

(Since 1950)



**ADDIS ABABA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES  
COLLEGE OF NATURAL SCIENCES  
SCHOOL OF INFORMATION SCIENCE  
DEPARTMENT OF INFORMATION SCIENCE**

**ASSESSMENT OF MAJOR FACTORS THAT LEAD TO  
SOFTWARE PROJECT FAILURE: A CASE STUDY OF ADDIS  
ABABA**

**BY  
MULLUALEM TEMECHE**

**ADDIS ABABA, ETHIOPIA  
MAY, 2014**

**ADDIS ABABA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES  
COLLEGE OF NATURAL SCIENCES  
SCHOOL OF INFORMATION SCIENCE  
DEPARTMENT OF INFORMATION SCIENCE**

**Assessment of Major Factors that Lead to Software Project Failure:  
A Case Study of Addis Ababa**

**A Thesis Submitted to the School of Graduate Studies, Addis Ababa  
University, in Partial Fulfillment of the Requirements for the  
Degree of Master of Science in Information Science**

**By:**

**Mullualem Temeche**

ADDIS ABABA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES  
COLLEGE OF NATURAL SCIENCES  
SCHOOL OF INFORMATION SCIENCE  
DEPARTMENT OF INFORMATION SCIENCE

Assessment of Major Factors that Lead to Software Project Failure:  
A Case Study of Addis Ababa

A Thesis Submitted to the School of Graduate Studies, Addis Ababa  
University, in Partial Fulfillment of the Requirements for the  
Degree of Master of Science in Information Science

By:

Mullualem Temeche

Signature of board members for this work approval:

Advisor	<u>Getachew Jemaneh</u>	signature		Date	<u>14/10/14</u>
Examiner	<u>Rahel Bekele</u>	signature		Date	<u>14/10/14</u>
Examiner	<u>Tibele Bestoh</u>	signature		Date	<u>13/10/2014</u>
Chairperson	<u>Tezome Alem</u>	signature		Date	<u>14/10/2014</u>

**Declaration**

I declare that the work I am submitting for assessment contains no section copied in whole or in part from any other source unless explicitly identified in quotation marks and with detailed, complete and accurate referencing.

*Signature*



**Dedication**

This work is dedicated to:

My father Ato Temeche Molla

## **Acknowledgement**

Glory be to God for giving me the strength to do this thesis. A special thanks to all organizations who were never tired to provide the needed information to complete this work, especially the departments of Information and Communication Technology (ICT).

I would like to express my sincere gratitude to my advisor, Ato Getachew Jemaneh, for his continuous guidance, encouragement, and excellent advice throughout the course of this work. Without his help, this work would not have been possible.

Finally, a very special thank you to my family, I am indebted to you all, especially my jewel of inestimable value, my wife, Elssa Mulate for her total support.

**Table of Contents**

Declaration ..... i

Dedication ..... ii

Acknowledgement ..... iii

List of Figures ..... vii

List of Tables ..... viii

List of Acronyms ..... ix

*Abstract* ..... x

CHAPTER ONE ..... 1

    1.1. Introduction ..... 1

    1.2. Statement of the Problem ..... 2

    1.3. Research Objectives ..... 4

        1.3.1. General Objective ..... 4

        1.3.2. Specific Objectives ..... 4

    1.4. Scope of the Study ..... 5

    1.5. Research Methodology ..... 5

        1.5.1. Data Collection Methods ..... 5

        1.5.2. Sampling Size and Selection Procedure ..... 6

        1.5.3. Data Presentation, Analysis and Interpretation ..... 6

1.5. Organization of the Research.....	7
CHAPTER TWO.....	8
Review of the Literature.....	8
2.1. Introduction.....	8
2.2. Software Project Failure.....	9
2.3. Major Causes of Software Project Failure .....	12
2.4. Software Defect Detections Strategies.....	19
2.5. System Development Methodologies: A framework for Comparison.....	19
CHAPTER THREE .....	22
Data Presentation, Analysis and Interpretation.....	22
3.1. Respondents Background.....	23
3.2. Software Project Failure from Different Perspectives .....	23
3.2.1. Software Development Services .....	23
3.2.2. Participation in software project development.....	25
3.2.3. Role of participants in Software development process.....	26
3.2.4. Number of projects in which respondents participated .....	27
3.2.5. Ratio of Successful to Failed Projects .....	28
3.2.6. Effects of Software Project Failure for Organizations.....	29
3.2.7. Skills and Abilities necessary for Team Members.....	30

3.2.8. Major Causes of Software Project Failure.....	31
3.2.9. Software Project Team.....	32
3.2.10. Methodology which can commonly minimize project failure rate.....	33
3.3. Defining Software Project Failure.....	34
3.3.1. Rated definitions of software project failure.....	34
3.4. Process of Developing Software.....	36
3.4.1. The process and the development team.....	36
3.4.2. The process and project management.....	37
3.4.3. The process and customer/users (internal/external).....	38
3.4.4. The process and project requirements.....	39
3.5. Discussion.....	40
CHAPTER FOUR.....	42
Conclusions, Recommendations and Suggestions for Future Researches.....	42
4.1. Conclusions.....	42
4.2. Recommendations and Suggestions for Future Researches.....	43
References.....	44
APPENDIXES.....	48

## List of Figures

Figure 3.1: The core business of the organization.....	19
Figure 3.2: Participation in Software Project Development.....	20
Figure 3.3: Role of participants in Software development process.....	21
Figure 3.4: Number of projects in which respondents are participated.....	22
Figure 3.5: Ratio of Successful versus Failed Projects.....	23
Figure 3.6: Effects of Software Project Failure for Organizations.....	24
Figure 3.7: Causes of Software Project Failure.....	26
Figure 3.8: Participants of Software Project Team.....	27
Figure 3.9: Methodology which can commonly minimize project failure rate.....	28

## List of Tables

Table 2.1 Software Project Failure and Its Cost from 1992-2005.....	8
Table 2.2 Yearly failure records or statistics of software projects in United States.....	12
Table 2.3 Early Warning Signs Ranked by Mean Importance Score.....	13
Table 3.1 Rated definitions of software project failure.....	29
Table 3.2 the process and the development team.....	31
Table 3.3 the process and project management.....	32
Table 3.4 the process and customer/users (internal/external).....	33
Table 3.5 the process and project requirements.....	34

## List of Acronyms

STRADIS	Structured Analysis, Design & Implementation of Information Systems
YSM	Yourdon Systems Method
IE	Information Engineering
JSD	Jackson Systems Development
OOA	Object-Oriented Analysis
RUP	Rational Unified Process
DSDM	Dynamic Systems Development Method
XP	Extreme Programming
ETHICS	Effective Technical & Human Implementation of Computer-Based Systems
SSM	Soft Systems Methodology
PRINCE	Projects in Controlled Environment
SSADM	Structured Systems Analysis & Design Method
IT	Information Technology
IS	Information System
ERP	Enterprise Resource Planning
CRM	Custom Relations Management
MIS	Mean Importance Score
ICT	Information and Communication Technology
CBE	Commercial Bank of Ethiopia
MES&T	Ministry of Ethiopian Science and Technology

## ***Abstract***

*Currently in most organizations everything becomes digitalized and every organization is influenced by this technological advancement. No organization is free from this effect in one or other ways. To cop up with this digital world, organizations propose to change or improve their systems each time. But most software projects can be considered at least partial failures because few projects meet all their cost, schedule, quality, or requirements. Failures are rarely caused by unsolved causes. The reason for software project failure can be technical or non technical. Whatever the reason is, it can be preventable and avoidable through deep investigation and understanding of factors that may lead to software project failure. The main objective of this research is to identify factors that lead to software project failure. The research was conducted based on the data collected from different software companies and from organizations that have long history in IT and software usage which are found in Addis Ababa. Different literatures were also reviewed and used as sources of secondary data for the research. Although not a comprehensive compilation of failure causes, the research has outlined several areas that may demand future researchers' attention.*

**Keywords:** Software project failure, factors for software project failure, software development methodologies

## CHAPTER ONE

### 1.1. Introduction

Companies spend much of their revenues on information technology each time to cop up with the changing business world. In other words, IT is now one of the largest corporate expenses (Charette, 2005, Grossman, 2003) outside employee costs. Much of that money goes into hardware and software upgrades, software license fees, and so forth, but a big chunk is for new software projects meant to create a better future for the organization and its customers. Most organizations which are found in countries like Ethiopia are at the transformation age to new technology and they spend much money to integrate their systems to a better one. They need new, outstanding (quality) and simple software that can enable them to control their financial transaction effectively and to handle their customers efficiently. On the contrary, absence of experienced software company in Ethiopia makes organizations very difficult to get software what they need and it leads to loss of valuable resources.

Software development is approaching 50 years as a profession. During this time there have been numerous success stories as well as many broadly publicized failures (Linberg, 1999).

IT projects usually fail when the rework exceeds the value-added work that has been budgeted for. Researcher proves that organizations waste billions of money each year on entirely controllable mistakes. Of the IT projects that are initiated, many are abandoned before or shortly after delivery. Many others are arrived late and over budget or require massive reworking. Few IT projects, in other words, truly succeed (Charette, 2005).

(Charette, 2005) "Governments, too, are big consumers of software." Failure of any one of these software projects can cost large amount of money and other important resources. IT failure in government can also endanger national security. But failure is common especially in countries like Ethiopia where there are no experienced and large software companies.

IT failures can also stunt economic growth and quality of life. A research by (Jones, 1998), shows that the failure or cancellation rate of large software systems is over 20 percent.

Some studies have shown that software specialists spend about 40 to 50 percent of their time on avoidable rework rather than on what they call value-added work, which is basically work that has done right the first time. Once a piece of software makes it into the failed, the cost of fixing an error can be 100 times as high as it would have been during the development stage (Jones, 2006).

The biggest tragedy is that software failure is for the most part predictable and avoidable. Unfortunately, most organizations do not see preventing failure as an urgent matter, even though that view risks harming the organization and maybe even destroying it (Charette, 2005).

### **1.2. Statement of the Problem**

The use of information technology (IT) has become pervasive in the management of organizations in all sectors in today's information economy (Linda, et al., 2004); it is impossible to imagine any organization, however small, not using computers in some manner to stay competitive. One of the main elements of the all-consuming influence of IT is software. Software is ubiquitous in our daily life. The increasing pressure of the information economy has driven executives in almost all organizations to look for that competitive and strategic advantage through the judicious use of software. Software has thus come to be viewed as a vital component of any organization's effort to remain competitive in the information economy. New software projects are constantly being proposed for development in organizations to meet their computing and information needs.

The expected growth and transformation of Ethiopia cannot easily be realized without software usage. Some of our undergoing projects are very big and complex that they cannot be managed by the old ways and means to reach the pinnacle of success according to plan and budget. By considering business and operating models pioneered by the software industry and tailoring them to their own needs, organizations can lower their costs, boost performance and turn software into a competitive advantage.

Ethiopia's private sector engaged in the software industry has not been able to show the fruits of technology and grow as observed in the other world. It is only the banking industry and a few big companies that have utilized the benefits gained from software of foreign companies. But, it is to be noted that these entities are using expensive software while there can exist ample resource locally with less price. The number of universities and colleges teaching IT has also grown but the country has not been capable of providing them the core job of software production.

Little achievements had been seen from most of the existing local software companies from the small opportunities offered to them. A few could have been competitive with the international companies if proper support was given from the state and the society.

Developing software systems is an expensive, and often a difficult process as software development projects are affected by a series of problems. It is never bug-free, and software bugs keep incurring monetary loss or even catastrophes. Despite the best efforts of these organizations, their well-thought-out plans for various software projects may end up as failures (Walid, et al., 2009). If we examine the literature we find some recent research that investigates failed software projects and provides details of the factors believed to be behind the project failures (Charette, 2005, Linberg, 1999, Verner et al., 2009). However, if we carefully study these projects most authors don't provide enough detail about them, and leave us with a number of questions regarding who or what was really behind the failure. Boehm suggested in 1991 that realistic schedule and budgets together with a continuing stream of changes in requirements are high risk factors for software development projects (Boehm, 1991). Besides, studies (Jones, 1998, 2004, 2006, 2008, Mohan et al., 2013, Procaccino, and Verner, 2006, Underwood, 2001, Sultan Ali Ebrahim, and Pawar, 2013, Bern et al., 2008, Verner et al., 2007) have confirmed that there are various factors that lead to software project failure. Researchers prove that organizations waste billions of money each year on entirely controllable mistakes. Of the IT projects that are initiated, many are abandoned before or shortly after delivery. Many others are arrived late and over budget or require massive reworking. Few IT projects, in other words, truly succeed (Charette, 2005). Software failure is becoming a big problem for most of the

organizations and there can be several reasons of software failure.

The researcher has interested in empirically checking the results of these prior studies and testing the validity of previously reported anecdotal evidence in a local context; thus this work builds on that previously reported by (Jones, 2004; Verner, et al., 2009; Bern, et al., 2008; CHAOS, 1995; Yeo, 2002; Kappelman, et al., 2006; Charette, 2005; Taylor, 2002; Al-Ahmad, et al., 2009; Jones, 2006) etc. Therefore, the researcher tries to solve the following basic research questions:

1. What factors lead to software project failure?
2. Can one make methodological setup to failing factors which can commonly minimize project failure rate?

### **1.3. Research Objectives**

#### **1.3.1. General Objective**

The main objective of the study is to identify the different factors that lead to software project failure and to investigate critical failure factor of software projects. The researcher finally recommends methodological solutions by studying the potentials of software development methodologies to minimize project failure rate in different situations.

#### **1.3.2. Specific Objectives**

Specifically, the research has tried to address the following areas.

- ✓ To identify different factors that lead to software project failure.
- ✓ To provide brief definition of software project failure.
- ✓ To show the recommended participants of software projects for its successful completion.
- ✓ To investigate software development process with respect to different parties ( like customers, managers, development team).

## **1.4. Scope of the Study**

The study focuses all about identifying software project failing factors and mechanisms to minimize project failure rate.

## **1.5. Research Methodology**

### **1.5.1. Data Collection Methods**

This thesis is aimed at identifying the critical failure factors for software project failure. The researcher used both primary and secondary data. The primary data was collected through questionnaires from some software companies and clients (users) and the questionnaires have four sections. The first section is about the respondents' background. The second section assesses software project failure from different perspectives. Software project failure have been tried to be defined from respondents point of view in the third section. The last section deals with the process of developing software. Besides, secondary sources include the annual reports, different related publications, organizational planning documents and search engines or web sources.

Based on extensive review of the software engineering literature, the researcher developed a questionnaire in order to investigate software developers' and software users' perceptions of software development practices that affect project outcomes. Some questions are developed by the researcher based on the main objective of the research and the remaining is adopted from other sources (Procaccino & Verner , 2006).

The researcher has considered items that have direct relevance to respondents, and items that are applicable to the outcome of the project as a whole (the final system). The following were the instructions included with the set of statements that respondents were asked to address using a five-point Likert scale and alternatives to be chosen by respondents.

- The first two sections of the questionnaire include questions with different alternatives and respondents were given the opportunity to select one or more alternatives.

- Here is the instruction for the third section: Each of the statements in this section relates to a specific aspect of the final product or result of developing software. For each of the following statements, rate how important you believe each is, in general, to your definition of software project failure.
- Finally, the fourth section instructs respondents as: Each of the statements in this section relates to a specific aspect of the process of developing software. Please rate how significant you believe each is in its contribution to what you consider a 'failed' software project.

### **1.5.2. Sampling Size and Selection Procedure**

Even though eleven questionnaires were distributed to ICT officers of eleven organizations, only ten of them were able to respond. Four governmental organizations, two banks and four software companies were the target organizations. The four governmental organizations were selected because of they have long experience of using different software (<http://www.mereja.com>). The two banks were also selected because of they are among the experienced banks in applying different e-banking systems (Tewdros, S., and Bhaskar, R.M., 2011). From the state owned banks Commercial Bank of Ethiopia (CBE) was selected and from private banks Wegagen bank was selected. Four software companies found in Addis Ababa, Ethiopia were also selected randomly. Random sampling technique was employed for the selection of sample software companies to minimize bias and to enhance the probability of all population to be included in the sample. The researcher first takes list of software companies from Ministry of Ethiopian Science and Technology (MES&t) and then four companies were selected randomly.

### **1.5.3. Data Presentation, Analysis and Interpretation**

For the research, data were collected, presented, analyzed and interpreted by employing different techniques to attain the already stated objectives of the study. The collected data were presented on pie chart, bar chart and table and then analyzed using quantitative technique.

## 1.5. Organization of the Research AFTER TWO

The research includes four chapters. The first chapter is about the introductory part of the research which emphasizes on the general overview of the study, the problem that the research focuses, objectives of the study, the scope of the study and the methodology used in conducting the research. In the second chapter related literatures are reviewed. And in the third chapter the collected data has been presented, analyzed, and interpreted. The last chapter includes conclusions and some recommendations forwarded by the researcher.

## CHAPTER TWO

### Review of the Literature

#### 2.1. Introduction

Software development is approaching 50 years as a profession. During this time there have been numerous success stories as well as many broadly publicized failures (Linberg, 1999).

Studies on system software projects show that the failure rate of large projects is higher than small projects. Since only few people like to admit failure, the real statistics of project failure may be more than what researcher says (CHAOS, 1995).

The aim of this chapter is to review relevant data available on software project failure factors from different sources. There is enough literature available to prove that software failure problem is faced by many organizations. In the past few decades, the business environment has been changed and facing rapid changes like technological advancement in sense of doing business and due to this almost everything partially or completely dependent on technology. Also the span of business is changing and shifting from local to global and now due to technological advancement, we are operating on global level and facing tough competition. "As the era of globalization proceeds and enterprises begin competing for the same customers". (Parker, 1995).

Today, IT is a major department in most of the organizations and its impact is felt throughout the company. As (Grossman, 2003) said in a report that "IT departments now consume, on average 50% of all corporate capital expenses. Yet, it is widely accepted that over 70% of IT projects fail, with software development representing most of them are over age". Further supported by (Saur & Cuthbertson, 2003) that an Oxford University survey reported that "only 16% of IT projects was successful, around 74% were 'challenged' and 10% were abandoned". In this situation it is important to examine some fundamentals, like "what is software failure?"

When the investigator started examining the failure phenomena, he came to know that how little is truly understood regarding failure which matter the most. It is further a problematic that there

is a lack of clarity as to what is software project failure? When a product/software is unable to perform, there can be countless reasons and on the other side sometimes difference between correct operation and failure is not clear. IT experts are agreed that such failures occur far more often than they should. Furthermore these failures are now well into the billions of dollars a year and that's why it is most important and urgent matter for an organization to primarily find its causes and secondly try to avoid them. So, the investigator needs to dig deep into this failure problem and look for its main causes and their solution. But due to limited time and scope of the research project, the researcher will just try to determine the root causes and present recommendations for avoiding software failure.

### **2.2. Software Project Failure**

After giving a brief introduction of the research topic, now the investigator is in a position to start with the main research problem and that is "software project failure". According to (George, 1988) in "Fortune 200 companies they had a chance of at least one software project runaway in their history and you can imagine for fortune companies what would be the cost of failure".

"As Software industry is growing along the time; the software is going more complex, diverse and costly. We still have some low budget software options available in market but in this case software quality suffers. On the other side maintaining successful rate in software industry is on stake" said by (Zeeshan, and Gerhard, 2007). In another words said by (Konstantinos, et al., 2005) that "the design and development of large scale software projects is a complex endeavor, often facing problems like cost and schedule overruns as well as low quality".

In spite of well-known research results and despite decades of individual and collective experience of managing projects, despite the rapid growth in membership of project management professional bodies and despite an increase in the amount of project working in the industry, project results continue to disappoint stakeholders (Davies, 2002).

Different scholars give different definitions for the phrase "software project failure". According to (May, 1998), a failure is defined as any software project with severe cost or schedule overruns, quality problems, or that suffers outright cancellation. Some other researcher also says

information systems projects may satisfy users requirements, may not be overruns, over budgeted, etc, but the project may have significant negative social ethical impacts. (Aken, 1996) even defined project success or failure as the satisfaction or dissatisfaction of all stakeholders. (Lyytinen and Hirschheim, 1987) as cited by (Yeo, 2002), looks Information System (IS) failure in to four notions:

**1. Correspondence Failure:** When the systems design objectives are not met, the information system is considered a failure. It is generally believed that design goals and requirements can be specified clearly in advance, and that their achievements can be accurately measured. Performance measures mainly based on cost-benefit analysis are employed for managerial control over the systems implementation. Correspondence failure, goal seeking in outlook, tends not to recognize that users may not necessarily accept systems that meet design objectives and specifications.

**2. Process Failure:** A process failure occurs when an IS cannot be developed within an allocated budget, and/or time schedule. There are two likely outcomes of process failure. Firstly, an outright failure occurs when no workable system can be produced. Secondly, a more common outcome is when an information system is developed with massive overspending in both cost and time, thus negating the global benefits of the system. This is a project level failure attributed to unsatisfactory project management performance.

**3. Interaction Failure:** The level of end-user usage of the information system is suggested as a surrogate in IS performance measurement. Some related measures of IS usage include user attitudes and user satisfaction, the amount of data transferred or the frequency of use. However, heavy usage does not necessarily mean high user satisfaction and improved task performance, and there is little empirical evidence supporting such a claim. Heavy systems usage might be a result of legal compulsion, persuasion, or that there are simply no other alternatives besides using the system.

**4. Expectation Failure:** The notion of expectation failure views IS failure as the inability of a system to meet its stakeholders' requirements, expectations, or values. Failure, therefore, does

not only involve the system's inability to meet design (technical) specifications. Expectation failure is perceived as the difference between the actual and desired situation for the members of a particular stakeholder group. Unlike the other three notions, IS failure is considered holistically in this case, as the views of different stakeholders are taken into account.

In the (CHAOS, 1995) report "only 16.2% of software projects were completed on time and on budget". According to (woodhapper and McManus, 2003) in the last decade a significant number of project failures have been reported. Research also indicates that in case of software projects there are many problems.

It is very important to present some concrete examples of software project failure in our economy. For this purpose the investigator has selected a table which presents some examples of software project failure. This table covers the time span from 1992 to 2005 and different industries as well as their amount of loss. After reviewing this table everyone will realize how big this failure problem and that is why the researcher needs to find out its real causes and focus on their solution.

Table 2.1 Software Project Failure and Its Cost from 1992-2005.

Year	Company Outcome	(Costs in US \$)
2005	Hudson Bay Co. [Canada]	Problems with inventory system contribute to \$ 33.3 million loss.
2004	Avis Europe PLC [UK]	Enterprise resource planning (ERP) system cancelled after \$ 54.5 million is spent.
2004	Hewlett-Packard Co.	Problems with ERP system contributes to \$ 160 million loss.
2003-04	AT&T Wireless	Customer relations management (CRM) upgrades problems lead to revenue loss of \$ 100 million.
2002	McDonald's Corp.	The information purchasing system cancelled after \$ 170 million is spent.
2001	Nike Inc.	Problems with Supply chain management system contribute to \$ 100 million loss.
2000	Washington, D.C.	City payroll system abandoned after deployment costing \$ 25 million.

1999	Hershey Foods Corp.	Problems with ERP system contribute to \$ 151 million loss.
1998	Snap-on Inc.	Problems with order entry system contribute to revenue loss of \$ 50 million.
1997	U.S. Internal Revenue Service	Tax modernization effort cancelled after \$ 4 billion is spent.
1996	Arianespace [France]	Software specification and design errors cause \$ 350 million Ariane 5 rocket to explode.
1995	Toronto Stock Exchange [Canada]	Electronic trading system cancelled after \$ 25.5 billion is spent.
1994	Chemical Bank	Software error causes a total of \$ 15 million to be deducted from 100,000 customer accounts.
1993	London Stock Exchange [UK]	Taurus stock settlement system cancelled after \$600 million is spent.
1992	Budget Rent A Car, Hilton Hotels, Marriott International, and AMR [American Airlines]	Travel reservation system cancelled after \$ 165 million is spent.

Source: Robert, N. Charette, 2005

### 2.3. Major Causes of Software Project Failure

Data exists on the causes of software project failure and several studies have mentioned some common causes of software project failure like “lack of project management, organizational culture and structure, user involvement and training, lack of project planning, scope and objectives setting” etc by (Jones, 2004; Verner, et al., 2009; Bern, et al., 2008; CHAOS, 1995; Yeo, 2002; Kappelman, et al., 2006; Charette, 2005; Taylor, 2002; Al-Ahmad, et al., 2009;

Jones, 2006). In different studies they have mentioned different causes and symptoms of software project failure. After reviewing the available literature; the investigator has listed out different software project failure factors which are believed to be major factors for software failure by different authors. Note that the factors are not presented in their order of importance.

According to (Jones, 2004), inadequate project planning, inadequate cost estimating, inadequate measurements, inadequate milestone tracking, inadequate quality control, and ineffective change control are the major causes of software project failure. (Jones, 2006) also adds some other factors like; Inaccurate Estimating and Schedule Planning, Incorrect and Optimistic Status Reporting, Unrealistic Schedule Pressures, Inadequate Quality Control, and New and Changing Requirements during Development and he was believed that these are the “root causes of software project failure.” Besides, (Verner, et al., 2009), after reviewing different literatures, concludes that the followings are “common project failure factors.”

- 1) The delivery date impacted the development process; 93% of the failed projects;
- 2) The project was underestimated; 81% of the failed projects;
- 3) Risks were not re-assessed, controlled, or managed through the project; 76% of the failed projects;
- 4) Staff were not rewarded for working long hours; 73% of the failed projects;
- 5) Delivery decision was made without adequate requirements information; 73% of the failed projects;
- 6) Staff had an unpleasant experience working on the project; 73% of failed projects.

On the other hand, (Bern, et al., 2008), categorizes project failing factors into nine categories as:

**Customer:** four contextual factors were identified: requirements clarity, level of involvement, customer orientation, and level of customer management formalization.

**Competence:** four factors were identified: level of software engineering skills, working experience, level of business knowledge, and level of application domain knowledge.

**Organizational culture:** four factors were observed: attitude to process improvement, attitude to quality, work time policy, and personal effect on working practices.

**Organizational structure:** two factors were identified: number of roles, and consistent understanding of software engineering practices.

**Business environment:** six factors were identified: level of competition, level of trust, attitude to subcontracting, cost level, language barrier, and physical distance.

**Operational management:** four factors were identified: resource management approach, schedule management approach, project management approach, and company quality assurance.

**Strategic management:** two factors were identified: business strategy, and awareness of the strategy throughout the organization.

**Knowledge transfer:** two factors were identified, efficiency of communication, and level of documentation.

**Company infrastructure:** six factors were identified: level of methods and process formalization, selection of tools, amount of training available, level of change management formalization, availability of reusable artifacts, and availability of a common architecture.

The Standish Group Report (1995) also shows the yearly failure records or statistics of software projects in United States. According to their report failure rate becomes large with respect to company and project size. For the purpose of simplicity the Standish Group Report classifies companies as large, medium and small based on the company's capital investment and also projects as project success, project challenged, and project impaired (canceled). Accordingly, the average success rate is 16.2%, challenged projects account for 52.7% in average and 31.1% of projects were cancelled before completion after they cost millions of dollars. The report also outlines project success factors, challenged factors and impaired factors as follows:

Table 2.2 Yearly failure records or statistics of software projects in United States

Project Success Factors	Project Challenged Factors	Project Impaired Factors
<ul style="list-style-type: none"> <li>- User Involvement</li> <li>- Executive Management Support</li> <li>- Clear Statement of Requirements</li> <li>- Proper Planning</li> <li>- Realistic Expectations</li> <li>- Smaller Project Milestones</li> <li>- Competent Staff</li> <li>- Ownership</li> <li>- Clear Vision &amp; Objectives</li> <li>- Hard-Working, Focused Staff</li> </ul>	<ul style="list-style-type: none"> <li>- Lack of User Input</li> <li>- Incomplete Requirements &amp; Specifications</li> <li>- Changing Requirements &amp; Specifications</li> <li>- Lack of Executive Support</li> <li>- Technology Incompetence</li> <li>- Lack of Resources</li> <li>- Unrealistic Expectations</li> <li>- Unclear Objectives</li> <li>- Unrealistic Time Frames</li> </ul>	<ul style="list-style-type: none"> <li>- Incomplete Requirements</li> <li>- Lack of User Involvement</li> <li>- Lack of Resources</li> <li>- Unrealistic Expectations</li> <li>- Lack of Executive Support</li> <li>- Changing Requirements &amp; Specifications</li> <li>- Lack of Planning</li> <li>- Didn't Need It Any Longer</li> <li>- Lack of IT Management</li> <li>- Technology Illiteracy</li> </ul>

(Kappelman, et al., 2006), identifies early warning signs that can lead to software project failure as shown in the following table.

## Why Software Projects Fail? 2014

Table 2.3 Early Warning Signs Ranked by Mean Importance Score (MIS) (7 = Extremely Important, 1 = Extremely Unimportant)

Rank	Item Description	Source	MIS
1	Lack of top management support or commitment to the project	Schmidt et al., 2001	6.59
2	Functional, performance, and reliability requirements and scope are not documented	Winters, 2002	6.58
3	Project manager(s) cannot effectively lead the team and communicate with clients	Schmidt et al., 2001	6.38
4	No change control process	Schmidt et al., 2001	6.33
5	Project stakeholders have not been interviewed for project requirements	Ward, 2003	6.32
6	No documented milestone deliverables and due dates		6.30
7	Undefined project success criteria		6.22
8	Project team members have weak commitment to the project scope and schedule	Schmidt et al., 2001	6.17
9	Communication breakdown among project stakeholders	May, 1998	6.17
10	Key project stakeholders do not participate in major review meetings		6.16
11	Project team members do not have required knowledge/skills	Barki et al., 2001	6.16
12	Project resources have been assigned to a higher priority project	Havelka et al., 2004	6.12
13	No business case for the project	Ward, 2003	6.11
14	No project status progress process	Havelka et al., 2004	6.11
15	Schedule deadline not reconciled to the project schedule		6.09
16	Early project delays are ignored — no revision to the overall project schedule	McKeeman, 2001	6.04
17	Subject matter experts are overscheduled: retain all prior duties yet expected to Provide substantial participation to the project	McKeeman, 2001	6.04

## Why Software Projects Fail? 2014

18	No planning and estimation documentation	Jones, 2004	5.96
19	Project managers have poor training	Schmidt et al., 2001	5.94
20	Key stakeholders do not review and sign off deliverables on a timely basis		5.93
21	Project stakeholder decision delays have caused due dates to be missed		5.93
22	No due diligence on vendor(s) and team members	McKeeman, 2001	5.91
23	No written commitment for the project outside of the project team		5.88
24	Significant goal, scope, or schedule requirements change immediately after project kickoff	Boehm, 1991	5.85
25	Team members have undefined roles and responsibilities	Jiang et al., 2002	5.83
26	No project communications plan or resources devoted to managing and communicating project expectations		5.80
27	Project team members are overscheduled	Schmidt et al., 2001	5.77
28	Users are not willing to cooperate	Schmidt et al., 2001	5.75
29	No team member experience with the chosen technology	Schmidt et al., 2001	5.73
30	No project management methodology	Schmidt et al., 2001	5.67
31	No project charter document at early stage of project		5.65
32	No risk analysis documentation and process	McKeeman, 2001	5.65
33	Failure to gather requirement via joint application design		5.63
34	No documented analysis of business strategy alignment	Winters, 2002	5.61
35	Major new risks are identified after the project kickoff		5.59
36	No performance and reliability requirements metrics tracking process	Jones, 2004	5.57
37	Approved project budget less than budget estimated by the project team		5.56
38	Budget, schedule, scope, and quality all mandated from outside the project team		5.56
39	Project manager(s) have never managed a project of this scale before	McFarlan, 1982	5.55

40	Deliverable due dates missed during the first 10 percent of the project schedule	McKeeman, 2001	5.54
41	IT operations infrastructure and network infrastructure problems have major impact on project team productivity		5.52
42	Difficulty in determining the input and output of the system		5.51
43	Cultural conflict among organizations involved	Winters, 2002	5.50
44	No contingency budget for known risks and rate of changes		5.50
45	Unstable organization environment (such as changes in senior management or restructuring)	Schmidt et al., 2001	5.49
46	Project team member(s) have low morale	McKeeman, 2001	5.48
47	Key team member turnover after project kickoff	Schmidt et al., 2001	5.45
48	Key stakeholders have not signed the project charter		5.36
49	Large number of interfaces to other system required	Barki et al., 2001	5.30
50	Users cannot get involved because of lack of understanding of new system Capabilities	McFarlan, 1982	5.29
51	Project involves implementing a custom or beta version of hardware or software	Schmidt et al., 2001	5.10
52	Users or technical support team feel threatened by a project to replace their legacy system	Jiang et al., 2002	4.80
53	Earned value systems not in place or used to control program		4.56

*Source: Kappelman, et al., 2006.*

(Charette, 2005), identifies project failure factors and he was believed the following are the most common: Unrealistic or unarticulated project goals, Inaccurate estimates of needed resources, Badly defined system requirements, Poor reporting of the project's status, Unmanaged risks, Poor communication among customers, developers, and users, Use of immature technology, Inability to handle the project's complexity, Sloppy development practices, Poor project management, Stakeholder politics and Commercial pressures. (Taylor, 2002), who is the president and CEO of Systems Engineering Corporation, identifies five major reasons for project failure: Lack of

Leadership, Lack of Clarity, Little or no Due Diligence is performed, Lack of Accountability and Lack of a Proven Process. (Al-Ahmad et al., 2009), classify root causes of IT project failure as:

*Project Mgt Factors:* User involvement, Scope and objectives, ...

*Top Mgt Factors:* Lack of a champion, Lack of commitment, ...

*Technology Factors:* Developer expertise, Lack of commitment, ...

*Organizational Factors:* Culture / structure, Conflicting interests, ...

*Complexity / Size Factors:* Large and multifaceted project, Complex project, ...

*Process Factors:* No/unsuitable project mgt process, Conflicting interests, ...

The good thing the researcher has found when he has reviewed different works on this area is almost all researchers are agreed on one thing. They have revealed that only some projects are affected by one or two factors and most other projects are affected by more than two factors.

### **2.4. Software Defect Detections Strategies**

There are different approaches to identify defects like inspection, prototyping, testing and correctness proof. Inspection is examination of human artifacts to detect defects at the early stages of software development. It is the most effective and efficient quality assurance technique. A prototype is an experimental version of software release. It helps both customer and developer to verify that the product meets all the stipulated requirements. It enables both parties to resolve ambiguous requirements to a well-defined specification. Thus, prototyping eliminates defects caused due to ambiguity. Testing is quality control activity that identifies defects at the time of implementation. It uncovers those defects, which could have escaped by identification at the early stage of development. Correctness proof discovers defects at coding stage. The code that fails to meet the requirements of correctness proof indicates existence of defect (Vasudevan, 2005).

### **2.5. System Development Methodologies: A framework for Comparison**

System development methodology (SDM) is a standard process followed by an organization to conduct all the steps necessary to analyze, design, implement, and maintain information systems

(Hoffer et al., 1999).

Various criterions can be used for the comparisons of methodologies. Some of these criterions are (Benzzine, 2002):

**Processes definition:** By process definition we mean does the methodology assume that every step in the development life cycle is well known? And that there is no room for environmental changes? Or does the methodology allow a big deal of flexibility in all stages of the development cycle to incorporate those changes that needs to be incorporated in the project before its closure.

**Final product determination:** Another criterion would be related to the final product. Does the methodology define the final product early on in the planning stage or does it define it during the project and close to project closure time

**Project cost:** This criterion is related to the estimation of the cost of the project and at what stage of the project it's done.

**Project completion date:** Estimation of a schedule of deliverable based on the estimation of the tasks to be accomplished. Is that done up front or as the project progresses?

**Responsiveness to environment:** This criterion measures the flexibility that the methodology allows to incorporate changes during the project, changes due to the environment, technology, competition or other.

**Team dynamic and creativity:** The steps predefined by the methodology could be an obstacle to creativity among teams, as the linear model suggests that some work needs to be done first by a small group and then the project moves to the next stage. This criterion tries to measure the ability of allowing some team work and interactions among the team members.

**Role of the upper management:** Is management an obstacle in creating a better system, or is the role of management to empower the team by taking care of any obstacles that impact the team performance.

**Training and knowledge:** Does the methodology step allow training and knowledge transfer during the project or does it put limitations to what a team member can do and learn.

**Probability of success:** This criterion measures the probability of success of a project using a certain methodology by embracing a certain degree of complexity and unpredictability.

## CHAPTER THREE

### **Data Presentation, Analysis and Interpretation**

This chapter presents, analyzes and interprets the data obtained from the primary source. In the second chapter, enough secondary data has been used to find the main causes of software project failure. In this chapter, the researcher tries to cross check these causes and validate them with the help of questionnaire survey which was designed for software developer companies and some of their clients. Due consideration and care has been taken to ensure that the data presented is from reliable sources. The primary data was obtained from the questionnaire which is designed to collect the necessary data to answer the research questions. The questionnaire was administered for eleven organizations, situated in Addis Ababa, Ethiopia and ten of them (91%) were correctly administered and used for analysis.

This chapter is organized in the same way the questions in the questionnaire are organized. Accordingly, the chapter has four sections. The first section is about the respondents' background. The second section assesses software project failure from different perspectives. Software project failure have been tried to be defined from respondents point of view in the third section. The last section deals with the process of developing software.

## 3.1. Respondents Background

As shown in the methodology section the target respondents were some software companies found in Addis Ababa and their customers who use different software in different situations. The questionnaire survey results revealed that almost all (90%) of respondents were male. 70% of respondents were between 25 to 34 years old. The remaining 20% and 10% of the respondents were found between 35 to 44 and 45 to 54 years old respectively. When we look the respondents' educational background, most of them (80%) have bachelor degree and the remaining have graduate degree.

From the respondents profile, in most organizations there is gender disproportion. But most of their employees are found in a productive age range. The Standish Group Report (1995) shows technology illiteracy is one factor for software project failure. In this respect, organizations need to train and upgrade the technological knowhow of their man power use this productive man power effectively for their success.

## 3.2. Software Project Failure from Different Perspectives

### 3.2.1. Software Development Services

In most situations software companies are requested for development of different types of software. In other words, some organizations may need software which cannot longer use after a certain task is completed. Or some other companies may need software which can be used for long period of time. The business of software companies is also different. The reason may be their financial capacity, man power, technological and technical understanding, market need and soon. The following chart shows the business of sample software companies and need of clients.

## Responses of software companies

## Response of clients

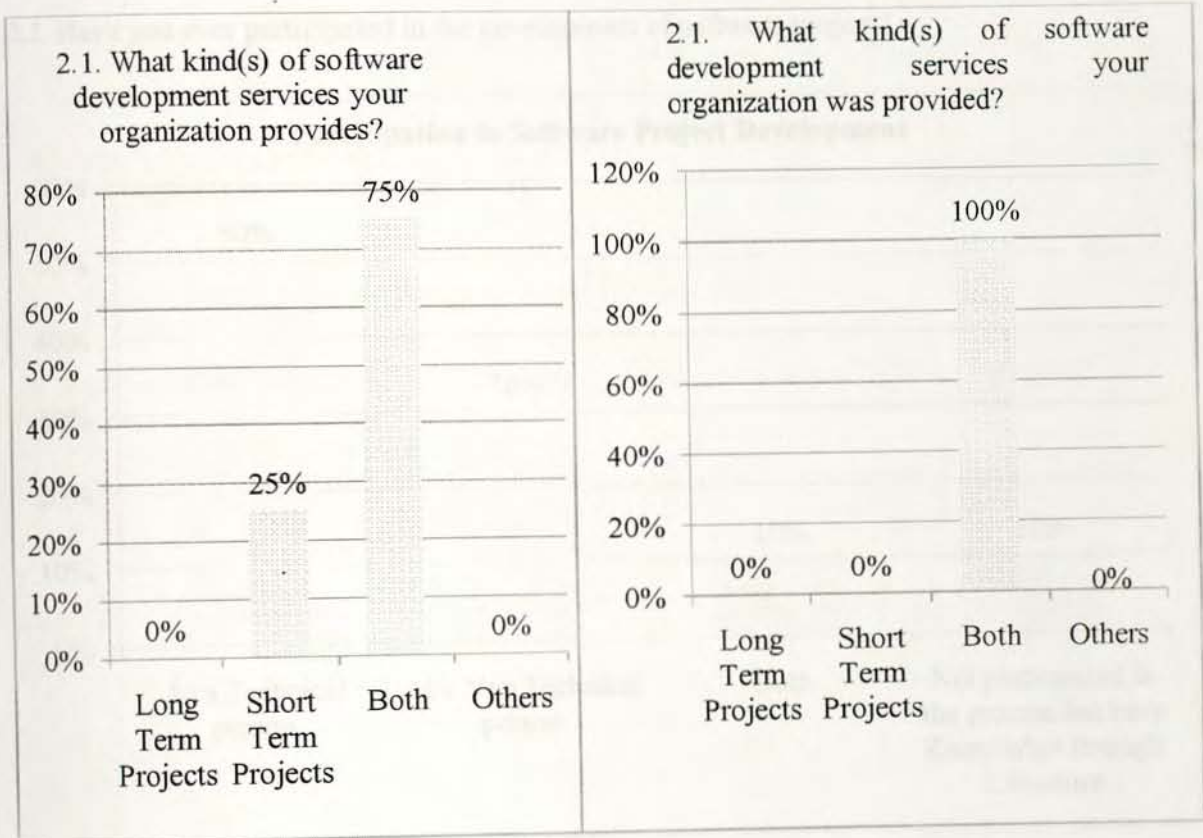


Figure 3.1: The core business of the organization

In Figure 3.1, there are 75% software companies who are working on both “long term and short term projects” and only one company (25% of the respondents) working on “short term projects”. This data shows that most software companies have the knowledge and experience of working on both the long term and short term projects. On the other hand, all sample client organizations invite software companies for the development of both short term and long term projects. To sum up, majority of respondents (software companies and client companies) are working on both long term and short term projects.

3.2.2. Participation in software project development

2.2. Have you ever participated in the development of software project?

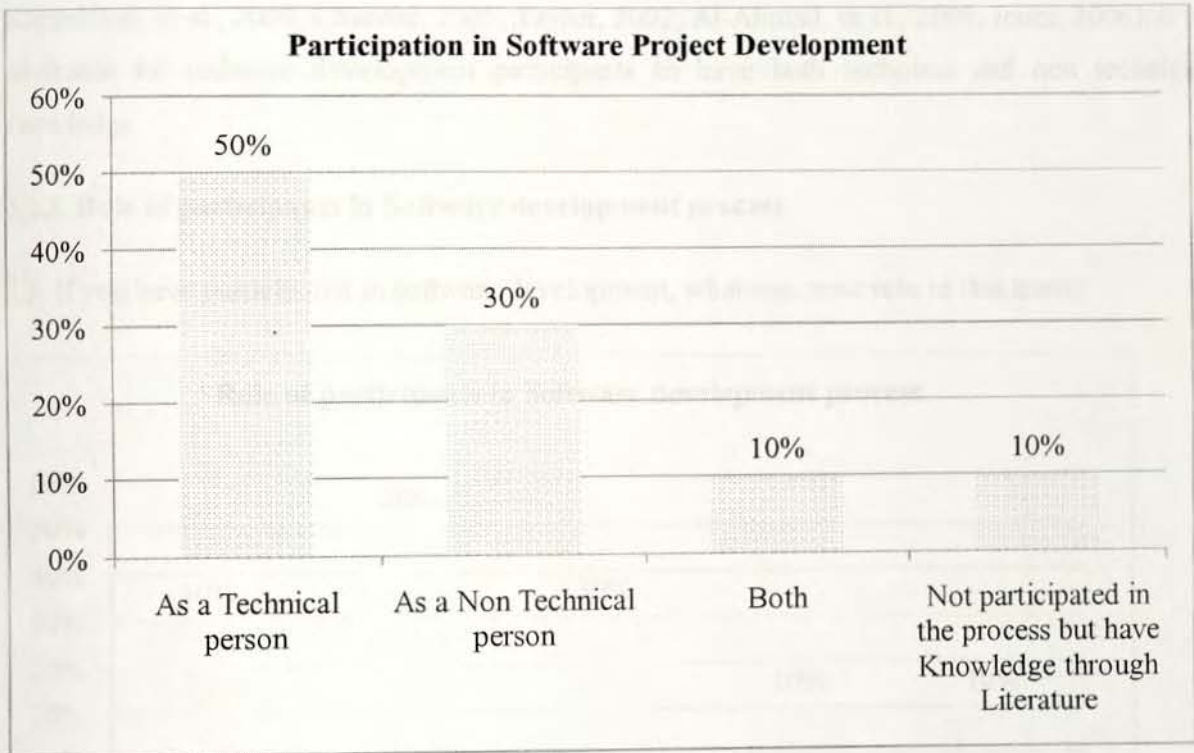


Figure 3.2: Participation in Software Project Development

The purpose of question 2.2 was to know about the respondent’s area of expertise and their knowledge about the software projects. Accordingly, as we see from Figure 3.2, there are 50% respondents with technical knowledge and experience. On the other hand 30% of the respondents were participating in software development projects as a non technical person. There were 10% respondents who have the knowledge about the process of software development through literature review and the same number of respondents (10%) involved in this research were participated in different software development projects as a technical as well as non technical person.

From this we can deduce that majority of the respondents have technical contribution in software projects. But since lack of effective project management is one of the major causes of software project failure (Jones, 2004; Verner, et al., 2009; Bern, et al., 2008; CHAOS, 1995; Yeo, 2002; Kappelman, et al., 2006; Charette, 2005; Taylor, 2002; Al-Ahmad, et al., 2009; Jones, 2006), it is advisable for software development participants to have both technical and non technical knowledge.

### 3.2.3. Role of participants in Software development process

2.3. If you have participated in software development, what was your role in that team?

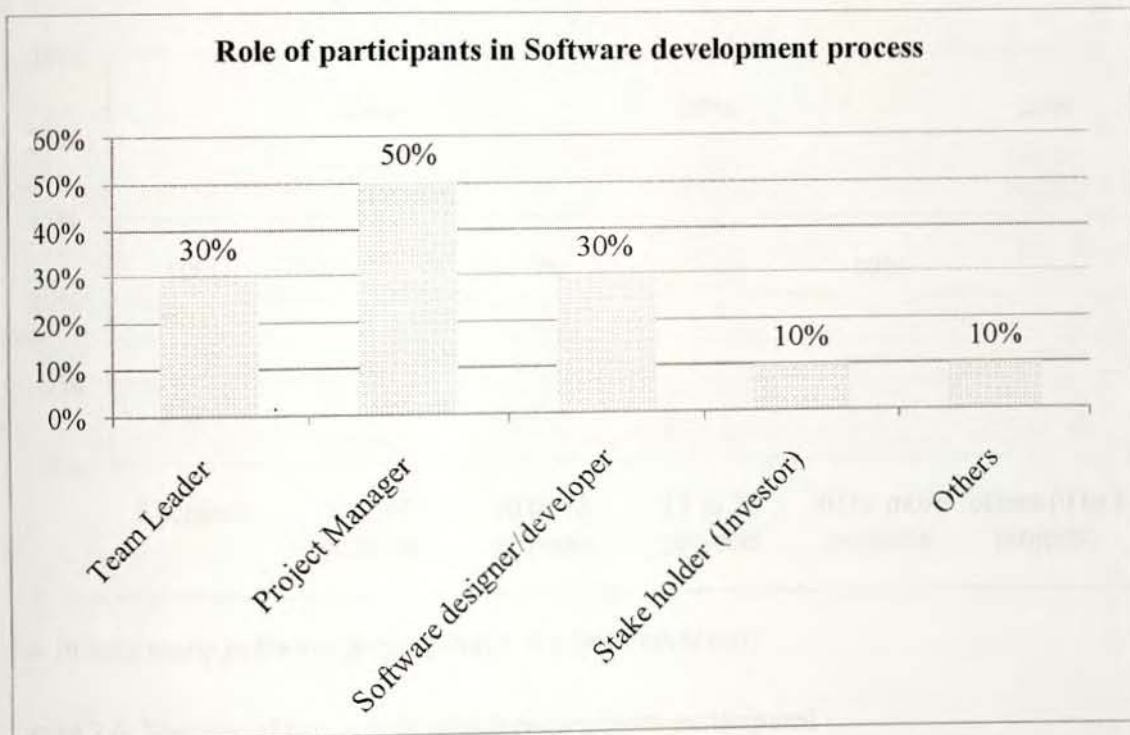
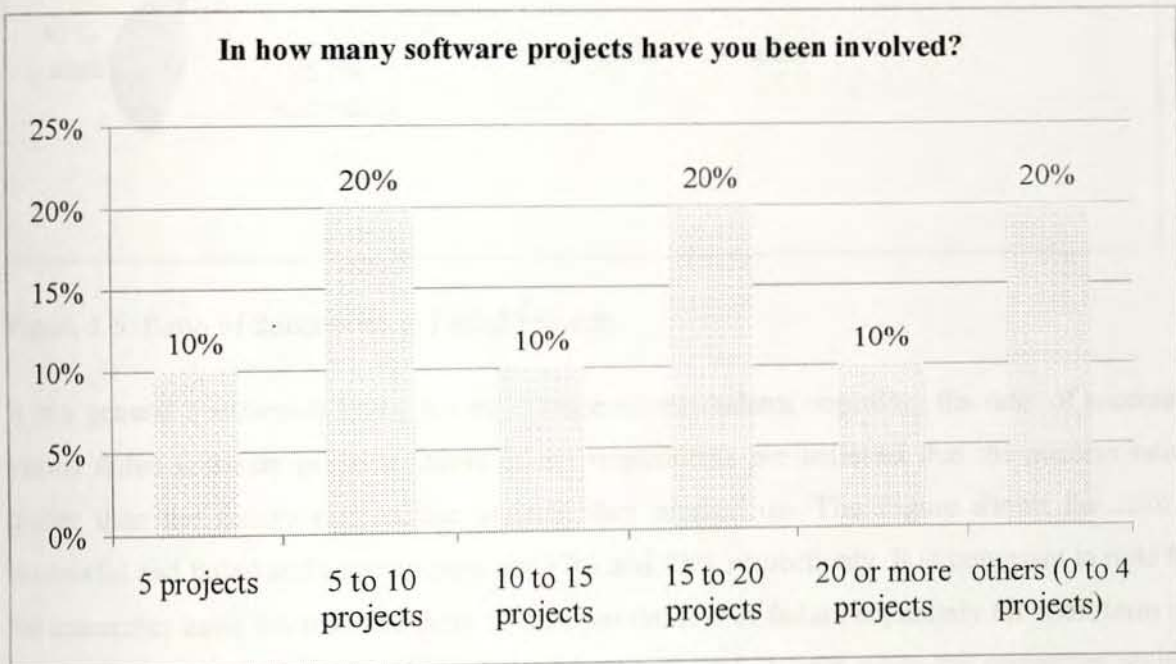


Figure 3.3: Role of participants in Software development process

In the above Figure, 50% of the respondents are project manager. Second highest ratio is 30% and these respondents are working as a team leader and the same ratio (30%) accounts for respondents who are software designers/developers. On the fourth level, there are people who are working as a project manager and as a programmer and it accounts for 10% of the respondents.

The remaining 10% of the participants are stakeholders (investors). As we remember from Figure 3.2 above, one respondent does not participate in software development but he/she knows most of software development procedures from literatures. This respondent does not specify its role as well. In conclusion, from Figure 3.2 and 3.3 above, we can deduce that the majority of the respondents are project managers who have technical knowledge and experience.

### 3.2.4. Number of projects in which respondents participated



2.4. In how many software projects have you been involved?

Figure 3.4: Number of projects in which respondents participated

In this Figure, respondents were asked to tell in how many software projects they have took part. Accordingly, there are 20% respondents who worked up to 4 projects. The same percentage is also for respondents who are participated for 5 to 10 and 15 to 20 projects. Further, there are some respondents who had worked on five projects, 10 to 15 projects and 20 or more projects

and those respondent accounts for 30% of the total respondents with the same ratio of 10% in each category and it also includes those who just worked on one single software project.

### 3.2.5. Ratio of Successful to Failed Projects

2.5. What was the ratio of successful versus failed software projects in which you were involved?

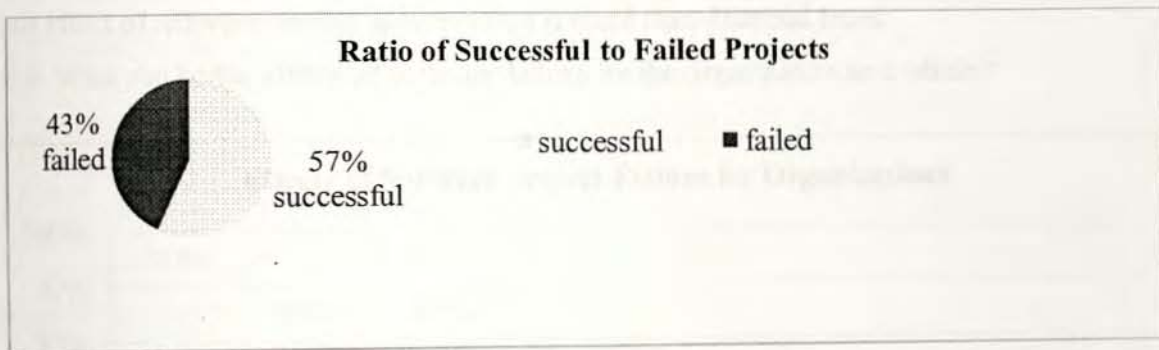


Figure 3.5: Ratio of Successful to Failed Projects

It is a general question to know the experience of respondents regarding the ratio of successful versus failed software projects. Most of the respondents are believed that the success rate is higher than the failure rate in the projects they worked on. The Figure shows the ratio of successful and failed software projects are 57% and 43% respectively. It is important to note that the researcher have never asked them to mention the rate of failure separately for short term and long term projects. It was briefly presented in the second chapter when the researcher reviews different literatures. Studies revealed that there are very less rate of failure in short term projects and in case of long term projects the rate is very high. In this survey response, the researcher has got totally opposite experience comparing to the results found from literature review. There could be different reasons; one is that respondents are sometimes mixing short term projects with long term projects. The other reason is that most of the respondents are not able to differentiate challenged with failed projects. They consider challenged projects as successful. The other most important reason is that sometimes they never feel good to accept that they have done something wrong because only few people like to admit failure.

### 3.2.6. Effects of Software Project Failure for Organizations

When software projects end up as a failure, it can have numerous impacts on organizations. But most organizations give more attention for financial losses even if failing to look other effects may harm the existence of the organization. Whatever the project failure factor is, if a project ends up as a failure, it has effects which are shown in the following chart with respect to its proportion of danger. The purpose of answering this question is to make organizations alert on the effect of software project failure which is more than financial losses.

2.6. What can be the effects of software failure for the organization as a whole?

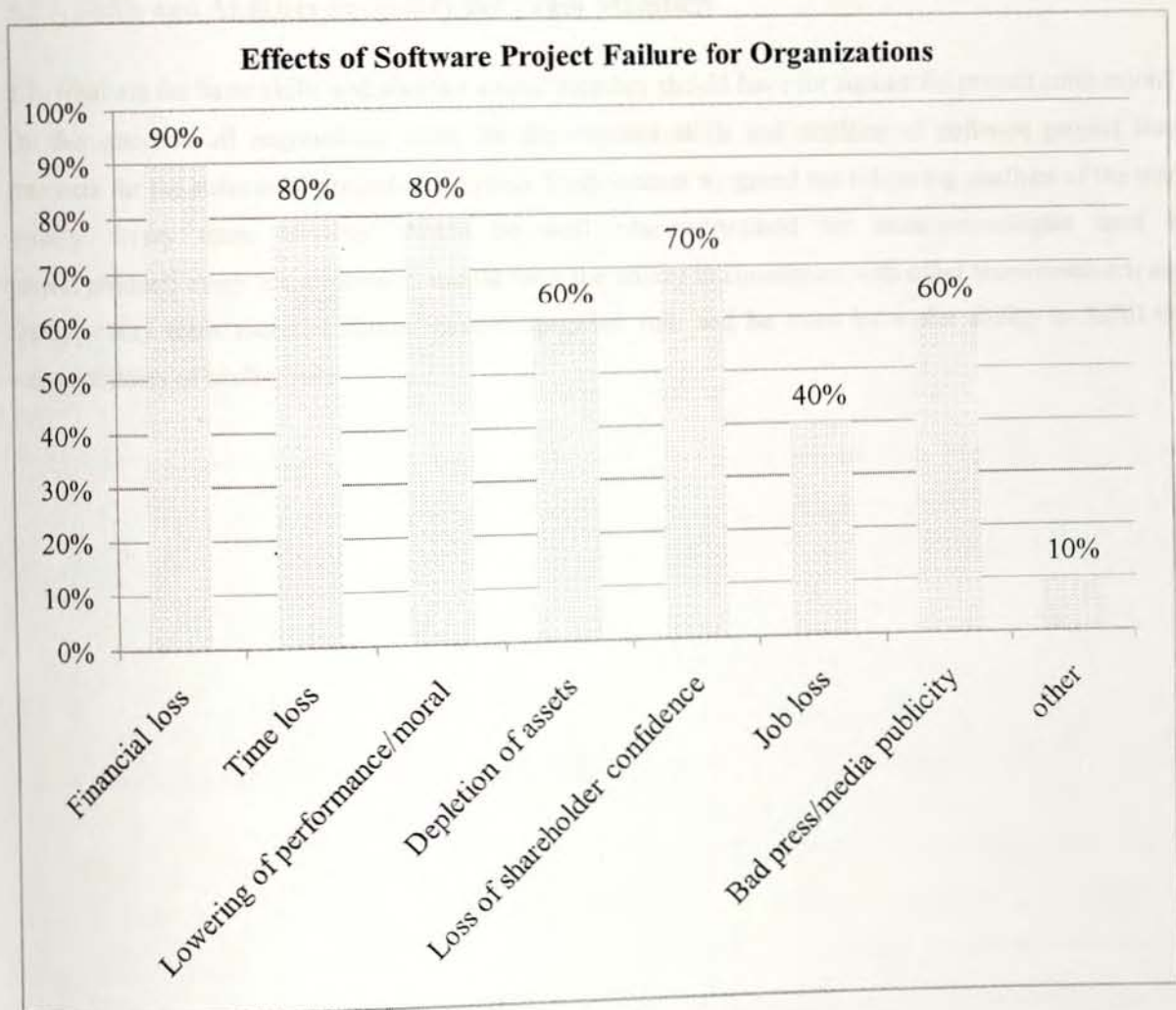


Figure 3.6: Effects of Software Project Failure for Organizations

In Figure 3.6, respondents were asked a general question to show you the possible effects of software project failure for an organization. In case of software failure, organizations could face one or more effects at the same time. Almost all respondents (90%) are agreed that “financial loss” is the main effect of software project failure for most organizations followed by “time loss” and “lowering of performance” accounting for 80% each. The other effects are “depletion of assets” and “bad press/media publicity” 60% each, and “job losses” 40%. These are few main effects of software project failure for the organization but some respondents also mentioned few other reasons like loss of organizations good will.

### 3.2.7. Skills and Abilities necessary for Team Members

2.7. What are the basic skills and abilities a team member should have for successful project completion?

On this question all respondents agree on the required skills and abilities of software project team members for the successful project completion. Respondents weighted the following qualities of the team equally: Every team member should be well educated/trained for tools/technologies used in project/product; every team member should have the ability to coordinate with other team members; and finally, every team member should have a specified role and he must have the ability to fulfill the responsibilities of his/her role.

## 3.2.8. Major Causes of Software Project Failure

2.8. What are the main causes of software project failure according to you?

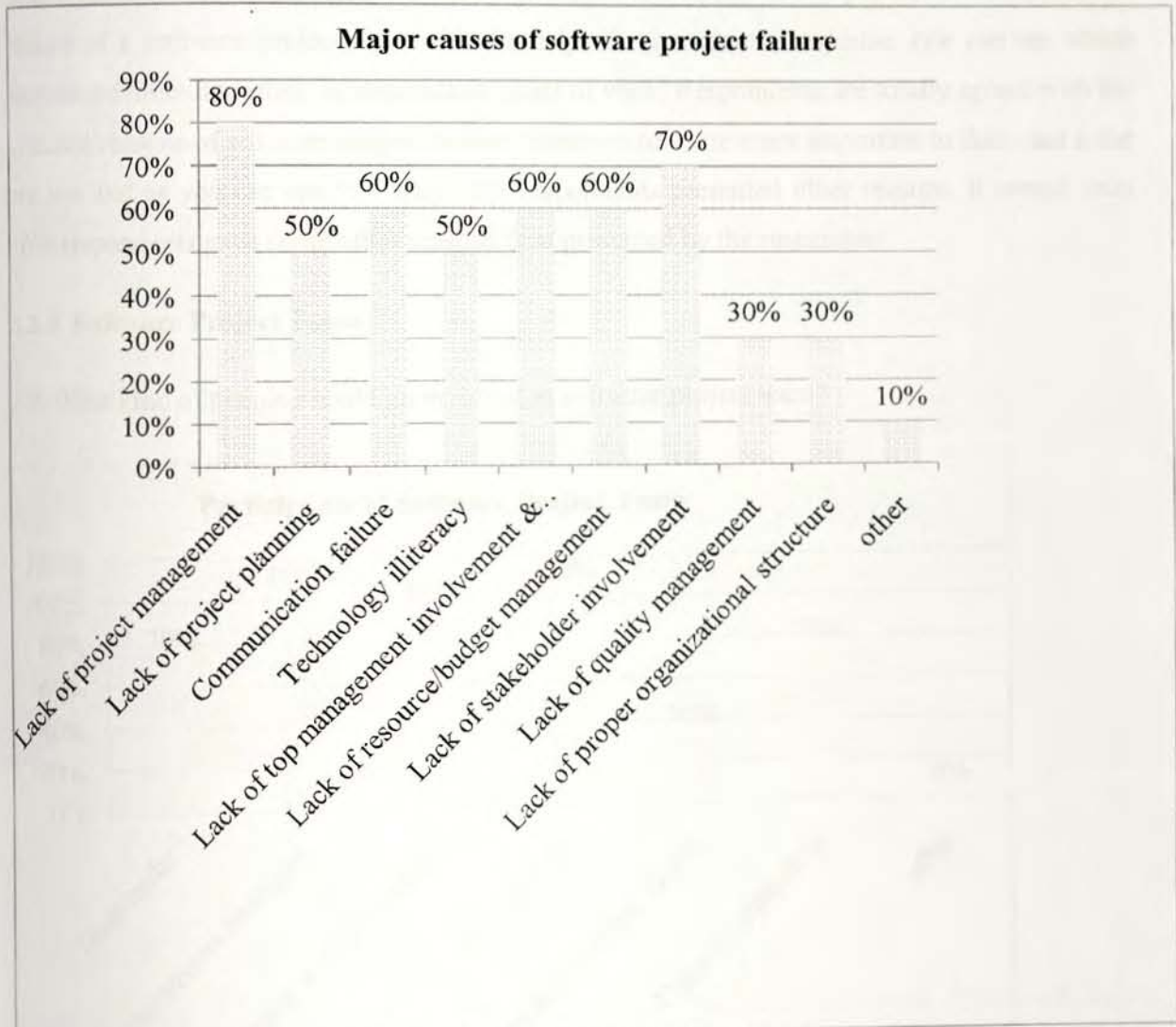


Figure 3.7: Major Causes of Software Project Failure

In Figure 3.7 above, respondents place “lack of project management” as a number one factor for software project failure and which accounts for 80% of the respondents. “Lack of stakeholder involvement” is voted as the second major factor for software project failure by 70% of the respondents. 60% of the respondents vote “lack of top management involvement & support”, “lack of resource/budget” and “communication failure” equally. “Lack of project planning” and

“Technology illiteracy” follows the above failure factors with the ratio of 50% each. 30% selected “lack of quality management” and “Lack of proper organizational structure” as a least major cause of software project failure. Overall, all the factors are important in the success or failure of a software projects but with the help of respondent’s response, one can see which factors are more important in respondents point of view. Respondents are totally agreed with the selected reasons of software project failure. Some reasons are more important to them and some are less but as you can see that only 10% respondents presented other reasons. It means, only 10% respondents gave some other reasons than presented by the researcher.

**3.2.9. Software Project Team**

2.9. What kind of people should be involved in software project team?

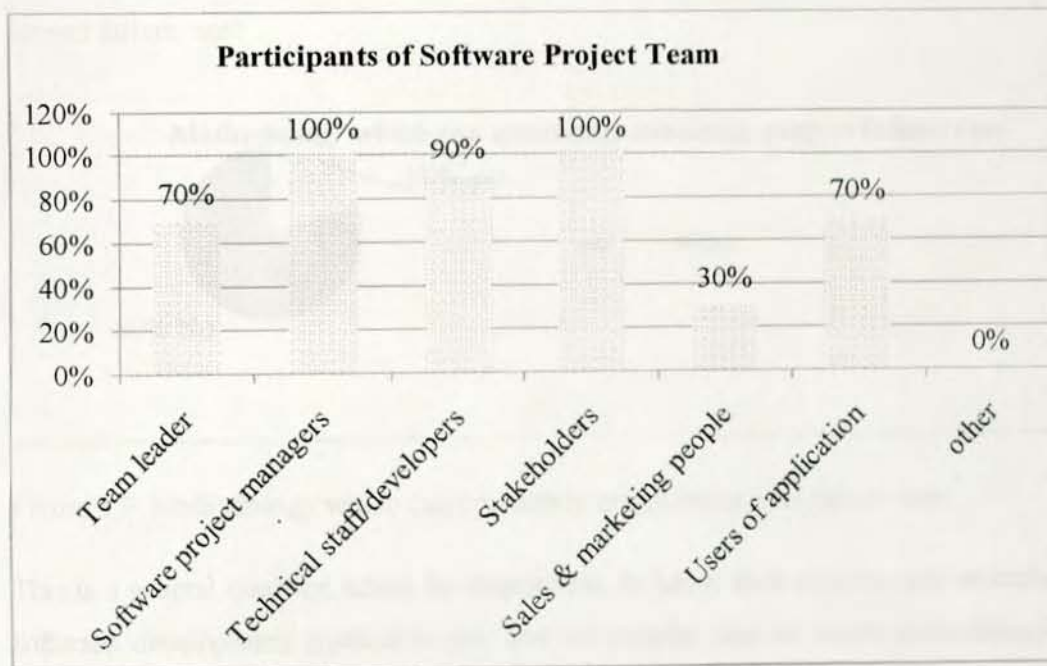


Figure 3.8: Participants of Software Project Team

In the above Figure, we can know the ideal team members for the development of software projects. Let us see which party is more important according to survey respondents. All respondents are agreed that software project managers and stakeholders should be involved in

software project team. Respondents with the ratio of 90% said software project team should involve software project managers, stakeholders and technical staff/developers. On the other hand, 70% of the respondents are believed that in addition to software project managers, stakeholders and technical staff/developers, team leaders and application users should also be included under software project team. The remaining 30% have given their vote to include sales and marketing people in the team of software project.

In conclusion, software project managers, stakeholders and technical staff/developers must be actively involved in software development for its successful completion.

### 3.2.10. Methodology which can commonly minimize project failure rate

2.10. Can one make methodological setup to failing factors which can commonly minimize project failure rate?

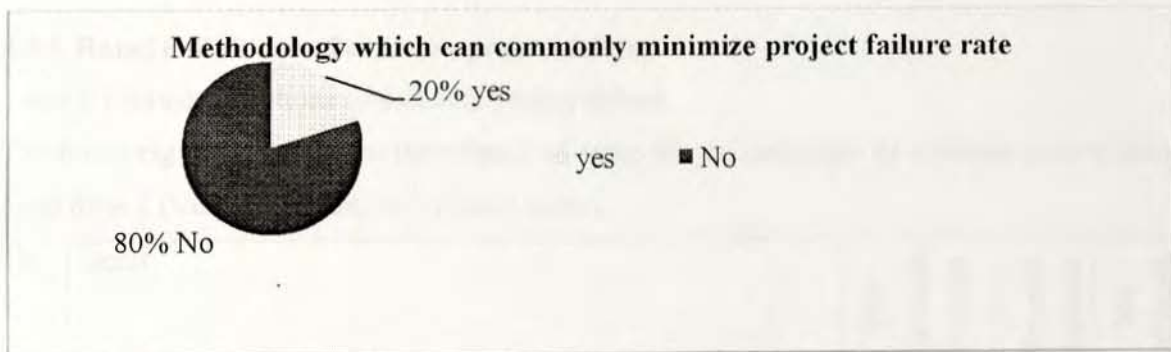


Figure 3.9: Methodology which can commonly minimize project failure rate

This is a general question asked for respondents to know their opinion and understanding about software development methodologies and to suggest one or more methodology which can commonly minimize software project failure rate based on their experience. As we see from Figure 3.9 above, the results found from the survey is totally similar with literatures reviewed in chapter two. Studies have revealed that different methodologies are used in different situations. From this survey also 80% of the respondents are agreed we cannot make methodological setup to failing factors which can commonly minimize project failure rate. What you are surprised here

is that the remaining 20% respondents who say we can make methodological setup to failing factors which can commonly minimize project failure rate are not agreed on one single methodology. Rather, majority of them believes Structured Systems Analysis and Design Method (SSADM) can commonly minimize project failure rate and some others agree Object-Oriented Analysis (OOA) as a good methodology than others for the success of software projects. But it is important to note that respondents were not provided any framework for comparison of system development methodologies rather they have given the opportunity to compare methodologies based on their knowledge about development methods.

### 3.3. Defining Software Project Failure

In third section, there are couples of general questions related to the research study. In this section, the researcher will try to know that how much respondents have knowledge about the software project failure and how they define it.

#### 3.3.1. Rated definitions of software project failure

Table 3.1 Rated definitions of software project failure

The following table shows the importance of items for the definition of software project failure rated from 1 (Very important) to 5 (Don't know).

No_	Items	N	Very important	Important	Not at all important	Not applicabl	Don't know
			%	%	%	%	%
1	Requirements of customers/users were not met by the completed system	10	60	40	0	0	0
2	Users do not found the final system easy to use	10	30	70	0	0	0
3	The final system does not consisted of solid, thoroughly tested code	10	10	60	10	10	10
4	The final system does not work as intended	10	40	60	0	0	0
5	The project was not completed within budget	10	30	70	0	0	0
6	The project was not completed (was cancelled)	10	20	70	0	10	0
7	The project was completed (was not cancelled, but the agreed upon functionality was reduced)	10	10	70	20	0	0

8	The project was not completed on time (within schedule)	10	10	80	10	0	0
9	Software project delivered according to specification but was not on time	10	20	70	10	0	0
10	Software project delivered according to specification but lacks some quality aspects	10	20	70	10	0	0
11	Software project delivered according to specification but was over budgeted	10	30	60	10	0	0
12	Software project was not according to specification but fulfills other requirements like time, quality and budget	10	10	70	20	0	0

Source: Questionnaire survey, 2014

Researchers in the area of software project failure get difficult to find commonly accepted definition for “software project failure.” Some of them define it from customers/users perspective; others from developers’ perspective; and the remaining investigators try to give definition by using different parameters as a standing point. As you can see from the above table, the researcher was tried to include numerous parameters which were assumed most people are agreed to use them when defining software project failure. After the items were identified respondents were given opportunities to rate each based on their perception. Accordingly, “Requirements of customers/users were not met by the completed system” was rated as the first (60% respondents rate it as “very important” and 40% as “important”) parameter followed by “The final system does not work as intended” (40% respondents rate as very important and 60% as important). Besides, the researcher has evaluated the combined results of “Very important” and “important” measures. Based on this result the following four items are given high weight by respondents and they suggest this items should be used as a standing point in defining software project failure so as to reach at the best approximation of the definition: “Requirements of customers/users were not met by the completed system”, “Users do not found the final system easy to use”, “The final system does not work as intended”, and “The project was not completed within budget”. But we need to note that respondents do not want to consider other items as unimportant. To sum up, a system that doesn’t meet customer/user requirements, is not easy to use, is not completed within budget and does not work as intended are considered as a failure.

## 3.4. Process of Developing Software

Respondents perception of software project failure rated from 1 (Very Significant) to 5 (Don't know)

### 3.4.1. The process and the development team

Table 3.2 The process and the development team

No_	Items	N	Very significant	Significant	Not at all significant	Not applicable	Don't know
			%	%	%	%	%
1	The development team is large and low performing.	10	40	50	10	0	0
2	There is no sufficient access to business and/or technical experts outside of the development team	10	20	70	10	0	0
3	The development team is not able to negotiate changes in necessary resources (time, money and/or personnel).	10	0	80	10	10	0
4	Overall, the development team is not sufficiently skilled	10	60	30	10	0	0
5	The development team is not included in the decision making process (estimation, scheduling, etc.).	10	20	80	0	0	0

Source: Questionnaire survey, 2014

In software development the quality of software project team can determine the success or failure rate of software projects. From the above table we can deduce that after all the development team

should be included in making decisions like cost estimation, scheduling, and so on. In addition to this according to the questionnaire survey result the development team should be sufficiently skilled and those technically skillful experts have to be sufficiently accessed.

### 3.4.2. The process and project management

Table 3.3 The process and project management

No_	Items	N	Very significant	Significant	Not at all significant	Not applicable	Don't know
			%	%	%	%	%
1	The project manager does not provide developers with feedback on their work on the project	10	20	70	0	0	10
2	The development team does feel pressured by the schedule/deadlines	10	30	50	10	10	0
3	There is no a defined development methodology (either your organization's own or another)	9	10	60	0	10	10

Source: Questionnaire survey, 2014

It is important to note Figure 3.7 that “lack of project management” is the major cause of software project failure. 90% respondents have believed that for the success of software projects project managers must provide developers with feedback on their work on the project. 80% were also agreed on the development team should feel free on schedules/ deadlines; and the other 70% need to have defined development methodology for the success of software projects.

**3.4.3. The process and customer/users (internal/external)**

Table 3.4 The process and customer/users (internal/external)

No_	Items	N	Very significant	Significant	Not at all significant	Not applicable	Don't know
			%	%	%	%	%
1	Customers/users have not provided the development team with feedback on their work.	1 0	20	60	10	10	0
2	Customers/users have not realistic expectations (including related to time and functionality)	1 0	20	60	0	0	20
3	The customers/users and development team have not a good relationship (cooperative, mutually responsive, etc.)	1 0	20	80	0	0	0

Source: Questionnaire survey, 2014

The above table shows all respondents have said absence of good relationship between customers/users and development team can lead to project failure. The same percentage of 80% responded that customers/users have not provided the development team with feedback on their work and they have not realistic expectations and this leads to project failure.

As remembered from table 3.3, lack of good communication between project manager and the development team (absence of feedback) is a major cause for failure. Here also, lack of good communication between users and the development team is a major cause. Therefore, the project

manager is responsible to make link and smooth the relationship between the project team and software users.

### 3.4.4. The process and project requirements

Table 3.5 The process and project requirements

No_	Items	N	Very significant	Significant	Not at all significant	Not applicable	Don't know
			%	%	%	%	%
1	There is a 'scope creep' (expanding requirements).	10	80	20	0	0	0
2	Requirements are not accepted by the project team as realistic/achievable.	10	40	60	0	0	0
3	Requirements are not clear and not understood by the development team	10	80	20	0	0	0

Source: Questionnaire survey, 2014

Practitioners of software development have agreed on requirement identification is an important step of any software development project. Respondents were also sensitive for this issue. Surprisingly all respondents perceive all measures mentioned above can lead to software project failure. So, if scope creep the case for software project failure, there should have flexible software project plan. Besides, software project objectives are formulated based on the requirements proposed by users. Therefore, to have clear and achievable objectives requirements must be clear and understood by the development team. If requirements are not clear, they are not accepted by the project team as realistic or achievable.

### 3.5. Discussion

In this section, the researcher will try to discuss and correlate questionnaire survey results with the literature review presented in 2nd chapter. Here, the researcher will only discuss the few core questions which are important with respect to this study. This discussion will try to cover all those core areas which are either covered or not above and the purpose of this discussion is to summaries the important areas here. In the first section, the researcher tried to get basic information about the respondents, their organizations, and their role in the projects etc. These things will help us to know about the respondents. In section three, the researcher tried to validate the definition of software project failure presented in literature review. Respondents are agreed and they have given high importance to delivering a system that doesn't meet customer/user requirements. It is second highest ratio of respondents those agreed that a software project is a failure if it is not easy to use. Others are those, who gave almost equal importance to the time, quality and budget for considering a project failure. The consensus among the respondents who participated in this study indicated that delivering a system that doesn't meet customer/user requirements, is not easy to use, is not completed within budget and does not work as intended are important aspects for them to consider a project to be a failure. So, final conclusion from section three is that a system that doesn't meet customer/user requirements, is not easy to use, is not completed within budget and does not work as intended are considered as a failure.

In section two questions 8 is basically made to validate all major causes of software project failure presented in chapter two. In the result, highest response was gotten in favor of lack of project management, lack of stakeholder involvement, lack of top management involvement and support, lack of resource/budget, communication failure, lack of project planning, and technology illiteracy are the main factors that can lead to software project failure. On above mentioned causes respondents were agreed that these are the most important and common factors. Few respondents mentioned some other reasons of software failure which were different than presented in the questionnaire. But this percentage was only 10%. It shows that all the respondents were almost agreed with the causes of software failure mentioned by me in literature

review chapter. In question 11, respondents were asked about the parties that should be involved in software project development process. In answer of this, they were agreed that software project managers, stakeholders, technical staff/developers, team leaders, application users and sales and marketing people should be the part of this team. A high ratio of respondents mentioned that software project managers, stakeholders and technical staff/developers are very crucial parties for this process.

In conclusion, it was a useful survey which helped the researcher to know the point of view of experts and software users on the software project failure and other concerned issues in our country context. A strong point about the survey is that majority of respondents involved in this survey were experienced people so that they can know the local software development environment. Most of them are currently involved in managing and developing software projects in different organizations. Thus, it can be approved from the critical review literatures and the survey result that lack of proper project management challenges all local software projects.

## CHAPTER FOUR

### Conclusions, Recommendations and Suggestions for Future Researches

#### 4.1. Conclusions

The consensus among the respondents who participated in this study indicated that delivering a system that doesn't meet customer/user requirements, is not easy to use, is not completed within budget and does not work as intended are important aspects for them to consider a project to be a failure. So, A system that doesn't meet customer/user requirements, is not easy to use, is not completed within budget and does not work as intended are considered as a failure. The findings from the questionnaire survey indicate that lack of project management, lack of stakeholder involvement, lack of top management involvement and support, lack of resource/budget, communication failure, lack of project planning, and technology illiteracy are the main factors that can lead to software project failure. Even though the above mentioned factors are considered as main factors, they are not the only factors for project failure. The project team must be involved in activities like cost estimation, scheduling activities and making deadlines for the project and in other decision making processes. The project team needs also to have good communication with project manager and users. Having flexible project plan with the expectation of requirement change is also an important finding from the survey result. Unclear requirement specification also leads to software project failure. The severity of the factors may change with technological, economical, societal, and other changes. That is why it is difficult to outline comprehensive failure factors. Each of the failure factors cannot also be controlled by setting and following specific system development methodologies, because by their nature system development methodologies are adaptive.

### 4.2. Recommendations and Suggestions for Future Researches

Based on the findings and conclusions, the researcher forwarded the following recommendations and suggestions:

- It is common that requirement specifications are changed during the software project development process. If project team is forced for any midway changes in large and complex projects, it demands to revise the whole project plan. In case of any midway changes in projects will not serve with "Linear Thinking". It means a project should be revised in sense of updating the project deadline, budget, staff and effort required for completing it successfully. There could be some other factors needed to revise depending on the project.
- Project "goals and objectives" should be realistic and attainable; otherwise it is difficult to achieve them. We should have a clear idea about the status of the project on every stage of its completion. It is only possible by true comparison between achieved and forecasted performance. Realistic goals are easy to achieve and a motivational factor for the project development team.
- Successful completion and implementation of a software project requires the collaboration and communication among different participants like Stakeholders, Management, Users, and Development Team etc. Every "team participant" has to play his role and responsibilities for making a project successful. We need to include all required participants in the process for successful completion of a project.

## References

1. Benzzine, A. (2002). System Development Methodologies a framework for comparison. MSIS 488 - Information Systems Analysis.
2. Bern, A., Nikula, U., Jaya, S., Pasi, A., and Smolander, K. (2008). Contextual Factors Affecting the Software Development Process – An Initial View.
3. Boehm B. W. (1991). Software Risk Management: Principles and Practices, *IEEE Software*, 8, 1, , pp.32-41.
4. Charette, R. N. (2005). Why software fails? Posted 2 Sep | 3:18 GMT.
5. Davies, T. C. (2002). The real success factors on projects, *International Journal of Project Management*, 2, 185-190.
6. Drew, J. P., and June, M. V. (2006). Software project managers and project success: An exploratory study. *The Journal of Systems and Software*, page 1541–1551.
7. Ebrahim, N. S. A. (2013). Identifying the Software Failure Mechanisms Using Data Mining Techniques. *International Journal of Soft Computing and Engineering (IJSCE)* ISSN: 2231-2307, Volume-3, Issue-4.
8. George, W. R. (1988). *Information Systems for Managers*, 3rd ed. West publishing company, pp. 01-429.
9. Grossman, I. (2003). Why so many IT Projects fail and how to find Success, *Financial Executive*, ABI/INFORM Global.
10. Hoffer, J. A., George, J. F., and Valacich, J. S. (1999). *Modern Systems Analysis & Design*, 2nd ed. Addison Wesley Longman, Inc. 24 p.
11. Jones, C. (1998). Project Management Tools and Software Failures and Successes. *The Journal of Defense Software Engineering Software Productivity Research*, Inc.
12. Jones, C. (2004). Software Project Management Practices: Failure Versus Success. *The Journal of Defense Software Engineering, Software Productivity Research LLC*.
13. Jones, C. (2006). Social and Technical Reasons for Software Project Failures. *The Journal of Defense Software Engineering, Software Productivity Research LLC*.
14. Jones, C. (2007). Software engineering: The state of the art in 2007. *Software Productivity Research LLC*, Version 11 – November 21.

15. Jones, C. (2008). Measuring defect potentials and defect removal efficiency. Provided by Rex Black Consulting Services, Version 4.1– March 1.
16. Kappelman, L.A., McKeeman, R. and Zhang, L. (2006). Early warning signs of it project failure: The dominant dozen. IT Project Management.
17. Linberg, K. R. (1999). Software developer perceptions about software project failure: a case study. *The Journal of Systems and Software* 49 177-192.
18. Linda, W., Mark, K., and Arun, R. (2004). How Software Project Risk Affects Project Performance: An Investigation of the Dimensions of Risk and an exploratory model. *Decision Sciences*, Volume 35, Number 2 Spring 2004.
19. May, L. J. (1998). Major Causes of Software Project Failures. *The Journal of Defense Software Engineering*.
20. Mohan, K., Ahlemann, F., and Kügler, M. (2013). Understanding the Nature of Use Regarding System Development and Management Methodologies - Using Psychoanalysis to Understand the Influences of Methodology Attributes. 46th Hawaii International Conference on System Sciences.
21. Parker, M. M. (1995). *Strategic Transformation & Information Technology: Paradigms for Performing while Transforming*, USA: Prentice Hall, Inc. pp. 03-508.
22. Procaccino J. D., Verner J. M. (2006). Software project managers and project success: An exploratory study. *The Journal of Systems and Software* 79 (2006) 1541–1551.
23. Sauer, C. & Cuthbertson, C. (2003). The state of IT project management in the UK. Templeton College, Oxford University.
24. Tewdros, S. A., and Bhaskar, R.M. (2011). E-Business: Application of software and technology in selected Ethiopian Banks: Issues and challenges. *International Journal of Computer Science Issues*, Vol. 8, Issue 6, No 1.
25. The Standish Group, (1995). CHAOS; The Standish Group International.
26. Underwood, R. M. (2001). Factors Affecting the Success Rate of Software Development Projects.
27. Van Aken, T. (1996). De weg naar project success: Eerder via werkstijl dan instrumenten, *De Tijdstroom*, p 411 quoted in Westerveld, E. The project excellence model: linking success criteria and critical success factors, *International Journal of*

*Project Management*, 21, 411-418.

28. Vasudevan, S. (2005). Defect Prevention Techniques, and Practices, *Proceedings of Fifth Annual International Software Testing Conference*, 2005.
29. Verner, J., Sampson, J., and Cerpa, N. (2009). What factors lead to software project failure?
30. Walid, Al-A., Khalid, Al-F., Khalid, K., Khalid, A., Saleem A., and Hani Abu-S. (2009). A Taxonomy of an IT Project Failure: Root Causes. Vol. 5 No. 1 2009.
31. wood-happer, T. & McManus, J. (2003). Information systems project management: the price of failure, *management services*, ABI/INFORM Global.
32. Zeeshan, A. & Gerhard, D. (2007). Contributions of PDM Systems in Organizational Technical Data Management, pp. 12-13.
33. <http://www.mereja.com/forum/memberlist.php?>, 04 Oct 2013, 17:46.

Appendixes

APPENDIXES

APPENDIX I- COVER LETTER

Dear Respondent,

First of all I would like to extend my heartfelt gratitude and respect to you for administering this questionnaire honestly and responsibly. The questionnaire is designed to collect the necessary information to undertake a research on the topic "Identification of factors that lead to software project failure" in partial fulfillment of the requirements of the degree of Master of Information Science at Addis Ababa University. It will hardly take a few minutes please try to complete it as soon as possible.

Finally, I would like to assure you that your responses will be kept confidential and no question I will be extremely grateful if you could return it to me as soon as possible.

Yours faithfully,

Mulualem Tesfaye MSc

Research Assistant

School of Information Science

Addis Ababa University

Adis ababa

# Appendixes

APPENDIX I – COVER LETTER

<Dear Respondents>

First of all I would like to forward my heartfelt gratitude and respect to you for administering this questionnaire honestly and responsibly. The questionnaire is designed to collect the necessary information to undertake a research on the topic “Identification of factors that lead to software project failure” in partial fulfillment of the requirements of the degree of Master of Information Science at Addis Ababa University. It will hardly take a few minutes; please try to complete as much questions as you can.

Finally, I would like to assure you that all the information you provide in this questionnaire will be kept confidential and will exclusively be used for research purpose. Please feel free to ask any question. I am available at [mtemeche@gmail.com](mailto:mtemeche@gmail.com).

I will be extremely thankful to you for your extended cooperation.

Yours faithfully,

**Mullualem Temeche Molla**

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

School of Information Science

Addis Ababa University

Ethiopia

## APPENDIX II – QUESTIONNAIRE FOR SOFTWARE COMPANIES

### Section 1: Respondents Background

1.1 Gender:

- Male  Female

1.2 Age range:

- 16–19 years  45–54 years  
 20–24 years  55–64 years  
 25–34 years  65+  
 35–44 years

1.3 Highest completed level of education

- Primary school  Bachelor degree  
 High school  Graduate degree  
 Some college, no degree  Doctorate (PhD)

1.4 Indicate the approximate number of year (full-time equivalent) that you have worked in each of the following jobs (if zero, leave blank):

- |   |  |
|---|--|
| <input type="text"/> Business analyst       | <input type="text"/> Project manager/leader    |
| <input type="text"/> Database administrator | <input type="text"/> Team leader               |
| <input type="text"/> Programming analyst    | <input type="text"/> Database developer        |
| <input type="text"/> Systems analyst        | <input type="text"/> Programmer                |
| <input type="text"/> Change control officer | <input type="text"/> Senior manager            |
| <input type="text"/> Network engineer       | <input type="text"/> Other (specify job) _____ |

### Section 2: Different Perspectives of Software Project Failure:

2.1 What kind(s) of software development services does your organization provide?

- A. Software projects with the specified time span (Short term)  
B. Software products developed over the years (Long term)  
C. Both  
D. Please specify if other \_\_\_\_\_

2.2 Have you ever participated in the development of software project?

- A. Yes, I have participated in software development as a Technical person  
B. Yes, I have participated in software development as a Non Technical person  
C. Both

D. No, I have not participated in software development but I know most of the procedure through literature survey

2.3 If you have participated in software development, what was your role in that team?

- A. Team leader
- B. Project manager
- C. Software designer/developer
- D. Stake holder (investor)
- E. Please specify if other \_\_\_\_\_

2.4 In how many software projects have you been involved?

- A. 5 projects
- B. 5 to 10 projects
- C. 10 to 15 projects
- D. 15 to 20 projects
- E. 20 or more projects
- F. Other (0 to 4) projects

2.5 What was the ratio of successful versus failed software projects in which you were involved?

(Divide 100% in A & B)

- A. Successful \_\_\_\_\_ %
- B. Failure \_\_\_\_\_ %

2.6 What can be the effects of software failure for the organization as a whole? (Please specify all those apply)

- A. Financial loss
- B. Time loss
- C. Lowering of performance/moral
- D. Depletion of assets
- E. Loss of shareholder confidence
- F. Job loss
- G. Bad press/media publicity
- H. Please specify if other \_\_\_\_\_

2.7 What are the basic skills and abilities a team member should have for successful project completion? (Please specify all those that apply)

- A. Every Team member should be well educated/trained for tools/technologies used in project/product
- B. Every team member should have the ability to coordinate with other team members
- C. Every Team member should have a specified role and he must have the ability to fulfill the responsibilities of his/her role
- D. All above
- E. Please specify if other \_\_\_\_\_

2.8 What are the main causes of software project failure according to you? (Please specify all those apply)

## Why Software Projects Fail? | 2014

---

- A. Lack of project management
- B. Lack of project planning
- C. Communication failure
- D. Technology illiteracy
- E. Lack of top management involvement & support
- F. Lack of resource/budget management
- G. Lack of stakeholder involvement
- H. Lack of quality management
- I. Lack of proper organizational structure
- J. Please specify if other \_\_\_\_\_

2.9 What kind of people should be involved in software project team? (Please specify all those apply)

- A. Team leader
- B. Software project managers
- C. Technical staff/developers
- D. Stakeholders
- E. Sales & marketing people
- F. Users of application
- G. Please specify if other \_\_\_\_\_

2.10 Can one make methodological setup to failing factors which can commonly minimize project failure rate?

- A. Yes
- B. No

2.11 If your answer is yes for question number 2.10, what is that methodology?

- A. Structured Analysis, Design & Implementation of Information Systems (STRADIS)
- B. Yourdon Systems Method (YSM)
- C. Jackson Systems Development (JSD)
- D. Information Engineering (IE)
- E. Object-Oriented Analysis (OOA)
- F. Rational Unified Process (RUP)
- G. Dynamic Systems Development Method (DSDM)
- H. Extreme Programming (XP)
- I. Effective Technical & Human Implementation of Computer-Based Systems (ETHICS)
- J. Soft Systems Methodology (SSM)
- K. Projects in Controlled Environment (PRINCE)
- L. Structured Systems Analysis & Design Method (SSADM)
- M. Please specify if other \_\_\_\_\_

## Section 3: Defining Project Failure

Each of the statements in this section relates to a specific aspect of the final product or result of developing software.

For each of the following statements, rate how important you believe each is, in general, to your definition of software project failure. Each question begins with the phrase, **“It is important to your definition of project failure that . . .”**

Please rate the factors on a scale of 1 to 5.

1	2	3	4	5
Very important	Important	Not at all important	Not applicable	Don't know

### Defining Failure—Final System/Project:

- 3.1 Requirements of customers/users were not met by the completed system.
- 3.2 Users do not found the final system easy to use.
- 3.3 The final system does not consisted of solid, thoroughly tested code.
- 3.4 The final system does not work as intended.
- 3.5 The project was not completed within budget.
- 3.6 The project was not completed (was cancelled).
- 3.7 The project was completed (was not cancelled, but the agreed upon functionality was reduced).
- 3.8 The project was not completed on time (within schedule).
- 3.9 Software project delivered according to specification but was not on time.
- 3.10 Software project delivered according to specification but lacks some quality aspects
- 3.11 Software project delivered according to specification but was over budgeted.
- 3.12 Software project was not according to specification but fulfills other requirements like time, quality and budget
- 3.13 If necessary, explain below your opinion on any of the statements in this section \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

3.14 Indicate any additional aspects of project development that are important to your definition of project failure \_\_\_\_\_

**Section 4: Process of Developing Software**

Each of the statements in this section relates to a specific aspect of the process of developing software. Please rate how significant you believe each is in its contribution to what you consider a 'failed' software project. Each question begins with the phrase, **“It is important to your perception of project failure that. . .”**

Please rate the factors on a scale of 1 to 5.

1	2	3	4	5
Very Significant	Significant	Not at all significant	Not applicable	Don't know

**The Process & Your Team:**

- \_\_\_ 4.1 The development team is large and low performing.
- \_\_\_ 4.2 There is no sufficient access to business and/or technical experts outside of the development team.
- \_\_\_ 4.3 The development team is not able to negotiate changes in necessary resources (time, money and/or personnel).
- \_\_\_ 4.4 Overall, the development team is not sufficiently skilled.
- \_\_\_ 4.5 The development team is not included in the decision making process (estimation, scheduling, etc.).

**The Process & Project Management:**

- \_\_\_ 4.6 The project manager do not provide developers with feedback on their work on the project.
- \_\_\_ 4.7 The development team does feel pressured by the schedule/deadlines.
- \_\_\_ 4.8 There is no a defined development methodology (either your organization's own or another).

**The Process & Customer/Users (Internal/External):**

- \_\_\_ 4.9 Customers/users have not provided the development team with feedback on their work.

\_\_\_ 4.10 Customers/users have not realistic expectations (including related to time and functionality).

\_\_\_ 4.11 The customers/users and development team have not a good relationship (cooperative, mutually responsive, etc.)

### **The Process & Project Requirements:**

\_\_\_ 4.12 There is a 'scope creep' (expanding requirements).

\_\_\_ 4.13 Requirements are not accepted by the project team as realistic/achievable.

\_\_\_ 4.14 Requirements are not clear and not understood by the development team.

4.15 If necessary, explain below your opinion on any of the statements in this section. \_\_\_\_\_  
\_\_\_\_\_

4.16 Indicate any additional aspects of the process of developing software that contribute to project failure. \_\_\_\_\_  
\_\_\_\_\_

4.17 Do projects fail because of a single overwhelming factor or is it true that they fail only when there is a combination of factors? \_\_\_\_\_

4.18 How many factors must be in conjunction before a project becomes a failure? \_\_\_\_\_  
\_\_\_\_\_

4.19 Are there any common patterns in those factors? \_\_\_\_\_  
\_\_\_\_\_

Thank you for your participation

## APPENDIX III – QUESTIONNAIRE FOR SOFTWARE USERS

### Section 1: Respondents Background

#### 1.1 Gender

- Male  Female

#### 1.2 Age range

- 16–19 years  45–54 years s  
 20–24 years  55–64 years  
 25–34 years  65+  
 35–44 years

#### 1.3 Highest completed level of education

- Primary school  Bachelor degree  
 High school  Graduate degree  
 Some college, no degree  Doctorate (PhD)

#### 1.4 Indicate the approximate number of year (full-time equivalent) that you have worked in each of the following jobs (if zero, leave blank)

- |                            |                               |
|----------------------------|-------------------------------|
| ___ Business analyst       | ___ Project manager/leader    |
| ___ Database administrator | ___ Team leader               |
| ___ Programming analyst    | ___ Database developer        |
| ___ Systems analyst        | ___ Programmer                |
| ___ Change control officer | ___ Senior manager            |
| ___ Network engineer       | ___ Other (specify job) _____ |

### Section 2: Different Perspectives of Software Project Failure:

#### 2.1 What kind(s) of software development services your organization was provided?

- A. Software projects with the specified time span (Short term)  
B. Software products developed over the years (Long term)  
C. Both  
D. Please specify if other \_\_\_\_\_

#### 2.2 Have you ever participated in the development of software project?

- A. Yes, I have participated in software development as a Technical person  
B. Yes, I have participated in software development as a Non Technical person  
C. Both

D. No, I have not participated in software development but I know most of the procedure through literature survey

2.3 If you have participated in software development, what was your role in that team?

- A. Team leader
- B. Project manager
- C. Software designer/developer
- D. Stake holder (investor)
- E. Please specify if other \_\_\_\_\_

2.4 In how many software projects have you been involved?

- A. 5 projects
- B. 5 to 10 projects
- C. 10 to 15 projects
- D. 15 to 20 projects
- E. 20 or more projects
- F. Other (0 to 4) projects

2.5 What was the ratio of successful versus failed software projects in which you were involved? (Divide 100% in A & B)

- A. Successful \_\_\_\_\_%
- B. Failure \_\_\_\_\_%

2.6 What can be the effects of software failure for the organization as a whole? (Please specify all those apply)

- A. Financial loss
- B. Time loss
- C. Lowering of performance/moral
- D. Depletion of assets
- E. Loss of shareholder confidence
- F. Job losses
- G. Bad press/media publicity
- H. Please specify if other \_\_\_\_\_

2.7 What are the basic skills and abilities a team member should have for successful project completion? (Please specify all those that apply)

- A. Every Team member should be well educated/trained for tools/technologies used in project/product
- B. Every team member should have the ability to coordinate with other team members
- C. Every Team member should have a specified role and he must have the ability to fulfill the responsibilities of his/her role
- D. All above
- E. Please specify if other \_\_\_\_\_

2.8 What are the main causes of software project failure according to you? (Please specify three or all those apply)

- A. Lack of project management
- B. Lack of project planning
- C. Communication failure
- D. Technology illiteracy
- E. Lack of resource
- F. Lack of top management involvement & support
- G. Lack of stakeholder involvement
- H. Lack of quality management
- I. Lack of proper organizational structure
- J. Please specify if other \_\_\_\_\_

2.9 What kind of people should be involved in software project team? (Please specify all those that apply)

- A. Team leader
- B. Software project managers
- C. Technical staff/developers
- D. Stakeholders
- E. Sales & marketing people
- F. Users of application
- G. Please specify if other \_\_\_\_\_

2.10 Can one make methodological setup to failing factors which can commonly minimize project failure rate?

- A. Yes
- B. No

2.11 If your answer is yes for question number 2.10, what is that methodology?

- A. Structured Analysis, Design & Implementation of Information Systems (STRADIS)
- B. Yourdon Systems Method (YSM)
- C. Jackson Systems Development (JSD)
- D. Structured Systems Analysis & Design Method (SSADM)
- E. Information Engineering (IE)
- F. Object-Oriented Analysis (OOA)
- G. Rational Unified Process (RUP)
- H. Dynamic Systems Development Method (DSDM)
- I. Extreme Programming (XP)
- J. Effective Technical & Human Implementation of Computer-Based Systems (ETHICS)
- K. Soft Systems Methodology (SSM)
- L. Projects in Controlled Environment (PRINCE)
- L. Please specify if other \_\_\_\_\_

## Section 3: Defining Project Failure

Each of the statements in this section relates to a specific aspect of the final product or result of developing software.

For each of the following statements, rate how important you believe each is, in general, to your definition of software project failure. Each question begins with the phrase, **“It is important to your definition of project failure that. . .”**

Please rate the factors on a scale of 1 to 5.

1	2	3	4	5
Very important	Important	Not at all important	Not applicable	Don't know

### Defining Failure—Final System/Project:

- 3.1 Requirements of customers/users were not met by the completed system.
- 3.2 Users do not found the final system easy to use.
- 3.3 The final system does not consisted of solid, thoroughly tested code.
- 3.4 The final system does not work as intended.
- 3.5 The project was not completed within budget.
- 3.6 The project was not completed (was cancelled).
- 3.7 The project was completed (was not cancelled, but the agreed upon functionality was reduced).
- 3.8 The project was not completed on time (within schedule).
- 3.9 Software project delivered according to specification but was not on time.
- 3.10 Software project delivered according to specification but lacks some quality aspects
- 3.11 Software project delivered according to specification but was over budgeted.
- 3.12 Software project was not according to specification but fulfills other requirements like time, quality and budget
- 3.13 If necessary, explain below your opinion on any of the statements in this section \_\_\_\_\_  
\_\_\_\_\_
- 3.14 Indicate any additional aspects of project development that are important to your definition

of project failure \_\_\_\_\_

### Section 4: Process of Developing Software

Each of the statements in this section relates to a specific aspect of the process of developing software. Please rate how significant you believe each is in its contribution to what you consider a 'failed' software project. Each question begins with the phrase, **“It is important to your perception of project failure that. . .”**

Please rate the factors on a scale of 1 to 5.

1	2	3	4	5
Very Significant	Significant	Not at all significant	Not applicable	Don't know

#### The Process & Your Team:

- \_\_\_ 4.1 The development team is large and low performing.
- \_\_\_ 4.2 There is no sufficient access to business and/or technical experts outside of the development team.
- \_\_\_ 4.3 The development team is not able to negotiate changes in necessary resources (time, money and/or personnel).
- \_\_\_ 4.4 Overall, the development team is not sufficiently skilled.
- \_\_\_ 4.5 The development team is not included in the decision making process (estimation, scheduling, etc.).

#### The Process & Project Management:

- \_\_\_ 4.6 The project manager do not provide developers with feedback on their work on the project.
- \_\_\_ 4.7 The development team does feel pressured by the schedule/deadlines.
- \_\_\_ 4.8 There is no a defined development methodology (either your organization's own or another).

#### The Process & (Internal/External) Customer/Users:

- \_\_\_ 4.9 Customers/users have not provided the development team with feedback on their work.
- \_\_\_ 4.10 Customers/users have not realistic expectations (including related to time and

functionality).

\_\_\_ 4.11 The customers/users and development team have not a good relationship (cooperative, mutually responsive, etc.).

### **The Process & Project Requirements:**

\_\_\_ 4.12 There is a 'scope creep' (expanding requirements).

\_\_\_ 4.13 Requirements are not accepted by the project team as realistic/achievable.

\_\_\_ 4.14 Requirements are not clear and not understood by the development team.

4.15 If necessary, explain below your opinion on any of the statements in this section. \_\_\_\_\_  
\_\_\_\_\_

4.16 Indicate any additional aspects of the process of developing software that contribute to project failure. \_\_\_\_\_  
\_\_\_\_\_

4.17 Do projects fail because of a single overwhelming factor or is it true that they fail only when there is a combination of factors? \_\_\_\_\_

4.18 How many factors must be in conjunction before a project becomes a failure? \_\_\_\_\_  
\_\_\_\_\_

4.19 Are there any common patterns in those factors? \_\_\_\_\_

Thank you for your participation