniform dimensional result is expected.

#### 6.3.1.4 Application experience (4 items)

Since Pilot study A reported 4 dimensions for this construct, the same result is expected in this study.

#### 6.3.1.5 Hardware experience (4 items)

It would appear that measurement of this construct has previously not been reported in the literature. Since both pilot studies of this research found hardware experience to be a unidimensional construct, the same result is expected in the main study.

#### 6.3.1.6 Leadership (4 items),

Since pilot B (Walsham 2002) reported 4 dimensions for this construct, the same result is expected in this study.

#### 6.3.1.7 Motivation (4 items),

Since pilot B (Feller 2005) reported 4 dimensions for this construct, the same result is expected in this study.

#### 6.3.1.8 Communication (4 items)

Since pilot B (Judith 2007) reported 4 dimensions for this construct, the same result is expected in this study.

#### 6.3.1.9 Team size (4 items).

It would appear that measurement of this construct has previously not been reported in the literature. Since both pilot studies of this research found hardware experience to be a multidimensional construct, the same result is expected in the main study.

**Table 6- 4 Exploratory Factor Analysis –Summary**

<b>Construct</b>	<b>Number of factors extracted</b>	<b>Proposed Dimensionality</b>	<b>Variance explained</b>
Language experience	5 factors	Expected	63.9%
Programming experience	4 factors	Expected	50.1%
Application experience	4 factors	Expected	75.2%
Hardware experience	4 factors	Expected	58.9%
Leadership	4 factors	Expected	59.6%
Motivation	4 factors	Expected	55.3%
Communication	4 factors	Expected	54.4%

#### **6.4 Validity**

The scales used in the analysis required initial development, further development, or modification to the research context. Hence, it was deemed necessary to assess the construct validity of the scales...*the extent to which an operationalization measures the concept it is supposed to measure....* (Bagozzi, 1994:20). This assessment can only be inferred; it cannot be assessed directly (Peter, 1981). In order to assess the construct validity for this research the following analyses were undertaken:

1. A factor analysis of each of the construct measures to assess if the expected dimensionality of each construct is achieved. This was obtained with each of the constructs of the study, as described in Table 6- 4 - Exploratory Factor Analysis –Summary, thereby providing supportive evidence of construct validity (Peter, 1981).

2. A factor analysis of all the items representing constructs in the model in order to evaluate discriminant validity (Sweeney 1995)

## 6.5 Factor Analysis

Factor analysis essentially tells us the extent to which measures relate to the same thing (Nunnally 1967). Therefore, a factor analysis of all the items that load on multi item constructs will demonstrate the amount of overlapping items that may exist between constructs. The factor analysis undertaken utilized principal components analysis with oblique rotation constrained to eight factors.

The factors extracted represented the following constructs: Factor 1, Language skill explaining 14.4% of the variance; Factor 2, Programming skill explaining 9.2% of the variance; Factor 3, application skill explaining 4.6% of the variance; and Factor 4, Hardware skill explaining 2.2% variance; Factor 5, Leadership explaining 4.3% variance; Factor 6, Motivation explaining 4.6% variance; Communication explaining 4.5% variance.

The unidimensionality, convergent validity and discriminant validity of the scales were assessed through exploratory Factor analysis. The varimax rotated factor solution gave adequate support for the unidimensionality and validity of the scales. Results of the varimax rotated analysis indicated the existence of eight significant factors with eigenvalues greater than one. The eight factors together explained about 60% of the total variance. The KMO measure of sampling adequacy value was 0.84 indicating sufficient level of intercorrelations. The Barlett's test of sphericity was also found to be significant (Chi-square= 482.046,  $p < 0.001$ ). All the items loaded to their predicted factors indicating the existence of unidimensionality. The factor loadings for the items were also quite high (above 0.5) indicating the existence of convergent and discriminant validity. The factor structure therefore effectively established the validity of the scales. Based on a review of the scale-items that loaded the most on each factor, factor labels were developed. The eight factors were labeled as: (i) Language experience (5 items), (ii) Programming experience (4 items), (iii) Application experience (4 items), (iv) Hardware experience (4 items), (v) leadership

(4 items), (vi) motivation (4 items), (vii) communication (4 items) and (viii) team size (4 items).

**Table 6.5 Factor analysis and scale reliabilities**

<b>Factor Label</b>	<b>No. of Items</b>	<b>Factor loading (range)</b>	<b>Coefficient of reliability</b>
Language experience on quality	5	0.512-0.758	0.80
Programming experience on quality	4	0.545-0.916	0.72
Application experience on quality	4	0.714-0.990	0.80
Hardware experience on quality	4	0.532-0.745	0.78

## **6.6 Reliability Analysis**

A high internal consistency estimate provides support for the construct validity of a measure of a unidimensional construct (or for separate subscales of a measure of a multidimensional construct) ..... (Peter 1981:136).

Reliability was undertaken on each of the dimensions identified in the exploratory analysis using Cronbach's coefficient alphas (Cronbach 1951). With the exception of role conflict, measurement of the constructs is still in the early stages of research. Hence, Nunnally's recommendation of accepting modest alpha reliabilities of .60 and .50 was taken.

*In the early stages of research on predictor tests or hypothesized measures of a construct, one spends time and energy by working with instruments that have only modest reliability, for which purpose reliabilities of .60 or .50 will suffice (Nunnally 1967: 226).*

Table 6 – 5, Reliability Analysis of construct Dimensions, describes the coefficient alphas that were obtained from each of the factors of the constructs. All the alphas fell above the acceptable lower limit of .50 for research in the early stages. All the alphas were generally lower than the previous reliability analysis of construct measures.

The reliability of the constructs was established through Cronbach alpha measurements. The reliability coefficient (alpha) of each element are as follows: Language experience (80 percent), programming experience (72 percent), application experience (80 percent), hardware experience (78 percent), leadership (81 percent), motivation (74 percent), communication (76 percent), team size (75 percent). This effectively establishes the internal consistency reliability of the scales.

### **6.7 Analysis of variance**

ANOVA gives a statistical test of whether the means of several groups are all equal, and therefore generalizes Student's two-sample *t*-test to more than two groups. In practice, there are several types of ANOVA depending on the number of treatments and the way they are applied to the subjects in the experiment.

In this analysis, a One-way ANOVA is used. A one-way ANOVA is to test for differences among two or more independent groups. Typically, however, the one-way ANOVA is used to test for differences among at least three groups, since the two-group case can be covered by a T-test (Gossett, 1908). When there are only two means to compare, the T-test and the F-test are equivalent; the relation between ANOVA and *t* is given by  $F = t^2$ .

The data for analysis were 125 samples from Indian and 158 samples from United States. In the first instance, the Indian and the US samples were compared using the Analysis of Variance technique. It is a useful technique, which allows us to compare means for two or more groups (Zar 1999). ANOVA test were conducted to compare the variations on the “impact of team factors on quality across the two groups – Indian sample and US sample with the grouping variable as country name. The summary of the results is depicted in Table 6 - 6. Table 6 - 7 shows the means and standard deviations of the variables considered.

As can be seen from Table 6 - 6, the extent to which language experience of team members' impact software quality is found to be varying across the Indian and US sample. The F value 5.63 is found to be significant at  $p < 0.02$ . This provides support to the hypotheses that the influence of team members' language experience on software quality varies between India and U.S. From Table IV, it is seen that the mean of the importance of language experience for the U.S. sample is 3.91 while the mean for this variable in the case of the Indian sample is 3.63, along with results from the ANOVA, it is therefore concluded that the language experience of team members is a stronger determinant of software quality in U.S than in India.

In the case of the impact of hardware experience of team members on determining software quality, again the ANOVA results point to a significant difference between the Indian and U.S. sample. From Table 6 - 6, the F value for this variable is 13.848, which is significant at  $p < 0.0005$ . From Table 6 - 7, it is seen that the mean of this variable for the U.S. sample (3.69) is much higher than in the case of the Indian sample (3.19). This again shows that the hardware experience of a team member is a stronger determinant of software quality in U.S. software development teams than in Indian software development teams.

The importance of application experience on software quality however gives a different result. While the F value (7.656) is significant at  $p < 0.001$ , and hence a difference can be inferred between the two countries, the mean value for the Indian sample is seen to be higher than the U.S. sample from Table. 6 - 7. This goes to indicate that there is a stronger relationship between application experience of team members and quality of the software produced in India when compared to the U.S. This is definitely an interesting result. A similar result is seen about the impact of programming experience of team members on software quality. Here too the ANOVA result indicate a distinct difference between Indian and U.S. software development teams and a higher value for the mean in the case of the Indian sample. This shows that the impact of team member's program experience on software quality is higher in India than in U.S.

In the case of the impact of team member motivation on software quality, the ANOVA doesn't point to a statistically significant result, as the F value is significant



only at  $p < 0.1$ . A similar result is seen in the case of the impact of team size on software quality. Thus, it can be concluded that in terms of the impact of team member motivation and team size on software quality, there is very little difference between Indian and U.S. software teams. However, there is a significant difference for the impact of leadership on software quality as well as for the level of communication on software quality between the Indian and U.S. software development teams. In both the cases the F value is shown to be significant at  $p < 0.001$ . An examination of the mean values, show that in the case of leadership's impact on quality, the Indian sample has a higher mean thereby pointing to a greater importance in India than in the U.S. At the same time in the of the importance of communication on software quality, it is the U.S. sample which has a higher mean than the Indian sample which goes to show that in the U.S. it is definitely more important as a determinant of software quality than in India.

**Table 6 – 6 ANOVA of the relative effect of team factors on software quality in terms of India and United States**

		<b>Sum of Squares</b>	<b>df</b>	<b>Mean Square</b>	<b>F</b>	<b>Sig.</b>
Language experience on quality	Between Groups	5.448	1	5.448	5.363	.021
	Within Groups	279.347	275	1.016		
	Total	284.794	276			
Hardware experience on quality	Between Groups	17.877	1	17.877	13.848	.000
	Within Groups	362.752	281	1.291		

	Total	380.629	282			
--	-------	---------	-----	--	--	--

Application experience on quality	Between Groups	13.943	1	13.943	7.656	.006
	Within Groups	511.795	281	1.821		
	Total	525.739	282			
Programming experience on quality	Between Groups	3.599	1	3.599	4.455	.036
	Within Groups	227.009	281	.808		
	Total	230.608	282			
Motivation on quality	Between Groups	3.299	1	3.299	2.762	.098
	Within Groups	335.676	281	1.195		
	Total	338.975	282			
Leadership on quality	Between Groups	18.173	1	18.173	15.468	.000
	Within Groups	327.791	279	1.175		
	Total	345.964	280			
Communication on quality	Between Groups	5.711	1	5.711	9.226	.003
	Within Groups	173.949	281	.619		
	Total	179.661	282			
Team size on quality	Between Groups	.291	1	.291	.192	.661
	Within Groups	425.687	281	1.515		
	Total	425.979	282			

**Table 6 – 7 Summary statistics of these variables.**

Variable name	Mean		Standard deviation	
	India	US	India	US
Language experience on quality	3.634	3.917	1.0101	1.0048
Hardware experience on quality	3.189	3.696	1.2474	.9773
Application experience on quality	3.879	3.728	1.0117	1.1871
Programming experience on quality	3.987	3.920	.8886	1.1681
Leadership on quality	4.278	4.112	.7890	1.1014
Motivation on quality	4.291	4.240	.7341	.8926
Communication on quality	4.113	4.400	.8444	.7071
Team size on quality	3.911	3.976	1.3132	1.1177

## 6.8 Summary

This chapter first described the sample profile. Though a multi stage approach in the administration of the main survey as outlined in the previous chapter, a final ‘clean’ response rate of 36% was achieved, which represented respondents. Evidence presented suggests that the characteristics of the sample data are similar to the population data.

Exploratory factor analysis found the expected dimensionality of the construct measures. Construct validation found modest coefficient alphas, which were considered acceptable given Nunnally’s (1967) recommendations. Discriminant validity was considered modest but acceptable given the results of three analyses of the scale items.

Analysis of variance was undertaken to compare the variations on the “impact of team factors on quality across the two groups – Indian sample and US sample with the grouping variable as country name.

This is not surprising as each of these items represent positive behaviors closely aligned to a superior quality of the software. Given the small amount of overlapping of factors, it is considered that this analysis provides support for an acceptable level of discriminant validity between measures.

## CHAPTER 7

### DISCUSSION AND CONCLUSIONS

#### 7.1 Introduction

The objective of this chapter is first, to present a summary, together with a brief discussion, of the research results presented in chapter 6. Second, discuss the marketing applications of this research. Third, outline the contributions of this study. Fourth, address the limitations of this research. Finally, the possible future research directions will be considered.

#### 7.2 Primary research purpose

The primary research purpose of this study was to empirically examine and extend our knowledge of the determinants of software quality in software development organizations. Specifically, the focus was on those who work in software teams working in India and United States. Importantly, this research was to be undertaken from the perspective of software development team members on software quality.

The research purpose was achieved using both qualitative and quantitative approaches among software development teams. The survey instrument measured the constructs of software quality, language skills, programming skills, application skills, hardware skills, leadership, motivation and communication, as depicted in the empirical model. Testing of the propositions proposed by the empirical model utilized “Analysis of variance”.

#### 7.3 Discussion of Research Results

The empirical results that have been detailed in chapters 6 will now be discussed in the context of the research questions that were outlined in Chapter 1 and which broadened the research purpose of this study.

The overall objective of this study was to investigate the differences in impact of team factors on quality in software development teams in India and United States.

The impacts of team factors were found to be higher in the US than in India. The next section explores on the research questions and discussions on it.

### **7.3.1 What is the extent of relationship between the Domain skills of the team members on quality of software product in India and United States?**

Krishnan (1998) referred to domain skills of team members on quality as one concept in his discussion of role of team factors on software quality; where as the current research specified domain skills as one of the factor among technical skills of the software team members determining software quality.

The application experience of team members was found to differ extensively in these countries. The impacts of application experience were higher in India. This confirms the findings that the Indian software and service industry must place more emphasis on domain and project management skills (Timgoles 2008).

### **7.3.2 what is the extent of relationship between the Language skills of the team members on quality of software product stronger in India and United States?**

The language experience and programming experience showed a huge significance on the effect of team factors on quality. This supports the argument that the service projects in India need an extensive skill set of low technical skills like language (Judith, 2007). This further confirms the findings that the project teams with different types of projects in different countries possess different skills in experience (Tim 2008). The software development teams in India were on services and maintenance which needed low level programming and language skills for low level coding and testing .The results indicate that the language skills of the team members on quality is higher in the US teams. This is a contradictory to the hypothesis that extent of language skills in Indian teams is higher in India than in US teams.

### **7.3.3 What is the degree of relationship between the Program skills of the team members on quality of software product on teams in India and United States?**

The programming experience of the team members showed huge variations projects types done in India and United States. The results substantiate the differences in programmer capability between the performances of programmers (Sheppard, 1979). The hypothesis of this study, examines the programming skills of the Indian teams necessitate to be higher for the reason that the service industry in Indian projects require higher skills in programming. The results show a contradiction to this assumption and stated that the US teams showed the extent of variations higher in US teams.

### **7.3.4 What is the extent of relationship between the hardware skills of the team members on quality of software product on India and United States?**

The next research question was the extent of relationship between the hardware skills of the team members and the quality of the software product. However, the findings indicate that the impact of hardware experience varies substantially in India and US. The Indian software development teams was not based on manufacturing hardware, but rather more on service and intellectual works (Tschang 2001, Jorge 2008). The result indicates that the disparity of hardware skills impact on software quality has a substantial importance and was higher in US than in India. The next section discusses on the project management skills of the software team and the extent of relationship of the project management skills in India and United States.

The Indian managers were critical of the Indian system of promoting software programmers to managers based on seniority rather than on proven managerial ability, which weakened their project management skills (Sahay 2001; Walsham 2002; Tracy 2007; Thomson 2007). Indian firms, on the other hand, cited this practice as a way of providing a career path to their professionals and a major part of their attempts to hold down employee attrition (Arora 2001).



### **7.3.5 What is the extent of relationship between the Leadership skills of the team members on quality of software product in India and United States?**

The results of this study revealed that the impact of leadership on software quality is the most dominant team factor that varies in India and United States. This emphasizes the concentration of service providers on business domain and project management skills (Timgoles 2008). Walsham (2002) point out that Indian managers and developers tend to be members of different social systems arising from both work and non work related members of different social systems such as intellectual groups, local community and family. Often these rules and resources drawn upon to form these systems are conflicting, for example the work norm of efficiency would clash with the family norm of helping.

Where as the western culture often exhibit high intelligence, dominance, self-confidence, integrity, energy levels and task relevant knowledge (Smith 1972). The results stated that the hypothesis that the impact of leadership diverges in Indian and US teams are true.

### **7.3.6 What is the extent of relationship between the motivation skills of the team members on quality of software product stronger in India than in United States?**

The revenue per employee is probably a far better indicator of productivity in the software services industry (Arunachalam 1999; DCosta 1998). In this industry, equipment is not very expensive compared to the other service industries in India and firms do not differ very significantly in terms of the capital employed per employee. Thus, revenue per employee provides a reasonable motivational measure of productivity (Sahay 2001; Roland 2000). The results substantiate this hypothesis that the effect of motivation on the team members had a greater impact on US teams .The effect may be because of well-structured career prospects in the US job culture.

### **7.3.7 What is the extent between the communication skills of the team members on quality of software product differ in India and United States?**

The findings from the research indicate that there has been a consistent variation in the team skills between the US and Indian communication skills. The results of this study also revealed that the impact of communication on software quality is the most dominant team factor that varies in India and United States. Another related weakness is the lack of familiarity of many Indian firms and professionals with the work culture and work norms in the West, and especially in the United States. Other cultural differences included the unwillingness of Indians to speak out in meetings, necessitating “offline” conversations to convey the needed information. The results indicate lack of skills of Indian teams to work in multicultural environments and the irritants of communication skills of Indians (Arora 2001), which substantiate the hypothesis of the study.

### **7.3.8 What is the extent of relationship between the team size on quality of software product stronger in India than in United States?**

The impact of team size of the software team was found to have insignificant contributions towards software quality in US and India. This indicated that whether the team size is small or the team size are big the quality of the software developed is not affected in the countries India and United States. The result is contradictory to the assumptions that the impact of team size on the quality of the software product is stronger in India than in United States.

## **7.4 Managerial Implications**

Important implications flow from the results of the research. The study’s greatest insight for management is that it highlights team variables that influence the software quality. This provides management with greater insight for policy implementation to ensure excellent quality for the software produced.

To further understand the nature of software quality, the summary statistics of each of the dimensions of software quality were assessed. Reference to the descriptive statistics of the dimensions of software quality reflect that on average, respondents participated in the positive behavior of leadership far more frequently than any other behavior dimensions. Since so many of the team factors measured can have disastrous results in the software quality arena, it is comforting to note that software quality engage in technical skills and project management skills of the team members are significantly more often than other factors. This supports Pressmen (2005) observation that team factors to be considered by the software quality personnel to endeavor the task done, prior to other factors.

### **7.5 Summary**

In summary, the findings of this study focus on the variability of impact of team skills software quality, where the team members to have a optimum skill on language and programming, high level of domain skills and hardware skills and a positive managements recognition of them within the organization, to ensure the delivery of superior quality. Also, the study confirms the influence the project management skills in an organization to initiate positive software quality.

The study highlights the need to make a concerted effort to implement strategic policy that will enable management to first, recruit the most suitable members in a team and second, exert influence over the software quality of their existing team. Key areas for consideration by management include:

1. The introduction of personality instruments for screening potential software employees for a software development team.
2. Assessment of the technical skills perspective of potential employees.
3. Conveyance to employees of management's appreciation and recognition of the work of software developer.

4. Improvement of project management skills experienced by team member that is driven internally by the service organization.
5. The implementation of software quality culture and philosophy that represents the team member role in the software quality encounter, so as to ensure a harmonious team behavior.
6. The importance of ongoing training, seminars and support structures to address key issues that impact software quality

This research highlights the impact of language skills, programming skills, hardware skills and application skills on the quality of the final software delivered. Of particular interest were the influence of different team project management perspectives and the managerial implications ensuing from this. The company's service philosophy and leadership, including human resources policy influence on the project management perspective are considered essential inputs to the management of the perceived perspective. Additionally, training is considered essential for the maintenance of software team members with management modeling the software quality they require.

A task of management will be to ensure the team skills of the software developer suitable in countries United States and India. A US team member is generally tasked with software design of the software development phases and so their skills in the domain of the software project is thought to be higher. The project management skills also need to be higher in the case of software development teams in United States. In India the software development is mainly the service and maintenance oriented and the individuals are to be taught with skills on programming and language and the project management skills. Achieving this may require a combination of management approaches including recruitment and selection, training, empowerment and provision of resources, team building and support and reorganization of work groups, coupled with clear policies and procedures supported by integral marketing campaign.

*Because managing an organization's human resources equates with managing its customer services (Smith, 1989:16).*

## 7.6 Contributions of the study

The major contribution of this research is the development of determinants of software quality of the software team. Importantly the research addresses the technical skills of the software team member together with the project management skills of the team member. The research offers initial findings concerning constructs that have been found theoretically and empirically to influence the software quality. This is an important area for the research given we know that the software quality impacts the overall productivity of the software in software development organization.

The focus of this study has been on software quality, where research into antecedents is in its infancy. Therefore, the empirical research offers new research findings in an area where the proposed relationships between constructs have had minimal discussion on research in the software quality literature. These areas include:

Software quality, which has only been measured once previously (Krishnan, 1998) among software developers in the United States. Additionally, one of the antecedents of this study's model i.e. domain experience was measured in the previous study.

- Language skills and programming skills have been rarely studied.
- Hardware skills which have not been studied in the service organizations.
- The investigation of project management skills has been rarely specifically measured
- Variable team size which have not been studied

Importantly, the research draws attention to the role of team skills in software quality and offer findings concerning the management of software quality in order to ensure superior quality of software. This is critical to the objective of software quality

## 7.7 Conclusion

As discussed earlier, the focus of this analysis is to study the variations in the effect of team factors on software quality based on organizations in India and the United States. Perceptions of the software development employees on the model were collected from 34 companies in India and 16 companies in the United States. To check for differences in perception among the US teams and Indian teams, the responses were analyzed using an ANOVA with respondent country as the grouping variable. The results were analyzed by taking the characteristics of the team and its members in each dimension and comparing them with the responses they gave in the form of questionnaire.

Experiences of the developers showed an improvement in the quality of the software product in United States when compared to the developers in India. But in the case of language experience or programming experience, hardware experience, and the effect of variations were highly significant. The skill requirements for the success of Indian projects were in programming skills, which varied the requirement of skills in US projects which were experience in the application domain (Arora 2001; Judith 2007). The experience in the application domain were an essential part in the software development in the United states, since they carried out more of high end software development .The Indian software teams developed mainly service oriented projects ,of which the success demanded higher knowledge in the language experience and programming capability

Effects of the project management skills communication and leadership skill on Indian software development teams and US were highly significant. The extensive variability on the effect of communication skills on quality were due to the cultural issues of India team members i.e., reluctant to speak on meetings. The impact of leadership skill on quality showed very significant variation in the teams in India and US. The software developments in the United States were more disciplined and followed a very good structural organizational hierarchy from the manager to team members. Thus the leadership and the communication skills improved the team dynamics in the teams in United States .So the Indian teams had greater impact on leadership and communication skills.

A very low variation among effect of application experience and motivation on quality were noticed. Effect of the motivational factor was seen higher in Indian teams as the software development industries were still young in India and large-scale projects where project managers are trained are still relatively rare. Team members were motivated paying them higher, and with higher family benefits given by the organization in the software industry compared to other industries. The motivation reduced the problem of large number of experienced professional emigrating to the U.S

The effect of the team size of the team on quality showed no variations in the United States and Indian teams.

### **7.8 Limitations and directions for future research.**

Even though the software quality determining variables in the study have been extensively analyzed in the past the degree of variations in the quality determining team related variables in countries like India and United states have not received much attention. The present study is an attempt to this direction.

The validity of the study is only a comparison of software development in United States and India. Further research is to extend the study to other Asian countries like china and the Far East countries like Korea and Singapore were software development services to Unites States is extensive.

Second, other team related variables such as education level of the team members and the performance evaluation ratings of the team members are not included. At the research sites, performance evaluations were kept confidential. The third limitation of the study concerns the sources of responses.

In addition, it should be noted that the main focus of the study is on the effect of team factors on software product. Other factors, such as the use of tools and technology and adopting various disciplined software practices, may also impact quality of the software product and are issues of future research.

## APPENDICES

**APPENDIX 1 LIST OF ORGANIZATIONS RESPONDED TO THE SURVEY**

Names of firms responded to the survey

## INDIA

1. Ramco systems
2. Tata consultancy services
3. Wipro Technologies Ltd
4. IBM
5. Hex aware technologies
6. HCL Corporation
7. Ifex Solutions
8. Inautix
9. Sacksoft Ltd.
10. Polaris
11. EDS
12. Sathya m computers
13. seimens
14. Tech Lead
15. Sri Jain Technologies
16. Bell
17. Sophla
18. SSI
19. Infomix
20. Extech
21. Cosmosot Technologies
22. Lieourdreams Ind pt Ltd
23. RBI
24. Excellent Automations
25. Sophia technologies
26. Omniscient global solutions
27. Alcatel elluent
28. RSA
29. Cap gemini
30. Sm steels
31. KGM Sotwares
32. DMiss
33. Micro objects
34. syntel

## UNITED STATES

1. IBM
2. Tmobile
3. Oracle Corporation Inautix
4. Best cross mark
5. Texas instruments
6. AT & T
7. Idearc media corporation
8. Quorum Business Solutions
9. GS
10. Signiti Ink
11. Amerigroup
12. Logica CMG
13. Texas department o agriculture
14. Wiquest
15. Epics systems corporation



**APPENDIX 2      DESCRIPTION OF RESPONDENTS PARTICIPATED IN  
THE SURVEY**

<b>Data collected from organizations under study</b>	<b>United States</b>	<b>India</b>
Total number of organizations	16 (123 samples)	34(158 samples)
Average organization size	32 employees	43 employees
Organization maturity level	CMM certified = 16	CMM certified = 24 ISO Certified= 5
Average project duration	9 – 15 months	10-15 months
Average project team size	4	4
Number of teams under study	32	34
Respondents data	Total respondent: 128 Male:77 Female:51	Total respondent:158 Male:92 Female:56
Duration of service in current organization	Less than 1 year - 26 Between 1-3 years-59 Between 3-5 years-23 5 years and above-10	Less than 1 year-32 Between 1-3 years-68 Between 3-5 years-44 5 years and above-14
Project domains of teams under study	Insurance Banking Telecommunication Ecommerce Finance	Telecommunication Finance Insurance Banking ERP SCM Ecommerce

### APPENDIX 3 QUESTIONNAIRE USED FOR THE SURVEY

*Benchmarking Study to identify the role of team factors on cost and quality in the Software Engineering Industry*

*The purpose of the study is to collect the range of factors which impact the cost and quality of the software product*

Questionnaire

Please complete all applicable sections of the questionnaire. Please mail or fax the completed questionnaire to Viji Vinod – vijivino@gmail.com. **All data is confidential.**

Name of person completing this questionnaire:

\_\_\_\_\_

Position title: \_\_\_\_\_

Company name:

Company address: \_\_\_\_\_

---

*Does your organization provide or make software product development ?*

yes \_\_\_\_, no, organization-sponsored training is not available to employees \_\_\_\_,

#### **A. Introduction**

1. Has a team software development been conducted in your organization?

yes \_\_\_\_, no \_\_\_\_, don't know \_\_\_\_

2. If yes, are you engaged in working with a team \_\_\_\_ ?

yes \_\_\_\_, no \_\_\_\_

3. In your opinion, what kind of improvement' does team related factors impact on the software quality of the product in your organization?

	None	some	Quiet a bit	Extreme amount	all
significant improvement	1	2	3	4	5
moderate improvement	1	2	3	4	5
some improvement	1	2	3	4	5
no improvement	1	2	3	4	5

**B. Personnel capability of the team members Questions:**

4. Is personality development training provided for all software engineering related job classification levels

Management	None	some	Quiet a bit	Extreme amount	all
High level	1	2	3	4	5
Middle	1	2	3	4	5
Low level	1	2	3	4	5
Support staff:	1	2	3	4	5

5. How do you rate the management's promotion towards employees for attending training?

Strongly encourage	encourage	Neutral	discourage	Strongly discourage
--------------------	-----------	---------	------------	---------------------

6. Do you feel that management supports team software development?

	<b>Never</b>	<b>Occasionally</b>	<b>Fairly many times</b>	<b>Very often</b>	<b>Always</b>
Executive management	1	2	3	4	5
Middle management	1	2	3	4	5
Supervisory (project) management	1	2	3	4	5

7. Are project delivery times achieved by the team?

None	Some	Quiet a bit	An extreme amount	All
------	------	-------------	-------------------	-----

8. How do you rate the effect of personnel capability to software development?

Best	Above average	Average	Below average	Worst
------	---------------	---------	---------------	-------

### **C. Leadership:**

9. Have you lead a team?

None	Some	Quiet a bit	An extreme amount	All
------	------	-------------	-------------------	-----

10. How do you rate the leadership skill of your team members?

Very good	good	fair	poor	Very poor
-----------	------	------	------	-----------

11. Do you feel that leadership has effect on the quality of the product?

Far too much	Too much	About right	Too little	Far too little
--------------	----------	-------------	------------	----------------

**D. Size of the team:**

12. What is the extent to which you believe that the size of the team has an effect on quality of the product ?

To large extent	Moderate extent	Some extent	Very minimal extent	Not at all
-----------------	-----------------	-------------	---------------------	------------

13. For software engineering and related disciplines, as size is decreasing, product quality is

	Never	Occasionally	Fairly many times	Very often	Always
Increasing	1	2	3	4	5
Decreasing	1	2	3	4	5
Same as last year	1	2	3	4	5
No difference	1	2	3	4	5

14. How do you rate the effect of team size to the cost of software development?

	Never	Occasionally	Fairly many times	Very often	Always
High	1	2	3	4	5
Medium	1	2	3	4	5
No difference	1	2	3	4	5

**E. Cost**

15. Do you have a separate group for providing software engineering and related budget?

Never	Occasionally	Fairly many times	Very often	Always
-------	--------------	-------------------	------------	--------

16. Is this group knowledgeable about software engineering and related functions?

Very knowledgeable	Knowledgeable	Neutral	Not knowledgeable	Worst
--------------------	---------------	---------	-------------------	-------

17. In general, is the quality procedures budget?

	Never	Occasionally	Fairly many times	Very often	Always
Increasing	1	2	3	4	5
Decreasing	1	2	3	4	5
Same as last year	1	2	3	4	5
No separate budget	1	2	3	4	5

18. How do you rate the effect of size to the cost of software development?

No effect	Some	Quite a bit	Extreme amount	all
-----------	------	-------------	----------------	-----

## F. Quality

19. Are Quality procedures documented?

	Always	To large extent	About 50% of time	Very rarely	Never
Analysis	1	2	3	4	5
Design	1	2	3	4	5
Testing	1	2	3	4	5
Evaluation	1	2	3	4	5
Maintenance	1	2	3	4	5

20. Is the quality process subject to continuous improvement?

Never	Occasionally	Fairly many times	Very often	Always
-------	--------------	-------------------	------------	--------

21. Is quality (check all that apply)

	Never	Occasionally	Fairly many times	Very often	Always
Manually evaluated	1	2	3	4	5
Computer based	1	2	3	4	5

22. How effective is your quality process?

Very effective	Effective	Neither	Ineffective	Very ineffective
----------------	-----------	---------	-------------	------------------

### G. Communication and coordination

23. Do the team members help each other?

Never	Occasionally	Fairly many times	Very often	Always
-------	--------------	-------------------	------------	--------

24. If yes, to what level?

Far too much	Too much	About right	Too little	Far too little
--------------	----------	-------------	------------	----------------

25. Is there hierarchy in co-coordinating and communication between team members?

<b>Management hierarchy level</b>	<b>None</b>	<b>Some</b>	<b>Quiet a bit</b>	<b>An extreme amount</b>	<b>All</b>
Top to bottom	1	2	3	4	5
Bottom to top	1	2	3	4	5
Middle to top /bottom	1	2	3	4	5
No hierarchy	1	2	3	4	5

26. Do you feel that co-ordination of work has an impact on the quality of software?

Much higher	Higher	Almost same	Lower	Much lower
-------------	--------	-------------	-------	------------

27. Are there corporate communication training available in your organization?

Never	Occasionally	Fairly many times	Very often	Always
-------	--------------	-------------------	------------	--------

28. Are communication done through circulars or you receive gossips before the circulars

	<b>Never</b>	<b>Occasionally</b>	<b>Fairly many times</b>	<b>Very often</b>	<b>Always</b>
Circulars	1	2	3	4	5
Gossips	1	2	3	4	5

29 .If circulars ,(check all apply)

	<b>Never</b>	<b>Occasionally</b>	<b>Fairly many times</b>	<b>Very often</b>	<b>Always</b>
Computer based	1	2	3	4	5
Manual	1	2	3	4	5



30. Do you have a proper communication channel?

None	Some	Quite a bit	Extreme amount	All
------	------	-------------	----------------	-----

31. Do you feel that communication has an effect on quality of software?

Very effective	Effective	Neutral	Ineffective	Very ineffective
----------------	-----------	---------	-------------	------------------

### H. Motivation

32. Are the team members self motivated?

Never	Occasionally	Fairly many times	Very often	Always
-------	--------------	-------------------	------------	--------

33. Are the team members motivated by the management?

	<b>Never</b>	<b>Occasionally</b>	<b>Fairly many times</b>	<b>Very often</b>	<b>Always</b>
Executive management	1	2	3	4	5
Middle management	1	2	3	4	5
Supervisory (project) management	1	2	3	4	5

34. Do you feel that the motivation among members has an impact on the quality of the product developed ?

Never	Occasionally	Fairly many times	Very often	Always
-------	--------------	-------------------	------------	--------

**I. Experience**

35. How do you rate each of the following factors in terms of its importance in determining the quality of the product?

Low

high

Total experience of members	1	2	3	4
Language experience	1	2	3	4
Hardware experience	1	2	3	4
Application experience	1	2	3	4
Programming capability	1	2	3	4

36. How do you rate each of the following factors in terms of its importance in determining the cost of the product?

low

high

Total experience of members	1	2	3	4
Language experience	1	2	3	4
Hardware experience	1	2	3	4
Application experience	1	2	3	4
Programming capability	1	2	3	4

37. How do you rate each of the following factors in terms of its importance in determining the quality of the product?

low

high

Personnel capability	1	2	3	4
Leadership	1	2	3	4
Motivation	1	2	3	4
Communication	1	2	3	4
Coordination	1	2	3	4
Size	1	2	3	4

39. How do you rate each of the following factors in terms of its importance in determining the cost of the product?

low

high

Personnel capability	1	2	3	4
Leadership	1	2	3	4
Motivation	1	2	3	4
Communication	1	2	3	4
Coordination	1	2	3	4
Size	1	2	3	4

**APPENDIX 4 COVERING LETTER****To the Employees of Software Organizations**

This is a study based on the “Role of Team Skills on Cost and Quality” of the software product. The respondents to the questionnaire are software employers working in IT organizations in India and United States.

Please fill in the questionnaire with your responses on your opinions on the role of team skills on cost and quality of the software developed.

Your responses are to be returned to

Ms. Viji Vinod

[vijivino@gmail.com](mailto:vijivino@gmail.com)

---

Thank You for your time and effort in completing this survey. Your help in this research is appreciated.

---

**APPENDIX 5 REMINDER FOR THE RESPONDENTS**

**APPENDIX 5 - A**

## Reminder 1

Dear software developers,

You may recall receiving a questionnaire on the “Impact of team skills on cost and quality of the software product”. If you have had not a chance to complete it as yet, it would be greatly appreciated if you could as soon as possible as your opinions and comments are essential to the success of the study. If you have already completed and returned the questionnaire, please note this as a note of thanks for your participation.

If you have misplaced your questionnaire and require another copy, please email me in the address [vijivino@gmail.com](mailto:vijivino@gmail.com).

Yours sincerely,

Viji Vinod

**APPENDIX 5 – B**

## Reminder 2

Dear software developers,

Please accept this as a note of thanks for participating in the survey on the “Impact of team skills on software cost and quality “.If you have had not have a chance to complete the survey,” THERE IS STILL TIME “.The closing date for replies has been extended to 11<sup>th</sup> December. Surveys should be returned to [vijivino@gmail.com](mailto:vijivino@gmail.com)

If you have misplaced your questionnaire and require another copy, please email me in the address [vijivino@gmail.com](mailto:vijivino@gmail.com).

Yours sincerely,

Viji Vinod

**REFERENCES**

1. Adams J.S. (1965), 'Inequity in social exchange' - Adv. Exp. Soc. Psychol , Vol 62, pp 335-343.
2. Adelson. and Eliot. (1985), 'Artificial intelligence and software engineering' IEEE Transactions on Software Engineering, Vol 11, No 11, pp. 1351-1360.
3. Alain Abran. and James Moore. (2004), 'Guide to the Software Engineering Body of Knowledge - 2004 Version', IEEE Computer Society, p. 1-1, ISBN 0-7695-2330-7. <http://www.swebok.org>.
4. Allison C.W. and Hayes. (1988), 'The learning styles questionnaire: an alternative to Kolb's inventory', Journal of Management Studies', Vol 25, No 3.
5. Ann K. Brooks. (2006), 'Power and the production of knowledge: Collective team learning in work organizations ', Human Resource Development Quarterly, Vol 5 No 3, pp. 213 - 235.
6. Arora. and Arunachalam V.S. (1999), 'The Indian Software Services Industry 'working paper, Heinz School, No. 99-19. Available at SSRN: <http://ssrn.com/abstract = 198968> DOI:10.2139/ssrn.10.2139/ssrn.198968. 07 December 2000.
7. Arora. and Ashish. (2006), 'The Indian Software Industry and its Prospects', working paper, Carnegie Mellon University, Available at SSRN: <http://ssrn.com/abstract = 964457/> (accessed 21 February 2007).
8. Arora, A. (2001), 'The Indian software services industry', Elsevier Science, Vol 30, No 8, pp.1267-1287.
9. Ashish A. and Arunachalam. (2001), 'The Indian Software Services Industry', Research Policy, Vol 30, No 8, pp.1267 - 1287.
10. Aswathappa. (2007), 'International Human Resource Management ', Tata McGraw hill.
11. Baddoo N. Hall T. (2003), 'De-motivators for software process improvement: an analysis of practitioner's views', The Journal of Systems and Software, Vol 66 pp. 23-33.
12. Bach J. (1995), 'Enough about Process: What We Need Are Heroes', Computer, Vol 12, No 2, pp. 96-98.
13. Bach J. (1999), 'What Software Reality Is Really About? ', Computer, Vol 32, pp. 148 - 149.

14. Bagozzi Richard P. (1994), 'Measurement in marketing research: Basic principles of questionnaire design' 'Principles of Marketing Research, 1 – 49, Cambridge and Blackwell.
15. Banker R. D. (1991), 'A model to evaluate variables involving the productivity of software maintenance projects', *Management Science*, Vol 37 , No 1, pp.1 -18.
16. Banker R D and Davis G B. (1998), 'Software development practices, software complexity, and software maintenance: a field study', *Management Science*, Vol 44 No 4, pp. 433 – 450.
17. Barstow D R. (1985), 'Domain-Specific Automatic Programming', *IEEE Transactions on Software Engineering*, Vol 11, No 11, pp. 1321-1336, Nov. 1985, doi: 10.1109/TSE.1985.231881.
18. Boehm. (2007), 'Early experiences in software economics', *software engineering*, Wiley, pp.219-226.
19. Broy M. and Denert E. (2002), 'Software Pioneers', Springer Verlag, Springer, p .632 - 640.
20. Barry W. and Boehm J R. (2007), 'Quantitative evaluation of software quality', *software engineering*, IEEE computer society, Vol 15, No 4, pp.21 , *software quality journal* ,ISSN 0963-9314 print,1573-1367online.
21. Bassellier G and Benbasat I. (2004), 'Business competence of information technology professionals: conceptual development and influence on IT-business partnerships', *MIS Quarterly*, Vol 28 No 4, pp. 673 – 694.
22. Basili V R and Briand L C. (1996), 'A Validation of Object-Oriented Design Metrics as Quality Indicators', *IEEE Transactions on Software Engineering*, Vol 22, No 10, pp.751-760.
23. Bateman A. (1990), 'Team Building: Developing a Productive Team', University of Nebraska, in Cooperation with US Department of Agriculture.
24. Beaver J M. and Schiavone G A. (2005), 'Predicting Software Suitability uses a bayesian belief network', In *Proceedings of the 4<sup>th</sup> International Conference on Machine Learning and Applications*, pp.82-84, IEEE Computer Society Press.
25. Belsley D A. and Kuh E. (1980), 'Regression Diagnostics: Identifying Influential Data and Sources of Co linearity', John Wiley & Sons, New York.



26. Berdie D R. (1989), 'Reassessing the value of high response rates to mail surveys', *Marketing Research*, Vol 1, pp. 52 – 64.
27. Bitner M J. and Booms B H. (1994), 'Critical Service Encounters: The Employee's Viewpoint', *Journal of Marketing*, Vol 58, No 4, pp.95 –106.
28. Bitner M J and Booms B H. (1990), 'The service encounter: diagnosing favorable and unfavorable incidents', *Journal of Marketing*, Vol 54, pp. 71 – 84.
29. Blackburn J D. (1996), 'Improving speed and productivity of software development', working paper, INSEAD, Fontainebleau, France.
30. Boehm B W. (1987), 'Improving Software Productivity', *Computer*, Vol 20, No 9, pp. 43 - 57, Sept. 1987, doi :10.1109/MC.1987.1663694.
31. Boehm E W. and Brown J R. (1978), 'Characteristics of software quality ', North Holland.
32. Boehm and Barry W. (1976), ' Quantitative evaluation of software quality', *International Conference on Software Engineering, Proceedings of the 2nd international conference on Software engineering*, pp. 592 - 605, Los Alamitos, CA, 1976, IEEE Computer Society Press.
33. Boehm B W. (2007), 'Early experiences in software economics', *software engineering*, Wiley - IEEE, pp.219 - 226.
34. Boehm B W. (1981), 'Software Engineering Economics', Prentice-Hall, Inc, Englewood Cliffs, NJ.
35. Brooks. and Boehm B. (1993), ' Software Engineering Economics' , Prentice-Hall, Englewood Cliffs, NJ, 1981.
36. Brooks F. (1993), 'The Mythical Man Month', Addison-Wesley, Reading, MA.
37. Broy. and Denert E. (2002), 'Software Pioneers' , Springer Verlag, Springer, pp .632 - 640.
38. Bruce J. and Avolio. (2007), 'Leadership: Current Theories, Research, and Future Directions', *Annual reviews*, Vol 60, pp. 421 - 449.

39. Calder. and Bobby J. (1977), 'Focus groups and the nature of qualitative marketing research ', *Journal of Marketing research* ,14 August , 353 – 364.
40. Carmel E. and Bird B J. (1997), 'Small is beautiful: a study of packaged software development teams ', *Journal of High Technology Management Research*, Vol 12, No 2, pp. 110 -123.
41. Carmel E. and Sawyer S. (1998), 'Packaged software development teams: what makes them different?', *Information Technology & People*, Vol 11, No 1, pp. 7 - 19.
42. Carmel E. and Agarwal. (2001), ' Tactical Approaches for Alleviating Distance in Global Software Development', *IEEE Software*, Vol 1, No 2, pp. 22 - 29.
43. Carnegie Mellon, Software Engineering Institute, Welcome to the CMMI ® Web Site, Carnegie Mellon, Software Engineering Institute, <http://www.sei.cmu.edu/cmml/cmmi.html>, 2004.
44. CASE definition In: Telecom Glossary 2000, Retrieved 26 Oct 2008.
45. Case A. (1985), 'Computer aided software engineering', *database*, Vol 17, No 1, pp. 35 - 43.
46. Castels M. and Hall P. (1994), 'Technologies of the World: The Making of 21st Century Industrial Complexes', Routledge, New York, and NY.
47. Catherine and Durnell. (2005), 'Relationships among geographical dispersion, team processes, and effectiveness in software development work team', *Journal of business research*, Vol 58, No 6, pp.758 - 765.
48. Charles Mitchell. (2000), 'International business culture', World trade press, California, P.37.
49. Chrysler E. (1978), 'Some basic determinants of computer programming productivity', *Communication of ACM*, Vol 21, No 6, pp. 478 - 483.
50. Churchill A. and Gilbert. (1979), ' A paradigm for developing better measures for marketing constructs, *Journal of Marketing research*, 16 February ,64 – 73.
51. Clampitt G. and Girard M. (1993), 'Communication satisfaction: A useful construct?' , *New Jersey Journal of Communication*, Vol 1, No 2, pp. 84 -102.

52. Conde Dan. (2002), 'Software Product Management : Managing Software development from Idea to Product to Marketing to Sales' , September 1, pp.24 – 29.
53. Cohen S. (2001), 'Job Satisfaction Survey: Desperate for Direction', Computer World magazine, Vol 29, No 2, pp.6 - 6.
54. Cougar J R. (1980), 'Zawacki Managing and motivating computer personal', Wiley.
55. Cronbach L J. (1951), 'Coefficient alpha and internal structure of tests ', *Psychometrika*, Vol 16, pp. 297 – 334.
56. Crosby P B. (1979), 'Quality is free: the art of making quality certain', McGraw-Hill New York.
57. Crowston K. and Howison J. (2006), 'Heirarchy and centralization in free and open source software team communications', *Knowledge, Technology and Policy*, Vol 18, No 4 ,pp.65 – 85.
58. Curtis Bill. and William E. (2001), 'An overview of people CMM', [http://www.adjugo.com/pdfs/120\\_page\\_doc.pdf](http://www.adjugo.com/pdfs/120_page_doc.pdf).
59. Curtis B. (1981), 'Substantiating programmer validity ', *proc IEEE*, Vol 69, No 7, pp.846 - 846.
60. Curtis B. and Krasner H. (1988), 'A field study of the software design process for large systems', *Communications of the ACM*, Vol 31, No 11, pp.1268 - 287.
61. Curtis B. and Miller S. (2001), 'People capability maturity model', Software engineering institute, Carnegie Mellon University.
62. Curtis B. and Hefley W E. (2007), 'The People Capability Maturity Model: Guidelines for Improving the Workforce', Dorling Kindersley (India) Pvt. Ltd, Delhi, India.
63. DCosta. (2002), 'Export growth and path independence: The locking of innovations in the software industry', *science technology and society*, Vol 7, No 1, pp.51 - 89.
64. DCosta. (1998), 'Technology Leapfrogging: Software Industry in India', Presented at the 2nd International Conference on Technology Policy and Innovation, Calouste Gulbenkian Foundation.

65. Demarco T. and Lister T. (1987), 'Peopleware Dorset House and Other Puzzles of the Information Age', Dorset House Publishing, New York, NY.
66. Demarco, T. (1982), 'Controlling software projects', Yourdon press, New York, NY.
67. Dilman. and Don. A (1978), 'Mail and Telephone surveys: The total design method ', USA, Wiley and Sons.
68. Deming, W E. (1988), 'Out of the crisis: quality, productivity and competitive position', Cambridge University Press.
69. Demarco, T. and Lister T. (1999), 'Peopleware: Productive Projects and Teams, Dorset house publishing Co., New York, USA.
70. Diaz M. and Sligo J. (1997), 'How software process improvement helped Motorola', IEEE Software, Vol 14, No 5, pp.75 – 81.
71. Dromey, R. G. (1996), 'Concerning the Chimera software quality', IEEE Software, Vol 1, No 1, pp. 33 - 43.
72. Dromey R G. (1995), 'A model for software product quality', IEEE Transactions on Software Engineering, No 2, pp. 146 -163.
73. Dickey T. (1981), 'Programmer variability', Proc IEEE, Vol 69, No 7, pp. 844 - 845.
74. Downs W. and Clampitt G. (1988), ' A Communication and organizational outcomes', In G. Goldhaber & G. Barnett, Handbook of organizational communication, pp. 171-211, Norwood, NJ, USA.
75. Ducker Peter. (1995), 'Management challenges for the 21<sup>st</sup> century', Harper Business, New York, P.61.
76. Durani. (2004), 'A survey to examine the effect of team communication on job satisfaction', ACM SIGSOFT, Software engineering notes, Vol 29, No 2, pp.1 - 10.
77. Dutoit A. and Breugge B. (1998), 'Communication Metrics for Software Development', IEEE Transactions on Software Engineering, Vol 24, No 8, pp. 615 - 628.

78. Dyke T. and Kappelman L. V. (1997), 'Measuring Information Systems service quality : concerns on the use of the SERVQUAL questionnaire', *MIS Quarterly*, Vol 21, No 2, pp. 195 – 208.
79. Ebert C and Alcatel A. (2001), 'Surviving global software development', *IEEE Software*, Vol 14, No 2, pp. 253 - 273.
80. Edgemon J. (1995), 'Right Stuff - How to recognize it when selecting a project manager', *Application Development Trends*, Vol 2, No 5, pp.37- 42.
81. Edmondson A C. and Winslow A. (2003), 'Learning how and learning what: Effects of tacit and codified knowledge on performance improvement following technology adoption', *Decision Sciences*, Vol 34, No 2, pp. 197 - 223.
82. Ethiraj S. and Kale P. (2005), 'Where do capabilities come from and how do they matter? A study in the software services industry', *Strategic Management Journal*, Vol 26, No 1, pp. 25 - 45.
83. England G W. (1975), 'The manager and his values: an international Perspective', Cambridge, Ma, Ballinger
84. Evans Isabel. (2004), 'Achieving software quality through team work', Published by Artech House, Norwood, MA, U.S.A.
85. Fahmi S. and Choi. (2008), 'A Survey on Team Software Process Supporting Tools', 3<sup>rd</sup> International conference on Convergence and Hybrid Information Technology, ICCIT '08, Vol1, pp. 987 - 990.
86. Faraj S. and Sproull L. (2000), 'Coordinating expertise in software development teams', *Management Science*, Vol 46, No 12, pp 1554 -1568.
87. Elashmawi Farid. (2001), 'Competing globally: mastering multicultural management and negotiations', *Elsevier science and technology*, p.144.
88. Feller B. and Fitzgerald S. (2005), 'Why hackers do what they do: Understanding motivation and effort in free/open source software perspectives on Free and Open Source Software', MIT Press.
89. Feigenbaum A V. (1983), 'Total quality control', McGraw-Hill.
90. Ferratt T W. and Agarwal R. (2005), 'IT human resource management configurations and IT turnover: theoretical synthesis and empirical analyses', *Information Systems Research*, Vol 16, No 3, pp. 237 – 255.

91. Francis Cherunilam. (2009), 'International Economics ', TataMcGraw-Hill.
92. Futrell R. and Shafer L. (2001), 'Software quality assurance: from theory to implementation', Prentice Hall, PTR Upper Saddle River, NJ, USA
93. Gibbons R. and Waldman M. (2004), 'Task-specific human capital', AEA Papers and Proceedings, Vol 94, No 2, pp. 203-207.
94. Gibbons R. and Waldman. (2006), 'Enriching a theory of wage and promotion dynamics inside firms', Journal of Labor Economics, Vol 24, No 1, pp. 59 -107.
95. Galin D. (2004), 'Software quality assurance: from theory to implementation', Addison- Wesley.
96. Galin D. (2004), 'Core and Periphery in Free/ Libre and Open Source Software Team Communications', Addison-Wesley.
97. Gambardella. and Alfonso. (2004), 'The Globalization of the Software Industry: Perspectives and Opportunities for Developed and Developing Countries', NBER Working Paper No. W10538, available at SSRN: <http://ssrn.com/abstract=556525> (accessed 17 June 2007).
98. Gerbing, Anderson. (1988), 'An updated paradigm for scale development incorporating unidimensionality and its assessment, Journal of Marketing Research, 25 May 186 – 192.
99. Gopal A. and Mukhopadhyay T. (2002), 'The role of software processes and communication in offshore software development', Communications of the ACM, Vol 45, No 4, pp.193 - 200.
100. Garvin D A. (1988), 'Managing Quality: The Strategic and Competitive Edge', The Free Press, New York, 1988.
101. Garvin D. (1984), ' what does product quality really mean? ', Sloan Management Review, Vol 26, No 1.
102. Grady R B. (1992), 'Practical software metrics for project management and process improvement', Prentice Hall.
103. Green G. (1989), ' Perceived importance of systems analysts' job skills, roles, and non - salary incentive ', MIS Quarterly, Vol 13, No 2, pp. 115 –133.

104. Haley T J. (1996), 'Software process improvement at Raytheon', IEEE Software, Vol 13 No 6, pp. 33 - 41.
105. Hargie O. and Dickson. (1999), 'Communication in Management Jossey – Bass', CA, USA.
106. Hasan H. (2008), 'Why and how can human-related measures support software development processes?', Journal of systems and software, Vol 81, No 7, pp.128 -1252.
107. Hayes J. (2002), 'Do You Like Pina Coladas? How Improved Communication Can Improve Software Quality', IEEE Software Society, Vol 20, No 1, pp. 90-92.
108. Heeks R. (1996), 'In India's Software Industry: State policy, liberalization and industrial development', Sage Publications.
109. Herbsleb J D. and Moitra D. (2001), 'Global software development', IEEE software, [ieeexplore.ieee.org](http://ieeexplore.ieee.org).
110. Hertel G. and Niedner S. (2003), 'Motivation of software developers in open source projects: An internet-based survey of contributors to the Linux kernel', Research Policy, 32, 1159 -1177.
111. Hofmann H F. and Lehner F. (2001), 'Requirements engineering as a success factor in software projects', IEEE Software, Vol 18, No 4, pp.58 – 66.
112. Hofstede G. (1991), 'Cultural and Organizations: software of the mind', London Mcgrawhill.
113. Holliday A. (2002), 'Doing and Writingh Qualitative Research', Sage Publications, London.
114. Hoyer R W. and Hoyer B Y. (2001), 'What is quality?', Quality Progress, No. 7, pp. 52-62, 2001.
115. Humphrey W S. (1989), 'Managing the Software Process', Addison-Wesley Publishing Co., and Reading, MA.
116. Humphrey M and Konrad D. (2005), 'Software process modeling' Springer, New York, Springer.
117. Humphrey W S. (2000), 'Introduction to the team software process', Addison -Wesley, 2000.

118. Humphrey W S. (1989), 'Managing the software process', Addison -Wesley, 1989.
119. Hyatt and Lawrence E. (1996), 'A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality', European Space Agency Software Assurance Symposium and the 8th Annual Software Technology Conference, 1996.
120. Ioannis S. and Lefteris A. (2003), 'Estimating the development cost of custom software' Information & Management, Vol 40, pp. 729 –741.
121. ISO/IEC 9126, ISO/IEC 9126:2001.Software engineering, (2001), Product Quality, International Organization for Standardization, Geneva, Switzerland.
122. Jacobson I. and Booch G (1999),' The Unified Software Development Process', Addison Wesley Longman, Inc., 1999.
123. Jackson. (2004), 'European management learning: a cross-cultural interpretation of Kolb's learning cycle', Journal of Management and Development, Vol 14, No 6, p.42.
124. James Jiang. and Gary Klein B. (1999), 'Software development risks to project effectiveness ', Available online 8 May 2000.
125. Javed T. and Durani.(2004), ' A survey to examine the effect of team communication on job satisfaction in software industry' , ACM SIGSOFT Software Engineering Notes ,Vol 29, No 2, pp. 6 - 6.
126. Jefferya R. and Ruheb M. (2000), 'Information and software technology ', Elsevier science ', Vol 9, No 50 PP. 01-08.
127. Jones C. (1986), 'Programming productivity', MC Grawhill, New York.
128. Jorge N. and Tschang T. (2008) ,'U Comparing Chinese and the Indian Software MNCs: Domestic and Export Market Strategies and their Interplay', NU-MERIT Working Paper Maastricht Economic and social Research and training centre on Innovation and Technology, UNUMERIT, August 2008.



129. **Jorge N. and Tschang T. (2009), 'The strategies of Chinese and Indian software multinationals: implications for internationalization theory', *Industrial and corporate change*, Vol 18, No 2, pp. 269 - 294.**
130. Judith C. and Kate M. (2007), 'Information Technology Workforce skills: Does Size matter?', *Information Systems Management*, Vol 24, No 4, pp.345-359.
131. Justin M. and Beaver Guy. (2006), 'The Effects of Development Team Skill on Software Product Quality' *ACM SIGSOFT Software Engineering Notes*, Vol 31, No 3 Copyright © 1995 Published by Elsevier Science B.V.
132. Kalwarski Tara and Daphne Mosher (2006), 'Best Jobs in America', *MONEY Magazine*, CNN, <http://money.cnn.com/magazines/moneymag/bestjobs/2006/>, Retrieved 2006-04-20.
133. Kaner C and Bach J. (2002), 'The Indian software services industry ', Vol 30, No 8, pp 1267-1287 October 2001.
134. Kan S. (2002), 'Metrics and models in software quality engineering', Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.
135. Katz E. (2001), 'Bias in conditional and unconditional fixed effects logistic estimation' *Political Analysis*, Vol 9, No 4, pp. 379 - 384.
136. Kekre S. and Krishnan M S.(1995) , 'Drivers of customer satisfaction for software products: implications for design and service support', *Management Science* ,Vol 41, No 9, pp. 1456 –1470
137. Kerlinger Fred N. (1973), 'Foundations of Behavioural research ', 2nd edition, NewYork, Holt, Reinhart and Winston.
138. Khoshgoftaar T M. and Munson J C. (1992), 'Predictive Modeling Techniques of Software Quality from Software Measures', *IEEE Transactions on Software Engineering*, Vol 18, No 5, pp. 979 - 987.
139. Kemerer C F. (1993), 'Staffing factors in software cost estimation models in Keyes', (Ed.), *Software Engineering Productivity Handbook*, Wind crest/McGraw-Hill.

140. Kemerer C F. (1992), 'Bridging the gap between research and practice in software engineering management: reflections on staffing factors paradox', in Rombach, H.D., Basili, V.R. and Selby, R.W. (Eds), *Experimental Software Engineering Issues*, Springer-Verlag .
141. Kettinger W J and Lee C. (1995), 'Global measures of information service quality: a cross-national study', *Decision Sciences*, Vol 26, No 5, pp. 569 – 588.
142. Konicki S. (2002), 'Growth in Enterprise Software Market', *Information Week*, May 31 2002.
143. Kortner A. (2002), 'Organizational Communication: Research and Practice' Educational Resources Information Center (ERIC) - Funded by US dept. of Education, ERIC Clearing house on Reading, English and Communication, Bloomington, IN, USA.
144. Kriebel C. (1979), 'Evaluating the quality of information systems in design and implementation of computer based information systems', chapter 2 , *The Netherlands*, pp .29-43.
145. Krishnan M S. (1998), 'The role of team factors in software cost and quality: an empirical analysis', *Information Technology and People*, Vol 11, No 1, pp. 20 –35.
146. Krishnan MS. Kriebel C H. (2000), 'An empirical analysis of productivity and quality in software products', *Management Science*, Vol 46 No 6, pp. 745 – 759.
147. Krishnan M S. and Ramaswamy S. (1998), ' An empirical analysis of customer satisfaction for Intranet marketing systems', *Decision Support Systems* ,Vol 24, No 1, pp. 45 – 54.
148. Krishnan M S. and Subramanyam R. (2004), ' Quality dimensions in ecommerce software tools: an empirical analysis of North American and Japanese markets', *Journal of Organizational Computing and Electronic Commerce*, Vol 14, No 4, pp. 223 – 241.
149. Kruchten P. (2000), 'The Rational Unified Process - An Introduction - Second Edition', Addison Wesley Longman, Inc., 2000.
150. Kluckholm F. and Strodtbeck F. (1961), 'Variations in value orientation', Westport, CT Greenwood Press.

151. Lakhani K. and Wolf R G. (2006), 'Motivation of software developers in Open Source projects: an Internet-based survey of contributors', *Research Policy*, Vol 32, pp.1159 -1177.
152. Lambert G N. (1984), 'A comparative study of systems response time on program developer productivity ', *IBM systems*, Vol 23, No 1, pp. 36 - 43.
153. Larman C. (2003), 'Agile and iterative development: a manager's guide 'agile software development series, series publications.
154. Laurent A .(1986), ' The cross cultural puzzle of international human resource management', *Human Resource Management*, Vol 25 p. 91, *The economist* , 1997 (September 27), p.119.
155. Leonardi P M. (2007), 'Activating the informational capabilities of information technology for organizational change' *Organization Science*, Vol 18, No 8, pp. 813 - 831.
156. Lethbridge T C. (2002), 'What knowledge is important to software professional?' *Computer*, Vol 33, No 5, pp 44 - 50I, SSN: 0018-9162 D O I: 10.1109/2.841783 Current Version Published: 2002-08-06.
157. Lee D M. and Trauth E M. (1995), ' Critical skills and knowledge requirements of IS professionals: a joint academic/ industry investigation', *MIS Quarterly*, Vol 19, No 3, pp. 313 – 340.
158. Litecky C. and Arnett K. (2004), ' The paradox of soft skills versus technical skills in IS hiring', *The Journal of Computer Information Systems*, Vol 45, No 1, pp. 69 –76.
159. Lorange P. (1986), 'Human Resource Management in Multicultural cooperative ventures', *Human Resource Management*, Vol 25 No1.
160. Marciniak J J. (2002), ' Encyclopedia of software engineering', Vol 3, 2nd ed., Chichester : Wiley.
161. Marc A. and Witold. (2007), 'In search for a widely applicable and accepted software quality model for software quality engineering', *Software quality journal*, Vol 15, No 4, pp.401 - 416.
162. Mayuram S. and Krishnan. (1998) , 'The role of team factors in software cost and quality An empirical analysis ', *Information Technology and people*, Vol 11, No 1, 1998, pp. 20-35, © MCB University Press, 0959-3845.

163. McCarthy Jim. (1995), 'Dynamics of Software Development', pp.10 – 30, August 1.
164. McCabe. (2006), 'A complexity measure', IEEE transactions of software engineering, Vol 2, No 4, pp.308 -316.
165. MCall's J A. and Richards P K. (1977), ' Factors in software quality ', Nat 'l tech Information service, Vol 1, No 1, pp 2- 3.
166. [Michael E.](#) and [Ray D.](#) (2007), 'An analysis of software quality management', software quality journal, Vol 15, No 4, pp.347 - 365.
167. Mitchell Charles (2000), 'International business culture', World trade press, California, P.37.
168. Mithas S. and Krishnan M S. (2007), ' Human Capital and Institutional Effects in the Compensation of Information Technology Professionals in the United States' , Management Science, Conditional Acceptance, 2007.
169. Mithas S. and Whitaker J. (2007), 'Is the world flat or spiky? Information intensity skills and global service disaggregation', Information Systems Research, Vol 18, No 3.
170. Murali Patibandla and Bent Petersen. (2002), ' Role of Transnational Corporations in the Evolution of a High-Tech Industry: The Case of India's Software Industry world development ' , Vol 30, No 9 sep 2002.
171. Myers T. and Myer E. (ed) (1982), ' Managing by Communication – An Organizational Approach', McGraw-Hill Book Company, New York.
172. Narendra M. and Agrawal. (2003), 'International Journal of Human Resources Development and Management', Vol 3, No 3, pp. 249- 264.
173. NASSCOM. (2005) National Association of Software and Service Companies , NASSCOM-McKinsey report 2005.
174. NASSCOM . (1999), 'The Software Industry in India: A Strategic Review', New Delhi, India
175. NASSCOM, (2000) National Association of Software and Service Companies, 'An Overview of the Indian Software Industry'.

176. Naur P. and Brian Randell. (2008), 'Garmisch, Germany: Scientific Affairs Division', NATO. <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF>. Retrieved 2008-12-26.
177. Nicholson B. and Sahay S. (2001), 'Some political and cultural issues in the globalisation of software development: case experience from Britain and India', *Information and organization*, Vol 11, pp.25 - 43.
178. Nicholson B and Sahay S. (2004), 'Global IT Outsourcing: Software Development Across Borders', *Communications of the ACM*, Vol 47 , No .4 , pp. 62 - 66
179. Nidumolu S R. and Goodman. (1993), 'Computing in India: An Asian Elephant Learning to Dance ', *Communications of the ACM*, Vol 36, No. 4, pp.15 – 22.
180. Nieva V F. and Fleishman E A. (1985), ' Team Dimensions, Their identity,Their Measurement and Their Relationships' , Washington D C US Army Research Institute for Behavioral and social Sciences.
181. Ohmae Kenichi. (1990), 'The Borderless World, New York, Harper Business.
182. Nunnally and Jum C . (1967), 'Psychometric theory ', New York, McGraw-Hill.
183. Paulk Mark and Weber. (1993), 'Capability Maturity Model for Software, Version 1.1', Software Engineering Institute, Carnegie Mellon University.
184. Peter Paul. (1981), 'Construct validity: a review of basic issues and marketing practices, *Journal of Marketing Research*, 28 May, 133 – 145.
185. Pradeep Sinha. (2003) , 'Computer fundamentals', BPB publications, New Delhi.
186. Philip Kotter. (1994), 'Marketing Management ', Prentice hall of India, New Delhi, p. 429.
187. Pressman R. (2005), 'Software engineering a practitioner's approach', Tata McGraw-Hill.
188. Pitt L F. and Watson R T. (1995), ' Service quality: a measure of information systems effectiveness', *MIS Quarterly* ,Vol 19, No 2, pp. 173 –187.
189. Pozzebon M. and Pinsonneault A. (2005), ' Global-local Negotiations for Implementing Configurable Packages: the Power of Initial Organizational

- Decisions', *Journal of Strategic Information Systems*, Special Issue Understanding the Contextual Influences on Enterprise System Implementation, (Part II), Vol 14 , No 2 , pp. 121-145
190. Prahalad C K. and Krishnan M S. (1999), 'The new meaning of quality in the Information Age', *Harvard Business Review*, Vol 77, No 5, pp. 109 –118.
  191. Ramasubbu N. and Mithas S. (2007), 'Work Dispersion, Process-Based Learning and Offshore Software Development Performance', Working Paper, Singapore Management University.
  192. Raccoon. (1995), 'The chaos model and the chaos life cycle', *ACM software engineering notes*, Vol 20, No 1, pp. 55 – 66.
  193. Reagans R and Boehm. (2005) , 'Individual experience and experience working together: Predicting learning rates from knowing who knows what and knowing how to work together, *Management Science*, Vol 51, No 6, pp. 869 - 881.
  194. Richard D C. (2000), 'Cross cultural business behavior', Copenhagen Business School Press, Denmark.
  195. Richard T. (2005), 'Software Engineering', Volume 1, The Development Process, 3rd Edition, Wiley-IEEE Computer Society Press.
  196. Richard W. and Selby. (2007), 'Software engineering', Published by Wiley-IEEE.
  197. Richard W. (2007), 'Lifetime Contributions to Software Development', Management and Research, Wiley-IEEE Computer Society Press.
  198. Robson C. (2002) , 'Real world research: a resource for social scientists and practitioner- Researchers', Blackwell Publisher Ltd.
  199. Robert S and Bradley R. (2006), 'The firm specificity of individual performance: Evidence from cardiac surgery' *Management Science*, Vol 52, No 4, pp. 473 - 488.
  200. Robert S and Bradley R. (2007), 'Team Familiarity, Role Experience, and Performance: Evidence from Indian Software Services' working paper, HBS Working Paper Number: 5785, September 2007.

201. Rose Jeremy. (2002), 'Interaction, transformation and information systems development – an extended application of soft systems methodology', *Information Technology and People*, Vol 15, No 3, pp. 242 - 268.
202. Royce W W. (1970), 'Managing the development of large software systems: Concepts and techniques', Wescon, August 1970.
203. Sackman H W. and Grant (1968), 'Exploratory experimental studies comparing online and offline programming performance' *Communication of ACM*, Vol 11, No 1, pp. 3 -10.
204. Sahay S and Brian N. (2001), 'Some political and cultural issues in the globalisation of software development: case experience from Britain and India's', *Information and Organization*, Vol.11, pp. 25 – 43.
205. Samer F and Lee S. (2000), 'Coordinating expertise in software development teams', *Management science*, Vol 46, No 12, pp.1554 -1568.
206. Sarker S. and Sahay S. (2003), 'Understanding Virtual Team Development: An Interpretive Study,' *Journal of the AIS (JAIS)*, Vol 4, Article 1, pp. 1-38.
207. Sheppard S B. (1979), 'First year results from a research program on human factors in software engineering', *National Computer Conference*, New York, NY, pp.1021- 27.
208. Shah Bankim. (1995), 'India's Software Industry', unpublished report, BRS Associates, Lake Worth, Florida.
209. Simon Commander. (2004), 'Must skilled migration be a brain drain? Evidence from Indian software industry', working paper IZA DP No. 1422.
210. Sinclair G. and Klepper. (2000), 'What's experience got to do with it? Sources of cost reduction in a large specialty chemicals producer', *Management Science*, Vol 46, No 1, pp. 28 - 45.
211. Smith B E. and Thomas J M. (1972), 'Cross cultural attitudes among managers: a case study', *Sloan Management Review*, Vol 13, No 3, p.35.
212. Stephen H. (2002), 'Metrics and Models in Software Quality Engineering', Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA
213. Smith K. (2001), 'The software Industry's Bug problem', *Quality Digest*, reproduced on <http://www.qualitydigest.com>, April 2003.

214. Sol E J. (2001), 'The embedded Internet towards 100 Billion devices', Eurostar conference, Stockholm, Sweden, 2001.
215. Stevens and Campion.(1994) , 'The knowledge, skill and ability requirements for team work: Implications for human resource management' , Journal of Management, Summer , pp.503 - 30.
216. Subramanyam R and Krishnan M S.(2003), 'Empirical Analysis of CK Metrics for Object-Oriented Design Complexity: Implications for Software Defects' , IEEE Transactions on Software Engineering, Vol 29, No 4, pp. 297 - 310, April 2003.
217. Suvillivan John. (2000), 'Preparing in the global economy', Economic reform today, No.1.
218. Sweeney Jillian. (1995), ' An investigation of a theoretical model of consumer perceptions of value, PhD thesis, curtain University, Australia.
219. Tracy H and Nathan B. (2007), 'Motivating developer performance to improve project outcomes in a high maturity organization in a high maturity organization', software quality journal, Vol 15, No 4, pp. 365 - 381.
220. Trompenaars F. (1993), 'riding the waves of culture: Understanding cultural Diversity in Business', Nicholas Brealey, and London.
221. Tschang Ted. (2001), 'The basic characteristics of skills and organizational capabilities in the Indian software industry', Working Paper, ADB Institute No. 137, February 2001.
222. Thadhani A. (1984), 'Factors affecting productivity during application development', IBM systems, Vol 23, No 2, pp. 19 - 35.
223. Thomson P. (2007). 'How much did the Liberty shipbuilders forget?' Management Science, Vol 53, No 6, pp. 908 - 918.
224. Timgoles. (2008), 'Information technology workforce skills: The software and IT services provider perspective', Information Systems Frontiers, Vol 10, No 2, pp.179 -194.
225. Todd P A and McKeen J D. (1995), 'The evolution of IS job skills: a content analysis of IS job advertisements from 1970 to 1990', MIS Quarterly, Vol 19 No 1, pp. 1 – 27.



226. Trienekens J. and Evan Veenendaal. (1997), 'Software quality from business perspective, Deventer , Netherlands, Kluwer .
227. Tuchman B W. (1980), 'The decline of quality', New York Times Magazine, November 2 1980.
228. Udell J. (1993), 'India's Software Edge', Byte, pp. 55-60.
229. Walsham G. (2002),' Cross-cultural software production and use: a structural analysis', MIS Quarterly, Vol 26, No 4, pp. 359 - 380.
230. Weinberg G. (1986), 'On Becoming a Technical Leader', Dorset House, New York, NY.
231. Whitaker K. (1994), 'Managing software Maniacs', Wiley and Sons publishers.
232. Watson H J. and Young D. (1990), ' Requisite skills for new MIS hires', ACM SIGMIS Database, Vol 21, No 1, pp. 20 – 29.
233. Zachary Pascal. (1998), 'Armed truce: software in an age of teams', Information Technology & People, Vol 11, No 1, pp. 62-65.
234. Zigurs I. (2003), 'Leading self-directed work teams: A guide to developing new team leadership skills', Organizational Dynamics, Elsevier.
235. Zollo M. and Winter S G. (2002), 'Deliberate learning and the evolution of dynamic capabilities', Organization Science, Vol 13, No 3, pp. 339 - 351.

**LIST OF PAPERS PUBLICATIONS**

1. Viji Vinod , Dr.Sunil Sahadev .(2009) , Software development team and quality – US and India International journal of quality and reliability management.
2. Viji Vinod and Dr. Sunil Sahadev .(2009), “Software team skills on software product quality”, *Asian Journal of Information Technology*, Vol 8, No 1 , pages .8-13.
3. Viji Vinod , Dr. Sunil Sahadev.(2008),” Impact of team skills on software quality : A comparative study of India and US “, *International Journal of Computing and Applications* , Vol 3, No 2, pages .135-160.
4. Viji Vinod. (2007), “Management and planning tools for quality”, national conference on Business Intelligence in Enterprise Resource planning “, 29 March 2007, page 5.
5. Viji Vinod, Golda Dilip,(2007),” Innovations of supply chain management”, National conference on Business Intelligence in Enterprise Resource Planning 29 March 2007.
6. Viji Vinod.(2005) ,” Modelling software processes : Directions for the future “, *Marketing in the age of convergence*, Jan 2005.

## VITAE

Born on the mainland of India in 1975, Professor Viji Vinod received her primary and secondary schooling at N.S.S arts and science co-educational College, India. She furthered her studies, graduating with a B.Sc (with Great Distinction) in Mathematics from Calicut University in 1996. Ms Viji Vinod has received her Masters in computer applications from Bharathiar University in 1999. Ms.Viji Vinod had started her career in working on real time software engineering. She joined as a software engineer at Pentafour communications ltd, Chennai, India in 1999 where she achieved in developing enterprise resource planning applications in client /server architectures. She has worked as a software developer with Lawrence and Associates ltd, a US based Software Company where she developed various web-based applications. She had the passion of doing research and reach great heights in academics, influenced her decision to join as the faculty of the Department of Information technology at Dr. M G R University in 2002, becoming senior lecturer of information technology management and computer science She was the first professor in object oriented analysis and design and component based technology at Dr. M G R University in 2004. From 2004 to 2005, she served as an Asst. Professor of department of Information Technology at Dr. M G R University and from 2006 to 2009, as an Asst .Professor of the department of Computer Applications at Dr. M G R University.

Professor Viji Vinod specializes in areas of information technology management and various software engineering issues with the goal of improving both software engineering practices and processes. She has edited one book and published articles and notes in professional journals and conferences.