Measuring the impact of the maturity of Requirement Engineering Process on the software project cost and schedule

by

Bushra Sharif

2009-NUST-MSPHD- CSE (E)-12

MS-09 (SE)

Submitted to the Department of Computer Engineering in fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
SOFTWARE ENGINEERING

Thesis Supervisor

Dr. Shoab A. Khan

College of Electrical & Mechanical Engineering

National University of Sciences & Technology

2012

DECLARATION

I hereby declare that I have developed this thesis entirely on the basis of my personal efforts under the sincere guidance of my supervisor Dr. Shoab A. Khan. All the sources used in this thesis have been cited. The contents of this thesis neither as a whole nor as a part has been copied out from any source. No portion of the work presented in this thesis has been submitted in support of any application for any other degree of qualification to this or any other university or institute of learning.

____

Bushra Sharif

ACKNOWLEDGMENTS

All the prayers and thanks to Almighty Allah, The Most Merciful and The Most Beneficent. I am thankful to Almighty for giving me knowledge, power and strength to accomplish this task

My special thanks are for my supervisor **Dr. Shoab A. Khan** for his sincere guidance, valuable opinion and faith in me. I appreciate his dedication for my thesis. I would specially like to thanks **Muhammad Wasim Bhatti** for his guidance at every step. I am thankful to all the faculty members of the department of computer software engineering for their support and help all the time. I appreciate **Dr Aasia Khanum, Dr Arslan Shaukat and Brig Rafiuddin** for their in time guidance.

Lastly, I would like to thank my family and friends for all their love and support. For my parents who believe in me and my efforts during my entire life.

Dedicated to my beloved parents and to all those, whose prayers always pave the way to success for me.

ABSTRACT

It is well acknowledged fact that software development is a dynamic process so during the development of a software project many requirement changes are proposed. These proposed changes have the potential to affect the software development in different dimensions. Cost and schedule of the software project are among those major dimensions that are affected by Requirement change.

This research work demonstrates the impact of a requirement change on Cost and estimated schedule of a software project. Cost in terms of development effort (i.e. total working hours) to implement that change and variance in schedule in the context of week days that occurs due to the implementation of change. This empirical study suggests a way to compute the effort and schedule variance with the help of Regression Equation by performing Correlation and Regression analysis on the change request data collected from 9 different Software projects of Pakistan Software Industry. This analysis is based on conceptual model for cost and schedule variance estimation. This research study also discusses the systematic impact analysis approach to analyze how the impact of a change in requirement propagates from one phase of Software Development Life Cycle (SDLC) to other phase.

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

REP	Requirement engineering process
RV	Requirement volatility
SRS	Software requirement specification
UC	use case document
RTM	Requirement traceability matrix
QTM	Quality traceability matrix
DTM	Development traceability matrix
CID	Change ID
QC	Quality control
QA	Quality Assurance
SDLC	Software development life cycle
SV	Schedule Variance

Chapter 1

INTRODUCTION

The requirement gathering and analysis phase is one of the most crucial steps in any project. Lack of unclear requirements or incorrect requirements has lead software project unable to meet cost, schedule or performance objectives or all three. The quality of requirements phase dictates the quality of all the subsequent phases in the project. The requirements should be clear concise and well written and the overall process should be well managed to ensure the smooth working of the project.

Literature reveals this evidence that well defined requirements have positive effects on downstream software development. A mature Requirement engineering process can increase the quality of software in terms of increased developer productivity, software within cost and estimated schedule. So with good Requirement Engineering practice the software development problems can be addressed in better way.

1.1. PROBLEM OVERVIEW

Requirements evolution is mandatory for any software project. Users can propose requirements change at any stage of SDLC. Although change in requirements may affect Cost, Schedule and Quality of software project [3] but change should be allowable when it is inevitable to meet the customer expectations. Change can be the one of the difficulties in software development [2].

When a change is occurred during the implementation of existing requirements its impact is not only limited to that particular phase where change was proposed but also propagate to other subsequent phases of SDLC [1]. Because of this propagation effect cost in terms of development effort and schedule of the project is directly affected [6]. Frequent changes in project requirements interrupt the project and contribute to greater effort each time work is resumed.

Initially the schedule is developed to complete a particular project on the basis of opening requirements provided by the client. Usually the clients add or modify requirements at later stages of the project development life cycle, due to which more tasks and activities are needed to add to the predefined scheduled tasks to accommodate those changes. To perform each task some additional effort in terms of working hours is required, these increased working hours extend the project duration and lead to delays in the initial estimation of the timelines.

1.1.1 Research Questions

i. How a change can affect the cost and schedule of software project?

ii. How the effort and variance in schedule due to implementation of change can be computed?

1.1.2 Hypothesis

Change in requirements during software development process increases the project effort.

1.2. RESEARCH OBJECTIVES

The purpose of this research study is to demonstrate the impact of a requirement change on cost of a software project in terms of development effort i.e. total working hours to implement that change and its impact on the estimated schedule of the Project .To suggest a way to compute the effort and variance in schedule with the help of Regression Equation by performing Correlation and Regression analysis on the change request data collected from 9 different Software projects.This research study also identifies those requirement change attributes which are the potential factors for the estimation of the effort and variance in schedule due to change by suggesting a conceptual frame work for cost and schedule variance estimation. This empirical study also discusses the systematic impact analysis approach to analyze how the impact of a change in requirement propagates from one phase of System Development Life Cycle (SDLC) to other phase.

1.3. THESIS OUTLINE

The thesis is logically broken down so that each chapter builds on the learning's from the previous chapters. Chapter 2 highlights the importance of REP and discusses the problems faced by software industry due to immaturity of REP. Chapter 3 describes the research work done in this area. Chapter 4 describes the research methodology that has been used to provide the solution of the problem statement and provides details about qualitative analysis and the information gathered during this activity and describes a frame work to analyze the impact of change with respect to phases of SDLC. Chapter 5 provides information about the data sources and the documents that are used to extract the required information. Finally chapter 6 presents the results and generic regression equations that can be used to compute the cost and variance in schedule that occurs due to change in requirements. Finally chapter 7 concludes the thesis and presents directions for future work.

Chapter 2

BACKGROUND

2.1 INTRODUCTION

This chapter is a detailed description of the related work that has been done in this area. It describes different impact analysis methods and models proposed by researchers to analyze the impact of a change on the existing software development process.

2.2 REQUIREMENT ENGINEERING

A research group at NATO [18] in 1967 introduced the term software engineering. NATO approved it as an engineering domain at the Software Engineering Conference in 1968 claiming that software development required integration of engineering principles like other engineering domains. Requirement engineering is considered as the branch of software engineering and can be defined as

i. RE is the process of instituting those required services that system should provide and it operates under some constraints [14].

ii. RE is the branch of software engineering and it is apprehensive with real world objectives for and functions of and constraints on software systems. The relationship of the stated above factors with the accurate measurement of the software behaviour is also analysed [13].

2.3 REQUIREMENT ENGINEERING PROCESS

According to Karl E. Wiegers RE process can be divided into two sections [15].

i. Requirements Development

ii. Requirements Management

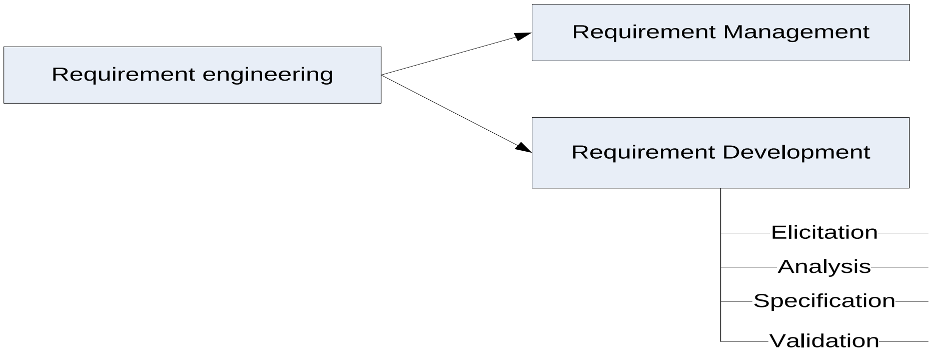


Figure 1 Requirement engineering process [17]

2.3.1 Requirement Development

Requirement development is further divided into four sub disciplines i.e. Elicitation, Analysis, and specification and validation.

2.3.1.1 Requirement Elicitation

During elicitation phase different requirements gathering techniques are used by RE engineers to gather the requirements from the customers for the new system to be developed. Following elicitation techniques can be used to extract these requirements from the user.

i. Interviews

ii. Requirement Workshops

iii. Competitive product analysis.

iv. Review of existing system documentation.

v. Meetings with customers

vi. Event Lists.

vii. Story Boarding

viii. Prototypes

ix. Questionnaires

x. Methodology

2.3.1.2 Requirement Analysis

During analysis phase the collection of unstructured requirements are analyzed so that related requirements can be grouped and prioritized. Conflicts among requirements are also resolved at this stage. Requirements analysis is very important for the success of the project. Requirements must be actionable, measurable, and testable, related to identified business needs, and defines to a level of detail sufficient for system design. Requirements can be functional and non-functional. Following are requirement analysis techniques.

i. Scenario Construction

ii. Goal Oriented Analysis

iii. CATWOE Analysis (Checklist method by Peter Checkland)

iv. Task Analysis

v. Domain Analysis

vi. Brainstorming

vii. JAD (Joint Application Design – a special form of prototyping)

viii. Prototype

2.3.1.3 Requirement Specification

Requirements Specification is a detailed description of the function of the system to be developed. During this phase requirements are documented in natural language amplified with graphical models. The graphical models can be the following

i. Data Flow diagrams (DFD)

ii. Entity-Relationship diagrams (ERD)

iii. State-Transition diagrams (STD)

iv. Dialog Maps

v. Use-Case Diagrams

vi. Class Diagrams

vii. Activity Diagrams

2.3.1.4 Requirement Validation

The purpose of this activity is to make sure that the user needs are properly understood and documented before they are incorporated into design and development. The verification activity starts with the following:

1. Find defects and gather data, where defects at verification activity refers to information that can be like:

i. **Unambiguous**

The description is vague and not easy to understand

ii. **Incomplete**

Things are missing and information is not complete, e.g. the response time, attachment types, attachment size, etc.

iii. **Inconsistent**

The information is inconsistent across document

iv. **Complex**

The goals mentioned are not clear. Features mentioned are not relevant.

v. **In-secure**

Security measures not mentioned and/or not required. e.g. session timeouts, concurrent login permitted, user authenticity secure, etc.

vi. **Unachievable**

The feature might not be implemented due to any reasons, such as lack of training, less number of resources, higher costs, non-availability of test environment, etc.

During this process following techniques are used [14].

i. Requirement reviews

ii. Prototyping

iii. Test case generation

iv. Formal inspections

v. Peer reviews

vi. Checklist

2.3.2 Requirement Management

Requirement management is concerned with maintaining an agreement on the requirements for a particular software project with the clients. Requirements keep emerging and changing due to change in stakeholder needs, environmental change or due to changes in policies and rules of the involved organization.

To properly mange the requirements following activities are carried out.

i. Requirements are baseline.

ii. When change occurs in requirements, impact analysis is performed.

iii. After the approval change is incorporated into the project.

iv. When change occurs in requirements, impact analysis is performed.

v. Requirements are baseline.

vi. When change occurs in requirements, impact analysis is performed.

vii. After the approval change is incorporated into the project

viii. Change is communicated to all the stakeholders

ix. Project plans and other work products are kept up to date with the requirements.

x. Traceability is maintained from design document to application code and from code to test cases.

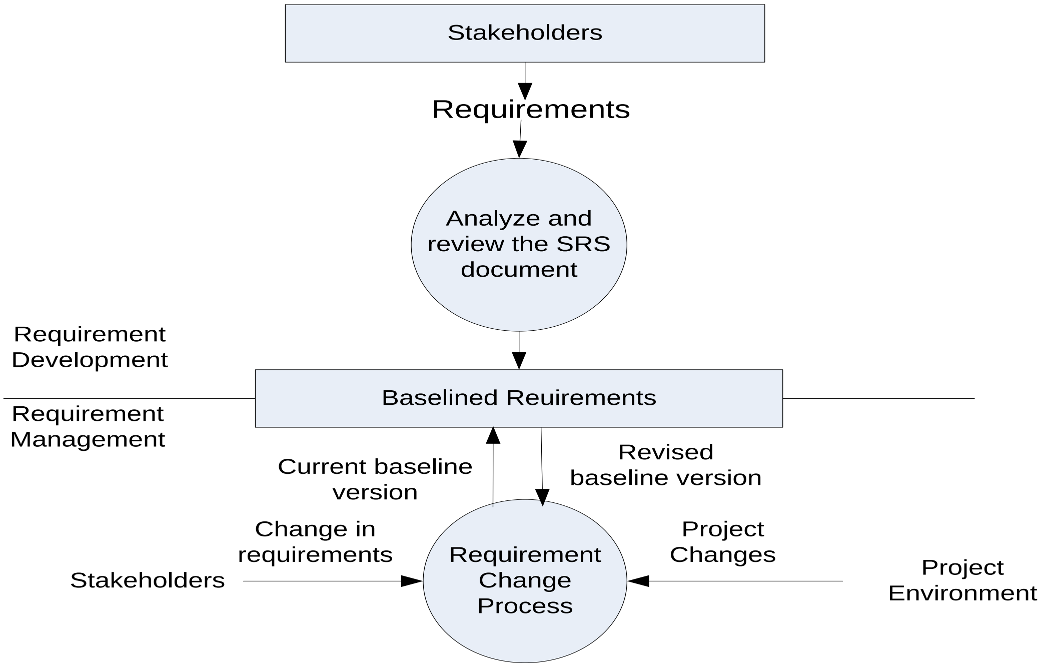


Figure 2 Procedure of Requirements Management [15]

2.4 PROBLEMS OF SOFTWARE INDUSTY

In spite of major software advances and introduction of various development techniques, software industry still faces the following challenges while developing software systems; they include, late product delivery, residual faults, over budgets, etc. The Standish group studied 20,000 development projects that were undertaken in the year 2000.The results of this study indicates that only 28 percent of the projects were successfully completed, 23 percent were cancelled before completion or never got implemented in the first place. The remaining 49 percent were completed but were over budget, late, had fewer features than those specified by the customers .Following can be the major causes of project failure.

2.4.1 Poorly defined, incomplete and missing requirements

Incomplete and changing requirements are considered one of the major causes of project failure [Standish]. Due to these missing and incomplete requirements few software products are delivered on time and within budget. To overcome these problems a mature REP is very important. This is because if REP is not mature and some of the requirements are not completely captured during elicitation phase then these missing or incorrect requirements can lead to extensive rework cost when the errors are identified in later phases of SDLC. According to Frederick P. Brooks

During the development of software system the hardest part is to decide what to build. The establishment of technical requirements is the most difficult conceptual work. Technical requirements include interfaces to people, to other systems and to machines. If this conceptual work is done wrong it cripples the resulting system. This is the most difficult part to rectify later [12].

The cost that incurs in a software development due to erroneous requirements increases with delay in their correction and depends on how soon these erroneous requirements are discovered. This cost to fix these errors is greater because the system design and implementation has to change to rectify the problem and increases with the flow of the development lifecycle towards deployment. As shown in the following diagram

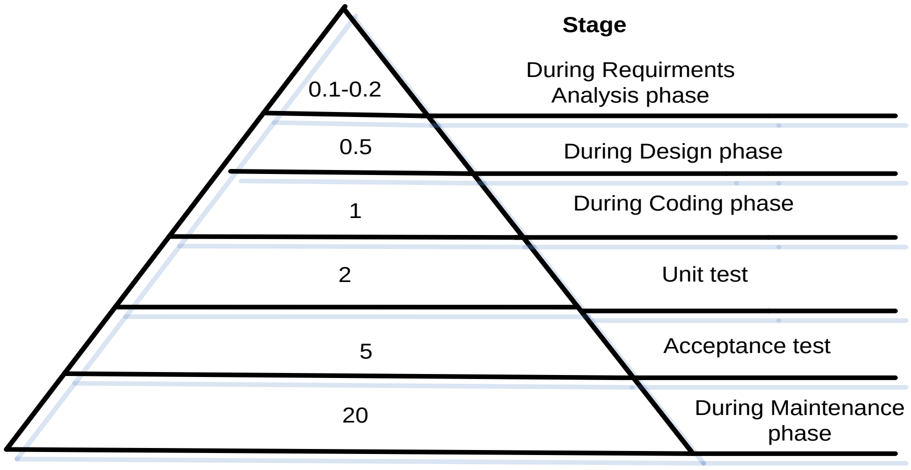


Figure 3 Cost to repair a defect during SDLC phases

Most major reason for erroneous requirements include starting the RE process with little or no planning, which eventually results in poor quality requirements and lesser control of overall RE process management. A list of often encountered RE process problems include [18, 19]

i. Vague stakeholder requirements
ii. Undefined software requirement process
iii. Inadequate requirements traceability
iv. Lack of proper stakeholder involvement.
v. Inadequate attention to business needs
vi. Absence of requirement management process
vii. Requirements not reflecting the real problem needs of the stakeholders
viii. Absence of requirement change management
ix. Misunderstandings/misinterpretation between customers and software engineers
x. Lack of stakeholder communication

2.1.1 Change in requirements

Requirement volatility is considered as one of the major challenges faced by software industry.

Requirement volatility is considered as one of the major challenges faced by software industry. Due to changing needs of stakeholders and work environment the requirements usually expand during the development of a software project. And this slow expansion in project requirements results in scope creep. This is because every change in requirement incorporates some new functionality; the longer the software project will go the more growth in scope would be experienced. This phenomenon of scope creep can affect the whole software project in different dimensions as shown below.

i. Changes in Requirements can lead to Expansion in Project Scope (RV+→ Scope Creep+)

ii. Scope Creep can lead to Increased Project size (Enhanced Functionality).(Scope Creep+→Project Size+)

iii. Increased Project Size can lead to More Effort (Man Days) (Project Size+→Effort(Man Days)+)

iv. More Man Days can lead to Extended Project Timelines (Man Days+→Project Duration+)

v. Extend project Timelines can lead to Schedule Delays.(Project Duration+→Schedule Delays+)

2.1.2 Causes of change in project requirements

The conciseness of requirement definition, the type of system development methodology used, software components, poor communication between user and developer [6] and technological changes are those factors that contribute to Requirement volatility.

2.5 EFFORT ESTIMATION

Effort estimation is the most significant aspect for project managers because it helps to plan the forthcoming activities. Effort estimation can be defined as the prediction of working hours and number of resources is needed to perform a particular task [19].

For effort estimation following approaches are used

i. **Expert estimation:** In expert estimation, the required effort for a software project is measured on the basis of judgment. The term “Expert” may be used for an individual or for a team. It is assumed that experts typically possessed more information and have the more flexibility that how information is processed [21]

ii. **Formal estimation models**: In model based estimation, the quantification step is mechanical. For example, use a formula derived from historical data like COCOMO. However, it may be complicated to build models for software development effort estimation because of lack of stable relationships and use of small data sets to build models [21].

iii. **Combination based estimation model**: It is the combination of above two approaches where effort estimation is based on a judgmental or mechanical combination [20].

2.6 SUMMARY

In this chapter a background study on REP is presented. Problems of software industry are discussed in detail. Different effort estimation approaches are briefly introduced.

Chapter 3

LITERATURE REVIEW

3.1 INTRODUCTION

This chapter is a detailed description of the related work that has been done in this area. It describes different impact analysis methods and models proposed by researchers to analyze the impact of a change on the existing software development process.

3.2 RELATED WORK

In [1], Muhammad Wasim Bhatti and Nadeem Ehsan analyzed the impact of requirement change with respect to the development phase of a software project. They reported that changes can be proposed during any development phase however more changes are proposed by customer during maintenance phase. Finally concluded that changes proposed in requirement phase and changes proposed in design phase & changes proposed in design phase and changes requested in testing phase have significant relationship.

In [9], Didar Zowghi and Nurmuliani defined different types of Requirement Volatility i.e. requirement volatility in early and later phases of software development life cycle [9] and they finally concluded that change in requirements during later phases of SDLC is more destructive because it affects the quality of software.

During the development of a software project the volatility of requirements in early phases is considered as Pre-SRS: Requirement Volatility. The early phases are Elicitaion and Analysis of requirements. Requirement Volatility in later phases during the development of a software project is referred as post-SRS: Requirement volatility. The later phases are Design. Implementation, testing, maintenance.

Due to changing needs of stakeholders and work environment the requirements usually expand during development of software project and due to this requirements become even more volatile. In [6], Didar Zowghi and Nurmuliani analyzed the impact of requirement volatility on software project in terms of Software project cost and schedule. They identified that conciseness of requirement definition, the type of system development methodology used, software components, poor communication between user and developer are those factors that contribute to Requirement volatility. To analyze the impact of requirement volatility on Software Project Performance they presented a conceptual Model as shown in figure 2.1.

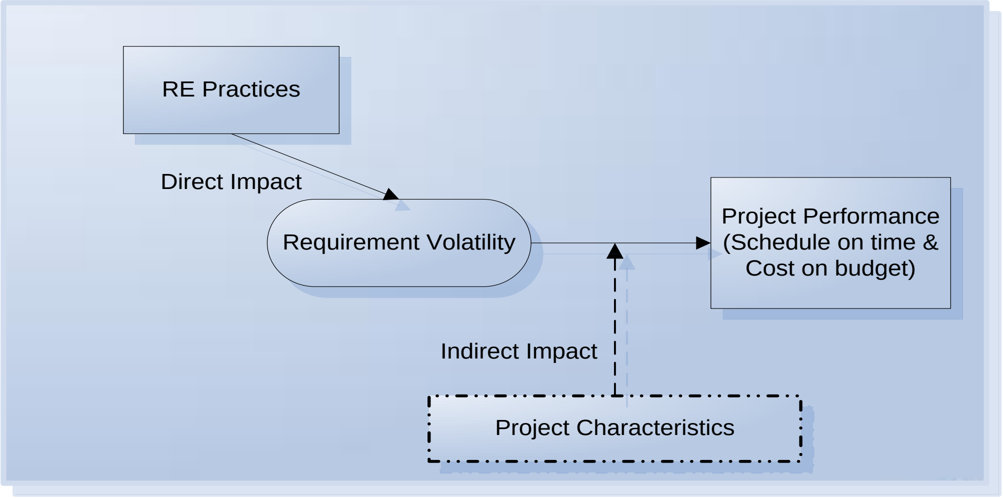


Figure 4 Conceptual Frame Work [6]

In this model they analyzed the direct relationship between Requirement Volatility and Project performance and the impact of other factors like Requirement Engineering Practices and Project characteristics such as Project size and Organization size on this relationship. They finally reported that project performance is being measured as the project that is being developed within budget and within schedule and requirement volatility can affect the project performance, and this impact of RV on project performance can be affected by other factors such as Organization size and project size The coefficient of correlation between requirement volatility and project schedule performance and correlation coefficient between requirement volatility and project cost performance was negative. This negative relationship depicts that the degree of RV is negatively associated with Project Performance.

In [5], Nurmuliani, Didar Zowghi and Susan P. Williams investigated the impact of Requirement Volatility on development effort that is total working hours to implement a change. Their findings reported that if new requirements are added in the later phases during software development it would be a high risk because it will cost the organization in the form of schedule delays or budget overruns.

They identified different requirement change attributes that can be used to estimate effort .These attributes are number of document affected, Source of change (Internal, External) and type of change(addition, deletion, modification). Their study demonstrates that these factors are significantly correlated with the amount of effort to implement a change.

Requirement Change Attributes	**Estimated Effort**
Number of documents affected	0.384**
Total Requirement changes	0.223*
Source of changes (Internal, External)	0.314**
Change types(Deletion, modification, addition)	0.404**

**.Correlation is significant at the 0.01 level (2 -tailed)
*Correlation is significant at the 0.05 level (2- tailed)

Table 1Correlation coefficients of change attribute and change effort [5]

In [7], Evelyn J. Barry, TRIDAS MUKHOPADHY and Sandra A. Slaughter analyzed the relationship between project duration and project effort by developing and evaluating a two-stage model.

They extended the traditional models of Software project effort by adding another variable i.e. Project Duration. The first conceptual frame work demonstrates that Project effort can be relate to various factors like Project duration, Project type, project size and team skill level. Many changes can occur during the project development time period like requirements can be modified, changes in assignments and tasks of development team. All these changes have the potential to affect the project effort, because each time when work is resumed after the interruption programmers spend additional effort and this effort increase the overall effort of the project. They also concluded that due to changing environment project requirements are increased over the Project duration which increases the project size and due to this project effort

is increased.

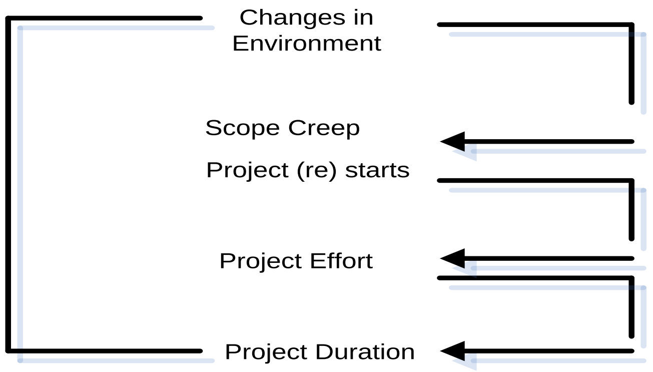


Figure 5 Relationship among Variables [7]

Figure 5 illustrates the relationship between environmental change, project duration and project effort. Project Effort= function (project duration, anticipated project size, project type, team skill level)

Their study demonstrates that project duration is a function of project effort. In the Project duration calculation two variables have more impact dependent project tasks and level of total project control. As the number of sequentially dependent tasks would increase the project duration would be longer.

Project Duration= function (project effort, sequentially dependent tasks, level of total project control)

To determine the impact of a requirement change on software development James S. O'Neal and Doris Carver [22] presented impact analysis method based on requirement traceability. They created classes of requirement changes by identifying attributes of different work products and traces. Then they prioritized those requirement classes according to their potential impact.

Their study define software development Project (SDP) as SDP= (Nodes, Arcs, i, c, e, p), where Nodes are those artifacts which are produced during development of project Arcs refer to requirement traces from source work products to target work products. i is influence function. It depicts the degree of influence that a source artifact can has on target work products. c is complexity function. It represents the estimated complexity of work product. e is the effort function that define the development effort in person hours. P is phase cost function that defines effort required to change a work product in later phases of software development life cycle.

Requirement change has defined as RC**=(Change set, Change Arcs, Nodes, i)**, where change Set-> sets of requirement changes and Change. Arcs->edge that links a requirement change to work product-> set work products during SDP .i represents influence factor that a requirement change has on changed requirement.

According to Yashwant K. Malaiya and Jason Dentson study [8], the requirement volatility is determined by change in requirement after the code has started. Those software projects where requirements are changed after implementation phase has high volatility and where requirements are stable has low volatility. Their study demonstrates that due to requirement volatility the software components have to redesign and this leads to higher defect density in the Software. They identified that Complexity of software, development team skill and development process maturity are those significant factors which contribute in the defect density of a software project.

They analyzed those interface errors which arise due to modification in software components. They made the comparison between defect density which occurs due to changes in requirement and that defect density which arises without changes in requirement. Their findings reported that that those changes in requirements which occur after testing effort have great influence on defect density and can significantly raise defect density.

In [10], Daniela Damian,James Chisan, Lakshminaranan Vaidy Thamsamy reported that there is a positive relationship between improved requirement engineering process and software productivity. According to them a mature requirement engineering process improves overall software development.

To demonstrate the impact of requirement instability on project performance D. Pfahl, K. lebsanft used the simulation models [11].

3.3 SUMMARY

In this chapter previous related work about measuring the impact of changing requirements on software project cost and schedule is presented. It describes that many impact analysis approaches are suggested by researches to analyze the impact of changing requirements but any generic equation is still not derived to calculate the impact of a change request.

Chapter 4

PROPOSED APPROACH

4.1 INTRODUCTION

In this chapter the research methodology is described in detail that has been used to propose the conceptual frameworks which will be used for effort estimation and computation of schedule variance of a software project against the associated Change.

4.2 RESEARCH APPROACH

Fig. 6 describes a novel approach which is used to derive generic regression equations for the estimation of cost and the computation of the schedule variance that came into being due to proposed changes.

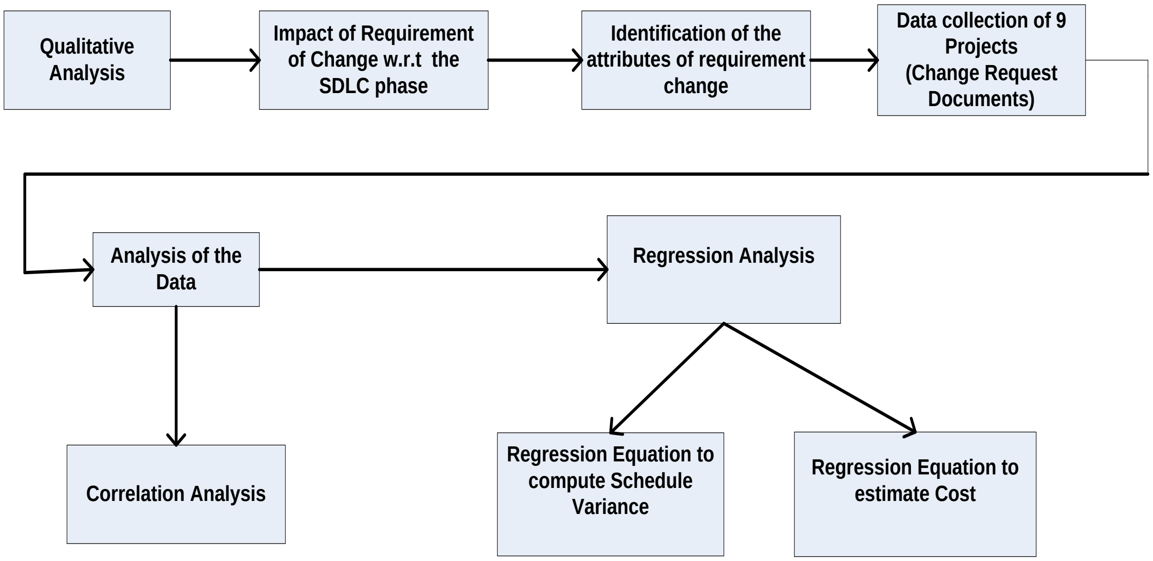

Qualitative Analysis
Impact of Requirement of Change w.r.t the SDLC phase
Identification of the attributes of requirement change
Data collection of 9 Projects (Change Request Documents)
Analysis of the Data
Regression Analysis
Correlation Analysis
Regression Equation to compute Schedule Variance
Regression Equation to estimate Cost

Figure 6 Research Approach

4.2.1 Qualitative Analysis

The first step in the research approach is qualitative analysis. Qualitative analysis is performed to explore the issues, to understand the phenomenon and to find out the answers of the different questions .It helps to search for the "why" of its topic. During this activity unstructured information that is obtained through interviews transcripts is analyzed.

In this research work to understand the impact of changing requirements on software cost and schedule, to identify the significant attributes of a requirement change request and to understand the impact calculation method of a requirement change, qualitative analysis is performed. To achieve this, following questions related to the research topic are asked from the experts of Pakistan Software Industry.

4.2.1.1 Research Questions

i. How changing requirements affect the cost and schedule of software?

ii. How do you perform impact analysis of a change request???

iii. Which type of requirement change requires extensive rework??? (E.g. UI change, Workflow Change, DB change, process change)

iv. If change request arises in Requirement Analysis phase what work products are affected from it?

v. If change request arises in Design phase what work products are affected from it?

vi. If change request arises in Implementation phase what work products are affected from it?

vii. If change request arises in Testing phase what work products are affected from it?

viii. If change request arises in Maintenance phase what work products are affected from it?

ix. How do you measure effort to implement a proposed change in Requirement analysis phase?

x. How do you measure effort to implement a proposed change in Design phase?

xi. How do you measure effort to implement a proposed change in Implementation phase?

xii. How do you measure effort to implement a proposed change in testing phase?

xiii. How do you measure effort to implement a proposed change in Maintenance phase?

xiv. How the change in requirement affects the Project Scope?

xv. How the change in requirement affects the timelines of Project? (E.g. Analysis, Design, Implementation, Testing, Maintenance)

The responses and answers given by experts are discussed in detail in Appendix-B.

4.2.2 Impact analysis approach

Whenever change is occurred during the implementation of existing requirements its impact is not only limited to that particular phase where change was proposed but also propagate to other subsequent phases of SDLC [1]. Because of this propagation effect Cost in terms of development effort is directly affected [6]. Frequent changes in project requirements interrupt the project and contribute to greater effort each time work is resumed. In the second step of the research approach to properly understand the impact of change with respect to development phases, change management procedure is discussed and impact analysis approach is proposed.

4.2.2.1 Requirement change management procedure

Requirement Change management procedure defines how a change is logged, and if approved how it becomes part of the project.

4.2.2.1.1 Change repository

Changes can be reported by any stakeholder of the project. Mostly, they come from client. Each project has defined its communication and reporting means. Reported change is logged into the change repository of that Project. Following are the different statuses of a change request this whole process.

New: Stakeholder logs a change request with the status “New”. Impact analysis in terms of application/requirements is done on the change request.

Approved: Against the approved change the status of change request is updated to “Approved”. Decision details need to be provided, which is usually based on impact of change.

Pending: Change is pending for analysis and further decisions in future. A pending change can be rejected or approved as per management decision

Rejected: Change is set to rejected and it is not implemented

Reopen: A rejected or approved change can be reopened as per management decision for further processing

Planned: A change is planned in project’s plan. Project Manager/Manager changes the status from “Approved” to “Planned”. The project manager/manager decides to implement the change in current release or in any of the future releases. Software’s Release No. is mentioned in change repository. A planned change can also be rejected or set pending. Also, while planning a change, the project authority also needs to mention Reference of Client’s Commitment (approval) and Due Date (the date to provide that change as per commitment) The impact of change in terms of effort and schedule is calculated. The project manager/manager updates the plan (for effort and schedule) if required and ensures the communication of change implementation to all stakeholders through either publishing the plan or generating emails.

Implemented: A planned change is mentioned as implemented in change repository to update that it is developed in software.

Closed: A change is closed when its required and related work products are updated

In Change Repository, along with a reported change the information logged is:

i. Change Request Details – Any information that help to take a decision on change.

ii. Decision Details – Any information regarding work to do or decision of CCB

iii. Notes – Any information that can help in taking decision. Impact analysis is also mentioned

iv. Change type and status

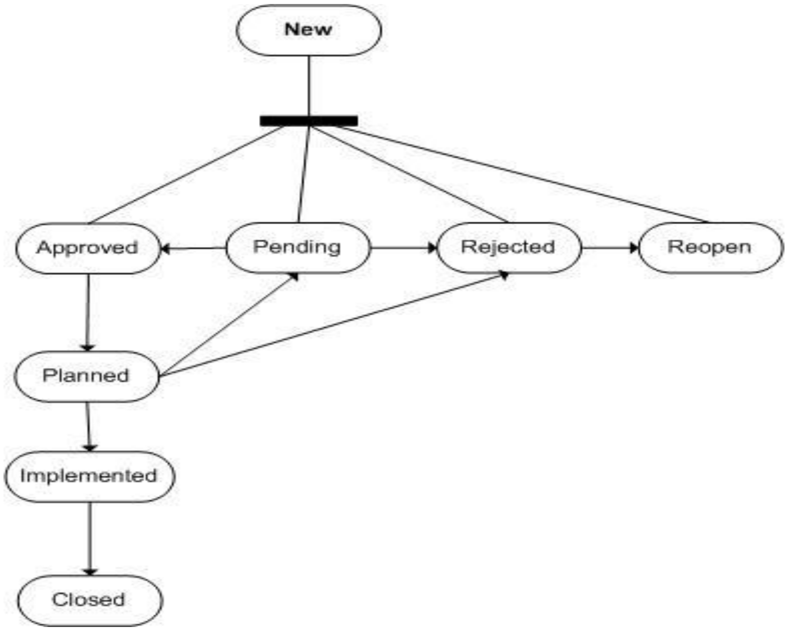


Figure 7 Change Request Status Transition

4.2.2.2 Impact Analysis

The impact analysis of requirement change is done and maintained in change request repository before approval. The impact analysis depicts the impact in following terms

4.2.2.2.1 Analysis and design team

i. Number of new uses cases

ii. Number of deleted uses cases

iii. Number of completely changed uses cases

iv. Number of partially changed uses cases

4.2.2.2.2 Development team

i. Quantum of change

ii. Size of change

iii. Components about to change

4.2.2.2.3 Testing (QC) team

i. Nature of change

ii. Complexity of change

iii. Number of new test cases

iv. Number of deleted test cases

v. Number of completely changed test cases

vi. Number of partially changed test cases

4.2.2.3 Procedure

When a change is requested from any stakeholder it is logged in change log with the "New" status. According to the decision of approving authority the request status is updated .If change is approved as per management decision then its status is updated to "planned' and it is planned for development. After the change implementation its status is updated to implement. All documents impacted by change must also be updated when a change is implemented. Finally the initiated change request is closed by setting the status of the request "closed". (Figure 8)

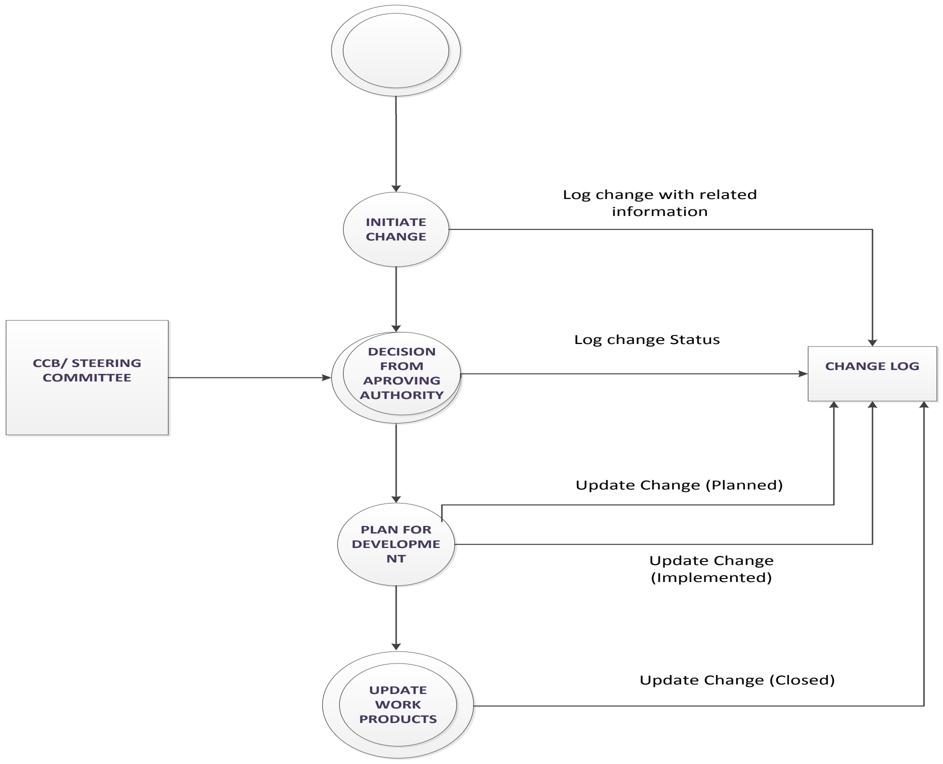


Figure 8 Procedure to handle change

4.2.2.4 Updateable work products and relationship among these artifacts

Figure 9 describes the different artifacts which are developed in each phase of SDLC and address the different research questions (Qualitative Analysis).

4.2.2.4.1 Artifacts of analysis phase

During the analysis and design phase following artifacts are developed.

i. SRS

ii. Use case document

iii. SDS

iv. RTM

v. DB design

4.2.2.4.2 Artifacts of implementation phase

During the implementation phase following artifacts are developed.

i. GUI
ii. DTM
iii. DB Queries
iv. Application Code document
v. Release notes

4.2.2.4.3 Artifacts of testing phase

During the testing phase following work products are developed.

i. Positive Test cases
ii. Negative Test cases
iii. Other Test cases
iv. Flow charts
v. Sanity documents
vi. QTM

4.2.2.4.4 Artifacts of maintenance phase

During the maintenance phase following artifacts are developed.

i. User manuals
ii. External release notes

Impact of a change is calculated using horizontal traceability. To identify those work products that can be affected from requirement change, first the SDLC Phase in which change is requested is spotted.

If change is requested during Requirement analysis phase, it has no chain effect. So only Analysis and Design phase related artifacts which are (SRS,UC,RTM,SDS,DB Design) would require modification which need less working hours and few number of resources, so computed effort would be low to implement change at this stage. If change is occurred during Implementation phase then the artifacts related to this phase (i.e. DTM, release notes, application code) and artifacts related to earlier analysis and design phase (RTM, DB/Design, SDS, SRS) would need modification so more effort would be required as number of artifacts has increased. In testing phase artifacts related to this phase e.g. QTM, test cases and artifacts of earlier phases which include analysis and design and development (RTM, DB/Design interaction (matrix), DTM) would need updation. In maintenance phase, RTM, DB/Design interaction (matrix), DTM and QTM are used to find out size of work (effort) required against a change.   Also, effort to review and update Software Manuals of the module/component is added in it.

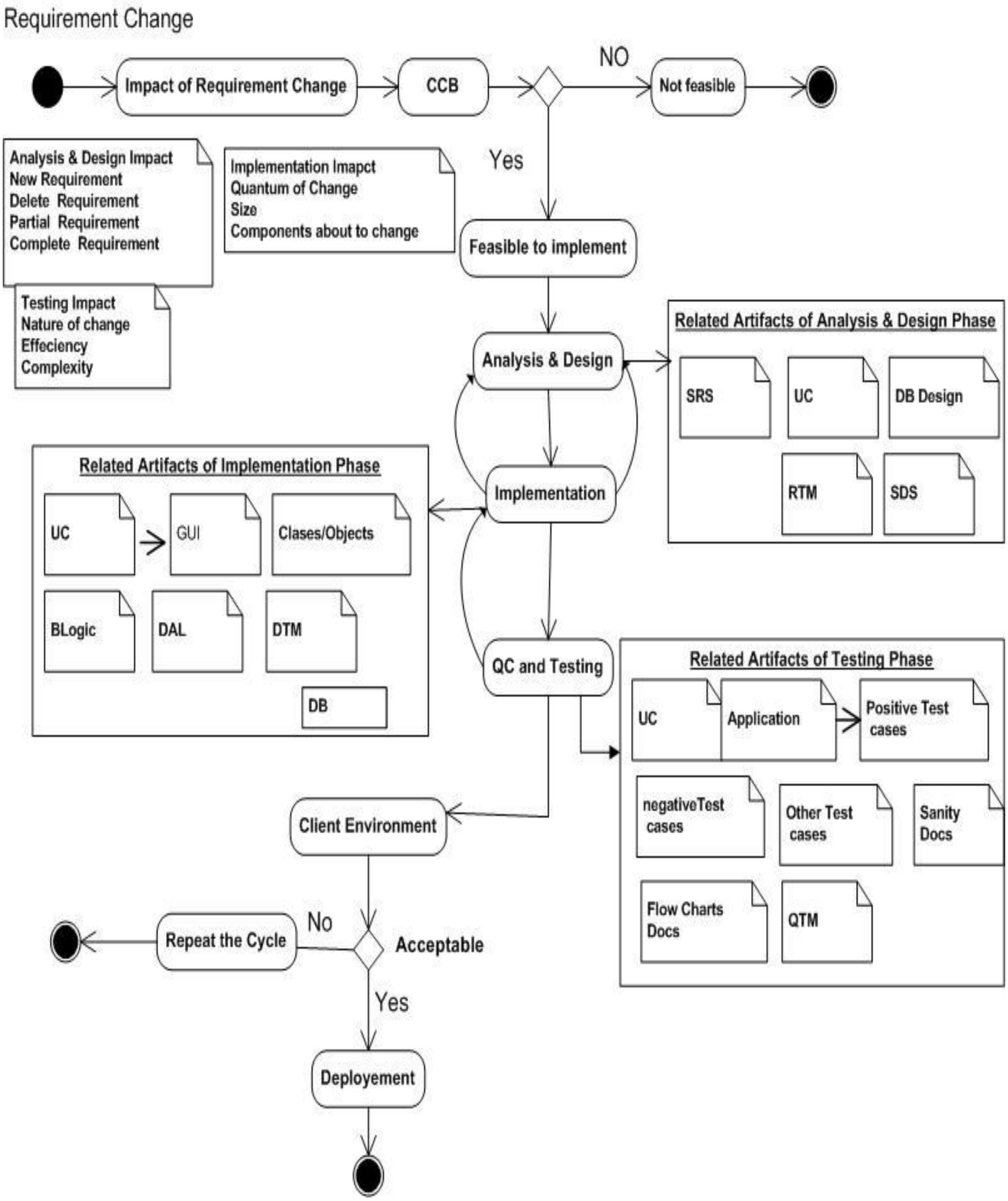


Figure 9 Conceptual Framework

4.2.2.5 Bi-Directional Traceability

Bidirectional traceability is maintained through traceability matrices. Traceability Matrices are used to find out size of work (effort) required against a change. This matrix links a requirement to a UC document and UC document to Interface and UI to different classes and objects and links UC document to Test cases.

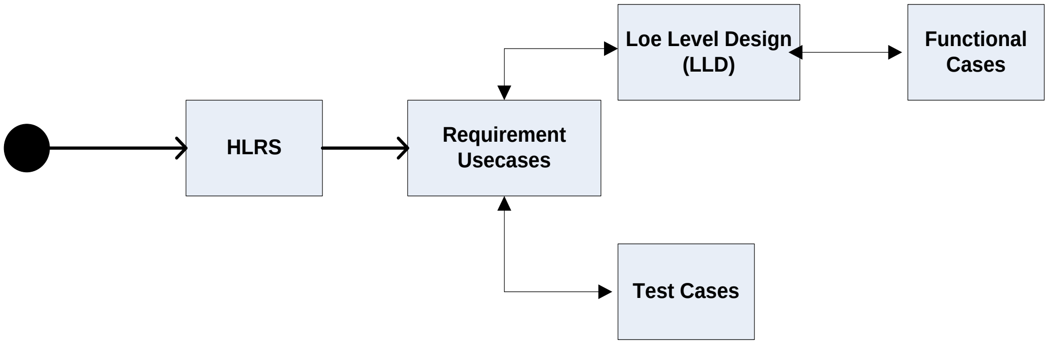


Figure 10 Bi-directional traceability

Analysis teams develop requirement traceability matrix (RTM) and provide this RTM to testing (QC) and Development teams. That traceability, a two column sheet normally in MS-Excel form, contains traces between specification document and use cases. This RTM document contains a link between requirement (i.e. FR1) and use case (i.e. UC1). QC team develops test cases on this matrix and develops their own traceability matrix QTM. QTM document contains a link between use case document and test cases. Same as Development team LLD based on traceability matrix and then develop functional/ classes traceability matrix DTM. DTM contains link between use cases and the developed pages and classes. Any change in any of traceability matrix will reflect change on all other traceability matrix, that’s why these are called bidirectional traceability matrix. Traceability among artifacts determines the link between requirement and its system components

4.2.3 Change request attributes

When change is suggested its impact is calculated on the basis of different attributes. These attributes are the significant factors for the effort estimation and the computation of schedule variance. So during question answer session with the experts the unstructured information would be analyzed to identify those significant change request attributes.

4.2.3.1 Proposed conceptual framework for cost estimation

In this phase, we identified significant attributes of requirement change from change request forms that can be useful in the estimation effort associated with change. Following are the required attributes

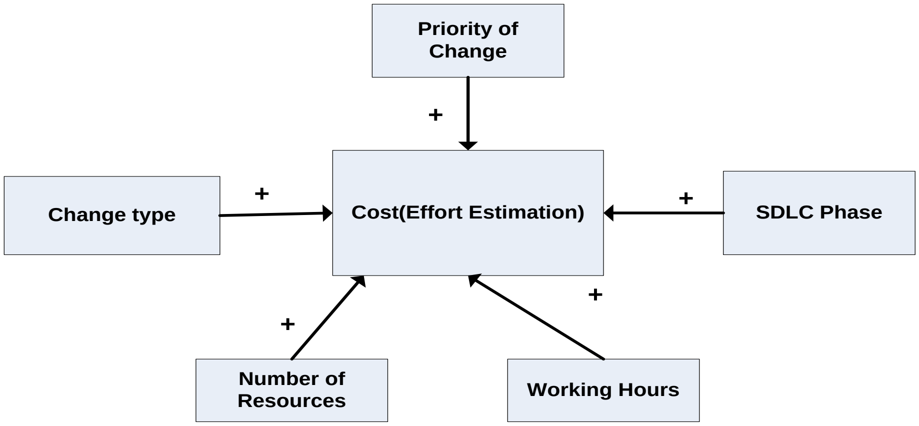


Figure 11 Conceptual Frame work for Cost Estimation

i. Type of requirement change (i.e. GUI, Functionality, Process, Work flow, DB Design),

ii. Software development life cycle phase (Analysis, Design, Implementation, Testing, Maintenance),

iii. Change Priority (Low, Medium, High),

iv. Number of working hours and

v. Number of resources

4.2.3.2 Proposed conceptual framework for schedule variance

In this phase, we identified significant attributes of requirement change from change request forms that can be useful in the estimation of variance in schedule due to implementation of change. Following are the required attributes

i. Type of requirement change (i.e. GUI, Functionality, Process, Work flow, DB Design),

ii. Software development life cycle phase (Analysis, Design, Implementation, Testing, Maintenance),

iii. Change Priority (Low, Medium, High)

iv. Effort (Man Days) as depicted in fig 12.

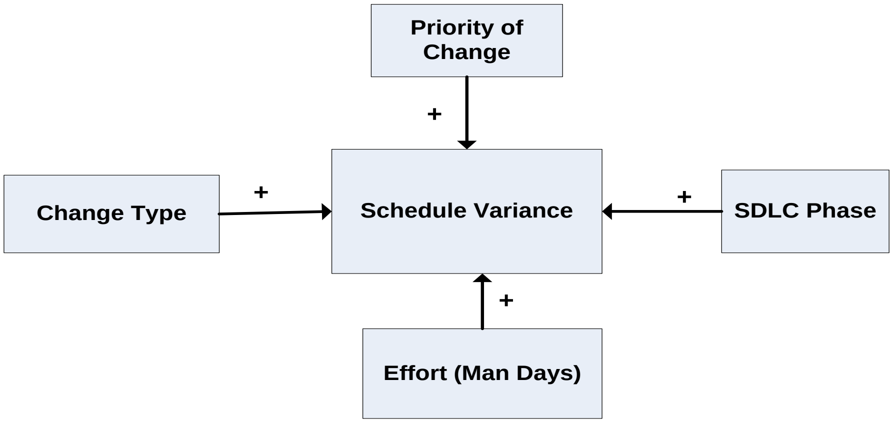


Figure 12 Conceptual Frame work for Schedule Variance Estimation

4.3 SUMMARY

In this chapter the research approach is discussed in detail that is used to derive the cost and schedule variance estimation equations. Impact analysis approach is described and cost and schedule estimation models are proposed.

Chapter 5

DATA COLLECTION

5.1 INTRODUCTION

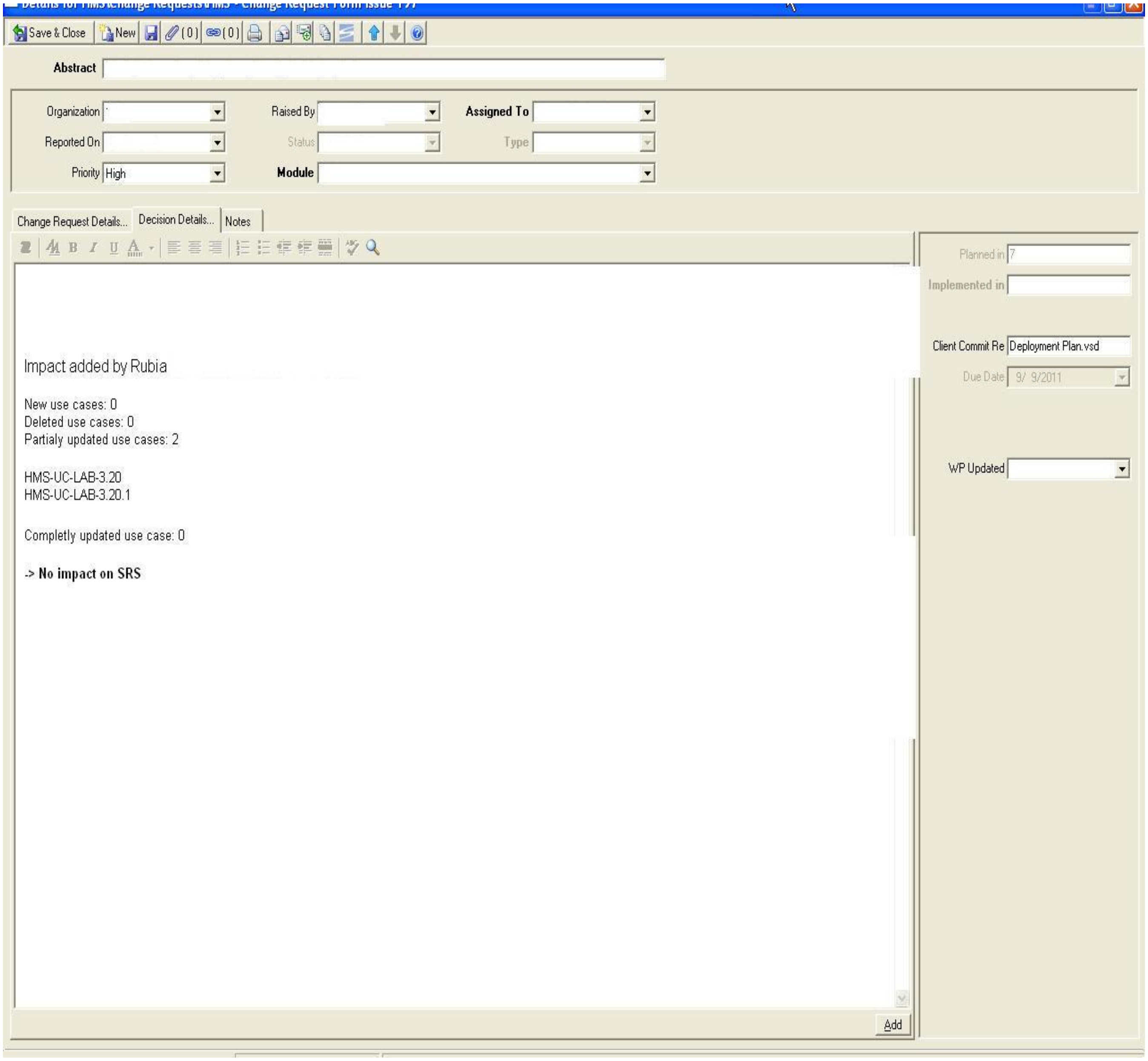
Fourth step in the proposed research approach is data collection. In this chapter we will discuss in detail how the data was collected for the analysis purpose and what were the sources of data.

5.2 DATA COLLECTION SOURCES

To analyze the impact of a proposed change data of different CIDS was required. This information can be obtained from change request forms and documents, timelines of different organizations. For this purpose change request forms and documents are collected from different organizations of Pakistan software Industry. The selected organizations are IAC (Interactive Convergence Ltd), ESOLPK (Electronic Solutions Pakistan (PVT.) LTD, Komatsu Pakistan and all these organization are CMMI certified.

5.2.1 Change request forms

Information about 101 CIDS of different nine projects was collected during this activity. Following change request documents are collected during this phase



Details for HMS\Change Requests\HMS - Change Request Form Issue 177

Abstract

Organization

Raised By

Assigned To

Reported On

Status

Type

Priority: High

Module

Planned in: 7

Implemented in

Client Commit Re: Deployment Plan.vsd

Due Date: 9/ 9/2011

WP Updated

Impact added by Rubia

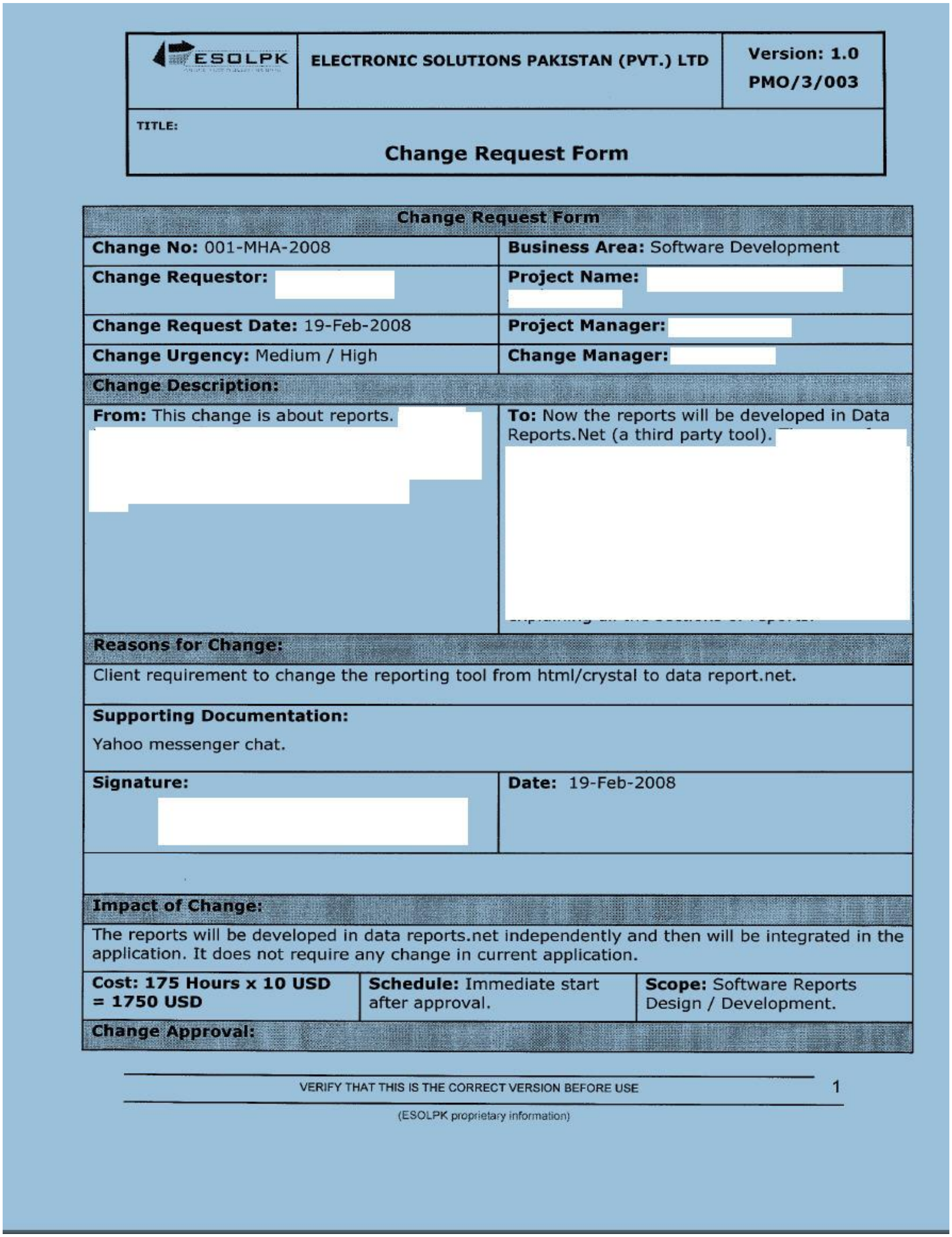
New use cases: 0
Deleted use cases: 0
Partialy updated use cases: 2

HMS-UC-LAB-3.20
HMS-UC-LAB-3.20.1

Completly updated use case: 0

-> No impact on SRS

Figure 13 Change Request Form (IAC)



ESOLPK	ELECTRONIC SOLUTIONS PAKISTAN (PVT.) LTD	**Version: 1.0** **PMO/3/003**
TITLE: **Change Request Form**		

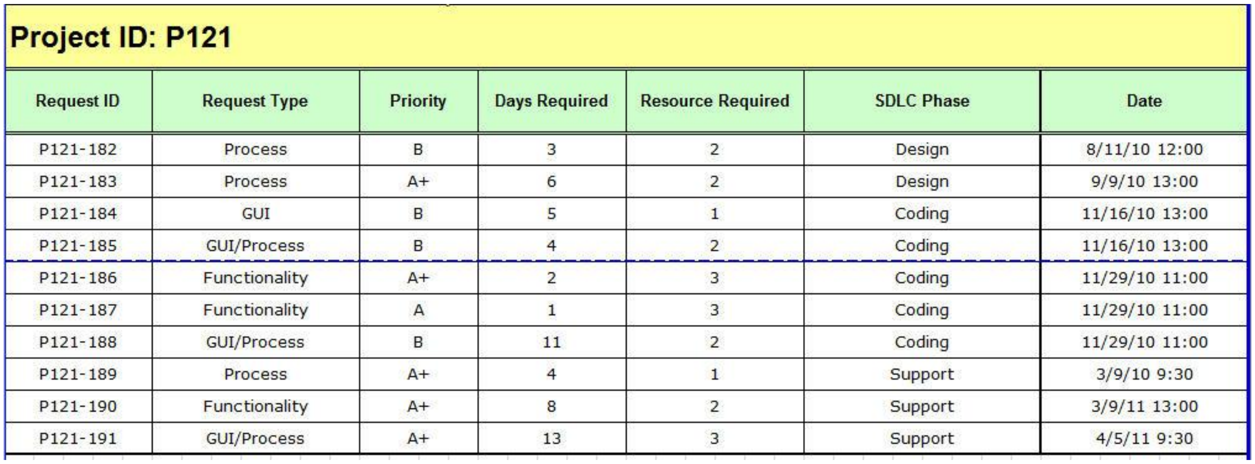
Change Request Form	
Change No: 001-MHA-2008	**Business Area:** Software Development
Change Requestor: [redacted]	**Project Name:** [redacted]
Change Request Date: 19-Feb-2008	**Project Manager:** [redacted]
Change Urgency: Medium / High	**Change Manager:** [redacted]
Change Description:	
From: This change is about reports. [redacted]	**To:** Now the reports will be developed in Data Reports.Net (a third party tool). [redacted]
Reasons for Change:	
Client requirement to change the reporting tool from html/crystal to data report.net.	
Supporting Documentation: Yahoo messenger chat.	
Signature: [redacted]	**Date:** 19-Feb-2008
Impact of Change:	
The reports will be developed in data reports.net independently and then will be integrated in the application. It does not require any change in current application.	

Cost: 175 Hours x 10 USD = 1750 USD	**Schedule:** Immediate start after approval.	**Scope:** Software Reports Design / Development.
Change Approval:		

VERIFY THAT THIS IS THE CORRECT VERSION BEFORE USE 1

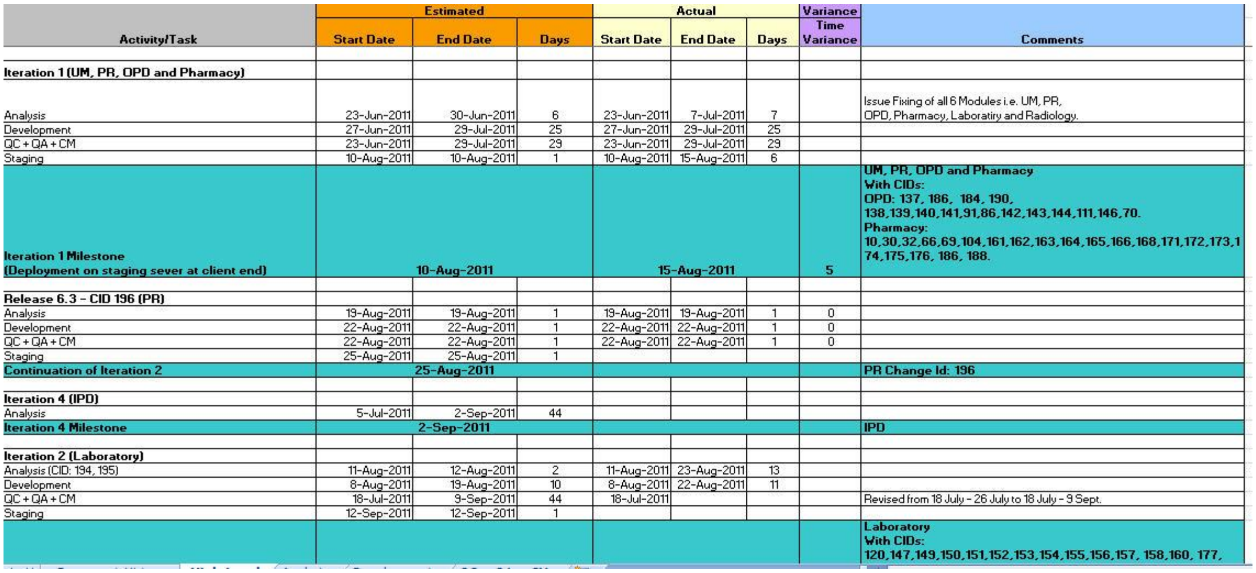
(ESOLPK proprietary information)

Figure 14 Change Request Form (ESOLPK)



Project ID: P121						
Request ID	Request Type	Priority	Days Required	Resource Required	SDLC Phase	Date
P121-182	Process	B	3	2	Design	8/11/10 12:00
P121-183	Process	A+	6	2	Design	9/9/10 13:00
P121-184	GUI	B	5	1	Coding	11/16/10 13:00
P121-185	GUI/Process	B	4	2	Coding	11/16/10 13:00
P121-186	Functionality	A+	2	3	Coding	11/29/10 11:00
P121-187	Functionality	A	1	3	Coding	11/29/10 11:00
P121-188	GUI/Process	B	11	2	Coding	11/29/10 11:00
P121-189	Process	A+	4	1	Support	3/9/10 9:30
P121-190	Functionality	A+	8	2	Support	3/9/11 13:00
P121-191	GUI/Process	A+	13	3	Support	4/5/11 9:30

Figure 15 Change request data (Komatsu)



	Estimated			Actual			Variance	
Activity/Task	Start Date	End Date	Days	Start Date	End Date	Days	Time Variance	Comments
Iteration 1 (UM, PR, OPD and Pharmacy)								
Analysis	23-Jun-2011	30-Jun-2011	6	23-Jun-2011	7-Jul-2011	7		Issue Fixing of all 6 Modules i.e. UM, PR, OPD, Pharmacy, Laboratiry and Radiology.
Development	27-Jun-2011	29-Jul-2011	25	27-Jun-2011	29-Jul-2011	25		
QC + QA + CM	23-Jun-2011	29-Jul-2011	29	23-Jun-2011	29-Jul-2011	29		
Staging	10-Aug-2011	10-Aug-2011	1	10-Aug-2011	15-Aug-2011	6		
Iteration 1 Milestone (Deployment on staging sever at client end)	**10-Aug-2011**			**15-Aug-2011**			**5**	**UM, PR, OPD and Pharmacy With CIDs: OPD: 137, 186, 184, 190, 138,139,140,141,91,86,142,143,144,111,146,70. Pharmacy: 10,30,32,66,69,104,161,162,163,164,165,166,168,171,172,173,174,175,176, 186, 188.**
Release 6.3 - CID 196 (PR)								
Analysis	19-Aug-2011	19-Aug-2011	1	19-Aug-2011	19-Aug-2011	1	0	
Development	22-Aug-2011	22-Aug-2011	1	22-Aug-2011	22-Aug-2011	1	0	
QC + QA + CM	22-Aug-2011	22-Aug-2011	1	22-Aug-2011	22-Aug-2011	1	0	
Staging	25-Aug-2011	25-Aug-2011	1					
Continuation of Iteration 2	**25-Aug-2011**							**PR Change Id: 196**
Iteration 4 (IPD)								
Analysis	5-Jul-2011	2-Sep-2011	44					
Iteration 4 Milestone	**2-Sep-2011**							**IPD**
Iteration 2 (Laboratory)								
Analysis (CID: 194, 195)	11-Aug-2011	12-Aug-2011	2	11-Aug-2011	23-Aug-2011	13		
Development	8-Aug-2011	19-Aug-2011	10	8-Aug-2011	22-Aug-2011	11		
QC + QA + CM	18-Jul-2011	9-Sep-2011	44	18-Jul-2011				Revised from 18 July - 26 July to 18 July - 9 Sept.
Staging	12-Sep-2011	12-Sep-2011	1					
								Laboratory With CIDs: 120,147,149,150,151,152,153,154,155,156,157, 158,160, 177,

Figure 16 Timelines (IAC)

5.3 Data in Analysis Mode-Cost Estimation

For the analysis purpose the extracted data was maintained in data sheets in SPSS software version 17. Following variables were defined in SPSS

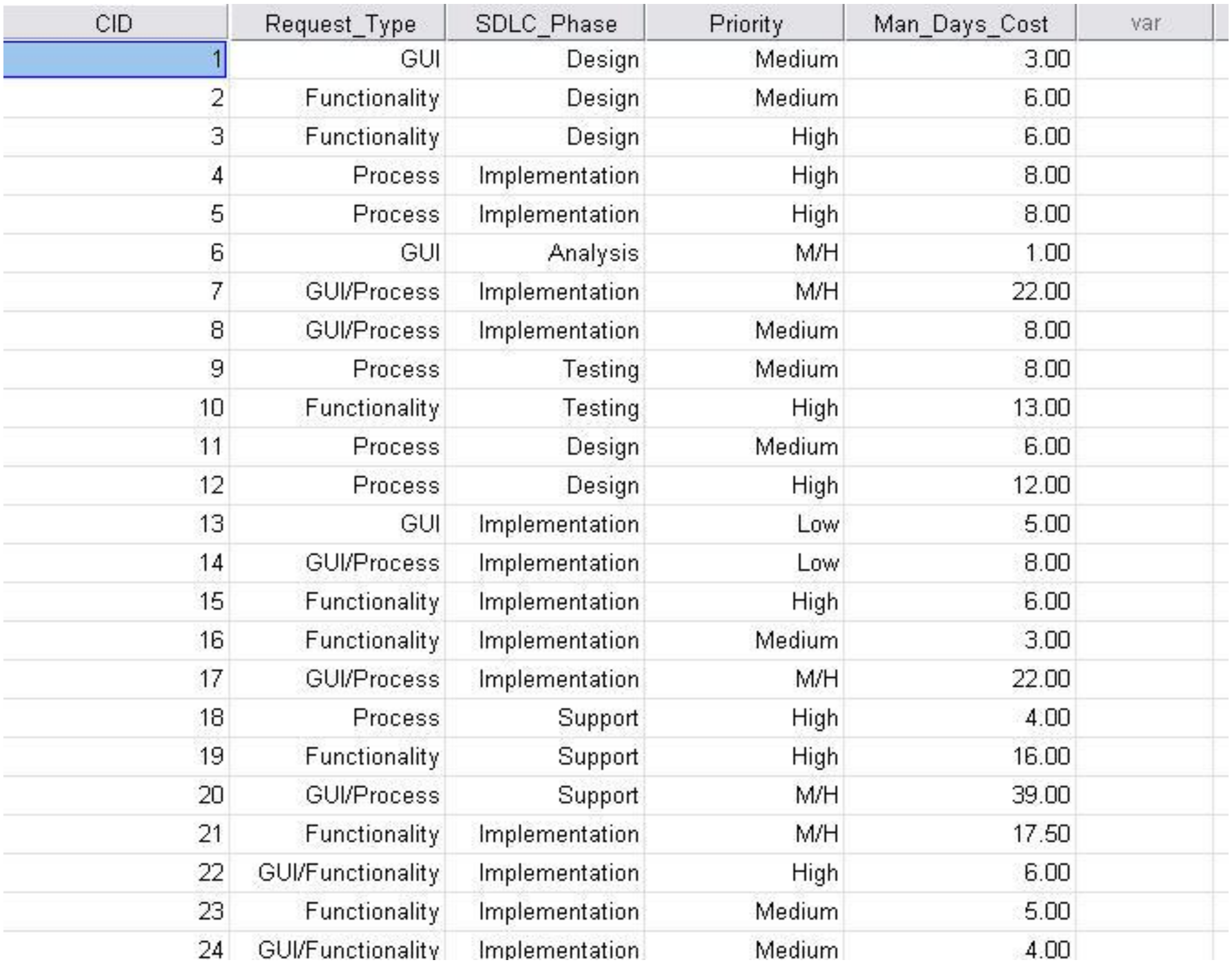
i. CID (Type=Numeric, Measure=Nominal)

ii. Request Type (Type=Numeric, Measure=Nominal)

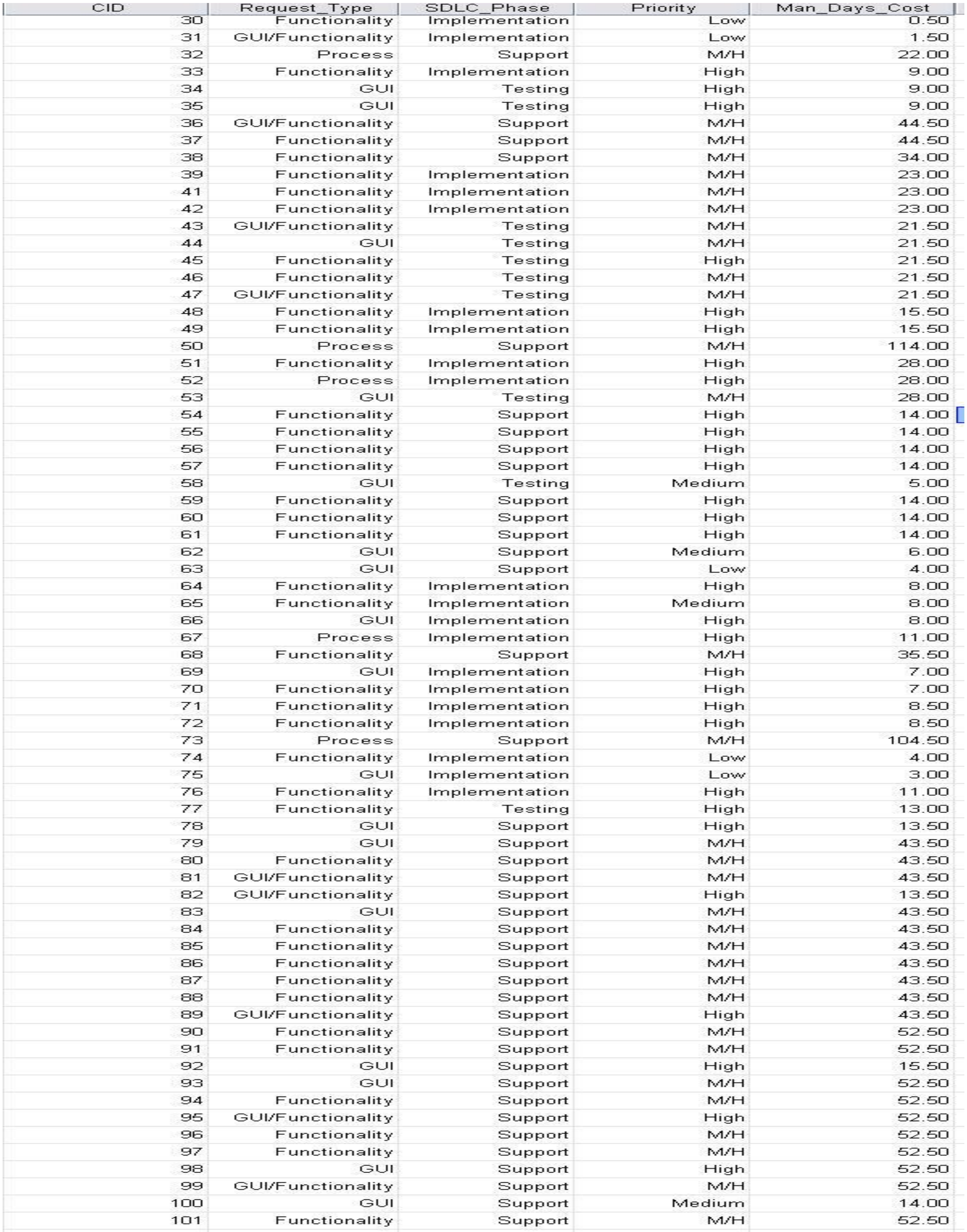
iii. SDLC Phase (Type=Numeric, Measure=Nominal)

iv. Priority (Type=Numeric, Measure=Ordinal)

v. Cost(Man Days) (Type=Numeric, Measure=Scale)



CID	Request_Type	SDLC_Phase	Priority	Man_Days_Cost	var
1	GUI	Design	Medium	3.00	
2	Functionality	Design	Medium	6.00	
3	Functionality	Design	High	6.00	
4	Process	Implementation	High	8.00	
5	Process	Implementation	High	8.00	
6	GUI	Analysis	M/H	1.00	
7	GUI/Process	Implementation	M/H	22.00	
8	GUI/Process	Implementation	Medium	8.00	
9	Process	Testing	Medium	8.00	
10	Functionality	Testing	High	13.00	
11	Process	Design	Medium	6.00	
12	Process	Design	High	12.00	
13	GUI	Implementation	Low	5.00	
14	GUI/Process	Implementation	Low	8.00	
15	Functionality	Implementation	High	6.00	
16	Functionality	Implementation	Medium	3.00	
17	GUI/Process	Implementation	M/H	22.00	
18	Process	Support	High	4.00	
19	Functionality	Support	High	16.00	
20	GUI/Process	Support	M/H	39.00	
21	Functionality	Implementation	M/H	17.50	
22	GUI/Functionality	Implementation	High	6.00	
23	Functionality	Implementation	Medium	5.00	
24	GUI/Functionality	Implementation	Medium	4.00	



CID	Request_Type	SDLC_Phase	Priority	Man_Days_Cost
30	Functionality	Implementation	Low	0.50
31	GUI/Functionality	Implementation	Low	1.50
32	Process	Support	M/H	22.00
33	Functionality	Implementation	High	9.00
34	GUI	Testing	High	9.00
35	GUI	Testing	High	9.00
36	GUI/Functionality	Support	M/H	44.50
37	Functionality	Support	M/H	44.50
38	Functionality	Support	M/H	34.00
39	Functionality	Implementation	M/H	23.00
41	Functionality	Implementation	M/H	23.00
42	Functionality	Implementation	M/H	23.00
43	GUI/Functionality	Testing	M/H	21.50
44	GUI	Testing	M/H	21.50
45	Functionality	Testing	High	21.50
46	Functionality	Testing	M/H	21.50
47	GUI/Functionality	Testing	M/H	21.50
48	Functionality	Implementation	High	15.50
49	Functionality	Implementation	High	15.50
50	Process	Support	M/H	114.00
51	Functionality	Implementation	High	28.00
52	Process	Implementation	High	28.00
53	GUI	Testing	M/H	28.00
54	Functionality	Support	High	14.00
55	Functionality	Support	High	14.00
56	Functionality	Support	High	14.00
57	Functionality	Support	High	14.00
58	GUI	Testing	Medium	5.00
59	Functionality	Support	High	14.00
60	Functionality	Support	High	14.00
61	Functionality	Support	High	14.00
62	GUI	Support	Medium	6.00
63	GUI	Support	Low	4.00
64	Functionality	Implementation	High	8.00
65	Functionality	Implementation	Medium	8.00
66	GUI	Implementation	High	8.00
67	Process	Implementation	High	11.00
68	Functionality	Support	M/H	35.50
69	GUI	Implementation	High	7.00
70	Functionality	Implementation	High	7.00
71	Functionality	Implementation	High	8.50
72	Functionality	Implementation	High	8.50
73	Process	Support	M/H	104.50
74	Functionality	Implementation	Low	4.00
75	GUI	Implementation	Low	3.00
76	Functionality	Implementation	High	11.00
77	Functionality	Testing	High	13.00
78	GUI	Support	High	13.50
79	GUI	Support	M/H	43.50
80	Functionality	Support	M/H	43.50
81	GUI/Functionality	Support	M/H	43.50
82	GUI/Functionality	Support	High	13.50
83	GUI	Support	M/H	43.50
84	Functionality	Support	M/H	43.50
85	Functionality	Support	M/H	43.50
86	Functionality	Support	M/H	43.50
87	Functionality	Support	M/H	43.50
88	Functionality	Support	M/H	43.50
89	GUI/Functionality	Support	High	43.50
90	Functionality	Support	M/H	52.50
91	Functionality	Support	M/H	52.50
92	GUI	Support	High	15.50
93	GUI	Support	M/H	52.50
94	Functionality	Support	M/H	52.50
95	GUI/Functionality	Support	High	52.50
96	Functionality	Support	M/H	52.50
97	Functionality	Support	M/H	52.50
98	GUI	Support	High	52.50
99	GUI/Functionality	Support	M/H	52.50
100	GUI	Support	Medium	14.00
101	Functionality	Support	M/H	52.50

Figure 17 CIDS sheets in SPSS 1-101

5.4 Data in Analysis Mode-Schedule Variance

For the analysis purpose the extracted data was maintained in data sheets in SPSS software version 17. Following variables were defined in SPSS

i. CID (Type=String, Measure=Nominal)

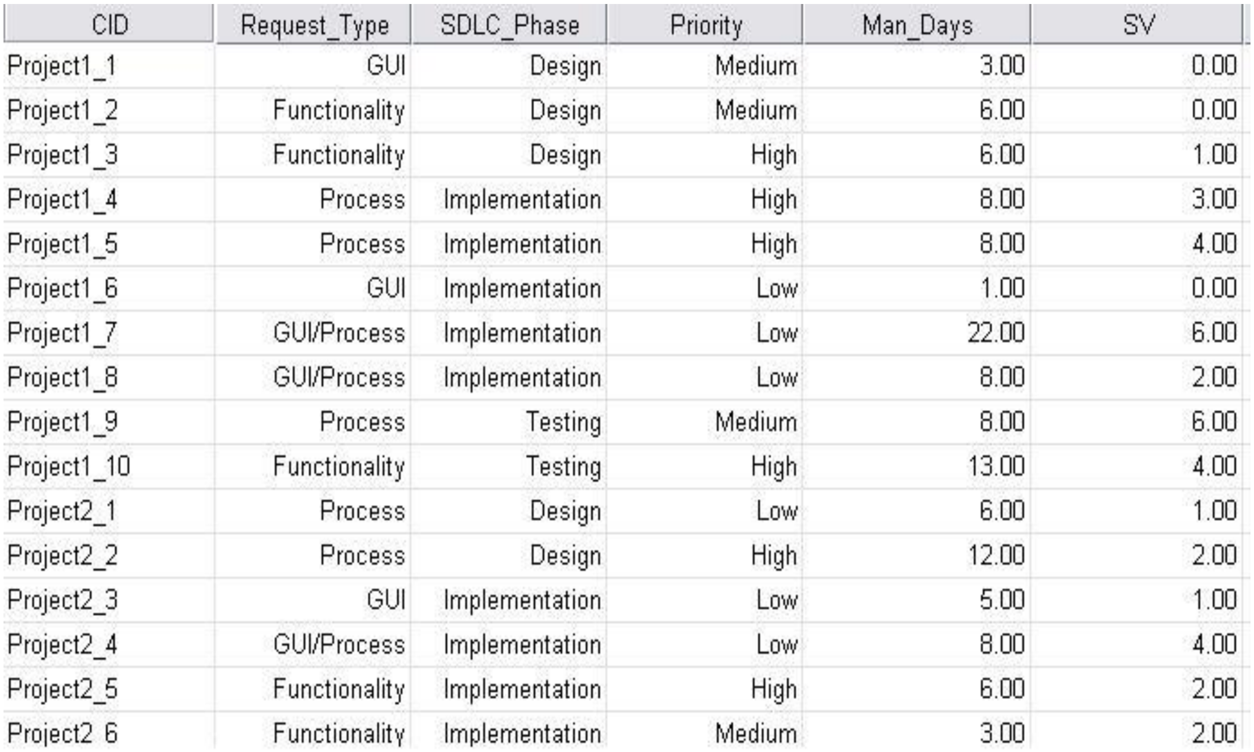
ii. Request Type= (Type=Numeric, Measure=Nominal)

iii. SDLC Phase (Type=Numeric, Measure=Nominal)

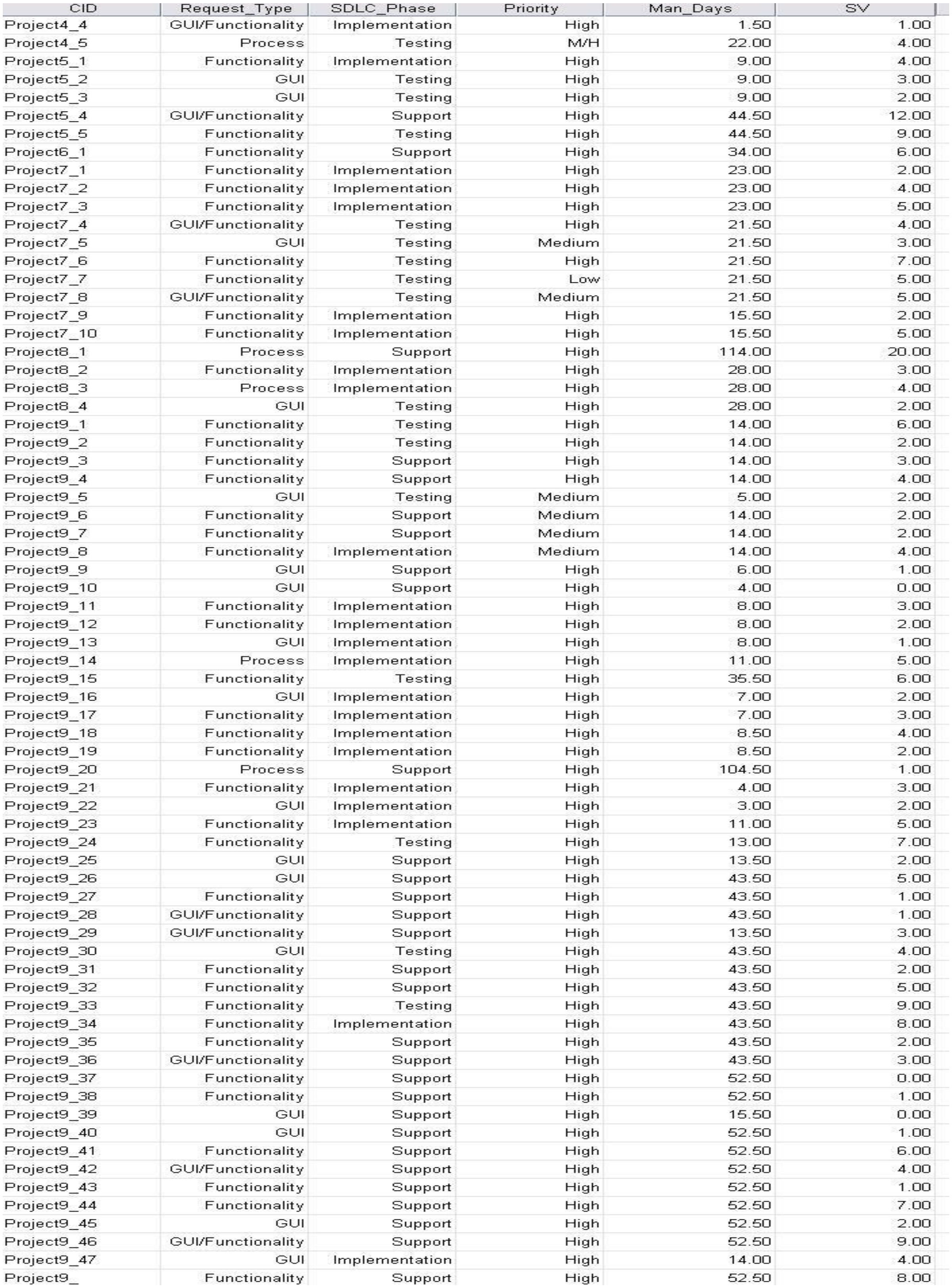
iv. Priority (Type=Numeric, Measure=Ordinal)

v. ManDays (Type=Numeric, Measure=Scale)

vi. SV(Schedule Variance) (Type=Numeric, Measure=Scale)



CID	Request_Type	SDLC_Phase	Priority	Man_Days	SV
Project1_1	GUI	Design	Medium	3.00	0.00
Project1_2	Functionality	Design	Medium	6.00	0.00
Project1_3	Functionality	Design	High	6.00	1.00
Project1_4	Process	Implementation	High	8.00	3.00
Project1_5	Process	Implementation	High	8.00	4.00
Project1_6	GUI	Implementation	Low	1.00	0.00
Project1_7	GUI/Process	Implementation	Low	22.00	6.00
Project1_8	GUI/Process	Implementation	Low	8.00	2.00
Project1_9	Process	Testing	Medium	8.00	6.00
Project1_10	Functionality	Testing	High	13.00	4.00
Project2_1	Process	Design	Low	6.00	1.00
Project2_2	Process	Design	High	12.00	2.00
Project2_3	GUI	Implementation	Low	5.00	1.00
Project2_4	GUI/Process	Implementation	Low	8.00	4.00
Project2_5	Functionality	Implementation	High	6.00	2.00
Project2_6	Functionality	Implementation	Medium	3.00	2.00



CID	Request_Type	SDLC_Phase	Priority	Man_Days	SV
Project4_4	GUI/Functionality	Implementation	High	1.50	1.00
Project4_5	Process	Testing	M/H	22.00	4.00
Project5_1	Functionality	Implementation	High	9.00	4.00
Project5_2	GUI	Testing	High	9.00	3.00
Project5_3	GUI	Testing	High	9.00	2.00
Project5_4	GUI/Functionality	Support	High	44.50	12.00
Project5_5	Functionality	Testing	High	44.50	9.00
Project6_1	Functionality	Support	High	34.00	6.00
Project7_1	Functionality	Implementation	High	23.00	2.00
Project7_2	Functionality	Implementation	High	23.00	4.00
Project7_3	Functionality	Implementation	High	23.00	5.00
Project7_4	GUI/Functionality	Testing	High	21.50	4.00
Project7_5	GUI	Testing	Medium	21.50	3.00
Project7_6	Functionality	Testing	High	21.50	7.00
Project7_7	Functionality	Testing	Low	21.50	5.00
Project7_8	GUI/Functionality	Testing	Medium	21.50	5.00
Project7_9	Functionality	Implementation	High	15.50	2.00
Project7_10	Functionality	Implementation	High	15.50	5.00
Project8_1	Process	Support	High	114.00	20.00
Project8_2	Functionality	Implementation	High	28.00	3.00
Project8_3	Process	Implementation	High	28.00	4.00
Project8_4	GUI	Testing	High	28.00	2.00
Project9_1	Functionality	Testing	High	14.00	6.00
Project9_2	Functionality	Testing	High	14.00	2.00
Project9_3	Functionality	Support	High	14.00	3.00
Project9_4	Functionality	Support	High	14.00	4.00
Project9_5	GUI	Testing	Medium	5.00	2.00
Project9_6	Functionality	Support	Medium	14.00	2.00
Project9_7	Functionality	Support	Medium	14.00	2.00
Project9_8	Functionality	Implementation	Medium	14.00	4.00
Project9_9	GUI	Support	High	6.00	1.00
Project9_10	GUI	Support	High	4.00	0.00
Project9_11	Functionality	Implementation	High	8.00	3.00
Project9_12	Functionality	Implementation	High	8.00	2.00
Project9_13	GUI	Implementation	High	8.00	1.00
Project9_14	Process	Implementation	High	11.00	5.00
Project9_15	Functionality	Testing	High	35.50	6.00
Project9_16	GUI	Implementation	High	7.00	2.00
Project9_17	Functionality	Implementation	High	7.00	3.00
Project9_18	Functionality	Implementation	High	8.50	4.00
Project9_19	Functionality	Implementation	High	8.50	2.00
Project9_20	Process	Support	High	104.50	1.00
Project9_21	Functionality	Implementation	High	4.00	3.00
Project9_22	GUI	Implementation	High	3.00	2.00
Project9_23	Functionality	Implementation	High	11.00	5.00
Project9_24	Functionality	Testing	High	13.00	7.00
Project9_25	GUI	Support	High	13.50	2.00
Project9_26	GUI	Support	High	43.50	5.00
Project9_27	Functionality	Support	High	43.50	1.00
Project9_28	GUI/Functionality	Support	High	43.50	1.00
Project9_29	GUI/Functionality	Support	High	13.50	3.00
Project9_30	GUI	Testing	High	43.50	4.00
Project9_31	Functionality	Support	High	43.50	2.00
Project9_32	Functionality	Support	High	43.50	5.00
Project9_33	Functionality	Testing	High	43.50	9.00
Project9_34	Functionality	Implementation	High	43.50	8.00
Project9_35	Functionality	Support	High	43.50	2.00
Project9_36	GUI/Functionality	Support	High	43.50	3.00
Project9_37	Functionality	Support	High	52.50	0.00
Project9_38	Functionality	Support	High	52.50	1.00
Project9_39	GUI	Support	High	15.50	0.00
Project9_40	GUI	Support	High	52.50	1.00
Project9_41	Functionality	Support	High	52.50	6.00
Project9_42	GUI/Functionality	Support	High	52.50	4.00
Project9_43	Functionality	Support	High	52.50	1.00
Project9_44	Functionality	Support	High	52.50	7.00
Project9_45	GUI	Support	High	52.50	2.00
Project9_46	GUI/Functionality	Support	High	52.50	9.00
Project9_47	GUI	Implementation	High	14.00	4.00
Project9_	Functionality	Support	High	52.50	8.00

Figure 18 Data sheets in SPSS

Chapter 6

ANALYSIS AND RESULTS

6.1 INTRODUCTION

Data extracted during qualitative analysis phase and from change request forms is collectively analyzed to evaluate our hypothesis. For this purpose correlation and regression analysis is performed on the data set. In this chapter we will discus these analysis techniques and the results in detail.

6.2 COST ESTIMATION

Effort associated with change is estimated as prediction of working hours and number of resources is needed to perform a particular task [21].

6.2.1 Correlation analysis and results

To show relationship among different variables correlation statistical technique is used. To show the relation among the variables of our cost estimation model, correlation analysis is performed on the selected data set. The statistics of table 2 shows that coefficient of correlation between SDLC Phase and Cost is .604 and this correlation is significant at 0.01 level (1-tailed). This positive relation indicates that if more changes are requested in the later phases of software development life cycle then more effort would be required to implement that change (figure 19).

		Request_ Type	SDLC_ Phase	Priority	Cost(ManDays)
Request_ Type	Pearson Correlation	1	-.048	.105	.121
	Sig. (1-tailed)		.318	.150	.116
	N	100	100	100	100
SDLC_ Phase	Pearson Correlation	-.048	1	.388**	.604**
	Sig. (1-tailed)	.318		.000	.000
	N	100	100	100	100
Priority	Pearson Correlation	.105	.388**	1	.620**
	Sig. (1-tailed)	.150	.000		.000
	N	100	100	100	100
Man_ Days_ Cost	Pearson Correlation	.121	.604**	.620**	1
	Sig. (1-tailed)	.116	.000	.000	
	N	100	100	100	100

**. Correlation is significant at the 0.01 level (1-tailed).

Table 2 Correlation Matrix - Significant factors for the Schedule variance estimation model

This is because if change is proposed in later phase e.g. in Maintenance phase its effect would be propagate to the earlier phases which are, Analysis, Design, Implementation and Testing. As more rework would be required so it would have great impact on the cost associated with change

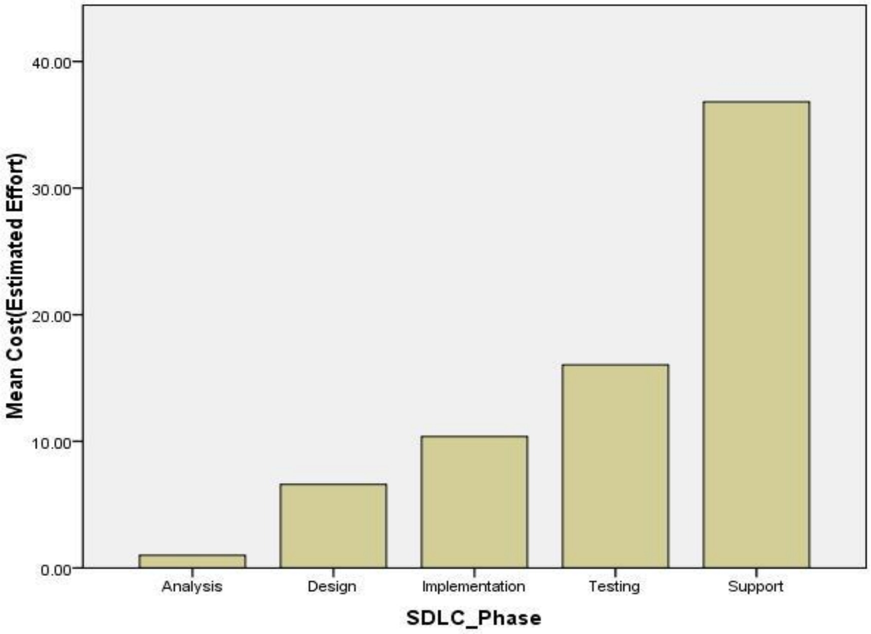


Figure 19 SDLC_ Phase and Cost of associated change

The coefficient of correlation between Priority and Cost is .620.This correlation is significant at 0.01 (1-tailed).These results indicate that there is positive relation between Cost and priority of change i.e. a high priority change in requirement results as intensive rework as compare to the low priority change (figure 20).

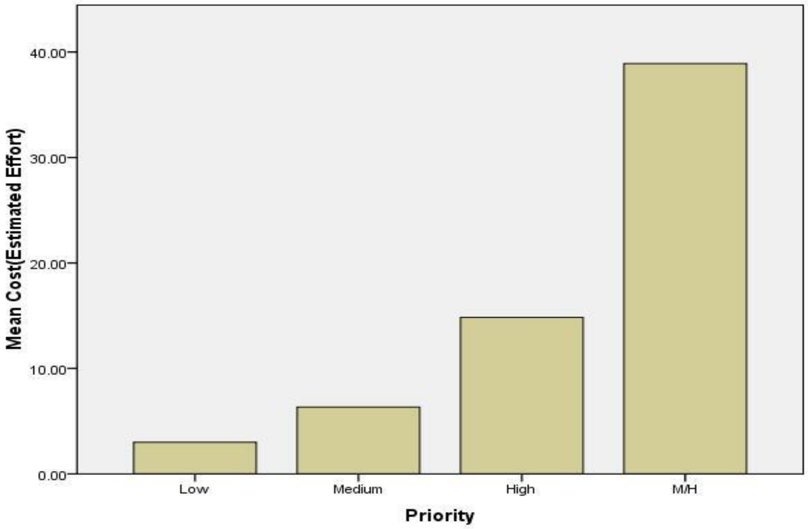


Figure 20 Priority of Change and Cost of associated Change

The coefficient of correlation between Change type and cost is 0.121.This weak positive correlation shows that up to some extent change type would affect the effort required to implement the change (figure 21). This is because it depends during which phase Of SDLC change was proposed. If GUI related change occurs in Design phase few work be required to make modifications in the related artifacts of analysis and design phase. If same change occurs in testing phase more effort would be required because artifacts related to testing phase plus artifacts of the prior phases would need modification.

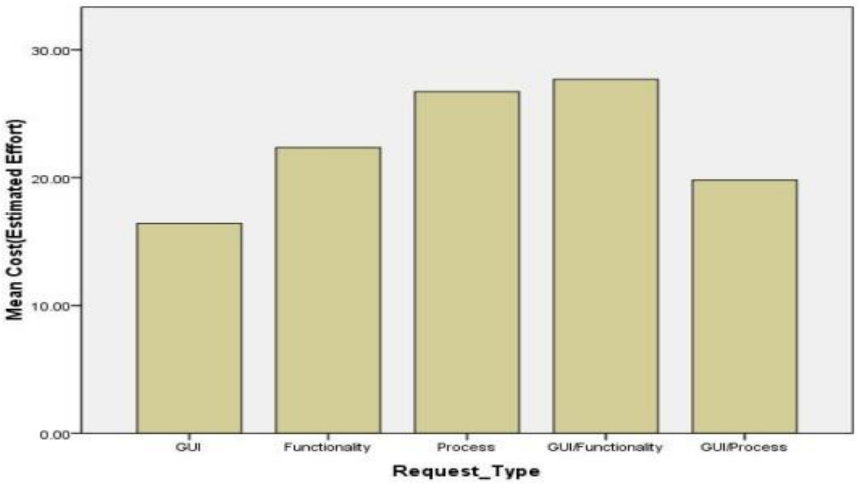


Figure 21 Request Type of Change and Cost of associated Change

6.2.2 Regression analysis and results

With the regression analysis the value of dependent variable can be predict from one or more independent variables In our research dependent variable is cost and the independent variables are SDLC phase, change request type and change priority. To find out the value of effort regression analysis is performed on the selected data set.

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics				
					R Square Change	F Change	df1	df2	Sig. F Change
1	.740[a]	.548	.534	14.16648	.548	38.817	3	96	.000

a. Predictors: (Constant), Request_ Type, SDLC_ Phase , Priority

Table 3Model Summary

Table 3 analysis shows that Adjusted R-Square (Coefficient of determination) is 0.534 and Sig F. change is .000.Which indicates that relationship of Cost in terms of development effort is significant with independent variables i.e. Change Priority, SDLC phase and Change type at

99.99 confidence interval. This indicates that 53.4 % variance in project cost is because of these three independent variables (Change Priority, SDLC phase and Change Type).

Model		Sum of Squares	Df	Mean Square	F	Sig.
1	Regression	23370.265	3	7790.088	38.817	.000[a]
	Residual	19266.163	96	200.689		
	Total	42636.428	99			

a. Predictors: (Constant), Request_ Type, SDLC_ Phase , Priority

b. Dependent Variable: Man_ Days_ Cost

Table 4 ANOVA

Analysis of variance of this model shows that independent variables are significantly contributing in the value of dependent variable. The values df, Mean Square, F of ANOVA table are calculated according to following formulas.

Df=Number of variables-1

Mean Square=Sum of squares/ df.

F= Mean Square of Regression/ Mean Square of Residual

		Unstandardized Coefficients		Standardized Coefficients			Collinearity Statistics	
Model		B	Std. Error	Beta	t	Sig.	Tolerance	VIF
1	(Constant)	-45.967	6.841		-6.720	.000		
	SDLC_ Phase	8.605	1.471	.437	5.850	.000	.842	1.188
	Priority	9.784	1.671	.440	5.856	.000	.834	1.199
	Request_ Type	1.844	1.337	.096	1.379	.171	.980	1.021

a. Dependent Variable: Man_ Days_ Cost

Table 5Coefficients

Model	Dimen sion	Eigenvalue	Condition Index	Variance Proportions			
				(Constant)	SDLC_Phase	Priority	Request_Type
1	1	3.768	1.000	.00	.00	.00	.01
	2	.154	4.939	.01	.06	.04	.84
	3	.048	8.872	.08	.31	.94	.00
	4	.030	11.240	.92	.63	.01	.15

a. Dependent Variable: Man_ Days_ Cost

Table 6 Collinearity Diagnostics

6.2.2.1 Regression Equation

Values of column B of Table 5 are used to derive regression equation. The derived equation is a generic regression equation for the computation of cost against a change. This equation is based on the model depicted in figure 11.

Cost= -45.967+8.605SDLC_Phase+9.784Priority+1.844Request_Type

Where

Cost is the dependent variable and is computed in terms of effort against a change.

Effort=Number of Resources* Number of working hours.

6.3 Schedule Variance

Variance in schedule which occurs due to change in requirements is estimated in terms of calendar days.

6.3.1 Correlation Analysis and Results.

To show the relation among the variables of our proposed schedule variance estimation model, correlation analysis is performed on the selected data set.

The statistics of table 7 shows that coefficient of correlation between Request type and SV (Schedule Variance) is .337.This correlation is significant at 0.01 level (1-tailed). These results indicate that there is a positive relation between the nature of change and schedule variance i.e. a dramatic change in requirement results as intensive rework as compare to the minute change in requirements. More complex change types are potential candidates to increase the project scope.

		Request_ Type	SDLC_ Phase	Priority	Man_ Days	SV
Request_ Type	Pearson Correlation	1	-.048	-.134	.121	.337**
	Sig. (1-tailed)		.317	.093	.116	.000
	N	100	100	100	100	100
SDLC_ Phase	Pearson Correlation	-.048	1	.142	.599**	.156
	Sig. (1-tailed)	.317		.080	.000	.061
	N	100	100	100	100	100
Priority	Pearson Correlation	-.134	.142	1	.132	.110
	Sig. (1-tailed)	.093	.080		.096	.138
	N	100	100	100	100	100
Man_ Days	Pearson Correlation	.121	.599**	.132	1	.455**
	Sig. (1-tailed)	.116	.000	.096		.000
	N	100	100	100	100	100
SV	Pearson Correlation	.337**	.156	.110	.455**	1
	Sig. (1-tailed)	.000	.061	.138	.000	
	N	100	100	100	100	100

**. Correlation is significant at the 0.01 level (1-tailed).

Table 7Correlation Matrix - Significant factors for the Schedule variance estimation model

These results indicate that there is a positive relation between the nature of change and schedule variance i.e. a dramatic change in requirement results as intensive rework as compare to the minute change in requirements. More complex change types are potential candidates to increase the project scope. The increment in project scope requires more effort and time to complete the project due to which the projects usually run out of time. The coefficient of correlation between SV and Priority is .110. This weak positive correlation shows that up to some extent change Priority would affect the variance in schedule. The coefficient of correlation between SDLC Phase and SV is .156 which is weak positive correlation. The coefficient of correlation between SV and Priority is .110. This weak positive correlation shows that up to some extent change Priority would affect the variance in schedule. The coefficient of correlation between SDLC Phase and SV is .156 which is weak positive correlation.

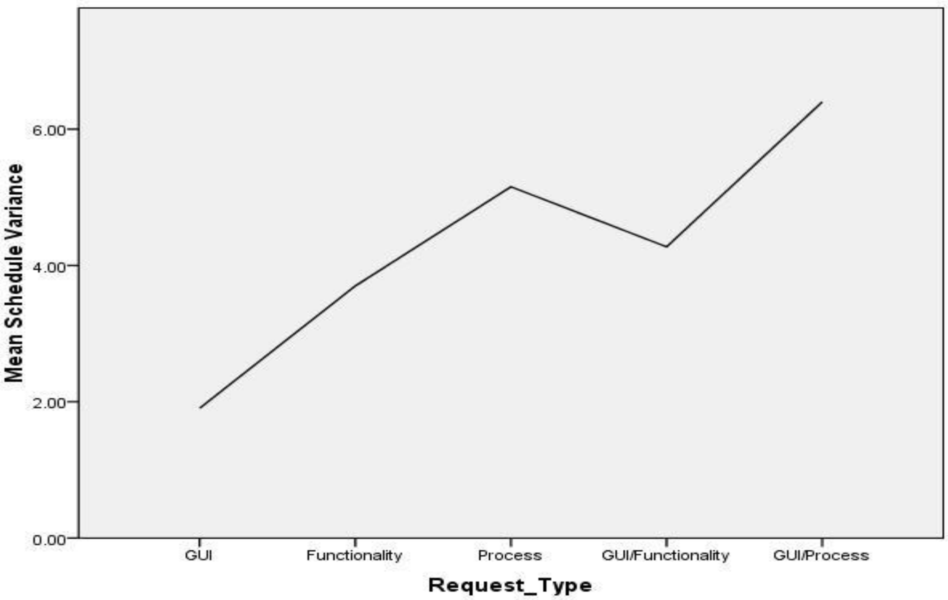


Figure 22 Request_ Type and Schedule Variance

The coefficient of correlation between Man Days and SV is .455 and this correlation is significant at 0.01 level (1-tailed).This positive relation indicates that Man Days significantly contribute towards the estimation of dependent variable i.e. SV .Man days represent the working hours * No of resources. Continuous changes in requirements directly affect the size of anticipated project because a change results as additional tasks that are needed to perform by the project development team or enhancement in project scope. To accommodate change requests more time in the context of Man Days is taken by development cycle. This is because some changes modify the existing work and some changes enhance the existing functionality of the software. Due to this enhanced functionally and rework more development time in terms of working hours is required to accommodate this proposed change. This greater effort leads to longer project duration and due to this longer duration delay is expected in the estimated schedule.

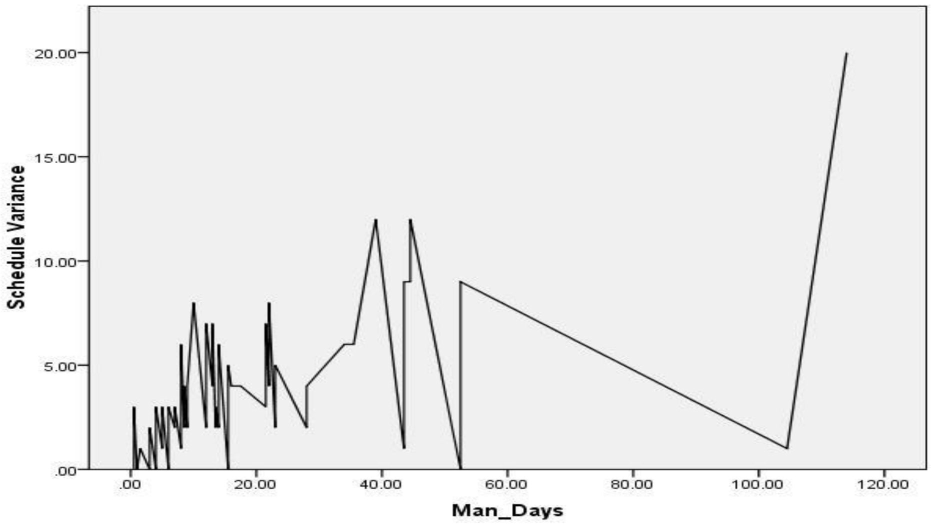


Figure 23 Effort (Man Days) and Schedule Variance

6.3.2 Regression analysis and results

In our research dependent variable is schedule variance and the independent variables are SDLC phase, change request type and change priority, effort (Man Days). To find out the value of schedule variance regression analysis is performed on the selected data set.

					Change Statistics				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	R Square Change	F Change	df1	df2	Sig. F Change
1	.556[a]	.309	.280	2.58738	.309	10.628	4	95	.000

a. Predictors: (Constant), Man_ Days, Request_ Type, Priority, SDLC_ Phase

Table 8 Model Summary

Statistical analysis of Table 8 shows that Adjusted R-Square (Coefficient of determination) is 0.280 and Sig F. change is .000.Which indicates that relationship of dependent variable schedule variance is significant with independent variables Change Priority, SDLC phase and Change_ Type and Effort (Man Days) at 99.99 confidence interval. This indicates that 28 % variance in project schedule is because of these four independent variables (Change Priority, SDLC phase and Change Type, Man Days).

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	284.607	4	71.152	10.628	.000[a]
	Residual	635.983	95	6.695		
	Total	920.590	99			

a. Predictors: (Constant), Man_ Days, Request_ Type, Priority, SDLC_ Phase

Table 9 ANOVA

Analysis of variance of this model shows that independent variables are significantly contributing in the value of dependent variable. The values df, Mean Square, F of ANOVA table are calculated according to following formulas.

Df=Number of variables-1

Mean Square=Sum of squares/ df.

F= Mean Square of Regression/ Mean Square of Residual

		Unstandardized Coefficients		Standardized Coefficients		
Model		B	Std. Error	Beta	t	Sig.
1	(Constant)	.661	1.712		.386	.700
	Request_Type	.807	.249	.285	3.242	.002
	SDLC_Phase	-.444	.344	-.139	-1.292	.200
	Priority	.457	.387	.103	1.181	.241
	Man_Days	.072	.016	.491	4.514	.000

Table 10 Coefficients

6.3.2.1 Regression Equation

Values of column B of Tab.10 are used to derive regression equation. The derived equation is a generic regression equation for the computation of variance in schedule that due to change implementation. This equation is based on the model depicted in figure 3.

Schedule Variance =.661+.807Request_Type-.444SDLC_Phase+.457Priority+.072Man_ Days

Where

Schedule Variance is the dependent variable and is computed in terms of calendar days.

Man Days=Number of Resources* Number of working hours.

6.4 SUMMARY

In this chapter we have discussed the data analysis techniques which are performed on the selected datasets and the findings are reported.

Chapter 7

CONCLUSION AND FUTURE WORK

In this chapter we conclude the thesis, summarize the work presented so far and provide directions for future work.

7.1 CONCLUSION

In this research work we have empirically investigated the impact of changing requirements and its associated cost from 9 different software projects. Our research has addressed following research question.

RQ1. How changing requirements affect the cost and schedule of software?

RQ2. How changing requirements affect the cost and schedule of software?

RQ3. When change request arises in any phase of SDLC, which work products are affected from it?

RQ4. How the effort is measured to implement a proposed change in different phases of SDLC?

The results help to understand the impact of a particular change and the associated cost to implement that change .This study has also identified major change attributes which significantly contribute in cost estimation of change .The momentous relation of change Priority and Cost shows that because of high priority of change more effort more working hours are required to implement that change. The significant relationship of SDLC phase and Cost indicates that if changes are proposed in later phases of SDLC then more rework is required to implement that change. Our Research also suggests a cost estimation model and a generic regression equation to compute the associated cost of a change.

In this work we have also tried to proof that the change in requirements has a strong impact on the estimated time duration of the anticipated project. This area of research has addressed following research questions.

RQ1. How the change in requirement affects the Project Scope?

A change in software requirement doesn’t always affect project scope. Nature of change and frequency of change are the two important factors that can become the cause of scope creep.

RQ2.How the change in requirement affects the timelines of Project? (E.g. Timelines of different phases of SDLC i.e. Analysis, Design, Implementation, Testing, Maintenance)

Due to enhanced functionally and rework more development time in terms of working hours is required to accommodate the proposed change. This increase the project size as more tasks are added in the already defined tasks which in turn results to greater effort and greater effort leads to longer project duration and due to this longer duration delay is expected in the estimated timelines of the project.

This research study helps to understand the impact of a particular change on the estimated schedule of the software project and highlights the potential factors for the estimation of schedule variance.

7.2 FUTURE WORK

This research work can be extended by considering more change request attributes that can be the potential factors for cost and schedule variance estimation of anticipated software project against change in requirements. Further detailed analysis of data can be performed by increasing the sample size of the data.

Appendix A

SNAPSHOTS

A.1 GUI of IAnalyzer

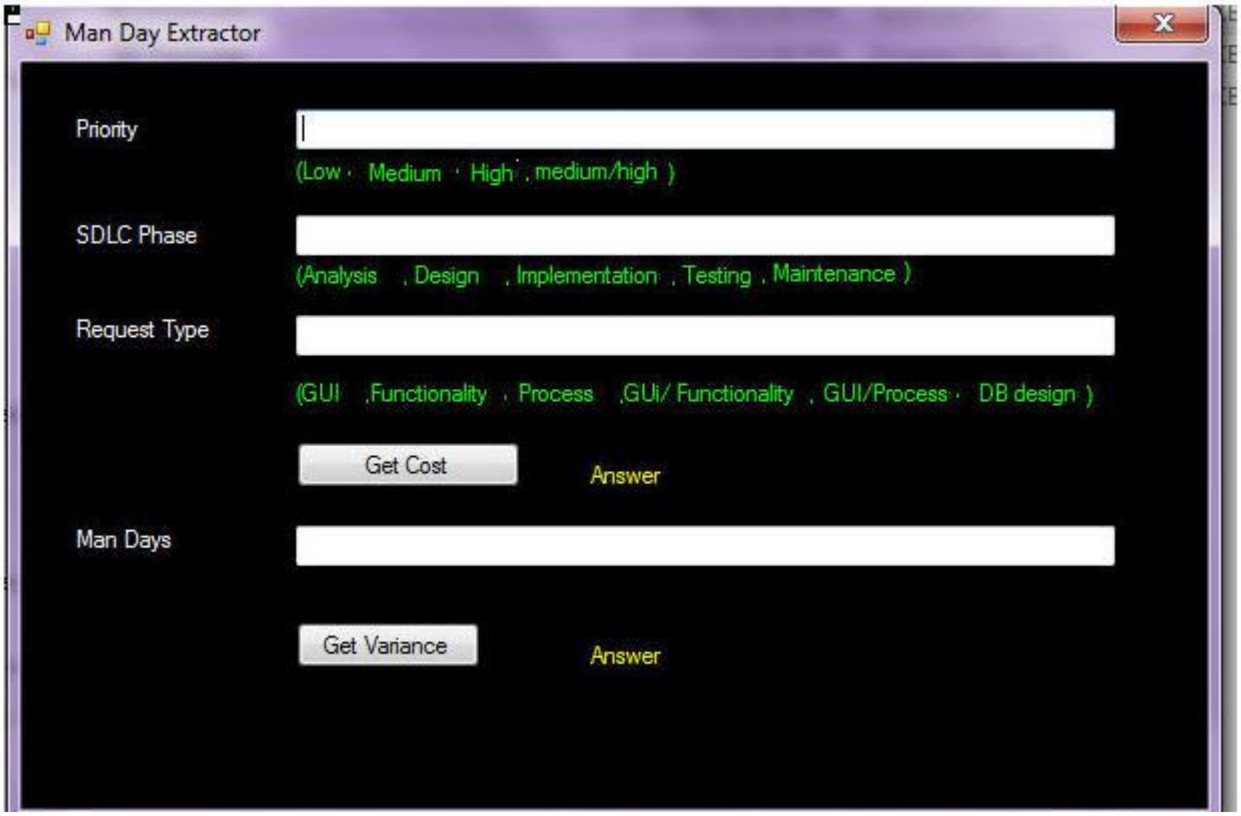
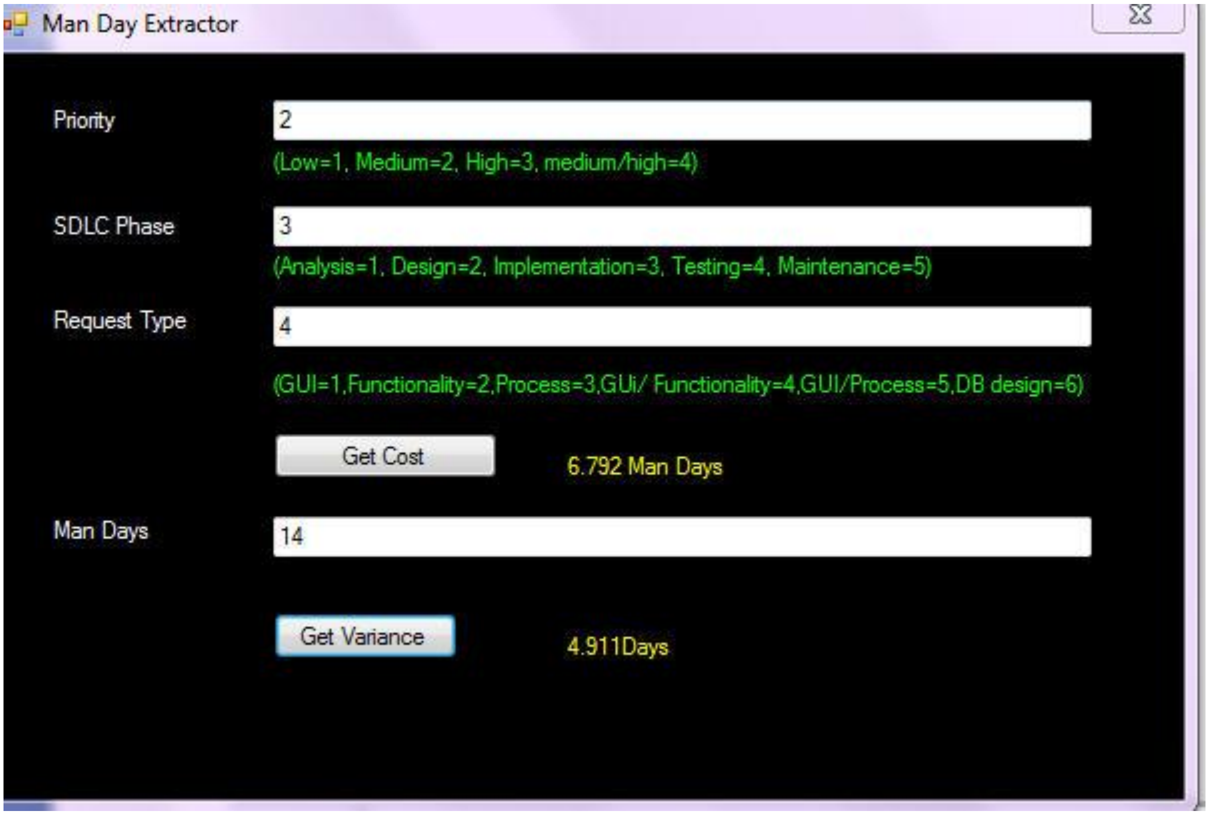

Man Day Extractor
Priority
(Low · Medium · High , medium/high)
SDLC Phase
(Analysis , Design , Implementation , Testing , Maintenance)
Request Type
(GUI ,Functionality , Process ,GUi/ Functionality , GUI/Process · DB design)
Get Cost
Answer
Man Days
Get Variance
Answer

Figure 24 GUI of IAnalyzer

A.2 Input and Results



Priority: 2

(Low=1, Medium=2, High=3, medium/high=4)

SDLC Phase: 3

(Analysis=1, Design=2, Implementation=3, Testing=4, Maintenance=5)

Request Type: 4

(GUI=1,Functionality=2,Process=3,GUi/ Functionality=4,GUI/Process=5,DB design=6)

6.792 Man Days

Man Days: 14

4.911Days

Figure 25 Input and Results

Appendix B

INTERVIEW TRANSCIPTS

DISCUSSION

During the interview session, experts gave the following answers.

Question-1

Generally while planning for a project margin for [expected] changes is added, depending upon the nature of work. Change in requirements affects cost and schedule if they go beyond expected effort. Then, there could be needs to revise/update plans and/or review for alternates like add/drop features from scope to accommodate necessary changes without changing project budget or time. It all depends on individual project [Manager].

Change in requirements can affect the existing schedule and overall calculated cost of project. This is because to accommodate change activities are added and each activity occupy some duration which can affect existing timelines. To perform each activity resources are allocated and each resource has some cost which adds up to existing cost of the project [Developer].

If estimated timelines are increased due to change in requirements then the cost would also be increased because cost is measured in effort and effort is equal to Human Resource * time [Test Engineer].

Question-2

Impact of a change is calculated using horizontal traceability, where, requirement and its solution are already traceable through matrices maintained between work items. So, impact analysis of a change can be performed by determining the effect of a change request to solution (Requirements Traceability) and further to its related work items (DTM and QTM traceability matrices) [Manager].

i. Analysis phase: Number of new use cases, number of completely changed use cases, Number of partially changed use cases and weight of change [Developer].

ii. Development Phase: Quantum of Change, Size, Components to be change

iii. Testing Phase: Nature of change, Complexity, Efficiency

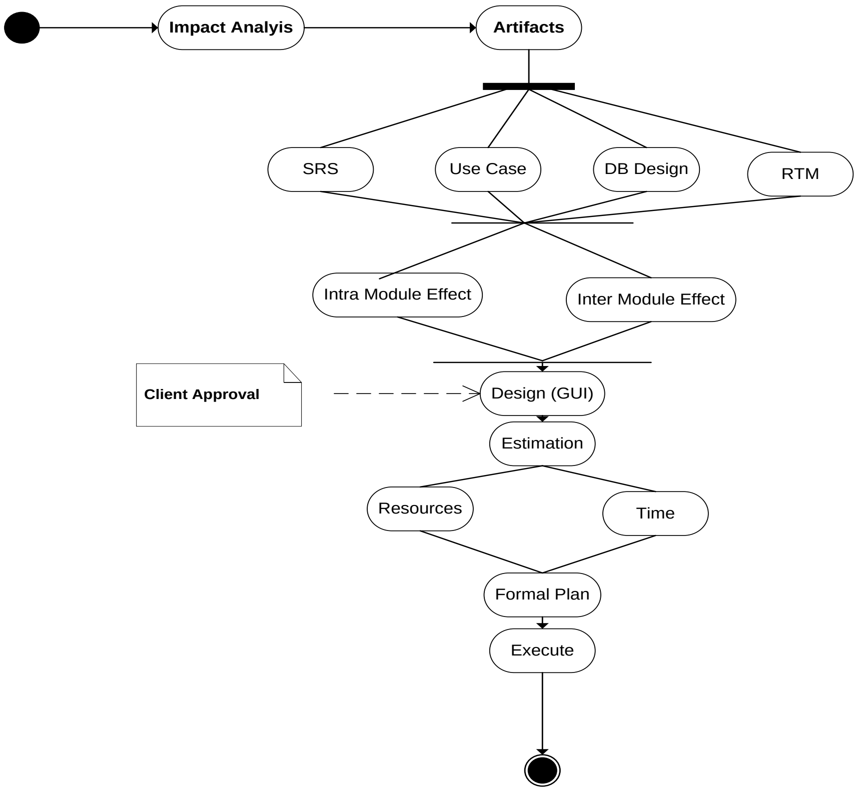


Figure 26 Change analysis process

Question-3

Assuming we are talking about rework when software is already developed with all its artifacts, any change (from mentioned above) could lead to extensive rework (in all application layers). If there is only change in UI, it would be minimal as compared to the other three, which are interrelated and rarely require little work (as they impact in all application layers) [Manager].

Workflow/Process can he need extensive rework then DB and GUI related change [Developer].

Question-4

Only Requirement Specification documents require modifications. It could be SRS, Prototype, Issue log or some UML or other notation representing requirement [Manager].

Artifacts related to analysis like Project plans, Task sheets, UC. SRS, RTMs, ERD are required to update [Developer].

Question-5

In design phase, design artifact is also built on proposed solution against requirements. So, a change at this phase would require updations in Requirement specifications, solution specifications and design documents [Manager].

All the design documents impacted by change must also be updated [Developer].

Question-6

In implementation phase, application is also built after solution and design against requirements. So, a change at this phase would require updations in Requirement specifications, solution specifications, and design documents as well as in application code of software [Manager].

All the artifacts impacted by change related to implementation phase like Application code, release notes, DTM plus artifacts of the prior phases (Analysis an Design) must also be updated [Developer].

Question-7

In testing phase, application is built with all its artifacts after analysis, design and development phases. So, a change at this phase would require updations in Requirement specifications, solution specifications, design documents, application code and its test cases [Manager].

All the artifacts developed during testing phase like Test Cases, Test reports, Sanity test documents, QTM plus all the artifacts of prior phases [Developer].

Question-8

In Maintenance phase, application is built with all its artifacts after analysis, design, development and testing phases. So, a change at this phase would require updations in Requirement specifications, solution specifications, design documents, application code, test cases and its manuals [Manager].

User manuals, External release notes plus all the artifacts of previous phases [Developer].

Question-9

A change during Requirement analysis phase has no chain effect. So, its effort is measured directly through requirement specification document’s expected change [Manager].

On the basis of expert judgment and number of target artifacts that would be affected by change [Developer].

Question-10

In Design phase, RTM - Requirement Traceability Matrix - and DB/design interaction (a mapping document from proposed solution to its design) are used to find out size of work (effort) required against a change [Manager].

Question-11

In Implementation phase, RTM, DB/Design interaction (matrix) and DTM –Development Traceability Matrix are used to find out size of work (effort) required against a change [Manager].

Question-12

In Testing phase, RTM, DB/Design interaction (matrix), DTM and QTM – QC (Quality Control) Traceability Matrix are used to find out size of work (effort) required against a change [Manager].

Number of test scenarios are identified against change .As number of test scenarios be increased the more man days be required to execute the whole testing activity [Test Engineer].

Question-13

In Maintenance phase, RTM, DB/Design interaction (matrix), DTM and QTM are used to find out size of work (effort) required against a change.   Also, effort to review and update Software Manuals of the module/component is added in it [Manager].

Question-14

A change in software doesn’t always affect project scope. If a change is determined not to be part of project scope and is inevitable then Project Scope document, project planning and related documents are updated. In Some cases it requires some legal process to establish consensus on updated Scope. Also, communication of the update to all stakeholders needs to be carried out [Manager]

Continuous change has the potential to increase the scope of project [Developer].

To find out the impact of requirement change on software scope three factors should be consider i.e. RE process Approach, Nature of client, and Change nature. If the RE development process is using Top Down approach then there is more probability, change in requirements can affect scope of project. If Bottom up approach is used them there is less chances of change in scope of project. If a change occurs in the policies of the client then scope can be affected [System Analyst].

Question-15

During change process development team calculates its impact for decision support. After approval for development it requires planning for development. Development plan is revised and may require additional effort for it to develop [Manager].

Planning for timelines is done on the basis of initial estimates when change occurs it adds up some duration to the initial estimates so with few changes estimated duration can be double [Developer].

SDLC Schedule Variance=Variance of Analysis phase + Variance of Development phase + variance of Testing phase+ variance of Maintenance phase [System Analyst].

REFRENCES

[1] Muhammad Wasim Bhatti, Nadeem Ehsan, "An Investigation of changing requirements with respect to development phases of a software project", 2010 International Conference on Computer Information Systems and Industrial Management Applications, pp 323-327.

[2] F.P., Jr. Brooks, "No Silver Bullet Essence and Accidents of Software Engineering", Computer, vol. 20, no. 4, Apr. 1987, pp. 10-19.

[3] M.R.Strens and R.C.Sudgen , "Change Analysis:A step towards the Challenge of changing Requirements", Engineering of computer based systems 1996 proceedings, IEEE symposium and workshop, pp 278-283.

[4] James S. O'Neal, Doris L. Carver, "Analyzing the Impact of Changing Requirements," ICSM, 17th IEEE International Conference on Software Maintenance (ICSM'01), 2001, pp.190.

[5] N Nurmuliani, Didar zowgh, Williams, S.P, "Requirements Volatility and Its impact on Change Effort: Evidence based Research in Software Development Projects", Australian Workshop on Requirements Engineering, AWRE 2006 Adelaide, Australia.

[6] Didar Zowfgi, N Nurmuliani, "A study of the Impact of Requirements Volatility on Software project performance", Proceedings of Ninth Asia Pacific Software Engineering Conference 2002, pp 3-11.

[7] Evelyn J.Barry,TRIDAS MUKHOPADHY and Sandra A. Slaughter, "Software Project Duration and Effort :An Empirical Study ", Journal Information Technology and Management volume 3, 2002, pp.113-136.

[8] Y.Malaiya and J.denton, "Requirements Volatility and Defect density", Proceedings of the

10th International Symposium on Software Reliability engineering ,Fort Collins, 1998.

[9] Didar Zowghi, n Nurm**u**liani "investigating Requirement volatility during software development: Research in progress", Proceeding of the 3rd Australian Conference on Requirement Engineering Australia 1998.

[10] Daniela Damian,James Chisan, Lakshminaranan Vaidy Thamsamy, "Requirements Engineering and Downsteam Software Devlopment:Finding from a Case Study", Empirical Software engineering 10, 2005, pp.255-283.

[11] D.Pfahl, K.lebsanft, "Using simulation to Analyze the impact of software requirement volatility on Project performance", Information and Software Technology vol.42, 2000, pp.1001-1008.

[12] Frederick P. Brooks, director of IBM 360 projects.

[13] “A core ontology for requirements” , Zave 1997

[14] Software Engineering,8th Ed- Sommerville

[15] Software Requirements second edition by Karl E. Wiegers

[16] Stephen R.Schach. *Object-Oriented & Classical Software Engineering*. Sixth edition 2005, pp. 1-95.

[17] Sommerville I., Sawyer P. and Viller S. “Improving the Requirements Process”. Fourth International Workshop on Requirements Engineering: Foundation of Software Quality, 1998, 71-84.

[18] Hall Tracy, Beecham Sarah and Rainer Austen. “Requirements problems in twelve software companies: An empirical analysis". IEE Proceedings- Software, August 2002, pp.153-160.

[19] Effort estimation , 2004 ,http://tracer.lcc.uma.es/problems/estimation/estimation.html, (25th

july, 2010)http://tracer.lcc.uma.es/problems/estimation/estimation.html, (25th july, 2010)

[20] Software development Effort estimation 22nd april 2010http://en.wikipedia.org /wiki/Sofftware_effort_estimation, (30th April, 2010)

[21] Jorgenson Magne, Forecasting of software development Work Effort: Evidence on Expert Judgment and formal models, May 2007, 1-34.

[22] James S. O'Neal, Doris L. Carver, "Analyzing the Impact of Changing Requirements", Proceedings of the 2001 International Conference on Software Maintenance, Florence, Italy, November 2001, pp. 190-195