STATUS OF THESIS

Title of thesis:	A PROPOSED FRAMEWORK FOR TAILORING AGILE-BASED SOFTWARE DEVELOPMENT PROCESSES FOR SMALL AND MEDIUM SIZED COMPANIES

I ______ REHAN AKBAR ______

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

☐ Confidential

√ Non-confidential

If this thesis is confidential, please state the reason:

____

____

____

The contents of the thesis will remain confidential for ___________ years.

Remarks on disclosure:

____

____

Endorsed by

______________________	______________________
Signature of Author	Signature of Supervisor
Permanent address:	Name of Supervisor
3-W-1 Madina Town	Assoc. Prof. Dr. Mohd. Fadzil Bin
Faisalabad, Pakistan	Hassan
Date :______________	Date :______________

UNIVERSITI TEKNOLOGI PETRONAS

A PROPOSED FRAMEWORK FOR TAILORING AGILE-BASED SOFTWARE DEVELOPMENT PROCESSES FOR SMALL AND MEDIUM SIZED COMPANIES

by

REHAN AKBAR

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fullfilment of the requirements for the degree stated.

Signature: ______________________________

Main Supervisor: Assoc. Prof. Dr. Mohd. Fadzil Bin Hassan

Signature: ______________________________

Co-Supervisor: Assoc. Prof. Dr. Azrai Abdullah

Signature: ______________________________

Head of Department: Dr. Jafreezal Bin Jafar

Date: ______________________________

A PROPOSED FRAMEWORK FOR TAILORING AGILE-BASED SOFTWARE DEVELOPMENT PROCESSES FOR SMALL AND MEDIUM SIZED COMPANIES

by

REHAN AKBAR

A Thesis

Submitted to the Postgraduate Studies Programme

as a Requirement for the Degree of

DOCTOR OF PHILOSOPHY

INFORMATION TECHNOLOGY

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR

PERAK, MALAYSIA

APRIL 2013

DECLARATION OF THESIS

Title of thesis	A PROPOSED FRAMEWORK FOR TAILORING AGILE-BASED SOFTWARE DEVELOPMENT PROCESSES FOR SMALL AND MEDIUM SIZED COMPANIES

I, REHAN AKBAR

hereby declare that the thesis is based on my original work except for quotations and citations, which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

________________________	________________________
Signature of Author	Signature of Supervisor
Permanent address:	Assoc. Prof. Dr. Mohd. Fadzil
3-W-1 Madina Town	Bin Hassan
Faisalabad, Pakistan.	
Date : ____________________	Date : ____________________

DEDICATION

To my father Dr. Shamshad Akbar, mother Rafia Khanam and especially to my beloved wife Shumaila Rehan for their prayers, continuous support and encouragement.

With Love to my two princesses Hira Akbar and Hadia Akbar

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the guidance, support and cooperation of all those persons who in one way or another contributed their assistance, helped me and provided their guidance in the completion of this study.

First of all, I am truly thankful to Almighty ALLAH for His unlimited favors, support and removing all obstacles in achieving this objective and making me able to complete this study.

I am sincerely thankful to my worthy supervisor Dr. Mohd Fadzil Hassan for his valuable suggestions, unselfish support and guidance throughout my study without which I was not able to complete it.

I am heartily thankful to my co-supervisor Dr. Azrai Abdullah for his guidance, input and encouragement during my study.

I am truly indebted and thankful to CambridgeDocs (Pvt.) Ltd., for extending their cooperation and collaboration in completion of my research work.

Besides, I would like to thank to my parents, my wife and daughters for their moral support, encouragement and prayers for successful completion of this dissertation.

In the last I would like to thank all my friends who helped and supported me throughout my studies in each and every respect.

ABSTRACT

High risks are involved as well as a large number of resources are required to construct the software development processes from scratch. Most of the software development companies follow ad-hoc approaches in informal ways to tailor an existing software development process according to their requirements. Instead of devising new tailoring strategies, these approaches describe and compare the similar tailoring operations at very superficial level and mainly focus on the large sized software development companies.

In this regard, very limited attention has been paid on the software development process tailoring in small and medium sized software development companies according to the agile based methodologies. Lightweight agile based methodologies and software process tailoring are preferred practices in these companies. Such companies due to resources constraints are unable to create new processes or follow traditional heavyweight approaches of software development. Despite the importance of process tailoring, very limited research has been carried out in this area which arises the need of a formal and systematic process tailoring approach for these small and medium sized software development companies.

This research addresses this issue and presents a framework for software development process tailoring in such small and medium sized software development companies. The present study is based on the project and project's client factors such that it analyzes the client's perspective, project's behavior and various states of the projects, and proposes the process tailoring operations, activities and strategies. Both qualitative and quantitative research methodologies have been followed to validate and complete this study. The results show that present study provides a lightweight approach to tailor the software development processes and activities in small and medium sized software development companies that apply agile based methodologies.

ABSTRAK

Risiko yang tinggi, begitu juga dengan sumber yang banyak akan diperlukan untuk menstrukturkan proses-proses pembangunan perisian daripada awal. Kebanyakkan syarikat pembangunan perisian menggunakan pendekatan informal yang segera untuk selari dengan proses pembangunan perisian sedia ada dan mengikut keperluan mereka. Dalam pada memulakan strategi yang baru, pendekatan segera ini memperlihatkan operasi yang hendak disamakan hanya pada peringkat luaran sahaja dan fokusnya hanyalah syarikat pembangunan perisian yang besar.

Oleh itu, didapati hanya sedikit perhatian diberikan kepada proses penyesuaian pembangunan perisian dalam metodologi berasaskan perubahan ini oleh syarikat pembangunan perisian yang bersaiz kecil dan sederhana. Metodologi berasaskan perubahan yang ringkas dan penyesuaian proses perisian selalunya menjadi pilihan syarikat jenis ini. Dek kerana kekangan sumber, syarikat kecil dan sederhana ini tidak dapat membina proses-proses baru pembangunan sesuatu perisian, walaupun mengikut pendekatan tradisional yang kompleks. Disamping pentingnya penyesuaian proses, tidak banyak kajian yang dijalankan dalam bidang ini bagi meningkatkan keperluan kepada penyesuaian proses yang sistematik dan formal untuk syarikat pembangunan perisian yang bersaiz kecil dan sederhana ini.

Kajian ini membincang dan membentangkan satu kerangka proses penyesuaian pembangunan perisian syarikat bersaiz kecil dan sederhana. Kajian ini berdasarkan kepada projek dan faktor pelanggan projek itu, jadi analisa adalah berkaitan dengan perspektif pelanggan, perlakuan projek dan peringkat-peringkat projek serta tujuan operasi penyesuaian proses, aktiviti dan juga strateginya. Kedua-dua metodologi kualitatif dan kuantitatif digunakan untuk mengesah dan menyempurnakan kajian ini. Hasil kajian menunjukkan pendekatan segera telah digunakan untuk menyesuaikan proses dan aktiviti pembangunan perisian dalam syarikat yang bersaiz kecil dan sederhana yang menggunapakai metodologi perubahan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CEO	Chief Executive Officer
CRC	Class Responsibility Collaboration
CTO	Chief Technical Officer
Dev	Developer
GSD	Global Software Development
PM	Project Manager
QA	Quality Assurance
RDP	Rule Description Practices
Team Lead	Team Leader
Tech Lead	Technical Leader

CHAPTER 1

INTRODUCTION

Overview

This chapter presents an overview of the software development process tailoring and its need in small and medium sized software development companies following agile based processes and practices, and role of client and client's perspective in software development process in context of Global Software Development (GSD). It also explains GSD and its influences on software development processes and trends. Further, it highlights the research problem and presents the research questions addressed by this thesis as well as research objectives that have been achieved. Finally, this chapter describes the organization of the chapters of this thesis.

1.1. Software Development Process Tailoring

Software development process tailoring is the activity of customizing or adapting a software development process. As defined by [1], *"Software development process is the collection of all processes and activities carried out during the entire lifecycle of a software product"*. Software development companies adapt these processes and activities of software development according to their own requirements. According to [2], software process tailoring is *"the act of adjusting the definitions and/or particularizing the terms of a general process description to derive a description applicable to an alternate (less general) environment"*, for example, tailoring the requirement management practices for small projects by adding necessary activities, deleting unnecessary activities, and merging or splitting the activities.

Constructing or redesigning a software development process from scratch is risky, inefficient, resource intensive and involves overheads [3], [4], [5]. Tailoring an existing software development process improves its performance and quality of the software product, minimizes risks and reduces the effort [4]. A good software development process improves the quality of the software product and brings stability in the progress of the project. Therefore, the software development process tailoring is an important practice to regenerate processes, and redefine and reuse existing processes.

Software development process tailoring can be carried out at two different levels such as organizational level and project level [6]. Complete understanding of organizational standards, procedures and project requirements is necessary to tailor a software development process at each level.

At project level, software development process tailoring activity requires complete understanding of the project, characteristics of the project and requirements of the project as well as understanding of the process to be tailored and expected regenerated process. Highly applicable skills of project manager are required for good process tailoring as required by the project and company. The consequences of bad process tailoring have been reported as follows [6]:

i. The project budget, development time, and software quality are greatly affected by bad processes.
ii. Bad process tailoring may include unnecessary activities in the software process that can lead to an increase in cost, wastage of time and omission of necessary activities.
iii. Bad process tailoring can produce a process that does not meet the organizational or international standards.

The main impact of bad process tailoring may appear on the quality of the software development process and software product. Therefore, software process tailoring is a challenging task for the project managers and software development companies.

1.2. Current Software Development Trends

With the advent of Global Software Development (GSD), there has been a rapid change in the overall software development practices since 2000 [7]. Globalization has connected the nations, their societies, and public and private sectors with each other in all aspects [7]. As a result, project outsourcing and distributed software development begin to be the most preferred practice during the recent years. These outsourced businesses are largely managed by offshore companies mostly operating in developing countries.

A number of factors such as cheap but competent and skilled resources, quick releases and deliverables, and launching products early in the market to meet the competitors have been reported as major factors behind the decisions of outsourcing projects to such offshore companies [7], [8]. According to an estimate, software industry generates 50% - 70% revenue from project outsourcing [8].

A general GSD environment is shown in Fig. 1.1 [9]. The owner of the project referred to as project client being in USA interacts with the offshore teams located at India and China following the processes required for requirement management, project management, change management, product release plans, and quality assurance. However, software development teams can be located in one country instead of different countries.

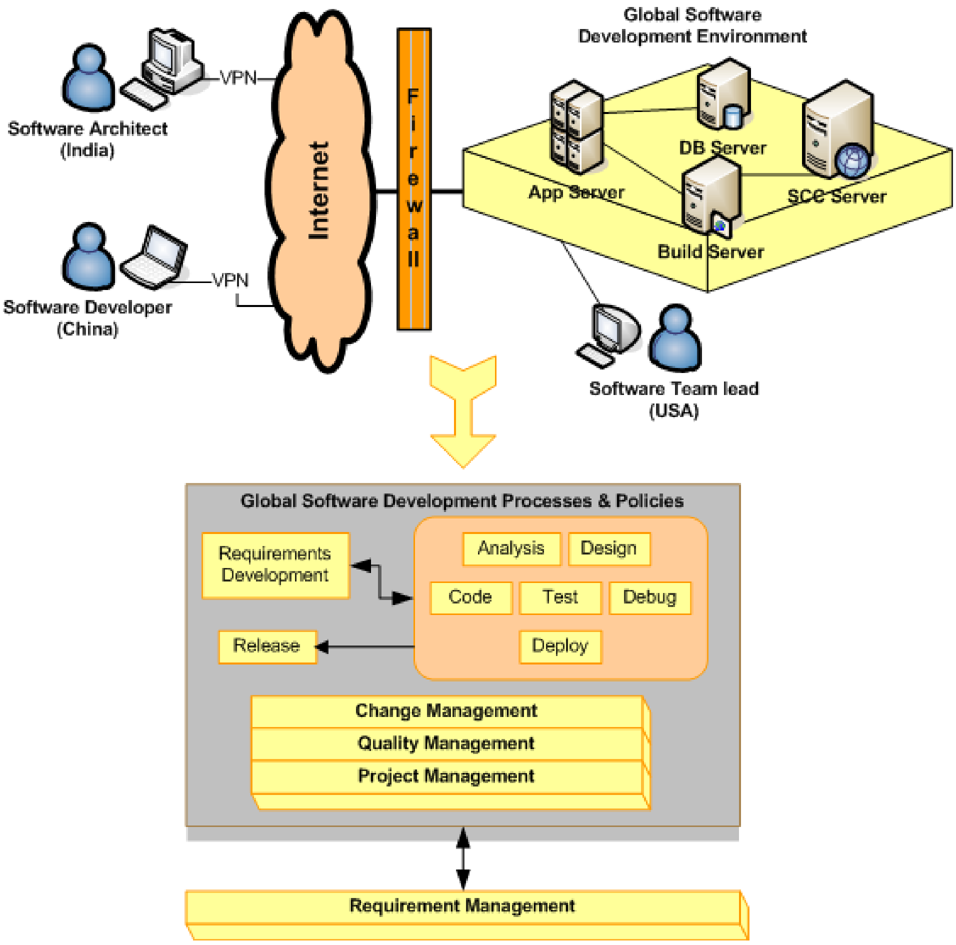

Fig. 1.1 GSD Environment [9]

The availability of low cost but skilled resources, information sharing, global markets and new ways of software development as a result of GSD [7] has made software development much faster. In order to meet the market competitors, clients need implementation of new requirements and modifications in existing software products without any delay, early completion of working code as well as quick releases of their products. Under such circumstances, clients have been more demanding and have set high expectations from the offshore software development teams.

There has also been observed a clear shift from traditional heavyweight software development approaches to the lightweight agile based methodologies [10]. Unlike large companies, small and medium sized software development companies have the human, technical, financial and infrastructural resources limitations such that these companies are unable to follow the traditional heavyweight approaches of software development and process standardization [11] such as CMMI [12], ISO [13], Six Sigma [14], and Lean Development [15]. The definition and size of these small and medium sized software development companies vary from country to country as reported by [16], [17], [18], [19], [20], [21].

Despite that CMMI and agile methodologies belong to two different schools of thought, both approaches present good practices for software development. Most of the misconceptions by agile followers about CMMI model belong to the CMM which is less flexible than the CMMI. Similarly, agile manifesto has been used by the agile followers as a reference to justify that processes are not necessary for the software development. The misconceptions about both approaches have been arisen due to the lack of correct information about their practices and their use which are based on the personal experiences [22]. CMMI describes what to do [23], whereas agile based methodologies describe how to do by providing lightweight practices for software development. In a sense, both agile based methodologies and CMMI can complement each other and both should be embraced by the software development industry [22].

It is a common practice for small and medium sized software development companies to follow lightweight agile based methodologies due to various factors such as short iterations, delivery of the software products in short time, accommodating changes quickly, minimum processes, high team interaction, dynamic prioritization, as well as their support to GSD [24], [25], [26], [27]. Agile based software development has also a strong and direct positive relation with the processes and project's outcome [27]. The support of agile based methodologies to meet the expectations of client, close interaction between client and software development team, accommodating changing requirements, short and quick releases, and iterations, and working code are the reasons for which small and medium sized companies prefer these methodologies.

1.3. Need for Process Tailoring

It is almost impossible to find identical projects or processes [4]. In fact, two companies are different from each other, two projects within the same company may also be different as well as a process applicable for one project may not be suitable for others [6].

Processes can be defined in two ways either by constructing a new one or customizing or adapting an existing process [28] according to the requirements of the company and project. The software process tailoring is the most applicable practice in small and medium sized software development companies which due to the limitation of resources are unable to create or follow new processes. Unfortunately, limited research has been carried out on software development process tailoring and most of which has focused on large-sized software development companies [6].

Although, the need of tailoring agile based methodologies has been recognized but existing literatures have not specified strategies for tailoring these methodologies [5]. This has arisen the need for a framework for software process tailoring in small and medium sized software development companies following agile based methodologies. This research addresses this issue and presents a framework to tailor the software development processes in small and medium sized software development companies following agile based methodologies.

Fig. 1.2 shows the process tailoring phenomena in context of GSD. As a result of GSD or offshore development, agile based methodologies got an overwhelming response from the small and medium sized software development companies. These companies follow agile based methodologies to meet the challenges of GSD. As mentioned earlier, these companies due to limitations of resources tailor the existing software development processes according to their requirements. Project managers are involved directly in tailoring the process according to the requirements of their projects.

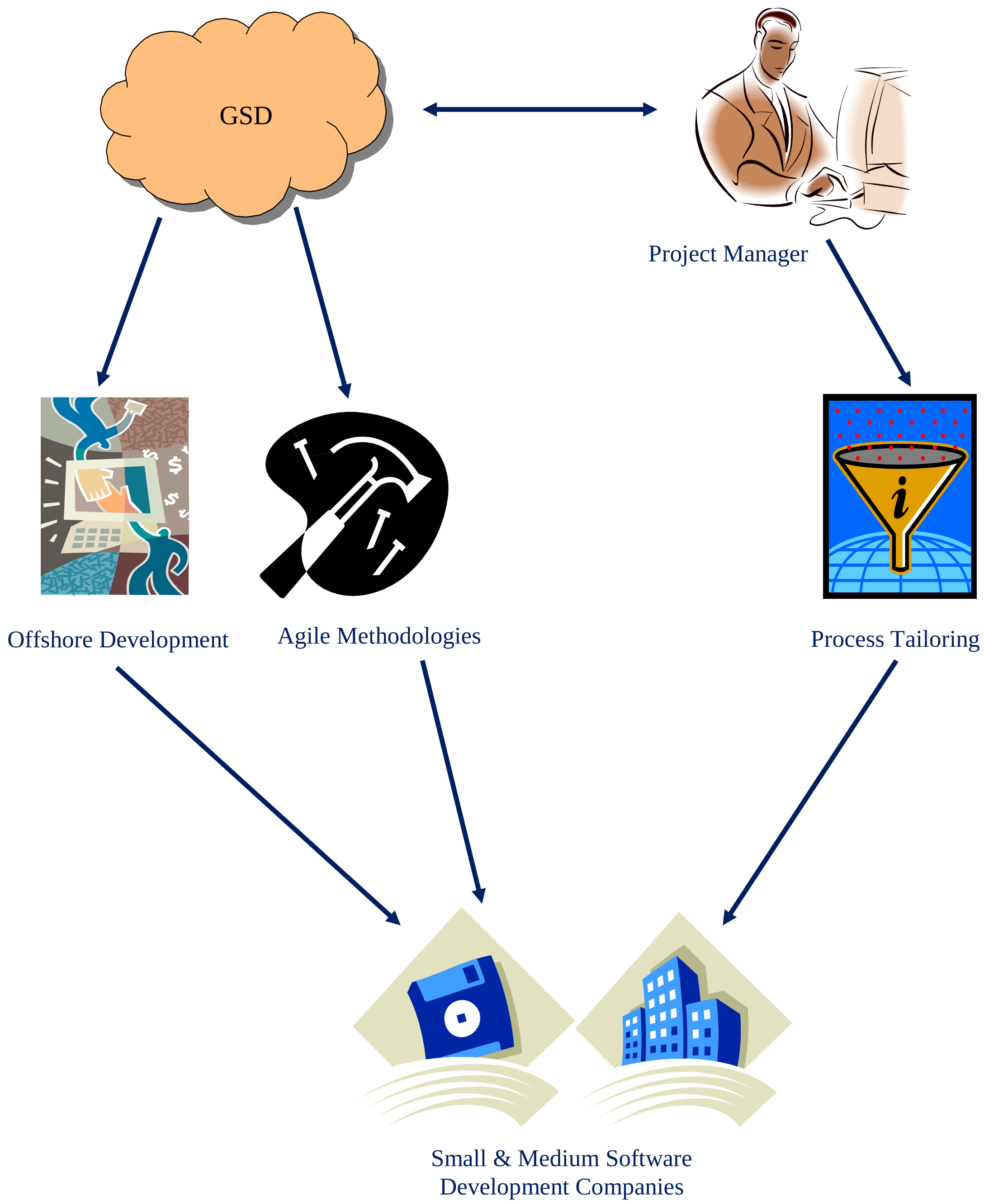

Fig. 1.2 Process Tailoring in GSD

1.4. Client Factor In Agile Based Software Development Projects

Weak and inappropriate processes in small and medium sized software development companies greatly affect the quality of the software product. A good software process produces good quality software and enhances the software development productivity [1].

Lower quality software products adversely affect the satisfaction level of the client and may lead to the cancellation or termination of project without completion. It has been reported that end of 85% of the software projects is failure while it has been further estimated that 31.1% of the projects might have been cancelled without completion [29].

The satisfaction of client has been considered very important in agile based methodologies [30]. Client's satisfaction is extremely important and significantly affect the company's reputation and business profile [31]. The agile based methodologies have also emphasized on the client based software development and have presented client as one of the most important and influential factor [10] such that involvement of client is the key success factor in software projects [32]. Therefore, small and medium sized companies consider client's requirements, expectations and satisfaction as important factors for the success of their projects which are preferred and maintained intentionally or unintentionally throughout the project.

As opposed to traditional software development methodologies, the lightweight agile based methodologies emphasize on the continuous involvement and close collaboration of client throughout the project lifecycle [33]. According to a study, in various process models, the role of client has been restricted to the requirement engineering process [34]. As compared to other processes of software development, in requirement engineering the role of client is mostly active. The success of agile development is dependent on the active participation of the client in the development process along with the software development team [10]. The importance of client has already been realized by human resource and business management group. Their business and operational strategies, and plans have also been modified accordingly since years [35] whereas, understanding and practicing client's perspective in software development projects should also be realized [34].

The research work reported in this thesis has emphasized on understanding and practicing the client's perspective during software development and has formulated a framework for software development process tailoring which tailors and derives the processes and activities mostly preferred and required by the client in small and medium sized software development companies. In order to meet the challenges of GSD, realizing client and client's perspective as an important part of software development processes particularly in small and medium sized software development companies is inevitable. Considering client as an important key factor, the proposed framework provides a systematic approach for tailoring agile based software development processes to derive lightweight processes and activities for software development in small and medium sized software development companies.

1.5. Research Problem

The existing process tailoring approaches provide very superficial guidelines to tailor a software development process. These approaches focus on first level tailoring which is suitable for domains such as informatics, medical, aviation and defense instead of tailoring at second level for specific project requirements [5]. These approaches are not based on the detailed understanding of the project's requirements, important factors, and characteristics which makes them inapplicable to tailor the software development processes of a specific software project.

Very limited work has been presented on tailoring software development processes in the context of agile based processes [36]. The existing models and frameworks on process tailoring are very general and mostly recommend similar practices. They do not tailor a software process specifically on project and client factors which have been considered as two most important and decisive elements of agile based software development environments of small and medium sized software development companies. There has been found a direct positive relation between stakeholder satisfaction and processes and outcome of the project [27]. The client and development team in agile based environments have been considered as the important factors [37], [38], [39], [5], [10].

The existing approaches tailor a software development process before the beginning of the project and follow that same tailored version of the process throughout the project [5]. This tailored process may not fulfill the requirements of the project at some later stage due to evolving software development practices and changing requirements of the client in agile based software development projects. Therefore, this arises the need to tailor the software development processes and their activities during the project development or at some specific level of project to accommodate the modifications in the processes or activities [5], [6].

One problem with existing process tailoring approaches is that most of them focus on large software development companies and less attention has been given to small and medium sized software development companies [6]. Therefore, the existing models and frameworks do not fulfill the requirements of small and medium sized software development companies following agile based methodologies. The major issues related to the software process tailoring have been reported by [5] such as:

i. The existing approaches do not take into account the effect of project characteristics on process tailoring practices.
ii. The limited work that is available on process tailoring is of very superficial level. It does not address agile methodologies properly and mostly compares them with the traditional approaches of software development.
iii. Existing work presents general principles for selected methodologies instead of elaborating process tailoring strategies.
iv. The software development and project management people have no proper guidelines and strategies to tailor a software process.

Such problems indicate that there is need of a systematic process tailoring framework for small and medium sized software development companies to meet the challenges faced by the projects [5], [6].

1.6. Research Questions

The research problems addressed by this thesis have been summarized by the following research questions:

RQ1. How would software development processes be tailored in small and medium sized software development companies following agile based methodologies?

RQ2. What critical factors play an important role in software development process tailoring in small and medium sized software development companies following agile based methodologies?

RQ3. How important client and client's perspective are for project success and process tailoring?

RQ4. How would software development projects progress and behave in small and medium sized software development companies following agile based methodologies?

RQ5. How would lightweight software development processes and activities be defined for small and medium sized software development companies?

The answers of these research questions will enable small and medium sized software development companies to generate lightweight processes, activities and practices for software development through process tailoring which will effectively address the issues of software development processes of these companies.

1.7. Research Objectives

The objectives of this research study can be described as:

i. To investigate the client factor and client's perspective in small and medium sized software development companies according to agile based methodologies.

ii. To analyze the behavior of risks or problems faced by the projects and classifying these risks into groups.

iii. To investigate how the software development projects behave and respond to various factors, issues and problems in small and medium sized software development companies and presenting project’s behavioral execution flow states.

iv. To formulate a software development process tailoring framework for small and medium sized software development companies according to agile based methodologies.

v. To generate a process tailoring schema specifying the implementation details of process tailoring operations.

1.8. Motivation

The significant motivation behind this study is addressing the issues of software development processes being faced by small and medium sized software development companies as consequences of GSD. The present study applies process tailoring technique to address this issue. It makes this research work more important and applied due to the limited research works presented in this regard and particularly in context of small and medium sized software development companies.

Secondly, emphasizing on realization of client and client’s perspective as dominant and critical factors in software development projects, and process tailoring is another motivational factor which increases the novelty of the present study.

Thirdly, presenting a comprehensive case study of real projects in software engineering research which lacks such case studies performed in real companies [6]. Fourthly, being both a professional project manager and an academic researcher focusing on the real industry scenarios and practices and giving them attention and recognition through published research papers following academic research methodologies.

Formulating a process tailoring technique with the blend of real software industry practices and existing research works is the real motivational factor behind this research study which makes this research work more significant.

1.9. Summary

Global software development (GSD) has changed the overall software development practices. Project outsourcing has been started and agile based methodologies have got overwhelming response from software development companies particularly from small and medium sized companies. Due to the limitations of resources these companies instead of creating a new process from scratch prefer to tailor an existing process according to their requirements.

In this regard, limited research works have been presented on software development process tailoring and particularly on tailoring agile based methodologies. The existing works provide superficial guidelines on process tailoring and do not focus on small and medium sized companies which adopt agile based methodologies. The present study addresses this issue and presents a process tailoring framework for such small and medium sized software development companies to tailor their processes.

Based on the research problem, the research questions have been formulated and research objectives have been set. The significant importance of process tailoring approach for agile based projects running in small and medium sized companies have been regarded as main motivation to conduct and complete this research work.

1.10. Organization of Thesis

The remaining thesis has been organized into seven more chapters as follows:

Chapter 2 – Review of Literature – This chapter presents an analysis on the existing research literatures that have been consulted during this research work.

Chapter 3 – Research Methodology – Next, the research methodology has been described that has been followed to complete this study.

Chapter 4 – Process Tailoring Framework – In this chapter the complete process of framework formulation has been presented and its various components have been explained.

Chapter 5, 6 – Case Studies – The case studies and their findings have been presented in both of these chapters. Chapter 6 additionally presents the findings of survey questionnaires to validate and support the case study findings as well as presents the comparative case (cross case) analysis of both case studies.

Chapter 7 – Results and Discussion – In this chapter the results of case study and survey questionnaires have been summarized and discussed in accordance with research questions of this study.

Chapter 8 – Conclusion – The summarized results, overall conclusion of the research study aligned with research questions and objectives as well as future research directions of the current study have been presented in this chapter.

CHAPTER 2

REVIEW OF LITERATURE

Overview

This chapter discusses the background study of Global Software Development (GSD), its effect on software development practices as well as consequences in the form of project outsourcing and migration to agile based methodologies. This chapter also highlights the various aspects of agile based methodologies, risks involved in software development projects, and software development process improvement and management issues. Moreover, in the context of GSD, agile based methodologies and issues related to the software development processes, this chapter presents an overview of the software development process tailoring and limitations and issues related to the existing approaches. Finally, this chapter presents the conclusion of the whole background study.

2.1. Background

Since the 2000, the Global Software Development (GSD) had started affecting the overall software development scenarios. The effects of GSD or IT globalization on the IT industry had been more prominent since early twenty first century. The evolutionary consequences of GSD had changed the existing software development trends from traditional practices to the emerging new lightweight methodologies. The traditional approaches of software development have been modified and reformed. On one hand GSD has changed the software development practices while on the other hand it has also affected the preferences and priorities of the software development teams, software development companies and project clients.

As a result of GSD, a new generation of software development processes and practices such as agile based methodologies, project outsourcing, distributed software development, process reusability and software process tailoring have been emerged. Software process improvement, process management, project management, risk management and process tailoring practices have become the most preferred areas of software engineering research due to the widely being used agile based methodologies.

Agile based methodologies and project outsourcing will be discussed in context of GSD. A critical review on risk management, project management, process improvement and process tailoring practices will also be presented. The existing software process tailoring approaches, their issues, advantages and disadvantages have been analyzed and discussed critically in detail.

The rapidly changing scenarios of software development will be comprehensively highlighted whereas, a big picture of the current situation of the software development industry particularly small and medium sized software development companies, their related issues, priorities and practices have also been discussed.

2.2. Effect of GSD on Software Development Practices

GSD has brought major changes in the overall software development paradigms. It has connected the world societies, economies and individuals on social, cultural and technological fronts and has promoted project outsourcing to geographically distributed offshore teams located in developing countries due to the factors such as availability of skilled resources at low costs, meeting market competitions, full day development due to time differences between different geographical zones, and benefits in terms of launching a product early in the market [7]. However, a number of issues have also been associated with GSD such as lack of effective communication channels, interaction and coordination issues in case of distributed work, delayed deadlines, loosing trust of the client as well as social and cultural issues. Weak communication channel has been attributed as the basic reason of project failures [40].

The traditional way of software development has also been changed from centralized development to the geographically co-located distributed software development teams. The projects have has been outsourced to some other local company or offshore company by the project outsourcing company considered as client in GSD environment [7]. The project outsourcing has been emerged as the most preferred practice of software development companies and clients during recent years. Similarly, project outsourcing has been presented as a strategy to standardize the solutions and functionalities within an organization [41] whereas, the decision of project outsourcing has been studied in relation to the stakeholders of the outsourced project. The importance of project outsourcing decision has been highlighted in the study. In addition to a number of benefits associated with the project outsourcing, many issues faced by the outsourced projects have also been reported.

2.2.1. Issues Related to Project Outsourcing

There have been many problems associated with the projects that have been outsourced to some other offshore companies or geographically distributed software development teams. A number of problems have been faced by such geographically distributed offshore teams. The various problems in project outsourcing namely preparing and managing outsourcing contracts, demand supply and billing management of the development team, documentation, process improvement, tools and technologies for software development, project planning and management, social cultural and communication issues, and motivation or morale of the offshore team have been highlighted whereas, creativity in software development in terms of coding, modeling, processes, and marketing has also been considered as the main problem of outsourced projects, and suggestions on resolving such issues have also been presented in [8]. The problems and their solutions have been reported as important in effectively managing the outsourced projects and improvement in the overall team performance.

In a work on risks associated with outsourced projects, the key risks have been grouped into six broad categories based on vendor (outsource service provider company) specific and project client specific risks [31]. The identification of the risks faced by the outsourced projects and proposing their solution has been made easy with these risk categories.

Outsourcing project to offshore company or team belonging to different country and culture has been identified as one of the key issues when both client and vendor company are in different countries [42]. In this regard, a relation between culture and control procedures in projects outsourced to offshore companies has been discussed. Control has been defined as the organizational level strategy to manage the activities of the team members according to a standard procedure to achieve the objectives. The effects of factors such as behavior and project skills on formal and informal modes of control have been identified. The relationship between various cultural parameters and both formal and informal control modes following the research model as shown in Fig. 2.1 have been derived as shown in Table in 2.1 [42]. The cross cultural issues in projects outsourced to offshore teams have been found as important for the success of the software development project.

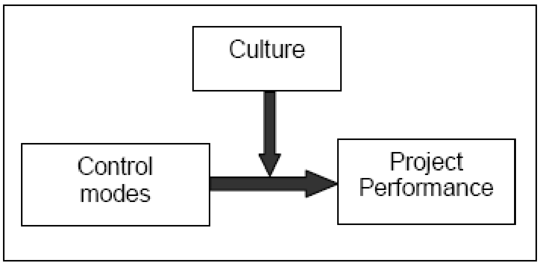

Fig. 2.1 Research Model [42]

Table 2.1 Relationship Between Culture and Control Mechanism [42]

Culture Dimensions	**Formal modes control**		**Informal modes control**	
	Outcome	Behavior	Self	Clan
High Individualism	√		√	
High Collectivism		√		√
Strong uncertainty avoidance	√	√		
Week uncertainty avoidance			√	√
Large power distance	√	√		
Small power distance			√	√

Project managers controlling cultural issues in offshore project may overcome the cultural barriers for making projects successful. To manage such distributed software development projects, recommendations have been made to manage the distributed software development project at infrastructure, planning, execution and organizational level [43].

In addition, the communication, interaction and coordination issues in outsourced projects have been reported as the main factors that affect the performance of the offshore teams. In this regard, a conceptual model for outsourced offshore projects as shown in Fig. 2.2 has been presented in which the role of performance parameters such as effort, rework and elapsed time has been analyzed in software development process, interaction and communication in outsourced software development projects [44]. The communication and coordination processes have been significantly found as the most important in projects outsourced to offshore companies while emphasis has been given on keeping balanced communication between client and offshore company.

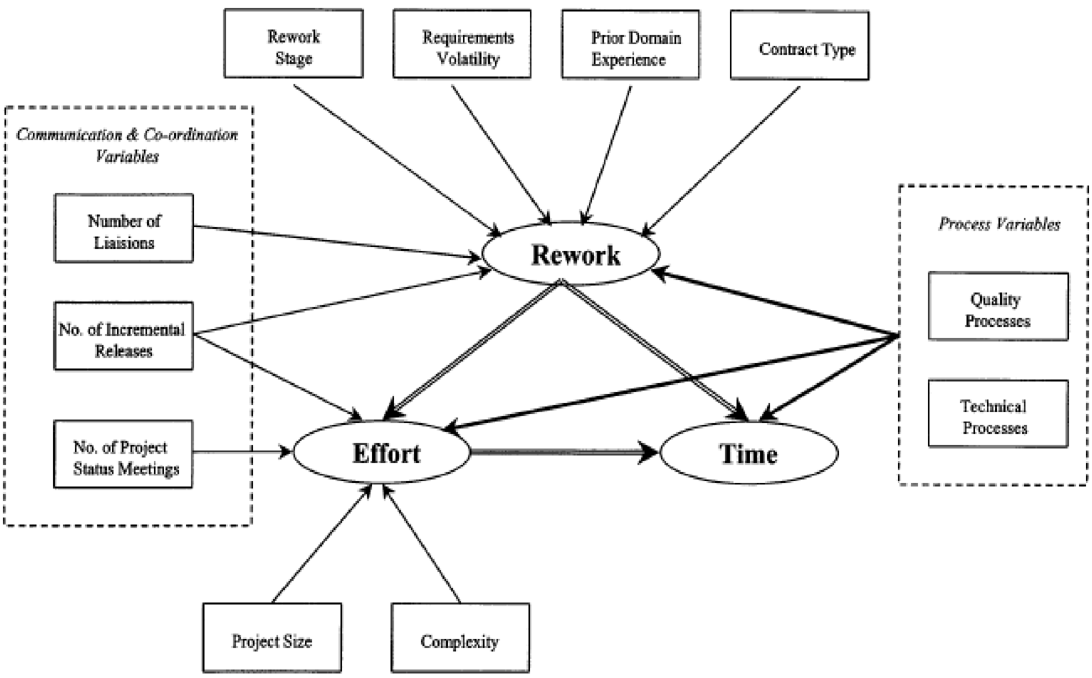

Fig. 2.2 Conceptual Model [44]

2.2.2. A Paradigm Shift in Software Development

As a result of project outsourcing and GSD, there has been occurred a major shift from traditional heavyweight approaches of software development to the lightweight agile based methodologies. The global software development trends have been clearly affected by this change as have been reported in a survey on the differences in international software development practices [45].

A large number of small and medium size software development companies has been appeared due to outsourcing projects to offshore companies most of which have been working in countries such as India, Pakistan, Malaysia, and Singapore since past many years.

In order to meet the consequences of GSD and requirements and expectations of the offshore clients for developing and delivering the software quickly in short iterations, agile based methodologies became the preferred choice of the software development teams. Such methodologies due to their support to GSD got an overwhelming response by the software development companies such that conventional software development practices have been increasingly replaced [10]. Agile based methodologies or hybrid approaches of software development have become the preferred choice of the software developers and companies.

2.3. Migration to Agile Based Methodologies

Software development methodologies have been evolving since years due to the rapidly changing tools and technologies and requirements of the client. In such rapidly changing scenarios, the lightweight agile based methodologies have been considered as a new generation of processes during recent years [10]. The agile models have been widely adopted due to their support to the offshore development in terms of quick and early iterations, fast coding and delivery of product, less documentation and adaptability to changing requirements.

There exist two prominent schools of thought regarding agile based methodologies such as supporters who consider code completion as the only important deliverable, and critics who support and emphasize on documentation and consider it necessary [46].

Many methodologies and processes claiming to be an agile method have been introduced since past many years. As such, there has been formed an agile alliance in a meeting in 2001 and agile manifesto comprising of twelve principles was presented to standardize such methodologies and processes [46], [30]. Each process claiming to be an agile have to have qualify these twelve principles of agile manifesto.

Migrating from traditional heavyweight software development approaches to lightweight agile based methodologies has always been considered as challenging. The extensive support and criticism on both methods has always increased the problems and created confusion in perception, understanding and selection of suitable method. A few of the fundamental characteristic differences between traditional approaches and agile based methodologies following principles of agile manifesto have been presented in Table 2.2 [10].

Table 2.2 Fundamental Differences Between Traditional and Agile Methods [10]

Factors	Traditional Approaches	Agile Methods
Software Development	Process oriented	People oriented
Project Management Style	Command and Control	Leadership and Collaboration
Team Role Assignment	Based on skill level of individual team members	Self organizing teams
Communication	Formal	Informal
Client's Role	Important	Critical
Process Model	Waterfall, Spiral, Prototype	Evolutionary approach
Project Lifecycle	Based on tasks or activities	Based on software product features

The software development approach has been found as the major difference between both methodologies as shown in Table 2.2 such that, unlike traditional approaches, agile methodologies follow practices adopted by the development team members [47].

In a comparison of agile and traditional methods, the factors such as organizational culture, process and project management, work appraisals, competent teams, client and team relationship, processes, and various software development and project management tools and techniques have been identified as the key issues in migrating from traditional software development approaches to agile based methodologies [10] whereas, agile methodologies have been found suitable for projects where requirements change rapidly. The agile methods have been found suitable for complex and rapidly changing projects based on exploratory problem solving approaches, and team centered and collaborative development environments [47]. Close collaboration between software development team, project manager and project client have been characterized as the basic requirement of following agile based methodologies. Unlike traditional approaches, the role of project manager in projects following agile based methodologies has not been more than a leader, facilitator, collaborator or coordinator [10], [48].

In order to clearly define the boundaries between traditional software engineering practices and newly emerging methodologies, a guideline framework has been presented to understand the relationship between these approaches [49]. The best practices of traditional software engineering approaches, agile methodologies and industry have been considered in the analysis. The analysis has been performed based on the best practices, relationship of methodologies and projects, history of practices of a methodology, similarities among various practices of different methodologies, understanding of developers on these practices, and terminologies used in various methodologies. It has been presented as an initial step towards making a reasoning model for selection of suitable methodology for particular project. As opposed to [10], the work presented by [49] finds the relationship between important software engineering methodologies.

The selection of a suitable methodology for a particular software development project has always been considered difficult because there have been found limitations in all software development methodologies [50]. Based on the analysis of prominent agile based methodologies such as Extreme Programming (XP) [51], [52],

[53], [54], Unified Process [55], and Scrum [56], [57], the limited support of agile methodologies to distributed development, contract management, reusability, large projects and large teams have been reported [46]. Traditional software development processes have been characterized by process overloading and unnecessary documentation which makes them inapplicable in many projects as well.

Since past few years, an overwhelming response has been received by the agile based methodologies due to the rapidly evolving software development practices. Due to the applicability and suitability of agile methodologies to various development environments [58], a number of software development companies have migrated to agile methodologies. However, it has been found challenging for the companies following traditional approaches since years to migrate to agile methodologies [10] where quality, cost and schedule are the motivational factors for process change [58].

2.3.1. Application Domain of Agile Methods

During last few years, web applications have gained much popularity and most of the projects being developed in software development companies are web based systems [17] such as online business applications, social networking, online banking and many others. Unlike traditional software development projects, the web application development has been attributed as very dynamic, interactive and challenging task. Majority of the software process models and frameworks that have been presented for developing web based applications are highly resource intensive [59] and have not been considered suitable for small size software development companies [17]. Recommendation has been made to follow agile approach for developing web based systems that have been evolved into complex information system applications [60].

A five step guideline model has been presented to address the issues and problems of small companies working on web applications but at very fundamental level without highlighting technical details about issues and problems of such companies [17]. Similarly, an agile based web application development method has been proposed considering requirement and analysis phases in agile methodologies important in web application development [61].

In agile based software development environments, Extreme Programming (XP) practices have been followed by majority of the software development companies and have been used in the form of pair programming [62]. The detailed analysis on pair programming, its various aspects, issues and problems, benefits and implementation strategies have been presented in [62], [63].

In another work, using XP as basic agile methodology, the compliance of agile methods with the ISO 12207:1995 software development standard have been analyzed and guideline has been provided on how agile based practices can retain their characteristics when adapted to ISO standards [64]. It has been concluded that the agile based methodologies can be adapted in compliance with ISO 12207:1995 standard. However, the principles of agile manifesto should not have been ignored during such practices.

2.3.2. Agile and People Factor

The client in software development projects following agile methodologies have been considered as the most important and critical factor [10] such that emphasis on client and face to face communication with the client has been given in agile manifesto [46], [30]. The factors such as communication, interaction, and coordination with the client throughout the project lifecycle have been considered important for agile based methodologies.

In this regard, three modes of interaction with the client such as user, designer and client centric have been proposed based on the principles of transparency, responsibility, staff training and education [65] such that designer, client or user might have been the decision maker in the software development process.

The impact of behavior of client on the performance of the software development project has been examined with respect to the relationship between traditional tools of software development and system dynamic models [34]. The emphasis has been given on the collaborative interaction between client and project manager. The deep understanding of client's behavior has been provided to the project managers. Understanding the client, client's perspective and realizing its importance by the managers have also been emphasized in [35].

The role of client in distributed development environments following agile methodologies has been considered more important and decisive because of the limited support provided in agile methodologies for these kind of environments [46]. Achieving the client's trust and confidence in such environments is necessary for the success of the project. In this regard, various aspects to increase the confidence of client in distributed development environments has been investigated in [66].

Infact, all the stakeholders of a software development project have their own importance in the overall software development process. In a study on relationship between agile methodologies and satisfaction of stakeholders, positive effect of agile methodologies on stakeholder's satisfaction has been found as being directly associated with the software development process and productivity of the project [27]. The various aspects of the role of stakeholders and users in the software and information systems development have been investigated and explained in many other research works for example [67], [68], [69], [70], [71], [72].

2.3.3. Team Roles in Agile Based Projects

Team building in agile based software development projects has been considered as an important initial level task that has to be done before a project starts. Self organizing teams encouraging role interchanging among the team members have been promoted by the people oriented agile based development environments [10]. However, despite this fact, role distribution in software development teams following agile methodologies has been realized very important. Software development teams with good distribution of roles and responsibilities produce good quality works [73]. Besides personality, knowledge and technical expertise of individual team members, the role assignment has been considered as the main element of software development team building [74].

The team roles have been divided into formal and informal types such that formal roles are defined by the project managers through a standard process whereas informal roles appear through interaction among team members itself [74]. Four informal roles have also been identified on the basis of a) tasks and related coordination, b) team work and group building through cooperation and interaction, c) lack of cooperation and technical expertise and d) high expertise and technical programming. The emphasis has been given on the need of defining roles based on tasks and groups factors. A comparative study has been presented on various role schemes in web based software development teams [75]. In addition to the original tasks of the team members, distribution of project management tasks among the various roles of the team members has also been recommended in [76].

The role distribution and team management in software development projects following agile based methodologies is very important. Self organizing teams in agile based development need more management by the project manager or leader of the project and also require more attention in case of offshore development of outsourced projects. Failing which may lead to the severe problems and issues in the projects.

2.4. Risks in Software Development Projects

The problems and issues are faced by all the projects throughout their lifecycle and have been termed as risks in software development projects. No software development project has been considered as identical [77]. Hence, the impact of these problems is more severe in projects outsourced to offshore teams following agile based methodologies characterized by faster development.

The problems and issues related to the outsourcing contracts, project documentation, software development processes, software development and project management tools, cultural differences, communication, team mentoring, and creativity have been found as the key risks for outsourced projects [8]. Other risks for example company's reputation, competition, client's trust and expectations, customization and complexity of the software, and geographically distributed locations have also been highlighted in [31]. Various risks associated with the software development projects have been examined, analyzed and discussed in many research works.

In addition to other factors, the progress of software development projects also gets affected by software processes and practices [78], [45]. The risks faced by the software projects have been divided into three main categories such as technical risks, management risks and scope change risks as shown in Fig. 2.3 [79].

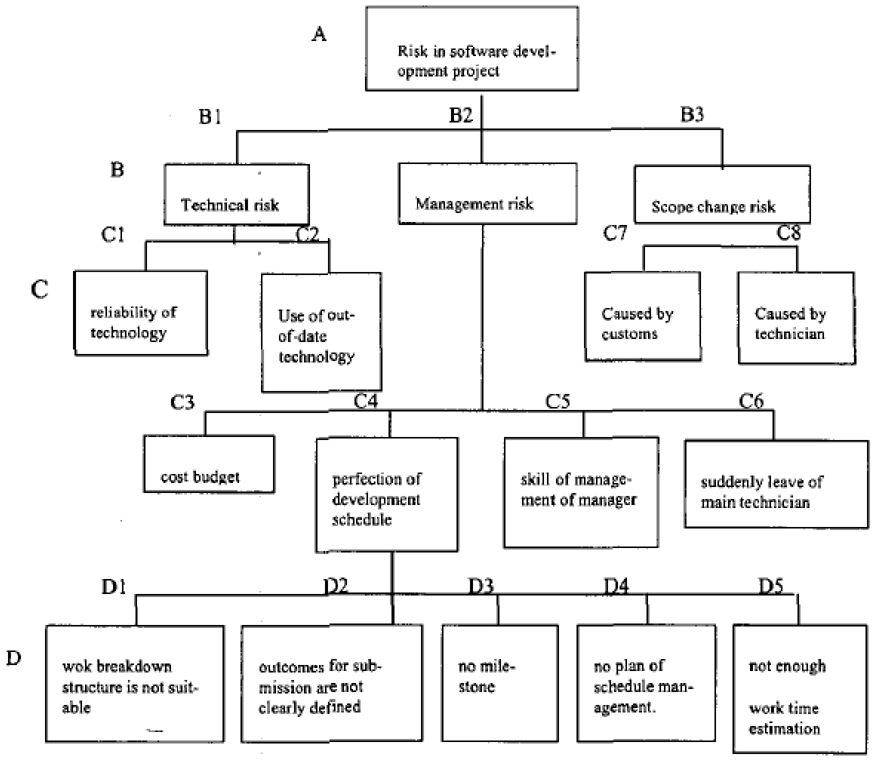

Fig. 2.3 The Structure of Risks in Software Development Project [79].

Technical risks have been divided into reliability of technology and use of outdated technology. Management risks have been classified into risks related to the cost, schedule, project manager’s skills, suddenly leaving of a main technician. Risks associated with both owners and contractors of the project have been identified based on the evaluation of each risk class (A, B, C, D) through empirical study. Feasibility of project, design of the project, implementation, and knowledge management during beginning, development, execution and completion stages respectively have been found as the risks mostly faced by the project owner. Whereas, change in project scope, project management, and technical risks have high impact on the projects from contractor's point of view. The major problem with this risk model is that it presents risks mainly from project management aspect and restricts the risks only to the management risks, technical risks and scope change risks. Further, risks have been discussed at very superficial level without considering other types of the risks.

In a similar kind of work, risks related to the project owners and contractors have also been analyzed [80]. Based on evidential assessment, a structured approach to find, analyze and rectify the risks in software development projects has also been presented [29]. The factors such as clear and understandable requirements, realistic project scheduling, budgeting, and an experienced, skilled and competent project manager have been found important to avoid risks and ensuring success of the projects [81]. Accordingly, building the team of good resources, maintaining and increasing the pace of development by minimizing drain of resources, quality improvement, good management, progress tracking and evaluation, good decision making, and post-mortem analysis have been identified as crucial factors for project success [82].

2.4.1. The Role of Project Manager in Risk Management

The importance of role of project manager has been widely accepted in software development projects. In projects that have has been outsourced to offshore companies following based methodologies, this role becomes more critical due to a number of risks involved in such kind of development environments where project manager has been considered not more than a facilitator, leader and collaborator [10].

In majority of the projects, project managers are not kept aware of the issues and problems in the project by the team members having fear of raising the questions on their performance [83]. This reluctant attitude creates troubles for the projects in future. For reporting a problem to project manager, the relation between two traits of team members such as assuming it as personal responsibility and/or doing it willingly has been found significant [83].

On the other hand, inability of a project manager to handle risks, project managers being unaware of project progress and status, or weak project management have also been considered as the basic reasons of project failure. Emphasis has been given on realistic project planning, scheduling, resource management, and tasks allocation for good software project management [84].

Resources and tasks allocation before execution of a project is an important task for project managers. A formal approach for project managers to manage project resources and processes has been presented in [85]. Project management tools play an important role in effective project planning, scheduling, resource management and tasks allocation such that integrating these tools with workflow management systems can be much effective [86].

Formal interaction, communication and coordination processes are implemented by successful project managers in order to keep track of the project progress and performance of the team members [87]. In this regard, detailed guidelines for good project management of project's beginning, planning, execution and completion phases have been provided in [88] which have been further reviewed in [89].

Good project teams and leadership qualities of a project manager have been considered very important in effective risk management as such a timely and right decision by a project manager handles the risk effectively but a slow response converts risks into losses in a software development project [90].

2.5. Software Development Processes – Improvement and Management

Since years, unreliable, over cost and low quality software products have been produced by the software development companies [91]. The situation has been more worsen due to rapidly evolving software development practices. However, good quality software development processes can improve the quality of the software and satisfaction of client [91]. Besides technology, time, and cost, it has also been realized that software development processes must be client oriented and must have to fulfill the requirements of the client [92]. Therefore, change, improvement and management of software development processes have been considered very important for software development projects. In this regard, a number of software engineering research works have been presented on various aspects of software development processes.

With respect to its size and type of projects, a set of processes is tried to be followed by each software development company according to its requirements. During recent years, the organizational structures have been practically changed from departmental level divisions to the business process teams [92]. The preferences and priorities of the organizations have been changed from business point of view such that project management and support process have been found important for progress of the project [92]. Further, in process oriented companies as shown in Fig. 2.4, it has been considered necessary to make changes to meet the requirements of the client [92].

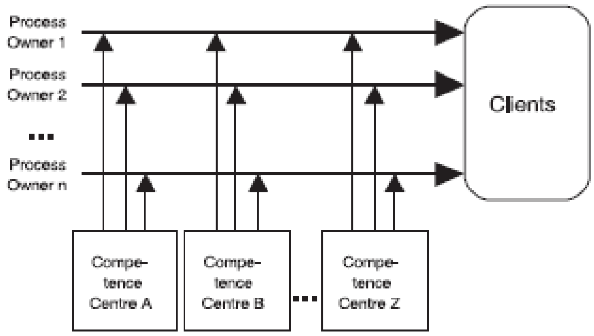

Fig. 2.4 Structure of Process Oriented Organization [92]

A structural hierarchy in which process management team holds the top position and bottom position as held by process execution team has been recommended to build up a process oriented company.

In the context of software development processes improvement and standardization practices, the traditional software engineering process models and process standardization approaches such as CMM, ISO, SPICE have been reviewed in [78]. The traditional waterfall model still has been considered as the fundamental model of software development. On the other hand, the response of small and medium sized software development companies towards process improvement and standardization approaches such as CMM and ISO have also been not found unsatisfactory. The priorities of the companies to follow a suitable process improvement, standardization and management approach may vary depending on the various factors such as project requirements, organizational objectives and preferences.

The various issues related to the standardized process improvement approaches have been reported in [93] such as:

i. Less contribution of academic research works on the effectiveness of these practices.
ii. All the approaches are based on very similar set of techniques and practices.
iii. Neither approach recommends best practices of software development. Mostly these approaches specify the improvement practices that works well with established practices or should exist in an organization.
iv. Only established processes have been improved instead of improving a new process.

As these approaches have been proposed by the industry, therefore more academic research contribution is required to further improve their applicability [93]. Due to the gap between academic research works and industry practitioners, software engineering research works most likely have been unable to address the issues and problems of the software development companies on real grounds. The basic reason behind the inapplicability of proposed works is the unavailability of real data of the projects running in the industry. Therefore, a close collaborative approach between academic research community and industry practitioners is required [93].

In a comparison of research methodologies used in different research papers of various science discipline, the software engineering research papers have been found with worst methodologies [94], [95] which has been attributed to the poor understanding of the research model and inability to perform testing due to unavailability of data [96]. Similarly many other research works have been presented on software engineering research methodologies such as [97], [98], [99], [100].

In currently building scenarios as a result of GSD, the project management and process improvement approaches have become more important for the software development companies. Therefore, the emphasis has been given on integrated process management for management of software development projects using automated tools [86]. Software development process change, improvement and management are part of continuous process development procedures in well established and large companies. Various aspects of software development processes and their improvement practices at organizational levels have been investigated in [101], [102], [103], [104].

The software development processes have always been a continuous part of debate among the software engineering research community. A number of issues, problems and misleading facts have been associated with them. In this regard, the leading misconceptions about software development processes have been corrected as follows [105]:

i. Business process improvement is not dependent on process improvement.
ii. Process improvement is not directed by process change.
iii. Software development processes are very critical.
iv. IT processes are required by business process activities.

Lack of best practices for software development has been found as the common deficiency of all process improvement approaches [93]. The software process change and improvement process is time consuming and demands commitment and dedication from the managers. Considering commitment as one of the most important factors in software process improvement, three types of commitment such as ‘affective’, ‘continuance’ and ‘normative’ have been described [106] whereas, continuous efforts are required soon as the improvement process starts [23].

Among many other research works such as [107], [108], [109], [58], [110], [11], [111], the work presented in [23] providing a platform comprising of common software improvement practices have been considered significantly important.

The heavyweight process improvement and standardization approaches have been considered ineffective and unnecessary by the project managers of small and medium sized software development companies [111] due to the limitations such as heavy documentation, resource intensive, unnecessary practices, reviews, high costs of training of staff members, and improper guidance [11]. In such circumstances the reusability of software development processes has been considered as an important solution of weaknesses in software development process change and improvement and approaches. To standardize the software process reusability, a meta-model has been presented that integrates the components such as people, processes, roles and infrastructure facilities using the reusability and standardization as the same process [109].

The satisfaction of client has also been considered as a parameter of software process improvement [111]. Software quality, budget, project planning and scheduling as well as organizational performance have been identified as the motivational factors for the project managers in making modifications and improvements in the software development processes [107], [58]. Furthermore, team expertise, automated software development, product complexity, deadlines and communication have been considered as major factors responsible for quality of the software and organizational performance [107].

Under the current scenarios when existing heavyweight approaches of software process change, improvement, and standardization have been unable to resolve the software development processes issues particularly of small and medium size software development companies, software process tailoring practices have emerged as the feasible practice to address these issues of software development processes. The following section presents prominent work on software process tailoring practices.

2.6. Software Process Tailoring - An Emerging Approach

Software process tailoring is a standardized practice to improve the quality and reusability of the software development process thus by removing unnecessary and unwanted practices and activities from it. Each software development project is different from other with respect to its size, scope and complexity therefore, a process suitable for one project might not be applicable for the other [112]. Designing a software process from scratch has been considered as quite resource intensive and expensive such that the overall cost of the project exceeds [3]. The main objective of process tailoring is to adapt and reuse a software process according to the requirements of the company [6].

2.6.1. Problems and Limitations

The efforts on process tailoring had been reported back in 1980s, but research works on software process tailoring has been presented since 2000. Relatively being a newly emerging approach, very limited research works have been published in this area [6] whereas, most of the works have described tailoring activities not the strategies to perform tailoring activities or operations.

In the existing work attention has not been paid to the small and medium size software development companies [6]. Most of the work that has been presented on software process tailoring is based on similar kind of factors such as organization, knowledge base and project. The procedures of standard compliance or conformance testing to some process standard such as CMMI, ISO or agile and various issues related to the compliance or conformance testing have been rarely addressed in the existing approaches [6]. Most of the approaches recommend tailoring a process before starting of the project which mostly fails in projects following agile based methodologies.

Most of the existing works that have been presented on tailoring agile based processes focus on tailoring of only a single agile methodology such as XP which makes it inapplicable for other agile based environments. Therefore, a very general process tailoring framework applicable to a broad range of agile based development projects is required [6].

2.6.2. Prominent Works on Process Tailoring

In the existing works on software process tailoring, similar operations have been performed to tailor a process. Few of the approaches have been found complex and inapplicable to certain environments and projects such as [113], [114] has presented an optimization based approach for software development process tailoring based on creating a balance between collaboration and discipline. Discipline has been considered as plan and processes used during design phase whereas, collaboration has been related to the interaction among people involved in the processes. The approach followed in the research work makes it difficult to follow and does not provide detailed process tailoring procedure based on the fundamental operations of process tailoring activities.

Most of the works have focused on general environmental and organizational factors. Such an evolutionary process improvement approach has been presented to tailor a software process based on the project goals and environmental factors by analyzing the errors, faults and failures of the software development projects in [115]. However, focusing only on defects or errors as being one of the improvement factors in software development projects makes the approach inapplicable in case of other factors.

In another work, process tailoring has been performed based on the product and activity tailoring following GV-model which is mostly used by German companies [116]. Only deletion and modification tailoring operations have been performed. The inapplicability and unsuitability of GV-model outside the Germany has been found as the major limitation of the work.

In most of the process tailoring approaches verification procedures for the tailored version of processes have not been described. In order to meet this issue, an Activity

Artifact Graph (AAG) based approach to tailor the process modules of a software process model has been presented in [3]. Reusability of process modules has also been considered during the tailoring process and four tailoring activities such as addition, deletion, splitting and merging have been performed to tailor the process module. Different process verification techniques namely syntactic error checking of AAG, type conformance checking and standard compliance with ISO/IEC 12207 have been followed and a case study has been conducted to verify the proposed approach.

In context of agile methodologies, XP method has been tailored and applied for large life critical systems [117]. Advantages and disadvantages of practices of tailored version of XP in accordance with large system have been investigated and their suitability to these systems has been analyzed. It has been found that various practices of XP or agile methodologies can be applied to the large systems as well as life critical ones.

In addition to other factors of tailoring a software process, the knowledge base or information repositories of past project data have also been considered an important technique to tailor the software process in many research works. Majority of the researchers rely on the data related to the past project's history. A knowledge based process tailoring framework comprising of project and organizational characteristics, process modules and context in which the process module had to be tailored has been presented in [118]. The process knowledge has been captured through a prototype tool developed and guidelines have been provided on acquiring the process knowledge required to tailor a software process. According to the findings of the study, the knowledge of the process that has to be tailored is very necessary to understand the tailoring process.

Similarly, the role of generalized and contextualized knowledge types in software process tailoring has also been analyzed [119]. In another work, the effect of experiences gained from the successful software development processes on software process tailoring have been found very significant and positive [120], [121]. Obtaining correct information about tasks, artifacts and deciding about relevant, necessary and irrelevant information is very important in knowledge based process tailoring [119].

In another type of knowledge based process tailoring, the processes have been tailored using structural similarity approach based on the idea of selecting and using similar processes for new projects from among the processes that have been previously used [122]. The processes have been categorized based on the factors such as lifecycle, number of iterations, type of application, project size, and software complexity and three tailoring actions such as add, delete and replace based on the tailoring rules have been applied to process meta-model [28]. Also, the method to find the structural similarity among the processes has also been presented. It has been found that following this approach, fewer modifications would have been required to tailor the process to make it suitable according to the requirements of new project which ultimately reduces the tailoring cost and effort.

The complete understanding and knowledge of a software development process that is to be tailored is very important for the successful process tailoring. In this regard, an approach of software process knowledge base consisting of information of existing techniques and practices to generate a process has been used to tailor agile based processes [112]. The process tailoring has been performed at framework level where knowledge base is maintained, and at application level where tailored version of the process has been applied to resolve the problems of the system. In another effort, RDP technique using CRC cards comprising of rules of engagement and rules of play have been used to tailor the software process [123]. The rules of XP have been modified very precisely and tailored using rule based RDP cards technique. Other research works that highlights the practices of XP in various environments are [124], [125], [126].

A general procedure to customize the activities, artifacts and roles of a software process meta-model has been presented in [28]. The identification of problems first has been considered necessary for deriving an effective process through process tailoring. Such kind of approach has been followed in [127] where four system analysis strategies have been defined based on the software and environmental factors considering both immediate and long term requirements of each factor. The same approach of system analysis as defined in [112] has been followed during the tailoring.

The software process tailoring has been considered as a time consuming process which is mostly done before the starting of the project. To reduce the time spent on software process tailoring, an artificial neural network approach using semi-automated filtering technique comprising of three phases namely process filtering, reconfiguration, and feedback have been used [128]. During the process filtering phase the tasks of a process are selected from the tasks repository and precedence is set during reconfiguration phase and finally a tailored process is derived. The approach followed in the study has been found complex and unsuitable to be followed by other companies.

In a comprehensive review of existing works on software process tailoring, the answers of the following questions raised from time to time about process tailoring has been found and discussed [6]:

- What are the main process tailoring approaches and methodologies that exist?
- What kind of process tailoring and standard conformance guideline is available?
- Are there any real case studies available on software development process tailoring?

Based on the analysis of existing process tailoring approaches and answers found of the above questions potential research areas of software process tailoring have been identified such as [6]:

- Development of a systematic process tailoring framework for small and medium size software development companies.
- Realizing the need of standard compliance procedure.
- Development of a general process tailoring framework applicable to various diversified development environments.

Further, in an evolutionary approach of software process tailoring, addition, deletion, splitting and merging operations have been applied on four different process blocks namely sequence, concurrence, selection and iteration block [4]. The process tailoring has been performed on activities of software evolution process.

In another approach, the software development challenges and real issues related to the resources, communication, requirement management, and political and technical faced by the project goals and environment have been addressed by process tailoring [5]. A number of tailoring operations namely add, delete, downsize, drop, expand, redefine and replace have been applied to tailor the tasks, sequence, artifacts and roles. Similarly, an outline to tailor the software improvement process for small projects running in small companies has been presented but no implementation details and findings have been provided [11].

2.7. Software Development Practices in Small and Medium Sized Companies

The consequences of GSD has also appeared in the form of mushroom growth of a large number of small and medium sized software development companies. Outsourced projects have been considered as the main source of business and clientage of these companies and have also been considered as the main reason behind the existence of these companies by the software engineering research community.

The definition of small and medium sized companies varies from country to country. According to a European Commission report, the size of the company is determined by the number of people working in the company such that companies having less than 50 and around 250 staff members qualifies for small and medium sized companies respectively [16]. The definition of small and medium sized companies has been presented according to the European companies which may not be applicable to other countries. Similarly, the companies having 50 or less employees have been reported as small companies according to the general definition of small companies whereas, small companies having 3 to 20 employees operating in a large number of countries such as Brazil have also been reported [17]. Many other definitions of small and medium sized companies according to different countries and environments have been given in [18], [19], and [20]. The contribution of small and medium sized companies in software production has been found significant. In Ireland that has been considered as one of the main country in the field of software development 99% companies are small and medium sized having less than 50 employees whereas, India has been exporting software since 1970 and is considered as one of the main software exporting countries [129].

As discussed earlier, these small and medium sized companies mainly adopt agile based methodologies for software development. In a review work on GSD and use of agile based methodologies, [130] has identified performance of global software development, governance related issues, and software engineering process as the main challenges faced by the companies in context of GSD. Similarly, the work presented on agile and scrum methodologies related to GSD has also been discussed. A comprehensive review has been presented in this article which helps in understanding the relation between GSD and agile methodologies. The limitations of the study have been found in the areas of GSD and its consequences on small and medium sized software development companies. It has also been identified that empirical findings in this context are very limited.

With the beginning of IT globalization, the interest of software engineering researchers has been developed and more attention is being paid towards the challenges faced by the small and medium sized companies related to their software development processes. In this regard, good quality works on various aspects of software development practices such as adoption of agile methodologies, software process improvement and process tailoring approaches of small and medium sized companies have been presented.

Out of many research works that have been presented on agile methodologies, limited works have been focusing on small and medium sized software development companies. Most of the available work on agile based processes in small and medium sized companies mainly highlights the importance of agile methods and their suitability to the development environments of these companies and have rarely presented any approach to measure the agility factor in these companies. [129] has presented a model to measure the agile factor in small and medium sized software development companies which is based on the existing models in this regard. Limitations of the existing models have been assessed as shown in Table 2.3 and a new model with enhanced features has been presented.

Table 2.3 Analysis of Agility Assessment Models [129]

Assessment Model	**Advantages**	**Disadvantages**
Boehm and Turner's Model	Thorough analysis of project	No analysis on administrative practices, environmental conditions, and agile values
4-Dimensional Analytical Tool	Thorough analysis of agile based practices and values, and software process characterization	Poor analysis of environmental variations and projects.
Model by ThoughtWorks Studios	Thorough analysis of development practices and administrative aspects	No analysis on environmental factors, superficial analysis of projects and agile factors.
Model based on Team agility assessment	Thorough analysis of software development teams	Poor analysis of agile and environmental aspects

It has been tried to include the environmental, project management and software development factors in the proposed model. The proposed model presents the guidelines to measure the agile factor in small and medium sized companies but does not provide implementation and validation details of this model. No results have been presented on the applicability and impact of the working of the model in such companies.

Though the support of agile methodologies to the development environments of small and medium sized companies has been recognized but as a matter of fact its support to the large companies, large projects and distributed applications cannot be ignored [131]. In this regard, a research study on the suitability of agile method such as extreme programming (XP) to the distributed development of large scale projects has been analyzed [131]. The study finds agile methodologies as the possible solution of the issues faced during the distributed development projects.

In another study, the impact of agile methodologies on various projects has been debated. Both opponents and proponents of agile methodologies from academics and industry have presented their point of view but study has been unable to reach any conclusion in this regards [132].

In addition to the role of agile methodologies in small and medium sized companies, a lot of works have been presented on software development process issues and other challenges faced by these companies. In this regard, software engineering practices followed by small and medium sized companies have been discussed in the context of changing development environments in [133]. Similarly, a model has been presented on domain specific modeling approach in small and medium sized companies by [134]. The model focuses on flexibility of requirements and iterative development factors and presents a pilot study in this regards.

Software process improvement practices in small and medium sized companies have been found as another area of main concern by the software engineering research community. The priorities and preferences of small companies for their process improvement have been highlighted by [135] basis on the SPICE model. similarly, the success factors for the success of the projects in small and medium sized companies have been presented by [136]. The study emphasizes on monitoring the processes based on internal success factors of these companies. The study presents a good insight into critical success factors for small and medium sized companies. In similar works, [137] and [138] have discussed small and medium sized companies in Brazil and Finland respectively and have elaborated software process improvement approaches in these companies of both countries. [139] has presented a systematic review of work presented on software process improvement approaches in small and medium sized software development companies. The paper highlights the important works in this area.

In small and medium sized companies software process tailoring has also been regarded as an important practice. A detailed discussion has been presented on software process tailoring in section 2.6 in this chapter. However, in the context of small and medium sized companies, very limited works have been presented. In a research study, software process slicing technique has been used to tailor the software process [140]. The process slicing has been made on the sub-processes and activities of the software process on the basis of case based reasoning of past experiences. The work is technically quite good but does not focus on the software development environments of small and medium sized companies.

Similarly many other researchers have presented good quality works on tailoring various processes such as [141] and [142]. These and other work on process tailoring present different approaches of tailoring software development processes. The main problem with the existing papers is that though they tailor those processes such as XP, RUP which are used by the small and medium sized companies but they do not consider issues, problems, factors and characteristics of small and medium sized companies. More specifically, their work does not focus on small and medium sized companies. [143] has presented a good work on software processes of small and medium sized companies but as mentioned earlier that based on the review and analysis of existing software tailoring works, [6] has found that very less amount of work has been presented on small and medium sized companies.

Agile based methodologies, software development process improvement and tailoring approaches have been found as the most potential areas of research in the context of small and medium sized companies. There is the need to continuously present more good quality work in this regard.

2.8. Summary

With the advent of GSD, the existing software development trends and practices have been tremendously changed. Agile based methodologies are widely being used by the software development companies. Project outsourcing to offshore software development companies have become the preferred choice of the companies and project clients which have ultimately promoted the distributed software development practices. Due to the latest tools and technologies as well as effective communication channels the barriers of geographical distances and cultural differences have been effectively handled.

The prominent effect of such rapidly evolving software development practices and environments has been appeared on software projects and software development processes. The preferences of the clients, companies and developers have been changed as well as commercialization and business aspects have changed the priorities of all the stakeholders. The software development has become very fast, whereas traditional approaches of software development have been unable to meet the requirements of the current software development paradigms. To overcome such situations, the existing software development practices and processes are increasingly being redesigned and regenerated. New methodologies and practices are being adopted to complete the work in earliest time to achieve the satisfaction and trust of the client for project success and to earn more business.

In this regard, the software engineering research works that have been presented so far have not properly accepted and addressed these changes. Most of the work presented has been unable to meet these challenges and is inapplicable in most of the development environments. It has been observed that industry factors, issues and problems have not been properly addressed in most of the works. Only a limited research works have properly addressed the global changes occurring and their effects on the IT industry. Most of the works are based on assumptions which are not in accordance with the real issues and problems of the IT industry. A gap between software engineering research community and industry practitioners is quite prominent. The more realistic work fulfilling the industry requirements could not have been produced due to this gap.

Moreover, the real processes, practices, approaches and problem solving techniques used by the industry rarely have been published by the industry practitioners due to which software engineering research community has been unable to focus on the real industry problems and issues. Due to these factors, the problems of the IT industry particularly small and medium sized software development companies still exist. Despite the good quality process models and frameworks of software development process design, development, change, reusability, tailoring, improvement and management, the project success rate could not have been increased. Rarely any research work has promoted reusability of software development process through process tailoring technique. The area of process tailoring lacks good quality systematic approaches in general and for small and medium size software development companies in particular.

Therefore, there is a need to produce realistic research works that fulfill the needs of the IT industry particularly small and medium sized software development companies and resolves the issues faced by the projects. Software process tailoring is needed to be adopted as a regular part of the software development process. More contribution of the software engineering research community is required in this area. Moreover, the academic-industry collaboration is also required to produce realistic models and framework to address the problems of small and medium sized software development companies which are exceeding in number.

This research has focused on these aspects and has addressed the issues of small and medium sized software development companies through software process tailoring approach. The real industry projects have been thoroughly investigated and analyzed to provide a systematic approach of process tailoring for small and medium sized software development companies.

CHAPTER 3

RESEARCH METHODOLOGY

Overview

This chapter presents the research methodology that has been followed to complete this research study. It explains the research phenomena in software engineering research work and highlights the characteristics of various qualitative and quantitative research methodologies. Details on qualitative and quantitative research methodologies and selection of suitable research methods for present study in this thesis have been presented. Further, this chapter presents the overall research design of the present study.

3.1. Research Design

The overall research design is comprised of two main parts of this study such as formulation of the framework and its validation. In the first part, framework for process tailoring has been formulated which has been validated in the second part of the study. Different qualitative and quantitative methodologies have been used in both parts, details on which have been presented in the following sections of this chapter.

Based on the comprehensive literature review as presented in chapter 2, the research problem has been formulated and research objectives have been defined as presented in chapter 1. To address the research problem a preliminary framework has been formulated which has been further validated.

The findings and results of the study have been interpreted and their comparative analysis has been performed. Finally, a process tailoring schema has been developed and a framework for process tailoring has been presented as an outcome of the research. Fig. 3.1 shows the overall design of research methodology followed in the present study.

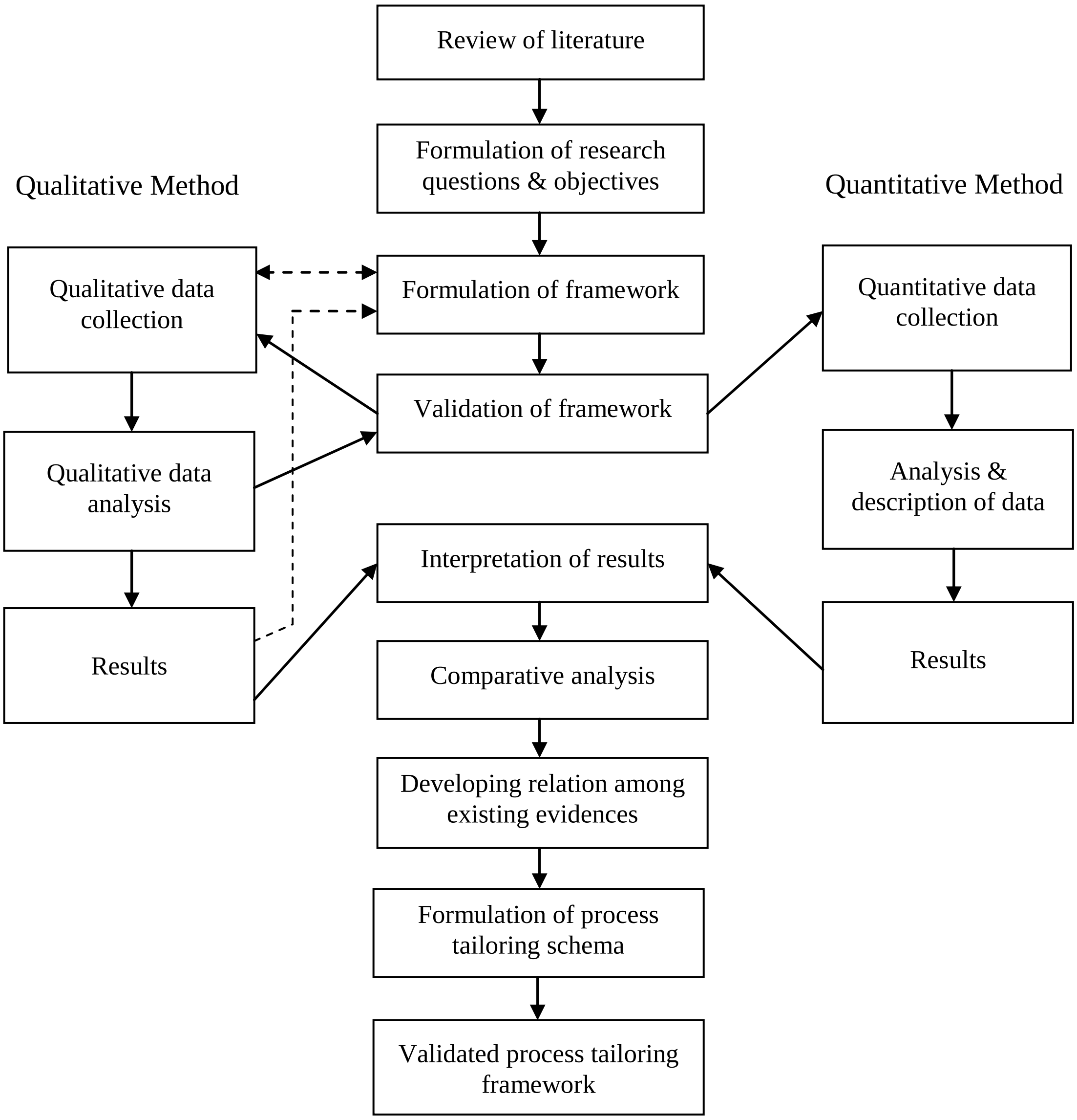

Fig. 3.1 Research Methodology

3.2. Software Engineering Research

The sciences have been classified into various disciplines such as social sciences, biological sciences, physical sciences, chemical, and engineering sciences. This classification of sciences has been made based on different criteria [144]. Two commonly accepted classifications of sciences are formal sciences such as mathematics, and empirical sciences like biological and social sciences such that formal sciences follow deductive research methodologies and inductive or empirical methodologies are followed by the empirical sciences [100]. The deductive or empirical methods that deal with the study of natural phenomena or objects refer to the quantitative research methodologies, and to study the theories, and social and cultural phenomena qualitative methods are used [100], [145].

The classified science disciplines have well defined and suitable research methodologies. These methodologies provide detailed guidelines to these social, basic or formal sciences on data collection, verification and validation procedures. Though software engineering is based on engineering principles, but it does not directly belong to such scientific disciplines. Therefore, no well defined guideline is available on software engineering research methodologies. In this regard, a number of issues have been reported in [94], [97], [98], [146].

In practice, software engineering research follows both the deductive and empirical techniques as defined by social, basic and engineering sciences. Table 3.1 shows the types of software engineering sciences, their character and research methods used [100].

Table 3.1 Research Methods in Software Engineering

	Science	Object of Study	Character	Methods
Type A	Software Engineering Science	Construction of new objects	Engineering	Qualitative Creative
Type B	Software Science	Objects constructed	Empirical	Quantitative
Type C	Science of Information Systems	Implementation and uses of objects constructed	Cultural and social	Qualitative

Based on different objects of study, the software engineering research uses either qualitative or quantitative approaches which have been discussed in the following section.

3.3. Research Methodologies

The words research method and methodology have been used interchangeably in the software engineering research works and existing literature. According to the common belief, both have been used in the same context by the researchers. Research method refers to the technique to collect data such as questionnaire whereas research methodology is an overall set of research activities [147].

As discussed earlier, research methodologies have been divided into two main categories such as qualitative methodologies and quantitative methodologies. Both qualitative and quantitative methodologies have their own methods, techniques and rules. Various research paradigms in the context of qualitative and quantitative research methodologies have been presented by a number of researchers such as [148], [149]. Generally, software engineering researchers always try to find better methods of software development to meet the research objectives such as quality, cost and on time delivery of the product, whereas the primary objective of the research, type of research question, proposed solution and expected results are the main factors that contribute in selection of the suitable methodology [99].

The most commonly used research methodologies in software engineering research are surveys [150], controlled experiment[150], [151], action research [150], [152], [153], [154], [155], case study [156], and ethnographic studies [157]. The characteristics of these methodologies have been summarized in Table 3.2 [156].

Table 3.2 Overview of Research Methodology Characteristics

Methodology	Primary Objective	Primary Data	Design
Survey	Descriptive	Quantitative	Fixed
Case Study	Exploratory	Qualitative	Flexible
Experiment	Explanatory	Quantitative	Fixed
Action Research	Improving	Qualitative	Flexible

The characteristics of methodologies as presented in Table 3.2 shows that case study methodology suits to many kinds of software engineering research works as it is flexible, and explores and analyzes the contemporary phenomena which is very difficult to study separately [156]. On the other hand action research methodology is followed in the cases where improvement is the primary objective. Surveys and experimentation are quantitative methodologies and their design is not flexible.

3.4. Selection of Research Methodology

In the present study, the triangulation approach has been used in which the qualitative research methodologies as the main research method and quantitative research methods as supporting method have been followed [156], [158]. The qualitative research methodologies have been selected in order to study and analyze the contemporary phenomena that occur in the real software development projects in their natural context. Furthermore, appropriate methods, understanding perspective of the participants, direct interaction and communication and variety of approaches in qualitative research [159] were the main factors in selection of qualitative methodology. The quantitative method has been used in order to support the results of qualitative research methods. Finding support to results from different source confirms the validity of the results [160].

The present study has been completed in two parts in which firstly, a framework has been formulated and secondly, its validation has been performed. The methodologies used both for framework formulation and validations are shown in Fig. 3.2.

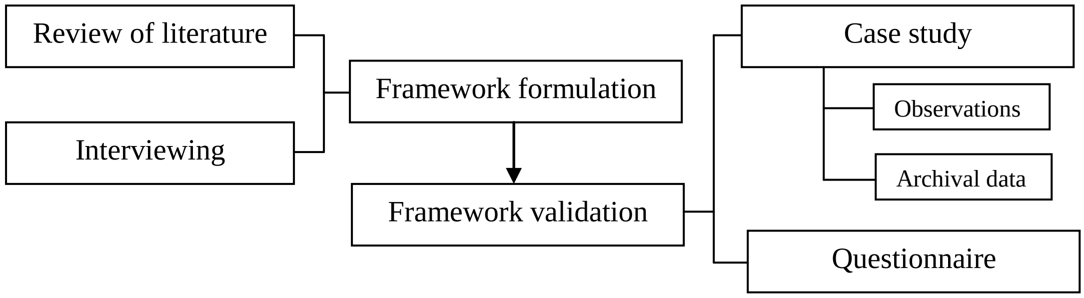

Fig. 3.2 Selected Research Methodologies

3.4.1. Selected Research Method for Framework Formulation

In the present study, the framework formulation has been made through the study of selected relevant research literature [114]. Interviews have been conducted to make further investigations on the components identified from the literature and verifying the correctness of the formulated framework.

3.4.1.1. Review of Literature

The primary study to formulate the framework has been made through review of literature [114] that has been consulted during the research work as presented in chapter 1 and discussed in detail in chapter 2. The detailed procedure of framework formulation is presented in chapter 4.

3.4.1.2. Interviews

The interviewing method has been used at the framework development level. The main categories and sub-categories of the framework have been identified and verified through this approach. Interviews have been divided into three types such as structured, semi-structured and unstructured [150], [156]. In the current research work, the structured interviewing approach has been used which is most likely a questionnaire. Expert interviews have been conducted through email from the people who are expert in the areas of software development processes [159]. These experts belonged to the academic researchers and industry practitioners working in that specific domain. The expert interviews have been conducted to verify the correctness of framework components which in the present research work have been designed through literature review. This type of interview helps to reduce the time of data collection and analysis as well as provides practical knowledge [161].

The interviews have been conducted through email due to unavailability of the experts for face to face interviews. This approach is applicable in situations where respondents are quite busy or lives in other city, country or region as well as detailed investigation is not required [162].

Interviews have been conducted with nine experts working in the domain of software development processes. According to [162], a single interview is enough to get required information, facilitate the research and deciding research direction.

3.4.2. Selected Research Method for Framework Validation

Various validation approaches have been reported in the literature to validate different theories, models, and frameworks. The valid projects and valid data have also been considered as proof of validity of work which has been provided in the form of the certificate attached herein Appendix A.

In the present study framework validation has been performed through case study followed by a survey questionnaire. It is important to maintain the validity of the case study since its beginning. Other ways that have been used in this study to maintain the validity of the case study findings are the detailed case study protocol, data and results reviewed by the case study subjects and sufficient time spent with each case [159], [156], [150].

To perform case study observations and analysis, observations and archival data approaches have been used. Archival data helps in obtaining rigorous information about the previous releases of the project such as previous processes, scenarios, issues and problems, and measures. The indirect methods such as audio, video recordings are more suitable in the situations where direct interaction is not possible and subjects are inaccessible. In this case, the information may be ambiguous and misleading that may affect the outcome of the study.

3.4.2.1. Case Study

Software engineering research is based on the contemporary phenomena that exist in their natural context in real software development projects. The research in software engineering in mostly related with the software development processes, models, standards, people or stakeholders, social factors, requirement management, organization, environment, project planning and management, and financial aspects. All these aspects of software engineering research depend on various situations and scenarios that exist only in real projects carried out in the software development companies. Simulation and controlled experiments are unable to take into account all possible situations or scenarios under certain restrictions. The response and behavior of these aspects is highly variable by projects depending on different factors that exist in real environments only.

According to [156], the case studies are normally the choice in software engineering research because software engineering research is multidisciplinary where scenarios and phenomenon are studied in their natural context such that boundary between scenarios or phenomenon, and their context is not very clear. In case of experimentation, there are many factors that affect the results of the study when these are repeated or replicated [163]. Case studies are flexible and provide a deep and close interaction between the objects of study and their environment where they exist [156]. As believed by the industry practitioners, the software engineering research needs to address and solve the issues related to the management, organization, and people in the order of their existence and progress of the project [160]. A case study generates research description of contemporary phenomena [164]. The case study helps researchers in understanding research phenomena with more details [165]. Above all, the case study has an added advantage of flexibility over other methodologies.

Such advantages of the case study over simulation and controlled experiments have made it the preferred choice of the software engineering researchers. Therefore, case study has been selected as the research method to validate the framework in the present study because of the support of available methods to study the contemporary phenomenon and scenarios in real projects running in the small and medium size software development companies.

a) Case Study Design

In order to conduct the case study, a complete case study protocol has been defined for making design of the study and deciding field procedures to be carried out such as data collection and analysis methods [156], [160]. A case study protocol describes the design of the case study. As shown in Fig. 3.3, the case study design in the present study has been adapted from [164].

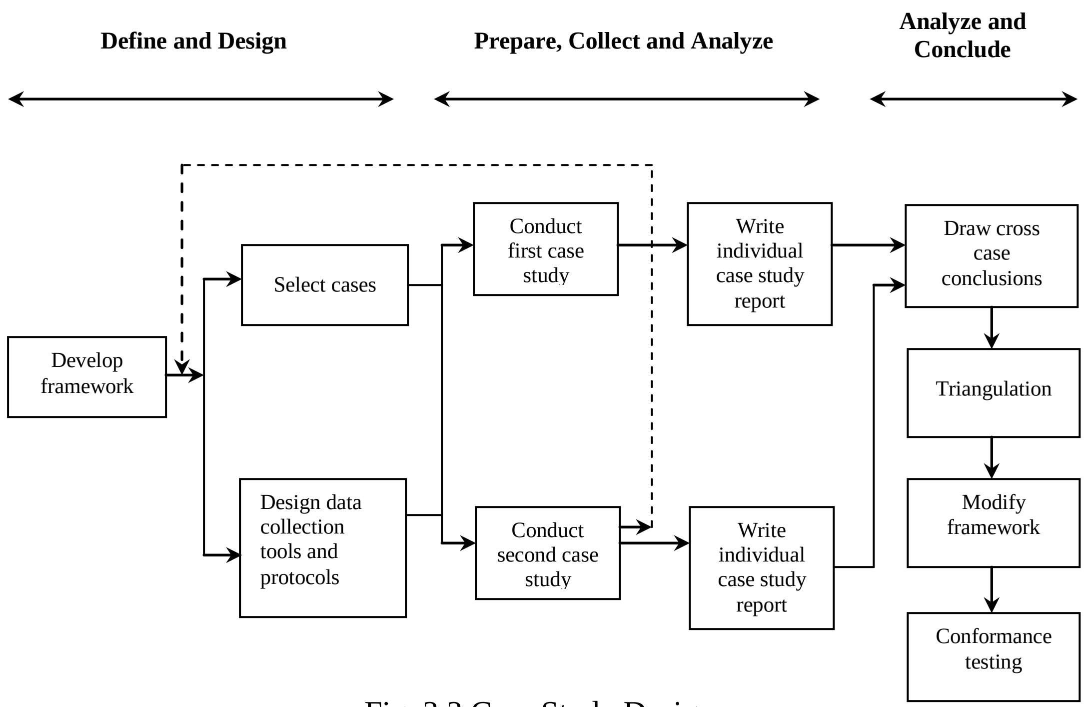

Fig. 3.3 Case Study Design

b) Objective of Case Study

The objective of the case study is to explore and explain [150] the real scenarios that exist in real software development projects in small and medium size software development companies to validate the framework in order to address the research questions presented in section 1.6 in chapter 1.

The case study has been completed in two iterations, therefore it is expected that research questions may evolve and be more focused and narrowed down after both iterations [166].

c) Defining a Case

According to the definition of [164], case is "contemporary phenomenon in its real-life context". In software engineering research, the case may be a software project, project team, individual member, process, product, organization, event or tools and technologies [156]. Case studies have been divided into two main types such as holistic case studies and embedded case studies [164]. In holistic case studies, the case is analyzed as a whole single unit whereas, multiple cases are studied in embedded type case studies and each case has multiple units of analysis for example two different projects of two different domains.

Based on the design, characteristics and suitability to the research objectives, the embedded type case study has been selected in this study as shown in Fig. 3.4 [156], [164].

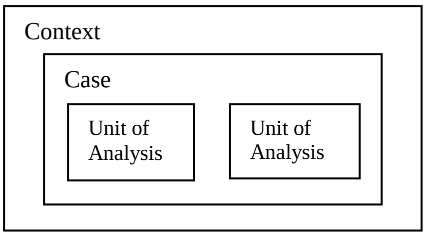

Fig.3.4 Embedded Case Study

According to the definition of embedded type case study, four different projects as four separate cases have been selected based on the following criteria:

1. Running in small and medium sized software development companies.
2. Should be following agile based methodologies.
3. GSD environment such as outsourced projects and offshore clients.
4. Multiple domain projects.
5. Teams might be centralized or distributed.

d) Selection of Case

Four units of analysis such as project, team, client and processes have been selected for each project based on the elements of the framework and research questions. The cases and unit of analysis are selected intentionally in case studies such as sampling is performed in controlled experiments and surveys [156]. However cases and units of analysis can be selected based on the comparison factors, and availability [167] as in the case of most of the experiments [168].

It is important to note that the names of the projects have been changed due to the company's privacy policy to 'Project A', 'Project B', 'Project C', and 'Project D' as well as information about the clients, correspondence and communication with the client and company policies have been kept confidential and have not been disclosed. The guidelines that have been used to implement the framework in case study projects have been given in Appendix L. The same guidelines have also been followed being the participant observer during the case study. In order to confirm the validity of projects, data, and case studies, a certificate of originality and completion of case study has been provided by the company as shown in Appendix A.

The selection of the company for performing framework validation case study has been made based on the criteria similar to the project selection criteria as described in section 3.4.2.1, sub-section 'c' of this chapter. However, emphasis has been given to that company should be working on outsourced projects, practicing agile based methodologies, and size of the company.

e) Background of Company

The selected company has two branches and one parent company operating from USA. Each branch of the company was working on different domain projects. All the projects were outsourced from USA. The company was working on different types of projects such as network security protocols, web based payments, chatting applications, social networking and its various web based applications, gaming applications and one of its software product for which it has various clients all over the world. The company started its business as a small company which in coming months grew enormously and successfully emerged as a medium sized software development company.

All the clients of the company were offshore, software development teams were centralized but in a few cases distributed development was also being done. In one of the project a team of people located in another country was also working in association with the team of the selected company.

The CEO of the company was a resourceful and innovative entrepreneur whereas, CTO of the company located in USA where he was dealing with company clients and was technically handling the main projects of the team along with their clients. The company had successfully completed many projects of different domains and was also successfully running its one software product for which it has a large international clientage. The company is considered as a successful company in the market that has captured a lot of business as compared to other companies.

f) Details of Selected Projects

The attributes and details of all four projects have been summarized in Table 3.3 and Table 3.4 respectively.

Table 3.3 Project Attributes

Name	Processes Followed	Team Structure	Team Resources Organization	Client location	Type of Development
Project A	Agile based methodologies	Hierarchical	Centralized	Offshore (USA)	Outsourced project, offshore development
Project B	Agile based methodologies	Hierarchical	Centralized	Offshore (USA)	Outsourced project, offshore development
Project C	Agile based methodologies	Hierarchical	Distributed	Offshore (USA)	Outsourced project, offshore development
Project D	Agile based methodologies	Hierarchical	Centralized	Offshore (USA)	Outsourced project, offshore development

Table 3.4 Project Teams Details

Project Name	No. of Team Members (Team size)	Skill level of Team Members	Experience	IT Background
Project A	12	Medium to High	2-5 + years	Graduates
Project B	8	Medium to High	1-3+ years	Graduates
Project C	6	Medium to High	2-3+ years	Graduates
Project D	3	High	1+ years	Graduates

Project ‘A’ was an online application for intelligently parsing the legal financial documents and converting their scanned softcopies into text and saving them into database for providing help to the clients to design their own legal documents. The client was offshore but his technical partner was often used to visit the project team.

Project ‘B’ was a multiple chat client application providing services of most popular chatting messengers like Yahoo, MSN, GTalk, XFire in one messenger. The SecondLife virtual world was the main chatting component of the application. The client was offshore but the chief technical officer (CTO) of the vendor company was handling and dealing with him at his own site.

Project 'C' was an online social networking application. Searching and making friends interested in each other's hobbies, cultures, and languages was the important feature of the project that made it different from other such applications. In addition to the offshore development team, client had also a software development team working in parallel to the offshore team.

Project 'D' was a gaming applications designed for facebook users. It was based on the concept of wars among the gangs. A user had to make his own gang, had to do different activities such as business, robbery to earn money to buy weapons and then to fight with other gangs to win the game. The actual client had a technical manager in the project who was leading the project from client side.

g) Organization of Cases (Projects)

The case study has been performed in two parts. In the first part the project 'A' has been analyzed and in second part the analysis of projects 'B','C' and 'D' has been performed. However, for the purpose of elaboration, explanation and support to the framework, a part of project 'C' has been discussed along with project 'A' to present the analysis of first three components. The projects 'B', 'D' and remaining parts of project 'C' have been discussed in detail in second part of the case study. A part of project 'D' has been discussed only in the analysis part of first three components of the framework in the second part of the case study.

The organization of projects in the discussion of analysis of the case study has been presented in Table 3.5.

Table 3.5 Organization of Cases/Projects

Case Study	Framework Components		Project
I	Client	Client's Composite Structure	A, C
		Client's Interaction Overview	A, C
		Client's Perspective Model	A
	Project	Project States Model	A
	Tailoring	Process Tailoring Schema	A
II	Client	Client's Composite Structure	B, D
		Client's Interaction Overview Model	B, D
		Client's Perspective Model	B, C
	Project	Project States Model	B, C
	Tailoring	Process Tailoring Schema	B, C
Cross case Analysis		Complete Framework	A, B, C, D

The first case study helped in understanding the whole phenomena of case study, which provided practical guidelines to conduct the second case study.

The division of projects in both case studies makes it easy to understand the analysis clearly and easily through relative comparisons. It increases the readability, understandability and consistency of the findings.

h) Observational Elements/Attributes of Each Component

Each component of the framework in both case studies has been analyzed based on similar elements or attributes as shown in Table 3.6. These observational elements or attributes have been identified based on their applicability in completely understanding the respective component. The value of each of these elements or attributes help in understanding the working, applicability, implementation, and role of each component in process tailoring framework.

Table 3.6 Observational Elements of Each Component

Framework Components		**Elements**
Client	Client’s Composite Structure	Project Name, Client/Role, Existence, Number of Persons in the Role, When Role Started, Role Job, Technical Skills.
	Client’s Interaction Overview	Project Name, Interaction Entities (Client, Project Team), Type of Interaction, Frequency of Interaction (No. of Times per day).
	Client’s Perspective Model	Key Area (Scenario), Client’s Satisfaction, Improvement (Sharing Interval(duration), Process/Sub-Process, Documentation).
Project	Project States Meta-model	State/Sub-state, Duration of State, Description, Activities performed during a state, Problem class/Factors, Problem Impact, Project Status.
Tailoring	Process Tailoring Schema	Key Process Areas, Tailoring Activities, Takeoff (Pre-Takeoff, Takeoff), Running (Running, Post-running), Landing (Landing, Post-landing), Hang up (Crawling, Swing, Pre-running).
Examples		The examples in component represent the scenarios that have been used to investigate and analyze each of the framework components. Scenarios for similar components in both case studies are same.

The examples as shown in Table 3.6 and later on presented in both case studies describe the scenarios that have been used to analyze the components in various situations. These examples for same components in both case studies are based on same parameters but are different from other components.

i) Data Collection

Unlike numeric data in quantitative approach, the qualitative data is comprised of words, pictures, audio, visuals and documents [169]. In a case study, there are many sources and techniques of data collection. Data collection methods have been divided into three levels as follows [170]:

i. First Degree: Direct Methods

In this method researcher collects real data in direct contact or interaction with the subject of study. The examples of direct methods include observations, interviews, and surveys.

ii. Second Degree: Indirect Method

In this method researcher collects raw data without directly interacting with the subject of the study. For example use of automatic monitoring tools, and audio video recordings.

iii. Third Degree: Independent Method

In this method researcher perform an independent analysis and collection of data from available documents such as project plans, schedules, requirement management and specification.

j) Selected Method for Data Collection

The following data collection methods have been followed during the case study analysis.

i. Observations

Observations have been made through direct contact and interaction with the project teams as case 2 in capacity of "observing participant" and "normal participant" respectively as shown in Table 3.7 [156], [160].

Table 3.7 Different Approaches of Observations

	High awareness of being observed	**Low awareness of being observed**
High degree of interaction by the researcher	Case 1	Case 2
Low degree of interaction by the researcher	Case 3	Case 4

In case 3 and case 4, the team members of the projects are aware that they are involved in a study which makes them conscious that may affect the observations and results may be biased. Following the case 2, the researcher takes the role of a project manager of the team without their knowledge of being observed which provides neutral and unbiased observations and results.

Observations provide a close and deep understanding of the real scenarios that exist in real projects which is not possible in any other method. Observation method is suitable for case studies where official view of scenarios is different from real view [171].

Particularly, observations have been made in the present case study following case 2 approaches where project teams were not aware of being observed and researcher was the part of the project in capacity of project manager. Data has been collected through think aloud protocol [156], [160], [172], participation in the meetings, project planning, requirements management, project scheduling, and resource planning documents. Furthermore, observations have also been made based on the client's feedback through emails, phone calls, chatting and personal observations [173].

ii. Archival Data

The archival data such as meeting minutes, project scope documents, documents of previously completed work, deliverables, milestones, resources, client's correspondence, and data repositories has been collected through this technique.

The data repositories such as MS Groove, GoogleDocs, and Sharepoint portal server have been used to collect the past data. The configuration management tools in archival data collection have significant importance due to the availability of previous versions of the documents and code base [156].

It is noted that in order to avoid the possibility of missing information, data collection method combined with other technique gives better results and provides a deep analysis. Therefore, in this case study both observation and archival data collection methods have been used together.

k) Data Analysis

During the case study, the results and conclusions have been derived from data analysis by associating evidences, order and sequence of information, discussions, explanation and justifications. The data is presented in a way that during reading the case study, the results and conclusions should be derived in the following sequence automatically [164]. In qualitative research, data collection and analysis are carried out in parallel and conclusions are derived.

Two data analysis techniques such as "cross-case analysis" and "within-case analysis" [5], [160], [172] have been applied on the collected data in this study. Various techniques of cross case analysis such as dividing a case into two groups based on their attributes, and analyzing the pair of cases have been presented in [174]. Thus, similarities, differences and variations in each case and pair are found then.

During the cross-case analysis, all the four cases (projects) have been compared with each other based on their similarities and differences in units of analysis and their respective attributes and findings. During with-in case analysis, each single case and its respective units of analysis such as team, process, project and client have been analyzed individually within a case and unit and as a whole case.

l) Converting Qualitative Data into Quantitative Through Coding

The open coding technique as presented by [159] ,[160] is used in the present study to convert the qualitative data into quantitative data in order to perform analysis on the values. Numbers have been assigned to the descriptive data as well as labeling of string values has been made to extract information from the data. For example, Ali Ahmad and Hassan are senior software engineers working in team 'Blue'. Converting this textual information into quantitative data will be as follows:

Name of team members: Ali, Ahmad, Hassan

Count: 3

Size of Blue Team = 3

To convert qualitative data into quantitative values and categorizing it to define variables for statistical analysis and comparisons, numerous coding techniques have presented in the software engineering research works [159], [175], [176]. The open coding technique as straightforward and direct extraction of data has been applied on data obtained from both observations and archives to convert into numeric values and labeling text strings. The transformation of qualitative data into quantitative data does not affect the subjectivity or objectivity of data [160]. However, after coding, subjective information should also be verified in order to minimize the chances of information lost during this process.

3.4.2.2. Questionnaire

In order to further validates the findings of the case study, the questionnaire method [159], [160] has been used to improve the validity of the results [156]. In this regard, a survey of various small and medium sized software development companies has been conducted. The sample questionnaire has been provided in Appendix I.

The questionnaire method facilitates in terms of collection and analysis of large amount of data quickly and supports the case study results by eliminating bias factor and improving their validity. The results of both qualitative case study and quantitative questionnaire have been used in investigating the various changes and improvements in the framework.

3.5. Conformance Testing

The finalized and validated framework has been aligned with the principles of agile manifesto [46] in order to confirm that the framework is agile based and the practices that it has derived are also based on agile principles. During conformance testing, various components of the framework and findings of the case study have been related with the principles of agile manifesto. The requirement of conformance testing is considered fulfilled when no defect is found [116].

3.6. Reporting Case Study

The report of the case study is the ultimate criteria to judge the quality of the work [156]. Normally case study reports are written but they can be presented in the form of audio, video or photographs [158]. The format and structure of the report of the case study is very important to understand the sequence, flow and findings of the case study. According to [164], the report audience may be different persons like researchers, industry professionals, and policy makers. Therefore, the report contents should be equally understandable by any audience. Anyhow, the reports are mostly prepared focusing on the researchers. The structure and format of the report is prepared based on the target audience.

3.6.1. Reporting Structure

A basic structure to report the case study as proposed by [177] was further modified by [178] and has been presented in [156]. In the present study, the case study has been reported following the linear-analytic and chronological structures as defined by [164] and shown in Fig. 3.5 [158].

LINEAR-ANALYTIC
Statement of Problem
Literature Review
Methodology
Findings/Analysis
Conclusions

COMPARATIVE
Case study 1: description A
Case study 1: description B

Fig. 3.5 Case Study Reporting Structures [158]

Both for single or multiple case studies, the linear-analytic structure is used when case study report is written for academic audience because its format is easily understandable by the academicians. On the other hand comparative structure is used when same case study is repeated two or more times and results are compared with each other [158]. In the present study, the case study has been presented following the both formats due to their suitability and support to the present case study.

3.6.2. Writing of Case Study Report

The contents of the case study can be written in different ways. However, [158] describes four structures to write the contents of the case study report as shown in Table 3.8.

Table 3.8 Four Written Forms of Case Study [158]

Type of Case Study	Report Structures		
Single case study	Case study description and analysis		
Multiple case study	Cross-case analysis and results	Appendix: Narrative Case Study 1 Narrative Case study *n*	
Multiple case study without narrative	Case study 1	Question 1 Question2	Answer Answer
	Case study 2	Question1 Question 2	Answer Answer
Multiple case study: integrated	Cross-case issue 1 – data and analysis from all cases Cross-case issue 2 – data and analysis from all cases		

In the present study, single and multiple case study types have been followed to present the contents of the case study. The body of the case studies consists of narrative descriptions consisting of analysis and supporting data of cross-cases whereas some contents are written in the appendices [158]. This approach avoids case study report from being bulky and presents it in the more readable form.

3.7. Summary

The present study has been completed using both qualitative and quantitative methodologies. The triangulation approach has been adopted in which both qualitative and quantitative methods have been combined. Finding support to the results from another source of data has been considered as more authentic for the validity and reliability of results. This is the main reason behind following the triangulation approach in present study. The study has been completed in two parts such as formulation of framework and validation of framework.

Formulation of the framework has been made through the primary study of literature and structured interviews approach, whereas the validation of the framework has been performed through case study of real projects and questionnaire method. Case study of the real projects comprises of the major portion of the present study. The research methodology adopted in this thesis has provided strong basis to complete the present study.

CHAPTER 4

PROCESS TAILORING FRAMEWORK

Overview

This chapter presents the detailed procedure of formulation of framework. It explains the primary study that has been conducted to identify the various components of the framework which have been further verified through structured interviewing approach. It explains the relationship between key components of framework and finally integrates them to formulate a framework.

4.1. Introduction

A software process meta-model comprises of phases, activities, artifacts and roles as shown in Fig. 4.1 [28]. The process meta-model truly reflects the structure of traditional software development approaches such as waterfall model. However, agile methodologies being development driven approaches have not well defined phases of software development and focus on software development activities [10], [27]. Therefore, following the process meta-model, the process tailoring framework tailors the activities of a software development process and derives the set of lightweight software development practices.

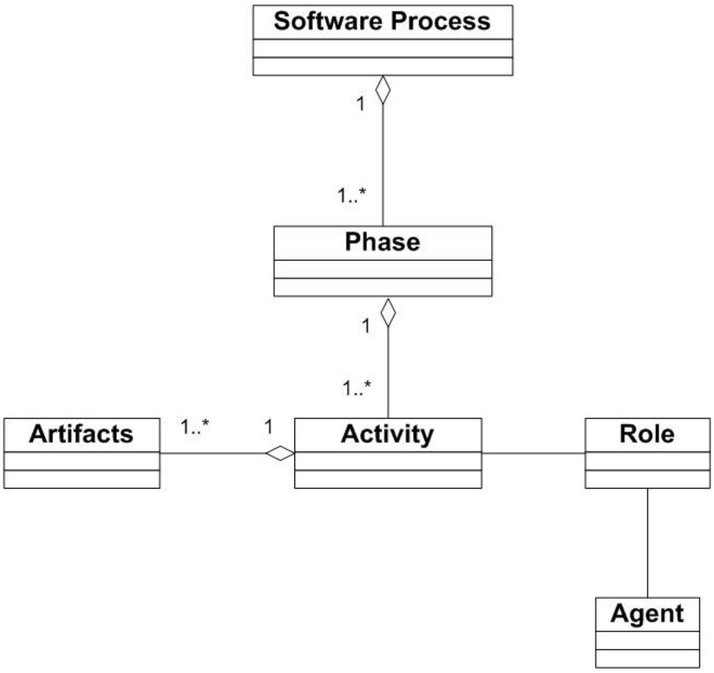

Fig. 4.1 Software Process Meta-model [28]

Existing research works have presented a number of factors related to the organization, project, team and stakeholders that affect the software development processes [5], [179], [180]. The proposed framework tailors the software development processes based on the agile based methodologies.

In agile based methodologies, client has been considered as the key role player in software development projects [10] therefore, understanding and maintaining the client's perspective throughout the project lifecycle is very important. In the proposed framework, the client and client's perspective have been identified as foundation elements on which the key process areas with which client is mostly concerned during the software development in small and medium sized software development companies have been identified.

The effect of client's perspective, problems or risks faced by the projects [79], and other factors such as team performance on behavior, response and progress of the project have been analyzed and project states have been derived. Project states are mainly based on the software project lifecycle phases and activities as discussed in the next sections of this chapter. A state refers to the particular condition of a project with respect to its behavior and response to various factors. The risks associated with software development projects as reported in the research literature have been categorized into main groups or classes with respect to their impact on the progress of the project and performance of the team members. These risks are the main elements in analyzing the project states.

Integrating these components, the process tailoring framework has been formulated which provides a schema to tailor the activities of the agile based key processes for a particular state of the project. The framework suggests which and when process tailoring activities should be performed, and how much tailoring is required during some specific state of the project.

The framework is based on the realistic critical success factors associated with the agile based environments of small and medium sized software development companies. Tailoring the software process and its activities during project execution throughout the project lifecycle or during any particular state is the novel approach of the framework within the context of small and medium sized software development companies following agile based methodologies.

4.2. Framework Development

The framework has been developed through primary study of literature review [114], and structured interviewing approach as described in chapter 3. The primary studies to develop the framework have been made through the study of existing research literature. The findings are then derived and verified through structured interviewing and framework has been formulated.

The existing research literatures specific to the research domain have been consulted during the primary study findings as followed in [6]. Deductive approach has been used during literature study. In deductive approach, the literature review provides a source of information about the research and also helps in generating hypothesis [158], which in present study has been used for the development of the framework.

Search has been made from the most famous and recognized electronic resources such as ACM, IEEE, Springer, and ScienceDirect. The search queries such as “software process tailoring”, “agile processes”, “problems in agile processes”, “software process improvement”, “project outsourcing”, “risks in software projects”, “project management issues” and “software development issues” have been used to find the required research papers. The papers not relevant to the current study were excluded after reviewing their abstract, methodology and findings. Following this inclusion and exclusion criteria, only the most relevant papers have been selected and reviewed. Whereas, [6], [5], [28], [35], and [46] are the key papers which provide foundations for this research work and framework development. The detailed review of research papers has already been presented in chapter 2. Only final findings have been discussed here to derive the components, and formulation of the proposed framework.

4.2.1. Critical Success Factors in Agile Based Projects

Software process tailoring is mostly performed at different levels such as organizational level and project level [181], [182], [183]. Process tailoring at organizational level is performed to meet the specific organizational needs. However, the need of process tailoring at project level has also been realized [6]. Therefore, software development project has been selected as the fundamental level of process tailoring in the proposed process tailoring framework.

It has already been discussed that small and medium sized software development companies follow agile based methodologies for which reasons have also been explained in chapter 1. According to the principles of agile manifesto, the project client has been considered as the key role player in agile based methodologies [46] and has also been recognized as third main player in context of GSD as shown in Fig. 4.2 [7].

Fig. 4.2 Three Main Players in GSD [7]

The project client has also been considered as a critical success factors in software development projects by [184], [185], [186], [187], [188], [189] and many other research works as reported in [81]. In software developments projects that follow agile based methodologies, the role of client has been found very critical being directly related with the project development practices [10]. Being most critical success factor, the client has been selected as the main element of projects following agile based methodologies as shown in Fig. 4.3.

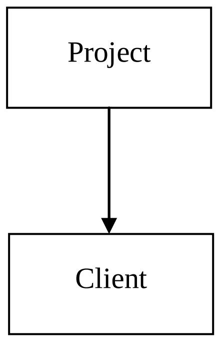

Fig. 4.3 Client as Critical Success Factor in Project

4.2.2. Client's Perspective

The client being the critical success factor in software development projects following agile methodologies holds the key decisive position in such projects. Therefore, the satisfaction of client is also very important for the progress and success of the project. The satisfaction of client is associated with fulfillment of his requirements and expectations such that activities in an organization should be centered around the client's needs [190]. The software development process must also be client oriented to fulfill his needs and to keep him satisfied continuously at permanent basis [92].

Empirical results show that satisfaction has a strong, positive and direct relationship with processes and outcome of the project following agile methodologies [27]. The satisfaction of client is the ultimate objective of principles of agile manifesto as has been emphasized in the first principle of agile manifesto [46], [26].

The satisfaction of client is significantly important for the software development companies due to its effect on market reputation, clientage and future business of the company [191]. The satisfaction of client is associated with meeting the client's expectations from project process and its outcome [31]. Achieving client's requirements and satisfaction of client are the important factors in software process improvement as well [111].

The satisfaction of client is achieved when client's requirements are fulfilled. This is achieved when client's perspective is fully understood and maintained throughout the project lifecycle. The client's perspective means understanding the client and his/her view point in order to meet his/her expectations, requirements and priorities in the project. The client's perspective has been referred to as putting yourself in the client's shoes [35]. The client's perspective has been considered as an important tool for business managers but this valuable approach has rarely been included in software development projects following agile methods in which client is also the most critical success factor. According to [35] understanding client's perspective needs complete understanding of:

a. What does client want?
b. What client has in mind?
c. How the client feels?
d. How the client thinks?

Furthermore, an exercise guideline has also been provided to understand client's perspective and practice it throughout the project such as [35]:

a. Consider yourself in the client's environment (physical environment).
b. Consider doing what client does (performing same tasks).

c. Consider yourself being in the mind of client (a & b).
d. Visualize the beliefs and values of the client (expectations, priorities, assumptions).

For achieving the client's satisfaction it is necessary to look at the situations from client's perspective throughout the project lifecycle [35]. It shows that the client's perspective is the most important factor of client in software development projects following agile based methodologies and has direct relationship with client, satisfaction of client and project success as shown in Fig. 4.4.

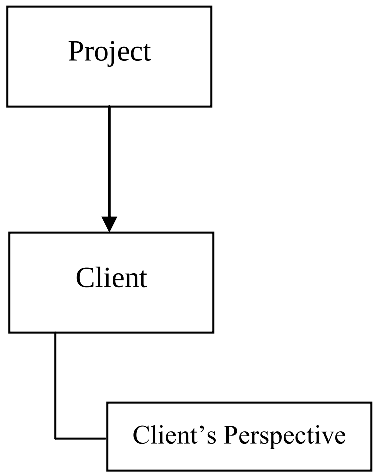

Fig. 4.4 Client's Perspective Factor of Client

4.2.3. Key Processes in Projects Following Agile Methods

Agile based methodologies mainly focus on a few key processes or phases of the software development project. On the other hand, client in small and medium sized software development companies following agile based methodologies are also mostly concerned with key processes such as team resources management, requirement gathering and tracking processes, communication process, and tasks allocation process. These key processes are the backbone of agile based methodologies and offshore development projects and are considered important for project success [44], [5], [81], [82], [43], [62].

Out of a number of challenges faced by the projects [5], the resource challenges, communication challenges, and requirement management challenges have been adapted as the challenging software development process areas in small and medium sized software development companies following agile based methodologies. The selection of these three key process areas is made on the basis of emphasis given to these in twelve principles of agile manifesto [26], [46]. The open coding technique [160], [159] has been used to extract the key elements from 2nd, 4th, 5th, 6th, 11th and 12th principles of agile manifesto as presented in Appendix B.

The main objective to refine these processes is the satisfaction of client which is the ultimate focus of the first principle of agile manifesto as discussed in section 4.2.2.

The communication process in agile methodologies is informal [10], therefore, proper attention is required to manage this important process. Clear requirements and specifications, effective communication, and project resources have also been identified important for project success and their frequency of reporting in literature is 60.5%, 46.5% and 25.6% respectively which are relatively higher than other factors [81]. Similarly, it has been found that communication and coordination between client and project team affects the performance and progress of the projects and emphasis has been given on further investigations in this important process area [44].

Therefore, as extracted from the principles of agile manifesto as shown in appendix B, adapted from [5] and identified from other literatures following are the key process areas in agile based methodologies:

a) Resource management
b) Communication (interaction & coordination) and
c) Requirement management
    i. Requirement gathering & tracking
    ii. Tasks allocation

Tasks allocation is an important part of team resources management process but more importantly it is directly affected by requirement management in which tasks are assigned to the team members. Therefore, it has been sub-categorized under requirement management process.

These key processes directly affect the progress of the project in agile environments. As much as these processes will be good and strong, the more and more client will be satisfied, and project's progress will be smooth in small and medium sized software development companies. These key processes in agile based methodologies significantly affect the client's expectations and requirements and achievement of client's perspective. Therefore, these processes are the main factors associated with achieving client's perspective as shown in Fig. 4.5.

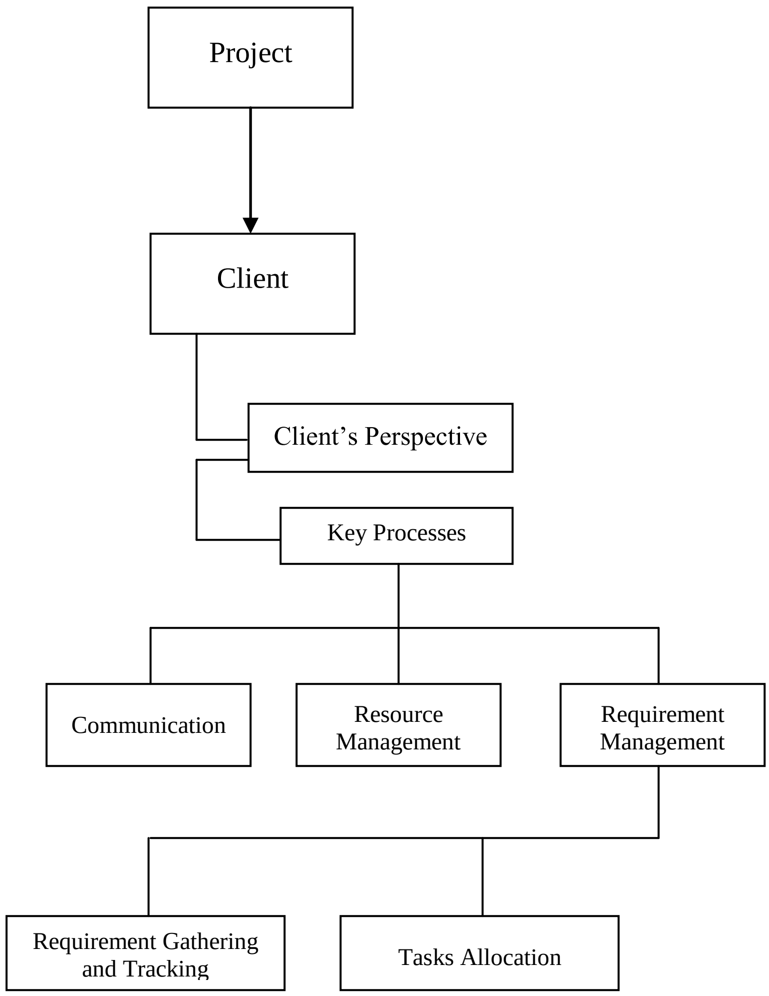

Fig. 4.5 Key Process Areas

The framework tailors these key processes according to the requirements of the project and client. The framework recommends that how to tailor these processes, their sub-processes and activities and what tailored practices should be performed in this regard.

As client's satisfaction is directly associated with these processes, the process tailoring of these three key processes is performed. The sub-processes, practices and activities that client wants to be followed have been adopted and tailored according to the expectations and requirements of the client. The good quality of the tailored software development processes ensure project stability and quality software product which ultimately leads to the client's satisfaction, trust and project success.

4.2.4. Behavior of Software Development Projects

As described in section 4.2.1 that software process tailoring is performed at organizational and project levels. Most of the research works have performed process tailoring to meet the organizational goals. However, it is required to refine the process at project level through software process tailoring [6]. The proposed process tailoring approach in this thesis performs process tailoring at project level. In this regard, it is important to understand the behavior of software development projects.

The behavior a software project depends on a number of factors such as performance of the team members, risks faced by the projects and response of the project to these risks. Risks are considered as critical success factors to manage a software development project [80]. They are the main determinants of the behavior of the software development projects and their execution flow. Risks affect the performance of the team members, progress of the project and project success. On the other hand, project success is directly related to the satisfaction of the client as discussed earlier. It shows that risks in software projects may adversely affect the satisfaction of the client which ultimately affects the success of the project. The relationship between risks, client's satisfaction and project success is ternary as shown in Fig. 4.6.

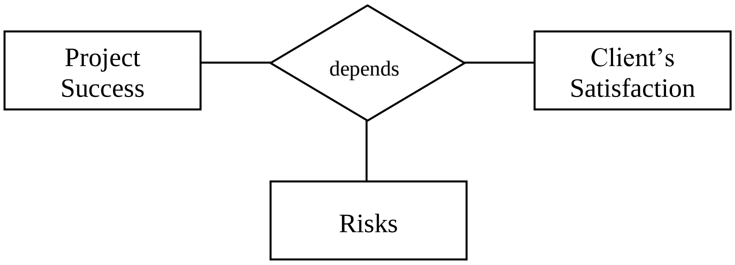

Fig. 4.6 Behavioral Aspect of Project

Identification of problem situations and their characterization is challenging for successful process tailoring [112]. Therefore, problems or risks as important behavioral determining factors of projects are discussed here. Issues and problems related to all other factors are also regarded as risks.

4.2.4.1. Risks in Software Development Projects

Major risks which have been considered responsible for project failure have been described as [192]:

- Improper requirement management.
- Poor requirement specifications.
- Weak project management and software development methodologies.
- Poor project planning, scheduling and project status reporting.
- Weak risk management.
- Mishandling of client.
- Improper and inefficient utilization of team resources.

Individually or collectively these and other kinds of risks affect adversely on the sustainability of software development processes, progress of the project and satisfaction of the client. The impact of these risks or problems is more severe in projects which are outsourced to small and medium sized software development companies following agile based methodologies such that risk management becomes very important for the smooth progress and success of the project.

Therefore, the role of project manager in risk management becomes more important. The various aspects of project management group have been highlighted in [193] and discussed further in other research works such as [194], [195], [196], [197], [198], [199]. The success of the software development project depends on the decision making abilities and expertise of the project manager. However, completion of the software development project is not the criteria of project success [200]. To handle the risks, various risk management strategies have been presented in a number of software engineering research works such as [201], [202], [203].

In this research, the existing risks as reported in the literatures have been grouped into three main classes with respect to their type and effect on the software development projects. These classes are then used to study the behavior of the project.

4.2.4.2. Problem/Risks Classification

There have been reported a large number of problems faced by the software development projects. The risks or problems as reported in existing research works such as [201], [202], [80], [90] [192], [79], [29] have been grouped into three main classes in this thesis. The classification is made on the basis of their impact on the performance of the software development team and progress of the project as follows:

i. Performance Minor
ii. Management Critical
iii. Progress Limiter

The effect of each problem class on the performance of the team and progress of the project is different which is based on the type of the problems it contains. A few examples of the problems that belong to each problem class have been summarized in Table 4.1. To understand the risk classes, the description of each risk class is given in the following sections.

Table 4.1 Problem Classification

Performance Minor	Management Critical	Progress Limiter
a) Incompetent or weak team resource(s) b) Poor problem solving approach c) Non cooperation d) Lack of coordination e) Professional jealousy f) Social and financial matters g) Lack of confidence h) Misguidance i) Lack of professionalism	a) Poor requirement management b) Weak project plans c) Unrealistic scheduling and deadlines and milestones d) Improper tasks allocation e) Unawareness of resources, their competencies and skill set f) Unusual pressure and work load on resources g) Stressed environment h) Strictness i) Weak tracking, monitoring and performance processes j) Communication gap k) Lack of coordination l) Weak project manager and control.	Unhandled and mismanaged performance minor and management critical problems which becomes of swear type.

Performance Minor problems are associated with the performance of the software development team. These types of problems do not affect the project greatly and can be easily handled. Problems such as incompetent team member, weak problem solving approach, non-cooperation with other team members, and social and financial problems normally affect the individual and team performance. Such problems can be handled through good project management practices.

Management Critical class problems are associated with the processes and practices of project management such as poor requirement management processes, weak project planning and scheduling, unrealistic deadlines and improper allocation of resources and tasks [29], and unsatisfied clients. Weak project management directly affects the software development and project management processes that leads to awry situations.

If not handled effectively, performance minor class problems may convert into management critical problems. Management critical problems can be handled and avoided with good project management practices. Experienced and skilled project managers by considering the factors such as learning from the previous projects, past experience with the same client can remove the weaknesses in project management practices.

Likewise, a lazy approach of a project manager towards the good project management practices to resolve the issues, may convert management critical problems into progress limiter problems.

Progress Limiter class problems are the mismanaged and unhandled severe problems. These problems are actually the severe threat to the progress of the project and its success in terms of high intensity and frequency. If remained unhandled, these problems may lead to the project failure.

These three risk classes occur throughout the project lifecycle as shown in Fig. 4.7.

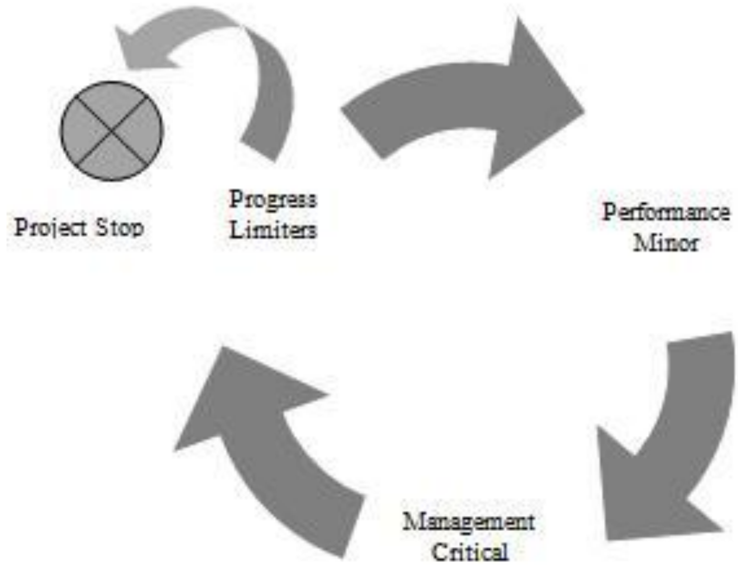

Fig. 4.7 Software Project Risks

Each risk class may lead to the other risk class; however risks in a software development project may never be avoided and ended. Even a small modification in any process or activity may cause risks at any level. Progress limiter class has problems of most severe nature that may terminate a project without completion as shown in Fig. 4.7 which shows that risks are handled and new risks are born that keeps this cycle continued until the project is stopped either completed or incomplete.

These risk classes are considered as main determinants of the behavior of a software development project which is the characteristic element of a software development project as shown in Fig. 4.8.

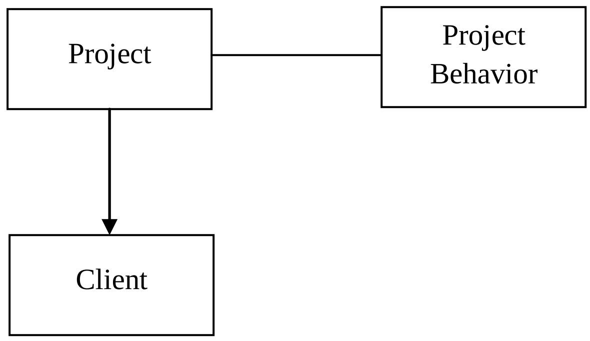

Fig.4.8. Project Behavior Characteristic Element

The following section investigates the project lifecycle elements and derives the project states as behavioral elements of the project. The project states describe the project execution flow and status of the project progress. Finally, based on the project execution flow, project states have been presented as an important element of the process tailoring framework.

4.2.5. Project Life Span Elements

Software development projects normally run in a standard way such that they start their life, completes all the tasks and then closes. This execution flow of the projects is generally determined by similar factors such as risks, team performance, work completion and client’s satisfaction but their effect varies from one project to the other project. This variable effect creates the variations in the execution flow of the projects.

Many factors such as skills of the team members [204], intervention of managers, use of advanced tools and technologies [205], size of the team [206], roles and responsibilities [207], support, encouragement and appraisal rewards by the client as well as the company are important factors which determines the performance of the project team whereas, personality diversity is one of the factors that plays an important role in efficient team building [208].

From client’s perspective point of view, the achievement of milestones in terms of completion of work and satisfaction of client is also an indicator of the team performance and progress of the project. The techniques to measure the team performance are limited however, a few general techniques have been discussed in [209] and [210].

A typical software development project has a start, middle and end [200]. The project passes through different phases throughout its lifecycle before completion, however, start and end of the software project lifecycle can be identified with respect to time [211].

Generally, project lifecycles have been divided into three broad phases such as start or initiation, middle or execution, and end or closure. In order to represent these phases of stages of the lifecycle, different terminologies have been used in the existing research works such that each represents the same meaning. The project lifecycle does not repeat itself except in special cases such as civil engineering projects for example construction projects, pipe lying therefore, it is better to term it as project life span [212]. The life span of each project is comprised of various phases as shown in Fig. 4.9 [211].

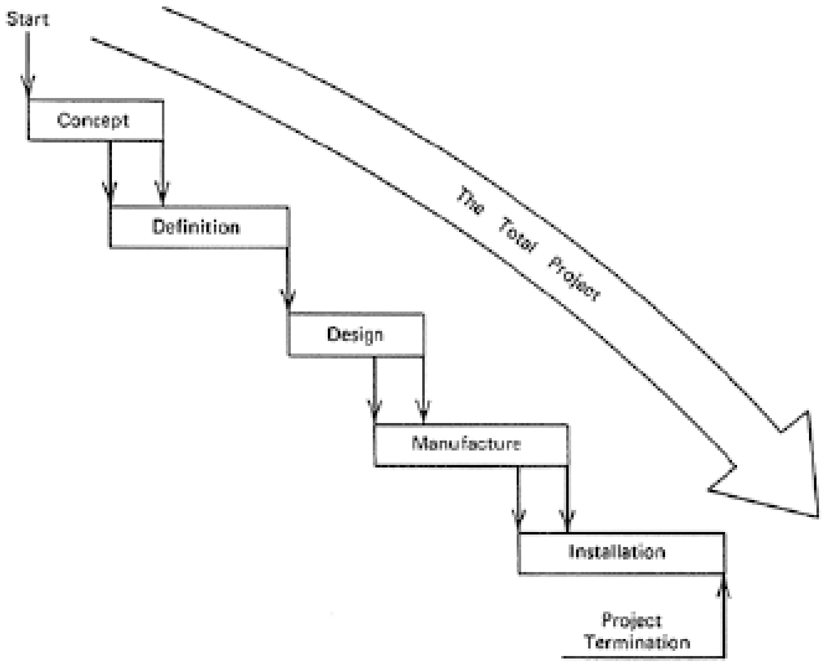

Fig. 4.9 Project Life Span [211]

Software development projects face many certain, uncertain, known, unknown, predictable and unpredictable situations throughout their life spans. The characteristic elements of project such as human resources, expertise and organizational resources change after each phase to start with the next phase [211].

A number of research works are available on project lifecycles such that each one presents similar stages and phases of the software development projects. Similarly, four phases of the project lifecycle such as project initiation, planning, execution and closure have been presented as shown in Fig. 4.10 [88].

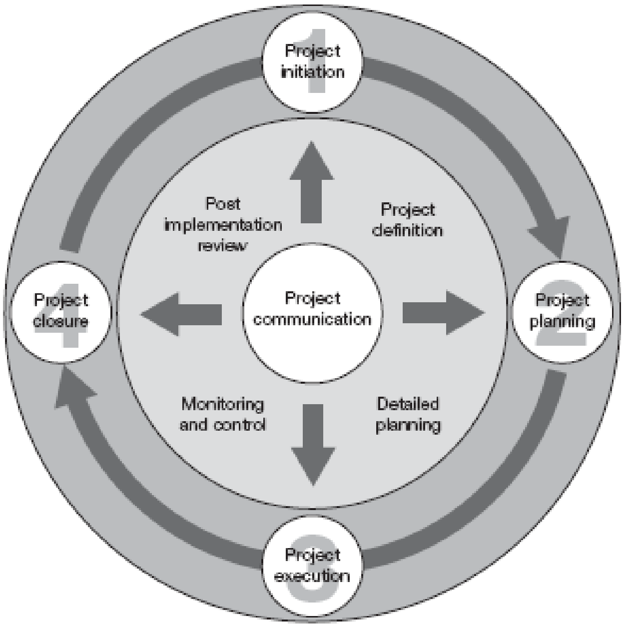

Fig. 4.10 Four Phases of Project Lifecycle [88].

Though a number of researchers have presented the lifecycle phases of a project differently from each other but these are built on basic set of phases which remains the same as shown in Fig. 4.9. Table 4.2 and Table 4.3 summarize the lifecycle phases of projects presented in different research works.

Table 4.2 Summary of Project Lifecycle Phases - I

<table>
<tr><th>Westland [89]</th><th>Song et. al [213]</th><th>Archibald [211]</th><th>Stuckenbruck [214]</th><th>Cavendish [215]</th><th>Wideman [212]</th><th>Webster [216]</th><th>Krezner [217]</th></tr>
<tr><td>Initiation
• Feasibility
• Team hiring
• Initial setup</td><td>Decision Making
• Requirement analysis
• Feasibility</td><td>Project Start
Concept
Definition</td><td>Initiation
• Project need
• Resource estimates
• Project Management
• Appointments</td><td>Conceptual
• Goals
• Management team
• Scope & Objectives
• Formal authority
• Estimating</td><td>Concept (Intermediate stages)
• Scope
• Initiation</td><td>Concept</td><td>Conceptual Definition</td></tr>
<tr><td>Planning
• Project
• Resource
• Communication</td><td>Design
• Supervision
• Preliminary design
• Detailed design
• Budget
• Tender</td><td>Design</td><td>Growth
• Project plan
• Project schedule
• Tasks (WBS)
• Resource allocation
• Team build up</td><td>Planning
• Budgeting
• WBS
• Define targets
• Make buy
• Scheduling</td><td>Development (Intermediate stages)
• Planning
• Prelim Eng.</td><td>Development</td><td>Production</td></tr>
<tr><td>Execution
• Deliverables
• Project Mgt.
• Time Mgt.
• Quality Mgt.
• Risk Mgt</td><td>Implementation
• Initial setup
• Coding/Implementation
• Testing
• Acceptance</td><td>Manufacture
Installation</td><td>Production
• Major work completion
• Design & Development
• Production & Testing
• Activation</td><td>Implementation
• Contracts
• Monitoring
• Problems Identification
• Re-planning</td><td>Implementation (Intermediate stages)
• Definition
• Design & Specs.
• Execution
• Build</td><td>Implementation</td><td>Operational</td></tr>
<tr><td>Closure
• Closing
• Review</td><td>Operation & Maintenance
• Operation
• Maintenance
• Evaluation</td><td>Project Termination</td><td>Shut-down
• Project Termination
• Resources released</td><td>Phase-out
• Problem resolution
• Rewards
• Resources re-allocation
• Review</td><td>Termination</td><td>Termination</td><td>Divestment</td></tr>
</table>

Table 4.3 Summary of Project Lifecycle Phases - II

PMI [218]	Kapur [219]	Morris [220]	Royce [221]		Mochal [212]	Cooper et. al [222]	Fish [223]	Allen [212]	Youker [224]
Start Initial	Idea Pre Launch Launch • Project request • Project charter • Planning	Inception • Project strategy	Engineering Stage	Inception • Idea	Analysis	Discovery Stage 1 Idea • Scoping	Discovery Requirement	Initiation	Pre-Identification Identification • Prelim. Feasibility • Resources • Plans
Intermediate	Execute • Scheduling • Tracking	Feasibility • Project brief		Elaboration • Architecture	Design	Stage 2 • Business case	Functionality Design Sanction	Definition	Preparation Approval Mobilization • Feasibility • Planning • Scheduling • Review
Final Finish	Implement • Completing	Design • Planning	Production Stage	Construction • Beta releases	Construct	Stage 3 • Development	Construct	Implementation	Implementation • Development
	Operation • Production	Procurement • Design • Contracting • Execution		Transition • Products	Test	Stage 4 • Testing • Validation	Check Review Audit	Completion	Operations • Production
		Commissioning & Start up			Implement	Stage 5 • Launch	Close-out		
		Post completion evaluation				Post-Launch			

The lifecycle phases and their respective activities as described in Table 4.2 and Table 4.3 are not specific to some particular type of projects or process models but present a general overview of the project lifecycles in context of different industries such as construction, and software development. However, division and definition of phases of a project may differ from project to project and industry to industry [212]. The same is the situation in small and medium sized software development companies where projects following agile based methodologies do not follow most of these phases and adapt or modify them based on the principles of agile manifesto as described earlier.

In the proposed process tailoring framework, typical phases of a project and their respective activities as shown in Table 4.2 and Table 4.3 have been adapted and rearranged. As agile based methodologies are product driven, evolutionary, and iterative approaches which do not follow proper process models [10] as well as their phases, the actual estimates of the performance and progress of the project are difficult to make. In such circumstances, the state of the project can describe the condition or status of the project, whereas a state is a condition of the project at a particular time. The phases of the projects as shown in Table 4.2 and Table 4.3 have been adapted and classified into project states. The sub-states of each main state are derived from the activities of the phases as shown in Table 4.2 and Table 4.3. The adaptation of phases and their activities has been made on the basis of principles of agile manifesto [46], [26] and characteristics of agile methodologies [10], [62], [27]. The adaptation process is shown in Fig.4.11. The activities that do not qualify the characteristics of agile methodologies [10], [27] have been eliminated.

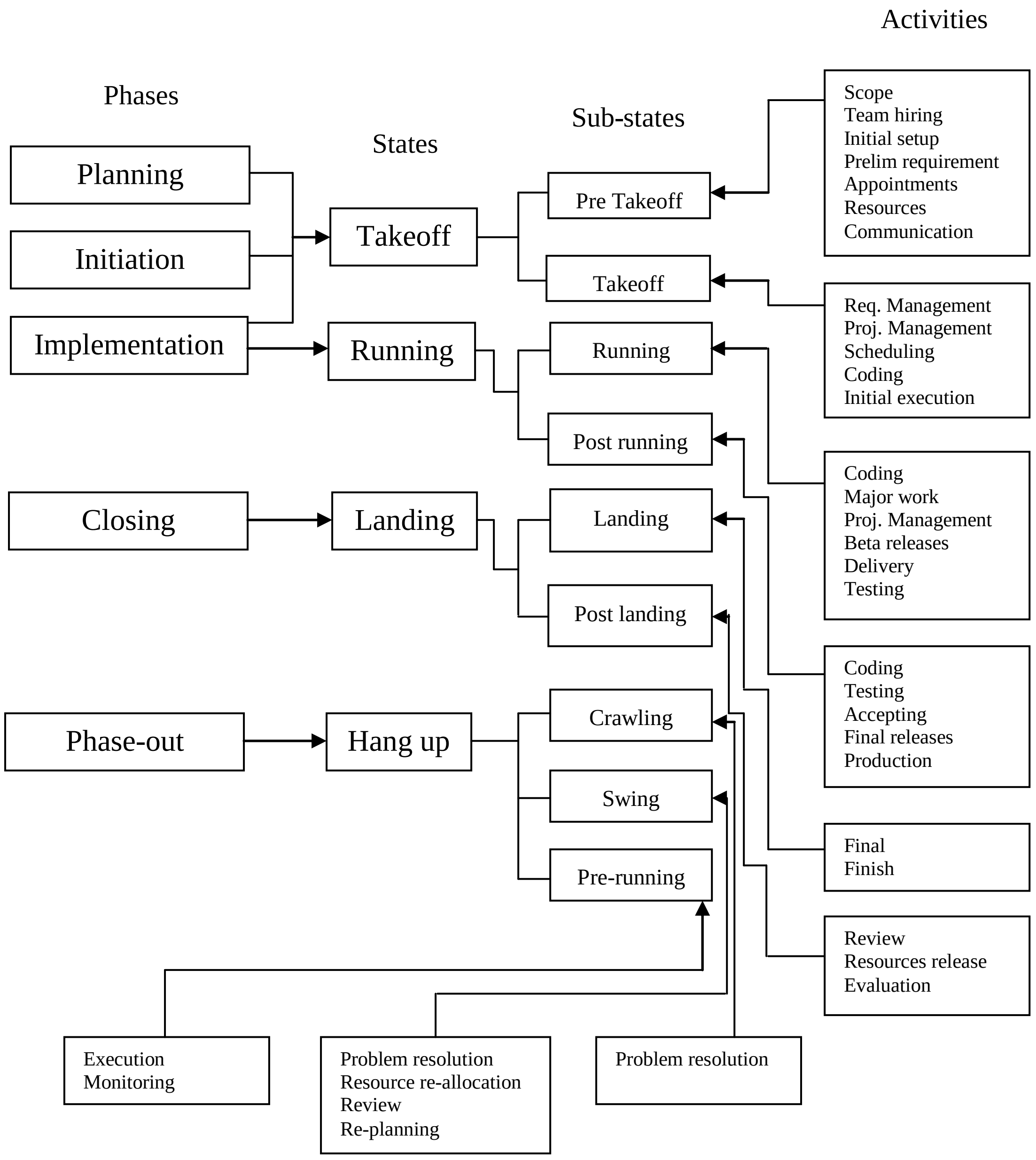

Fig.4.11 Software Project States

The names of the project states have been defined based on the resemblance of the software development project with the air flight where necessary arrangements are made before the flight takeoff, and important controls, measures and cares are adopted during the flight based on which the flight successfully reaches to its destination and lands on the ground successfully. Sometimes during the flight due to some unforeseen problems, emergency situations or incidents occur. Though emergency measures are taken then but based on the severity and critical nature of the problems, the airplane may either crash or lands on the ground safely or makes the crash landing successfully. Similar scenarios or situations occur during the software development project lifecycle that have been represented by the project states.

Take off is a starting state of a project and is further divided into two sub-states namely pre-take off and take off.

a) Pre-take off

It is an initial state of a project when preliminary requirements are received, project team is selected and initial set up is being made.

b) Take off

A project state where initial level development gets started and early iterations are in progress. It is the beginning for the development of the first milestone.

Running is a state when after project takes off, development and coding is in progress. It has two sub-states namely:

a) Running

A project state after the completion of few early iterations. Some deliverables have been released to clients.

b) Post running

It is a state when project has become matured after the completion of major iterations. For example, release of beta versions.

Landing is a state when after completion of all development tasks project is near to end. Further division of landing state is as follows:

a) Landing

A state when project is near to its completion. Last few deliverables are in progress. Deliverables might be a part of plan or newly received requirements from clients.

b) Post-landing

A state after completion of the project when priority tasks and requirements are about to finish. No new requirements are received to work on. Bugs are being fixed and necessary documentation is done. Status of all requirements is set to be verified and closed.

Hang up is a state when project progress gets slow down due to the presence of problems. Further division of hang up state is as follows:

a) Crawling

When project is in this state the progress of the project tends to slow down, delivery dates are not met, requirements are not fulfilled and tasks are not properly allocated. Client is also not satisfied and project seems to lose client's trust. Progress limiter class problems exist in this state. Success of the project might be at stake.

b) Swing

A state after crawling when revolutionary project management measures are taken up to overcome the issues faced by a project. Development processes are slowed down during this state due to the streamlining of the overall process. Quick project management measures are taken at micro-level. Senior management mostly plays its role in this state to save the project from failure.

c) Pre running

A state when development processes are speed up after swing state modifications. Project again enters into its running state.

The agile methodologies focus directly on coding and implementation instead of designing and planning. Based on the preliminary requirements coding is started. Therefore, planning initiation and implementation phases have been transformed into takeoff state. During takeoff state initial level coding of basic functionalities is performed.

Phase-out represents a condition when project deviates from its normal execution which could be due to the factors such as risks, problems or issues. The measures are taken to resolve the problems such as problem resolving strategies, re-planning, review, monitoring, and resource reallocations. However, as a result the project may proceed to the next phase, may be cancelled or go back to previous phase [212]. This scenario is transformed into hang up (phase-out) state and its sub-states.

During crawling sub-state problems can be easily handled and project keeps its execution continued, otherwise it enters into swing sub-state where in case of severe unhandled problems it may be cancelled. If problems are resolved, it enters into next state which is termed as pre-running and completes its life. This execution flow of project states is shown in Fig. 4.12.

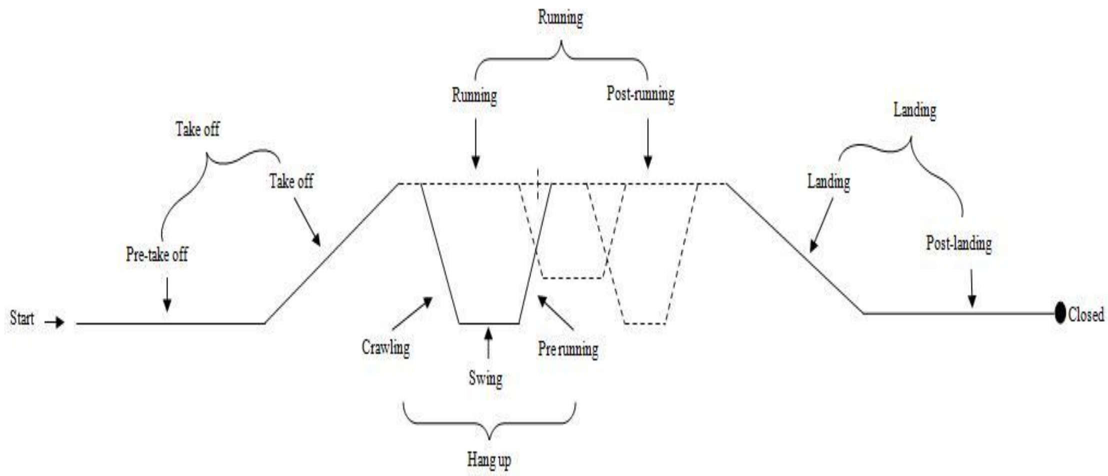

Fig. 4.12 Project States Execution Flow

This new nomenclature truly reflects the status and behavior of the software development project following agile based methodologies. Based on the execution flow of project states as shown in Fig. 4.12, the meta-model of project states has been presented as shown in Fig. 4.13.

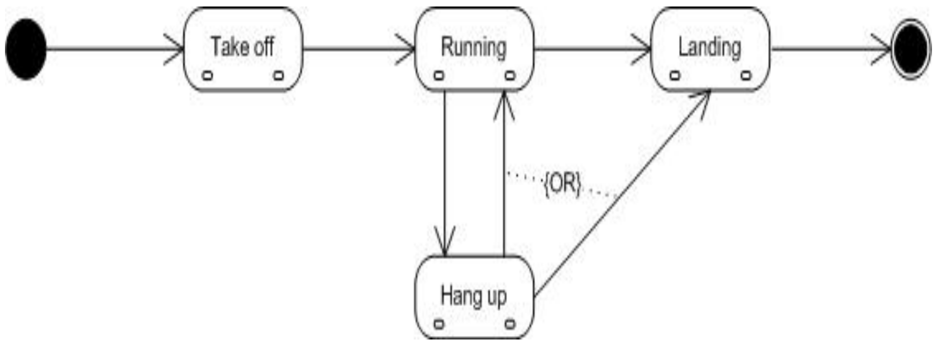

Fig. 4.13 Software Project States Meta-model

The meta-model of software development project states as shown in Fig. 4.13 represents the execution flow of a software development project following agile based methodologies. As such, the project takeoffs, starts running (execution) and lands (completion). During running state severe or mild nature problems may occur [215] as described in Table 4.2. With effective risk and project management practices these problems may overcome.

Hang up state represents uncertain situation of the project when there are problems in the project. So project may recover from the hang up state after problems have has been resolved and starts its normal execution. In case of failure, it may go to landing state and stops. The sub-states derived as shown in Fig. 4.12 have been represented through small rounded rectangles inside big rectangles labeled with names of project states as shown in Fig. 4.13. These sub-states are shown in Fig. 4.14, Fig. 4.15, Fig. 4.16, and Fig. 4.17.

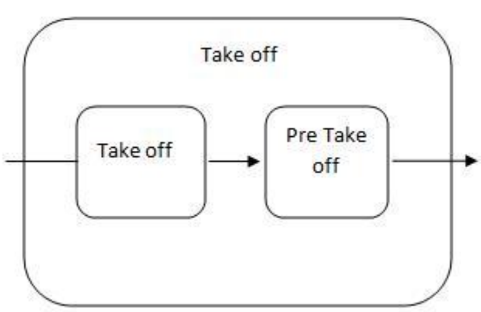

Fig. 4.14 Sub-states of Takeoff State

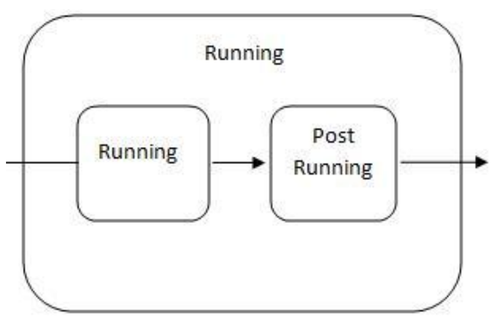

Fig. 4.15 Sub-states of Running State

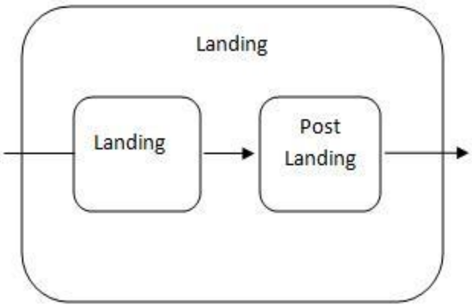

Fig. 4.16 Sub-states of Landing State

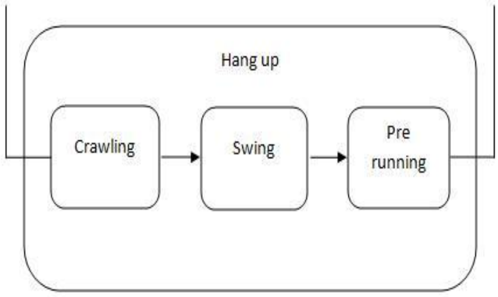

Fig. 4.17 Sub-states of Hang up State

These states represent the behavior of a software development project following agile based methodologies and have been represented as characteristic elements of project behavior as shown in Fig. 4.18.

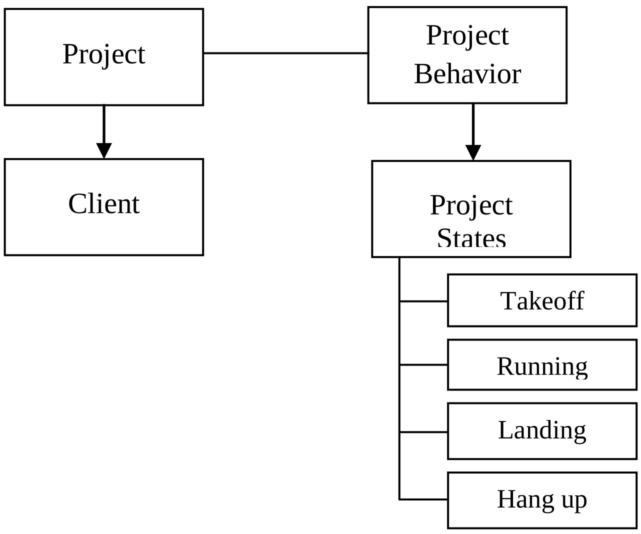

Fig. 4.18 Elements of Project Behavior

4.2.6. Software Process Tailoring Operations

Despite the realization of the importance of software process tailoring technique, the research works presented in this area are quite limited [6]. Majority of the work tailors the software processes based on the knowledge base of previous projects and focus on tailoring only one agile method such as XP. However, the existing approaches have used similar process tailoring operations such as add, delete, modify skip, and downsize [5], [123], [4]. Table 4.4 summarizes the process tailoring operations in prominent existing approaches.

Table 4.4 Process Tailoring Operations

Xu [5]	Welzel et. al [116]	Dai [4]	Yoon [3]	Xu [118]
Add	Delete	Adding	Addition	Addition
Downsize	Modify	Deleting	Deletion	Deletion
Drop/Skip		Splitting	Splitting	Modification
Expand		Merging	Merging	
Redefine				
Replace				

Table 4.4 shows that add, delete and modify operations have been commonly used in all approaches. Delete and modification operations have been modified, redefined and divided into further operations as well as replace and downsize have been used [5]. Furthermore, splitting and merging operations have also been used. It shows that tailoring operations can be used, modified and defined according to the requirements of the project and tailoring approach.

Therefore, based on the tailoring operations as described in Table 4.4, the tailoring operations for proposed framework have been defined as shown in the Table 4.5.

Table 4.5 Process Tailoring Operations Description

Activity/Operation	Notation/Symbol	Description
Add	(+)	Addition or adoption of one or more processes sub-processes or activities in an existing set of processes.
Delete/Skip	(-)	Removing or leaving one or more non required processes, sub-processes or activities from existing set of processes.
Modify	(Δ)	Updating or changing the existing one or more processes, sub-processes or activities.
Split & Select	($\neg$)	Dividing a process, sub-process or activity into one or more sub-parts and selection of the most suitable and most required one or more activities or processes or sub-processes.
Merge	($\veebar$)	Combining two or more processes, sub-processes or activities into single process, sub-process or activity based on their definition and performance.
Shrink	($\bowtie$)	Reducing or downsizing the number of steps, sub-activities or sub-processes from inside an activity or process.
Wrap up	($\oplus$)	Winding up or closing the activities or processes being performed to close the project.

4.3. Verification of Framework Elements

The elements of the framework that have been presented in sections 4.1 and 4.2 have been further verified through expert interviewing method (see section 3.4.1.2). The structured expert interviews have been conducted with 11 professionals having expertise in the areas of software development processes, software development, and project management. Being distantly located interviews have been conducted through emails [159]. The quick interviewing format and style of the interview has been set keeping in view the busy schedules of industry professionals and their time constraints. Table 4.6 shows the demographic data of all the experts.

Table 4.6 Information of Experts

Expert	Experience (years)	Company Size	Agile Based	Designation
X1	4.5	Medium	Yes	Senior Software Engineer
X2	4.5	Medium	Yes	Principal Software Engineer
X3	10	Medium	Yes	Academics
X4	7	Medium	Yes	Senior Software Engineer
X5	5+	Small	Yes	Principal Software Engineer
X6	7	Medium	Yes	Product Manager
X7	3	Small	Yes	Application Developer
X8	3.8	Large	Yes	Software Engineer
X9	4	Small	Yes	Director
X10	16	Medium	Yes	Director Projects
X11	10+	Large	No	Senior Manager

Table 4.6 shows that except two experts all others were working in small and medium size software development companies. Therefore, not fulfilling the criteria of small and medium size software development companies, these two interviews (X8 & X11) have been excluded. The designations of remaining participants show their level and domain of expertise. One interview has been conducted with a faculty members (X3) to get it verified from academic researchers as well. Table 4.6 shows that agile based methodologies were being followed by all the companies.

4.3.1. Findings of Interviews

The findings of the interviews have been summarized in Table 4.7. See Appendix C for details of the interview questions. Only main categories and their respective elements represented through questions are shown in Table 4.7.

Table 4.7 Findings of Interviews

Categories	Elements/Components	Answer (n=9)		
		Yes	No	Agreed Percentage
Client	Critical and influential factor.	9	0	100%
	Composite entity/role.	8	1	88.9%
	Interaction - project manager and team lead.	7	2	77.8%
	Client's perspective - understanding and maintaining	8	1	88.9%
	Client's perspective - each iteration	9	0	100%
Average:				91.1%
Key Processes	Resource management, Communication, and Requirement management	7	2	77.8%
	Requirement management - Tasks allocation, requirement gathering and tracking.	7	2	77.8%
Average:				77.8%
Project States	Takeoff, Running, and Landing	6	3	66.7%
	Hang up, Recovery, Cancellation	8	1	88.9%
Average:				77.8%
Process Tailoring	Small and medium sized companies	9	0	100%
	Project's requirements, and client requirements and expectations	9	0	100%
	Project states and client's perspective	8	1	88.9%
Average:				96.3%

The descriptive statistics [173] show that 88.9% experts have considered client's perspective as an important element of client factor whereas, 77.8% have agreed that resource management, communication, and requirement management are the key processes in agile based methodologies on which client is mostly concerned. It shows that client's perspective is an important element of client and resource management, communication and requirement management are the key processes in software development projects following agile based methodologies, whereas tasks allocation and requirement gathering and tracking are the sub-processes of requirement management.

Takeoff, running, landing and hang up have also been successfully recognized as behavioral states of the project on which 66.7% and 88.9% experts are agreed. 100% experts have agreed that process tailoring should be performed on the basis of project's requirements and client's perspective factors whereas, 88.9% experts have specifically recommended process tailoring based on the project's behavioral states and client's perspective elements.

The results of the structured expert interviews as shown in Table 4.7 completely verifies the framework elements based on the opinion of the expert professionals working in the IT industry as well as academic researchers.

4.4. Process Tailoring Framework

Based on the elements as identified from thorough primary study and then verified through expert interviewing method, the process tailoring framework has been formulated as shown in Fig. 4.19.

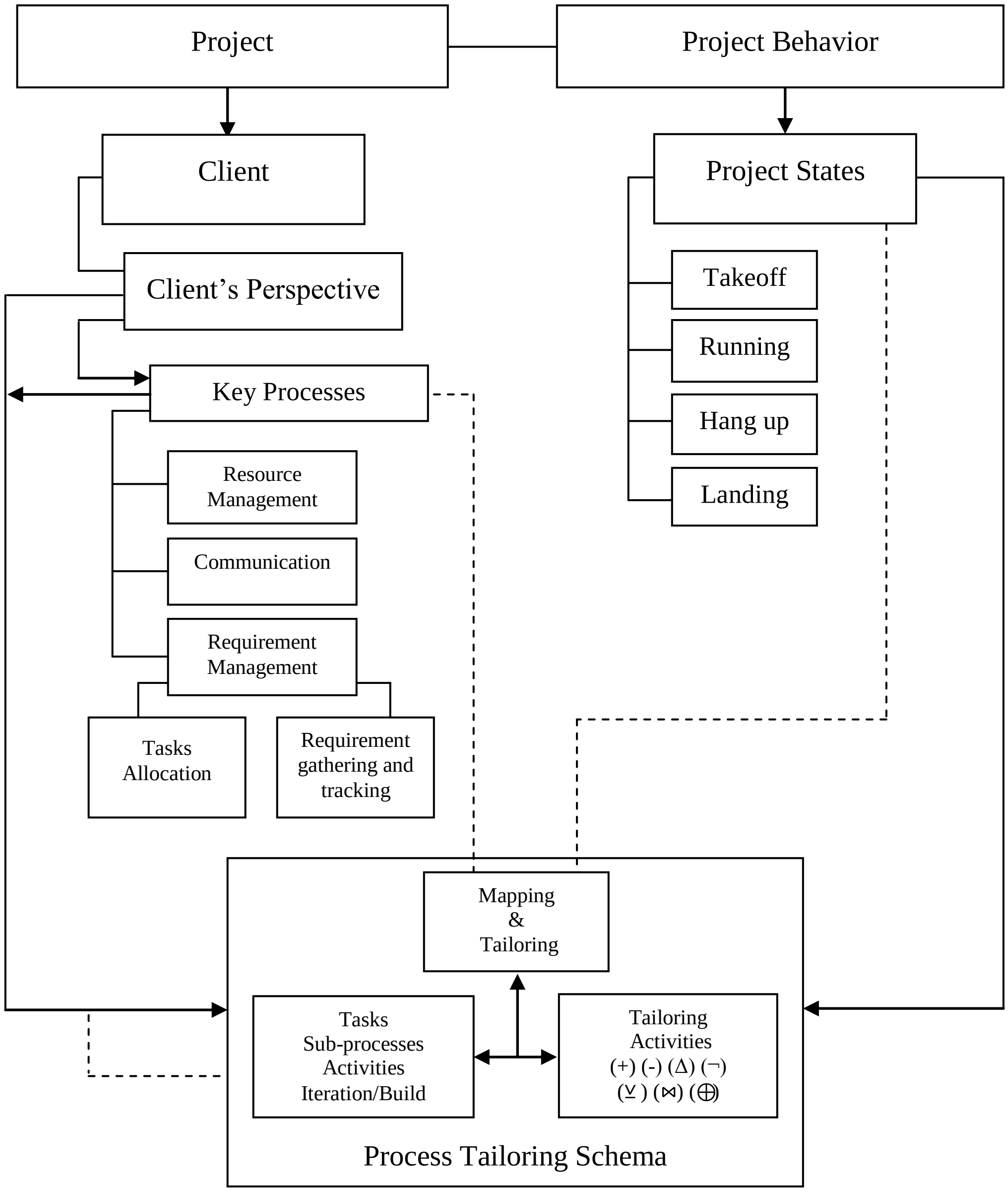

Fig. 4.19 Process Tailoring Framework

In addition to other components described in previous section, the mapping schema associates the project states, key processes and process tailoring activities or operations with each other. It tailors the tasks, sub-processes and activities of key processes when project is in a particular state as shown in Fig. 4.19.

Based on the various elements and process tailoring schema, the framework recommends the best tailoring strategies or policy at micro levels of the project for small and medium size software development companies. Furthermore, the framework derives the lightweight client based practices through process tailoring.

The framework provides the fundamental systematic guidelines on software process tailoring. The framework can be further inherited, extended and modified based on the basic scheme of tailoring it has provided. The framework and its components have been validated through both qualitative and quantitative methodologies as presented in chapter 5 and 6.

4.5. Summary

The process tailoring framework and its components have been derived from the research literature related to the agile based methodologies, process tailoring, software development processes, software development projects lifecycle, project management and risk in software development projects. Structured interviewing approach has also been used to formulate the framework.

The client of software development projects, client's perspective and software project states have been identified as the fundamental components of the process tailoring framework. Three key processes such as resource management, communication, interaction and coordination, and requirement management have been identified as the key processes in agile based methodologies on which client is also mostly concerned. After deriving these components, finding support from the literature and structured interviews, the framework has been formulated by integrating all these components.

The framework emphasizes on tailoring three key processes according to the respective state of the project based on client's perspective in projects running in small and medium sized companies according to the agile based methodologies. Finally the framework generates a process tailoring schema that provides guidelines on how and when process tailoring should be performed.

CHAPTER 5

CASE STUDY – I

Overview

This chapter presents the first case study and describes how it has been conducted. It also explains its various scenarios, sequence, flow and chain of evidences to validate the proposed framework. Further, this chapter presents separate analysis of each individual component of the framework as well as an overall analysis of the complete case study as a whole. Finally summary of the complete analysis has been presented.

5.1 Understanding Project Client

The client of the software development project being the key component of the framework has been analyzed in detail because to understand client's perspective, it is necessary to first investigate and understand its structure, interaction and role in software development projects. Sections 5.1.1 and 5.1.2 present a detailed analysis of these aspects of the framework's main component client for its complete understanding while section 5.2 presents analysis of client's perspective.

5.1.1 Client's Composite Structure

Client's composite structure helps in understanding the concept of term software project client and its role in software development projects. In social sciences a lot has been written in this regard. The term client, its role and importance also needs to be realized in software engineering. Therefore, observations and analysis have been made on the role of client, its importance and involvement in software development projects.

Table 5.1 shows the observations made on the structure of client, existence of its various roles, number of persons in the same role, starting date of the role, its job responsibilities and technical skills in two projects, project 'A' and project 'C'. The starting date represents the hiring or starting date of the respective role.

Table 5.1 Observations of Client's Composite Structure-I

Project Name	Client/ Role	Existence	Number of Persons in the role	When Role Started	Role Job	Technical Skills
Project A	CEO	Yes	1	Beginning of the project.	i. Owner of the project. ii. Monitoring and planning. iii. Defining requirements, iv. Meetings and checking project progress.	Naive
	Tech Manager	Yes	1	From sixth month of the project start.	i. Providing assistance to CEO in technical matter. ii. Offshore technical manager. iii. Defining requirements and handling technical matters. iv. Monitoring team performance. v. Evaluating build releases and code quality.	Expert
Project C	CEO	Yes	1	Since the beginning of the project.	i. Owner of the project. ii. Requirement specifications. iii. Setting deadlines. iv. Meetings	Expert
	Project Manager	Yes	1	After five months.	i. Handling project plan and schedule with offshore team. ii. Requirements iii. Meetings iv. Functionalities v. Debugging	Expert
	Tech Lead/ Manager	Yes	2	After four months.		
	Developer's Team	Yes	3	After five months.	i. Coding ii. Debugging	Expert

It has been observed that not all the roles of the client exist in a single project as shown in Table 5.1. Therefore, analysis has been made on two projects in first case study such as project 'A' and project 'C' for analyzing client component completely.

5.1.1.1 Analysis of Client's Composite Structure

In order to investigate and analyze the structure of client, two different projects have been analyzed. All the roles as shown in Table 5.1 have been considered as client by the offshore project team. On the other hand, client may also perform various roles. It is not necessary that all such roles exist in a single project. The role of the client or the persons that client hires to perform these roles are need based and varies from project to project.

As shown in Table 5.1, in project 'A', the CEO of the client's company was the actual client of the project. As a client and owner of the project he was used to do meetings with the offshore team members regarding project planning, scheduling, and for looking into technical details such as coding and functionalities. The client not having that much technical skills, appointed a technical manager (Tech Manager) on his side to handle the technical matters of the project and to coordinate with the project team on the technical details of the project.

The technical manager was appointed five months after the project had started. His appointment was purely need based. Inexpertness of client in technical matters of the project was the main reason behind this hiring. These two roles remained present throughout the project lifecycle. The technical manager was used to visit offshore team off and on to monitor the project progress, discussing technical details of the project with the team and whenever there was the occasion to release the final version to some investor or other client.

On the other hand, the scenarios in project 'C' were more tedious as compared to the project 'A'. Since the beginning of the project, client had a plan to take over the project development completely from the offshore team after some certain level of completion of the project. As per his plan, he started appointing project manager, technical lead/manager and project team. Till the six months of the project progress he had hired all these roles in his project at his own side. Each of them was corresponding with the offshore project team members. Dealing with multiple roles, and importance and preference of requirements of each person from client side created unpleasant situations for the offshore team.

5.1.1.2 Summary of Client's Composite Structure

The existence of different roles at client side in both projects as shown in Table 5.1 shows that the term client is a composite entity that may have more than one role. Based on the analysis of both project'A' and project 'C' the client's composite structure has been formulated as shown in Fig. 5.1 which has been further validated in second part of the case study.

Fig. 5.1 shows that techlead/manager and project manager may be the types of roles of CEO who is the actual client of the project. It is not necessary that techlead/manager and project manager exist in all the projects. Moreover, it is also not necessary that project team at client side exist in all the projects.

For a techlead/manager the existence of a project team is not necessary, but in most of the cases existence of project team is necessary for the existence of project manager's role. Therefore, it can be concluded that a project may have no or more than one techlead/managers as well as one or more project managers. The optional existence of project team has been represented by dotted line between project manager and project team as shown in Fig. 5.1.

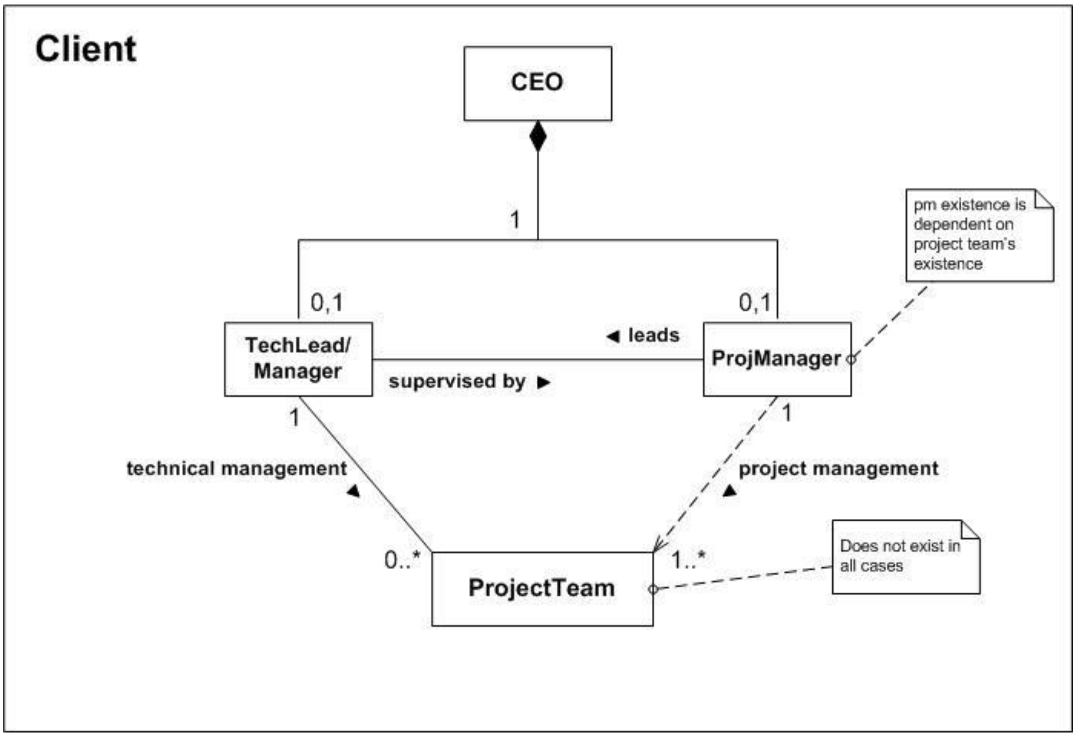

Fig. 5.1 Client's Composite Structure

For project team each role is important as they have to manage all of their requirements and preferences. Each role being equally important for project team is considered as client by them. Therefore, client has a composite structure that comprises of various roles that are performed by the client himself or by other hired people. Mostly these roles are appointed when:

i. Client is less technical having low level or no expertise in technical matters, algorithms, and coding part of the application.
ii. Project is complex and has big scope.
iii. Requirement of the project.

The analysis shows that these roles may or may not exist in all the projects. More than one instances of a same role in different capacities may also exist. For example, technical manager, technical lead, and project manager who have their own responsibilities to perform during the project without overlapping each other's.

5.1.2 Client's Interaction Overview

As software project client has various roles as shown in Fig. 5.1, these roles interact with key roles of the software development teams. The interaction between various roles of client and software development team has been summarized in Table 5.2.

Table 5.2 Observations of Client's Interaction-I

Project Name	**Interaction Entities**		**Type of Interaction**	**Frequency of Interaction (No. of times per day)**
	Client	**Project Team**		
Project A	CEO	PM	Direct	1 – 2
	CEO	Team Lead	Direct	2
	CEO	Developer	Indirect	~ 1 – 2 / month
	Project Manager	PM	N/A	N/A
	Project Manager	Team Lead	N/A	N/A
	Technical Lead/Manager	PM	Indirect	~ 1 – 2
	Technical Lead/Manager	Team Lead	Direct	~ 2
	Technical Lead/Manager	Developer	Direct	~ 1
Project C	CEO	PM	Direct	2 – 3
	CEO	Team Lead	Direct	2 – 3
	CEO	Developer	Indirect	~ 1
	Project Manager	PM	Direct	1 – 2
	Project Manager	Team Lead	Direct	2 – 3
	Technical Lead/Manager	PM	Direct	1 – 2
	Technical Lead/Manager	Team Lead	Direct	2 – 3
	Technical Lead/Manager	Developer	Indirect	~ 1 – 2 / week

~ almost

5.1.2.1 Analysis of Client's Interaction

The case study analysis of client's interactions has also been made on two different projects with the understanding that various client based scenarios, interaction and communication channels are not worthy to analyze in a single project. It is not necessary that all the possibilities could exist in a single project. Therefore, in order to conduct a complete analysis of possible available situations, the same two projects as in section 5.1 have been analyzed for client's interaction with the project team.

In project 'A', there exist only two roles of client such as CEO and Technical Lead/Manager as the project manager at the client side was not hired. Client being the naïve as shown in Table 5.1 made him realized the need of a technical person (Technical Lead/Manager). Both CEO and technical lead/manager were in direct correspondence with the project manager, team lead and the developer. Though the interaction between technical lead/manager and project manager of the team was not direct but project manager was used to remain present in almost all the meeting sessions between technical lead/manager and team lead. He was also CC'd in all the emails. The frequency of interaction was 1 – 2 times in a day. It has been observed that the interaction was mostly 2 times in a day in case of problems in the project or close deadlines. On average it was 1 time in a day and sometimes reaches upto 2 times.

Likewise, in project 'C', there exist three roles of client such as CEO, Technical Lead/Manager and Project Manager. The project manager was appointed in order to handle the development team at client side. Like CEO, both technical lead/manager and project manager were directly interacting with the project manager (PM) and team lead of offshore project team. Very often and indirectly they were communicating with the developers. After very long intervals and at the time of utmost need such as some issue in a particular module or functionality they had to have talked to the respective developer. Otherwise, team lead was enough to discuss with them on all technical matters.

As compared to project ‘A’, the project ‘C’ faced problems since its beginning. Therefore, the frequency of interaction in this project has been more as 2 – 3 times in a day as shown in Table 5.2. On average it has been observed as 2 times per day. The factors such as higher number of persons at the client side, their multiple requirements and issues have been found as the reasons behind this frequent interaction.

Moreover, in both project ‘A’ and project ‘C’, the direct interaction between client and the team members has been found quite rare and limited which is represented by dotted line as shown in Fig. 5.2. The Fig 5.2 shows a complete overview of interaction among various roles of client and software development team. The Fig. 5.2 has been further elaborated in second part of the case study.

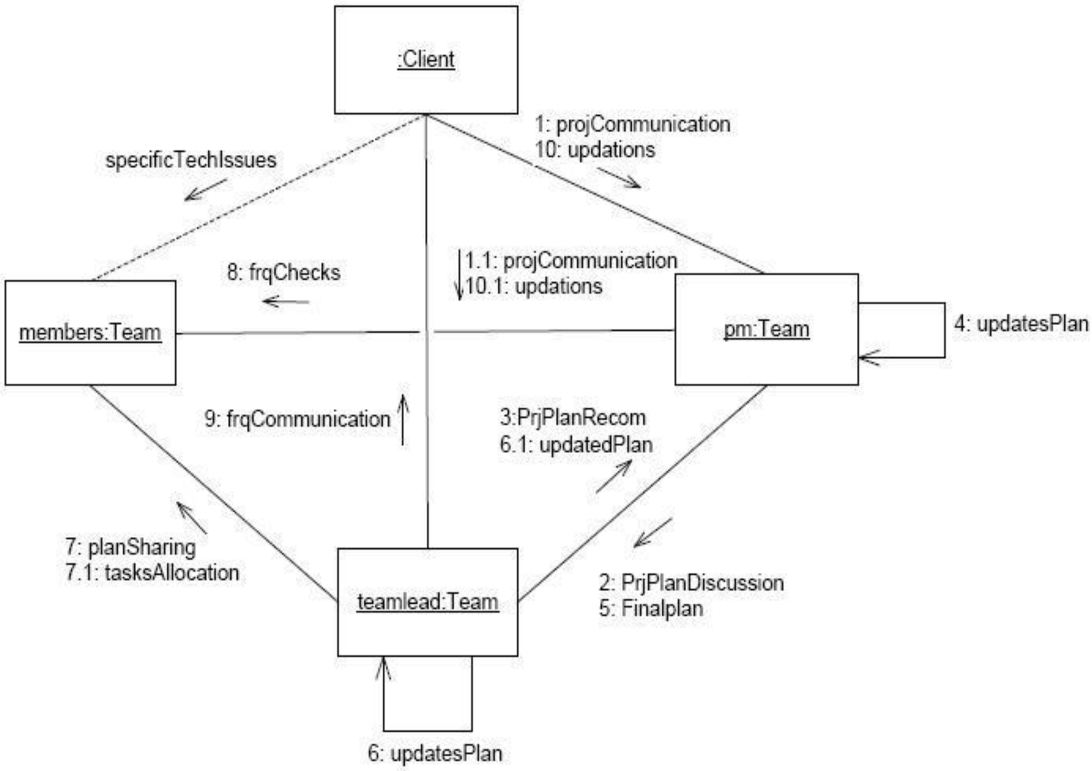

Fig. 5.2 Client’s Interaction Overview

5.1.2.2 Summary of Client's Interaction

The analysis shows more than one roles at the client side that made the proper interaction and communication channels necessary in both projects. From the analysis of both projects it can be concluded that the frequency of interaction is dependent on the following factors:

i. New requirements and modifications in the previous requirements.
ii. Technical issues such as server side for example some service disruption.
iii. Close deadlines.

The following tools/options have been used for communication with the client in both projects:

i. MS Groove
ii. Yahoo Messenger
iii. Live Phone Calls
iv. Net meeting services

The analysis shows that the interaction and communication between client and offshore team resources is quite necessary and important. Software teams in the same company may use same options and tools for communication while similar techniques are used in most of the other companies. The above mentioned three factors mainly determine the frequency of interaction & communication among all the resources of a project. The same set of factors was also observed in the study of other projects too.

The remaining analysis of client's perspective, project states meta-model and process tailoring schema has been made only on project 'A' in the first phase of the case study. This approach has been used for the better understanding, clarity and readability of the remaining portion of the case study.

5.2 Role of Client's Perspective

The analysis of client's perspective shows interesting facts about the behavior of client and its role in software development projects. Table 5.3 summarizes the observations of client's perspective in project 'A'. The details of each key area, client's satisfaction and improvements as shown in Table 5.3 have been presented in Appendix D. For understanding client's perspective and its role in successful project progress has been analyzed as shown in Table 5.3. Client's responses have been observed from his feedback through emails, phone calls, and chatting, expressing in terms of showing happiness and satisfaction on good work as described in [173], [225], [226], [227].

Table 5.3 Client's Perspective Analysis

Key Area (scenario)	Client's Response before changes in Project	Improvement			Client's Respon-se after changes in Project
		Sharing interval (duration)	Process/Sub-process/ Activities	Documentation	
Requirement gathering and tracking document's sharing with the client.	Client was not satisfied with existing approaches	Monthly Weekly Daily	Requirement's status, Requirement management, Requirement tracking, Document sharing, scheduling.	Master requirement tracking document, Project plan, Progress status.	Client was satisfied with improve-ments made.
Status updates		Monthly Daily	Team progress check, Tasks status, Project status	Status reports	
Tasks allocation		Monthly Daily	Feedback from team, Manage tasks repository, Update tasks status, Tasks extraction from repository, Tasks assignment	Project schedule and plan, Status reports, Tasks allocation document, Feedback document.	
Releasing build plan and meetings		Weekly Monthly (milestones) Daily (meetings, in different forms)	Requirement evaluation, Requirement extraction, Project scheduling, Tasks assignment & allocation, Structured and unstructured meetings	Requirement management, Project planning, Status reports, meeting minutes.	

Table 5.3 Client's Perspective Analysis (Continued)

Key Area (scenario)	**Client's Response before changes in Project**	**Improvement**			**Client's Respon-se after changes in Project**
		Sharing interval (duration)	**Process/Sub-process/ Activities**	**Documentation**	
Resource allocation and project staffing		Once in the start Need based Monthly (project staffing)	Requirement analysis, tasks analysis, resource assignment, effort estimation, resource allocation, % effort project staffing.	Project staffing document, Resource plan.	
Frequent meetings and updating client		Daily (unstructu-red, informal) Weekly/ monthly (structured, formal)	Tasks status, Project status, progress report, demonstrations	Project plan, schedule, status report, resource allocations.	
Automated project Management	Client was satisfied.	Daily	Requirement gathering & tracking, Tasks allocation and management, Bugs management.	-	Client was satisfied.

The satisfaction of client has been observed from his expression of satisfaction, and complaints during the communication with the project team. Since the beginning of the project the client's perspective was not that much understood by the project team and project manager of the project. The project progressed smoothly in the beginning but started facing problems in just few early months. Table 5.3 shows the level of dissatisfaction of client, improvement measures adopted based on the client's perspective and outcome of those measures in the form of improvement in the satisfaction level of client.

The client was not that much satisfied with the project team as shown in Table 5.3. Client was continuously communicating with the project manager through emails, chatting and phone calls. He expressed this during the meetings but project manager and team members could not realized it. Ultimately he wrote to the CEO of the company and brought this into his notice. He had serious concerns regarding the following mismanaged areas:

1. Requirement gathering and tracking document's sharing with him.
2. Status updates and tasks allocation of the team.
3. Build release plans and deadlines.
4. Resource allocation and project staffing.
5. Frequent and regular communication.

The project manager was unable to manage these process areas up to the expectations of the client that made the client dissatisfied with the performance of the whole team. With the direct intervention of CEO and a senior project manager, processes and practices were adopted as expected by the client and project planning and management documents were shared with the client first on monthly, then weekly and finally on daily basis as shown in Table 5.3.

The analysis of client's perspective through his emails and phone calls enabled the senior project manager to understand the interest, requirements and expectations of the client. The client's perspective approach was then adopted in all above five process areas. The dissatisfaction level of client that was increasing before proper measures were adopted was then tend to decrease and ultimately client became satisfied with the overall progress of the project as shown in Table 5.3.

5.2.1 Analysis of Client's Perspective

The client in this project has been found very much concerned about his project, its progress and resources assigned to the project. Since the beginning of the project, he wanted to remain updated with the whole project status, team members, their performance and tasks allocation and completion. Unluckily, the project manager did not pay attention to fulfill these important requirements of the client. A senior project manager was also putting his 25% effort on the project. The senior project manager was assigned to facilitate the tasks of project manager and overall take care of the project based on the information provided by the project manager.

The actual processes and documentations in this regard were found different from those verbally communicated by the project manager to the senior project manager. It led to a situation where the team members were not assigned proper tasks, tracking of tasks and requirement was improper, requirement management had no proper process and the deadlines were normally being late. Client remained unsatisfied with the overall performance of the team and progress of the project. On the other hand, many errors in the developed modules also made him realized that his project is not going good and created bad impression.

Ultimately, an email was sent by the client to the CEO and senior project manager expressing his dissatisfaction over the project in quite harsh words. CEO directly jumped into the project. CEO spent hours with the team day and night, senior project manager put his 75% effort on the project and another experienced project manager from some other project was also assigned as a silent resource on the project with 50% effort allocation. Keeping in view the client's perspective, the above mentioned measures were adopted to streamline the whole project.

Understanding and maintaining the client's perspective, the whole project was refurbished. With the whole effort of 3 – 6 months very lightweight approaches were defined to manage requirement gathering and tracking, resource allocation, tasks allocations, effective communication, back log and other activities. It took around six months in gaining back the trust of the client.

5.2.2 Summary of Client's Perspective

The client's perspective factor in this project was very prominent. CEO of the company, a senior project manager and a newly assigned experienced project manager judged the client's expectations from his behavior, emails and meeting sessions, took the notes, recorded the meeting audios and converted them into a set of activities and processes.

As shown in Table 5.3, all the expectations, concerns and requirements of the client were limited to the activities or processes as grouped into:

i. Communication (meetings, document sharing such as project plan & schedule)
ii. Requirement management (Requirement gathering and tracking, tasks allocation, project plan, project schedule)
iii. Resources management (effort distribution of resources, scheduling)

These processes have been found as the key processes on which client is mostly concerned. Transparency and streamlining in these areas made him satisfied with the performance and progress of the project. His level of satisfaction, trust and comfort was revealed from his emails, chatting and meetings.

The analysis shows that understanding and maintaining the client's perspective in the project is very important and critical factor for the success of a project. In agile methodologies following iterative development approach, it is necessary to follow the client's perspective during each iteration, phase, and process of the project throughout the project lifecycle as shown in Fig. 5.3 and Fig. 5.4.

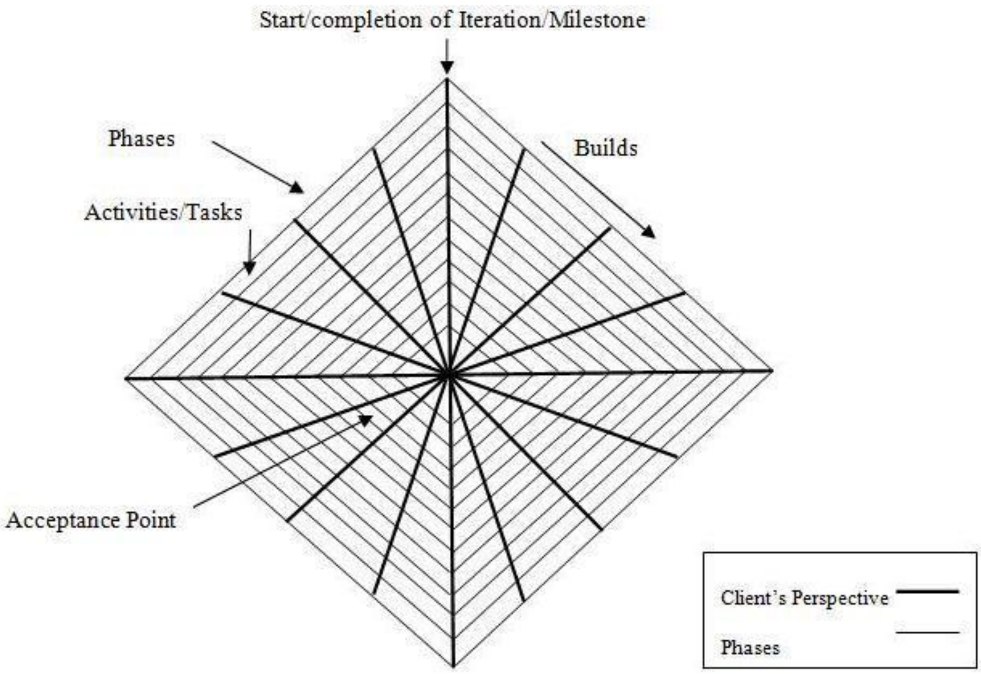

Fig. 5.3 Practicing Client's Perspective

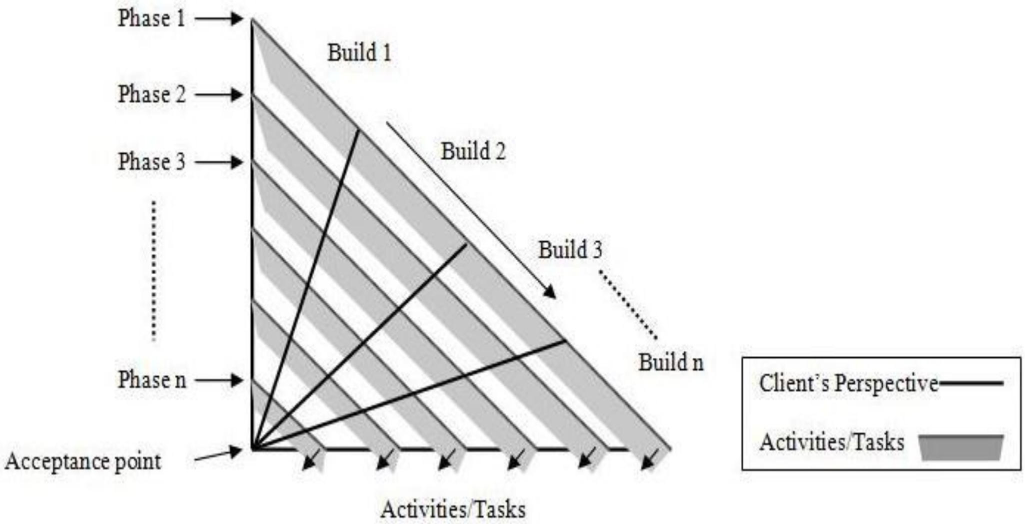

Fig. 5.4 Cross Section View

Each circle in Fig. 5.3 represents an iteration (also called milestone) having phases, activities or processes which is the characteristic of agile methodologies [10]. Phases are actually the key processes of agile methodologies in agile based software development projects as described in section 4.2.3 in chapter 4. Each inner cycle is smaller than the outer one which shows that as the project progresses and gets matured the number of functionalities, tasks and activities tend to decrease in number. There may be more than one build in each iteration, whereas, a build is the set of functionality to be delivered to the client. Upon reaching the completion point, a build is released or delivered to the client. The acceptance point is the state when client approves the build release after verifying that all the respective functionalities and specifications have been implemented in the current build. The acceptance point concept has been adapted from [212]. The solid and bold black lines represent client's perspective showing that all the phases, their activities and tasks are inside the boundaries of client's perspective from the start of each build till its end. The cross section view gives better understanding as shown in Fig. 5.4.

It is important and necessary in the projects to understand and maintain the client's perspective for satisfaction of client, their smooth progress and successful completion in addition to other factors. If client's perspective is missed or overlooked at some point it may lead to unpleasant situations and may cause severe problems at later stages of the project. Therefore, according to analysis, the client's perspective is quite necessary for the success of a project following agile based methodologies in small and medium sized software development companies.

5.3 Behavior of Software Development Project – Project States Meta-model

The observations made during the case study on the behavior of project and existence of project states have been summarized in Table 5.4 describing the behavior of projects and project states meta-model. The analysis has been made on the basis of problem factors, client's factors and project response which have been presented in detail in Appendix E. The activities performed during each state, factors or problems (see section 4.2.4 and 4.2.5), duration of each state and response of project to these states have been investigated and presented in this section.

Table 5.4 States of Project 'A'

State/ Substate		Duration	Description	Activities	Problem class/Factors	Problem Impact	Project Status
Take off	Pre-takeoff	1.5 months	Exists	System setup, Resource allocation, Preliminary meetings	Performance Minor, Client's Perspective	Low	Stable
	Takeoff	1.5 months	Exists	Alpha version tasks, resource allocation & meetings.	Performance Minor, Management Critical, Client's Perspective	Low Medium	Stable
Running	Running	4 months	Exists, Problems started	Beta versions, new modules and applications, post beta versions.	Performance Minor, Management Critical, Client's Perspective	Medium High	Stable-Instable
	Post running	6 months	Exists with major problem	Final versions, milestones, distributed teams for QC.	Performance Minor, Management Critical, Client's Perspective	High High	Stable-Instable

Table 5.4 States of Project 'A' (Continued)

State/ Sub-state		Duration	Description	Activities	Problem class/Factors	Problem Impact	Project Status
Hang up	Crawling	3 months	Exists in parallel to running state	Light weight processes for requirement management, resource and tasks allocation, project staffing and automated approaches for project management, resource shuffling.	Progress Limiter, Client's Perspective	High	Instable
	Swing	4 months	Exists in parallel to post running and last month of crawling state, development tasks got slow down, major problems solved, development processes slow down.		Progress Limiter, Client's Perspective	High	Instable
	Pre running	2 months	Exists, started in later months of swing state.	Milestones decided, proper team management and tasks allocation, requirement tracking and bugs tracking.	Management Critical, Client's Perspective	Medium	Instable-Stable
Landing	Landing	2 months	Exists, but existing project work got finished and a new development site emerged.	The client at the end of the project started the design of a new insight into the project. With completion of existing project, a new idea emerged from the existing and project got a new direction.	Performance Minor, Management Critical, Client's Perspective	Low Medium-Low	Stable
	Post landing	1 month	Project successfully completed.		Performance Minor, Management Critical, Client's Perspective	Low Low	Stable

5.3.1 Analysis of Project States

Appendix E presents the detailed analysis of project behavior in context of both project states meta-model and problems classes. The project states have been analyzed in context of problem classes and client's perspective factors. Identification of existence of problem class, reasons and response of state to that class as well as cause and effect of client's perspective on each project state has been analyzed thoroughly. Both project states and role of problem classes has been studied and analyzed in parallel due to the relation between problems classes and project states. Moreover, the analysis has also been made on the behavior of project states and their characteristics, and the characteristics of problem classes.

As shown in Table 5.4 it took total 3 month to the project during takeoff state in its initial setup, resource allocations, and getting and implementing requirements of the early releases (takeoff state). The duration of the running state in the project was 10 months while swing state was comprised of 9 months out of which 2 months belonged to pre-running sub-state when project started its normal execution again. It took 3 months in closing the project therefore, landing state was comprised of the same duration. It shows that all the states were present in project 'B'.

5.3.1.1 Impact of Problem classes and Client's Perspective on Project states

Performance minor class problems have been observed mainly during this state in addition to a few management critical class problems. However, overall they did not affect the project progress which remained stable during the takeoff state.

The duration of running state has been observed 10 months in which beta versions and final versions of the application have been released. Though project had has been matured but high frequency of management critical problems affected the project adversely.

As a result, the expression of dissatisfaction of client created instability in the project. Due to such issues, the progress of the project during running state had been slowed down. The crawling sub-state of the hang up state started appearing. The development processes have had been notably slowed down. Further, major modifications in the processes and activities were made during the swing sub-state of the project. After successful recovery from the swing sub-state, the project entered into its pre-running sub-state and remained in there for 2 months for stability and again started its running to progress normally with more stability. During hang up state, the major modifications in the processes were made as required and expected by the client. The processes were adapted as client wanted to see the project. The hang up state that had started in parallel to the running state remained for about 10 months. Later on, the project completed successfully. The modifications made during the hang up state won the trust and satisfaction of the client that remained persistent till the completion of the project.

It has been observed that projects always face problems throughout their life such that no project is safe from them. Inspite of having experienced project manager, skilled team resources, and well established processes such problems are unavoidable. Most of the problems are associated with some certain process, scenario, artifact or activity. Minor performance minor problems may also lead to some awry situation. Therefore, problems cannot be ignored and their postponement to be handled at some later stage can adversely affect the project.

Each problem that belongs to some particular class helps in determining the existing state of the project. The analysis shows that a project may not always be in hang up state in case of problems. Apparently, project enters into a hang up state when problems are more severe and their repercussions are adverse. However, the circumstances and factors that lead a project to a hang up state and how project responds are variable and uncertain, and are difficult to foresee.

The response of the project during hang up state also varies and depends on the factors such as:

i. Intensity and frequency of the problems.
ii. Type and nature of the problems.
iii. Measures taken to avoid and handle the problems.

iv. Effect on client's perspective and

v. Response of the client.

At several times, during the project lifecycle, the response of the project during hang up state has been found different. In one instance, the project entered into crawling sub-state, got recovered from the problems and passing through pre-running sub-state again started its running state. Swing state did not occur as shown in Fig. 5.5. Also hang up state has been found overlapping running state as shown in Fig. 5.6. In such cases the progress of the project does not get affected.

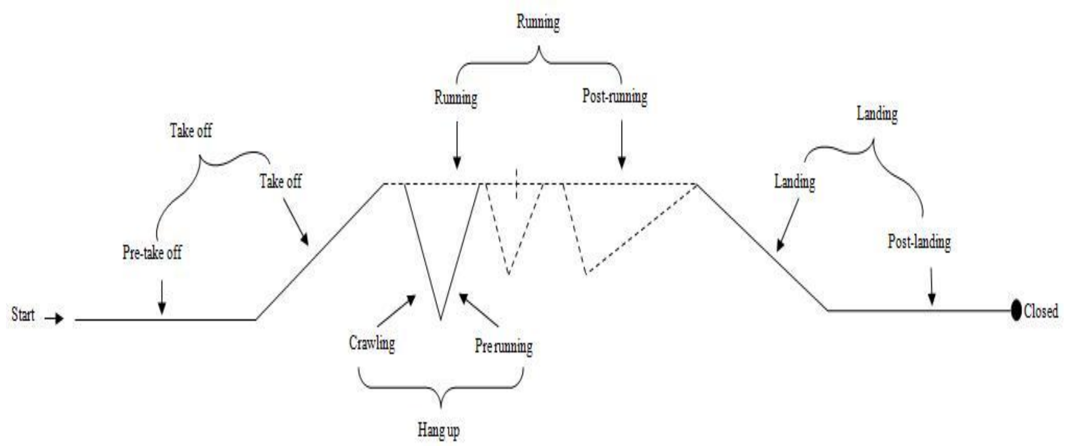

Fig. 5.5 Swing State Variation

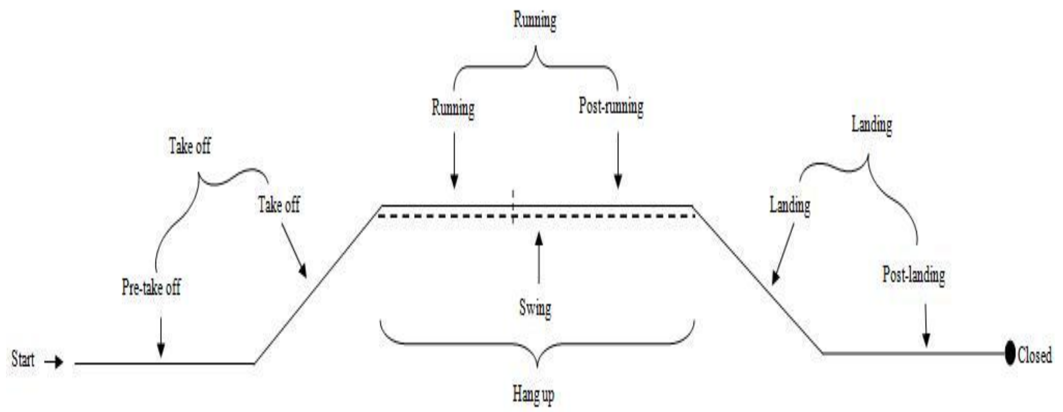

Fig. 5.6 Hang up State Overlapping

Critical problems such as management critical are when left unhandled becomes progress limiter problems and greatly affect the progress of the project. These problems adversely affect the client's perspective, satisfaction and trust such that this severe impact pushes the project into swing sub-state of hang up state as shown in Fig. 4.12. The hang up state may takes a project towards either of the two different situations as:

i. If project recovers from the problems, it enters into its running (either running or post-running sub-state level) state.
ii. If project does not recover, it directly enters into landing state and may get terminated or closed.

Both situations are shown in Fig. 4.12 and Fig. 5.7 respectively. With good project management approaches a project can be recovered from the hang up state but a large number of projects also get closed without completion. It means many factors affect a project when it is in hang up state. Hang up state may occur during the project life more than once. Each time its behavior is expected to be different.

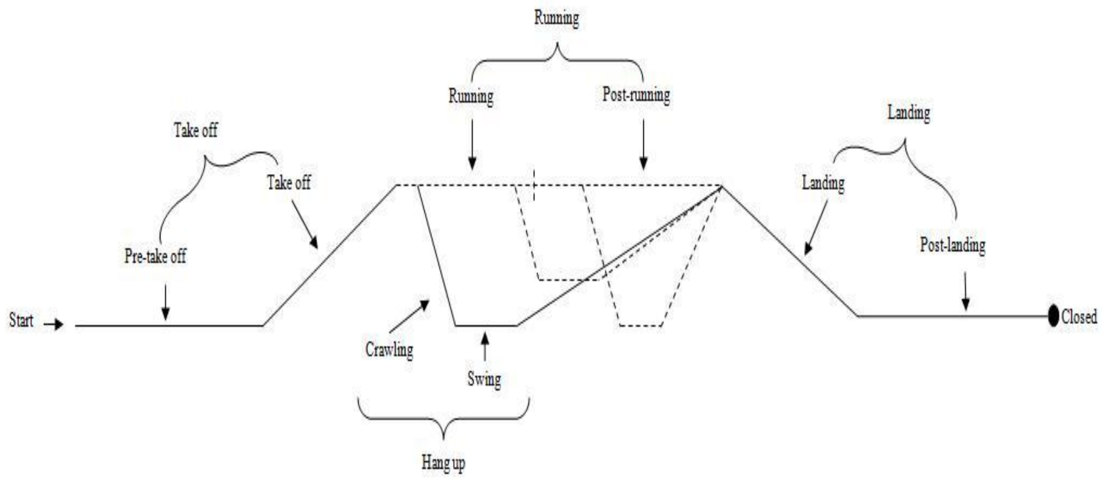

Fig. 5.7 Hang up State Variation

5.3.2 Behavior of Project States

As discussed in the previous section, each project passes through these execution flow states during its lifecycle. The behavior of these project states that has been studied during the case study, its characteristics and response of the project to each state has been presented in the following sections.

5.3.2.1 Takeoff State

Project Sub-states: Pre-Takeoff, Takeoff

Behavior:

1. The takeoff state provided sufficient time to:
   a. Define the project's scope.
   b. Project specifications and requirements.
   c. Setup initial project setup.
2. Allowed team members and client to understand each other.
3. Provided enough time to team members to develop cooperative and friendly environment.
4. Developed a sense of understanding and responsibility.
5. No hurdles in the initial releases.

Characteristics:

i. Accommodative.
ii. Flexible.
iii. No constraints.
iv. Process intensive.
v. Supportive.

Frequency of Occurrence: 1

Project Response: Project entered into running state after completing takeoff state requirements and tasks. Minor problems of mild nature were present.

Findings: Takeoff state existed in the project prominently. It was also observed that:

i. The duration of takeoff and its sub-states may vary from project to project depending upon the type, nature, domain and client's requirements.
ii. The takeoff state provides a base to the project through processes adopted, defined procedures and approaches

followed.

iii. Most likely projects get succeeded with good takeoff.

5.3.2.2 Running State

Project Sub-states: Running, Post-running

Behavior:

i. Problems started in running sub-state.
ii. Management-Critical problems left the project at stake.
iii. Full risk of project termination was realized in the early post-running state.
iv. Running state was full of problems and mismanaged.

Characteristics:

i. Accommodative
ii. Less flexible.
iii. Vulnerable to risks.
iv. Margin of improvement in running sub-state but less in post-running.
v. Deterministic in project success.
vi. Process intensive.
vii. Supportive.

Frequency of Occurrence: 2

Project entered twice in running state at running sub-state level.

Project Response: Maximum time project spent in running state, went into severe problems and from running state it went into hang up state and started crawling (sub-state).

Findings:

i. Running state was the most critical state of the project.
ii. Only very necessary modifications in the processes and approaches were made during running sub-state.
iii. Problems of the running states were resolved.

iv. Good project management practices were adopted to resolve the issues in shorter time.
v. Project successfully passed through the issues of running state and completed it.

5.3.2.3 Hang up State

Project Sub-states: Crawling, Swing, Pre-running

Behavior:

i. Project entered into crawling sub-state in parallel to running sub-state.
ii. Most of the processes adopted and project rehabilitation measures were taken in swing sub-state.
iii. The progress of the project was slowed down.
iv. Swing sub-state allowed adopting light weight processes and modifying existing approaches.
v. In pre-running state project resumed with new processes and project management activities.

Unlike running state, hang up state allowed major modifications in the communication, requirement management, tasks allocation and management processes as well as project management activities.

Characteristics:

i. Less accommodative.
ii. Less flexible.
iii. Risk bearing.
iv. Problem solver.
v. Provides time and space to revamp the processes.
vi. Process intensive.
vii. Supportive

Frequency of Occurred 2 -3 times in the project.

Occurrence:

Project Response:

i. Only in the first occurrence project passed through all of its sub-states. In following occurrences project mainly entered into crawling sub-state and pre-running, not in the swing sub-state.
ii. Project utilized the time in hang up states intelligently to recover from the problems.
iii. Project successfully recovered from the hang up state.

Findings:

i. The project entered into hang up states 2 – 3 times.
ii. Good project management practices and processes during hang up state mostly recover the projects from problems and lead to running state again.
iii. Hang up state in this project was less flexible and did not provide ample time and space to the team to resolve the issues.
iv. But with good project management approach, project recovered from the hang up state and started its normal execution.

5.3.2.4 Landing State

Project Sub-states: Landing, Post-landing

Behavior: Allowed following activities when project entered into it:

i. Major bug fixing.
ii. Minor new requirements and modifications to finalize the application.
iii. Code based completion & handover.
iv. Handing over implementation details.
v. Preparation of manuals.
vi. Changing important login information and passwords.

Characteristics:
i. Accommodative
ii. Flexible
iii. Time constraints.
iv. Supportive.
v. Process intensive.

Frequency of Occurrence: 1

Project Response:
i. No major problems observed during this state.
ii. Project was closed in a normal way without any serious concerns.
iii. The client's consent and approval was included in the whole process during landing state.
iv. Project's progress and closure was according to the requirement of the client.

Findings:
i. Project completed successfully.
ii. Project may enter into landing state through two different channels i.e.
    a. Upon normal completion
    b. Upon incomplete termination
iii. This project entered into landing state in a normal way of completion.
iv. The response of the projects on both a and b do not vary that much during landing state.
v. No major issues and problems were observed.
vi. Client wanted to start another scope of the same project, after completion of this one.

5.3.3 Analysis of Problem Classes Existence

In addition to the analysis of project states and their behavior, the analysis of problem classes, their effect, characteristics and response of project to these classes has also been analyzed. A combined analysis of all three problem classes such as performance minor, management critical and progress limiter has been presented below.

Project States Observed:	Takeoff, Running, Landing, Post-landing
Effect:	Irrespective of their class, type, and severity they do affect the project. Their effect appeared on: a. Team performance. b. Project progress. c. Client’s satisfaction. d. Company’s profile.
Characteristics:	i. Proportionate ii. Associative iii. Controllable iv. Reincarnation
Frequency of Occurrence:	i. Multiple occurrences. ii. More than once cycles appeared during the project.
Project Response:	i. Hindered the progress of the project. ii. Accumulated and adversely affected the project. iii. Over all project growth became sluggish. iv. Project faced the risk of incomplete termination
Findings:	Performance minor, Management critical and Progress limiter class problems were observed during the whole project lifecycle. After completing their first lifecycle they may start another. Multiple lifecycles of problem are present in a project. The duration and intensity of each lifecycle varies among

different phases of a project and among different projects.

All the problems that belong to either class, irrespective of their severity, effect, minor or major type must be avoided and handled right away. Minor problems must not be ignored.

5.3.4 Summary of Project Behavior and States

During takeoff state mainly Performance minor problems have been observed. Management critical problems are mainly associated with running state of the project. Though both types of the problems may exist in landing state but at that time project manager and team members are not that much concerned about because their intensity and frequency is not that much devastating. So problems of both classes are usually not handled except a few.

Takeoff and landing states normally do not have too much problems and uncertainties like running and hang up states. Landing may be of a completed project or incomplete project. In either case a project has to pass through it and complete the necessary requirements of the client during landing state.

Sensitivity and critical importance of running and hang up states make them crucial for the success of a project. As discussed earlier that many factors are involved that affect the project in hang up state. The same is true for takeoff, running and landing state. Unlike takeoff and landing states, they are more critical in running state as well. Hang up state is another state that may occur at anytime in a project. These factors are equally critical for running and hang up states and to somehow for takeoff and landing state.

The analysis supports the existence of these states in a project and problems associated to these states. It has also been found that these problems and other factors affect the duration of the project, length of each state and scope of the project and project schedule.

It has been observed that each project passes through all such issues and problems as well as these states throughout its life span. The nomenclature and taxonomy that has been defined in project states meta-model truly represents the project's behavior and response. It is expected that the project states meta-model is a state-of-art in addressing project development and management issues being a proved prediction and avoidance approach in software engineering and project management.

It has been further analyzed that an extensive study in this regard will help project managers and company executives to defined effective project management practices specific to each project state. It may be termed as micro project management. The project states meta-model and micro project management practices in a combination are found as important for project's success. However, a efforts are required by the research community to develop a proper framework of micro project management practices.

5.4 Process Tailoring Framework

The analysis of software process tailoring framework has been performed with respect to the key processes and project states making it more extensive and explanatory. The overall analysis is comprised of major eight milestones, their modules and components, and number of requirements or tasks completed in their releases as shown in Table 5.5.

Table 5.5 Major Milestones Released in Project ‘A’

<table>
<tr><th>Milestones</th><th>No. of Iterations /Builds</th><th>No. of Modules/ Components</th><th colspan="2">Quick Patch</th><th>Interval between Iterations (Days)</th><th>No. of Requirements/ Tasks</th></tr>
<tr><td rowspan="17">Beta 1</td><td>A (b8074.1)</td><td>2</td><td colspan="2">-</td><td>-</td><td>4</td></tr>
<tr><td>B (b8081.x)</td><td>3</td><td colspan="2">-</td><td>7</td><td>25</td></tr>
<tr><td>C (b8082.x)</td><td>3</td><td colspan="2">-</td><td>7</td><td>14</td></tr>
<tr><td>D (b8083.x)</td><td>3</td><td colspan="2">-</td><td>7</td><td>2</td></tr>
<tr><td>E (b8084.x)</td><td>4</td><td colspan="2">-</td><td>4</td><td>9</td></tr>
<tr><td>F (b8085.x)</td><td>4</td><td colspan="2">-</td><td>7</td><td>18</td></tr>
<tr><td>G (b8091.1)</td><td>4</td><td colspan="2">-</td><td>6</td><td>20</td></tr>
<tr><td>H (b8092.1)</td><td>4</td><td colspan="2">-</td><td>8</td><td>12</td></tr>
<tr><td>I (b8093.1)</td><td>4</td><td colspan="2">-</td><td>7</td><td>14</td></tr>
<tr><td>J (b8094.1)</td><td>4</td><td colspan="2">-</td><td>7</td><td>20</td></tr>
<tr><td>K (b8095.1)</td><td>4</td><td colspan="2">-</td><td>4</td><td>11</td></tr>
<tr><td>L (b8102.1)</td><td>4</td><td colspan="2">-</td><td>10</td><td>13</td></tr>
<tr><td>M (b8103.1)</td><td>4</td><td colspan="2">-</td><td>7</td><td>17</td></tr>
<tr><td>N (b8104.1)</td><td>4</td><td colspan="2">-</td><td>7</td><td>10</td></tr>
<tr><td>O (b8105.1)</td><td>4</td><td colspan="2">-</td><td>7</td><td>19</td></tr>
<tr><td>P (b8111.1)</td><td>4</td><td colspan="2">-</td><td>7</td><td>14</td></tr>
<tr><td>Q (b8112.1)</td><td>4</td><td colspan="2">-</td><td>7</td><td>16</td></tr>
<tr><td>Post Beta 1</td><td>A</td><td>3</td><td colspan="2">-</td><td>6</td><td>7</td></tr>
<tr><td rowspan="2">Beta 1.1</td><td>A</td><td>3</td><td colspan="2">-</td><td rowspan="2">14</td><td>18</td></tr>
<tr><td>B</td><td>3</td><td colspan="2"></td><td>7</td></tr>
<tr><td rowspan="5">Beta 1.2</td><td rowspan="5">A</td><td rowspan="5">3</td><td rowspan="2">1.0</td><td>1.1</td><td>5</td><td rowspan="2">13</td></tr>
<tr><td>1.1.1</td><td>10</td></tr>
<tr><td rowspan="3">2.0</td><td>2.1</td><td>7</td><td>8</td></tr>
<tr><td>1.1</td><td>2</td><td>27</td></tr>
<tr><td>1.1.1</td><td>7</td><td>3</td></tr>
</table>

Table 5.5 Major Milestones Released in Project 'A' (Continued)

Milestones	**No. of Iterations/ Builds**	**No. of Modules/ Components**	**Quick Patch**	**Interval between Iterations (Days)**	**No. of Requirements/ Tasks**
Beta 1.2	B	5	-	In parallel to iteration A, modules and resources were different.	32
	-	3	1.0 For Beta 1.2	7	13
	-	3	2.0 For Beta 1.2	11	31
Release 1.0	-	2	-	14	66
Comment Checker	-	3	-	Parallel	40
Search Engines – Proof of Concept	-	-	-	Parallel	8
Miscellaneous	-	-	-	Parallel	1

The intervals between the releases of milestones range from 7 – 14 days depending on the number of tasks, their complexity and percent effort of each resource on its tasks. Among all the releases, only milestone Beta 1.2 released quick patches as shown in Table 5.5. The complete analysis presented in the following sections is based on the observations made during the release of each milestone as presented in Table 5.5.

Following the tailoring scales as shown in Table 4.6, only the tailoring activities or operations mostly performed as well as found common throughout the project have been selected and presented as shown in Table 5.6. The same approach has been followed during second case study.

Table 5.6 Selected Tailoring Activities in all Projects

	Tailoring Activities	Takeoff		Running		Landing		Hang up		
		Pre-Takeoff	Take-off	Running	Post-running	Landing	Post-landing	Crawling	Swing	Pre-running
Resource Management	+	√	√	√	√			√	√	√
	−		√	√	√	√		√	√	√
	Δ		√	√	√			√	√	√
	⊢			√				√	√	
	⊻			√				√	√	
	⋈						√			
	⊕						√			
Communication (Interaction & Coordination)	+	√	√	√	√			√	√	√
	−		√	√	√			√	√	√
	Δ		√	√	√			√	√	√
	⊢							√	√	√
	⊻							√	√	√
	⋈									
	⊕						√			
Requirement Management: Requirement Gathering & Tracking	+	√	√	√	√			√	√	
	−			√	√	√		√	√	√
	Δ			√	√			√	√	√
	⊢			√	√			√	√	√
	⊻			√				√	√	√
	⋈						√			
	⊕						√			
Requirement Management: Tasks Allocation	+	√	√	√	√			√	√	√
	−		√	√	√	√		√	√	√
	Δ		√	√	√			√	√	√
	⊢		√	√	√			√	√	√
	⊻		√	√	√			√	√	√
	⋈						√			
	⊕						√			

Table 5.7 shows the number of each of the tailoring operation performed on the key processes with respect to the project states. Some tailoring operations have been performed only one time or more than one time in more than one state or sub-states collectively and commonly as shown in Table 5.7 having single value against more than one sub-states.

Table 5.7 Tailoring Activities Performed in Project 'A'

		Tailoring Activities	Takeoff		Running		Landing		Hang up		
			Pre-Takeoff	Takeoff	Running	Post-running	Landing	Post-landing	Crawling	Swing	Pre-running
Resource Management		+	3		5					4	
Resource Management		–			1		5			2	
Resource Management		Δ			2					1	
Resource Management		⊢			1				1		
Resource Management		⊻			1				1		
Resource Management		⋈						2			
Resource Management		⊕						1			
Communication (Interaction & Coordination)		+	2		5					7	
Communication (Interaction & Coordination)		–			2					3	
Communication (Interaction & Coordination)		Δ			3					3	
Communication (Interaction & Coordination)		⊢								1	
Communication (Interaction & Coordination)		⊻								1	
Communication (Interaction & Coordination)		⋈									
Communication (Interaction & Coordination)		⊕						4			
Requirement Management	Requirement Gathering & Tracking	+	1	1	2	3			4		
Requirement Management	Requirement Gathering & Tracking	–			1	1	1			1	
Requirement Management	Requirement Gathering & Tracking	Δ			1	2			2		1
Requirement Management	Requirement Gathering & Tracking	⊢			2	1				2	
Requirement Management	Requirement Gathering & Tracking	⊻			1					2	
Requirement Management	Requirement Gathering & Tracking	⋈						3			
Requirement Management	Requirement Gathering & Tracking	⊕						2			
Requirement Management	Tasks Allocation	+	1		3					1	
Requirement Management	Tasks Allocation	–			2		1			1	
Requirement Management	Tasks Allocation	Δ			2					2	
Requirement Management	Tasks Allocation	⊢			2					1	
Requirement Management	Tasks Allocation	⊻			2					1	
Requirement Management	Tasks Allocation	⋈						4			
Requirement Management	Tasks Allocation	⊕						3			

The reasons to tailor the software development process, the processes tailored, frequency of tailoring operations and the outcome of the overall tailoring performed on key each key process have been described below.

5.4.1 Resource Management

Process States:	Takeoff, Running, Landing, Hang up
Reasons to Tailor:	i. Adopting an effective communication channel, interaction and coordination process. ii. Bridging communication gap. iii. Avoiding ambiguities. iv. Setting priorities. v. Client’s management. vi. Team work.
Processes Tailored:	i. Resource hiring ii. Defining roles iii. Effort distribution iv. Resource transition v. Resource shuffling
Tailoring Operations Performed:	See Table 5.6 and Table 5.7
Frequency of Tailoring:	1 – 2
Repetition:	Yes

Outcome: Produced a set of minimum light weight activities to manage the resources such as:

i. Resource transition
ii. Silent resources
iii. Effort distribution

These finally selected activities replaced other activities and proved to be the best in resource management. Hiring a new resource and its training, appointing hidden or silent resources to overcome the resource limitations over the project and how effectively skilled resources can be utilized on different projects in a company were the beneficial outcome of this set of activities.

5.4.2 Communication, Interaction and Coordination

Process State: Takeoff, Running, Landing, Hang up

Reasons to Tailor:

i. Adopting an effective communication channel, interaction and coordination process.
ii. Bridging communication gap.
iii. Avoiding ambiguities.
iv. Setting priorities.
v. Client's management.
vi. Team work.

Processes:

i. Formal and informal communication process.
ii. Formal and informal sessions.
iii. Structured and unstructured meetings.
iv. Meeting recordings.
v. Meeting minutes.
vi. Document sharing.
vii. Automated communication.

Tailoring Performed:	See Table 5.6 and Table 5.7
Frequency of Tailoring:	2 – 3
Repetition:	Yes

Findings: Communication, interaction and coordination processes were very important in this project. A large number of issues, client's expectations and mismanaged work made it a critical process.

Modifications were repeatedly done during this part. Finally the emphasis was given to the following processes throughout the project:

i. Formal and informal communication channels.
ii. Automated interaction and coordination.
iii. Structured and unstructured meetings.
iv. Meetings recordings and meeting minutes.

Informal approaches were adopted in order to solve the problems of the project, fulfill the requirements of the client and better team coordination, to work in internet time without formalities. As a result project got stable and client remained satisfied with the project progress.

5.4.3 Requirement Management

Process State: Takeoff, Running, Landing, Hang up

Reasons to Tailor:

i. Defining a suitable requirement gathering and tracking process.
ii. Keeping track of the progress of the project.
iii. Avoiding missing requirements.
iv. Full utilization of resources.

v. Tasks completion.
vi. Performance check of the resources.
vii. Project audit

Processes Tailored:

i. Master requirement tracking documentation.
ii. Client’s approval.
iii. Requirement freezing.
iv. Requirement reopen, fixed close and verified close.
v. Project scheduling.
vi. Tasks extraction.
vii. Tasks assignment.
viii. Status reports.
ix. Tasks updates

Tailoring Performed: See Table 5.6 and Table 5.7

Frequency of Tailoring: 1 – 3

Repetition: Yes.

Findings: Tailoring of requirement management and tasks allocation processes was necessary for the project progress and to satisfy the client. The final set of processes and sub-processes that was selected to manage these this phase was the following:

i. Automated requirement gathering and tracking.
ii. Automated project scheduling.
iii. Daily tasks allocation.
iv. Daily feedback and status reports.
v. Daily tasks updates.
vi. Automated document sharing.

This set of processes and their respective sub-processes proved to be the most suitable for this phase and its activities throughout the project life cycle. The project performed good, recovered from the problems, maintained the client's satisfaction level and became precedence for other projects too.

5.4.4 Analysis of Process Tailoring

For validating this part of the framework, proper planning has been made. The activities performed during this validation have been selected based on the requirements of the client, processes, project requirements, project management and team management practices. Sections 5.4.1, 5.4.2, and 5.4.3 explain the reasons of process tailoring, and processes and activities adopted and modified, and effect of process tailoring on the project, and frequency of tailoring (number of times performed). Table 5.8 summarizes the average number of activities tailored as shown in Table 5.7 for each of the tailoring operations, key processes and project states.

Table 5.8 Overall Process Tailoring Performed in Project 'A'

		Key Processes		
		Resource Management	**Communication (Interaction & Coordination)**	**Requirement Management**
Total number of activities	**Total Tailored**	30	31	34+26=60
	Added	12	14	11+5=16
	Deleted	8	5	4+4=8
	Modified	3	6	6+4=10
	Split & Select	2	1	5+3=8
	Merge	2	1	3+3=6
	Shrink	2	0	3+4=7
	Wrap up	1	4	2+3=5
Activities tailored during a state	**Takeoff**	5	4	2+5=7
	Running	8	8	14+7=21
	Landing	8	4	6+8=14
	Hang up	9	15	12+6=18

The numbers of tailoring activities of resource management and communication phases are almost same with the difference of one activity. The highest numbers of activities have been tailored during requirement management process phase which is 60 as shown in Table 5.8. It shows that:

i. Requirement management processes comprises of a large number of processes and activities.
ii. This is the most important process of the project.

The other processes have also been considered important, but from technical aspect of the project, requirement management process of the project is very critical for the project success.

Also, the number of activities tailored such as add, delete, modified, split & select, merge, shrink and wrap up have been found greater in number during requirement management process as compared to the others. During takeoff, running, landing and hang up states of the project, requirement management processes are mostly tailored as shown in Table 5.8. Total 7 activities have been tailored at takeoff state, 21 at running, 14 at landing and 18 at hang up state.

5.4.5 Summary of Process Tailoring

It is found that requirement management process is the most technical and important part of the project and adapting processes to manage it requires more efforts and skills. The least number of activities tailored are for shrink and wrap up, split & select, and merge operations respectively. It shows that these four strategies are very specific to specific situations and requirements of the project and client. Mostly add, delete, and merge activities are performed.

In some cases tailoring has been performed repeatedly and frequently. As presented in sections 5.4.1, 5.4.2, and 5.4.3, the frequency of tailoring in communication process is higher than the requirement management. The communication process is considered as a very generic process and it is always tried to find the best option due to which the frequency of tailoring is higher in this process. On the other hand, when the requirement management process gets stable with good set of sub-processes and activities, it is not recommended to modify it again and again, being considered as a critical process for project's success.

In general, process tailoring is a necessary result oriented practice for projects following agile methodologies running in small and medium sized software development companies enabling the project managers and higher management of the these companies to select a suitable set of processes for their projects.

5.5 Overall Analysis of Case Study-I

Being an outsourced project, the client was offshore and agile based methodologies were being followed by the company. The project started in a good way after getting preliminary requirements from the client and setting up the initial project setup. The project team was interviewed, hired and approved by the client from the existing employees of the company. Soon after its beginning, only in few months problems started appearing. Initially the problems were of mild severity but gradually they started affecting the project progress. During the takeoff sub-state such problems had started. Till the running state of the project they had have been accumulated.

Later in the running sub-state and early post running sub-states the problems had become more severe and client had sent his complaint to the CEO of the company. The mismanagement was on the part of project manager who was unable to handle the situations. He ignored many facts and couldn't realize their importance. Ultimately the project management got weaker and weaker, and problems got more severe. Mainly the client was much concerned about the resource allocation (team member tasks), communication, interaction and coordination with the client and intra-team processes, and requirement gathering and tracking processes. He was not being kept updated with all such key areas. The team members were underutilized, project plans were inconsistent and redundant, back log tasks and bugs were too much. During the meetings with the team members and release of builds to him, he clearly observed this mismanagement and expressed his disliking. It was realized that project will be terminated by the client if this situation prolongs.

To handle the situation, the CEO and a senior project manager intervened. They slowed down the project’s progress for 1-2 months, defined new processes and modified the existing processes. During this whole time, the builds were released to the client with lesser functionalities on decided deadlines. After 2 months the project emerged as totally a new project, the processes were good, and project management had been improved. The earlier deadlines were changed and new milestones were defined. In the mean time, client had also appointed a technical manager at his side, and an experienced project manager was appointed at offshore team site. The new set of processes proved to be very result oriented. The project manager on daily basis improved the processes and activities. Transparency in project management, requirement management, resources management through effective and frequent communication and interaction coordination processes was achieved. Though the transparency was not 100% in case of resource management but client was quiet satisfied with the project progress, plans and schedules. Later on, the team delivered many milestones and project completed in a successful way.

The client as a composite entity was present in the project as client hired a technical manager on hid site. A large number of problems were present in the project that started hang up state of the project which was in parallel to the running sub-state. The problems or risks arose repeatedly. Process tailoring was performed to derive the lightweight processes to overcome the issues. The framework of process tailoring with all its components was totally implemented in this project and successfully improved the project progress and project management through lightweight processes.

The case study shows that application of the framework in the project work brought very positive changes. The project was most likely to be terminated and after using proposed techniques, it was completed successfully. The framework and its components brought revolutionary changes in the project, modified the processes, helped in understanding the client and making improvements in the project management. The positive changes were made in the following major areas of the project as:

5.5.1 Understanding the Client

For the success of a project, it is very necessary to understand the client, perceive his requirements, understanding his point of view clearly, understanding his thoughts and mind reading, and behavior and act or perform accordingly. Through the proposed framework, all these objectives have been successfully achieved. During each and every phase and state of the project, the client was given the foremost preference and his likes, dislikes and required things were always identified, implemented, adopted and delivered. This approach helped in improving 70% - 80% of the project processes.

5.5.2 Requirement Management Process

Getting requirements from the client, converting them into structured documents, defining requirement gathering and tracking processes, updating requirements, changing requirements into tasks and allocation of tasks to the team members are the most important part of project management activities. In agile based environments there lacks such processes. The tailoring framework has also proposed the technique to define a process for requirement management in agile based environment. More than 500 requirements/tasks were completed using this approach and back log and tracking processes were defined through tailoring mechanism and understanding of the client's attitude and behavior.

5.5.3 Communication, Interaction and Coordination

In agile based environments, communication, interaction and coordination among the team members and client is very important. The whole agile building block is standing on this process. Unluckily, for distributed agile based environments [46] there is no such support available in the agile model. The proposed framework has provided a communication and interaction mechanism by identifying the main key roles involved in this process and how they communicate. Identification of these communication roles helped in devising communication process and strategies in agile based projects. In project ‘A’, 3 communication channels, 2 roles, and 3 interaction and coordination sub-processes were defined through the framework. While 3 communication channels among 3 roles of client and project team were successfully defined. Till the project completion these were remained in use with slight modifications throughout. Such processes bridged the gap between client and offshore team and provided geographical transparency.

5.5.4 Resource Management

The team members are the actual resources of a project besides others such as software, hardware etc. The client is the investor, and he always wants to know the actual resources and their tasks. The framework has also defined and modified the resource allocation, tasks allocation and their effort distribution processes through tailoring technique. All the resources were 100% on the project, except project manager whose effort was distributed on two projects. 100% improvement was brought in the resource management processes.

In general, the framework had uplifted the project up to 90%, by defining and modifying the new and old processes respectively. It helped a lot in understanding the client throughout the project, devising strategies in this context and improved the processes for successful completion of the project.

Overall there was 90% improvement in the project from its previous condition. The project was rated as the best project in the company having well defined light weight processes. The same approaches were then followed in other projects too. This is the characteristic of the project that it has the ability to tailor the activities throughout the project life cycle not only once.

5.5.5 Project Management

The big challenges during the whole project such as resolving all the problems, satisfying client, good performance of the team members, project planning and scheduling are always on the part of project manager. Sometimes, such kind of circumstances are created that project manager's inspite of their skills are unable to handle the situations.

During the current project, such situations arose many times. The client's perspective component, project states meta-model and tailoring schema helped project manager in managing the whole project. The practices recommended in the framework are quick and light weight that in shorter time period of about two months, the project management was improved too much and proved very successful throughout the project.

5.6 Summary

The case study findings show that client's perspective is very important and critical for the success of projects running in small and medium sized companies which adopts agile based methodologies. Therefore, it is very important to understand and maintain the client's perspective throughout the project life, whereas resource management, communication, interaction and coordination, and requirement management have been found and verified as the main processes on which client is mostly concerned in agile based software development projects in small and medium sized companies.

Software development projects face various risks and problems throughout their life. These risks and other factors such as client's perspective affect the progress of the projects and determine the behavior and states of the projects such as takeoff, running, landing and hang up. Each project passes through these states throughout its life. These states determine the status and condition of the project with respect to various factors.

The process tailoring framework effectively tailors the three key processes as mentioned earlier with respect to the particular state of the project. The analysis and findings of the case study completely supports the process tailoring framework, its applicability and working in small and medium sized companies.

In Project 'A', the process tailoring has been performed according to the client's perspective. The process tailoring framework helped the project manager to resolve the issues and problems faced by the project, deriving lightweight activities of software development, ensured smooth progress and successful completion of the project. The fulfillment of client's requirements, client satisfaction, lightweight processes and successful completion of the project shows that framework is applicable in small and medium sized companies and provides a lightweight process tailoring approach to these companies.

CHAPTER 6

CASE STUDY – II

Overview

This chapter presents the detailed analysis, comparisons and findings of the second case study. A cross case analysis of all the projects in both case studies has also been presented in this chapter. Furthermore, the results of the questionnaire method have been described. Finally, the schema of process tailoring activities and tailoring framework has been presented.

6.1. Understanding Project Client

The client component, its structure, interaction and role in software development projects has been analyzed and discussed in second case study similarly it has been presented in first case study in chapter 5.

6.1.1. Client's Composite Structure

The roles of the client may vary from project to project. Entities that play role of the client are different in different projects. During the second case study two projects, project 'B' and 'D' have been analyzed for this component. According to the requirement and structure of the case study, the client's composite structure of project 'C' has already been presented in section 5.1 in chapter 5. Table 6.1 shows the observations made on the client's composite structure. The observations made are based on similar elements as presented in section 5.1.

Table 6.1 Observations of Client's Composite Structure-II

Project Name	Client/ Role	Existence	Number of Persons in the role	When Role Started	Role Job	Technical Skills
Project B	CEO	Yes	1	Beginning of the project.	i. Owner of the project. ii. Defining requirements. iii. Setting priorities. iv. Demonstrations. v. Meetings.	Moderate
	CTO	Yes	1	Beginning of the project.	i. Technical issues. ii. Requirement specifications. iii. Technical advice. iv. Upcoming tasks.	Expert
	Tech Lead	Yes	1	Middle of the project.	i. Development. ii. R&D. iii. Expert opinion.	Expert
Project D	CEO	Yes	1	Beginning of the project then off and on.	i. Requirement specifications. ii. Deadlines. iii. Meetings.	Expert
	Tech Lead	Yes	1	Beginning of the project.	i. Requirement specifications. ii. Meetings. iii. Current and upcoming tasks and functionalities. iv. Project's progress. v. Team performance.	Expert

6.1.1.1. Analysis of Client's Composite Structure

In project 'B', three roles of client have been found. The main client was the CEO itself. CEO had also hired a technical resource at its side for the sake of help and technically running the project. The main purpose of this technical resource was a kind of in house development of some components as well as guiding offshore development team. Both offshore team and in house technical resource collaboratively worked on so many task and issues.

An interesting observation has been made that a third technically strong resource was also coordinating, guiding, and helping the offshore development team from the client side. This resource was actually the Chief Technical Officer (CTO) of the offshore vendor company, but he was located in USA and was coordinating with both client and offshore team. He was used to attend all the meeting sessions and providing technical assistance ship to the offshore team. The offshore project team was treating all these three resources as client. Prioritizing their requirements and expectations, and their fulfillment was always being given preference by the offshore team.

In project ‘D’, since the beginning of the project, there remained two main roles, client (CEO) itself, and a technical person. Till the release of beta version of the system, CEO itself attended all the meetings, provided requirements and set the deadlines. The other technical resource was also involved technically with the client and offshore team. Development was done on both sides i.e. by the offshore team and technical person of the client. Making sure of completion of all the functionalities and requirements, code quality was on the part of technical person. Later after the release of beta version, the CEO gave the whole project’s responsibility to his technical resource. Till the end of the project the same format was followed.

6.1.1.2. Summary of Client’s Composite Structure

During the analysis it has been found that the appointment of various resources at the client side was due to the reasons such as:

i. Complexity of the project.
ii. Client is busy in some other project or work.
iii. Scope of the project.
iv. Client is less technical.

The second reason of client being busy in some other project or work was observed in the project D, while i, iii, and iv reasons were observed in project B.

As observed in this case study and as shown in Fig. 5.1 in section 5.1.1.2 in first case study, the structure of the client shows that client is a composite entity and there may be present multiple roles at the client side, each being considered as client by the offshore team.

6.1.2. Client's Interaction Overview

The detailed observations of interaction of various roles of client's with roles at offshore team have been presented in Table 6.2. During the analysis of roles in both projects, very interesting facts have been found as discussed in the analysis part.

Table 6.2 Observations of Client's Interaction-II

Project Name	**Interaction Entities**		**Type of Interaction**	**Frequency of Interaction (No. of times per day)**
	Client	**Project Team**		
Project B	CEO	PM	Direct	1 – 2 / week
	CEO	Team Lead	Direct	1 – 2 / week
	CEO	Developer	Indirect & rare	~ 1 – 2 / month
	CTO (collaborative role)	PM	Direct	1 – 3 / week
	CTO (collaborative role)	Team Lead	Direct	1 – 3 / week
	CTO (collaborative role)	Developer	Direct	1 – 2 / month
	Project Manager	PM	N/A	
	Project Manager	Team Lead		
	Technical Lead/Manager	PM	Indirect	~ 1 – 2 / month
	Technical Lead/Manager	Team Lead	Indirect	~ 2 / month
	Technical Lead/Manager	Developer	Indirect	~ 1 / month
Project D	CEO	PM	Direct	1 – 2
	CEO	Team Lead	Direct	1 – 2
	CEO	Developer	Direct	1 – 2
	Project Manager	PM	N/A	
	Project Manager	Team Lead		
	Technical Lead/Manager	PM	Direct	1 – 2
	Technical Lead/Manager	Team Lead	Direct	1 – 2
	Technical Lead/Manager	Developer	Direct	3 – 5

~ approximate

6.1.2.1. *Analysis of Client's Interaction*

It has been observed that project 'B' was quite a stable project. Therefore, the interaction among various roles of client and project team was not that much frequent as in other projects. The client (CEO) of the project was used to interact with the project manager or team lead only in the case of some urgent matter, issue or problem. Most of the discussions and decisions were taken during the weekly or fortnightly meetings.

On the other hand, interaction with the CTO was more frequent as compared to the client. CTO was interacting and coordinating with all the team members including developers mostly 1 to 3 times in a week. The interesting fact that was observed was the direct interaction of CTO with the developers and even QA engineers throughout the project. We had rarely observed this fact in any other project. The technical lead of the client was also used to contact the project team rarely. The technical lead was more towards development of some of the components which were different from the developers of the offshore team. Therefore, his interaction was indirect and less frequent with the team members. Table 6.2 shows the frequency and type of interaction in all cases of project 'B'.

It is found during the analysis that type and frequency of interaction among various roles of a project depends mainly on the status and progress of the project. Normally projects with good smooth progress have less frequency of interaction as compared to the others with problems and issues. Interestingly, this is not the rule of thumb, scenarios may vary and interaction may be opposite.

The project D was also a stable project with smooth progress and performance of the team members. Despite the minor issues and problems did exist in the project, the frequency of interaction throughout this project was quite high. CEO of the project directly interacted with all the team members till the release of beta version on daily basis. He was used to keep the check on the completion of the tasks and performance of the team. Later on, his appointed technical manager, who was with the project since the beginning, took over the whole project and ran it till its completion. The technical person was very frequently interacting with the whole team and especially the developer i.e. almost 4 times in a day. He was used to remain online on the messenger the whole day. This project had the highest frequency of interaction among client side and offshore team side roles.

6.1.2.2. Summary of Client's Interaction

The analysis found that type and frequency of interaction depends on:

i. The status and progress of the project.
ii. Complexity of the project.
iii. Issues and problems in the project.
iv. Behavior and attitude of the client.

The analysis and discussion of the project supports the roles and their interaction as described in the framework. The roles and their interactions may vary from project to project, but fundamental interactions remain the same as shown in Fig. 5.2 which has been further elaborated as shown in Fig. 6.1 and Fig. 6.2 through sequence diagrams respectively.

The big rectangle labeled as 'par' in Fig. 6.1 shows the repeating interactions and activities that are followed throughout the project lifecycle. The appointment of any role by the client of the project at client side is also shown in Fig. 6.1 labeled as <<creates>>. Fig. 6.2 is an improved version Fig. 6.1 showing each of the complete interaction sequence in the form of blocks of sequences. This illustration helps in understanding sequences of interactions more clearly.

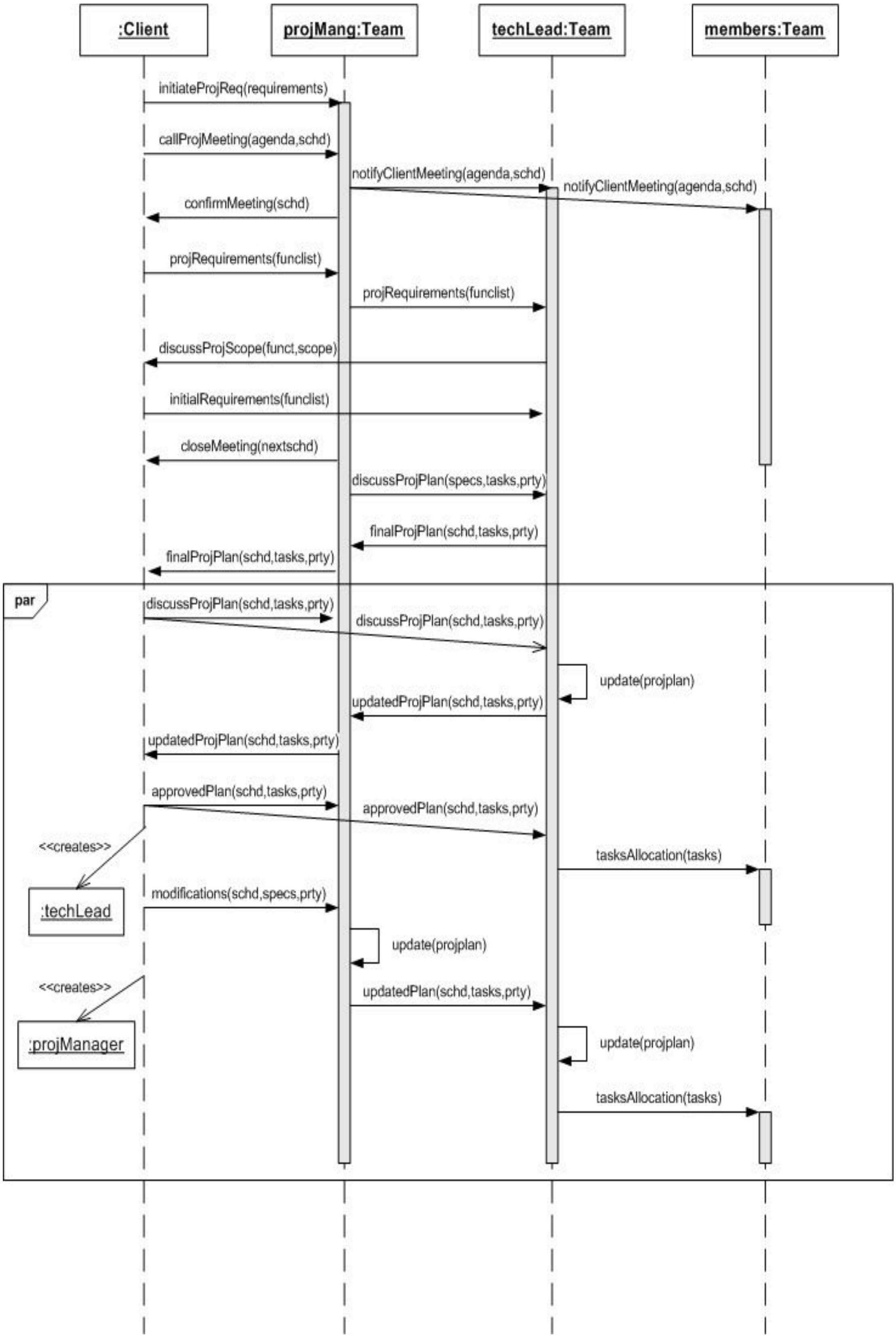

Fig. 6.1 Client's Interaction Sequence-I

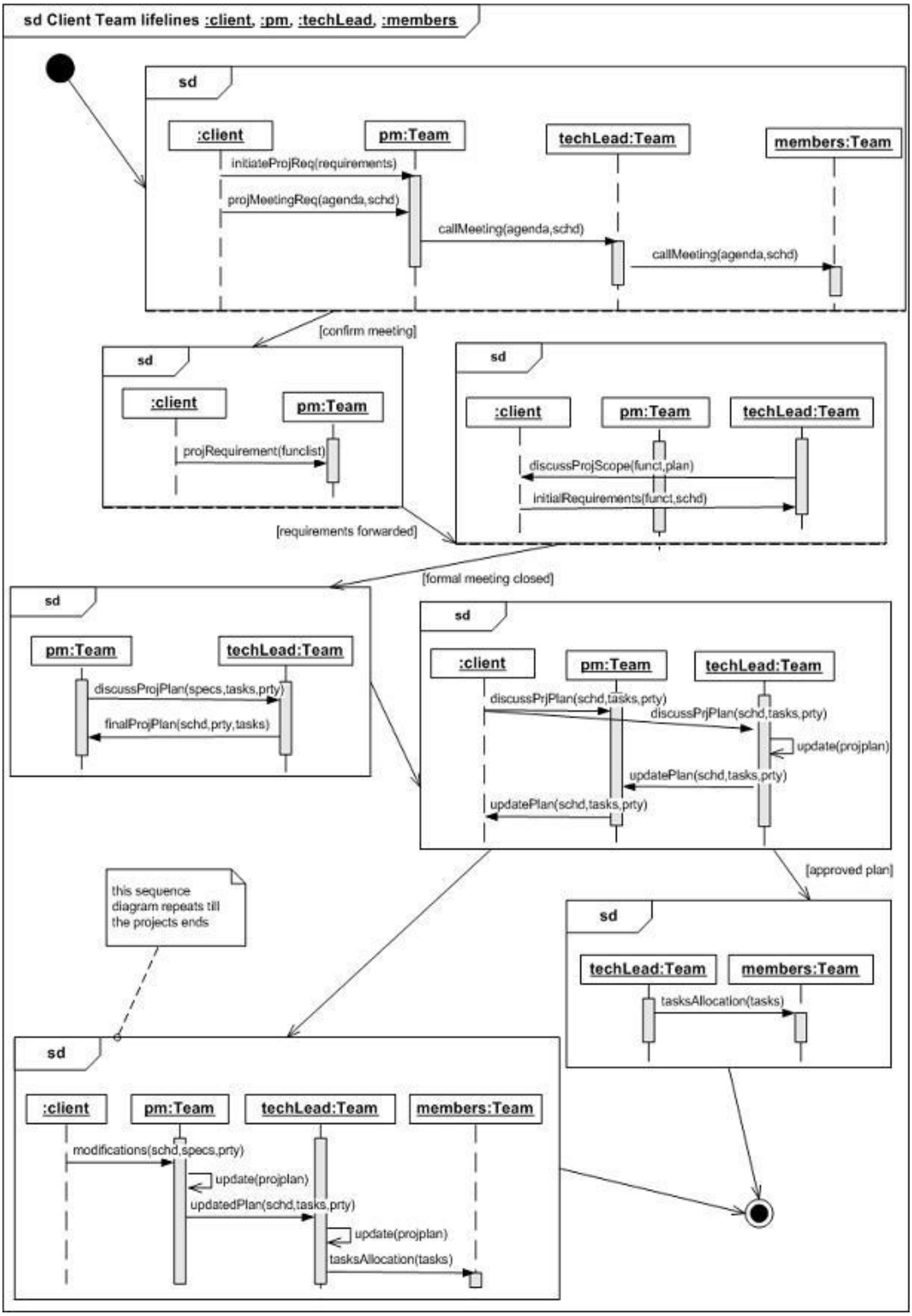

Fig. 6.2 Client's Interaction Sequence-II

6.2. Role of Client's Perspective

The client's perspective in all three projects i.e. project 'B', 'C' and 'D' has been found amazingly different from each other. Though the basic requirements of all three clients have been found almost same but their attitudes, behavior and responses were completely different and unpredictable. The description of analysis has been summarized in Table 6.3 whereas details have been presented in Appendix F.

The processes adopted and documentation maintained as presented in Table 5.3 in section 5.2 (chapter 5) have been the same. Therefore, instead of repeating them Table 6.3 shows only the findings. The client's perspective analysis has been performed on the same set of processes as identified in the first case study.

In project 'B', since the beginning of the project the client was cool, calm, supportive and cooperative. Though the project was stable, but in case of major and minor issues in the project, he never made any complaint. Overall he was satisfied from the project and later his satisfaction level increased more with the good performance of the team and progress of the project as shown in Table 6.3.

Clients in the project 'C' were very difficult to handle. Due to the presence of project managers, tech leads at client side it was hard to convince them on some issues due to their differences of opinions and perspectives. The client in this project was not that much satisfied from the team performance and his dissatisfaction level tend to increase as shown in Table 6.3.

Table 6.3 Client's Perspective Analysis of Projects-II

<table>
<tr><th rowspan="2">Key area
(scenario)</th><th colspan="2">Client's Response before changes in Project</th><th colspan="2">Sharing interval (duration) of Project</th><th colspan="2">Client's Response after changes in project</th></tr>
<tr><th>B</th><th>C</th><th>B</th><th>C</th><th>B</th><th>C</th></tr>
<tr><td>Requirement Gathering and Tracking Document's Sharing with the Client.</td><td rowspan="6">Client was satisfied</td><td rowspan="6">Client was not satisfied</td><td>Weekly</td><td>Daily and
Weekly</td><td rowspan="6">Client became satisfied</td><td rowspan="6">Client tend to become dissa-tisfied</td></tr>
<tr><td>Status Updates</td><td>Daily
Weekly</td><td>Weekly</td></tr>
<tr><td>Tasks Allocation</td><td>Weekly
Daily</td><td>Weekly</td></tr>
<tr><td>Releasing Build Plans and Meetings</td><td>Weekly</td><td>Weekly (releases)

Daily (meetings, informal, irregular)</td></tr>
<tr><td>Resource Allocation and Project Staffing</td><td colspan="2">Start
Need based
Monthly (project staffing)</td></tr>
<tr><td>Frequent Meetings and Updating Client</td><td>Weekly/ monthly (structured, formal)
Biweekly (unstructur ed)</td><td>Daily
1-2 times</td></tr>
<tr><td>Automated Project Management</td><td>N/A</td><td>N/A</td><td>N/A</td><td>Daily</td><td>N/A</td><td>N/A</td></tr>
</table>

N/A – Not Applicable

During the analysis of different scenarios of both projects as presented in Appendix F, the strange behavior of clients in project 'C' has been realized. Despite all the measures taken according to the requirements of the client and processes adopted, the client's could not be satisfied with the overall performance of the team.
As observed in results of project 'C' in 6.2(d) in Appendix F, the unfavorable response of the client was tending to increase day by day. Client had no issues with the processes and approaches used by the project team. With keen observation of these issues, interesting facts were found which have been presented in the following section.

6.2.1. Client's Dissatisfaction Analysis in Project 'C'

Level of Dissatisfaction:	Increasing
Reasons Found:	i. Coding problems. ii. Inconsistent records and data. iii. Bugs found in each build.
Client's Comments:	i. Mistakes of the teams. ii. Carelessness of the team members. iii. Problems in the coding.
Team's Comments:	i. No access on the client's database at server. ii. No access rights on the server. iii. Lengthy procedures to download the data from the server. iv. Inconsistent database, redundant. v. Requirements not clear.
Project's Progress:	Inspite of the problems, the project team successfully completed and delivered all the components upto the client's expectations.
Client's Setup:	The client had already started hiring his own team for the project. He had hired the following persons since the beginning of the project: i. Director of operations (Project Manager role) ii. Director of product development. (Product manager role) iii. Director of engineering (Technical lead role) iv. Developer / Web designer (Developer role)

Summary of the Analysis: The client's comments regarding the performance of the team were found correct. The team made mistakes as mentioned above in each build but the reasons of those mistakes were quite genuine and justified as mentioned above.

The database being inconsistent and redundant always created troubles in the coding. The team was used to download the required records from the server. Always the whole process took 2 – 3 working days. The code was implemented on the downloaded records and, in the mean time records have had been changed in the database. Duplication of records was also a big issue.

Client never gave access to the team members on the server and live database. The project manager of the team brought this into the notice of client many times, but client was not willing to give access on their servers. This inconsistency of the database, always produced bugs and problems in the system.

Problems and issues were unacceptable to the client, on the other hand he was not convinced in the requirement of the project team. With the hiring of the whole team at his site, it was clear that after completion of certain level of the project, he wanted to take over it to start the in house development.

All these factors affected the level of trust and satisfaction between clients and the team members. Ultimately, the CEO of the company, by mutual decision with the clients, closed the project.

Result: i. The project was closed after 75% completion.

ii. Required documentation and code base was handed over.

6.2.2. Analysis of Client's Perspective

The analysis of project 'B' provides a detailed insight into the behavior and attitude of the project. It also highlights that how it affects the project progress. The analysis shows that the client of project 'B' was very cooperative and wise person. He completely agreed with the processes and activities of the project team. He found those processes and activities suitable for his project. He was not very demanding. Except a few changes, he never raised any objection or complaint.

The project manager had already understood the behavior of the client and he maintained the minimum requirement of the client. From requirement gathering to the delivery of the system, there were not observed any major issues. Though a critical major issue crashing a system due to malfunctioning of a library file occurred, but the trust of the client on the team never allowed him to be offensive or problematic. It took more than two months to resolve the problem, but the cooperation of the client was excellent during that time.

The reason behind it was that project team had already won the trust and confidence of the client, by understanding and maintaining his perspective throughout the project. Therefore, even in the presence of minor and major issues, client remained cooperative. Ultimately, the project completed successfully and product had been launched.

On contrary, the scenarios in the project 'C' were very different and unrealistic. From the scenarios presented in the above examples, it was clear that though all the processes and activities were according to the requirement of the client, but client was not satisfied. The team had understood his perspective, though they had fulfilled it somehow, but they were unable to maintain it throughout the project. It is fact that mistakes were also in the process defined by the client, but project team was unable to coop with it.

Observations as presented in key areas 6.2(a) to 6.2(g) in Appendix F show that client's requirements have been fulfilled upto the maximum. Only the few scenarios as presented in client's dissatisfaction analysis in section 6.2.1 let the whole project team down. Clients always encouraged the team, showed his satisfaction on the performance, but under hand they had finally decided to take over the project. The team inspite of all the efforts could not take up their concerns and got failed to avoid it. This failure of understanding the client resulted into the termination of the project.

6.2.3. Summary of Client's Perspective

The analysis shows that understanding the client's perspective and maintaining it throughout the project is very important for the project. The success of the project in small and medium size software development companies following agile based methodologies is mainly based on the client's perspective which have been found as the most important and critical success factors in projects in these companies.

6.3 Behavior of Software Development Project and Project States Meta-model

The project states meta-model has been validated through the analysis of both project 'B' and project 'C'. The problems faced by the projects, response of the projects and project states have been analyzed during the case study. Both Table 6.4 and Table 6.5 summarize the analysis of project states meta-model for which the detailed analysis has been presented in Appendix G.

Table 6.4 Project ‘B’ States Analysis

State/ Sub-state		Duration	Description	Activities	Problem class/Factor	Problem Impact	Project Status
Takeoff	Pre-takeoff	0.5 months	Exists	Initial system setup, Resource hiring, Requirement gathering, finalizing and project scope defined. Preliminary meetings	Performance Minor	Low	Stable
	Takeoff	1 month	Exists	Pre-beta tasks, Resource allocation, Resource shuffling Meetings, Requirement gathering	Performance Minor	Low	Stable
Running	Running	3 months	Exists (problem started)	Beta version, Server updates, New components	Performance Minor Client’s Perspective	Medium	Stable
	Post running	6 months	Exists (major problem in application library file)	Post beta versions, Major functionalities, Code factoring, Fixing library file crashes	Management Critical Client’s Perspective	High	Instable
Hang up	Crawling	1 week	Exists, but was not prominent	Two major issues existed and fixed i.e. a) Picture not loading b) Library file crashes The second problem took more time than the first one. The problem (a) was fixed in parallel to the actual tasks. During fixing (b) the major development was stopped	Management Critical	High	Instable
	Swing	5 months (parallel to running state)	Exists in parallel to the running state. Minor tasks and fixes were completed. Major tasks were stopped for the time being		Progress Limiter Client’s Perspective	High	Instable

Table 6.4 Project 'B' States Analysis (Continued)

State/ Sub-state		Duration	Description	Activities	Problem class/Factor	Problem Impact	Project Status
	Pre running	1 months (minor tasks still performed during hang up state)	Exists The problem was resolved and being tested against existing application	(continued) (b) the major development was stopped.	Management Critical	Medium	Stable
Landing	Landing	1 month	Exists	Major and minor fixes Modifications. Code base handover	Performance Minor	Low	Stable
	Post landing	2 weeks	Exists	Documentation Server side updates and passwords handover	N/A	N/A	Stable

N/A – Not Applicable

Table 6.5 Project 'C' States Analysis

State/ Sub-state		Duration	Description	Activities	Problem class	Problem Impact	Project Status
Take off	Pre-takeoff	0.5 months	Exists	Initial system setup, Team allocation, Requirement gathering, Preliminary meetings	Performance Minor Client's Perspective	Medium	Stable
	Takeoff		Exists	Newsletter releases, Meetings, Requirement gathering	Performance Minor Client's Perspective	Medium	Stable

Table 6.5 Project ‘C’ States Analysis (Continued)

State/ Sub-state		Duration	Description	Activities	Problem class	Problem Impact	Project Status
Running	Running	6 months	Exists (problems started)	Final builds releases, Data downloading, Major and Minor fixes	Performance Minor Management Critical Client’s Perspective	High	Stable
	Post running		Exists (major problem)		Management Critical Progress Limiter Client’s Perspective	High	Instable
Hang up	Crawling	No	Does not exist.	N/A	N/A	N/A	Instable
	Swing	5 months (parallel to running state)	Exists in parallel to the running state. In parallel to major and minor fixes, actual development work was not stopped.	Major and Minor Bug fixes, Code fixes. Development of new Builds.	Performance Minor Progress Limiter Client’s Perspective	High	Instable
	Pre running	No	Does not exist.	N/A	N/A	N/A	Stable
Landing	Landing	1 week	Exists	Newsletter release, Code handover Documentation Passwords changing	Performance Minor	Low	Stable
	Post Landing	3-4 days	Exists		N/A	N/A	Stable

N/A – Not Applicable

6.3.1 Analysis of Project 'B' States

The project states of both projects have been analyzed mainly on problem existence, and client's perspective factors. The behavior of project states has been explained, and effect and characteristics of three main problem classes have been presented for both of the projects.

Takeoff state in the project was comprised of 1.5 months and running state was comprised of 9 months. Project entered into hang up state twice in its life and overall spent 6 months and 1 week in this state while landing state was comprised of 1 month and 2 weeks. Swing sub-state in this project was executing in parallel to the running state due to the supportive behavior of the client regarding problems and issues. Table 6.4 shows that all the states of project states meta-model were present in the project 'B' with variable durations.

6.3.1.1 Impact of Problem Classes on Project 'B' States

The problems in the project 'B' had started early during takeoff state. The progress of the project though was not affected by it. The same scenarios were present during running state. Due to two major problems, the project went into hang up state twice during its life but recovered from it. As shown in Table 6.4, the majority of the problems in project 'B' were performance minor with low impact on project progress. Whereas, management critical problems were present having high impact. The progress limiter class problems were present only during hangup state due to which project entered into this state. From the analysis it was observed that the project team had achieved the client's perspective since the beginning of the project. Therefore, in case of hang up state, the client of the project still had trust and confidence on the team's abilities to resolve the problems. The project team successfully resolved the problems and put the project again on the smooth path. Duration of each state has been presented in Table 6.4.

6.3.1.2 Impact of Client's Perspective on Project 'B' States

While comparing the problems, the problems of project 'B' were more severe than those of project 'C' where client was not satisfied with the approach of the project team. The attitude of the client was inflexible in certain matters that left the team with unresolved issue.

The client's perspective approach was used in both of the projects. The outcome though is quite different in both cases. During the project 'B', the client was not demanding too much. Though he wanted his work completion well in time, but he never put pressure on the project team. The project manager adopted the processes keeping in view the behavior, response and expectations of the client. The minimum set of light weight processes such as requirement management, development and feedback were proved to be very successful. Throughout the project, the client remained cooperative, made discussions with the team, agreed with them on their point of views and requirements. He always gave weightage to the team's point of view instead of imposing his own, but he had also prioritized his work in this context.

Due to such a cooperative attitude and response of the client, the project team worked in a very relaxed environment and successfully completed the project. The team had established very good working relationship with the client that proved to be very beneficial for the project. This positive and cooperative attitude of client remained persistent throughout the project. Even during hang up states when there was a major problem in the project, client gave free hand to the team members to solve the problem. Such that the problems were resolved during the crawling sub-state and project entered directly into pre-running sub-state and then started normal execution as shown in Fig. 5.5.

The analysis of project 'B' shows that client's perspective plays a very critical role in the success of the project. Understanding it and creating good working relationship with the client at the early stage of the project is necessary for the project's success and progress.

The project ‘B’ passed through all the states and its hang up state appeared as a separate state than the running state. All the states in project ‘B’ were of normal duration as shown in Table 6.4 except hang up which took more time to finish. The project ‘B” entered into hang up state twice throughout its life and recovered successfully as shown in Fig. 4.12 and Fig. 5.5.

During the project’s states analysis, the states of the project ‘B’ were found more flexible, accommodative and supportive due to the attitude of the client. The states provided enough time and support to resolve the problems. Cooperative and friendly environment and good relations with the client, helped the project team to resolve the problems and settle all the major issues.

6.3.2 Analysis of Project ‘C’ States

As shown in Table 6.5, the running state was the largest state comprising of 6 months out of which project remained in hang up state for 5 months after which within 1.5 weeks it was closed. Though project entered in swing sub-state (hangup) which overlapped the running state due to which it never entered into crawling and pre-running sub-states. Despite the uncertainties in the project and these variable durations, Table 6.5 shows that all the states were present in project ‘C’.

6.3.2.1 Impact of Problem Classes on Project ‘C’ States

The impact of performance minor problems in project ‘C’ was medium and in some cases was high. Management critical and progress limiter class problems throughout the project were of high impact. Due to which project during post running, crawling, and swing sub-states remained instable. The project manager and team members tried their best to understand the clients, clarifying them their problems and proposed solutions, but non cooperative behavior of the client always let them down. The client was never convinced with the approach of the team, while project team was also not satisfied with the response of the client on critical technical matters. It led the project to enter its hang up state during the running state. The hang up state started early during the running sub-state and overlapped the running state as shown in Fig. 5.6.

Though during later states project was stable as shown in Table 6.5 but it could not be recovered from the hang up state and was closed without completion by the CEO and client through a mutual decision.

6.3.2.2 Impact of Client's Perspective on Project 'C' States

The project team could not understand the client's expectations due to their uncertain and rigid attitudes. It has been observed that if client would have had been agreed with the project manager on the solutions of the problems as he suggested, then the team could have resolved all the problems successfully.

In project 'C' the client's behavior was the main hurdle for the project manager to understand the client's perspective. In addition to the response, the client's behavior itself is an important element of client's perspective and is important in understanding the client's perspective. The client's perspective model as derived from the analysis is shown in Fig. 6.3.

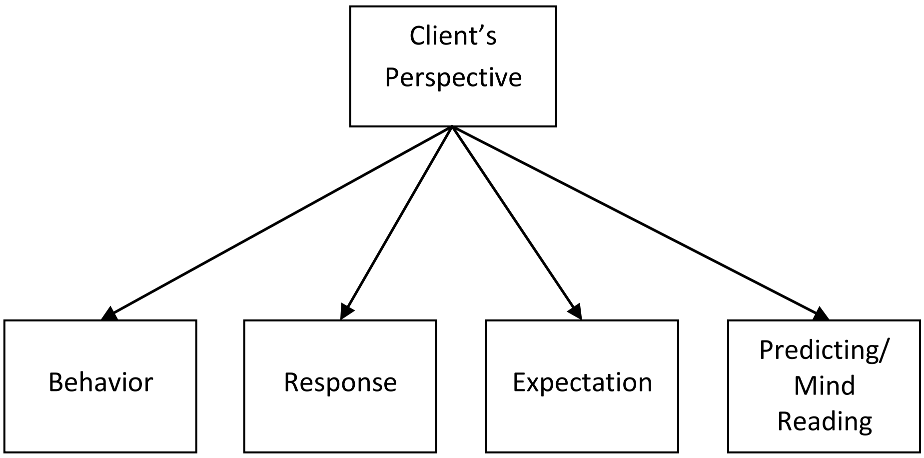

Fig. 6.3 Client's Perspective Model

The project 'C' had never entered into the pre running state and was directly closed during the hang up state. Therefore, pre running state never existed in project 'C' as shown in Fig. 5.7

The states of project ‘C’ were inflexible to make modifications in the processes or procedures. The apparently friendly but inflexible and rigid attitude of the client was the main reason behind the unsupportive behavior of the project states. Insipte of the problems being minor, those could not be resolved. These unhealthy developing scenarios ultimately lead to the incomplete closure of the project.

Following section explains the behavior and characteristics of the project states in both project ‘B’ and project ‘C’. Behavior is actually the response of the state to different scenarios. Characteristics have been described in terms of attributes or properties of that state in response to their particular behaviors.

6.3.3 Behavior of Project States

Similar approach as followed in section 5.3.2 in chapter 5 has been followed to analyze the behavior of project states.

6.3.3.1 Takeoff State

Project Sub-states:	Pre-Takeoff, Takeoff	
Behavior:	Project B:	i. Provided enough time for initial development setup. ii. Provided cooperative and friendly environment. iii. Provided enough time to establish the understandability and good working relationship with the client. iv. No hurdles in the initial releases.
	Project C:	i. Provided less time for initial development setup. ii. Provided non cooperative but friendly

environment.

iii. No enough time was available for understandability and good working relationship with the client.

iv. A few hurdles in the initial releases.

Characteristics:	Project B:	Accommodative. Flexible. No constraints. Less process intensive. Supportive.
	Project C:	Non accommodative. Inflexible. Few constraints. Process intensive. Unsupportive.
Frequency/ Occurrence:	Project B:	1
	Project C:	1
Project Response:	Project B:	Project entered into running state after successful completion of takeoff state.
	Project C:	Project entered into running state along with the problems started facing during takeoff state.
Findings:	Project B:	i. Takeoff state existed in the project prominently. ii. Takeoff state provided strong basis to the project. iii. Most likely projects got succeeded with good takeoff.

Project C: Takeoff state was for shorter duration, therefore:

i. Did not provide a strong base to the project.

ii. Made the reason of project failure.

6.3.3.2 Running State

Project Substates: Running, Post-running

Behavior:

Project B:

i. Provided enough time for releases, bug fixing and modifications.

ii. Did not create hurdles to achieve the milestones.

iii. Helpful in establishing processes.

iv. Enhanced good working relationships with the client.

v. Provided margin of improvements.

Project C:

i. Provided very less time for releases, test cycles and modifications.

ii. Created time constraint hurdle to achieve the milestones.

iii. Not helpful in process adoption and modification.

iv. Could not develop good working relationships with the client.

v. Did not provide margin of improvements.

Characteristics:

Project B:

Accommodative.

Flexible.

Supportive.

Less vulnerable to risks.

Moderate process intensive.

	Project C:	Non accommodative. Inflexible. Non supportive. Vulnerable to risks. Moderate process intensive.
Frequency/ Occurrence:	Project B:	2 Project entered twice into running state, at post running sub-state level.
	Project C:	1
Project Response:	Project B:	Project spent maximum time in the running state. Did not face critical problems and entered into hang up state twice from the running state. After successful completion, it entered into landing state towards its closure.
	Project C:	Project spent its whole life in the running state after takeoff. From running state it was directly closed.
Findings:	Project B:	i. Running state was the longest state of the project. ii. Processes got established and matured during this state. iii. Good trust relationship and confidence built up with the client. iv. Stability in running state led to the stable project.

	Project C:	i.	Project spent almost whole life in running state.
		ii.	Processes got established during this state.
		iii.	Processes could not be matured due to the client's attitude.
		iv.	Project team lost the trust relationship with the client.
		v.	Instability tend to increase.

6.3.3.3 Hang up State

Project Sub-states:	Crawling, Swing, Pre-running		
Behavior:	Project B:	i.	Provided enough time to resolve the problems.
		ii.	Supported the slower development approach.
		iii.	Got support from the client.
		iv.	Supported the effort redistribution of the resources.
		v.	Did not create any kind of hindrance in the existing approach.
		vi.	Let the team resolve the problems with relaxed environment and support of the client.
	Project C:	i.	Ran in parallel to the running state.
		ii.	No time was available to the developers to resolve the critical issues.
		iii.	Provided enough time to resolve the minor or major, but not the critical problems.
		iv.	Did not let the processes be matured.

		v. Did not help to establish trust relationship with the client.
Characteristics:	Project B:	Accommodative. Flexible. No constraints. Less process intensive. Supportive. Less vulnerable.
	Project C:	Non accommodative. Less flexible. Major constraints. Moderate process intensive. Less supportive. Vulnerable.
Frequency/ Occurrence:	Project B:	2
	Project C:	1, but continued.
Project Response:	Project B:	i. Project resolved the problems during hang up state. ii. Successfully recovered from this state and started running again. iii. Project supported the approach of slowing down the actual development tasks and limiting it to independent minor tasks.
	Project C:	i. Problems could not be resolved. ii. Project could not be recovered from this state. iii. Project did not complete its life.

iv. Project remained in the hang up state till its end.

Findings:	Project B:	i. Good project management practices and processes during hang up state helped in recovering from this state. ii. The good trust and confidence relationship of running state with the client supported a lot during the hang up state.
	Project C:	i. Project manager could not be succeeded in winning client's trust and creating good working relationship. ii. Lack of trust and confidence relationship with the client existed throughout the hang up state. iii. This was the major hindrance in recovering the project from hang up state. iv. The project could not be recovered.

6.3.3.4 *Landing State*

Project Sub-states:	Landing, Post-landing	
Behavior:	Project B:	i. Provided enough time to: a) Handing over implementation details. b) Hand over login information and passwords. c) Minor changes and fixes. d) Handing over code base. ii. Created no more problems.

	Project C:	i. Did not create any major issues and problems. ii. Provided enough time to hand over the code base, and documentation to the client.
Characteristics:	Project B:	Accommodative Flexible No constraints. Less process intensive. Supportive. Un vulnerable.
	Project C:	Accommodative Flexible No constraints. Less process intensive. Supportive. Un vulnerable.
Frequency/ Occurrence:	Project B:	1
	Project C:	1
Project Response:	Project B:	i. Project successfully completed all the remaining tasks. ii. Hand over was successfully completed. iii. Project was successfully completed and closed.
	Project C:	i. Project successfully completed all the requirements of the client related to project hand over tasks. ii. Project was stopped and closed but without full completion.

Findings:	Project B:	i.	During landing state no major problems or issues were present.
	Project C:	ii.	No major processes were adopted.
		iii.	Resources got free from the project

6.3.4 Analysis of Problem Classes Existence

The analysis of effect, characteristics, frequency of occurrence and response of both projects to these problem classes has been presented in the following section. The analysis has been presented with respect to all three classes together.

Project States Observed:	Takeoff, Running, Landing, Post-landing		
Effect:	Project B:	i.	Team performance.
		ii.	Individual performance.
		iii.	Working environment
	Project C:	i.	Project management.
		ii.	Team performance.
		iii.	Project progress.
		iv.	Client's satisfaction and trust.
		v.	Working environment
		vi.	Company's profile.
Characteristics:	Project B:	i.	Proportionate
		ii.	Dependent and Independent
		iii.	Controllable
	Project C:	i.	Proportionate
		ii.	Dependent and Independent
		iii.	Associative
		iv.	Uncontrollable
		v.	Reincarnated.

Frequency of Occurrence:	Project B:	Low.
	Project C:	High, multiple times, repeating.
Project Response:	Project B:	i. Project progress was not affected. ii. Project successfully resolved all the problems.
	Project C:	i. Hindered the progress of the project. ii. Accumulated and adversely affected the project. iii. Over all project growth was greatly affected. iv. Problems could not be resolved.
Findings:	Project B:	i. All three types of classes of problems were present in the project. ii. With good project management practices problems were under control and resolved. iii. Good project management practices avoided many problems before time.
	Project C:	i. Project management was though good, but client's behavior was unexpected and non cooperative. ii. Problems could not be resolved. iii. Even minor problems, converted into progress-limiter ones. iv. Project progress was adversely affected

6.3.5 Summary of Project Behavior and States

It has been found during the analysis that the problem classification directly or indirectly affects the team performance, individual performance and project's progress. These are also responsible for the happiness and satisfaction of the client. The severity and intensity of these problems depends on the proportion of other problems and their dependency on other components as well as each other. Whatever are the reasons and effects, they can be controlled.

The analysis supports that these states do exist in all the projects. It is possible that depending upon varying scenarios that occurs in the projects, any state my not exist, some may be repeating and some may be quite longer or shorter. The same research questions arise here as have been described in the first case study. The case study was successful and has validated the project states meta-model through deeper and extensive analysis of the projects.

6.4 Process Tailoring Framework

The process tailoring component is validated through the analysis of both projects ‘B’ and ‘C’. The major milestones achieved during the tailoring process, number of tasks, and numbers of iterations of these milestones are shown in Table 6.6.

Table 6.6 Major Milestones Released in Project ‘B’

Milestones		**No. of Iterations/ Builds**	**No. of Modules/ Components**	**Interval between Iterations (Days)**	**No. of Requirements/Tasks**
Pre Beta		16	3	7 – 20	84
Beta 1.0		6			46
Beta 1.2	Air Client Beta 1.0	8	1	12 – 30	60
	Air Client Beta 1.1	1	1	120	3
	Air Client Beta 1.2	1	2	16	24
Open Fire upgrading		2	1	10	-
Library File Integration at Local Server		1	Variable	-	-
Library File Integration at Production Server		1	Variable	60	-
Transition to New Library		2-3	2	90	-
Post Beta		1	1	-	31

After the release of beta 1.0 version of the application, there was faced the major problem of crashing of a library file. It took around 3-4 months to fix the issue. During that time another component of adobe air client was completed and other major and minor modifications and fixes were made. After post beta the final version was released.

Table 6.7 Major Milestones Released in Project 'C'

Milestones	No. of Modules/ Components	Interval between Iterations (Days)	No. of Requirements/Tasks
Newsletter # 3	1	-	7
Newsletter # 4	1	12	27
Newsletter # 5	1	10	27
Newsletter # 6	1	50	10
Build 1.0	3	-	19
Build 1.1	2	14	29
Build 1.2	3	10	37
Build 1.3	3	7	88
Build 1.4	1	9	31
Build 1.5	3	10	24
Build 1.6	5	15	71
Build 1.7	1	16	6
Build 1.8	1	24	25
Friends Invite Build 1.0	9	20	20
Build 1.9	4	21	182
Newsletter # 11	1	8	5
Build 1.10	1	6	7

The project 'C' was started with the development and release of newsletter # 3. All the build were of variable durations and tasks. After the release of build # 10, the project was closed. The client had started in house development of the project by his own team. The process tailoring of both projects i.e. project ‘B’ and ‘C’ has been described in following section. Table 6.8 and Table 6.9 show the number of activities tailored for project ‘B’ and project ‘C’ respectively.

Table 6.8 Tailoring Performed in Project ‘B’

<table>
<tr><th colspan="2" rowspan="2"></th><th rowspan="2">Tailoring Activities</th><th colspan="2">Takeoff</th><th colspan="2">Running</th><th colspan="2">Landing</th><th colspan="3">Hang up</th></tr>
<tr><th>Pre-Take off</th><th>Take-off</th><th>Running</th><th>Post-running</th><th>Landing</th><th>Post-landing</th><th>Crawling</th><th>Swing</th><th>Pre-running</th></tr>
<tr><td colspan="2" rowspan="7">Resource Management</td><td>+</td><td colspan="2">3</td><td colspan="2">4</td><td></td><td></td><td>1</td><td>4</td><td>1</td></tr>
<tr><td>−</td><td></td><td>1</td><td colspan="2">3</td><td>2</td><td></td><td colspan="3">1</td></tr>
<tr><td>Δ</td><td></td><td>2</td><td colspan="2">3</td><td></td><td></td><td colspan="3">3</td></tr>
<tr><td>⊢</td><td></td><td></td><td>1</td><td></td><td></td><td></td><td colspan="2">2</td><td></td></tr>
<tr><td>⊻</td><td></td><td></td><td>1</td><td></td><td></td><td></td><td colspan="2">2</td><td></td></tr>
<tr><td>⋈</td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td></tr>
<tr><td>⊕</td><td></td><td></td><td></td><td></td><td></td><td>2 – 3</td><td></td><td></td><td></td></tr>
<tr><td colspan="2" rowspan="7">Communication (Interaction & Coordination)</td><td>+</td><td>2</td><td colspan="3">3</td><td></td><td></td><td colspan="3">2</td></tr>
<tr><td>−</td><td></td><td colspan="3">2</td><td></td><td></td><td colspan="3">1</td></tr>
<tr><td>Δ</td><td></td><td colspan="3">1</td><td></td><td></td><td colspan="3">2</td></tr>
<tr><td>⊢</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">1</td></tr>
<tr><td>⊻</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">3</td></tr>
<tr><td>⋈</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>⊕</td><td></td><td></td><td></td><td></td><td></td><td>3</td><td></td><td></td><td></td></tr>
<tr><td rowspan="14">Requirement Management</td><td rowspan="7">Requirement Gathering & Tracking</td><td>+</td><td>1</td><td>1</td><td colspan="2">2</td><td></td><td></td><td colspan="2">3</td><td></td></tr>
<tr><td>−</td><td></td><td></td><td colspan="2">1</td><td>2</td><td></td><td colspan="3">1</td></tr>
<tr><td>Δ</td><td></td><td></td><td colspan="2">2</td><td></td><td></td><td colspan="2">1</td><td></td></tr>
<tr><td>⊢</td><td></td><td></td><td colspan="2">1</td><td></td><td></td><td colspan="3">1</td></tr>
<tr><td>⊻</td><td></td><td></td><td>2</td><td></td><td></td><td></td><td colspan="3">2</td></tr>
<tr><td>⋈</td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td></tr>
<tr><td>⊕</td><td></td><td></td><td></td><td></td><td></td><td>3</td><td></td><td></td><td></td></tr>
<tr><td rowspan="7">Tasks Allocation</td><td>+</td><td>2</td><td colspan="3">3</td><td></td><td></td><td colspan="3">3</td></tr>
<tr><td>−</td><td></td><td></td><td colspan="2">2</td><td>1</td><td></td><td colspan="3">2</td></tr>
<tr><td>Δ</td><td></td><td colspan="3">2</td><td></td><td></td><td colspan="3">1</td></tr>
<tr><td>⊢</td><td></td><td colspan="3">1</td><td></td><td></td><td colspan="3">1</td></tr>
<tr><td>⊻</td><td></td><td colspan="3">2</td><td></td><td></td><td colspan="3">2</td></tr>
<tr><td>⋈</td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td></tr>
<tr><td>⊕</td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td></tr>
</table>

The process tailoring in project ‘B’ was comprised of very few activities. The reason was the stability in the project. There were not a number of changes required to be made in the existing processes, sub-processes and activities. Also, not even a single activity was modified at pre-running sub-state during the requirement gathering and tracking processes.

Also shrink operation was not performed on any activity during landing state of communication process. The analysis shows that as client was much satisfied with the performance of the team members and progress of the project, therefore, major changes in the existing processes were not made.

Table 6.9 Tailoring Performed in Project ‘C’

<table>
<tr><th rowspan="2" colspan="2"></th><th rowspan="2">Tailoring Activities</th><th colspan="2">Takeoff</th><th colspan="2">Running</th><th colspan="2">Landing</th><th colspan="3">Hang up</th></tr>
<tr><th>Pre Takeoff</th><th>Take off</th><th>Running</th><th>Post running</th><th>Landing</th><th>Post landing</th><th>Crawling</th><th>Swing</th><th>Pre running</th></tr>
<tr><td rowspan="7" colspan="2">Resource Management</td><td>+</td><td colspan="2">6</td><td colspan="2">1</td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>−</td><td colspan="2">4</td><td colspan="2">1</td><td>1</td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>Δ</td><td></td><td colspan="3">1</td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>⊢</td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td>0</td><td></td></tr>
<tr><td>⊻</td><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td>0</td><td></td></tr>
<tr><td>⋈</td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td></tr>
<tr><td>⊕</td><td></td><td></td><td></td><td></td><td></td><td>4</td><td></td><td></td><td></td></tr>
<tr><td rowspan="7" colspan="2">Communication
(Interaction & Coordination)</td><td>+</td><td>4</td><td colspan="3">2</td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>−</td><td></td><td colspan="3">1</td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>Δ</td><td></td><td colspan="3">1</td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>⊢</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>⊻</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td></tr>
<tr><td>⋈</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>⊕</td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td></tr>
<tr><td rowspan="7">Requirement Management</td><td rowspan="7">Requirement Gathering & Tracking</td><td>+</td><td>3</td><td colspan="3">1</td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>−</td><td></td><td></td><td colspan="2">1</td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>Δ</td><td></td><td></td><td colspan="2">1</td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>⊢</td><td></td><td></td><td colspan="2">1</td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>⊻</td><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td>0</td><td></td></tr>
<tr><td>⋈</td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td></tr>
<tr><td>⊕</td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td></tr>
</table>

Table 6.9 Tailoring Performed in Project ‘C’ (Continued)

		Tailoring Activities	Take off		Running		Landing		Hang up		
			Pre take off	Take off	Running	Post running	Landing	Post landing	Crawling	Swing	Pre running
	Tasks Allocation	+	2	1						1	
		−			1					1	
		Δ		1						1	
		⊢		1						1	
		⊻		2						2	
		⋈						1			
		⊕						1			

The results of process tailoring during project ‘C’ were interesting and amazingly different from other projects. Throughout the project only internal resource shuffling was done. During the project the processes and activities adopted during the takeoff state were hardly modified throughout the project. Majority of the activities were tailored only once.

Interestingly, the deletion during takeoff state of resource management process was made only in this project, not in any other. Likewise, the two scenarios where there was no process tailoring performed as shown in Table 6.9 are:

1. No Split & select, and merging activities at swing state during resource management, and
2. No merging activity at swing state during requirement gathering and tracking.

The reason found behind these interesting results was the inflexible and non cooperative attitude of the client. The project team wanted to make the modifications in the processes but client was never motivated and convinced. The reasons of this problem have already been discussed.

The process tailoring was not according to the requirements of the project therefore, the process could not be modified and refined. The minimum tailoring could not be fruitful and project was closed without completion. Following examples explain the various scenarios of process tailoring performed.

6.4.1 Resource Management

Process States:	Takeoff, Running, Landing, Hang up	
Reasons to Tailor:	Project B:	Skilled and cooperative resources. Improvement in coordination and collaboration. Project requirements. Client's expectations. Meeting deadlines.
	Project C:	Skilled and cooperative resources. Process improvement. To overcome problems. Full utilization of resource on the project.
Processes:	Project B:	Resource hiring Resource transition Resource shuffling Defining roles Effort distribution
	Project C:	Resource shuffling (without the consent of the client) Effort distribution.
Tailoring Performed:	Project B:	See Table 6.8
	Project C:	See Table 6.9
Frequency of Tailoring:	Project B:	1 – 2
	Project C:	1
Repetition:	Project B:	Yes

	Project C:	No
Findings:	Project B:	Light weight set of processes was obtained. Activity of silent resources was originated from the practice. Smooth project progress. Good resources appointed. Good team work and performance. Refined processes. Work load leveling. Client's satisfaction.
	Project C:	No major modifications in the existing resources could be made. Resources overloaded. No improvement in the progress and performance.

6.4.2 Communication, Interaction and Coordination

Process State:	Takeoff, Running, Landing, Hang up	
Reasons to Tailor:	Project B:	Bridging communication gap. Client's management. Coordinated team work. Effective communication channel, interaction and coordination process. Avoiding ambiguities in the requirements. Setting priorities Live demonstrations of builds. Keeping client and team members updated.

	Project C:	Resolving issues of the project. Developing trust relationship with the client. Confidence built up of client on the team. Clarifying the problems and increased understandability.
Processes:	Project B:	Formal and structured meetings. Meeting recordings. Meeting minutes. Document sharing. Automated communication channels. Feedback process.
	Project C:	Formal and informal meetings. Structured and unstructured meetings. Automated communication channels.
Tailoring Performed:	Project B:	See Table 6.8
	Project C:	See Table 6.9
Frequency of Tailoring:	Project B:	1 – 2
	Project C:	1
Repetition:	Project B:	Yes
	Project C:	No
Findings:	Project B:	Well defined communication processes. Well coordinated both client and team. Understandability of client on project's progress and work increased.

		Good working relationship and environment.
	Project C:	Informal meetings created further problems. Coordination could not be developed. Client's understandability and satisfaction could not be achieved.

6.4.3 Requirement Management

Process State	Takeoff, Running, Landing, Hang up	
Reasons to Tailor	Project B:	i. To manage a large number of requirements. ii. To define a proper requirement tracking process. iii. To manage backlog. iv. Keeping track of the progress of the project. v. Avoiding missing requirements. vi. Proper utilization of resources. vii. Timely completion of tasks. viii. Project audit.
	Project C:	To avoid missing requirements and functionalities. Understanding each and every major and minor requirement and functionality. To resolve coding problems. Full utilization of resources. In time completion of the builds.

Processes	Project B:	i. Automated requirement management. ii. Requirement extraction and allocation of tasks to the team members. iii. Tasks statuses and updates. iv. Requirement reopen, fixed close and verified close. v. Project scheduling. vi. Assigning numbers to the tasks and allocating to the team members. vii. Modular approach to develop the system.
	Project C:	i. Master requirement tracking documentation. ii. Client's approval and verification. iii. Tasks ownership through master requirement document.
Tailoring Performed	Project B:	See Table 6.8
	Project C:	See Table 6.9
Frequency of Tailoring	Project B:	1 – 3
	Project C:	1
Repetition	Project B:	Yes.
	Project C:	No

Findings:	Project B:	Well managed requirement management processes.
		Light weight approach of tasks allocation and feedback.
		Backlog tasks started to be completed.
		Enough time to meet the deadlines.
		Resource not overloaded as well as underutilized.
		Transparency of the whole project and team.
	Project C:	Requirements were managed.
		All the tasks allocated to the team members.
		Resources overloaded.
		Overtimes of the resources started.

6.4.4 Analysis of Process Tailoring in Project 'B'

Table 6.10 summarizes the process tailoring performed for each of the tailoring activity, key process and project states as well as total number of activities tailored.

Table 6.10 Overall Process Tailoring in Project 'B'

		Key Processes		
		Resource Management	**Communication (Interaction & Coordination)**	**Requirement Management**
Total Number of Activities	**Total Tailored**	**39**	**20**	**25+25=50**
	Added	13	7	7+8=15
	Deleted	7	3	4+5=9
	Modified	8	3	3+3=6
	Split & Select	3	1	2+2=4
	Merge	3	3	4+4=8
	Shrink	2	0	2+1=3
	Wrap up	3	3	3+2=5

Table 6.10 Overall Process Tailoring in Project 'B' (Continued)

		Key Processes		
		Resource Management	**Communication (Interaction & Coordination)**	**Requirement Management**
Activities Tailored During a State	**Takeoff**	6	4	2+5=7
	Running	12	4	8+7=15
	Landing	7	3	7+4=11
	Hang up	14	9	8+9=17

The project 'B' due to its stability was very accommodative for process tailoring. Process tailoring in this project was not that extensive. Total 39 activities during resource management process, 20 during communication, interaction and coordination process and 50 were performed during requirement management process. The Table 6.10 shows that addition, deletion, modification, merging and wrap up activities were performed the most. While addition, deletion and modification are top most tailored operations performed respectively.

During Hang up state the maximum tailoring activities which are 40 has been performed. The project entered into hang up state twice, therefore the number of tailoring activities in this state are the highest ones. Due to this highest tailoring during hang up state, the project could successfully resolved its problems, got recovered and started running normally. The second highest of tailoring activities was in running state which are 31.

The project manager adopted, deleted and modified a large number of processes during the project and particularly during these states because of the support from the client side. To meet the requirements and expectations of the client lightweight approaches were generated through process tailoring such as explained in 6.4.1 to 6.4.3.

The frequency of tailoring through the project was the same which is 2 times based on the situation, requirements and expectations of the project and the client as described in 6.4.1 to 6.4.2. Also the repetition in some cases was performed as per requirement. For example, an activity deleted/skipped once was added again and modified as required. Therefore, repetition of tailoring was made throughout the project.

6.4.5 Summary of Process Tailoring in Project 'B'

According to the analysis it has been found that the requirement management was the most important phase or process of the project. The highest tailoring performed during this process shows that most suitable set of processes were required during this phase. Resource management processes on the other hand were emerged as the second highly prioritized by the client. It shows that the project required well established and refined process in both phases. Light weight processes generated during these phases through process tailoring played the important role in the overall progress of the project.

The project was stable therefore not that much tailoring was performed during the other states of the project. The process tailoring generated a light weight process set containing a very few sub-process. Due to the tailoring of activities and support of the client project had resolved two critical issues and was successfully completed. The client's trust and confidence on the team members was also increased. Due to which he always gave a go ahead to the project manager in certain decisions and processes.

6.5 Overall Analysis of Project 'B' in Case Study-II

Since its beginning, the client of project 'B' was cool minded, cooperative and supportive. On the other hand, the project had also won the trust of the client with good performance. The team had resolved a large number of minor and major problems, delivered the stable builds, and met the deadlines successfully. Project manager had also used the approach in the areas in which he found the client most concerned and did stabilize the processes.

The client's perspective was met throughout the project. A light weight set of minimum processes was used to manage the whole project. The client was not very demanding. His attitude was quite flexible. Even in case of minor, major and critical problems, he never put pressure on the team. He always welcomed and supported the suggestion of the team in such matters.

The project team had already built up the trust relationship with the client. The client had also confidence in the abilities of the team and approach of the project manager. Therefore, he let the team to resolve the problems and issues as they found it better. He always gave valuable suggestions and recommendations to the team on certain matters. The team according to the client's suggestions, following their own process always successfully came up with some solution and ideas.

The project manager had defined good processes for interaction and coordination with the client and among the team members. The project passed through all the states, got recovered twice from the hang up state and was completed successfully.

6.6 Analysis of Process Tailoring in Project 'C'

Table 6.11 summarizes the operation performed on each of the key process and project states as well as total number of activities tailored.

Table 6.11 Overall Process Tailoring in Project 'C'

		Key Processes		
		Resource Management	**Communication (Interaction & Coordination)**	**Requirement Management**
Total Number of Activities	**Tailored**	**26**	**15**	**15+16=31**
	Added	8	7	5+4=9
	Deleted	7	2	2+2=4
	Modified	2	2	2+2=4
	Split & Select	1	1	2+2=4
	Merge	2	2	2+4=6
	Shrink	2	0	1+1=2
	Wrap up	4	1	1+1=2

Table 6.11 Overall Process Tailoring in Project ‘C’ (Continued)

		Key Processes		
		Resource Management	**Communication (Interaction & Coordination)**	**Requirement Management**
Activities Tailored During a State	**Takeoff**	10	5	3+3=6
	Running	6	3	6+5=11
	Landing	7	1	2+2=4
	Hang up	3	6	4+6=10

Since its beginning, the project ‘C’ was not a stable project with processes. The clients were not that much satisfied with the performance of the project team. The factors responsible for it have already been discussed in earlier sections. Throughout the project, total 26 activities during resource management, 15 during communication, interaction and coordination process and 31 during requirement management process were tailored. As shown in Table 6.11, in addition to the addition, deletion, modification, split & select operations, merging operation had the highest number of tailoring performed. Likewise highest tailoring was performed during takeoff, running and hang up states respectively.

Interestingly the difference of tailoring activities among these three states is very minor which is of only 1 activity. It shows that tailoring throughout the project was consistent. As presented in sections 6.4.1 to 6.4.3, the frequency of tailoring was only 1 and there was no repetition. It means minimum process tailoring was performed during the project.

Both of the requirement management and resource management processes were the highest tailored processes respectively. The client in this project was also very much concerned regarding these two processes.

6.6.1 Summary of Process Tailoring in Project ‘C’

The low process tailoring in this project could not generate a stable and agreed upon set of processes. As we had mentioned earlier, the client’s inflexible attitude and non cooperative behavior did not let the project manager to devise a suitable set of processes. Due to this attitude of client, he could perform less process tailoring and generated a few good processes of requirement management and resource management. These processes were not enough but client was satisfied with them. As a matter of fact, a few more processes were required to modify in the long run, but inflexible attitude of the client did not let the project manager do it.

The process tailoring in this project also proved to be good and generated two very light processes of requirement management but could not be continued because of lack of understanding between client and the project team.

6.7 Overall Analysis of Project 'C' in Case Study-II

Since its beginning, the project 'C' was not that much stable. There were 3 – 4 persons at the client side and everyone was an acting client in addition to the CEO. They were not cooperative and flexible. They were used to hide facts from the project team that was realized at many times.

Each of them was used to handle the project in its own way. The project since its beginning had a very critical problem. The client had not granted access rights to the team on their server. The team had defined a process to download the records from their server. The process was used to take 2 days to be completed. Unluckily, the database was inconsistent as records were redundant. It created a lot of problems for the team members during coding, testing and delivery of builds.

The project manager tried his best to convince the client to modify the processes but client was not at all agreed. The processes continued and problems got accumulated. The instability in the project had increased. The problems were simple but became critical gradually. The project manager was unable to modify, delete or adopt processes due to the lack of understanding of the client.

Project manager had understood the point of views of the client but could not do anything due to his inflexible attitude. Problems could not be resolved. Though project manager had successfully generated the processes through process tailoring as client wanted to do, but further improvements and refinements were required. These could not be possible anyhow due to client's attitude.

The client's perspective had been understood by the project manager during the project, but he could not maintain it. The client's own attitude was proved to be the hindrance in practicing and maintaining it. Except a few, other processes could not be defined and established. The client wanted to take over the project for in house development; therefore, he was not interested in the suggestions of the project manager.

Project faced the problems since the beginning. Client also wanted to take over the project after the completion of some certain level. Ultimately, keeping in view the situation, the CEO of the project team, decided to close the project with the consent of the client. The client's perspective was achieved during the project and processes were defined accordingly, but it could not be maintained and practiced any further due to the client's own attitude. The project was the typical example of client's composite structure, interaction, variations in project states and various aspects of client's perspective.

6.8 Combined Analysis of Project 'B' and Project 'C'

The analysis of the case study of both projects presented many interesting findings. The response to the framework in both projects was variable. The case study covered and supported all aspects of the framework. The framework supported both projects in terms of producing light weight processes of project development and management. It helped project 'B' to establish the processes and refine them. While in case of project 'C' it helped in defining the processes. In either way, the framework provided a good support to both of the projects in terms of good project management and processes. The model supported the improvement in the following major areas of the project:

6.8.1 Understanding the Client

The framework helped in both projects to understand the client, predicting his behavior, and understanding his expectations and point of views. All these objectives through the framework have successfully been achieved.

During the project ‘B’, the processes were defined and modified throughout the project to meet the client’s and project requirements. Client’s points of views were very clear, understandable that helped a lot to set the achievable milestones.

Project ‘C’ had also practiced it and defined processes for the smooth progress of the project. Client’s non cooperative behavior did not allow this approach to grow further. Inspite of critical problems, the project manager had successfully completed about 80% work of the project.

The client’s perspective approach proved to be very helpful in the smooth progress of the project. In project ‘B’, 80% light weight processes were defined, got matured, established and succeeded based on this approach. While in project ‘C’ it was only 50% - 60% improvement.

6.8.2 Requirement Management Process

The requirement management process of project ‘B’ was consisted of requirement gathering and tracking, updating requirement, modifications, and converting them into achievable tasks processes. In project ‘B’ the process was different as it was consisted of only requirement gathering, tracking, documentation and assignment.

In either way, such processes in both projects got matured and helped a lot in the overall progress and performance of the project and team. More than 248 requirements in project ‘B’ and more than 615 requirements in project ‘C’ were successfully managed and completed. The processes to keep track of the requirements and backlog tasks were successfully defined, matured and improved using through client’s perspective, interaction & coordination, and process tailoring approaches.

6.8.3 Communication, Interaction and Coordination

The communication, interaction and coordination processes being the back bone of agile based projects were specifically addressed during the case study. In project 'B' 2 communication channels among 3 roles at client side and project team were successfully developed and managed. In project 'D' 3 effective communication channels were defined and maintained among the 2 roles of client side and project team. The processes defined for communication, interaction and coordination remained in use with modifications as required in both projects throughout. They helped a lot in bridging geographical distances between the client and team members.

6.8.4 Resource Management

The team members are considered as the resources of the project other. Clients in both of the projects were concerned about the proper and full utilization of resources.

In both projects, resource allocation, their effort distribution and tasks allocation processes were defined with the consent of the client. Any change in the resources was always made by the approval of the client. Both in project 'B' and 'C' all the resources were putting 100% effort in the project. Only the effort of the project manager was distributed that was 50% on project 'B'. A few hidden resources were also added to the projects about which client was unaware. Those resources after training, at suitable time were added properly in the team after the client's approval.

On the part of resource management, 90% - 95% improvement was observed in both projects. The light weight processes and activities were defined to manage the resources in both projects. The clients of both projects were completely satisfied with the management of the resources, their utilization and resources. Moreover, resource leveling was also performed in order to manage the work load that improved the performance of the team members and progress of the project.

6.8.5 Project Management

The real applications and advantages of the framework were observed in the processes of project management. The framework generated very light weight processes to manage both of the projects.

Each state of the project was well managed with effective project management. Though there were flaws in the project 'C' but project management was improved too much. Particularly, during hang up, the micro practices were adopted to find quick solutions in short timings. Project 'C' had brought improvements in the project management but could not be sustained. Project 'B' on the other hand was proved to be one of the best managed projects. The client had trust in the project manager.

The framework is actually a short and quick solution for the project managers to manage their projects through effective, short and quick techniques. During analysis it has been observed that the framework actually supports the project management work and provides the project managers a light weight approach to manage their projects.

6.9 Cross Case Analysis

The cross case analysis of all the projects has been made to compare the findings of both case studies. The results of the comparison will be helpful in understanding the findings of the case study.

The cases have been partitioned into three main groups based on the units of case study. Similarities and difference in each case have been examined. This analysis has made easy the understanding of the response of each project to the framework.

Table 6.12 Cross Case Analysis

	Project A	Project B	Project C	Project D
Project Team				
Structure	Hierarchical	Hierarchical	Hierarchical	Hierarchical
Location	Centralized	Centralized	Distributed	Centralized
No. of People	12	8	6	3
Level of Expertise	Medium to High	Medium	Medium to High	Medium
Experience	2-5+ years	1-3+ years	2-3+ years	1+ years
Project's Client				
Location	Offshore, Visiting	Offshore	Offshore	Offshore
Expertise	Medium	Low	High	High
Attitude	Aggressive Positive	Positive	Normal	Uncertain
Behavior	Cooperative	Very Cooperative	Non cooperative	Cooperative
Response	Good	Good	Normal	Good
Project				
Size	Large	Medium	Small - Medium	Small
Type	Web based	Web based	Web based	We based
Domain	Legal financial documents	Multiple chat application	Social networking	Facebook
Process Model	Agile	Agile	Agile	Agile
Processes	Normal	Good	Normal	Good
Type of Development	Centralized	Centralized	Distributed	Distributed
Product Complexity	High	Medium	Low – Medium	Low
Problems Complexity	High	Low – High	Low – High	Low

The numbers of team members in each project were different. The difference of the team members directly had no impact on the performance of the framework. The numbers of the team members were according to the scope, size and complexity of the project. As shown in Table 6.12, project 'A' was the large and most complex project as compared to other projects. It had the highest number of team members i.e. 12. The projects 'B', 'C' and 'D' had 8, 6, and 3 members respectively. Project 'B' was a medium size project, while project 'C' was likely to be medium and project 'D' was a small project.

The product and problem complexity of project 'B' was also greater than the project 'C', and 'D'. The level of expertise and experience of the team members of projects 'A' and 'C' were also greater than 'B'. Inspite of these facts, project 'A' and 'C' having better resource pool than project 'B', the project 'A' and 'B' had resolved the problems, completed all the requirements and were completed successfully while project 'C' could not. The support of the framework to both of the projects 'A' and 'B' was very good as compared to the project 'C'. The complexity of the problems in the project 'C' was also low as compared to the other projects. The resources were as good as in project 'A' but better than 'B'. Despite these facts, the project could get advantage of the framework to some extent which was not enough for the good progress of the project and performance of the team members.

These facts show that the characteristics of the team members alone are not enough for the progress of the project. The teams in all the projects were good with slight differences, but other factors were also involved.

Moreover, the development work of project 'C' was distributed. As discussed in previous sections, there was another team at the client side for in house development and management of the some other components of the application. Both teams could not understand each other and the work was greatly affected. Unlike project 'A' and

‘B’, the client’s perspective was not clear in the project that provided partial support to the framework. It was realized that for the distributed development environments, the client’s perspective is very important to understand and bridges the gap between the teams.

The client’s perspective was fully met in both projects ‘A’ and ‘B’. The clients in all the projects were offshore. Only the client of project ‘A’ was used to visit the team during the days of final releases for completion of work well in time. For this purpose, to keep the whole team highly motivated and creating good working environment, he was used to give them incentives and rewards such as financial, and recreational in the form of dinners, tours etc. Though he showed aggressive but positive attitude at some points but in general was a cooperative person. Due to these factors the framework fully supported the project and played a role in its success. While the client of project ‘B’ was good and more cooperative with positive attitude than project ‘A’ and ‘C’. Due to the cooperative and positive attitudes of clients in both project ‘A’ and ‘B’, the full support of the framework could be possible. This was lacking in project ‘C’.

The limited study of project ‘D’ was made only for the structure and interaction overview of the client. This was a small project with only three team members including project manager. The experience of the team members was 1 – 2 years only. The project was being developed at two geographically distributed locations. The client was doing the coding itself and developers at the offshore team. This was a simple project with few problems. The client’s attitude was quite uncertain but cooperative. His response to the team on certain problems and issues was quite good and quick. Unlike other projects ‘A’, ‘B’ and ‘C’, the processes in this project were quite different. The communication with the client was the most frequent as compared to others which many times in a day. The client’s uncertain mood or attitude was sometimes the problem for the team; otherwise it had no big problems.

The analysis showed that the framework supported all the projects. Though support to the project ‘C’ was not same as in other projects, but it was found that the existing support helped the project manager more than enough as per condition of the project.

It also concluded that client's perspective plays an important role in the overall software project development and management. The framework for the first time has provided a deep insight into and guidelines to this aspect.

6.10 Common Findings of Case Study I & II

The observations, discussions and analysis of both case studies found that in small and medium sized software development companies following agile based methodologies:

i. The role of client is the most important success factor in agile based projects in small and medium sized software development companies.
ii. Client is mainly concerned with the project resources, communication and interaction, and requirement management.
iii. Client's perspective is very important and necessary to understand and maintain through the project lifecycle to achieve client's satisfaction and for project success.
iv. The execution flow, progress and behavior of the projects are greatly affected and determined by client factors, and problems faced by these projects.
v. The projects respond variably to these factors and pass through various states during their lives.
vi. Process tailoring of three key process areas such as resource management, communication, interaction and coordination, and requirement management generates lightweight processes for these companies.
vii. Tailoring these three key process areas when project is in a particular state is an effective and realistic tailoring approach.
viii. The agile based processes and activities tailored based on client's perspective factor are more effective and address the software development processes issues in these companies.

6.11 Findings of Questionnaire Method

The questionnaire has been prepared based on the findings of the case study as presented in section 6.10 as well as based on the components of the process tailoring framework. The questionnaire has been prepared in accordance with the case study analysis and has been made consistent with the case study findings. Similarly as the case study has been performed according to the main components of the framework, the questionnaire has been prepared based on the same pattern of case study.

The questions have been divided into three main sections according to the main components of the framework such as client's perspective, project states and process tailoring. The survey has been conducted in small and medium sized software development companies in order to provide support to the framework and case study findings.

6.11.1 Data Collection

Data has been collected using a structured questionnaire consisting of 39 questions out of which 5 questions belonged to demographic information. The questions of client's perspective are related to the importance, satisfaction and preferences of client as well as key processes on which client is mostly concerned in agile based development. The questions related to the project states have been described in terms of behavior and response of the project to various factors. The questions of process tailoring are comprised of process tailoring activities as well as tailoring scales. The five point Likert scale [27] ranging from 1 = Strongly Disagree, 2 = Disagree, to 5 = Strongly Agree has been used in feedback and response analysis.

The questionnaire was sent to various small and medium sized software development companies as well as posted on different online forums of project managers, software developers, software development projects, and project management.

6.11.2 Response Analysis

Out of 143 total responses received, 103 complete responses have been selected for further analysis. [228], [229], [230], [231] have considered 100 sample size enough for the analysis whereas minimum sample sizes such as 42, 59 as well as less than 100 have also been reported by [232], [27] and [233]. The details of the demographic data have been presented in Appendix H. All the responses represent the small and medium software development companies operating in different countries such as Malaysia, UK, UAE, Pakistan, China, Austria, and India. A large number of companies in these countries consist of small and medium sized software development companies. Due to their growing environments, these companies in these countries follow similar kind of practices.

The maximum numbers of respondents were senior persons belonging to software engineers group (38.8% & 26.2%) and project management group (13.6% & 11.7%) respectively. The response shows that project management professionals being busy in their projects showed less interest in giving feedback on the questionnaires. This has been realized as the general attitude of industry professionals towards the academic research works. The experience of the respondents has been categorized into four groups according to the number of years.

The majority of the respondents i.e. 52.4% have more than 5 years and less than 10 years of experience which is more than half of all the respondents. Overall majority of the respondents have 1 – 10 years of experience. The higher numbers of responses from more experienced persons show that the findings of the survey are more reliable and accurate. All the respondents belonged to various small and medium sized software development companies following agile based methodologies.

Demographic data as presented in Appendix H shows that software engineers and project management professionals particularly the senior people having upto 10 years of experience working in small and medium sized software development companies following agile based methodologies have actively participated in the survey and have provided valuable information on the process tailoring practices in their respective companies.

6.11.3 Empirical Findings

The empirical results of the analysis have been presented in Table 6.13 which shows the mean values of each main dimension of the questionnaire whereas dimensions represent the components of the framework. The mean scores above the midpoint i.e. 3 on 5 point Likert scale [27] shows that framework of software process tailoring is fully applicable in small and medium sized software development companies.

The cronbach alpha coefficient of the overall data which has the value of 0.869 shows high reliability and consistency of results. Nonetheless, there actually has not been defined lower limit of this value but the coefficient value more closer to 1 has been considered good and acceptable for more internal consistency [234]. As cited by [234], [235] has considered 0.7 cronbach alpha coefficient as an acceptable value for reliability and consistency of data. However, all the three components have the cronbach alphas greater than 0.7 as shown in Table 6.13.

Table 6.13 Quantitative Results of Main Components

Component	Item	Mean	Cronbach Alpha
Client	Client's Perspective	3.63	0.7228
Project	Project States	3.66	0.7025
Process Tailoring	Tailoring schema	3.78	0.8179

The 3.63 mean value of client's perspective shows that client's perspective has been considered as an important factor affecting the software development processes in small and medium sized software development companies following agile based methodologies. Related to client's perspective the mean values of client's satisfaction and client's preferred processes (key process areas) in agile based environments of such companies have been calculated as 3.88 and 3.80 respectively as shown in Table 6.14.

Table 6.14 Sub-components Results

Item	Focus	Mean
Client's Perspective	Client's Response/Satisfaction	3.88
	Key Process Areas (Resource Management, Communication, Requirement Management	3.80
Tailoring Schema Scale	Moderate tailoring	3.42
	Maximum tailoring	3.25
	Minimum tailoring	3.95
	Specific tailoring (for two types of operations)	3.68 & 3.72

It shows that client's satisfaction has been considered very important in such companies and has been regarded as an important measure of client's perspective. Whereas, resource management, communication and requirement management have been found as the key process areas on which client is mostly concerned in such small and medium sized software development companies.

Similarly, above average mean values of 3.66 and 3.78 of project states and tailoring schema respectively as shown in Table 6.13 have also supported these two components as an important part of the framework. As shown in Table 6.14, the high mean values of 3.42, 3.25 and 3.95 supports that moderate, maximum and minimum process tailoring activities should be performed during the project lifecycle with respect to the state of the project whereas, the high mean values of 3.68 and 3.72 also shows that only specific tailoring activities should be performed when project is about to complete and close. These maximum, minimum, moderate and specific tailoring operations have been explained further in following section 6.12.

Therefore, all the above average means support the validity of the components of the framework as identified and validated in qualitative part such that in small and medium sized software development companies following agile based methodologies key software development processes should be tailored based on the client's perspective with respect to the particular state of the project.

The findings of both qualitative and quantitative studies validate the process tailoring framework and schema. Further results of quantitative survey in relation to the qualitative findings have been discussed in chapter 7 of results and discussions.

6.12 Process Tailoring Scales

Based on the findings of both qualitative and quantitative analysis of the tailoring framework, the process tailoring scales have been derived as shown in Table 6.15. The scales as shown in Table 6.15 describes that how much process tailoring should be performed during each state of the project.

Table 6.15 Process Tailoring Scales

Scale /State	**Take off**		**Running**		**Landing**		**Hang up**		
	Pre-take off	Take off	Running	Post running	Landing	Post landing	Crawling	Swing	Pre-running
Minimum	·				·				
Moderate		·		·					·
Maximum			·				·	·	
Specific						·			

These scales have been derived on the basis of number of tailoring activities or operations performed during each state of the project during case study analysis as presented in chapter 5 and 6, and supporting quantitative results as shown in Table 6.14. On the basis of these results, process tailoring scales have been defined as follows:

a) Minimum

Refers to the minimum or least number of tailoring operations performed during a project state or sub-state.

b) Moderate

Refers to neither least number of tailoring operations nor highest number of operations performed during a project state or sub-state.

c) Maximum

Refers to the highest number of tailoring operation performed during a project state or sub-state.

d) Specific

Refers to the particular tailoring operations performed only during specific project state or sub-state.

Takeoff is an initial state of the project when project is at early stage with less number of processes which are maturing gradually then minimum or moderate levels of process tailoring should have been considered with respect to the sub-states.

Running state is the proper development state of the project. A few early iterations have been completed and released and processes have has been matured enough to be tailored. Therefore, maximum process tailoring should have been performed. However, during post running state when project and processes are stable, the process tailoring should be minimum in order to avoid any problem caused by process manipulation/tailoring that may create instability in processes or projects.

Landing is the last or completion state of the project when no more new functionalities are implemented and only remaining works are completed. Therefore, minimum process tailoring should be performed because most of the processes have has been closed and later during post landing sub-state only specific tailoring operations should have been performed such as shrink and wrap up according to their definitions and suitability for this state.

Hang up is the problem state of the project which decides the future of the project. Maximum measures are taken during this state therefore, maximum process tailoring should be done to resolve the problems when project is in crawling or swing sub-states, and during recovery (pre-running) moderate process tailoring is considered because of not disturbing the tailored processes that have resolved the problems and recovered the project from hang up state.

Based on the tailoring activities performed and tailoring scales as shown in Table 6.15, a process tailoring schema has been defined which recommends the tailoring operations to be performed for each key process area with respect to each state of the project as shown in Table 6.16.

6.13 Process Tailoring Schema

The process tailoring schema has been derived based on the process tailoring activities performed during case studies findings shown in Tables 5.7, 5.8 and 6.9 to 6.11 and quantitative results presented in chapter 6. The process tailoring schema has been presented in Appendix J.

The procerss tailoring schema maps the project states with the key process area and recommmends that which tailoring operatons should be performed for each key process when project is in a particular state. The takeoff sub-state has been tailored alongwith running state. As at pre-takeoff sub-state, the project is new and just beginning therefore, processes are started to add or adopt at this level. Therefore, among all other tailoring operations only add tailoring operation is peroformed. When processes have has been added, later on during takeoff sub-state further tailoring operations can be peroformed. It is possible that the tailoring operation performed at one state or sub-state may or may not be performed again at the next state. The same scenario happened at takeoff sub-state and running state. In such sccenarios overall tailoring activities have been peroformed during each state instead of separate tailoring during each sub-state.

For example, adding an activity during runnning sub-state, deleting that activity during post-running sub-state and again adding the same activitiy during post-running sub-state had been found during the case study and most likely was an overhead of the process tailoring in terms of resource utilization and time consumption. Though this scenario is not neglectable and may exist but to keep the tailoring proces more simple and lightweight overall process tailoring has been performed for each state as shown in process tailoring schema in Appendix J.

6.13.1 Tailoring During Takeoff State

The tailoring schema sugggests that when project is in takeoff state, only add tailoring operations should be performed for each of the three key processes. The project is just in its initial stage therefore, process and activities have to be added only, no processes exist to be deleted or modified. Later during takeoff sub-state processes have has been added and can be tailored. Therefore, tailoring for takeoff sub-state has been performed in association with runnning state.

6.13.2 Tailoring During Running State

The schema shows overall tailoring operations performed during takeoff sub-state and running state. When project is in initial running sub-state then add, delete/skip, and modify tailoring operations should be performed for each of the key process areas. When processes and project gets matured during post running sub-state, it is not required to manipulate or tailor the processes unnecessarily. Only a few tailoring operations such as split & select and merge should be performed during running sub-state only. Tailoring processes or activities during post running sub-state when projects are fully matured and stable may create problems and cause instability in their progress. Therefore, tailoring during post-running sub-state is not recommended and should be need based only. Process tailoring schema as presented in Appendix J shows this scheme and suggests tailoring operations that should be performed to tailor the three key processes with respect to the running state of the project.

6.13.3 Tailoring During Landing State

During landing state when projects are going to complete and close, only specific tailoring operation should be performed. At this stage, existing processes have been closed, lengthy and complex, and unrequired processes are winded up and deleted. Number of activities of complex and large processes are tend to reomve. Only specific tailoring operations are performed with the view that project is being closed and no excessive or normal process tailoring is required at this stage. Process tailoring schema in Appendix J shows the specific tailoring operations performed during landing state.

6.13.4 Tailoring During Hang up State

As described earlier, Hang up is the state when project faces problems and sometimes its progress is in danger and in most sewere cases projects may be closed without completion. Therefore, to overcome the problems, resolve the problems and recovering the project from this state, maximum process tailoring should be performed so that in either way project should be recovered. During pre-running sub-state when project is again resuming its progress after getting recovered from the problems then except basic tailoring operations such as add, delete or modify not other operation should be performed. Basic tailoring should be performed as required and project should let be more stable and smooth. Process tailoring schema table as shown in Appendix J shows the tailoring operation that should be performed during hang up state for each key process area.

The process tailoring schema provides complete guidelines on how process tailoring should be performed, which operations should be performed and when those operations should be performed. The tailring schema generates the lightweight set of processes and activities for small and medium sized companies according to the agile based methodologies.

6.14 Process Tailoring Framework Guidelines

The process tailoring framework as shown in Fig. 4.19 in chapter 4 has been validated through case study of real projects and questionnaire method. The questionnaire method has been conducted after the completion of the case study because of the fact that questions in the questionnaire are based on the findings of the case study and working on the framework components. This approach has been adopted to find more authentic results and support to the framework from both methods. The detailed results, their comparisons and explanations have been presented in chapter 5 and 6.

The specific stepwise process tailoring guidelines of the validated process tailoring framework as shown in Fig. 4.19 have been presented in Fig. 6.4.

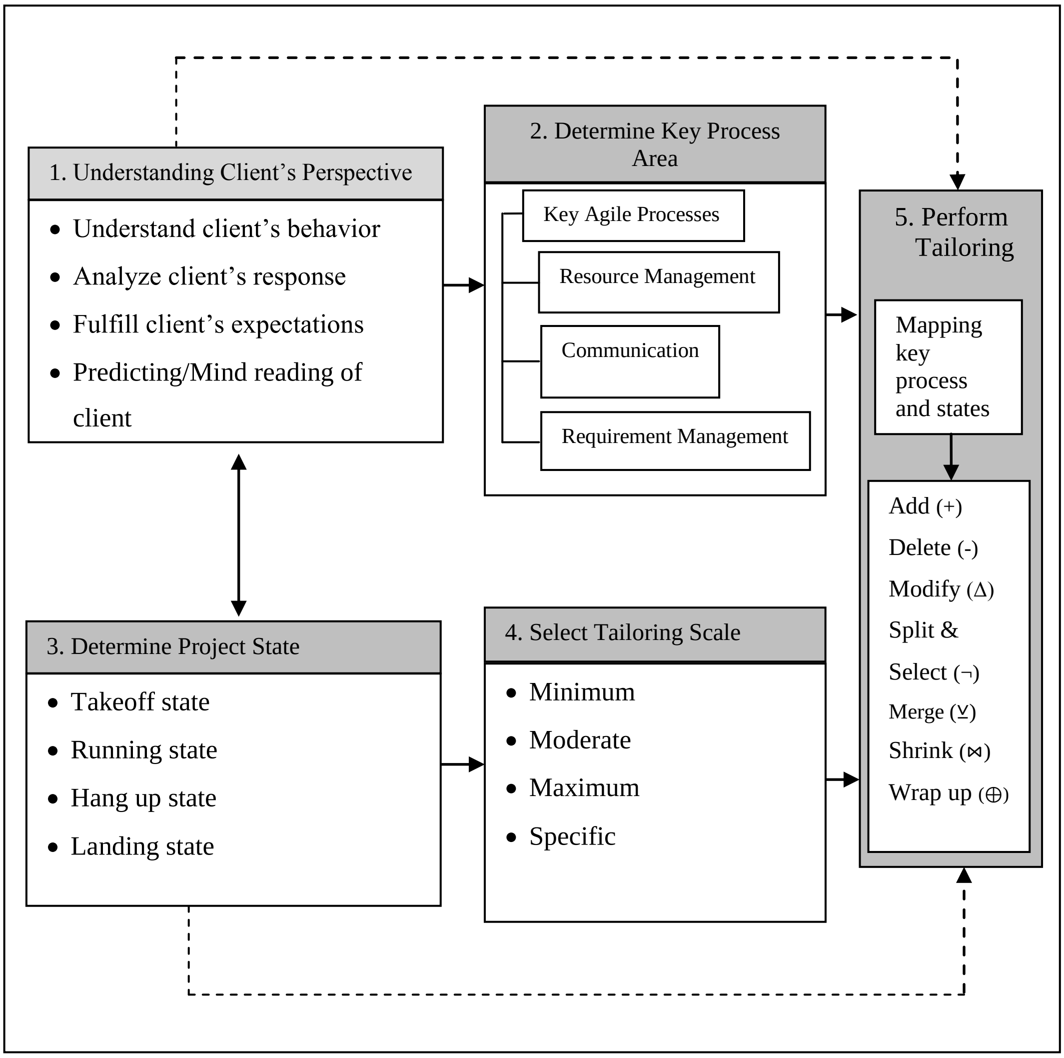

Fig. 6.4 Stepwise Framework Guidelines for Process Tailoring

The guidelines as shown in Fig. 6.4 describes that to tailor a software development process, following steps would have been followed:

1. Understand the client's perspective, the expectations and requirements of the client.
2. Determine which key process is required to be tailored based on the requirements of the project or client.
3. Find the current state of the project.
4. With respect to the project state, select the tailoring scale.
5. Apply the tailoring operations and tailor the process with respect to the state of the project according to the requirements of the project and client.

The details on understanding client's perspective, determining key processes, finding the project state and performing tailoring have been presented in chapter 5 and 6. The process tailoring schema as shown in Appendix L also provides hands on guidelines on performing process tailoring. Fig. 6.4 only presents simplified steps to perform process tailoring. As shown in Fig. 4.19, the validated framework is a fundamental framework to tailor a software development process that can be further inherited and extended. Further models and components can be derived from it.

6.15 Summary

The findings of case study-II have been found similar to case study-I as presented in section 5.6. The basic difference between both case studies is in the number of projects. The case study-I has been performed on a single project, whereas two projects have been analyzed in case study-II. Though a part of project 'C' has been discussed in case study-I due to relevance with project 'A' but its detailed analysis has been presented in case study-II.

In project 'B' the client's perspective was fully met due to which process tailoring performed in this project brought positive improvements in the project and project was successfully completed. On the other hand, in project 'C' though client's perspective was understood but could not be met due to his unsupportive and non cooperative behavior. In such circumstances, the process tailoring could not bring positive improvements in the project and processes required by the software development team could not be derived and implemented in project ‘C’. Due to non cooperative behavior of client, the project had been closed in a mutual decision. Unlike project ‘B’, project ‘C’ spent much of its life in hang up state.

The findings of case study-II also support the findings of case study-I such that process tailoring should be performed based on the client’s perspective in small and medium sized companies. The findings also support that only three key processes on which agile methodologies are based should be tailored according to the respective state of the project. Similar findings of case study-I have been presented in section 5.6.

The quantitative results as presented in section 6.11 completely support the findings of case study-I & II. The above average high mean values of all framework components validate the process tailoring framework and results of both case studies. Both of the process tailoring scales and process tailoring schema as shown in Appendix J which have been defined based on the case study findings and quantitative results truly describes that when, how and how much process tailoring should be performed. The tailoring schema provides a complete guideline on which process tailoring operations should be performed to tailor each of the key processes when project is in a particular state.

The process tailoring schema provides guidelines to the project managers of small and medium sized companies on how effectively they can tailor the processes according to their requirements. The quantitative results in relevance to case study findings validates the process tailoring framework and support the applicability of framework to the agile based projects in small and medium sized software development companies.

CHAPTER 7

RESULTS AND DISCUSSION

Overview

This chapter summarizes the results of the present study, compares the results of both qualitative and quantitative methods and presents a detailed discussion. It also finds the answers of the research questions presented in chapter 1. Furthermore, it conforms the results to the principles of agile manifesto and lastly, presents a comparison of framework with an existing work on software process tailoring.

7.1. Results in Software Engineering

Research produces new knowledge which is interpreted through particular results [99]. The results in software engineering research are expressed in various ways. The result may be a particular technique of software development, general findings about a model, a solution to some problem or outcome of the analysis of some phenomena expressed through clear chain of evidences based on experience or systematic analysis [99]. The chain of evidence enables a reader to derive the results and conclusions following the analysis of data which is carried out in parallel with the data collection [156]. Similar to the nature of the problems in software engineering research, qualitative results often are considered very difficult to summarize and express being considered as “softer” or “fuzzier” as compared to the quantitative findings [160].

The qualitative results are more authentic portraying the real phenomena, scenarios and practices of software development. Qualitative data increases the validity and confidence in results derived by applying multiple analysis techniques on collected large amount of diversified data containing rich information [160].

This chapter summarizes the results of case study comparing them with the quantitative results and presents a detailed discussion. The results have been divided into three sections according to the process tailoring framework and have been elaborated in terms of answers of the research questions as described in chapter 1.

7.2. Client's Perspective as Critical Factor in Small and Medium Sized Companies

The case study results as shown in Table 7.1 and quantitative results presented in the following sections addresses the RQ2 and RQ3 research questions as described in section 1.6 in chapter 1.

The results related to the critical factors such as client and client's perspective in software development projects based on the case study have been summarized in Table 7.1.

Table 7.1 Client's Perspective Summary of Results

Results of the Study	Project			
	A	B	C	D
Composite Structure (No. of Roles)	2	3	4 + 1 team	2
Interaction Overview Model (Avg. No. of Interactions per day within each pair of roles)	1.6	1	2.2	2.5
Client's Perspective	Understood and Achieved, Satisfied client	Understood and Achieved, Satisfied client	Not fully achieved. Client not fully satisfied	N/A
Project Success **Team Performance** **Project Completion**	Successful Satisfactory Completed	Successful Satisfactory Completed	Not really successful Not satisfactory Closed before completion.	

N/A – Not Applicable

The results show that client's perspective throughout all the three projects was successfully achieved. The existence of various roles of client in all the projects show client as a composite entity which may contain more than one role in a project. The average number of interactions among different roles of the client and offshore team ranges from 1 – 3 per day. The clients in projects 'A' and 'B' were found satisfied with the performance of the team and progress of the project. However, the client of project 'C' was not that much satisfied as discussed earlier because his own attitude could not let him be satisfied with the performance of the project team throughout the project. However, the client's perspective in project 'C' was also understood but not as much as in project 'A' and 'B'. Infact the non-cooperative attitude of client was itself a hindrance in accomplishing client's perspective in project 'C'.

The high mean value of 3.88 as shown in Table 6.14 in chapter 6 also verifies that client's satisfactory response is very important in projects following agile based methodologies in small and medium sized software development companies. Similarly in another study on agile based development, the mean value related to the satisfaction level of stakeholders having client as one of them with respect to the project outcome have been reported as 3.65 [27]. The results of present study have been found better than the former because of the participants of the survey who belong to the senior people in development team and management group who directly interact with the client, and secondly the former study is based on South African companies while present study is not confined to any specific country and specifically focuses on small and medium sized companies.

The findings show that client's perspective and satisfaction have been considered very important in projects running in small and medium sized software development companies following agile based methodologies whereas, the satisfaction of client can be achieved through understanding and maintaining client's perspective, and achieving his/her requirements and satisfaction itself [111] throughout the project. Client satisfaction is very important for the software development companies because it greatly affect their future business and clientage [31]. In general, agile based development positively affects the progress of the project and satisfaction level [27].

7.2.1. Quantitative Findings on Client's Perspective as Success Factor

Table 7.2 shows the descriptive results [173] of questionnaire method on response of client related to the project success, team performance, and project completion. The quantitative analysis has also found satisfaction of client very important in terms of achieving and maintaining client's perspective for the success of the projects. The mean scores above the average or midpoint value of 3 at 5 point scale [27] shows full support to the qualitative findings as shown in Table 7.2 whereas, the standard deviation shows the spread of data or distance from mean value. The number of diversified respondents may increase the value of standard deviation such as in the present analysis the respondents belonged to four major categories of industry practitioners as shown in Appendix G. The difference of opinion in such a case may increase the data spread across the mean value but it cannot be negative. Appendix I shows the list of questions that have been grouped here according to their relevance to the qualitative analysis of various cases.

Table 7.2 Client's Response Assessment

Client satisfaction	Mean	Std. Dev.
Project Success (Q8)	4.40	0.9429
Team Performance (Q9)	3.99	0.9235
Project completion (Q10)	3.24	1.0238

As shown in Table 7.2, the high mean value 4.40 shows that satisfaction of client leads to the success of the project. Whereas, the above average high mean values of 3.99 and 3.24 supports that performance of software development team is an important factor in achieving client's satisfaction which is ultimately necessary for the completion of the project.

The results as discussed earlier and also as summarized in Table 7.2 show that client's satisfaction in small and medium sized software development companies has been considered very important factor for the success of the project and team performance such that most likely projects tend to close due to the dissatisfaction of client. Client's satisfaction is very important in software development projects running in small and medium sized software development companies following agile based methodologies, and is an important element of client's perspective. It shows that client has been considered as an important factor in software development projects running in small and medium sized software development companies.

Table 7.3 Client Factor Assessment

Client	Mean	Std. Dev.
Influence & Importance (Q6)	4.20	1.0324
Critical & Decisive (Q7)	4.16	0.8255
Priorities & Expectations (Q16)	3.99	0.8459
Processes & Work Completion (Q13)	3.81	0.8524

Table 7.3 shows the role of client factor in overall software development process. The high mean values of 4.20 and 4.16 show that client is an influential, critical and decisive factor in projects running in small and medium sized software development companies. The expectations and priorities of the client are always given preference as supported by 3.99 mean value whereas, the value 3.81 shows that client is more interested in completion of its work instead of processes. These results show that majority of the industry practitioners agree that client in small and medium sized software development companies following agile based methodologies is an important factor and its requirements are given preference over processes.

Despite this fact, the processes have been found necessary for the software development projects as well as for the client's satisfaction. Slightly above average mean values of 2.68 and 2.89 as shown in Table 7.4 describes that more than 50% industry practitioners also do not consider processes important for satisfaction of client who is mostly interested in working code completion.

Table 7.4 Client-Process Assessment

Client – Process	Mean	Std. Dev.
Interest in Process/Working Code (Q14)	2.68	1.0592
Satisfaction & Happiness (Q15)	2.89	1.0928

In addition, client's perspective has also been considered necessary for the success of the projects as well as for the satisfaction of the client with high mean value of 3.85 as shown in Table 7.5. The mean value of 2.02 shows that less number of the respondents have disagreed that client's perspective is not necessary for the success of the project.

Table 7.5 Client's Perspective Assessment

Client's Perspective	Mean	Std. Dev.
Project Success (Q12)	2.02	1.1374
Client's Satisfaction (Q11)	3.85	1.0233
Project Risks (Q17)	3.97	0.9015

Furthermore, the very high mean value of 3.97 shows that majority of the industry professionals agree that a number of issues and risks in software development projects are associated with the lack of understanding of client's perspective.

These results show that overall satisfaction of client from the growth and development perspective of the software development project is very important for the success of the project. The satisfaction of client is achieved when client's perspective is met throughout the project development.

7.3. Key Processes for Software Development Process Tailoring

The communication, resources management and requirement management processes have been found as the key processes in small and medium sized software development companies following agile based methodologies. The client in these companies remains mostly concerned with these processes throughout the project lifecycle. As shown in Table 7.6 the high mean values of 4.02 show that majority of the industry professionals consider communication, interaction and coordination, resource allocations and requirement management processes as the key process areas in which process tailoring needs to be performed because of importance of these areas from client's perspective. The value of 3.58 as shown in Table 7.6 also supports this fact.

Table 7.6 Key Processes in Process Tailoring

Client – Key Processes	Mean	Std. Dev.
Requirement Management, Communication & Resource Management (Q18)	4.02	0.8281
Concern with Key process areas in agile (Q19)	3.58	0.7608

The result of both qualitative and quantitative analysis show that client is a critical factor in software development projects running in small and medium sized software development companies following agile based methodologies. The project success depends on the satisfaction of client in these companies which is ultimately dependent on the understanding and maintaining client's perspective throughout the project lifecycle in these companies.

The results show that highly experienced industry practitioners accept and agree with the importance of client's perspective in projects following agile processes in these companies. Therefore, client's perspective and the key processes areas are the two most important elements of software development processes in these companies. Therefore, tailoring of these key processes based on client's perspective is more effective process tailoring procedure in small and medium size software development companies.

The results as discussed in section 7.2 finds the answers of research questions RQ2 and RQ3 as described in chapter 1 whereas, both sections 7.2 and 7.3 partially addresses the RQ1 by identifying the factors that provide basis for process tailoring in small and medium size software development companies. The research question RQ1 is completely answered through the results and discussion presented in section 7.3 and 7.4.

7.4. Behavior of Software Projects in Small and Medium Sized Companies

In order to tailor a software development process in small and medium sized software development companies, it is important to understand the behavior and growth of software development projects in these companies. This section finds the answer of the research question RQ4 and RQ1. The results of qualitative study on project behavior and behavioral states in context of problem classes have been summarized in Table 7.7.

Table 7.7 Project Behavioral States Summary of Results

Results of the Study		**Project**											
		A				**B**				**C**			
		No. of Problems	**Impact**			No. of Problems	**Impact**			No. of Problems	**Impact**		
			Low	Medium	High		Low	Medium	High		Low	Medium	High
Problem Class (No. of Existence during all states)	Performance Minor	6	4	1	1	4	3	1	-	5	1	2	2
	Management Critical	6	1	3	2	3	-	1	2	2	-	1	1
	Progress Limiter	2	-	-	2	1	-	-	1	2	-	-	2
Total Number of Problems in Each Project		14	5	4	5	8	3	2	3	9	1	3	5
Total Duration of a State in Months	Takeoff	3				1.5				0.5			
	Running	10				9				6			
	Landing	3				1 + 2 weeks				9 days			
	Hang up	9				6 + 1 week				5			

The results as presented in Table 7.7 show that number of performance minor class problems are 6, 4, and 5 respectively in project A, B, and C which is higher than other problem classes in the same projects.

On contrary, the management critical class problems were mostly in the range of medium to high impact and progress limiter were mostly of high impact. Unlike project ‘C’, performance minor problems in both project ‘A’ and ‘B’ were mostly in the low impact range. The high impact of performance minor problems converted these into progress-limiter that led to instability in project ‘C’. The impact of management critical and progress limiter problems was higher only in project ‘A’ and ‘C’. Fig. 7.1 shows the existence of number of problems of each problem class in project A, B, and C.

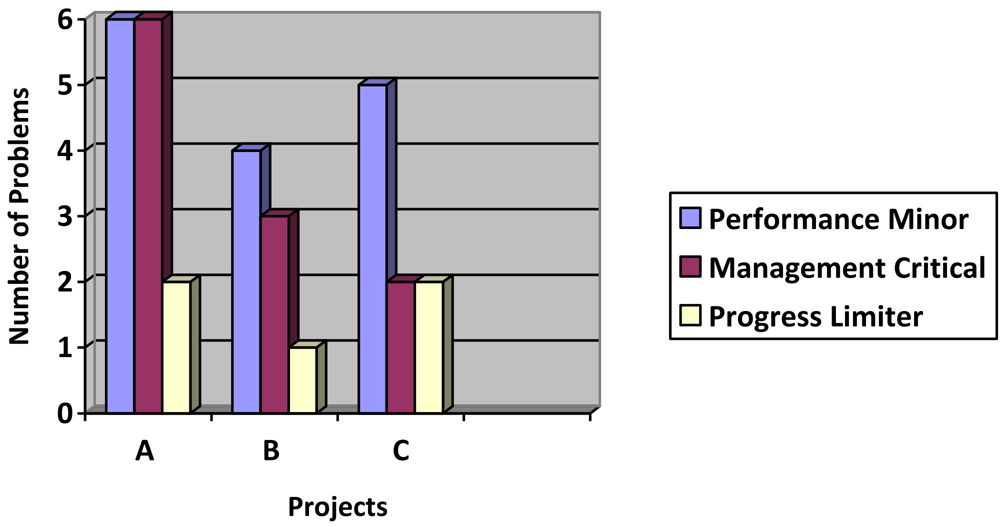

Fig. 7.1 Problem Class Existence in Each Project

As shown in the Fig. 7.1, the project ‘A’ had the highest number of problems. The progress-limiter class problems in both project ‘A’ and ‘C’ were same, but unlike project ‘C’, client’s perspective in project ‘A’ was fully met, therefore it successfully resolved all the problems and issues. Fig. 7.1 also highlights two interesting facts about client’s perspective such as:

i. Client’s perspective helps in resolving the problems and issues in the project.

ii. Client’s perspective helps avoiding or saving the project from problems and issues.

Both scenarios were present in project 'B'. The client was satisfied with the performance and progress of the team so despite the problems, the project's progress was smooth and good. Progress limiter class problems in all the projects were very few as compared to other problems as shown in Fig. 7.1. The client's perspective proved to be very helpful and supportive in resolving the problems and bringing stability in the projects.

7.4.1. Flow and Behavior of Software Project States

Table 7.7 summarizes the duration of Takeoff, Running, Hang up and Landing states in each of the project A, B, and C. Fig. 7.2 shows the comparison of the durations of each state in all the projects.

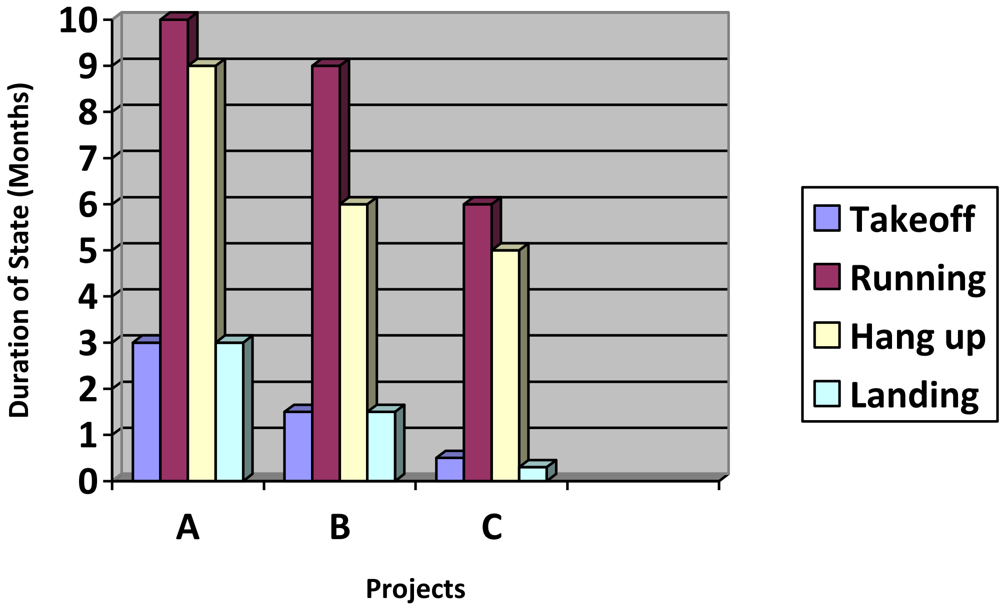

Fig.7.2 Duration of Project States in Each Project

All the projects spent more than 5 months in running state as shown in Fig. 7.2. The hang up state in both of the projects 'A' and 'B' was longer than that of project 'C', but client's perspective helped both projects to recover from the hang up state as opposed to project 'C' where client was not that much supportive. The minimum time of a project in a particular state is 9 days that project 'C' spent during landing state as well as takeoff state that was comprised of only half of the month as shown in Fig. 7.2. Fig. 7.2 also shows that hang up states in both projects 'A' and 'C' started shortly after running state had started. Each project entered into a hang up state at least once in its life. All these states do exist in all types of projects separately, in parallel or overlapping with other states.

7.4.2. Quantitative Findings on Problem Existence and Project States

As discussed in previous sections on qualitative findings that all the projects face problems throughout their life. As shown in Table 7.8, the mean value of 3.81 shows that all the projects face problems during their lifecycle. Majority of the industry professionals believes that all the projects pass through these problems throughout their lives.

The states of the project are determined by response of the project to these problems and key processes. As shown in Table 7.8, the mean value of 3.80 describes the strong relationship between a state of the project, and problem classes and key processes.

Table 7.8 Problem Existence Assessment

Problem Existence	Mean	Std. Dev.
Project – Ideally No Problem Exist (Q20)	3.81	1.1551
Project States – Response to Problems, Key Processes (Q26)	3.80	0.8558
Project States – Different Behavior & Response of projects (Q27)	3.71	0.7749

The duration of a state varies from project to project and can be measured with great difficulty based on an extensive study of a large number of diversified projects. The duration of a particular state depends on the scenarios, problems, issues, behavior and response of project to the key processes as well as client's perspective as evident by high mean value of 3.71 as shown in Table 7.8.

As presented in Table 7.9, the high mean value 4.00 shows that due to the severe problems such as progress limiter class problems, the progress of the project gets affected and can be slowed down (crawling sub-state). A project may continue (performance minor/management critical problems) as in running state, and either can be terminated by the client due to severe problems (progress limiter) as in hang up state or may recover from it (pre-running / running states) and complete its life (landing state). The mean values 3.73 and 3.17 as shown in Table 7.9 support this behavior of the project states and their existence.

Table 7.9 Project States and Response Assessment

Project States	Mean	Std. Dev.
Progress Hindrance (Q21)	4.00	0.7921
Project Termination (Q22)	3.17	1.0703
Project Success (Q23)	3.73	0.8188
Early phase (Q24)	3.38	0.8869

The mean value of 3.38 as shown in Table 7.9 finds that majority of the problems are faced by the projects during running state after completing few early milestones (takeoff state) of the project.

The high mean values show that industry practitioners also support that a software development project passes through takeoff, running, landing and hang up states till its completion such that it may complete its life successfully or may be closed before completion. Furthermore, the quantitative analysis fully supports the findings of the qualitative part.

The results of both qualitative and quantitative part show that project states exist in all software development projects whereas, hang up state may exist more than once in a project. The results further verify that all the states of software projects as shown in Fig. 4.13 were present in all the projects and thus validate the project states meta-model.

Furthermore, it shows that the project where client's perspective was met could have resolved their minor and major problems successfully as compared to those where client's perspective could not be achieved upto the expectations irrespective of the reason such as project ‘C’. Both project ‘A’ and ‘B’ who resolved their issues and problems by successfully understanding of the client’s perspective completed their lives as compared to project ‘C’ in which project was closed before completion because of the problems and not achieving the client’s perspective. This shows that project’s behavior and growth is associated with the problems and issues present and understanding client’s perspective.

The results as discussed in this section address the research question RQ4 (see section 1.6) by analyzing and discussing the behavior of software development projects with respect to the problem classes and client’s perspective factors. Both client’s perspective and problem classes have been discussed in section 7.2 and section 7.3 respectively. It shows that in response to these factors, the software project passes through various states and sub-states as shown in Fig. 4.13. The response and behavior of the project to these factors determines its future. A part of research question RQ1 (see section 1.6) has also been addressed in this section by discussing and identifying the behavioral states of the software projects. It shows that key processes of the software development projects following agile based methodologies should be tailored based on the client’s perspective with respect to each state of the project.

7.5. Software Development Process Tailoring

Table 7.10 summarizes the results of qualitative part of the study on software development process tailoring procedure. A number of tailoring activities were repeating as shown in sections 5.4.1 to 5.4.3 in chapter 5, and sections 6.4.1 to 6.4.3 in chapter 6. Repeating activities have been calculated in Appendix K and are shown in Table 7.10. The research questions RQ1 and RQ5 as presented in chapter 1 have been addressed in this section.

Table 7.10 shows that during software process tailoring add, delete and modify operations/activities were performed most in number in all the projects except project ‘C’ where merge tailoring activity was performed greater in number than the modify activity as shown in Fig. 7.3. On average add, delete and modify were the highest performed tailoring operations with 33.66, 17.66 and 14.66 average values respectively as shown in Table 7.10.

Table 7.10 Process Tailoring Summary of Results

Results of the Study		Project A	Project B	Project C	Average Number of Tailored Processes/ Activities
Total Number of Processes/Activities Tailored	**Add**	42	35	24	33.66
	Delete	21	19	13	17.66
	Modify	19	17	8	14.66
	Split & Select	11	8	6	8.33
	Merge	9	14	10	11
	Shrink	9	5	4	6
	Wrap up	10	11	7	9.33
Total Number of Processes/Activities Tailored in a state	**Takeoff**	16	17	21	18
	Running	37	31	20	29.33
	Landing	26	21	12	19.66
	Hang up	42	40	19	33.66
Total Number of Processes/Activities Tailored in a Phase	**Resource Management**	30	39	26	31.66
	Communication	31	20	15	22
	Requirement Management	60	50	31	47
Total Activities Tailored		**121**	**109**	**72**	**100.66**
Repeating Activities		**75.62**	**83.84**	**0**	**53.15**

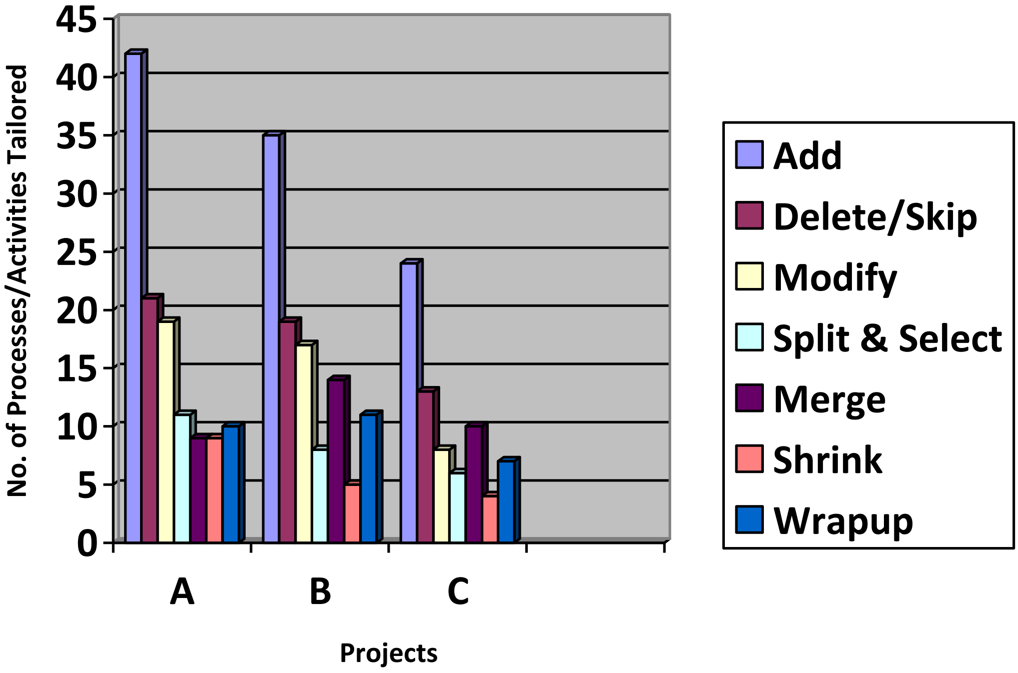

Fig.7.3 Process Tailoring in Each Project

Fig. 7.3 shows the comparison of tailoring activities in each project and describes that these basic set of activities are required to be performed throughout the project lifecycle following agile based methodologies in small and medium sized software development companies.

The results as summarized in Table 7.10 show that the highest tailoring activities having average values of 33.66 and 29.33 were performed during hang up and running states (project maturity) respectively which shows that maximum process tailoring should be performed during these states and particularly when project is in hang up to resolve the issues and problems.

The numbers of activities tailored during each state of the projects are shown in Fig. 7.4. Process tailoring was performed minimum during takeoff state in projects ‘A’ and ‘B’. As discussed earlier that takeoff state in project ‘C’ being quite short was not prominent. It was overlapping with running state therefore, tailoring during takeoff state in project ‘C’ was more than the other projects as shown in Fig. 7.4.

Due to the client's non cooperative behavior overall less process tailoring was performed in project 'C' than the other two projects.

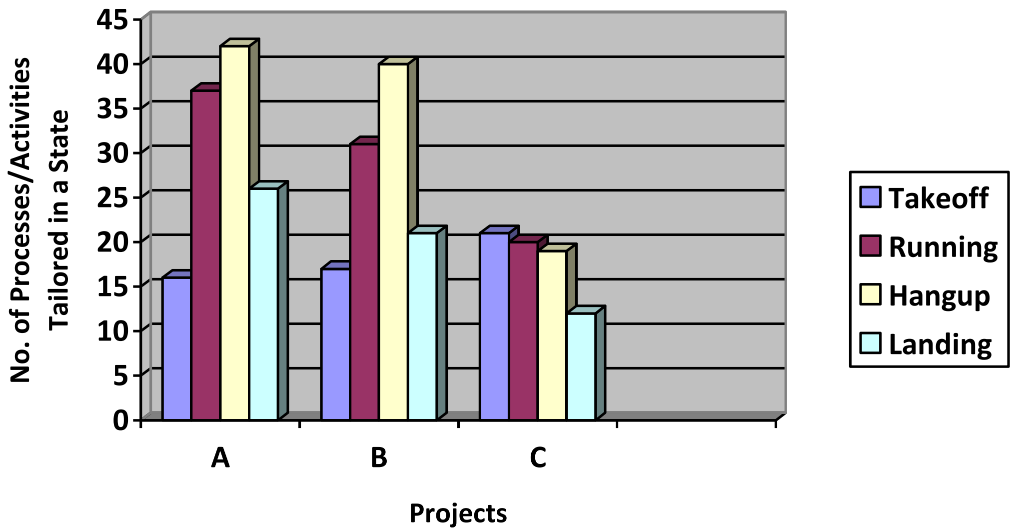

Fig. 7.4 State wise Process Tailoring

Fig. 7.4 shows that minimum process tailoring has been performed during takeoff state, and maximum during hang up state. On the other hand moderate process tailoring has been performed during running and landing states.

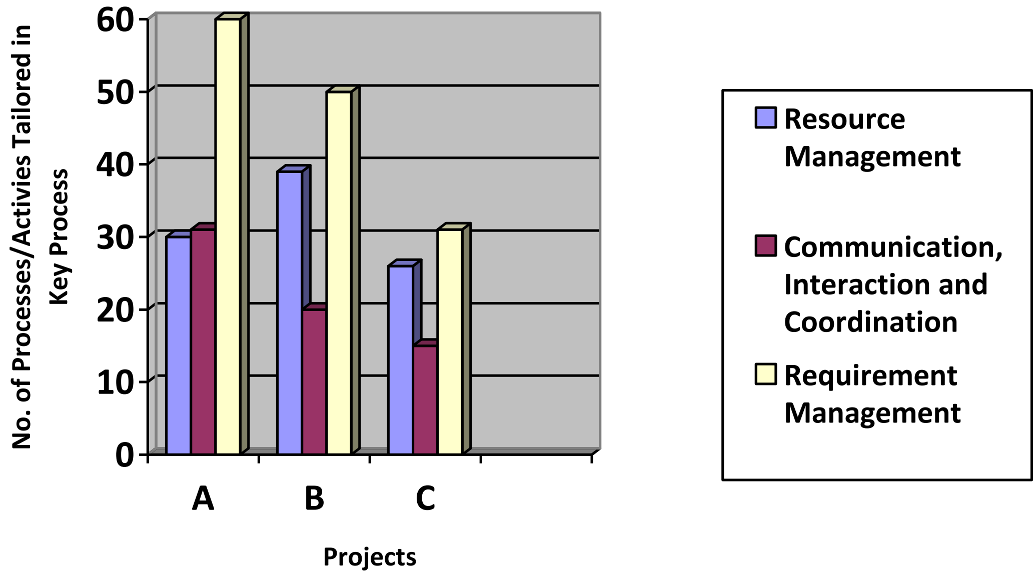

Fig. 7.5 Key Process wise Process Tailoring

The requirement management and resource management processes were the highest tailored processes with 47 and 31.66 average numbers of activities tailored respectively as shown in Table 7.10 and Fig. 7.5. The results show that clients in all three projects were mostly concerned with the resource management and requirement management processes or activities as compared to the communication, interaction and coordination process which was considered as internal process of the project team. However, communications processes in project ‘A’ were tailored more as compared to the resource management processes as shown in Fig. 7.5 which truly represents the behavior and expectations of the client regarding these processes in all the projects.

The findings support that resource management, requirement management and communication processes in software development project following agile based methodologies are the most important for the client as well as the projects.

Fig. 7.6 shows the total number of tailoring activities and total number of repeating activities in each project. The highest tailoring was performed in project ‘A’, then ‘B’ and ‘C’ respectively. The maximum numbers of activities 83.84 were repeated in project ‘B’ as shown in Table 7.10. The repetition factor 0 of project ‘C’ shows that there was no repetition in this project as has been shown in Fig. 7.6.

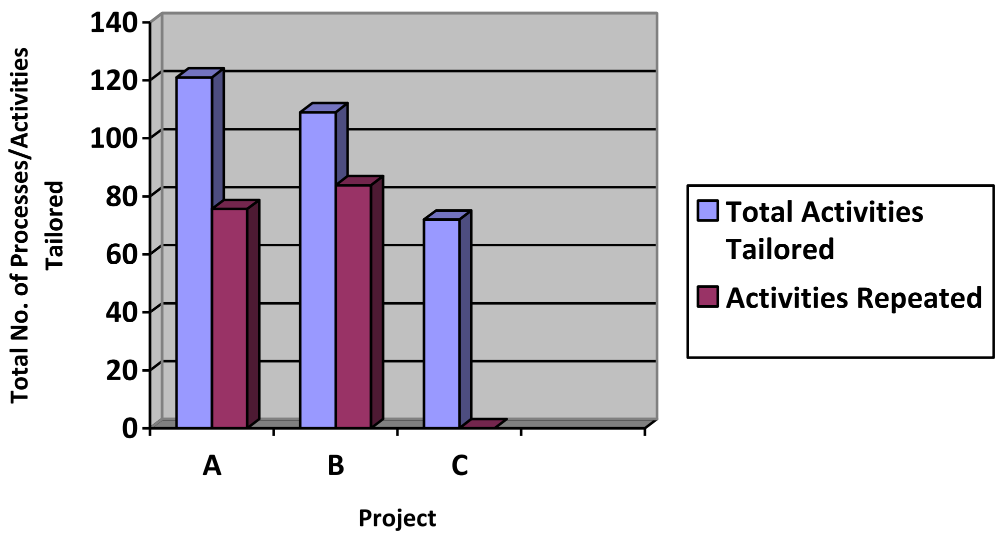

Fig. 7.6 Overall Process Tailoring in Projects

The repetition of the activities is the part of the tailoring process. Tailoring activities during the whole process can be repeating as per requirement. However, the repetition factor has been calculated to have an idea about the number of activities repeated during the tailoring process, otherwise it has no implications on the applicability of framework for process tailoring.

Project ‘A’ was facing problems since its beginning therefore maximum process tailoring was performed in this project. Client’s perspective in this project was not initially met but later on by understanding it maximum tailoring was performed to resolve the issues in the project. In project ‘B’ client’s perspective was met since its beginning therefore process tailoring was performed at moderate levels whereas, in project ‘C’ though client’s perspective was understood but client’s satisfaction could not be achieved. More process tailoring was required in this project, but unsupportive and uncooperative attitude of client could not make it possible.

7.5.1. Quantitative Findings on Process Tailoring

As discussed about qualitative results in the previous section that add, delete, modify are the basic tailoring operations which are performed in all the projects. The same results have been found through quantitative part. As shown in Table 7.11, the high mean value of 3.99 shows that majority of the industry practitioners agree that mostly process tailoring is performing based on these basic tailoring operations in the projects.

Table 7.11 Process Tailoring Activities Assessment

Process Tailoring	Mean	Std. Dev.
Basic Operations (Q31)	3.99	0.8342
Remaining Works Completion (Q36)	3.68	0.8989
Process Activities Reduction (Q37)	3.72	0.8565

Similarly, shrink and wrap up tailoring operations as performed during the landing state of the projects have also been validated by the quantitative results having mean values of 3.68 and 3.72 respectively as shown in Table 7.11. It shows that during landing state these specific tailoring operations are required to perform because of the completion of the project.

Table 7.12 shows the quantitative results on the minimum, maximum and moderate tailoring scales having mean values of 3.25, 3.95 and 3.42 respectively. The mean value of 3.25 shows that maximum process tailoring should be performed when projects have has been matured. Similarly, as shown in Tables 6.14, 7.11 and 7.12 that 3.68 and 3.72 high mean values support specific process tailoring scales.

Table 7.12 Process Tailoring Scales Assessment

Tailoring Scale	Mean	Std. Dev.
Maximum Tailoring (Q34)	3.25	1.0358
Minimum Tailoring (Q35)	3.95	0.8210
Moderate Tailoring (Q33)	3.42	0.9953
Specific (Q36) (Q37)	3.68 3.72	0.8989 0.8565

In addition, for fully matured and stable projects minimum process tailoring should be performed while moderate or minimum process tailoring should be performed at early stages of the project. Furthermore, as discussed earlier specific tailoring activities should be performed as shown in Table 7.12 with mean values of 3.95 and 3.42, and as shown in Table 7.10.

Furthermore, as discussed earlier that client's perspective is an important factor for the project's success in small and medium sized software development companies therefore process tailoring should be performed according to the client's perspective which is an ultimate requirement of the projects in these companies as shown by high mean value of 3.91 in Table 7.13.

Table 7.13 Process Tailoring Approach Assessment

Tailoring Approach	Mean	Std. Dev.
Project Requirement-Client's Perspective (Q30)	3.91	0.8867
Traditional Approaches – Small and Medium Companies (Q28)	4.01	0.9851
Tailoring Approach – Small and Medium Companies (Q29)	4.04	0.8034

The mean value of 4.01 shows that these small and medium size companies do not prefer to follow traditional software engineering approaches rather tailor their processes as shown by high mean value of 4.04 in Table 7.13.

7.5.2. Lightweight Processes and Activities for Software Development

Addressing the research question RQ5, the case study analysis shows that process tailoring generates a small set of lightweight and effective processes and activities for software development and project management for projects running in such small and medium sized software development companies following agile methodologies. Those activities and processes have been described in Appendices C to F and Table 5.1. The quantitative results as shown in Table 7.14 also recommends performing software process tailoring in these companies for generating and improving the quality of software development processes, process handling and project management practices. The mean value of 3.82 shows that process tailoring effectively generates lightweight processes in these companies. On the other hand, values 3.83 and 3.76 shows that process tailoring is an effective practice in these small and medium companies for project management and management of each state of the project.

Table 7.14 Process Tailoring of Agile Methods Assessment

Process Tailoring	Mean	Std. Dev.
Produce Lightweight Processes (Q32)	3.82	0.8254
Good for Project & Processes in agile based (Q38)	3.83	0.7289
Project State Management (Q39)	3.76	0.7854

The framework provides a lightweight systematic approach of software process tailoring for small and medium sized software development companies.

7.6. Conformance to Agile Manifesto

The framework is based on the practices followed by agile methodologies. The framework conforms the principles of agile manifesto such that the three key process areas of the framework have been derived and adopted from agile manifesto. However, to perform conformance testing [116] it has been compared with the principles of agile manifesto [46], [30] to verify that it is aligned with the agile manifesto and is suitable for projects following agile based methodologies.

The principles of agile manifesto and their compliance to the framework have been described as follows:

1. Satisfaction of client is the highest priority through early and continuous delivery of software.

The framework is based on the client's perspective and emphasizes on the client's satisfaction and expectations throughout the project lifecycle. The framework focuses on the delivery of software instead of unnecessary documentation. Build release plans, milestones and deadlines are given priority to make sure on time delivery to the client. Three components of the framework such as client's composite structure, client's interaction overview model and client's perspective model satisfy this important principle.

2. Always welcome modifications in requirements even late during the development work.

Requirement management has been given the foremost priority in the framework and is one of the important key process area in the framework as shown in Fig. 4.19. The framework fully supports the priority of the client's requirements and managing the requirement process according to the client's requirements.

3. Deliver working software regularly with shorter deadlines.

During the implementation of the framework, deadlines and milestones of all the projects were set. The client's perspective as shown in Fig. 5.3 and 5.4 promotes the iterative based development with shorter deadlines and quick releases. The processes in the framework make sure on time delivery of the builds to the client. The framework considers it as one of the important requirements of the client.

4. Clients and project team must work together daily.

The key roles in the framework have been identified and interaction amongst them has been defined. The client's interaction overview sequences as shown in sections 5.1.2 and 6.1.2 as well as shown in Fig. 6.1 and Fig. 6.2 define this principle. The study emphasizes on frequent, close and regular communication between client and the software development team.

5. The project team should be motivated. Provide them full support they require and develop strong trust relationship with them to complete the project.

The separate processes of resource management and tasks allocation have been defined and emphasis is given on these processes in the framework to manage software development teams that qualify this principle.

6. Face to face communication with the client and among the team members.

Client's interaction overview sequence as shown in Fig. 6.2 has been presented to highlight efficient communication between the client and project team. Even in case of client being offshore, the model bridges the communication gap and derives the quick and lightweight communication techniques. Communication, interaction and coordination has been identified and made an important element of the framework as shown in Fig. 4.19.

7. The working software is the best measure of the progress of the project.

The number of milestones achieved, number of builds released and planning of upcoming builds are also the set criteria to measure the progress of the project and performance of the team members in the framework.

8. Sustainable development through constant pace among all the key roles till the project ends.

Lightweight processes derived through the framework make sure the continuous pace of development throughout the project. The technique presented in the framework ensures the continuity and maturity of the processes. Interaction among all the key roles has been emphasized in the framework as shown in Fig. 6.2.

9. Emphasis on technical expertise.

The framework emphasizes and derives the processes of team hiring, resource management, effort distribution, resource leveling and tasks allocation, as well as problem solving approaches to improve the technical expertise of the resources.

10. Simplicity is necessary.

Only three most necessary processes such resource management, communication and requirement management have been used by the framework. All three processes have been kept limited to most necessary artifacts. Only minimum documentation related to project plans and status reports have been maintained by the framework to keep the development simple. Emphasis is given only to the minimum and most required processes only. The framework derives the simplest processes through process tailoring component.

11. The teams should be self organizing and should be competent enough to derive the requirements and architectures at their own.

12. The team should become more effective by the self tuning and adjustments in its approach and behavior as required.

The process tailoring of resource management, and communication, interaction and coordination processes enables the project manager and team members to improve themselves.

The framework of process tailoring and its all components comply completely with the principles of agile manifesto. The framework is based on agile approach and the practices and processes that are the outcome of the framework are also agile. They meet all the principles of agile manifesto and fully qualify to it.

7.7. Comparison With Existing Work

The results of the research work have been compared with another similar kind of process tailoring approach presented by [5]. The comparison has been presented in Table 7.15.

Table 7.15 Comparison of Results with Existing Work

Comparison Factors	Existing Work [5]	Present Work
Number of Companies	2	More than 15
Participants	Developer, Designer, Requirements Analyst, Test Engineer, Project Manager, Process Engineer,	Software Engineer, Senior Software Engineer, System Analyst, Senior/Top Management, Management position
Research Methodology	Qualitative	Qualitative and Quantitative
Data Collection	Semi-structured interviews, documentation review	Case study (action research) – Observations, archival data, Questionnaire
Data Analysis	With-in case analysis Cross case analysis	With-in case analysis Cross case analysis
Aligning with Agile Manifesto	No	Yes
Tailoring Factor(s)	Project goals and environment	Project client, Software Project
Challenges	Resources, Communication, Requirement Management, Political, Technical	Resource Management, Communication, Interaction and Coordination, Requirement gathering and Tracking, Tasks allocation
Tailoring Strategies	Add, Downsize, Drop/Skip, Expand, Redefine, Replace	Add, Delete/Skip, Modify, Merge, Split & Select, Shrink, Wrap up
Tailoring Procedure	Before project	Before and during project
Implementation Procedure	Not defined	Well defined through tailoring scales and schema

Table 7.15 Comparison of Results with Existing Work (Continued)

Comparison Factors	Existing Work [5]	Present Work
Scope/Level of Tailoring	Overall processes, whole project at broad level	Micro-level with respect to project states and challenges
Recommendations	Continuous monitoring of processes and Iterative tailoring.	No iteration, One time tailoring for an iteration or phase. Tailoring only for specific project state for a specific process with specific scale (amount). No need to continuously monitor.
Applicability	General	Small and Medium software development companies
Contribution	General Four step process tailoring framework guideline.	Client's Perspective Meta-model Project States Meta-model Process Tailoring Scales Process Tailoring Schema
Outcome	Time consuming, Excessive tailoring, Process damage, Repetition and inconsistency in processes, Time wastage, Suitable for general development environments. Tailoring strategies are not suitable for agile based methodologies. Suitable for general type of software development companies	Quick and lightweight approach, No repetition, less effort required, Specific to agile methodologies and client factors. Derives lightweight processes. Tailoring strategies most suitable for agile methodologies. Suitable for small and medium sized software development companies.

7.7.1. Similarities and Differences in Implementation and Conduct

The comparison of present study with similar existing work [5] finds prominent and important similarities and differences between both approaches. These similarities and differences clearly highlight the importance of present study and specify how it is different and better than the existing work.

The comparison of similarities and differences has been made on the basis of factors as shown in Table 7.15. The basic similarity between both works is direct contribution to the software development process tailoring research area. On the part of research methodology, in addition to the qualitative research methodology used by both works, the present work also finds supports to the qualitative results from results of the quantitative method. The primary study through review of literature as well as structured interviews has been conducted to develop the framework, while case study of real projects and questionnaire method has also been used to validate the framework in the present study. The results of the present study have been presented in detail in chapter 5, 6 and previous sections of this chapter. The existing work formulates the framework based on the findings of the interviews but results of further validation of the derived framework have not been presented. Similarly, data has been collected from the industry practitioners in both studies but more than 15 companies and three real projects were involved in present study which is quite more as compared to the existing work. Both approaches have made within case and cross analysis of the findings but more comprehensive and detailed analysis has been presented in the present study.

These implementation specifications of both works show that present study has been conducted in more real software development environments as compared to the existing work. It is noticeable that unlike proposed framework, the existing study has not been implemented in real projects in any company in order to be validated. It is based only on the inductive research approach. The present study uses the inductive research approach but also further validates it by implementing it in real projects and through questionnaire method. This makes proposed framework more authentic and applicable.

7.7.2. Improvements and Enhancements of Present Study

As shown in Table 7.15, the proposed framework is applicable in the agile based software development projects of small and medium sized software development companies. On the other hand, the existing work presented by [5] focuses only on general kind of software development environments.

Similarly, out of various challenges on which focus is given in the existing work, the proposed framework addresses only three areas which are directly related to the agile based methodologies as shown in Table 7.15. It makes the proposed framework more specific and applicable to agile based software development projects of small and medium sized companies. Unlike present study, no implementation details have been provided for the existing approach.

It has been found that proposed framework tailors the processes, sub-processes and activities when project is in progress while approach of tailoring the process only once before the beginning of the project has been adopted in the existing work. Furthermore, the proposed framework is based on the critical success factors of projects running in small and medium sized companies. The present study presents client's perspective model, project states meta-model which makes the proposed framework more applicable and suitable for small and medium sized companies due to their close relevance with agile based methodologies.

The comparison of both studies shows that the proposed framework provides a specific and lightweight process tailoring technique which has not been provided in the previous process tailoring works. The basic advantage of current approach over previous approach is its non-iterative behavior and target oriented process tailoring following a process tailoring schema.

The previous approach though has presented good general level guidelines for process tailoring but has been unable to focus on critical factors of a specific process and project. In general, this is a good approach but practically it is inapplicable for small and medium sized software development companies. Further, missing compliance testing raised the questions on applicability and suitability of this approach in agile based environments. The proposed framework precisely describes when and how to perform process tailoring and recommends best tailoring practices. Focusing on the most critical factor client makes it more applicable in all kinds of projects according to agile based methodologies.

The present study has been seen as an enhancement in the existing works on process tailoring because it is completely based on the new factors as described in chapter 4 and case studies in chapter 5 and 6. Properly for the first time, the structure of the client and client's perspective model has been presented. The approaches defined by the framework have been found very supportive and helpful in achieving the objectives of principles of agile manifesto. Further, the real practices of software development industry and approaches of the project managers have been highlighted through the framework and provide guidelines in understanding the actual problems and issues of the small and medium sized software development companies.

The present study has significantly contributed a systematic solution of software process tailoring for small and medium sized software development companies following agile based methodologies.

7.7.3. Suitability For Small and Medium Sized Companies

The framework is based on the real practices and processes followed by small and medium sized software development companies. It provides guidelines to these companies to define good and lightweight processes for their agile based software development projects. The framework performs process tailoring at run time during the execution of project throughout its lifecycle. Though it may be considered as an extra work in small and medium sized software development companies, but actually the framework recommends tailoring a software process or its activities for a particular state of the project instead of whole process, and thus eliminates this factor. Suggesting process tailoring at micro level for a particular state and key process does not involve resources cost overhead as in tailoring the whole process at once. The process tailoring approach meets the properties of a good process such as fulfilling the expectations, followed by the team members of the company and meeting their growing needs [28].

The comprehensive and detailed case studies of real projects running in small and medium sized software development companies provide a deep insight into the real issues and practices of such companies making the framework a more realistic approach. The present study addresses the issue of absence of a realistic case study in software engineering research works [6] by presenting these real case studies.

Focusing on the real industry practices, the research emphasizes on the needs of industry oriented research work to address the real industry issues by producing realistic models and frameworks in this regard. This is required by the software engineering research community as well as industry professionals to start collaborative research projects with the effective involvement of industry professionals. In order to achieve this objective, the industry professionals may need to realize the importance of academic research works and should be encouraged to find the solutions of their problems through it.

Realizing the importance of challenging role of project manager in agile based development environments [10], the present study would also be helpful to the project managers of small and medium sized companies for effective project management through process tailoring.

The success of agile based development demands the active participation of the client in the software development process [10]. It is the requirement of small and medium sized software development companies to realize the client as a major factor of success in their projects and formally making client's perspective a part of the process.

The study presented in this thesis can resolve the issues of small and medium sized software development companies by providing complete guidelines to select, derive or reuse an existing software development process. It will make such companies able to produce more good quality software through lightweight software development processes.

7.8. Summary

The results and discussion show that results of questionnaire having high mean values fully support the findings of the case study as well as completely validates the process tailoring framework. Detailed discussion and analysis has been presented by comparing both case study and questionnaire results with each other.

Furthermore, compliance of the framework with principles of agile manifesto has been made which shows that process tailoring framework qualifies to be an agile based process tailoring approach. The comparison of results with an existing work shows that the present study has made significant improvements and enhancements in the area of software development process tailoring.

CHAPTER 8

CONCLUSION

Overview

This chapter presents the conclusion of the present research study. It describes the application of work presented in this thesis as well as its role in addressing the software processes issues in small and medium sized companies. It further describes how this work is different and novel from existing works. This chapter highlights the need of process tailoring for the agile based development environments of small and medium sized companies and usefulness of present study in this regard. Finally, it presents the future work and directions of the present study.

8.1. Summary of Results

The results of both qualitative and quantitative analysis as presented in chapter 5, 6 & 7 have been concluded in this section according to the process tailoring framework.

i. The client of the project is a composite entity that contains various roles of client as shown in Fig. 5.1 but it is possible that any project have only one client (see sections 5.1.1, 6.1.1, 6.11.3 and 7.2).
ii. In agile based projects, client's perspective is one of the most important factors for project success (see sections 5.2, 6.2, 6.11.3 and 7.2).
iii. The client's perspective should be understood and maintained throughout the project lifecycle (see sections 5.2 and 6.2, 6.11.3 and 7.2).
iv. In projects following agile based methodologies, the minimum processes should be adopted according to the expectations of the client from the project team (see sections 5.2.1, 5.2.2 and 6.2.2).

v. Each project passes through different conditions throughout its life which are termed as states of the project which indicates the status and progress of the project (see sections 5.3, 6.3 and 7.4).

vi. Each project faces problems and issues and thus enters into the hang up state at least once in its life (see sections 5.3.1, 5.3.2, 5.3.4, 6.3.1, 6.3.2, 6.3.3 and 7.4).

vii. The response and behavior of the states vary from project to project (see sections (see section 5.3.2, 5.3.4, 6.3.3 and 7.4).

viii. The states may exist in parallel to other states/sub-states and can be overlapping (see sections 5.3.4, 6.3.3, 6.3.5 and 7.4).

ix. Software development processes should be tailored according to the tailoring scales with respect to project states and key processes (see sections 5.4, 6.4, 6.11.3, 6.12, 7.5, and Appendix L).

x. Process tailoring generates a small set of very lightweight processes for projects following agile based methodologies running in small and medium sized software development companies (see sections 5.4, 6.4 and 7.5.2).

xi. Software process tailoring based on client's perspective effectively addresses the issues of software development processes in small and medium sized software development companies (see sections 5.4, 6.4 and 7.5).

xii. The process tailoring provides a short and quick approach to the project managers to manage the projects and understanding the client's expectations (see sections 5.4, 6.4 and 7.5).

These results have also been discussed in chapter 7 in relation to the quantitative results. The quantitative results in detail have been presented in sections 6.11.3 in chapter 6 and sections 7.2.1, 7.3, 7.4 and 7.5 in chapter 7.

Quantitative results on average provide 80% and above support to the findings of the case study and working of the process tailoring framework. The support of both qualitative and quantitative results to each others as presented in chapter 7 shows that proposed tailoring framework is completely applicable in small and medium sized software development companies.

The process tailoring framework and all of its components provide full support to the project managers to manage their projects and software development processes. It has made quite easy for them to understand the client, customize the processes and selecting suitable processes for their project. The framework supports the software process tailoring at the runtime during execution of the project. It helps the project managers to understand the condition and status of the project and adopt the processes and strategies accordingly.

The process tailoring framework presents the fundamental components such as client and client's perspective model, project states meta-model, key processes and problems classes which provide ground work for tailoring processes based on client's perspective. Moreover, the present study shows that process tailoring approach can effectively be used to redesign, regenerate and reuse the existing processes. The process tailoring framework can be further extended and inherited as well.

8.2. Application of Present Study

As discussed in chapter 1, Global Software Development (GSD) has changed the overall software development trends. An increasing number of small and medium sized software development companies as a result of GSD, due to the limitations of resources follow the agile based methodologies and adapt the software development processes according to their requirements. The tailoring framework presented in this study provides guidelines to the small and medium sized software development companies on tailoring their agile based processes to derive lightweight software development processes according to their requirements.

The framework focuses on tailoring three key processes on which agile based methodologies mainly emphasize such as resource management, communication, interaction and coordination, and requirement management [5]. Good software process tailoring is important for producing good software development processes as well as improving their quality. In GSD environments where software development practices are rapidly changing, software process tailoring should be practiced by the software development companies as a part of process to meet the requirements of their projects and clients [5]. The process tailoring framework presented in this thesis considering GSD environment suggests the process tailoring practices.

The value of end product delivered to the client is very important in projects following agile based development [10]. The lightweight processes and practices derived through tailoring framework contribute high value to the end product in terms of quality and satisfaction of the client which are important for these small and medium sized companies in establishing their good repute in international markets.

8.3. Contribution of Research

The major contribution of the present study is identification of client's perspective factor, resource management, communication, interaction & coordination, and requirement management as three key processes based on agile methodologies, and project states meta-model as the fundamental elements of the framework that provide ground work for tailoring the software development processes in small and medium sized companies. To the best of my knowledge, this is the first applied work of this kind on software development process tailoring for small and medium sized companies.

The detailed discussion in the context of these elements as well as research questions and objectives has been made in the following section.

- *The client's perspective model describing how client's perspective should be understood practiced and maintained throughout the project lifecycle.*

The importance of project client has been realized in the software engineering process models, frameworks and research works but unfortunately rarely any research work highlight its various aspects and analyze this critical success factor. Similarly client's perspective and its importance in software development projects have not been presented in existing research works. The present study explains the concept of client's perspective, how it should be understood, maintained and practiced throughout the software development project lifecycle. In this regards, the client's perspective model as shown in Fig. 5.3, section 5.2.2 in chapter 5 is the major contribution of the present study which has not been presented in any other software engineering research works.

- *The classification of problems/risk into three main classes according to their effect on software development processes and projects.*

The present study to the best of my knowledge for the first time classifies the problems or risks faced by the projects into three main categories as presented in Table 4.1, section 4.2.4.2 in chapter 4. Problems and risks have been presented in many existing research works but no classification has been made. The classification nomenclature presented in this thesis helps in understanding the problem types, their area of impact and resolution. This is the contributory work of the present study in problems or risks management areas.

- *The project states meta-model that describes the behavior and status of software development projects.*

A lot has been written on project lifecycle or lifespan phases in the existing research works. Project lifecycle or lifespan phases do not comply with the working of agile based methodologies which do not follow typical process meta-model definitions. According to my strong belief, project states meta-model truly reflects the flow and progress of software development projects according to the agile based methodologies. The states presented in the project states meta-model as shown in Fig. 4.13 helps in understanding the condition and current status of the project as well as its past progress pattern. The project states meta-model is the contributory work towards the project lifecycle, project management, process management and risk resolution areas that helps in understanding the behavior of the projects with respect to various factors.

- *A lightweight software development process tailoring framework and implementation schema for small and medium sized software development companies.*

Existing research works on software development process tailoring do not describe implementation strategies of process tailoring activities. The proposed process tailoring framework and process tailoring schema addresses this limitation of existing works and describes that how various process tailoring activities or operations to tailor the agile based software development processes and activities should effectively be performed and practiced in small and medium sized software development companies. Both of the process tailoring framework and tailoring schema are original contribution of the present study in the area of software development process tailoring of agile based methodologies being practices in small and medium sized companies.

- *Generating smart and lightweight software development processes and activities through process tailoring for client intensive environments.*

The process tailoring framework effectively produces the agile based lightweight processes and activities through process tailoring. Small and medium sized companies mostly rely on lightweight processes and activities for their software development projects. The existing approaches of process tailoring do not produce such lightweight processes and activities which have been addressed by the proposed process tailoring framework in the present study. This has been considered as the major contribution of the present study to produce lightweight processes and activities through software development process tailoring.

In short, the present study presents a novel and lightweight approach of software development process tailoring that gives a new direction to the process tailoring research works and opens new doors for the researchers.

8.3.1. Achieving Research Objectives and Satisfying Research Questions

The contribution of the present study to the body of knowledge completely addresses all the research questions and achieves the research objectives as presented in sections 1.6 and 1.7 respectively in chapter 1 at page 11 . In this section the contribution of research as presented above in relevance to each of the research objectives has been discussed in detail.

The *first objective* of this study has been achieved by addressing the research questions *RQ2 and RQ3* as presented in section 1.6 in chapter 1. It has been achieved by analyzing client, its structure, various roles in software development projects, interaction with project teams, and expectations and requirements. Findings of the case studies as presented in chapter 5 & 6 and quantitative results in chapter 7 show that client is an important and critical success factor in small and medium sized software development companies which practice agile based methodologies. Moreover, it is necessary to understand and maintain client's perspective throughout the project lifecycle which is important for the success of the projects running in small and medium sized software development companies.

Therefore, client and client's factor being important success factors in software development projects in small and medium sized companies are the fundamental components of the proposed tailoring framework. The present study shows that process tailoring should be performed based on the client's perspective keeping in view the client's requirements, expectations, and priorities.

The client's composite structure helps in understanding the client of the project and existence of its various need based roles in the project. Whereas, client's perspective model describes how client's perspective should be understood and maintained throughout the project lifecycle. The meanings of client's perspective, its role and importance in software development projects have also been presented and should be practiced as a continuous part of process and project management. The study shows that role of project's client and client's perspective are very important for the success of agile based projects running in small and medium sized software development companies.

The *second and third objectives* have been achieved by addressing the research questions *RQ2 and RQ4* through presenting a project states meta-model which is based on the problem classes and client's perspective factors. The present study shows that problems or risks faced by the projects are of three main types such as performance minor, management critical and progress limiter. Each problem class truly reflects the nature and impact of the problems or risks that it contains. These problem classes show that problems or risks in projects never ends and are part of an ongoing problem or risks lifecycle as shown in Fig. 4.7 in chapter 4.

The problem classes and client's perspective elements are the major determinants of the project progress and project execution flow states. Behavior of the projects is also mainly dependent on these two factors.

Project states meta-model as shown in Fig. 4.13 in chapter 4 describes the behavior of the software development projects and their execution flow states such as takeoff, running, landing and hang up on the basis of problem classes and client's perspective factors. Project states meta-model presents the real picture of how project's behave in small and medium sized companies, how they respond to various factors and how they progress and pass or enter from one state to the other. Agile based methodologies do not follow the typical definitions or phases of a process lifecycle which makes it difficult to observe different phases of a project. Therefore, project states meta-model address this issue and presents a true picture of lifespan of projects which are based on agile based methodologies. It has been found that project states meta-model actually describes the condition of project at some particular time which helps in understanding its current status in terms of completion, success and growth.

The *fourth objective* has been achieved by addressing the research questions *RQ1 and RQ5* through formulating a framework presented in chapter 4 at page 100 for tailoring agile based processes in small and medium sized software development companies. The framework is mainly based on the principles of agile manifesto which makes it more applicable to agile based software development environments of small and medium sized companies. As in principles of agile manifesto, more emphasis has been given on involvement of client in software development projects therefore the process tailoring framework particularly considers client as an important factor in process tailoring. Three key process areas such as resource management, communication, interaction and coordination, and requirement management are the key processes on which client is mostly concerned and are based on principles of agile manifesto are the applied part of the framework.

The framework recommends tailoring these three key processes of agile based methodologies such as resource management, communication, interaction and coordination, and requirement management. These key processes are tailored when software development project is in a particular state. Therefore, process tailoring at micro level of project can be performed. Instead of tailoring all the phases of a complete process, the framework focuses tailoring only key processes as mentioned earlier and presented in case studies in chapter 5 & 6. The approach of tailoring a key process when project is in a particular state makes the framework more realistic and applicable for agile based software development projects in small and medium sized software development companies. This characteristic approach of the framework has not been followed previously and is the most original part of the proposed framework and is a novel approach.

The *fifth and last objective* has also been achieved by addressing the research questions *RQ1 and RQ5* through generating a process tailoring schema as presented in Table 6.16 in chapter 6 which describes that when and how many tailoring activities should be performed. The tailoring schema applies minimum, moderate, maximum and specific tailoring operations as identified during case studies as presented in chapter 5 & 6 and summarized in Table 6.15.

Process tailoring framework suggests that for agile based projects running in small and medium sized software development companies, only three key processes such as resource management, communication, interaction and coordination, and requirement management are required to be tailored with respect to the particular state of the project such as takeoff, running, landing, and hang up. In addition, the tailoring schema also describes the tailoring activities that should be performed at minimum, moderate, maximum and specific levels. The proposed tailoring schema provides hands on guideline to the project managers on performing tailoring operations effectively to manage their projects and processes.

In addition to achieving all these objectives of the present study, the process tailoring framework and its components such as client's perspective, key processes, and project states meta-model are the significant contribution of the research study. The case study of real projects makes the proposed framework more applicable and suitable for the agile based software development projects running in small and medium sized software development companies.

8.4. Impact of Framework on Case Study Projects

Qualitative results show that process tailoring framework effectively tailored the software development processes in real projects and helped to resolve the various issues faced by these projects. Mainly all the projects were facing issues related to the resource planning, effort distribution of team resources, requirement gathering and tracking as well as communication and coordination with the client and amongst the team members. To address these issues, framework modified and customized the respective processes and activities to fulfill the client's expectations and requirements.

The findings of case studies and their analysis as presented in chapters 5 and 6 shows that process tailoring practices successfully ensured that:

i. Client's requirements, expectations and priorities have been fulfilled.
ii. Client is satisfied with the processes and progress of the projects.
iii. Major milestones have been achieved.
iv. Issues and problems in the projects have been resolved.
v. Projects have been successfully completed.

In addition, the tailoring framework maintains consistency with project goals and scope. Accordance of detailed case study protocol with framework elements helped in spending sufficient time with each case during the case study. It provided complete support to the framework to be fully implemented in the real projects and making observations and analysis more strong and realistic.

Overall, the tailoring framework helped case study projects to resolve, overcome and avoid existing and anticipated issues and problems. All the projects completed their lives successfully with complete satisfaction of client. Furthermore, it identified those lightweight processes and activities which are most important for the success of agile based software development projects running in small and medium sized companies. The framework also emphasizes on project managers of such companies to focus on the lightweight processes and activities as produced by the framework through process tailoring. The achievement of milestones in projects, fulfillment of project goals and successful completion of the projects show that proposed process tailoring framework successfully achieves all its objectives and effectively addresses the issues of software development processes in small and medium sized companies.

8.5. Future Work

In an extension to the existing work, a research study will be performed to identify the attributes of both client's perspective model and project states meta-model. The identification and validation of attributes of various components of a meta-model has been considered as very complex and time consuming work, therefore, it will be completed in a separate research study.

A detailed study on the project hang up state has also been planned in the future research works. A lot of effort is required to identify the factors affecting the hang up state, and measures and practices that should be adopted to avoid or eliminate such factors. The average time a typical software development project passes in a particular state and whether this time increases the total duration of the project is yet to be explored in future works. Formalizing the micro project management concept into a framework for small and medium sized software development companies is another potential research area.

In another study, the scaling of process tailoring framework suitable to the development environments of growing and larger companies will also be performed. The framework is expected to be inherited for future models in similar kinds of works as well as improvements will be also be made.

REFERENCES

[1] S. Zahran, “Software Process Improvement: Practical Guidelines for Business Success,” *Addison-Wesley.*, 1998.

[2] M. P. Ginsberg, and L. H. Quinn, “Process Tailoring and the Software Capability Maturity Model,” *Technical Report, Software Engineering Institute (SEI)*, 1995.

[3] I. C. Yoon, S. Y. Min, and D. H. Bae, “Tailoring and Verifying Software Process,” *Proceedings of the 8th Asia-Pacific Software Engineering Conference*, pp. 202-209, 2001.

[4] F. Dai, and T. Li, “Tailoring Software Evolution Process,” *8th ACIS International Conference on Software Engineering, Artifical Intelligence, Networking and Parallel/Distributing Computing*, pp. 782-787, 2007.

[5] P. Xu, and B. Ramesh, “Using Process Tailoring to Manage Software Development Challenges,” *IEEE Computer Society ITPro*, 2008.

[6] O. Pedreira, M. Piattini, M. R. Luaces *et al.*, “A Systematic Review of Software Process Tailoring ” *ACM SIGSOFT Software Engineering Notes*, vol. 32, no. 3, 2007.

[7] J. Cho, “Globalization and Global Software Development,” *Issues in Information Systems,* vol. VIII, no. No.2 2007, 2007.

[8] N. M. Rao, “Challenges in Execution of Outsourcing Contracts,” *ACM ISEC'09*, 2009.

[9] M. Malarvannan, “Managing Offshore Software Teams,” *Available: http://outsourceportfolio.com/managing-global-software-teams/*, 2009.

[10] S. Nerur, R. Mahapatra, and G. Mangalaraj, “Challenges of Migrating to Agile Methodologies,” *Communications of the ACM,* vol. 48, no. 5, pp. 73-78, 2005.

[11] J. G. Brodman, and D. L. Johnson, “A Software Process Improvement Approach Tailored for Small Organizations and Small Projects,” *Proceedings of International Conference on Software Engineering (ICSE 97).* pp. 661, 1997.

[12] “CMMI Development Team, Capability Maturity Model - Integrated Version 1.1, Software Engineering Institue ” *Available: http://www.sei.cmu.edu/cmmi/*, 2001.

[13] “International Organization for Standardization, ISO Standard 9001: Quality Management Systems,” 2000.

[14] M. J. Harry, and R. Schroeder, “Six Sigma: The Breakthrough Management Strategy Revolutionizing the World’s Top Corporations,” *Crown Publishing Group*, 2006.

[15] M. Poppendieck, and T. Poppendieck, “Lean Software Development An Agile Toolkit,” *Addison-Wesley.*, 2003.

[16] “European Commission, Commission Staff Working Document on the Implementation of Commission Recommendation of 6 May 2003 Concerning the Definition of Micro, Small and MediumMedium-sized Enterprises,” *Available: http://ec.europa.eu/enterprise/policies/sme/files/sme_definition/sme_report_2009_en.pdf*, 2009.

[17] H. Tarawneh, A. Elsheikh, and S. Lahawiah, “Web-Based Applications Development in Small Firms ” *Proceedings of the 6th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems,* 2007.

[18] "Small & Medium Enterprises Development Authority, SME Definitions," *Available: http://www.smeda.org.pk/main.php?id=2*, 2007.
[19] "Small & Medium Enterprises Development Authority, SME Definitions (Proposed under SME act)," *Available: http://www.agroasia.net/iac/presentations/SMEDA.pdf*.
[20] O. P. Ardic, N. Mylenko, and V. Saltane, "Small and Medium Enterprises A Cross Country Analysis with new Data Set," *The World Bank Policy Research Paper*, 2011.
[21] "Define SMEs, Defining Small and Medium Enterprises (SMES)," *Available: http://www.smeinfo.com.my/index.php/en/sme-definition/sme-definition/ defining-sme*, 2011.
[22] H. Glazer, J. Dalton, D. Anderson *et al.*, *CMMI or Agile: Why Not Embrace Both!*, Software Engineering Institute, 2008.
[23] J. A. Kim, S. Y. Choi, and T.-H. Kim, "Management Environment for Software Process Improvement," *International Symposium on Computer Science and its Applications, IEEE*, 2008.
[24] J. Highsmith, and A. Cockburn, "Agile Software Development: The Business of Innovation," *IEEE Computer*, vol. 34, no. 9, pp. 120-127, 2001.
[25] B. Boehm, "Get Ready for Agile Methods, with Care," *Computer*, vol. 35, no. 1, pp. 64-69, 2002.
[26] "Principles Behind the Agile Manifesto Agile Alliance," *Available: http://agilemanifesto.org/principles.html*, 2001.
[27] C. Ferreira, and J. Cohen, "Agile Systems Development and Stakeholder Satisfaction: A South African Empirical Study," *Proceedings of SAICSIT'08*, pp. 48-55, 2008.
[28] G.Ibraguengoitia, J. A. Salazar, M. G. Sanchez *et al.*, "A Procedure for Customizing a Software Process," *Proceedings of the Fourth Mexican International Conference on Computer Science (ENC'03)*, 2003.
[29] N. Feng, M. Li, and H. Gao, "A Software Project Risk Analysis Model Based on Evidential Reasoning Approach," *Proceedings of the 2009 WRI World Congress on Software Engineering*, vol. 2, pp. 224-228, 2009.
[30] T. A. Manifesto, "The Manifesto for Agile Software Development," *Available: http://www.agilealliance.org/the-alliance/the-agile-manifesto/*, 2001.
[31] H. Taylor, "The Move to Outsourced IT Projects: Key Risks from the Provider Perspective," *Proceedings of the 2005 ACM SIGMIS CPR Conference on Computer Personnel Research*, pp. 149-154, 2005.
[32] G. G. Gable, and W. W. Chin, "Client Versus Consultant Influence On Client Involvement in Computer System Selection Projects: A Two Actor Model of The Theory of Planned Behavior," *Twenty- Second International Conference on Information Systems*, pp. 249-260, 2001.
[33] Y. B. Leau, W. K. Loo, W. Y. Tham *et al.*, "Software Development Lifecycle Agile vs Traditional Approaches," *2012 International Conference on Information and Network Technology*, vol. 37, 2012.
[34] A. G. Rodrigues, and T. M. Williams, "System Dynamics in Project Management: Asessing the Impacts of Client Behavior on Project Performance," *Journal of the Operation Research Society*, vol. 49, no. 1, pp. 2-15, 1996.

[35] J. Hills, “Putting Yourslef in the Client's Shoes A Little-appreciated Skill that can Boost the Success of HR People,” *Human Resource Management International Digest,* vol. 15, no. 7, pp. 35-38, 2007.
[36] E. Hossain, P. L. Bannerman, and R. Jeffery, “Towards an Understanding of Tailoring Scrum in Global Software Development: A Multi-case Study,” *2011 International Conference on Software and System Process,* pp. 110-119, 2011.
[37] Z. Bakalova, and M. Daneva, “A Comparative Case Study on Clients Participation in a Traditional and in an Agile Software Company,” *12th International Conference on Product Focused Software Development and Process Improvement,* pp. 74-80, 2011.
[38] Z. Racheva, and M. Daneva, “Client's Participation in Software Projects: Comparative Case Study Between an Agile and a Traditional Software Company,” *First Workshop on Leveraging Empirical Research Results For Software Business Success,* pp. 5, 2010.
[39] L. McLeod, and S. G. MacDonell, “Factors that Affect Software System Development Project Outcomes: A Survey of Research,” *ACM Computing Surveys,* vol. 43, no. 4, pp. 24-56, 2011.
[40] D. Pernas, “Agile Methods and GSD: The Wrong Solution to an Old but Real Problem,” *Communications of the ACM,* vol. 49, no. 10, 2006.
[41] C. Sterba, T. Grechenig, and M. Pazderka, “Outsourcing as a Strategy for IT Harmonization –A Public Sector Case Study Proposing an Approach in Independent Stakeholder Scenarios,” *Proceedings of ICEGOV'08,* 2008.
[42] R. Narayanaswamy, and R. M. Henry, “Effects of Culture on Control Mechanisms in Offshore Ourtsourced IT Projects,” *SIGMIS-CPR’05,* pp. 139-145, 2005.
[43] N. Ramasubbu, and R. K. Balan, “Towards Governance Schemes for Distributed Software Development Projects,” *Proceedings of SDG’08,* 2008.
[44] A. Gopal, T. Mukhopadhyay, and M. S. Krishnan, “The Role of Software Processes and Communication in Offshore Software Development,” *Communications of the ACM,* vol. 45 no. 4ve, pp. 193-199, 2002.
[45] M. Cusumano, A. McCormack, C. F. Kemerer *et al.*, “A Global Survey of Software Development Practices,” *IEEE Software,* vol. 178, pp. 1-17, 2003.
[46] D. Turk, R. France, and B. Rumpe, “Limitations of Agile Software Processes,” *Proceedings of 3rd International Conference on eXtreme Programming and Agile Processes in Software Engineering, ACM,* 2002.
[47] A. Cockburn, and J. Highsmith, “Agile Software Development: The People Factor,” *IEEE Computer,* vol. 34, no. 11, pp. 131-133, 2001.
[48] J. Highsmith, “Agile Project Management: Principles and Tools,” *Cutter Consortium Agile Product & Project Management Report,* vol. 4, no. 2, 2003.
[49] L. Jiang, and A. Eberlein, “Towards A Framework for Understanding the Relationship between Classical Software Engineering and Agile Methodologies ” *Proceedings of APSO'08,* pp. 9-14, 2008.
[50] V. R. Basili, “The Role of Experimentation in Software Engineering: Past, Current, and Future,” *Proceedings of the 18th International Conference on Software Engineering,* pp. 442-449, 1996.
[51] K. Beck, “Extreme Programming Explained: Embrace Change,” *Addison-Wesley,* 1999.
[52] K. Beck, and M. Fowler, “Planning Extreme Programming,” *Addison-Wesley.,* 2000.

[53] K. Auer, and R. Miller, “Extreme Programming Applied: Playing to Win,” *Addison-Wesley*, 2002.

[54] S. Fraser, K. Beck, W. Cunningham *et al.*, “Hacker or Hero? - Extreme Programming Today,” *Proceedings of the 2000 ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2000),* pp. 5-7, 2000.

[55] C. Larman, “Applying UML Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process,” *Prentice Hall Professional,* 2002.

[56] K. Schwaber, and M. Beedle, “Agile Software Development with Scrum,” *1st Edition, Prentice Hall, New Jersey.,* 2001.

[57] L. Rising, and N. S. Janoff, “The Scrum Software Development Process for Small Teams,” *IEEE Software,* vol. 17, no. 4, pp. 26-32, 2000.

[58] Y. Guo, and C. Seaman, “A Survey of Software Project Managers on Software Process Change,” *Proceedings of ESEM'08,* pp. 263-269, 2008.

[59] L. Carr, S. Kampa, and T. Miles-Board, “MetaPortal Final Report: Building Ontological Hypermedia with the OntoPortal Framework,” *Technical Report, IAM, ECS, University of Southampton*, 2001.

[60] V. E. S. Souza, and R. d. A. Falbo, “An Agile Approach for Web Systems Engineering,” *Proceedings of the 11th Brazilian Symposium on Multimedia and the Web*, pp. 1-3, 2005.

[61] G. B. Wills, N. Abbas, R. Chandrasekharan *et al.*, “An Agile Hypertext Design Methodology,” *Proceedings of HT’07,* pp. 181-184, 2007.

[62] J. Aiken, “Technical and Human Perspective on Pair Programming,” *ACM SIGSOFT Software Engineering Notes,* vol. Volume 29 Number 5, 2004.

[63] L. Williams, and R. R. Kessler, “Pair Programming Illuminated,” *Addison-Wesley*, 2003.

[64] W. H. M. Theunissen, D. G. Kourie, and B. W. Watson, “Standards and Agile Software Development,” *Proceedings of SAICSIT 2003*, pp. 1-11, 2003.

[65] C. Olberding, “Modes of Client Interaction: A Proposal,” *Available: http://www.stationfour.com/blog/post/2007/11/27/modes-of-client-interaction-a-proposal.aspx*, 2007.

[66] R. Jagnannathan, and P. A. G. Sivilotti, “Increasing Client-Side Confidence in Remote Component Implementations,” *Proceedings of ESEC/FSE'01,* 2001.

[67] R. Giordano, and D. Bell, “Participant Stakeholder Evaluation as a Design Process,” *Proceedings of the 2000 Conference on Universal Usability,* pp. 53-60, 2000.

[68] K. D. Joshi, K. M. Kuhn, and F. Niederman, “Multiple Stakeholders’ Perceptions of Critical Attributes of a Top Performer: Exploring the Effects of Congruency,” *SIGMIS-CPR’07,* 2007.

[69] S. Biffl, I. Mordinyi, and A. Schatten, “A Model-Driven Architecture Approach Using Explicit Stakeholder Quality Requirement Models for Building Dependable Information Systems,” *5th International workshop on Software Quality (WoSQ'07), IEEE*, 2007.

[70] A. Leonard, “A Conceptual Framework for Explaining the Value of End User Maturity Levels for IT Management,” *Proceedings of SAICSIT 2005*, pp. 152-158, 2005.

[71] H. W. Lawson, “Defining Stakeholder Relationships,” *Computer,* vol. 32, no. 3, pp. 110-112, 1999.

[72] K. C. Strong, R. C. Ringer, and S. A. Taylor, “The Rules* of Stakeholder Satisfaction (* Timeliness, Honesty, Empathy),” *Journal of Business Ethics,* vol. 32, pp. 219-230, 2001.
[73] B. Mennecke, and J. Bradley, “Making Project Groups Work: The Impact of Structuring Group Roles on the Performance and Perception of Information Systems Project Teams,” *Proceedings of 12th Annual Conference of the International Academy for Information Management*, pp. 19-24, 1997.
[74] G. Beranek, W. Zuser, and T. Grechenig, “Functional Group Roles in Software Engineering Teams,” *Proceedings of the 2005 Workshop on Human and Social Factors of Software Engineering,* vol. 30, no. 4, pp. 1-7, 2005.
[75] C. Jensen, and W. Sacchi, “Role Migration and Advancement Processes in OSSD Projects: A Comparative Case Study,” *29th International Conference on Software Engineering (ICSE'07), IEEE*, 2007.
[76] Y. Dubinsky, and O. Hazzan, “Using a Roles Scheme to Derive Software Project Metrics,” *Proceedings of the 2004 Workshop on Quantitative Techniques For Software Agile Process*, pp. 34-39, 2004.
[77] W. S. Humphery, *Managing the Software Process*: Addison-Wesley 1989.
[78] M. S. Krishnan, T. Mukhopadhyay, and D. Zubrow, “Software Process Models and Project Performance,” *Information Systems Frontiers 1:3, 267-277*, 1999.
[79] X. Lu, and Y. Ge, “Risk Analysis in Project of Software Development,” *Engineering and Management Conference (IEMC'03),* pp. 72-75, 2003.
[80] X. N. Lu, and Q. G. Ma, “Risk Analysis in Software Development Project with Owners and Contractors,” 2004.
[81] M. H. N. Nasir, and S. Sahibuddin, “Critical Success Factors for Software Projects: A Comaprative Study,” *Scientific Research and Essays,* vol. 6, no. 10, pp. 2174-2186, 2011.
[82] J. S. Reel, “Critical Success Factors in Software Projects,” *IEEE Software,* vol. 16, no. 3, pp. 18-23, 1999.
[83] M. Keil, H. J. Smith, and S. Pawlowski, “Why Didn’t Somebody Tell Me?: Climate, Information Asymmetry, and Bad News About Troubled Projects,” *ACM SIGMIS Database,* vol. 35, no. 2, pp. 65-84, 2004.
[84] C. S. Wu, W. C. Chang, and I. K. Sethi, “A Metric-Based Multi-Agent System for Software Project Management,” *2009 Eigth IEEE/ACIS International Conference on Computer and Information Science*, pp. 3-8, 2009.
[85] C. A. L. Reis, R. Q. Reis, H. Schlebbe *et al.*, “A Policy-Based Resource Instantiation Mechanism to Automate Software Process Management,” *SEKE '02*, 2002.
[86] A. Bahrami, “Integrated Process Management: From Planning to Work Execution,” *BSN '05 Proceedings of the IEEE EEE05 international workshop on Business services networks*, 2005.
[87] C. M. Beise, “IT Project Management and Virtual Teams,” *SIGMIS’04,* 2004.
[88] J. Westland, “The Project Management Life Cycle A Complete Step-by-Step Methodology for Initiating, Planning, Executing & Closing a Project Successfully ” *Kogan Page Limited*, 2006.
[89] J. Westland, “The Project Management Lifecycle,” *The Project Management Lifecycle A Complete Step by Step Methodology For Initiating, Planning, Executing & Closing a Project Successfully*, 2007.

[90] N. C. Debnath, R. Uzal, G. Montejano *et al.*, "Software Projects Leadership: Elements to Redefine "risk management" Scope and Meaning," *2006 IEEE International Conference on Electro/Information Technology*, pp. 280-284, 2006.

[91] M. Visconti, and C. R. Cook, "A Meta-Model Framework for Software Process Modeling," 2009.

[92] J. M. Fernandes, and F. J. Duarte, "A Reference Framework For Process-Oriented Software Development Organizations," *Softw Syst Model (2005) 4: 94–105*, 2005.

[93] D. N. Card, "Research Directions in Software Process Improvement," *Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04)*, 2004.

[94] P. Luckowicz, E. A. Heinz, L. Prechelt *et al.*, "Experimental Evaluation in Computer Science: A Quantitative Study," *Journal of Systems and Software,* vol. 28, no. 1, pp. 9-18, 1995.

[95] W. F. Tichy, "Should Computer Scientist Experiment More? 16 Excuses to Avoid Experimentation," *IEEE Computer,* vol. 31, no. 5, 1997.

[96] F. XIA, "What's Wrong with Software Engineering Research Methodology," *ACM SIGSOFT Software Engineering Notes,* vol. Vol 23 no 1, pp. 62 - 64, 1998.

[97] M. V. Zelkowitz, and D. Wallace, "Experimental Validation in Software Engineering," *Information and Software Technology,* vol. 39, no. 11, pp. 735-744, 1997.

[98] M. V. Zelkowitz, and D. Wallace, "Experimental Models for Validating Technology," *IEEE Computer,* vol. 31, no. 5, pp. 23-31, 1998.

[99] M. Shaw, "What Makes Good Research in Software Engineering?," *International Journal of Software Tools for Technoogy Transfer,* vol. 4, no. No. 1, pp. 1-7, 2002.

[100] E. Marcos, "Software Engineering Research versus Software Development," *ACM SIGSOFT Software Engineering Notes,* vol. 30, no. 4, 2005.

[101] B. Hansen, J. Rose, and G. Tjornehoj, "Prescription, Description, Reflection: the Shape of the Software Process Improvement " *International Journal of Information Management,* vol. 24, no. 6, pp. 457-472, 2004.

[102] J. D. Herbsleb, and D. R. Goldenson, "A Systematic Survey of CMM Experience and Results," *Proceedings of the 18th International Conference on Software Engineering,* pp. 323-330, 1996.

[103] D. Stelzer, and W. Mellis, "Success Factors of Organizational Change in Software Process Improvement " *Software Process - Improvement and Practice,* vol. 4, no. 4, pp. 227-250, 1999.

[104] A. Rainer, and T. Hall, "Key Success Factors For Implementing Software Process Improvement: A Maturity-Based Analysis," *Journal of Systems and Software,* vol. 62, no. 2, pp. 71-84, 2002.

[105] P. L. Bannerman, "Capturing Business Benefits from Process Improvement: Four Fallacies and What to Do About Them," *BIPI'08,* 2008.

[106] P. Abrahamsson, "Commitment Development in Software Process Improvement: Critical Misconceptions," *23rd International Conference on Software Engineering, ICSE 2001*, pp. 71-80, 2001.

[107] D. J. Paulish, and A. D. Carleton, "Case Studies of Software Process Improvement Measurement," *IEEE Computer,* pp. 50-57, 1994.

[108] B. Morgan, and G. Lowry, “Software Process Assessment and Improvement , Discussion Summary,” *IEEE Explore*, pp. 497-499, 1996.
[109] G. Succi, L. Benedicenti, P. Predonzani *et al.*, “Standardizing the Reuse of Software Processes,” *Standard View - Supporting Article*, vol. Vol 5, No.2 pp. 74-82, 1997.
[110] P. M. S. Bueno, A. N. Crespo, and M. Jino, “Analysis of an Artifact Oriented Test Process Model and of Testing Aspects of CMMI,” *LNCS 4034, pp. 263 – 277, 2006.*, 2006.
[111] Y. Xiaoguang, W. Xiaogang, L. Linpin *et al.*, “Research on Organizational-level Software Process Improvement Model and Its Implementation,” *2008 International Symposium on Computer Science and Computational Technology*, pp. 285-289, 2008.
[112] F. Keenan, “Agile Process Tailoring and probLem analYsis (APTLY),” *Proceedings of the 26th International Conference on Software Engineering (ICSE’04)*, 2004.
[113] A. M. Magdaleno, “An Optimization-based Approach to Software Development Process Tailoring,” *2nd International Conference on Search Based Software Engineering*, pp. 40-43, 2010.
[114] A. M. Magdaleno, “Balancing Collaboration and Discipline in Software Development Processes,” *32nd ACM/IEEE International Conference on Software Engineering*, vol. 2, pp. 331-332, 2010.
[115] V. R. Basili, and H. D. Rombach, “Tailoring The Software Process To Project Goals and Environments,” *Proceedings of the 9th international conference on Software Engineering*, pp. 345-357, 1987.
[116] D. Welzel, H. L. Hausen, and W. Schmidt, “Tailoring and Conformance Testing of Software Processes: The ProcePT Approach,” in Proceedings of the 2nd IEEE Software Engineering Standards Symposium, 1995, pp. 41-49.
[117] J. Bowers, J. May, E. Melander *et al.*, "Tailoring XP for Large System Mission Critical Software Development," *Extreme Programming and Agile Methods — XP/Agile Universe 2002*, Lecture Notes in Computer Science, pp. 269-301, 2002.
[118] P. Xu, and B. Ramesh, “A Tool For the Capture and Use of Process Knowledge in Process Tailoring,” *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS’03)*, 2003.
[119] P. Xu, and B. Ramesh, “Knowledge Support in Software Process Tailoring,” *Proceedings of the 38th Hawaii International Conference on System Sciences*, 2005.
[120] S. Henninger, “Turning Development Standards Into Repositories of Experiences,” *Software Process Improvement and Practice*, vol. 6, no. 3, pp. 141-155, 2001.
[121] S. Henninger, “An Environment for Reusing Software Process,” *Proceedings. Fifth International Conference on Software Reuse*, pp. 103-112, 1998.
[122] D. Kang, I.-G. Song, S. Park *et al.*, “A Case Retrieval Method for Knowledge-Based Software Process Tailoring Using Structural Similarity,” *15th Asia-Pacific Software Engineering Conference*, pp. 51-58, 2008.
[123] M. Mirakhorli, A. K. Rad, F. S. Aliee *et al.*, “RDP Technique: a Practice to Customize XP,” *APSO'08, Leipzig, Germany*, 2008.
[124] J. Grenning, “Launching Extreme Programming at a Process Intensive Company,” *IEEE Software*, vol. 18, no. 6, pp. 27-33, 2001.

[125] O. Murru, R. Deias, and G. Mugheddu, "Assessing XP at a Europian Internet Company," *IEEE SOftware*, vol. 20, no. 3, pp. 37-43, 2003.
[126] J. Drobka, D. Noftz, and R. Raghu, "Piloting XP on Four Mission-Critical Projects," *IEEE Softw.*, vol. 21, no. 6, pp. 70-75, 2004.
[127] D. W. Bustard, and F. Keenan, "Strategies for Systems Analysis: Groundwork for Process Tailoring," *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, pp. 357-362, 2005.
[128] S. Park, H. Na, S. Park *et al.*, "A Semi-Automated Filtering Technique for Software Process Tailoring Using Neural Network," *Expert Systems with Applications*, vol. 30, no. 2, pp. 179-189, 2006.
[129] V. E. Sarmiento, and M. L. Vasquez, "A Model For Measuring Agility in Small and Medium Software Development Enterprises," *Informatica (CLEI) 2012*, vol. 38, pp. 1-10, 2012.
[130] R. Sriram, and S. K. Mathew, "Global Software Development using Agile Methodologies: A Review of Literature " *Proceedings of the 2012 IEEE ICMIT*, pp. 389-393, 2012.
[131] T. Hildenbrand, M. Geisser, T. Kude *et al.*, "Agile Methodologies For Distributed Collaborative Development of Enterprise Applications," *International Conference on Complex, Intelligent and Software Inntensive Systems*, 2008.
[132] M. K. Mattson, G. A. Lewis, D. Siracusa *et al.*, "Long-Term Lifecycle Impact of Agile Methodologies," *22nd Innternational Conference on Software Maintenance*, pp. 422-425, 2006.
[133] S. Jantunen, "Exploring Software Engineering Practices in Small and Medium-sized Organizations " *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*, pp. 96-101, 2010.
[134] D. Koznov, "Process Model of DSM Solution Development and Evolution For Small and Medium-Sized Software Companies," *15th IEEE International Enterprise Distributed Object Computing Conference Workshops* pp. 85-92, 2011.
[135] T. K. Varkoi, "Process Improvement Priorities in Small Software Companies," *Portland International Conference on Management of Engineering and Technology and Innovation Management,* vol. 1, 1999.
[136] J. Soini, H. Keto, and T. Makinen, "An Approach to Monitoring the Success Factors in Software Business in Small and Medium Size Software Companies," *Proceedings of Technology Management For the Global Future PICMET'06*, vol. 6, pp. 2801-2808, 2006.
[137] G. Santos, M. Montoni, J. Vasconellos *et al.*, "Implementing Software Process Improvement Initiatives in Small and Medium-Size Enterprises in Brazil," *6th International Conference on the Quality of Information and Commmunications Technology*, pp. 187-198, 2007.
[138] I. Saastamoinen, and M. Tukiainen, "Software Process Improvement in Small and Medium Sized Software Enterprises in Eastern Finland: A State-of-the Practice Study," *Software Process Improvement Lecture Notes in Computer Science* vol. 3281, pp. 69-78, 2004.
[139] F. J. Pino, F. Garcia, and M. Piattini, "Software Process Improvement in Small and Medium Software Enterprises: A Systematic Review," *Software Quality Journal* vol. 16, pp. 237-261, 2008.

[140] S. H. Park, and D. H. Bae, "Tailoring A Large-Sized Software Process Using Process Slicing and Case-based Reasoning Technique," *IET Software,* vol. 7, no. 1, pp. 47-55, 2013.
[141] J. Li, and M. Mao, "A Case Study on Tailoring Software Process For Characteristics Based on RUP," *International Conference on Computational Intelligence and Software Engineering*, pp. 1-5, 2009.
[142] H. Westerheim, and G. K. Hanssen, "The Introduction and Use of a Tailored Unified Process - A Case Study " *Proceedings of 31st EUROMICRO Conference on Software Engineering and Advanced Applications* 2005.
[143] C. G. V. Wangenheim, S. Weber, J. C. R. Hauck *et al.*, "Experiences on Establishing Software Processes in Small Companies," *Information and Software Technology,* vol. 48, no. 9, pp. 890-900, 2006.
[144] M. A. Bunge, "Scientific Research," *Berlin, Newyork, Springer-Verlag* vol. 2 vols, 1967.
[145] M. D. Myers, "Qualitative Research in Information Systems," *MIS Quarterly,* vol. 21, no. 241-242, 1997.
[146] W. F. Tichy, "Should Computer Scientists Experiment More? 16 Reasons to Avoid Experimentation," *IEEE Computer,* vol. 31, no. 5, pp. 1-12, 1998.
[147] J. Mingers, "Combining IS Research Methods: Towards a Pluralist Methodology," *Information Systems Research* vol. 12, no. 3, pp. 240-259, 2001.
[148] J. W. Cresswell, "Research Design: Qualitative, Quantitative and Mixed Methods Approaches," *3rd Ed. Sage Publications,* 2009.
[149] R. D. Galliers, "Choosing Appropriate Information Systems Research Approaches: A Revised Taxonomy," *Information Systems Research: Contemporary Approaches and Emergent Trends,* pp. 327-345, 1991.
[150] C. Robson, "Real World Research," *Blackwell Publishing,* vol. 2nd Ed., 2002.
[151] C. Wohlin, P. Runeson, M. Host *et al.*, "Experimentation in Software Engineering: An Introduction," *Kluwer Academic Publishers,* 2000.
[152] Y. Dittrich, K. Ronkko, J. Eriksson *et al.*, "Cooperative Method Development," *Empirical Software Engineering,* vol. 13, no. 3, pp. 231-260, 2008.
[153] J. H. Iversen, L. Mathiassen, and P. A. Nielsen, "Managing Risk in Software Process Improvement: An Action Research Approach," *MIS Quarterly,* vol. 28, no. 3, pp. 395-433, 2004.
[154] D. Avison, R. Baskerville, and M. Myers, "Controlling Action Research Projects," *Information Technology and People,* vol. 14, no. 1, pp. 28-45, 2001.
[155] R. L. Baskerville, and T. W. Harper, "A Critical Perspective on Action Research as a Method for Information Systems Research," *Journal of Information Technology,* vol. 11, pp. 235-246, 1996.
[156] P. Runeson, and M. Host, "Guideline for Conducting and Reporting Case Study Research in Software Engineering," *Empirical Software Engineering,* vol. 14, no. (2009):14, pp. 131-164, 2009.
[157] S. Easterbrook, J. Singer, M.-A. Storey *et al.*, "Selecting Empirical Methods for Software Engineering Research," *Guide to Advanced Empirical Software Engineering (2008) Springer,* pp. 285-311, 2008.
[158] D. E. Gray, *Doing Research in the Real World* 2nd ed.: SAGE Publishers, 2009.

[159] U. Flick, "An Introduction to Qualitative Research," *Sage Publications Ltd. London,* vol. 4th Ed., 2009.
[160] C. B. Seaman, "Qualitative Methods in Empirical Studies of Software Engineering," *IEEE Transactions on Software Engineering,* vol. 25, no. 4, pp. 557-572, 1999.
[161] M. Meuser, U. Nagel, and M. Pfadenhauer, *Michael Meuser, Ulrike Nagel, Michaela Pfadenhauer,* Palgrave Macmillan, pp. 281, 2009.
[162] B. Gillham, *Research Interviewing: The Range of Techniques,* Open University Press McGraw Hill Education, pp. 200, 2005.
[163] F. Shull, V. Basili, J. Carver *et al.*, "Replicating Software Engineering Experiments: Addressing the Tacit Knowledge Problem," *Proceedings of the 2002 International Symposium on Empirical Software Engineering*, pp. 7-16, 2002.
[164] R. K. Yin, "Case Study Research: Design and Methods," *Sage Publications Ltd. London*, vol. 5, Applied Social Research Method Series, 2003.
[165] A. Tashakkori, and C. Teddlie, *Mixed Methodology: Combining Qualitative and Quantitative Approaches*: Sage Publications Inc. , 1998.
[166] C. Andersson, and P. Runeson, "A Spiral Process Model for Case Studies on Software Quality Monitoring - Method and Metrics," *Software Process: Improvement and Practice*, vol. 12, no. 2, pp. 125-140, 2007.
[167] I. Benbasat, D. K. Goldstein, and M. Mead, "The Case Research Strategy in Studies of Information Systems," *MIS Quarterly,* vol. 11, no. 3, pp. 369-386, 1987.
[168] D. I. K. Sjoberg, T. Dyba, B. C. D. Anda *et al.*, "Building Theories in Software Engineering," *Guide To Advanced Empirical Software Engineering*, pp. 312-336, Section III, Chp. 12, 2008.
[169] J. F. Gilgun, and K. Daly, "Definitions, Methodologies, and Methods in Qualitative Family Research," *Qualitative Methods in Family Research, Thousan Oaks, Sage Publications Inc.*, pp. 22-39, 1992.
[170] T. C. Lethbridge, S. E. Sim, and J. Singer, "Studying Software Engineers: Data Collection Techniques for Software Field Studies," *Empirical Software Engineering,* vol. 10, no. 3, pp. 311-341, 2005.
[171] H. Robinson, J. Segal, and H. Sharp, "Ethnographically-Informed Empirical Studies of Software Practice," *Information and Software Technology,* vol. 49, no. 6, pp. 540-551, 2007.
[172] B. G. Glaser, and A. L. Strauss, "The Discovery of Grounded Theory: Strategies for Qualitative Research," *Aldine Publishing*, 1967.
[173] J. Lackey, and L. Christy, "Evaluating Client Satisfaction: A Quantitative Approach," *30th Annual ACM SIGUCCS Conference on User Services,* pp. 191-194, 2002.
[174] K. M. Eisenhardt, "Building Theories from Case Study Research," *The Academy of Management Review*, vol. 14, no. 4, pp. 532-550, 1989.
[175] K. Charmaz, "Constructing Grounded Theory: A Practical Guide through Qualitative Analysis," *Sage Publications Ltd.*, 2006.
[176] J. M. Corbin, and A. L. Strauss, "Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory," *3rd Ed. Sage Publications, Inc.*, 2008.

[177] A. Jedlitschka, and D. Pfahl, “Reporting Guidelines for Controlled Experiments in Software Engineering,” *International Symposium on Empirical Software Engineering*, pp. 95-104, 2005.
[178] B. Kitchenham, H. Al-Khilidar, M. A. Babar *et al.*, “Evaluating Guidelines for Reporting Empirical Software Engineering Studies,” *Empirical Software Engineering*, vol. 13, no. 1, pp. 97-121, 2008.
[179] A. Tiwana, and M. Keil, “The One-Minute Risk Assessment Tool,” *Communications of the ACM*, vol. 47, no. 11, pp. 73-77, 2004.
[180] R. W. Woolridge, D. J. McManus, and J. E. Hale, “Stakeholder Risk Assessment: An Outcome-Based Approach,” *IEEE Software*, vol. 24, no. 2, pp. 36-45, 2007.
[181] B. Fitzgerald, N. L. Russo, and T. O'Kane, “Software Development Method Tailoring at Motorola,” *Communications of the ACM*, vol. 46, no. 4, pp. 64-70, 2003.
[182] N. Vivek, “On Tailoring an Organizational Standard Software Development Process for Specific Projects,” *Proceedings of 11th International Conference on Software Quality*, vol. 11, pp. 1-13, 2001.
[183] B. Fitzgerald, N. Russo, and T. O. Kane, “An Empirical Study of System Development Method Tailoring,” *Proceedings of 8th European Conference on Information Systems*, pp. 187-194, 2000.
[184] R. N. Charette, “Why Software Fail,” *IEEE Spectrum*, vol. 42, no. 9, pp. 42-49, 2005.
[185] C. Clegg, C. Axtell, L. Damodaran *et al.*, “Information Technology: A Study of Performance and The Role of Human and Organizational Factors,” *Ergonomics*, vol. 40, no. 9, pp. 851-871, 1997.
[186] J. Glaser, “Management' Role in IT Project Failures,” *Healthc Financ Manage*, vol. 58, no. 10, pp. 90-92, 2004.
[187] K. Ewusi-Mensah, and Z. H. Przasnyski, “Factors Contributing to The Abandonment of Information Systems Development Projects,” *Journal of Information Technology*, vol. 9, pp. 185-201, 1994.
[188] J. Jiang, and G. Klein, “Software Development Risks to Project Effectiveness,” *Journal of Systems and Software*, vol. 52, no. 1, pp. 3-10, 2000.
[189] C. Standing, A. Guilfoyle, C. Lin *et al.*, “The Attribution of Success and Failure in IT Projects,” *Industrial Management & Data Systems*, vol. 106, no. 8, pp. 1148-1165, 2006.
[190] M. Hammer, *Beyond Reengineering: How the Process-Centered Organization Will Change Our Work and Our Lives*: HarperCollins Publishers, 1996.
[191] B. Russell, and S. Chatterjee, “Relationship Quality: The Undervalued Dimension of Software Quality,” *Communications of the ACM*, vol. 46, no. 8, pp. 85-89, 2003.
[192] J. Verner, J. Sampson, and N. Cerpa, “What Factors Lead to Software Project Failure?,” *Second International Conference on Research Challenges in Information Sciences, 2008, RCIS, IEEE Explore*, pp. 71-80, 2008.
[193] G. S. Kearns, and R. Sabherwal, “Antecedents and Consequences of Information Systems Planning Integration,” *IEEE Transactions On Engineering Management*, vol. 54, no. 4, pp. 629 - 643, 2007.
[194] W. J. Doll, “Avenues for Top Management Involvement in Successful MIS Development,” *MIS Quarterly*, vol. 9, no. 1, pp. 17-35, 1985.

[195] R. D. Galliers, “Information Systems Planning in the United Kingdom and Australia - A Comparison of Current Practices,” *Oxford Surveys in Information Technology, Oxford University Press, Inc.*, pp. 223-255, 1987.

[196] J. Hann, and R. Weber, “Information Systems Planning: A Model and Empirical Tests,” *Management Science,* vol. 42, no. 7, pp. 1043-1064, 1996.

[197] G. Premkumar, and W. R. King, “Organizational Characteristics and Information Systems Planning: An Empirical Study,” *Information Systems Research,* vol. 5, no. 2, pp. 75-109, 1994.

[198] R. Sabherwal, “The Relationship between Information System Planning Sophistication and Information System Success: An Empirical Assessment,” *Decision Sciences,* vol. 30, no. 1, pp. 137-167, 1999.

[199] R. Sabherwal, and W. R. King, “An Empirical Taxonomy of the Decision-Making Process Concerning Strategic Applications of Information Systems,” *Jounral of Management Information Systems,* vol. 11, no. 4, pp. 177-214, 1995.

[200] J. Favaro, “Guest Editor's Introduction: Renewing the Software Project Management Lifecycle,” *IEEE Software* vol. 27, no. 1, pp. 17-19, 2010.

[201] H. Sertic, R. Filjar, and Z. Pozgaj, “Efficient Software Development Organisation Based on Unified Process,” *46th International Symposium Electronics in Marine, ELMAR'04,* 2004.

[202] K. Bandyopadhyay, P. P. Mykytyn, and K. Mykytyn, “A Framework for Integrated Risk Management in Information Technology,” *Management Decision,* vol. 37, no. 5, pp. 437-445, 1999.

[203] D. Klappholz, L. Bernstein, and D. Port, “Assessing Attitude Towards, Knowledge of, and Ability to Apply, Software Development Process,” *Proceedings of the 16th Conference on Software Engineering Education and Training (CSEET’03)*, 2003.

[204] D. C. Smith, M. Becker, J. B. Howell *et al.*, “Creating High Performance IS Teams,” *South African Computer Journal*, no. 28, pp. 21-29, 2002.

[205] P. J. Guinan, J. G. Cooprider, and S. Faraj, “Enabling Software Development Team Performance During Requirements Definition: a Behavioral Versus Technical Approach,” *Info. Sys. Research,* vol. 9, no. 2, pp. 101-125, 1998.

[206] S. Biffl, and M. Halling, “Investigating the Defect Detection Effectiveness and Cost Benefit of Nominal Inspection Teams,” *IEEE Transactions on Software Engineering,* vol. 29, no. 5, pp. 385-397, 2003.

[207] N. Gorla, and Y. W. Lam, “Who Should Work with Whom? : Building Effective Software Project Teams ” *Comunications of the ACM*, vol. 47, no. 6, pp. 79-82, 2004.

[208] V. Pieterse, D. G. Kourie, and I. P. Sonnekus, “Software Engineering Team Diversity and Performance,” *Proceedings of SAICSIT 2006*, pp. 180-186, 2006.

[209] V. Ambriola, R. Conradi, and A. Fuggetta, “Assessing Process-Centered Software Engineering Environments,” *ACM Transactions on Software Engineering and Methodology,* vol. 6, no. 3, pp. 283 - 328, 1997.

[210] A. Farooq, and R. R. Dumke, “Research Directions in Verification & Validation Process Improvement,” *ACM SIGSOFT Software Engineering Notes,* vol. Volume 32 Number 4, 2007.

[211] R. D. Archibald, *Managing High-Technology Programs and Projects,* 3rd ed.: Wiley Publishers, New York, 2003.

[212] R. M. Wideman, “The Role of the Project Lifecycle (Life Span) in Project Management,” *Available: http://www.maxwideman.com/papers/index.htm*, 2004.
[213] X. Song, F. Jingchun, and L. Ming, “Research on IT Project Life Cycle,” *Proceedings of the 2009 Second International Conference on Intelligent Computation Technology and Automation*, vol. 4, pp. 244-247, 2009.
[214] L. C. Stuckenbruck, *The Implementation of Project Management: The Professional's Handbook*, pp. 254: Addison Wesley, 1981.
[215] P. Cavendish, and M. D. Martin, "Negotiating & Contracting for Project Management," *The Principles of Project Management*, p. 285, Project Management Institute, 1997.
[216] F. M. Webster, "What Project Management is All About," *The AMA Handbook of Project Management*, p. 544: AMACOM Div American Mgmt Assn, 2010.
[217] H. Kerzner, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, 8th ed., pp. 1120, Wiley, 2009.
[218] W. R. Duncan, *A Guide to the Project Management Body of Knowledge*: Project Management Institute, 1996.
[219] G. K. Kapur, “The Seven Deadly Sins of Project Management,” *Available: http://www.scribd.com/doc/50446197/Kapur-Seven-Dealy-Sins-of-PM*, 1995.
[220] P. W. G. Morris, *Key Issues in Project Management*, pp. 5, Jossey-Bass, 1998.
[221] W. Royce, *Software Project Management: A Unified Framework*, pp. 75, Addison Wesley, 1998.
[222] R. G. Cooper, S. J. Edgett, and E. J. Kleinschmidt, *Portfolio Management for New Products*, 2nd ed., pp. 382, Perseus Publisher, 2001.
[223] E. Fish, “An Improved Project Lifecycle Model,” *Available: http://www.maxwideman.com/guests/plc/intro.htm*, 2003.
[224] R. Youker, “Managing the Project Cycle For Time, Cost and Quality: Lessons From World Bank Experience,” *International Journal of Project Management,* vol. 7, no. 1, pp. 52-57, 1989.
[225] M. Kohlbacher, E. Stelzmann, and S. Maierhofer, “Do Agile Software Development Practices Increase Customer Satisfaction in Systems Engineering Projects?,” *2011 IEEE International Systems Conference (SysCon)*, pp. 168-172, 2011.
[226] A. Hamidovic, and S. Krajnovic, “An Example of a Novel Approach to Measuring Projects Success within ICT Industry,” *8th International Conference on Telecommunications*, vol. 2, pp. 677-682, 2005.
[227] P. G. Patterson, L. W. Johnson, and R. A. Spreng, “Modeling the Determinants of Customer Satisfaction for Business-to-Business Professional Services,” *Joournal of the Academy of Marketing Science* vol. 25, no. 1, pp. 4-17, 1997.
[228] R. C. McCallum, K. F. Widaman, S. Zhang *et al.*, “Sample Size in Factor Analysis,” *Psychological Methods,* vol. 4, no. 1, pp. 84-99, 1999.
[229] R. L. Gorsuch, *Factor Analysis*, 2nd ed.: Routledge Taylor & Francis Group, 1987.
[230] P. Kline, *Psychometrics and Psychology*, Illustrated ed.: Academic Press, 1979.
[231] L. Hatcher, *A Step-by-Step Approach to Using the SAS System for Factor Analysis and Structural Equation Modeling*, pp. 608, SAS publishing, 1994.

[232] R. K. Henson, and J. K. Roberts, “Use of Exploratory Factor Analysis in Published Research Common Errors and Some Comment on Improved Practice,” *Educational and Psychological Measurement,* vol. 66, no. 3, pp. 393-416, 2006.
[233] L. R. Fabrigar, D. T. Wegener, R. C. MacCallum *et al.*, “Evaluating the Use of Exploratory Factor Analysis in Psychological Research.,” *Psychological Methods,* vol. 4, no. 3, pp. 272-299, 1999.
[234] J. A. Gliem, and R. R. Gliem, “Calculating, Interpreting, and Reporting Cronbach’s Alpha Reliability Coefficient for Likert-Type Scales,” *2003 Midwest Research to Practice Conference in Adult, Continuing, and Community Education*, pp. 82-88, 2003.
[235] D. George, and P. Mallery, "SPSS for Windows Step by Step A Simple Guide and Reference (11.0 Update)," Boston: Allyn & Bacon, 2003.

PUBLICATIONS

No.	Paper Title	Journal/Conference	
1	A Review of Prominent Work on Agile Processes, Software Process Improvement and Process Tailoring Practices.	Book Chapter (Springer)	CCIS Vol. 181, pp. 571-585
2	Structured Role Based Interaction Model for Agile Based Outsourced IT Projects: Client's Composite Structure.	Journal (Scopus)	Information Tech. J. 10(5), pp. 1009-1016
3	Directions and Advancements in Global Software development: A Summarized Review of GSD and Agile Methods.	Journal (Scopus)	Res. J. Inf. Tech. 3(2), pp. 69-80
4	A Framework of Software Process Tailoring For Small and Medium Size IT Companies.	Proceedings	IEEE
5	The Client as a Real Option in Successful Software Project Management (The CPPM Model).	Proceedings	IEEE
6	An Insight into Real Software Industry Paradigms and Software Engineering Research.	Proceedings	IEEE
7	A Collaborative-Interaction Model of Software Project Development An Extension to Agile Based Methodologies.	Proceedings	IEEE
8	Limitations and Measures in Outsourcing Projects to Geographically Distributed Offshore Teams.	Proceedings	IEEE
9	Client's Perspective: Realization as a New Generation Process for Software Project Development and Management.	Proceedings	IEEE

APPENDIX A

Validity Certificate of Company

CambridgeDocs
A Global Outsourcing Company

Dr. Mohammad Fadzil Hassan (Supervisor)

Head of Department

Computer and Information Sciences

Universiti Teknologi PETRONAS

Malaysia.

Subject: Certificate for PhD research work case study.

This is to certify that Mr. Rehan Akbar (G01001) has performed case study of his PhD research work on real projects at CambridgeDocs.

The data collected and observations made during the case study are allowed to use for research work and educational purposes only.

CambridgeDocs finds it as an opportunity to take part in IT/CS research works and would extend its cooperation and collaboration for future projects as well.

Manager HR	Sr. Manager
CambridgeDocs	CambridgeDocs

80 A , Abu Baker Block,
New Garden Town
Lahore, Pakistan.
Tel: +92 42 591 3500-9
Fax: +92 42 591 3510

405 Mountain Laurel Court,
Mountain View,CA 94043
USA.

www.cambridgedocs.com

APPENDIX B

Derivation of Key Processes

1. String values have been assigned to the keyword strings in the principles of agile manifesto.

   Principle 2: "Welcome changing requirements, even late in development……"
   Requirement management

   Principle 4: "Business people and developers must work together daily throughout the project." ***Team/resources***

   Principle 5: "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."
   Team/resource management

   Principle 6: "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation."
   Communication

   Principle 11: "The best architectures, requirements, and designs emerge from self-organizing teams." ***Requirement***
   Resource management & tasks allocation

   Principle 12: "Project teams evaluate their effectiveness at regular intervals and adjust their behavior accordingly." ***Team/Resource management***

2. The derived key string values are i) Requirement management, ii) Team/Resources management, iii) Communication, iv) Requirement management and tasks allocation.

3. i, ii, and iii & iv have been adopted as the key process areas in agile based methodologies.

APPENDIX C

Structured Interview Questions

1. Experience (No. of Years):____________________
2. Company Size: (Small or Medium):_____________
3. Follow Agile processes (Y/N):_______ or Hybrid Approach_______.
4. Designation______________.

Kindly provide your opinion on the following in Yes or No. You may also comment.

5. Client is the most critical and influential factor in projects following agile methods?
6. Client is the composite term used equally for different professionals hired by the client?
7. Project manager and team lead are the main roles with which client directly interacts?
8. Do you think that Client's Perspective should be understood and maintained throughout the project lifecycle particularly in agile based development?
9. Client's Perspective is necessary to maintain during each phase and iteration of software development.
10. Resource Management, Communication, Requirement management are the key process areas of software development on which client is mostly concerned.
11. Requirement gathering & tracking as well as tasks allocation to team members are the most important sub-processes of requirement management process.
12. Similar to an air flight, a software project takeoff, runs (execute), and then lands (complete).
13. Similarly to crash landing, a software project may hang up, makes crash landing or may recover from the emergency situation.
14. Most of the small and medium sized software development companies customize (tailor) their processes.
15. Process tailoring performed on the basis of project requirements, and client's requirements and expectations gives realistic results.
16. Behavior of projects (project states) and client's perspective are the main factors in agile based projects based on which processes are tailored.
17. When project is in a particular state (takeoff, running, landing, or hang up), the process tailoring of key processes (see 6 & 7) is quite effective.

APPENDIX D

Key Process Scenarios of Client's Perspective of Case Study-I

Key Process Areas Observations

Key Area 5.2(a)	**Requirement Gathering and Tracking Document's Sharing with Client**
Client's Expectation	Frequent and regular sharing and updates, scheduled.
Existing Approach	Off and on, monthly, unscheduled
Identified Problem	Project manager was unable to understand client's expectations being in-experienced. There were no proper requirements tracking documents maintained. Sharing these documents with the client was completely unscheduled.
Solution Approach	Analysis of client's emails, verbal communication and feedback. Understanding client's perspective.
Execution	All other processes were left. The schedule of document sharing with the client was prepared and got approval from the client. It was mutually decided with the client that following pattern would be followed for document sharing with him: 1. Weekly (each weekend) sending completed requirement gathering and tracking documents to the client for feedback on previous work and approval for next plan. As decided, gradually in a month, the sharing plan was shifted on daily basis sharing. 2. Daily (by the day end) the completed documents were

	shared with the client.
Result	The plan worked well and increased the satisfaction level of the client by being updated. It was clearly expressed in his emails and phone calls by using words such as “good work”, “you people are doing very well”, “great job”.

Key Area 5.2(b)	**Status Updates**
Client’s Expectation	Regular and scheduled status updates and sharing with the client.
Existing Approach	Off and on, monthly, unscheduled.
Identified Problem	Project manager did not pay attention on status updates and client was unaware of the progress of the team on his project.
Expression of Dissatisfaction	Client sent an email directly to the CEO on this issue and expressed his dissatisfaction.
Solution Approach	Direct intervention of senior project manager and CEO to resolve the issue.
Execution	Updating status on daily basis and sending the document to the client daily by the day end. In the beginning ordinary word documents were being used to prepare status document and reports. Later on to make the process more formalize, MS Project was used. This transformation is presented in Key Area 5.2(f).
Result	Client appreciated this approach and expressed his satisfaction through an email to the project manager and CEO.

Key Area 5.2(c)	Tasks Allocation
Client's Expectation	Each team member must know his/her daily tasks.
Existing Approach	Tasks were assigned irregularly to the team members and no document was being maintained in this regard.
Identified Problem	Improper requirements/tasks management process.
Expression of Dissatisfaction	During meetings client was interested to know about the tasks on which each of the team member was working. The project manager could not present any such structured document to the client and client complained about it in his email as mentioned in 5.2(b).
Solution Approach	Structured documents using ms project, bugzilla and ms word were started being used by the project manager.
Execution	On daily basis tasks were allocated to the team members, requirement repository was updated regularly on daily basis. Also a copy of each document was sent to the senior project manager.
Result	The whole project work tends to be streamlined. Each team member was aware of its daily tasks and it helped them to complete their tasks according to the deadlines. The completion of the tasks and every team member being busy on the completion of his own tasks started establishing the trust of client on the team members.

Key Area 5.2(d)	Releasing Build Plans and Meetings
Client's Expectation	Scheduled release plans with achievable deadlines.

Existing Approach	Improper planning, late releases, delayed deadlines, no proper release plan was decided with the client, variable intervals between releases.
Identified Problem	Weak project management and planning, improper communication with the client.
Expression of Dissatisfaction	After few early months, client took serious notice of this mismanaged work and communicated his dissatisfaction over team performance to the senior project manager and CEO, same case above.
Solution Approach	1. Mutually with the client it was decided that builds would be released by each weekend. 2. Later on, after stability of the project, monthly based milestones were decided 3. Informal frequent but regular daily communication with the client was done. 4. Formal and structured meeting sessions were scheduled on the day of build release.
Execution	For more than six months the process was strictly followed. A senior project manager, a project manager and CEO personally observed the process for months.
Result	i. Smooth progress of the project ii. Stability achieved iii. Client’s confidence restored iv. Realistic and achievable plans

Key Area 5.2(e)	**Resource Allocation and Project Staffing**
Client's Expectation	Full resource utilization with his approval, neither resource sit idle, Each resource must have daily tasks to complete.
Existing Approach	Improper resource allocation which did not match as reflected in project staffing document shared with the client.
Identified Problem	Improper and unplanned tasks allocation to the resources. The actual effort that each resource was putting was different as mentioned in project staffing document.
Expression of Dissatisfaction	Client mentioned it during meeting sessions. He wanted to know exactly who is working on which task and how much effort is he putting on the project. Client had to pay for each resource, so he was much concerned about number, quality and tasks of the resources.
Solution Approach	Numbers of resources were approved by the client, resources were appointed with his consent and requirement. He was kept completely aware of on which module each resource is assigned. Also Key Area 5.2(c) was applied.
Execution	Resource and tasks allocation. Updating client about progress of each module and resource. 100% effort of each resource on his/her module.
Result	i. Client became aware of each resource, its effort and tasks. ii. Client's satisfaction in terms of his project progress. iii. Client's satisfaction in terms of his investment and payments. iv. Willingness to add new resource as required. v. Trust developed in project manager.

Key Area 5.2(f)	Frequent Meetings and Updating Client
Client's Expectation	Regular meetings. Communicating him each and every minor major issues. He wants to be updated on everything.
Existing Approach	Team lead and project manager both were not replying to client's emails well in time. No status of the whole project was being sent to him. Key Areas 5.2(a) to 5.2(e) applies here.
Identified Problem	Lack of interest and not realizing its importance by project manager and team lead. As in all previous cases project was not managed so project manager and team lead were reluctant to have meeting and discuss issues and progress with the client regularly.
Expression of Dissatisfaction	Client did not show his concern on this matter directly. Instead he showed his concerns as in Key Areas 5.2(a) to 5.2(e).
Solution Approach	Regular and frequent meetings.
Execution	i. Scheduled & unscheduled ii. Structured & unstructured iii. Phone calls, E-mails, Chatting
Result	i. Awareness of client with all the details of the project. ii. His level of satisfaction increased. iii. Understandability and good working relationships developed. iv. Cooperation and coordination, open discussions without fear. v. Gap between team and client reduced. vi. Friendly environment developed.

Key Area 5.2(g)	Project Management Tools
Client's Expectation	Client had no concern with the tools but was interested in managed work in either way.
Existing Approach	MS Word documents were being maintained.
Identified Problem	i. Poor requirement management and tracking. ii. Missing requirements. iii. Resources underutilized. iv. Wrong project tracking.
Expression of Dissatisfaction	See Key Areas 5.2(a) to 5.2(f).
Solution Approach	Project management tools adopted. Automated processes adopted.
Execution	i. GoogleDocs using MS Excel ii. MS Project iii. MS Groove iv. Sharepoint portal server v. SVN
Result	i. Structured requirement management and tracking. ii. Proper tasks allocation. iii. Proper bug management and tracking. iv. Resource utilization and effort estimations. v. Easy project progress and tracking.

APPENDIX E

Project Behavior and States Factors of Case Study-I

Project States Behavior Observations

Problem classes

Factor 5.3(a)	**Problem Class Existence**
Project State:	Takeoff
Project Sub-states	Pre-Takeoff, Takeoff
Problem Class Identified	Performance-Minor
Other Problem Class (if exists)	Management-Critical at the initial level
Reasons of Problems	i. Carelessness of team members. ii. Weak coordination. iii. Poor requirement tracking and tasks allocation processes.
Project's/ State's Response	Soon after the project started, the mild problem started arising. Project anyhow kept continued. No problem created severe hurdle in the work.
Solution Approach	Project manager took small measures in a few cases and left some others unhandled.
Outcome	The problems started to accumulate. Mismanagement became more prominent but was still unnoticed.

Factor 5.3(a1)	**Problem Class Existence**
Project State:	Running
Project Sub-states	Running, Post-running
Problem Class Identified	Management-Critical Performance-Minor
Other Problem Class (if exists)	Performance-Minor
Reasons of Problems	i. Weak processes of requirement gathering & tracking ii. Tasks allocation iii. Client communication & coordination. iv. Bugs and late releases.
Project's/ State's Response	i. The accumulated problems became more severe. ii. Whole project work got mismanaged. iii. Deadlines sometimes became late. iv. Client noticed it and emphasized on managed work during meeting sessions and emails time to time.
Solution Approach	Project Manager tried to handle the unmanaged work and processes but the efforts were not right directed.
Outcome	The severity of the problems became more intense and affected the project. As a result client became unsatisfied with the overall performance of the team and progress of his project. The severity increased in the later stage of running sub-state.

Factor 5.3(a2)	**Problem Class Existence**
Project State	Hang up
Project Sub-states	Crawling, Swing, Pre-running
Problem Class Identified	Progress-Limiter
Other Problem Class (if exists)	Management-Critical
Reasons of Problems	i. Weak processes of requirement gathering & tracking. ii. Tasks allocation. iii. Client communication & coordination. iv. Bugs and late releases.
Project's/ State's Response	i. Whole project work got mismanaged. ii. Deadlines many times became late. iii. Client expressed his level of dissatisfaction many times. Ultimately he conveyed it to the CEO and senior project manager.
Solution Approach	CEO personally jumped in to redesign the processes of requirement gathering & tracking, tasks allocation and status updates. Senior project manager along with an experienced project manager planned and rescheduled the whole project. They used project management tools to manage the whole work.
Outcome	After the severity of the problems, the project started crawling to revamp the processes and handle the problems. The intensity of problems reported by the client led it to the swing state. It was most likely that if the same problems and conditions prevailed for a little more time, the client may think to terminate the project and handover it to some other company. Treated it as

	very critical, the measures taken by CEO, senior project manager and another project manager brought project back to pre-running state and project started its smooth progress with the confidence building up of the client.

Factor 5.3(a2)	**Problem Class Existence**
Project State	Landing
Project Sub-states	Landing, Post landing
Problem Class Identified	Performance-Minor Management-Critical
Other Problem Class (if exists)	Not observed.
Reasons of Problems	i. Lack of proper process for minimum documentation. ii. Undefined procedure of code base handover and necessary libraries. The project team being busy in completing the remaining work, bug fixing could not paid attention to these winding up tasks. Project manager himself did not spend his effort on it.
Project's/ State's Response	Such problems were not of severe nature so did not directly affect the project. But such problems of mild intensity though were present till the end.
Solution Approach	Scheduling by project manager to complete the last tasks and the required minimum documentation.
Outcome	i. Project completed and closed successfully. ii. The client's trust that had been developed helped the team in starting another project with the same client that was actually the second part of the same project.

Client's Perspective Factor

Factor 5.3(b)	Client's Perspective
Project State:	Takeoff
Project Sub-states	Pre-Takeoff, Takeoff
Observation	Poor requirement tracking and tasks allocation processes. Factor 5.3(a) (iii) of project states.
Others (if exist)	-
Cause	i. Weak project management. ii. Not taking care of client's interest.
Effect	In early stages the effect was not that much negative. It caused to accumulate problems for later stages.
Solution Approach	Micro management practices approach.
Outcome	Not all but few were handled.

Factor 5.3(b1)	Client's Perspective
Project State	Running
Project Sub-states	Running, Post-running
Observation	Factors 5.3(a1) (i), (ii), (iii) of project states. Management-Critical class problems.
Others (if exist)	-
Cause	i. Weak project management. ii. Inexperienced project manager. iii. Hiding the facts from senior managers.
Effect	i. Mismanaged project. ii. Client unsatisfied. iii. Weak plan and unrealistic schedule.
Solution Approach	i. At early running sub-state no proper solution was adopted.

	ii. In post-running few measures were taken by senior project manager.
Outcome	i. A proper process to manage the project and meet the client’s requirements was still lacking. ii. A light weight process was started to be defined but till then it was quite late.

Factor 5.3(b2)	**Client’s Perspective**
Project State:	Hang up
Project Sub-states	Crawling, Swing, Pre-running
Observation	Factors 5.3(a2) (i), (ii), (iii), (iv) of project states. Progress-Limiter class problems.
Others (if exist)	-
Cause	Poor project management. Hiding facts from senior managers. Difference between actual and communicated.
Effect	i. Client expressed his level of complete dissatisfaction. ii. Project progress got disturbed. iii. Slow progress. iv. Unable to evaluate the team performance.
Solution Approach	Well defined light weight approaches to manage: i. Requirement gathering & tracking process. ii. Tasks allocation. iii. Resource management. iv. Communication & coordination.
Outcome	i. Realistic plan and schedule. ii. Achievable milestones. iii. Client’s satisfaction and trust.

Factor 5.3(b3)	Client's Perspective
Project State	Landing
Project Sub-states	Landing, Post-landing
Observation	i. Documents and manuals prepared as client required. ii. Minor bug fixing.
Others (if exist)	Performance-minor
Cause	No major problems identified as missing requirement and lack of client's interest.
Effect	Progress of the project remained smooth.
Solution Approach	-
Result	i. Project completed successfully. ii. Second part's scope of the project defined and started by the client.

APPENDIX F

Key Process Scenarios of Client's Perspective of Case Study-II

Key Process Areas Observations

Key Area 6.2(a)	**Requirement Gathering and Tracking Document's Sharing with the Client.**	
Client's Expectation	Project B	i. Not demanding, weekly. ii. Structured requirement gathering and tracking document.
	Project C	i. Very demanding, daily sharing and must be regular, at least daily. ii. Detailed requirement specification document with resource allocation.
Existing Approach	Project B	i. Weekly sharing the document and updating the client. ii. Well formatted and structured document was being shared with the client.
	Project C	i. Regular sharing but on weekly basis. ii. Master requirement tracking document was being shared.
Identified Problem	Project B	No such problem or issue raised by the client. He was satisfied with the existing approach.
	Project C	i. The clients were very demanding. ii. Project manager was used to update clients irregularly in the beginning. iii. He could not have an idea of what they require from the team.
Solution Approach	Project B	The same process was followed regularly.
	Project C	i. Daily sharing of the documents with the client. ii. A comprehensive master requirement

		document was prepared and shared with the client.
Execution	Project B	i. Updated project progress document was shared with the client on every weekend. ii. The status of the requirements was clearly mentioned in the document.
	Project C	Master requirement tracking document for each build was sent to the client for his comments and approval. It always took 2-3 cycles to the final approval. Tasks lists of the team members, requirements and functionalities updates were sent to the client on daily basis as follows: 1) Daily (day end) sending updated requirement gathering and tracking documents to the client to provide them the current progress of the project. 2) Communicating through daily informal meetings. 3) The process remained intact till the project ended.
Result	Project B	i. The existing plan worked well throughout the project. ii. The satisfaction level of the client remained persistent. During the meeting sessions he always expressed his positive comments such as *"I have no problem with it", "ok", "good work"* etc.

	Project C	i. The clients were satisfied with the process, but because of being 2-3 technical persons at that end, they were used to make frequent changes in the requirements of the upcoming build. ii. Unable to freeze the requirements till the deadline. iii. The project team was not comfortable with this practice. This created a level of dissatisfaction between offshore team and the client.

Key Area 6.2(b)	**Status Updates**	
Client's Expectation	Project B	i. Regular and scheduled. ii. Structured document. iii. Status of the project, tasks and team members.
	Project C	i. Frequent and Daily. ii. Formal and Informal. iii. Well structured document with detailed status.
Existing Approach	Project B	i. Regular on daily basis. ii. Structured document with specific format.
	Project C	i. Regular but not daily. ii. Requirements were not properly interpreted.
Identified Problem	Project B	i. The client wanted scheduled status but not necessarily on daily basis. ii. No specific issues or problem ever rose.

	Project C	i. Project Manager was used to update the client on irregular basis. ii. Clients wanted to meet and being updated on daily basis. iii. There was realized kind of lack of trust from the client side. iv. The clients were also very concerned about the progress of the project and team performance.
Expression of Dissatisfaction	Project B	i. The client never objected on the existing process. ii. He just wanted to let him know before starting anything new or implementing server side changes.
	Project C	i. Directly clients hardly mentioned it. ii. Though from their conversations it was realized that they wanted more frequent interaction and iii. Detailed status of the project, tasks, developers and major as well as minor issues.
Solution Approach	Project B	i. Just existing process was kept continued. ii. Weekly status updates to the client. iii. More detailed status document of tasks (for his satisfaction) using different color legends was started sharing with the client. iv. Separate statuses of tasks such as *'Done'*, '*In Progress', 'On Going'*, and '*To Be Done*' were set in the status and shared with the client.
	Project C	i. More frequent meetings on daily basis started.

		ii. Updating client frequently on emails, chatting and phone calls.
Execution	Project B	i. Regular status update document in the form of complete progress of the project was used to send to the client by every weekend. ii. If there were some major changes or modifications, then were notified to the client the same day. iii. Color legends and tasks status separately and clearly mentioned in the document.
	Project C	i. Due to the geographical time difference, the team members changed their office timings in order to sit and waiting for the client's calls and emails till late at night. ii. The client never wanted some formal document in this regard. Therefore, informal and unstructured meeting session through live calls, emails and messengers fulfilled the requirements of the client. iii. Formal status update was being sent on weekly basis only.
Result	Project B	Due to the regularity of the process, the relationship between the client and project team got more trustworthy and stable.
	Project C	i. The frequent meeting sessions though proved helpful in keeping clients updated. ii. Also increased the level of discomfort at both ends. iii. Project apparently looked like stable but actually was not.

Key Area 6.2(c)	**Tasks Allocation**	
Client's Expectation	Project B	i. Proper tasks assignment to the team members. ii. Tasks status and tracking. iii. Neither resource should sit idle.
	Project C	i. Each team member must have tasks. ii. Proper tracking of completed and incomplete tasks.
Existing Approach	Project B	i. Internal tasks lists were prepared and updated on daily basis. ii. Team members were being assigned tasks once or twice in a week. iii. Weekly sharing tasks status with the client.
	Project C	i. Master requirement tracking document with owners of the tasks was prepared. ii. According to build release plan, tasks were allocated to the team members. iii. Shared with the clients according to the release plan.
Identified Problem	Project B	Status of completed, incomplete, ongoing and backlog tasks was unclear.
	Project C	No major problem was reported in the process.
Expression of Dissatisfaction	Project B	Not really.
	Project C	Not really.
Solution Approach	Project B	i. Tasks and bugs added to the automated system of tasks management. ii. Daily extraction of tasks from the repository. iii. Assignment of the tasks.

		iv. Feedback from the team members. v. Updating tasks status in the repository.
	Project C	No modification was made in the existing process.
Execution	Project B	i. First on weekly basis. ii. Later on, daily basis, in order to make it a formal process. iii. Bugzilla an open source system was used as tasks repository. iv. Tasks were given specific number. v. Their status was set in the bugzilla. vi. Preparing MS project plan and word documents were used for tasks allocation to the team members and status updates.
	Project C	i. Daily for tasks received from daily informal meetings. ii. Weekly for formal structured requirements of the releasing build.
Result	Project B	i. The persistent and formal process. ii. The level of client's satisfaction was increased. iii. He was satisfied with the existing project progress. iv. More stability in the project. v. Client's trust increased on the team's performance.
	Project C	i. The clients were satisfied with this tasks allocation and sharing process. ii. They expressed their satisfaction on this process and insisted on following it for upcoming builds.

Key Area 6.2(d)	Releasing Build Plans and Meetings	
Client's Expectation	Project B	i. Weekly releases and meetings. ii. Well prepared project plans. iii. Details of current and upcoming builds and tasks. iv. Newly explored functionalities. v. Demonstration of released build. vi. Backlog issues. vii. Well scheduled meetings with full preparation. viii. Server side management. ix. Prioritizing the tasks and functionalities.
	Project C	i. Weekly or fortnightly releases. ii. Regular formal or informal meetings. iii. Tasks allocation. iv. Bug fixing status. v. Backlog and modifications in previous releases.
Existing Approach	Project B	i. Weekly releases and meetings. ii. Scheduled project plans. iii. Research on new functionalities. iv. Demonstrations of released builds. v. Prioritizing the functionalities. vi. Backlog issues slightly presented.
	Project C	i. Uncertain meeting timings. ii. Immature release plans. iii. No approval of master requirement document of upcoming build from the client.
Identified Problem	Project B	No major problem was reported by the client.
	Project C	i. Informal decisions about upcoming builds. ii. No requirement freezing process.

		iii. Frequent informal meetings. iv. Each client prioritizing his/her own requirements.
Expression of Dissatisfaction	Project B	No
	Project C	i. Clients did not show major concern about such minor irregularities of the processes. ii. Project team though sometimes complained about making a proper process but client was not that much interested.
Solution Approach	Project B	i. As client was satisfied with the existing process and progress of the project, therefore, no major modifications made. ii. Reprioritizing the tasks both by the team lead and client.
	Project C	i. No major changes made. ii. Structural and formatting improvements in the existing master requirement document and project plans.
Execution	Project B	i. Reprioritization of upcoming tasks made during meeting sessions. ii. Live demonstrations of builds. iii. Feedback of the client. iv. Meeting minutes were prepared and shared. v. Meeting recordings.
	Project C	i. Only master requirement document and project plan was used on daily basis throughout the project. ii. Client's feedback. iii. Extracting requirements from client's documents.

		iv. Meeting minutes.
Result	Project B	i. Very stable project. ii. Very satisfied client. iii. Good team performance.
	Project C	Despite the major concerns of the client related to the processes, following issues were observed: i. Client's dissatisfaction. ii. Increasing complaints of client related to the code, bugs and other inconsistencies.

Key Area 6.2(e)	**Resource Allocation and Project Staffing**	
Client's Expectation	Project B Project C	i. Full resource utilization. ii. No resource should sit idle. iii. Tasks allocation to the resources. iv. Approval of resources and tasks allocation from the client. v. Resources must have tasks daily. vi. Effort of each resource on project. vii. Overall team effort on the project.
Existing Approach	Project B Project C	i. Determining tasks complexity. ii. Analyzing expertise of resources. iii. Defining % effort of each resource on the project. iv. Approval of resources from the client. v. Project staffing document and vi. Monthly approval from the client.
Identified Problem	Project B	i. Under estimated tasks complexity. ii. More resources required as estimated and approved earlier.

		iii. The client was not kept aware of the situation. iv. Client was reluctant to add more resources to the team.
	Project C	No problem observed.
Expression Of Dissatisfaction	Project B	Client was not aware of the issue, therefore, no problem was risen from his side.
	Project C	No.
Solution Approach	Project B	i. Determining the actual project work load. ii. Used hidden/silent resources.
	Project C	Existing processes continued.
Execution	Project B	i. Recalculation of tasks complexities. ii. Effort distribution of resources. iii. Resource leveling. iv. Addition of a new resource. v. Addition of silent resources. vi. Got client's approval at some appropriate time. vii. Monthly project staffing documentation. viii. Resource management and ix. Resource planning sheet.
	Project C	i. Approval of resources at the beginning of the project. ii. Monthly project staffing. iii. Resources management and iv. Resource planning sheet.
Result	Project B Project C	i. Full utilization of resources on the project. ii. Less idle time. iii. Client's satisfaction on his investment of time and money. iv. Transparency of project. v. Stability and client's trust.

Key Area 6.2(f)	**Frequent Meetings and Updating Client**	
Client's Expectation	Project B	i. Regular weekly meetings. ii. Need based meetings. iii. Communicating project status, major and minor issues, and next plans.
	Project C	i. Regular weekly meetings. ii. Informal daily meetings. iii. Project status updates. iv. Communicating major and minor problems, and bug fixing issues. v. Improvements and modifications required.
Existing Approach	Project B	Same as required by the client.
	Project C	Same as required by the client.
Identified Problem	Project B	i. Not observed. ii. Not reported by the client.
	Project C	Frequent and informal daily meetings became an overhead in the project due to being unable to freeze the requirements.
Expression of Dissatisfaction	Project B	No
	Project C	i. Not really, but ii. Client showed his unhappiness on the problems in previous releases. See results of 6.2(d).
Solution Approach	Project B	Not required.
	Project C	Not required regarding the process. Complaints about the problem have been presented in 6.2(h).
Execution	Project B	Not Applicable.
	Project C	Not Applicable.

Result	Project B	i. Gap between team and client reduced. ii. Awareness of client with all the details of the project. iii. His level of satisfaction and trust increased. iv. Cooperation and coordination, open discussions without fear.
	Project C	i. Number of requirements increased day by day. ii. Problems and issues rose. iii. Prioritizing and reprioritizing tasks by each of the client became the routine. See 6.2(h) for details.

Key Area 6.2(g)	**Automated Project Management**	
Client's Expectation	Project B Project C	i. Client had no concern with the tools. ii. Well managed project. iii. Project manager need to be more keen and concerned.
Existing Approach	Project B	MS Word documents i.e. i. Tasks sheets ii. Requirement management iii. Resource sheets iv. Status updates
	Project C	MS Word documents i.e. Master Requirement document.
Identified Problem	Project B	i. Poor requirement tracking. ii. Backlog work.
	Project C	Missing requirements.

Expression of Dissatisfaction	Project B	No
	Project C	During meeting sessions.
Solution Approach	Project B	Project management tools used.
	Project C	Improvement in the existing process.
Execution	Project B	i. GoogleDocs using MS Excel ii. MS Project iii. MS Groove iv. Messengers v. SVN
	Project C	Master requirement document was made more structured and well formatted. i. MS Excel ii. Messengers
Result	Project B	i. Proper requirement gathering and tracking. ii. Proper tasks allocation. iii. Proper tracking of backlog work. iv. Easy project progress and tracking. v. Bugs management. vi. Client's satisfaction and confidence increased.
	Project C	An improvement in the project management.

APPENDIX G

Project Behavior and States Factors of Case Study-II

Project States Behavior Observations

Problem Classes

Factor 6.3(a)	**Problem Existence**	
Project State	Takeoff	
Project Sub-states	Pre-Takeoff, Takeoff	
Problem Class Identified	Project B	Performance-Minor
	Project C	
Other Problem Class (if exists)	Project B	No
	Project C	
Reasons of Problems	Project B	i. Weak problem solving. ii. Carelessness. iii. Shortcuts in coding. iv. Bad coding practices.
	Project C	i. Bad coding practices. ii. Carelessness. iii. Inconsistent data. iv. Coordination problems with the client. v. Noncooperation from the client.
Project's/ State's Response	Project B	Problems were not critical and client was cooperative, therefore, i. Project progress was not affected. ii. Project ran smoothly. iii. Team performance improvement. iv. No hurdles during the takeoff state.
	Project C	Problems were not that much critical, but client was much concerned about them. In a way they: i. Affected the release of builds. ii. Produced bugs in delivered build. iii. Frequent modifications after build released.

		iv. Project was instable. v. Hurdles started during the takeoff state.
Solution Approach	Project B	Problems were minor; therefore gradually with the maturity of the processes they were automatically settled and improved. i. Refined the process and problem solving approach.
	Project C	i. Processes defined by the project manager. ii. Frequent coordination with the client. iii. Processes modified and improved.
Result	Project B	i. Stability in project with smooth progress. ii. Client's trust and satisfaction achieved.
	Project C	i. Problems persistently occurred. ii. Client's non cooperative behavior continued. iii. Project remained instable.

Factor 6.3(a1)	**Problem Existence**	
Project State	Running	
Project Sub-states	Running, Post-running	
Problem Class Identified	Project B	Performance-Minor
	Project C	Performance-Minor Management-Critical
Other Problem Class (if exists)	Project B	Management-Critical
	Project C	Progress-Limiter
Reasons of Problems	Project B	i. Bad coding practices. ii. Hiding facts by the developers. iii. Misleading answers by the developers. iv. Lack of attention of the project manager.

	Project C	i. Project manager was unable to convince the client on certain matters. ii. Client never tried to understand the team's point of view. iii. Improper process of development and release.
Project's/ State's Response	Project B	i. Project was not greatly affected. ii. Development and release plans not got affected. iii. Client was satisfied. iv. The running state was smooth. v. During post running sub-state two major problems were faced twice.
	Project C	i. Client's confidence shakes. ii. Increased inconsistency in released builds. iii. More instability in the project. iv. Problems faced throughout the running state.
Solution Approach	Project B	i. Good project management practices. ii. More refined processes.
	Project C	i. Processes defined by the project manager. ii. Increased coordination among the team members. iii. More frequent communication with the client.
Result	Project B	i. No major issues and problems. ii. Smooth progress of the project. iii. Good team performance. iv. Client's trust and satisfaction.
	Project C	i. Lack of trust increased with the client. ii. Mistakes by the developers remained. iii. Repeated modifications in the released

		builds. iv. Client's dissatisfaction.

Factor 6.3(a2)	**Problem Existence**	
Project State	Hang up	
Project Sub-states	Crawling, Swing, Pre-running	
Problem Class Identified	Project B	Progress-Limiter
	Project C	Progress-Limiter
Other Problem Class (if exists)	Project B	Management-Critical
	Project C	Performance-Minor
Reasons of Problems	Project B	i. Issues in the open source library. ii. Team unable to fix the problems in the library file. iii. Lack of time for the developers. iv. Upcoming plans and deadlines.
	Project C	i. Unresolved problems as described in Examples 5.7.4(a) to 5.7.4(a1). ii. Problems accumulated and couldn't be resolved because of behavior of the client.
Project's/ State's Response	Project B	i. Main development tasks were stopped for the time being. ii. Only minor fixes were made as development tasks. iii. Client was agreed with the approach. iv. Project entered into hang up state two times during its life. v. Hang up state was successfully recovered by the projects.
	Project C	i. Increased instability in the project insipte of hard work by the team members. ii. Hang up state ran in parallel to the

		running state.
Solution Approach	Project B	i. Development of a new library file. ii. Dependent tasks and functionalities were stopped. iii. Independent and minor tasks were completed only.
	Project C	Project manager communicated the situation and non cooperative attitude of clients to the CEO at last to avoid any complication in future.
Result	Project B	i. Client was satisfied with the development of new library file. ii. Increased client's confidence on the project team. iii. Stability in the project. iv. Friendly and relaxed development environment.
	Project C	i. No improvement in the project's progress. ii. Unpredictable attitude of client. iii. Termination of the project was predicted by the project manager and communicated to the CEO.

Factor 6.3(a3)	**Problem Existence**	
Project State	Landing	
Project Sub-states	Landing, Post landing	
Problem Class Identified	Project B	Performance-Minor
	Project C	Performance-Minor
Other Problem Class (if exists)	Project B	No
	Project C	No

Reasons of Problems	Project B	i. Minor coding issues. ii. Bug fixing. iii. Code improvement.
	Project C	i. Coding problems. ii. Inconsistent records. iii. Decrease in the number of team members. iv. Improper testing due to only one resource's availability.
Project's/ State's Response	Project B	i. No major effect on the project. ii. During the landing state, project was successfully completed.
	Project C	i. Same as in Examples 5.7.4(a) to 5.7.4(a2). ii. No major problem during the landing state.
Solution Approach	Project B	Light weight project management approaches.
	Project C	No solution was applicable to the situation.
Result	Project B	Project was successfully completed and closed.
	Project C	Project was closed without completion.

Client's Perspective Factor

Factor 6.3(b)	**Client's Perspective**	
Project State	Takeoff	
Project Sub-states	Pre-Takeoff, Takeoff	
Observation	Project B	Project was started as expected by the client, therefore at this initial level of the project no client related issues were present.
	Project C	Since the beginning of the project, there were mistakes in the builds. Client noticed it with great

		concern. i. Client's perspective was not achieved. ii. Problems in the project.
Others (if exist)	Project B	See 'Project Response' and 'Result' of 6.3(a).
	Project C	See 'Project Response' and 'Result' of project 'C' in 6.3(a).
Cause	Project B	i. Project manager ran the project according to the requirement and expectations of the client. ii. No effect on the project's progress.
	Project C	i. Project manager was unable to convince the client on the process of build development and release. ii. See 'Reasons of Problems' of project 'C' in 6.3(a) – (iii), (iv), (v).
Effect	Project B	i. Smooth project progress. ii. Satisfaction of client. iii. No pressure on the team members, no work overloads. iv. Friendly and cooperative environment. v. Take off state had no major issues and problems.
	Project C	i. Instability in the project started. ii. Pressure on the team members developed. iii. Lack of understanding developed between client and team members was started creating. iv. Takeoff state had been effected by the minor but critical problems.
Solution Approach:	Project B	The existing practices were continued.
	Project C	i. Modifications in the existing processes. ii. Frequent communication with the client.

		iii. Project manager tried to explain the problems to the client which were due to the existing process and motivating him to modify the process as required.
Result:	Project B	Project progress was quite successful.
	Project C	i. Problems and mistakes were persistent. ii. Client not agreed with the project manager. iii. Client was unsatisfied. iv. Same problems raised in each build. v. Instable builds and releases.

Factor 6.3(b1)	**Client's Perspective**	
Project State	Running	
Project Sub-states	Running, Post-running	
Observation	Project B	i. Project manager had defined the processes as expected by the client to fulfill his requirements. ii. Client's was agreed and satisfied with the approach of the team and project management.
	Project C	i. Client's perspective was not met throughout the running state and the project. ii. Client was much concerned about the problems. iii. Project was not running as expected by the client.
Others (if exist)	Project B	See 6.3(a1)
	Project C	See 6.3(a1)
Cause	Project B	i. The approaches were followed on which the client was agreed.

		ii. All the processes were decided in coordination with the client during meetings.
	Project C	i. Lack of trust from the client side. ii. Project was unable to convince the client on certain technical issues. iii. Non cooperative attitude of the client. iv. The client and project manager were unable to understand each other's point of views. v. Both of them were unable to reach at some mutual decision to resolve the issues.
Effect	Project B	i. No major effect on the project's progress. ii. Running state had no major concerns related to the client. iii. Client's perspective was fully met during the running state.
	Project C	i. Problems accumulated. ii. Instability in the project increased. iii. Client's satisfaction level decreased. iv. Pressure built up on the whole team. v. Client's perspective was missing throughout the running state. vi. Progress hurdles during running state.
Solution Approach	Project B	See 'Solution Approach' of project 'B' of 6.3(a1).
	Project C	CEO was communicated with the issues. See 'Solution Approach' of project 'B' of 6.3(a1).

Result	Project B	i. No major problems in the project. ii. Smooth progress of the project. iii. Good team performance. iv. Client's trust built up on the team.
	Project C	i. Inspite of the approaches followed to resolve the problems but the project could not be stable. ii. Lack of trust increased. iii. Client's showed his dissatisfaction even on the minor problems. iv. Instability in the project remained as it is.

Factor 6.3(b2)	**Client's Perspective**	
Project State	Hang up	
Project Sub-states	Crawling, Swing, Pre-running	
Observation	Project B	i. Hang up state occurred twice throughout the project. ii. Client had no objection on the approach followed by the team to recover from the hang up state. iii. The progress of the project was slowed down but was continued. iv. Client's expectations were fully met.
	Project C	i. Minor but critical problems were present throughout the hang up state. ii. Hang up state proved to be the most critical state of the project. iii. Client's expectations could not be met. iv. Client was not satisfied with the team

		performance.
Others (if exist)	Project B	See 6.3(a2).
	Project C	See 6.3(a2).
Cause	Project B	i. Client's cooperative attitude. ii. Client's understandability of the situation. iii. Team had already built up its trust relationship with the client.
	Project C	i. Client's unresponsive and non cooperative attitude. ii. Project manager unable to convince the client.
Effect	Project B	Client's expectations were fully met therefore, project ran smoothly even during the hang up state.
	Project C	i. Project got more instable and client's behavior became very uncertain. ii. Uncertainty and instability increased in the project. iii. Project manager had realized the termination of the project.
Solution Approach	Project B	The existing approach was followed during the hang up state. No major changes were made in the processes.
	Project C	Project manager openly discussed the issues with the client during the meeting. Insipte of it, client was not convinced.
Result	Project B	i. Project successfully recovered from the hang up state. ii. Project development started at normal

		pace.
	Project C	i. Project could not recovered from the hang up state. ii. Client stopped sending more requirements and tasks.

Factor 6.3(b3)	Client's Perspective	
Project State	Landing	
Project Sub-states	Landing, Post-landing	
Observation	Project B	Nothing special was observed or reported.
	Project C	
Others (if exist)	Project B	See 6.3(a3)
	Project C	See 6.3(a3)
Cause	Project B	i. The project was successfully completed and being closed. ii. Client being not demanding throughout the project.
	Project C	i. Project was being closed unsuccessfully. ii. Client had no more expectations from the team.
Effect	Project B	No effect during landing state.
	Project C	
Solution Approach	Project B	Not applicable.
	Project C	Not applicable.
Result	Project B	Project was successfully completed and closed.
	Project C	Project was terminated/closed without completion.

APPENDIX H

Demographic Information

Respondents Data

Table H1. Designation and Number of Respondents

Designation	Number of Responses	Percentage
Senior/Top Management	14	13.6%
Management	12	11.7%
System Analyst	10	9.7%
Senior Software Engineer	40	38.8%
Software Engineer	27	26.2%

Respondents Experience

Table H2. Experience of Respondents

No. of Years	Frequency	Percentage
1 – 4	40	38.8%
5 – 10	54	52.4%
More than 10	7	6.8%
Less than 1	2	1.9%

Company Responses

Table H3. Responses from Company

Company Size	Frequency	Percentage
Small	39	37.9%
Medium	64	62.1%

APPENDIX I

Sample Questionnaire

General Information

No	Question	Answer
1	What is your designation?	
	Senior/Top Management Position	
	Management Position	
	System Analyst	
	Senior Software Engineer	
	Software Engineer	
2	Which is your country?	
3	How many years of experience do you have?	
4	What is the size of the company where you work?	
	Small	
	Medium	
5	Does your company follow agile processes?	
	Yes	
	No	

No	**<u>Client's Perspective:</u>** **Means understanding the client. Understanding his/her requirements, attitude, behavior. Taking care of and giving preference to client's satisfaction. It is more likely mind reading of client. Judging his thinking.**	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
6	Client is one of the most important and influential factor in the software development projects especially in agile based development.					
7	The role of client is critical and decisive in software projects.					
8	The satisfaction of client is important and necessary for project success.					
9	Project success depends mainly on client's satisfaction which is dependent on team's performance.					
10	Most of the projects are closed without completion because of dissatisfaction of client.					
11	Understanding and following client's perspective throughout the project makes sure the client's satisfaction.					
12	It is not necessary to understand the client's perspective for the success of a project.					
13	Client is not interested in processes very much but wants his/her work completion in either way.					
14	It is not necessary to stick to the processes if client is not that much concerned with the processes and needs working codes in due time.					
15	Processes are not that much important to make a client satisfied.					
16	The preference is given to the client's priorities and expectations in projects					

	following agile based methodologies.					
17	Some problems are associated with the poor or no understanding of the client’s perspective.					
18	Requirement management, communication & coordination, and Resource (team) management are the key phases in which major process tailoring is required and done.					
19	Client in agile based environment is mostly concerned with these three phases of a project and process management.					
<u>Project States</u> **A project starts, run and completes it lifecycle. When there is no problem, projects progress normally and gets completed. When there is any problem in a project it may slow down, stuck, slow, stop or close without completion.**						
20	Ideally there does not exist any project without any problem or issue.					
21	Sometimes project get stuck at some point due to severe nature of problems or project progress may get slow down.					
22	Projects that face severe problems may be terminated/closed without completion if problems persist and couldn’t be resolved.					
23	Most of the projects survive during problem phases and completes their life cycle and closes after completion.					
24	Majority of the problems normally appear in the projects after they have passed through their early phases (delivery of few early milestones).					
25	Each project faces problems or enters into a problem phase at least once in its life and may face it many times till its completion.					
26	The existence of the issues & problems and response of the project to the key procsses determines the condition and state					

	of the project.					
27	Behavior and response of each project to these states is different from other projects.					
Process Tailoring (customization) **Add, Delete/Skip, Modify activities of a process, Splitting an activity and selecting the most suitable part of the process, or Merging similar activities of a process into a single activity.**						
28	Small and medium size companies due to resource limitations usually do not follow the traditional heavyweight processes and approaches like CMMI and ISO etc.					
29	Tailoring (customizing) an existing process according to their project requirements is the suitable choice for such companies.					
30	A software process should be tailored (customized) based on the client's perspective as the requirement of a project.					
31	During process tailoring new activities are added, existing activities are modified, deleted or skipped as per requirement of the project and client.					
32	Process tailoring in agile environment makes processes more effective, light weight, focused and result oriented.					
33	During early phases of a project moderate process tailoring is required because project is premature at that time and processes are not well established.					
34	During later phases of the project maximum process tailoring can be done because project and processes are quite mature till then.					
35	When project is completely mature and stable, minimum process tailoring should be performed to avoid creating any problem in the smooth progress of the project.					

36	When project is about to complete; only remaining works should be completed and no new activity or process should start.					
37	When project is about to complete, processes should be winded up, un-necessary or needless phases, activities or artifacts should be reduced in number and closed.					
38	Process tailoring is a good option in agile based development to manage projects, and handle processes through lightweight approaches.					
39	Tailoring a process (based on requirement) for each state of a project can be very effective in handling processes and managing projects.					

APPENDIX J

Process Tailoring Schema

<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="2">Take off</th><th colspan="2">Running</th><th colspan="2">Landing</th><th colspan="3">Hang up</th></tr>
<tr><th>Pre Takeoff</th><th>Take off</th><th>Running</th><th>Post running</th><th>Landing</th><th>Post landing</th><th>Crawling</th><th>Swing</th><th>Pre-running</th></tr>
<tr><td colspan="2" rowspan="2">Resource Management</td><td rowspan="8">Add (+)</td><td colspan="3">Add (+)
Delete/Skip (-)
Modify (Δ)</td><td rowspan="2">Delete/Skip (-)</td><td>Shrink (⋈)</td><td colspan="3">Add (+)
Delete/Skip (-)
Modify (Δ)</td></tr>
<tr><td></td><td colspan="2">Split & Select (⊢)
Merge (⊻)</td><td rowspan="3">Wrap up (⊕)</td><td colspan="2">Split & Select (⊢)
Merge (⊻)</td><td></td></tr>
<tr><td colspan="2" rowspan="2">Communication (Interaction & Coordination)</td><td colspan="3" rowspan="2">Add (+)
Delete/Skip (-)
Modify (Δ)</td><td rowspan="2">-</td><td colspan="3">Add (+)
Delete/Skip (-)
Modify (Δ)</td></tr>
<tr><td colspan="2">Split & Select (⊢)</td><td></td></tr>
<tr><td rowspan="4">Requirement Management</td><td rowspan="2">Requirement gathering & tracking</td><td colspan="3">Add (+)
Delete/Skip (-)
Modify (Δ)</td><td rowspan="4">Delete/Skip (-)</td><td rowspan="4">Shrink (⋈)
Wrap up (⊕)</td><td colspan="3">Add (+)
Delete/Skip (-)
Modify (Δ)</td></tr>
<tr><td></td><td colspan="2">Split & Select (⊢)
Merge (⊻)</td><td colspan="2">Split & Select (⊢)
Merge (⊻)</td><td></td></tr>
<tr><td rowspan="2">Tasks allocation</td><td colspan="3">Add (+)
Delete/Skip (-)
Modify (Δ)</td><td colspan="3">Add (+)
Delete/Skip (-)
Modify (Δ)</td></tr>
<tr><td></td><td colspan="2">Split & Select (⊢)
Merge (⊻)</td><td colspan="2">Split & Select (⊢)
Merge (⊻)</td><td></td></tr>
</table>

APPENDIX K

Calculation of Repetition Factor

Repetition Factor

The overall frequency of tailoring in each key process 6.4(a), 6.4(b), and 6.4(c) in chapter 6 has been found in the range of 1 – 3 times. Therefore, frequency 1 means that 1 time process tailoring is performed. If frequency is 2, it means that there is a repetition of 1 time i.e. 1 time actual and 1 time repeated. So it means the frequency of 3 times will be considered as follows:

Total Frequency: 3 times.
Actual: 1 time
Repetition: 2 times

Therefore, in order to determine the actual number of tailoring activities, the repetition factor has been calculated and excluded from each project. The formula is as follows:

Repetition Factor = ((Frequency of Tailoring of example*N1* – 1) + (Frequency of Tailoring of example*N2* – 1)… + (Frequency of Tailoring of example*Nn* – 1)) ÷ N

By applying the above formula:
Repetition Factor of Project ‘A’ = 1.6
Repetition Factor of Project ‘B’ = 1.3
Repetition Factor of Project ‘C’ = 0*
*Repetition factor 0 means there was no repetition.
Number of Activities Repeated = Total number of activities tailored in Project’X’÷ Repetition Factor

APPENDIX L

Framework Implementation Guidelines

Tailoring Framework
Implementation Guidelines

Prepared By:

Rehan Akbar

1. What is Process Tailoring?

Customizing or adapting a software development process is termed as "tailoring". During process tailoring procedure, the processes, sub-processes and activities of a software development process are tailored according to the requirements of the project.

2. What is client's Perspective?

Client perspective means to understand the client, anticipate him/her, and think in the way he/she thinks. Furthermore, it deals with understanding client's expectation, requirements what does client has in his/her mind, putting yourself into client's place and environment as well as putting yourself into client's shoes.

3. Activities of Process Tailoring

To customize the software development process, the tailoring operations as shown in the following table are performed.

Table 2L: Process Tailoring Operations

Activity/Operation	Description
Add (+)	Addition or adoption of one or more processes sub-processes or activities in an existing set of processes.
Delete/Skip (-)	Removing or leaving one or more non required processes, sub-processes or activities from existing set of processes.
Modify (Δ)	Updating or changing the existing one or more processes, sub-processes or activities.
Split & Select (¬)	Dividing a process, sub-process or activity into one or more sub-parts and selection of the most suitable and most required one or more activities or processes or sub-processes.
Merge (⊻)	Combining two or more processes, sub-processes or activities into single process, sub-process or activity based on their definition and performance.
Shrink (⋈)	Reducing or downsizing the number of steps, sub-activities or sub-processes from inside an activity or process.
Wrap up (⨁)	Winding up or closing the activities or processes being performed to close the project.

4. Project States Definitions

Definitions of each state as presented in Table 3L below, helps to identify the current state of the project.

Table 3L: Process Tailoring Operations

State	Sub-state	Description
Takeoff	Starting state of the project.	
	Pre-Takeoff	An initial state of the project when preliminary requirements are received, project team is selected and initial set up is being made.
	Takeoff	A project state when initial level development gets started and early iterations are in progress. It is the beginning for the development of the first milestone.
Running	The state when after project takes off, development and coding is in progress.	
	Running	A project state after the completion of a few early iterations. Some deliverables have been released to clients.
	Post running	It is a state when project has become matured after the completion of major iterations. For example, release of beta versions.
Hang up	The state when project progress gets slow down due to the presence of problems.	
	Crawling	When project is in this state the progress of the project tends to slow down, delivery dates are not met, requirements are not fulfilled and tasks are not properly allocated. Client is also not satisfied and project seems to lose client's trust.
	Swing	A state after crawling when revolutionary project management measures are taken up to overcome the issues faced by a project. Development processes are slowed down during this state due to the streamlining of the overall process. Quick project management measures are taken at micro-level.
	Pre running	A state when development processes are speed up after swing state modifications. Project again enters into its running state.
Landing	The state when after completion of all development tasks project is near to end.	
	Landing	A state when project is near to its completion. Last few deliverables are in progress. Deliverables might be a part of plan or newly received requirements from clients.
	Post landing	A state after completion of the project when priority tasks and requirements are about to finish. No new requirements are received to work on. Bugs are being fixed and necessary documentation is done. Status of all requirements is set to be verified and closed.

The definitions of each state of the project help in understanding the state and its identification during the project execution.

5. Process Tailoring Framework Implementation

To implement the framework follow the steps as labeled on the framework diagram as shown in Fig 4L.

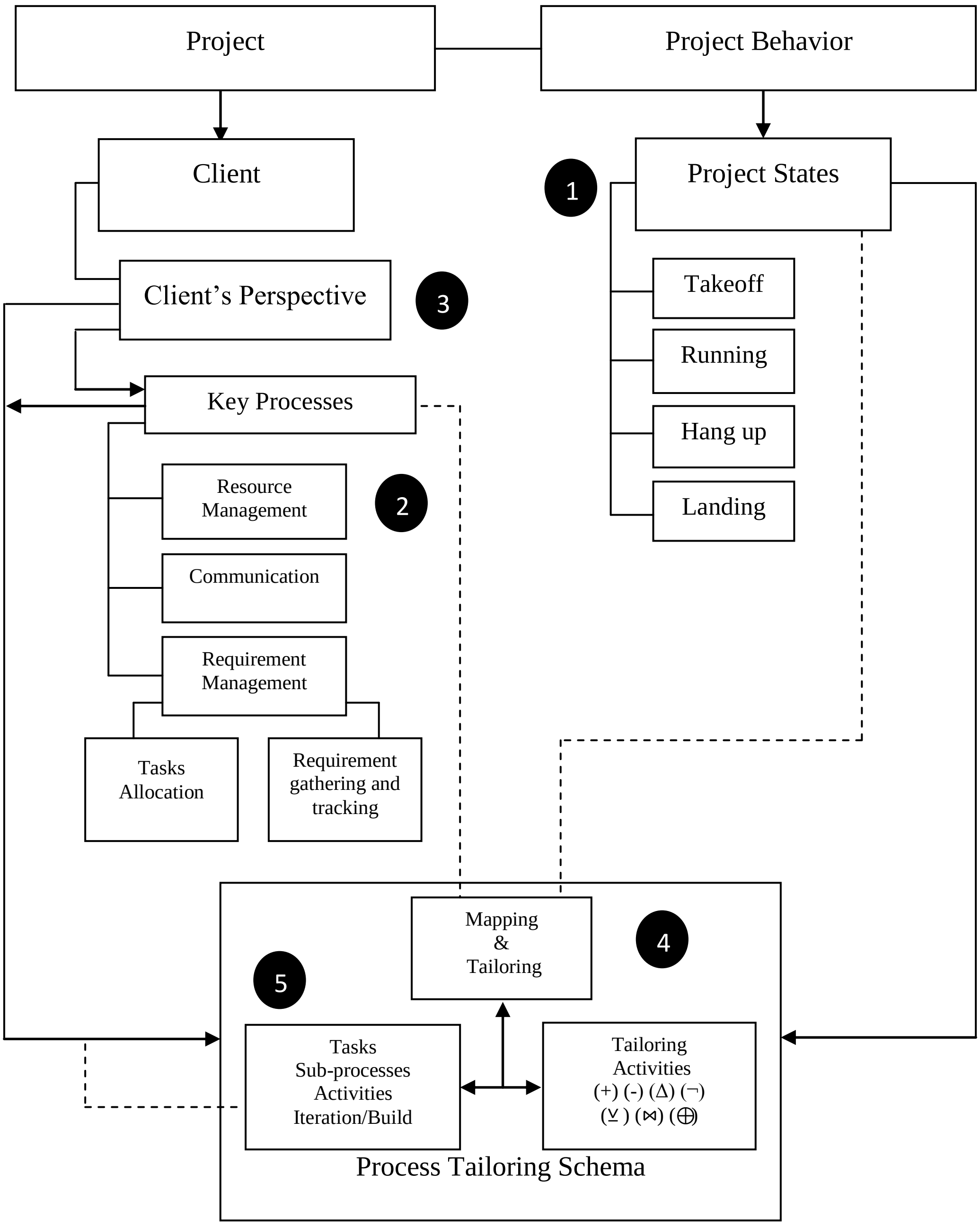

Fig. 4L. Process Tailoring Framework Implementation Steps

A. Steps to Tailor the Key Processes

Step 1: Identify and select the state of the project according to the definition of states as given in Table 3L.

Step 2: Select the key process as shown in Fig. 4L which is required to be tailored.

Step 3: Understand the client's perspective, his/her requirements and expectations from the selected key process.

Step 4: Select one or more tailoring operations as given in Table 2L which are required to perform tailoring of the selected key process.

Step 5: Tailor the selected key process according to the respective state of the project.

Note: Tailor all three key processes one by one as required according to the states of the project. Follow steps 1 – 5 every time.

B. Observations

1. Note down the processes, sub-processes and activities tailored as well as how many times tailoring is done.

2. Record the following observations/data:

    i. Most common tailoring operations performed for each key process during each state of the project.

    ii. Number of processes, sub-processes and activities tailored for each key process during each state of the project.

    iii. Calculate average number of processes, sub-processes and activities tailored for each key process during each state of the project.

    iv. Count the number of problems faced by the projects.

    v. Record the duration of each state of the project.

    vi. Which activities, processes and sub-processes are finally selected after tailoring?

    vii. Examples/scenarios which exist during process tailoring procedure.

C. Outcome

1. Remove garbage or unnecessary data from the recorded observations.

2. Present observations in a more structured and understandable format.