IMPROVING EXPERT JUDGMENT FOR SOFTWARE ESTIMATION

Supervisor:

Dr.Naveed Ikram
Associate Professor
Internatioal Islamic University
Islamabad.

Researcher:

Khalid Abubakr
Reg No. 172-FBAS/MSSE/F07
MS Software Engineering
Internation Islamic University
Islamabad.

July 2011
18 Shaban 1432

This thesis is submitted in partial fulfillment of the requirements for the Degree of Master of Philosophy/Science in Software Engineering at the

Department of Computer Science
Faculty of Applied Sciences
International Islamic University Islamabad

1- Software maintenance

MESSAGE

The original piece of improvement comes when you identify the right problem and the most effective solution. This identification requires in depth understanding of the concepts and knowledge related to the problems and available solutions. Estimates inaccuracy is one of the major problems in software engineering. Numerous models and methods are proposed to solve this problem. Expert Judgment has proven itself to be one of the most effective solutions considering its wide usage and accuracy of estimates it provides. However lack of in-depth understanding is the prevailing hurdle in its improvement to achieve even higher accuracy in estimates. This study is a means to resolve this obstacle and to improve Expert Judgment estimation.

DECLARATION

I hereby declare that this thesis, neither as a whole nor as a part thereof, has been copied out from any source. It is further declared that I have prepared this thesis entirely on the basis of my personal effort made under the kind and sincere guidenance of my supervisor.

No portion of the work, presented in this thesis, has been submitted in support of any application for any degree or qualification of this or any other university or institute of learning.

Khalid Abubakr (MSSE)

Department of Computer Science

Faculty of Applied Sciences

Interantional Islamic University

Islamabad.

IN THE NAME OF

ALLAH,

THE MOST MERCIFUL AND BENEFICIENT

Dedication

"To my great parents who nourished me and guided me to the right path, what I am today is only due to their relentless efforts for my sake"

ABSTRACT

Software industry is continuously demanding for accurate and reliable estimates as they form the basis for the core activities of project management i.e. bidding. planning and budgeting. Expert Judgment is most widely used estimation method. However. it lacks the research rigor as compared to the research work done in other estimation methods. Therefore directing more research resources to understand and improve the Expert Judgment is highly potential to fulfill the software industry's requirement of accurate estimates. The objectives of this research work was to provide preliminary understanding of Expert Judgment and reveal information variables that experts utilize to make judgment of software estimates. A list of information variables resulting from conducted survey identifies the required information variables for Expert Judgment. Afterwards a crossover experiment is performed to check the effect of these information variables on judgment estimates. Finally experts who participated in experiment are interviewed to understand the way they make judgment about software estimates. Transcripts are generated and textual analysis is performed for each interview to identify the *steps taken* and *information variables used* in each step. At the end a descriptive model is constructed to understand the way in which experts make judgment about software estimates. The purpose of constructing that descriptive model was to provide preliminary understanding of Expert Judgment to researcher for further improvement of this method.

ACKNOWLEDGEMENT

All the praises are attributed to the sole creator of the universe " the Almighty Allah", the Compassionate, the Merciful, the Source of all knowledge and wisdom, who bestowed upon me health, thoughts, talent, sincere and cooperative teachers, friendly brothers, helping friends and who gave me the strong courage and endurance to complete this thesis.

I express my deepest and sincere gratitude to my honorable and dignified supervisor Dr. Naveed Ikram (Associate Professor, Department of Software Engineering) for his inspiring guidance and continuous encouragement during the course of this research study.

I am also grateful to my friends Hameed-ur-Rehman, M.Fawad, Haider Abbas, Zareen Abbas, Waqar Hussain, Hamood-ur-Rehman, Adeel Akber, M. Rashid, M. Abid, M. Farhan and others for their kind support in data colloection, experimentation, reviewing and evluation of my research work.

I am also grateful to Mr.Shahzad Ahmad Qureshi, Mr.Zaheer Ahmed, Mr.Waseem Ahmad and Syed Afraz Ahmad who endowed their valuable time for experiment and intervew.

Khalid Abubakr

MSSE

International Islamic University Islamabad

TABLE OF CONTENTS

LIST OF TABLES 11

LIST OF FIGURES 12

LIST OF ABBRIVIATION 13

CHAPTER 1: INTRODUCTION 14

1.1 Software Estimation 15

1.2 Definition of Expert Judgment Estimation 15

1.3 Significance 16

1.4 Why Expert Judgment? 17

1.5 The Problem Statement (Gap) 18

1.6 Aim of Research 19

1.7 Research Questions 19

1.8 Scope 20

1.9 Research Process 20

1.10 Thesis Structure 21

CHAPTER 2: STATE OF THE ART 22

2.1 Research Dearth 23

2.2 Model vs. Expert Judgment 23

2.2.1 Estimates Accuracy 23

2.2.2 Practical Usefulness 24

2.2.3 Categorization of Approaches 24

2.3 Information Effect and Expert Learning 25

2.4 Historical Data Maintenance 26

2.5 Development Models 26

2.6 Size, Complexity and Simplicity 27

2.7 Expert Judgment Improvement 28

2.7.1 Environment Enhancement 28

2.7.2 Knowledge Base 28

2.8 Expert Judgment Process Description 29

CHAPTER 3: RESEARCH PROCESS 32

3. Research Process 33

3.1 Questionnaire Survey 33

3.1.1 Questionnaire Objectives 33

3.1.2 Target Population 34

3.1.3 Questionnaire Design 35

3.2 Survey Follow-up 40

3.3 Experiment 40

3.3.1 Project Task-1 41

3.3.2 Project Task-2 41

3.3.3 Design 42

3.3.4 Hypothesis 42

3.3.5 Experiment Elements 43

3.3.6 Experiment Execution 43

3.4 Follow-up Interviews 45

CHAPTER 4: RESULTS 46

4.1 Questionnaire 47

4.2 Survey Results Analysis 49

4.2.1 Environment Data 49

4.2.2 Project Type 51

4.2.3 Team Productivity 53

4.2.4 Domain Knowledge 55

4.2.5 Technology ... 56

4.2.6 Development Process ... 57

4.2.7 Resources ... 59

4.2.8 Customer ... 61

4.2.9 Questionnaire Result Categorization ... 62

4.3 Experiment ... 64

4.4 Experiment Analysis ... 65

4.4.1 T-test Analysis ... 68

4.5 Follow-Up Interview ... 70

4.5.1 Model Explanation ... 70

4.6 Discussion ... 72

CHAPTER 5: CONCLUSION ... 75

5.1 Research Summary ... 76

5.2 Research Work Limitations ... 77

5.3 Conclusion ... 77

5.4 Contribution ... 78

5.5 Future Work ... 79

7. APPENDIX - I: ASSOCIATED INFORMATION ... 84

I-a Questionnaire ... 84

I-b Project Task 1 ... 86

I-c Project Task 2 ... 90

I-e Textual Analyzed Transcripts ... 92

I-f Textual Model of Estimators ... 105

LIST OF TABLES

Chapter: 3

Table 1: Summary of literature review 30

Table 2: Information Categories 35

Table 3: Detailed Representation of Information Categories 37

Table 4: Actual Figures of Task 1 41

Table 5: Actual F41igures of Task 2 42

Table 6- Experiment Design 42

Table-7 Experiment Elements 43

Chapter: 4

Table-8 Questionnaire Results 47

Table-9 Response Summary of Environment Variables 50

Table-10 Response Summary of Project Type Variables 51

Table-11 Response Summary of Team Productivity Variables 53

Table-12 Response Summary of Domain Knowledge Variables 55

Table-13 Response Summary of Technology Variables 57

Table-14 Response Summary of Development Process Variables 57

Table-15 Response Summary of Resource Variables 59

Table-16 Response Summary of Resource Variables 61

Table-17 Results Summary with Response Rate 62

Table-18 Experiment Execution and Experts Estimates 64

Table-19 Informed and Uninformed Actual, Estimate and Difference %... 65

Table-20 Difference % of PT1 and PT2 Estimates 68

Table-210 Statistical T-Test Implementation 68

LIST OF FIGURES

Chapter 1

Figure-1.1: Research Process 20

Chapter 4

Figure 4.1: Descriptive Model for the Way Expert Judge Effort,

Time and Cost Estimates 71

LIST OF ABBRIVIATION

FP – Function Point

LOC – Line Of Code

LRC – Linear Responsibility Chart

MIS – Management Information System

OP – Object Point

PSP – Personal Software Process

PSP PROBE – Personal Software Process PROxy Based Estimation

RAM – Resource Assignment Matrix

SPM – Software Process Management

TSP – Team Software Process

UCP – Use Case Point

WBS – Work Breakdown Structure

CHAPTER 1: INTRODUCTION

1.1 Software Estimation

The two of primary concerns in Software Project Management are predicting the required cost and time for a project. Numerous studies are carried out and various models/methods are proposed for what we know as "Software Estimation". At an abstract level, estimation methods are divided in two categories: *Heuristic* e.g. Delphi, Wideband Delphi, Expert Judgment etc. and *Parametric* e.g. Function Point, UC Point, COCOMO etc (Yenduri, Muagala & Perkins, 2007).

In *Oxford Dictionary* Greek word Heurisko is the origin of Heuristic which means *to discover*. There exist multiple interpretations for the word *heuristic* e.g. useful, even indispensable cognitive processes for solving programs that cannot be handled by logic and probability theory or a rule of thumb for solving a problem without the exhaustive application of an algorithm (Jorgensen & Sjoberg, 2002).

1.2 Definition of Expert Judgment Estimation

Judgment-based estimation is the utilization of personal knowledge, abilities and information e.g. intuitions, educated guess etc. by human without any extensive use of mathematical or statistical formulas to make an estimate.

Parametric models follow the 'Tayloristic' principle because these estimation models focus on well-defined processes and the matrices that result from statistical analysis of previously completed projects (Jorgensen & Sjoberg, 2002). Primarily parametric models base their estimate on the size -in different notions e.g. Function Point (FP), Use Case Point (UCP) etc. - of the software and then adjust these estimates for different factors (Jorgensen & Sjoberg, 2005: Molokken-Ostvold & Jorgensen, 2005). An assumed yet imprecise advantage of models is to provide unbiased estimates i.e. estimates that are unaffected by gut feeling or likelihood (Jorgensen & Boehm, 2007).

1.3 Significance

Estimates are important to determine the success or failure of software projects. Projects having substantial deviation from their estimates are considered far from success. The Standish Group CHAOS reported the 'reliable estimates' as one of the top ten criteria for determining project success or failure (Grimstad, Jorgensen & Molokken-Ostvold, 2005).

Software project management becomes intimidating in the absence of accurate and reliable estimates (Yenduri et al, 2007). Bidding, planning and budgeting for project are the core activities of software project management. The knowledge of efforts required for projects-under-development is critical as it sets the basis for these core activities. The unrealistic estimates result in erroneous and flawed bidding, planning and budgeting and take the software projects to failure which could eventually damage the whole software company (Grimstad et al, 2005; Jorgensen, 2007; Jorgensen, 2006). Numerous survey findings show that most of the estimates produced are inaccurate and tend to be over-optimistic (underestimated) (Molokken-Ostvold & Jorgensen, 2005). The worst affect of overoptimistic estimates is devastation of software company (Jorgensen, 2006). On the other hand pessimistic estimates (overestimated) cause loss of business opportunities (Grimstad et al, 2005).

The major concerns of today's software industry include the overrun of software estimates and effects of overrun are manifolds e.g. poor planning, delays, contractual loses, decline in product quality etc (Jorgensen, 2007; Furulund & Moloken-Stvold, 2007; Grimstad & Jorgensen, 2007). Software industry requires more accurate estimates. Heuristic and parametric methods are present for software estimation but they do not fulfill the industry requirement. Standish Group reports that only 33% of all the projects get completed on their estimates. These evidences emphasizes that there is a need for improvement of software estimation methods (Jorgensen & Sjoberg, 2002; Jorgensen & Boehm, 2007).

1.4 Why Expert Judgment?

Model based software estimation has been the focus of research for the last four decades and many studies are conducted for improving these models but results coming from these extensive and continuous research works are not as effective as expected to be. Academic as well as industrial research show huge differences in effectiveness of research and accuracy of estimates produced which indicate the failure of estimate models to achieve their goal (Molokken-Ostvold & Jorgensen, 2005).

Expert and Parametric estimations methods considerably yield equally accurate results, yet favors judgment based estimations because, despite 40 years research on estimate models, judgment based model is still the most widely used estimation method in the software industries (Molokken-Ostvold & Jorgensen, 2005; Grimstad & Jorgensen, 2007).

A further factor is that even the parametric models themselves require expert or experienced persons for estimation. A novice person cannot make accurate estimates by using parametric models (Fraser, Boehm, Erdogmus, Jorgensen, Rifkin & Ross, 2008). Yet another factor for the use of Expert Judgment is its simplicity and less time consumption. Parametric models may give the same estimation accuracy but they are complex to use and time consuming than the Expert Judgment. Jorgenson (Jorgensen & Sjoberg, 2002) stated "Software effort estimation doesn't necessarily require introducing sophisticated formal models...it can be as simple as reframing questions to capture more accurately the project context and characteristic".

The Expert Judgment method is flexible to include much contextual information than estimation models. This flexibility signifies its superiority over estimation models. Software estimates are built over the requirements and requirements in software development are never static. Estimation methods need to be highly flexible to accommodate changes and new information. In this regard Expert Judgment provides the highest level of flexibility (Yenduri et al, 2007; Jorgensen & Boehm, 2007).

1.5 The Problem Statement (Gap)

Prerequisites to improve any method are in-depth understanding of that method plus the assumptions it makes. It is almost impossible to improve any method without such understanding. Similar rule apply to Expert Judgment. Lack of precision understanding is the prevailing hurdle in the improvement of this method (Jorgensen, 2007, 2006). In literature the process between understanding the problem and quantification of efforts is described as the 'magic step'. The cause is the low understanding and fewer amount of research on that topic (Jorgensen, 2006).

An example adapted form (Jorgenson, 2006); in real world scenario a manager is asked that how much time and efforts will be required to complete a project X with functionality A and B, for a bank Z? Manger responds that it will take 4 months however an extended time line of two months will be good because security issues.

The numbers are not coming from an objective observation that requires previous knowledge and contextual information to produce these numbers e.g. information to determine the security issues. The Expert Judgment works as a black box to produce the estimates and the literature of software engineering too, lacks the information to give knowledge about strengths and weaknesses of Expert Judgment estimation. This knowledge can be used to build a *descriptive model* of that process (Jorgensen & Sjoberg, 2002) for better understanding. A study scrutinizing 100 journals reported only one research paper attempting to model the analogy based expert process (Jorgensen, 2006). This dearth of research on Expert Judgment demands further work for understanding of the process to improve the Expert Judgment. Psychological research cited by (Jorgensen, 2007, 2006) describes human judgmental as something that "has not culminated in any theory of estimation, not even in a coherent framework for thinking about the process"

1.6 Aim of Research

Aim of this research work is to provide better understanding of the Expert Judgment and reveal information variables that the experts use to make judgment about software estimates. Furthermore, to format results to guide the estimators recognizing the minimum information they need to produce better estimates.

1.7 Research Questions

RQ1: - What information variables experts use in judgment-based software estimation?

RQ2: - What is the impact of the information variables on accuracy of judgment-based software estimates?

RQ3: - How do the experts make judgment-based software estimates?

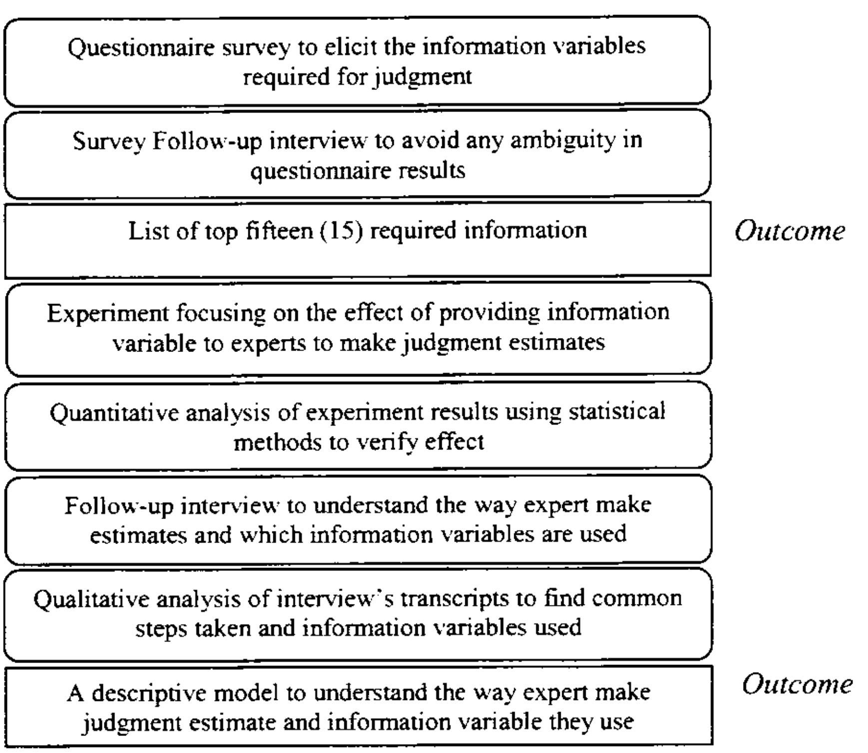
1.8 Scope

The main focus of research was to indicate the required information variables for Expert Judgment. Although the research literature on human judgment and decision making is considered important to understand judgment process (Jorgensen, 2007) but the personal capabilities and tacit knowledge related information variables are excluded from the scope. The scope of this study is limited specifically to software engineering and project development related information variables. A study based on behavioral science and/or human judgment theories could be appropriate for the excluded information variables elicitation.

1.9 Research Process

Figure-1.1 briefly describes the design of research process. Further details are provided in Chapter-3

Figure-1.1 Research Process



1.10 Thesis Structure

The thesis consists of 5 chapters. Chapter 2 synthesizes literature related to research topic. Chapter 3 describes the research process adopted to solve the research problem. The results of research are presented in Chapter 4. Finally Chapter 5 gives a summary and conclusion of research work with research limitations and future work.

CHAPTER 2: STATE OF THE ART

2.1 Research Dearth

This dearth of research on Expert Judgment demands further work for understanding of the process to improve the Expert Judgment (Jorgensen, 2006). There are barely any studies discussing the way experts judge software estimates or required information variables. Jorgensen reported only 17% out of 100 papers, analyzing or discussing Expert Judgment for software estimation (Jorgensen, 2004). Scrutinizing 100 journals resulted in only one research paper that attempts to model the analogy base software estimation (Jorgenson, 2006). However, the dominance of Expert Judgment is reported in various studies (Jorgensen & Boehm, 2007; Jorgensen, 2007; Grimstad & Jorgensen, 2007; Nasir, 2006; Hughes, 1996).

2.2 Model vs. Expert Judgment

2.2.1 Estimates Accuracy

There are expectations that 1) future research on model-based estimation may improve the estimate accuracy and 2) combination of model based estimation and Expert Judgment can yield more accurate results (Fraser et al, 2008). However predictable results indicate that sophisticated and complex models are unlikely to improve estimate accuracy because of dynamic nature and unstable relationship of requirements (Fraser et al, 2008; Boehm, 1981). It is recommended to combine model and judgment methods to build accurate estimates (Jorgensen & Boehm, 2007). However, a mixed approach of model based estimation and Expert Judgment raise the problem of scarce research and industrial resources and only increases the confidence level for estimates unless and until each method compensate for the strengths and weaknesses of other (Fraser et al, 2008).

Estimate produced using Expert Judgment is criticized for biased and influenced by organizational political pressure (Hughes, 1996). However a precise knowledge of 'how estimation models work' reveals that no method is protected from these effects (Fraser et al, 2008). More accurate estimates are produced using Expert Judgment than model based or mixed estimation (Furulund & Moloken-Stvold, 2007).

2.2.2 Practical Usefulness

A comparison of Expert Judgment and estimation models indicates 62% to 83% use of Expert Judgment than models and equal performance with respect to accuracy of estimates (Jorgenson, 2004). Although the accuracy of models is conditioned to find out a fit match of data yet it does not compensate for the widespread usage of Expert Judgment. It is also evident that the replacement of Expert Judgment with model based estimation is unlikely to happen (Grimstad & Jorgensen, 2007). The Expert Judgment is preferred over models because the dynamic nature of software requirements makes it difficult to use predefined values for model parameters therefore model must use the judgment to determine the initial value for required parameter (Fraser et al, 2008).

2.2.3 Categorization of Approaches

B. Boehm categorized the estimation methods into six categories and described the utility of estimates with four perspectives that are bidding, tradeoff and risk analysis, project planning and controlling and software improvement investment analysis (Boehm, 1981). Prior to use for estimation, the strengths of leading estimation techniques should matched to achieve the respective perspective.

Most famous techniques under expert based estimation are Work Breakdown structure (WBS), Delphi, and Wideband Delphi (Bohem & Chulani, 1998). Although techniques such as Parkinson's Law, Price to Win, Rule of Thumb, Buy vs. Make Decision, Personal Software Process PROxy Base Estimation (PSP PROBE) and Team Software Process (TSP) is also included under heuristic approach (Nasir, 2006) yet Parkinson's Law, Price to Win are already criticized for being scope setting and pricing tactic rather than estimation technique (Hughes, 1996).

2.3 Information Effect and Expert Learning

(Grimstad & Jorgensen, 2007) reported overestimation of most likely efforts when expert estimation is used in existence of irrelevant information. The recommendation of removing irrelevant information from requirements specification before using it for estimation (Grimstad & Jorgensen, 2007) arise the question of separating irrelevant information form relevant information. The expert's resistance too, increases towards misleading information when they are exposed to such information after making estimates i.e. the order of exposure to misleading information affect the Expert Judgment (Grimstad & Jorgensen, 2007).

The experts show low learning level from their experience because their work is based on unconscious processes and unavailability of explicit process description based their inability to update their initial estimates (Jorgensen, 2008). The experts are unlikely to accurately adjust their initial estimates when less biased, conflicting or irrelevant information props up because *mental and intuitional process, beliefs and experience* are involved in Expert Judgment which are difficult to change at a later stage (Jorgensen, 2008).

The experience level of the expert is not a realistic indicator for the accuracy of estimates i.e. there are situations when highly experienced experts may show low learning through their experience (Jorgensen & Sjoberg, 2005). Despite low learning abilities, the experience of the experts is not taken to be unimportant for judgment because experts use any *available information* to 'somehow' produce the estimates (Jorgensen & Sjoberg, 2005).

2.4 Historical Data Maintenance

Estimation is a prediction work and accuracy of the prediction depends on availability of reliable information (Hughes, 1996). The situations when historical data is unavailable or unreliable constantly occur in the field of software engineering because few organizations put up resources to maintain historical data (Furulund & Moloken-Stvold, 2007; Fraser et al. 2008). The basic reason for limited resources is the low value given to estimates by the software industry (Fraser et al, 2008). The use of experience data (if available) together with human judgment can improve the accuracy of estimates but data reliability and maintenance cost becomes obstacles. (Furulund & Moloken-Stvold, 2007; Yenduri et al, 2007). The unique nature of each project decreases the reliability of experience data whereas 5% to 10% of total project cost cause organizations avoid maintaining historical data. (Valerdi, 2007) reported two heuristic techniques i.e. representativeness and anchoring for estimation and the ability of experts to develop estimates by making certain assumptions *in absence* of required information. Although credibility of these assumptions is not justified but software estimates are influenced by these assumptions (Valerdi, 2007).

2.5 Development Models

Researchers are now struggling to explore *factors* that affect the accuracy of estimates and Software Development Lifecycle (SDLC) models are considered one of the factors (Molokken-Ostvold & Jorgensen, 2005). Parallel to reliable estimates, better management and control process and early feedback offered by flexible models are suggested to avoid the estimates overrun (Molokken-Ostvold & Jorgensen, 2005). However, neither the management and control nor the reliable estimates alone seems to resolve the estimates overrun. They must go hand in hand for a successful project.

2.6 Size, Complexity and Simplicity

Due to many reasons (change in requirements, change in business process (BP) etc.) software may grow in size and importance, but parallel to its size and importance the complexity of software also increases (Boehm, 1981) that makes it more difficult to predict an estimate for the software. The complexity factors attached with software estimation (such as intrinsic and unique nature of task, productivity, reliability, changeability and testability and relation among the factors) are anticipated to increase difficult of the task (Collopy, 2007) that is complexity factors should also be dealt to increase the accuracy of estimates.

Expert Judgment is a simple process. Experts use very few but important information variables to make judgment. Computer specialists assumed that managerial decision making uses plenty of information but the managers use very simple and easy process and typically use the information that worked well in past to make a decision (Hughes, 1996). Managers anticipate the overlooked task and unexpected event the main reason for estimates inaccuracy (Furulund & Moloken-Stvold, 2007). (Jorgensen, 2005) stated "Software effort estimation doesn't necessarily require introducing sophisticated formal models...it can be as simple as reframing questions to capture more accurately the project context and characteristic".

2.7 Expert Judgment Improvement

2.7.1 Environment Enhancement

The best practices for Expert Judgment are proposed to achieve accurate estimates and need to have a description of *actual processes* of Expert Judgment is highlighted (Jorgensen, 2004,2005). The main focus of these guidelines is to set the best environment to get accurate results. For example asking for *justification* motivates experts to achieve more accurate estimates but invisibility of process and information utilization still persists.

2.7.2 Knowledge Base

The estimation of 70 to 80 % software projects is done using Expert Judgment (Grimstad & Jorgensen, 2007) however; the reported lack of its knowledge base indicates the need of better understanding of the method for its efficient use and improvement.

A common understanding of estimation terminologies and their underlying goals can improve the practicality of estimates and make learning possible from these estimates (Grimstad et al, 2005). Three prominently used terminologies are bid, planned effort and most-likely effort. Each one has specific goals at their respective phases but different estimation terms are used regardless of their underlying goals which may render them vulnerable to inaccuracy.

A debate supporting Expert Judgment is reported in (Jorgensen & Boehm, 2007). Jorgensen pointed out that there is less likelihood of people to use the models because of model's inability to include extremely definite knowledge of experts as estimation inputs and argued that "more *structured and supporting elements* can further improve the judgment-based effort estimation" (Jorgensen & Boehm, 2007). Boehm recommended the maintenance of estimates *related information* with the use of mixed approach (expert and model based estimation) to improve estimation accuracy (Jorgensen & Boehm. 2007) but the benefits of mixed approach are only realized if we know the strengths and weaknesses of both model (Fraser et al, 2008). They recommended that the improvement of Expert Judgment to be the focus of current research. However, the goal of improvement cannot be achieved unless we have the basic knowledge about how experts make judgment and what information they need to make judgment estimates.

(Mannhart, Bilgic & Bertsche, 2007) sated that the reliability of Expert Judgment must be checked in the early stages of decision making in software development lifecycle (SDLC) but it is only possible if we can handle *experts' knowledge related data*. Although the experts' knowledge is not quantifiable and consists of qualitative relationships of *multiple variables*, eliciting experts' knowledge is a mandatory part for Expert Judgment and decision making (Mannhart et al, 2007).

2.8 Expert Judgment Process Description

Formal models and extensive experience based databases are not the primary requirement to improve the Expert Judgment but description of *the steps taken or the information used* for making judgment about software estimates can be the prevailing step of its improvement (Jorgensen, 2005). Research Literature lacks the knowledge about strengths and weaknesses of Expert Judgment estimation that can be used to build a *descriptive model* of that process (Jorgensen & Sjoberg, 2002) for better understanding and improvement.

Expert Judgment is hardly studied for the discovery of underlying processes that are involved in making estimates. (Jorgensen, 2007) based on human judgment literature proposed theory consisting on three (03) concepts which are considered important for Expert Judgment i.e. a) human using intuitional and unconscious process categorize and classify the world around them in order to understand the complex relationship among multiple variables, b) they tend to understand new information conforming to their first impression however, this long lasting, hard to change *first impression* may based on surface information, c) they tend to have representative values "anchors" from their past experience/projects and base these values to build new estimates i.e. people prefer relative value than absolute value (Jorgensen, 2007). The experienced people have high confidence in their anchors and low willingness to change or deviate from these anchors which affect their judgment and decision making process (Jorgensen, 2007). According to (Jorgenson, 2007) Expert Judgment involves many intuitional and mental processes and '*contextual information*' that are not explicit while proper understanding of the processes and explicating the contextual information can help us improve the Expert Judgment (Jorgensen, 2007).

Table-1 represents a synthesis of literature survey for better understanding and categorization of research article for different concepts.

Table 1- Summary of literature review

Articles Reference	Concepts										
	Dearth of Research	Model vs. Expert Judgment			Information Effect and Expert Learning	Historical Data Maintenance	Development Models	Size, Complexity and Simplicity	Expert Judgment Improvement		
		Estimates Accuracy	Practical Usefulness	Categorization of Approaches					Environment Enhancement	Knowledge Base	Expert Judgment Process Description
(Boehm, 1981)	○	●	○	●	○	○	○	●	○	○	○
(Bohem & Chulani, 1998)	○	○	○	●	○	⊙	○	○	○	○	○
(Collopy, 2007)	○	●	⊙	○	○	○	○	●	○	○	○
(Fairy, 1992)	○	⊙	○	○	○	○	○	⊙	○	○	○

(Fraser et al. 2008)	⊙	●	○	○	○	●	○	●	○	●	⊙
(Furulund & Moloken-Stvold. 2007)	⊙	●	○	○	⊙	●	○	○	⊙	○	⊙
(Grimstad & Jorgensen. 2007)	●	⊙	●	○	●	○	○	○	○	●	⊙
(Grimstad et al. 2005)	⊙	⊙	○	○	○	○	⊙	○	⊙	●	⊙
(Hughes. 1996)	●	●	○	●	○	●	○	●	○	○	○
(Jorgensen & Boehm. 2007)	●	●	⊙	○	○	⊙	○	○	⊙	●	⊙
(Jorgensen. 2008)	⊙	○	○	○	●	○	○	○	○	○	⊙
(Jorgensen. 2007)	●	⊙	○	○	○	⊙	○	⊙	⊙	⊙	●
(Jorgenson.2006)	●	⊙	○	○	○	○	○	○	○	⊙	●
(Jorgensen. 2005)	⊙	⊙	⊙	⊙	○	⊙	○	●	●	⊙	●
(Jorgenson.2004)	●	○	●	○	○	○	○	○	●		●
(Jorgensen & Sjoberg. 2005)	⊙	○	○	○	●	○	○	●	○	○	○
(Jorgensen & Sjoberg. 2002)	○	⊙	⊙	○	○	○	○	⊙	○	●	●
(Little. 2006)	○	⊙	○	○	○	●	⊙	⊙	⊙	○	○
(Mannhart. Bilgic & Bertsche. 2007)	○	⊙	○	○	○	⊙	⊙	○	○	●	○
(Molokken-Ostvold & Jorgensen, 2005)	○	⊙	●	○	○	○	●	⊙	⊙	⊙	○
(Nasir. 2006)	●	○	○	●	○	○	○	○	○	○	○
(Valerdi. 2007)	○	⊙	○	○	○	●	○	⊙	○	○	○
(Yenduri et al, 2007)	○	⊙	⊙	○	○	●	○	⊙	○	○	○
Concept Discussion Level	**●Fully Discussed**					**⊙Partially Discussed**			**○ Not Discussed**		

CHAPTER 3: RESEARCH PROCESS

3. Research Process

Difficulty to study the Expert Judgment is evident in numerous research studies. Expert Judgment involves many intuitional, mental and subconscious processes (Jorgensen, 2008) which indicate that the issue cannot be studied through a single research method. Inspired by facts, a fine blend of qualitative and quantitative methods is prepared. The process of this research work consists of four steps.

3.1.1 Questionnaire Survey

3.1.2 Survey Follow-up Interview

3.1.3 Experiment

3.1.4 Follow-up Interview

3.1 Questionnaire Survey

The literature reported different information variables vital for software project development but not precisely for Expert Judgment. A precise literature survey was performed and all participants from different organization were asked to respond whether they use information variables elicited from the literature to make judgment estimates.

3.1.1 Questionnaire Objectives

- *To find out the required information variables in real-time Expert Judgment estimation.*

A number of information variables are generated during software development. However, the experts utilize selected and limited information variables in judgment estimates (Hughes, 1996). Therefore it was important to find out the information variables considered *required* by on-job experts to develop early estimates. (Jorgensen. 2007) reported that each individual expert contains highly specific and valuable information and they use this information to make their judgment estimates. Thus there might be information variables that are used by individual expert but are not reported in literature. The open-ended question in the questionnaire enabled us to extract these information variables.

3.1.2 Target Population

The experts working in different software houses of Pakistan were our target population. The eligibility criteria's set for target population were:

- Related experience of expert is >= 3 years
- Expert is PMI certified professional with related experience >= 1 years

105 experts working in different software companies situated in Lahore, Islamabad and Karachi are selected in random order. All of them fulfilled the eligibility criteria. Prior to conducting questionnaire survey a participation request was sent through email to these experts. The promise of sharing research results with them was used as a tool for their motivation. Ninety (90) out of one hundred and five (105) people showed their interest for participation by replying to our email. The questionnaire was sent to these people setting our n = 90. Fifty six (56) participants form thirteen (13) organizations sent back filled questionnaire providing a response rate of approx 62%. The remaining thirty four either did not reply or replied after deadline and were excluded from the study.

- **Main characteristics of population**

All the participants are working on position of senior software engineers or higher positions. Their estimation related experience varies from 3 to 7 years or more. They have different educational levels and professional certifications.

3.1.3 Questionnaire Design

The questionnaire designed includes thirty three (33) close-ended questions to get opinion of experts and an open-ended question to get their personal suggestions and insights of individual expert to detect missing or new information specially related to software estimation. The information categories extracted from software engineering literature and deemed important for Expert Judgment estimation are:

Table 2- Information Categories

S. No	**Information Category**	**Frequency**	**Literature Survey**
1	Technology expertise level	5	Multiple technologies are used in organizations. When a project is estimated the expertise level of technology in which the project should develop is considered (Molokken-Ostvold & Jorgensen, 2005; Furulund & Moloken-Stvold, 2007; Hughes, 1996; Valerdi, 2007; Fairly, 1992).
2	Uncertainty level in project	4	Typically the projects involve a level of uncertainty of work with some new innovative work (Jorgensen & Boehm, 2007; Hughes, 1996; Mannhart et al, 2007; Little, 2006).
3	Team productivity	4	How much efficient work a team / an individual can deliver in a specific time (Molokken-Ostvold & Jorgensen, 2005; Jorgensen & Boehm, 2007; Hughes, 1996; Collopy, 2007).
4	Domain knowledge	3	Involving the people with significant knowledge of domain to which project belongs. Example: if we have banking software we must have finance and banking knowledge (Molokken-Ostvold & Jorgensen, 2005; Jorgensen, 2004; Hughes, 1996).

5	Product type	3	It is important to know whether project is standalone, embedded or distributed application etc. Some time project is also considered as small, medium, large projects (Grimstad et al, 2005; Fraser et al, 2008; Fairly, 1992).
6	Team motivation	3	Team motivated to perform work efficiently and take problems as challenges (Jorgensen, 2004; Hughes, 1996; Fairly, 1992).
7	Resource availability	2	It is important to know whether the resources required can be available as and when needed (Hughes, 1996; Fairly, 1992).
8	Customer expectation	2	Willingness of customer that how much cost s/he can spend and how long s/he can wait for the project (Grimstad & Jorgensen, 2007; Fairly, 1992).
9	Development environment and process	2	Friendly and collaborative environment and flexible development process affect the outcome (Valerdi, 2007; Fairly, 1992).
10	Scope	2	Project is single product or a family of products. Is there any potential to further market for the current developing product? It may notify utility of product resulting from a particular project (Molokken-Ostvold & Jorgensen, 2005; Valerdi, 2007).
11	Team composition	1	An organization can have static or dynamic composition of team. Static teams may have low communication gap and high level of trust among team members. These types of characteristic affect the outcome of teams (Furulund & Moloken-Stvold, 2007).

Although information categories in Table-2 are reported to effect software estimates but they were on higher abstraction level. These information categories are scrutinized to lower abstraction level to find aspects that affect the Expert Judgment estimates. This activity increased the reliability of collected data and revealed relationship between them. Table-3 below presents more detailed view of information categories based on (Jorgensen, Indahl & Sjoberg, 2003; Yin, Peterson & Arellano, 2004; Wu, 2006; Lionel, Khaled, Dagmar, Isabella, & Katrina, 1999).

Table 3- Detailed information categories and information variables

S. No	Category	Sub-Category	Information Variable
1	Environment Data	Deployment Environment	No. of Users, Expertise of User, Platform Used. Organizational Structure of client organization, Size of Organization, Standards Policies Followed
		Development Environment	Organizational Structure, Type of Development (In-house, Project, Product), Standards and Policies Followed, Communication Interfaces (within and out of organization), Maturity Level
2	Project Type	Application Type	Customer Services, MIS, Office IS, Process Control and Automation, Network Management, Transaction Process, Product Control and Legislation, Online and Information Services.
		Organization Type	Bank, Wholesale/Retail, Insurance, Manufacturing, Public Administration, Space.

			Military, Business Industry, Educational, Health Care
		Non-Analogy	Size (LOC, FP, OP,UCP)
		Analogy	Small, Medium, High
		Complexity	Internal: Algorithmic Complexity, Logical Complexity, Defects Occurrence Interval, No. of Modules Included, Execution Time, Storage Constraints, Level of Innovation
			External: Deadline pressure, conformance from customer, opportunity window, Product age, change in compatibility of platform over time, variability of features requested, required reliability
3	Team Productivity	LOC/hr	Programmer experience, Team size, Tool used.
			Team member collaboration, Team motivation, Customer participation
		Morale	*Career Planning, Confidence, Determination, Accountability*
4	Resource Availability	WBS	Activities involve, Resource Needed, Activities dependency
		Components	Reusable component exist, Break through point

		& Tools	of reusable component. Required tools available
		Staff Availability	Staff plan histogram, Resource assignment matrix(RAM)/ Linear Responsibility Chart (LRC)
5	Customer	Satisfaction	*Customer loyalty, Customer Variability*
			Budget Expectation, Delivery Expectation, Support and Training
6	Domain Knowledge	Development	Application Experience, Analyst Capability, Development platform maturity, virtual machine volatility,
		Market	*Market size, market growth, market potential*
		Other	*Company policies, Legislation, Standard dominance*
7	Technology	Complexity of technological problems	Language Experience, technological constraints, Software tools usage
			Leading/bleeding edge
8	Development Process		Maturity of SPM process, heavy/light weight(level of documentation required) process, fixed/flexible process(defined or undefined)

Table-3 provides drilled down view of information category and their related information variables. However '*market*' related variables were excluded in our study because these variables were specifically related to product development. Similarly *'others'* category containing information variables e.g. *legislation* information of organization, typically unrelated to project was also considered unimportant. *Team moral, customer loyalty and variability* are highly subjective, volatile variables which are rarely considered important for Expert Judgment (Bekkers, Ven de Weerd, Brinkkemper & Mahieu, 2008).

3.2 Survey Follow-up

There was a possibility to get ambiguous or unrealistic responses in survey. Therefore follow-up interviews were planned for such situations to avoid ambiguity and unrealistic responses. Telephonic interviews were conducted from respondents who did not respond to some of the questions of survey. They replied that it was an omission on their part. Hence these individual questions were excluded from the calculation of percentages for the respective information variable. In a follow up interview one of the respondents showed unfamiliarity of term *breakthrough point.* Hence, it was assumed that other respondents who replied *'uncertain'* for *breakthrough point* were also unfamiliar with the term so their responses for that question were also excluded that is the frequency of the question was decreased.

3.3 Experiment

There were two software projects to perform experiment. Project task 1 was a medium size telecom billing and customer management application. Project task 2 was comparatively smaller solution developed for money exchange company to manage its daily local and foreign currency transactions and customer's information. The questionnaire highlighted the importance of diverse information variables for making judgment about estimates of a software project. However, the consequence of providing these important information variables to expert for judgment was unknown. The experiment's objective was to discover the effect of providing information variables explicitly to experts for judgment estimates. The objective of experiment was

- *To find out the effect of providing information variables to experts on their judgment estimates*

3.3.1 Project Task-1

Project task-1 (PT1) was a telecom project. It was a customer support and billing application and developed to fulfill all the basic requirements of back office Billing and Customer Care of any Wireless or Wire-line Service Providers. The project's goal was to convert existing Enterprise System to an Open Source, Web2.0 enabled and NGOSS compliant suite of applications that would facilitate the Cellular and Landline Service Providers and integrate Telecom Service Providers. Actual figures of project task-1 are given in Table-4:

Table 4 - Actual figures of PT1

Actual Efforts	12936 man hours
Actual Time	7 calendar months
Actual Cost	PK RS: 4,200,000

3.3.2 Project Task-2

Project task-2 (PT2) was a finance application. It was developed for a money exchange and transfer company. This company has branches in different cities of Pakistan and provides money transfer service to many foreign countries. The project was to manage the daily local and foreign currency transaction. The application also includes customer registration and management. Actual figures of PT2 are given blow:

Table 5 - Actual figures of PT2

Actual Efforts	1050 man hours
Actual Time	2 calendar months
Actual Cost	PK RS: 235,000

3.3.3 Design

Crossover experiment (Senn, 2002) design was followed to avoid sequence affect and biases in Expert Judgment. Experts were divided in two groups. In Group-1 each individual was assigned PT1 to make judgment estimates. After a *wash away* time of one week PT2 was assigned to each individual of Group-1 for estimation. Similar fashion was followed for the Group-2 except the project tasks were assigned in reverse order. The information considered important by the questionnaire respondents was included only with PT1 while PT2 was assigned without that information. Tabular representation of the experiment design is given below:

Table 6 - Experiment Design

Treatments / Subjects	Informed	Uninformed
GR1	GR1 estimates PT1 with information	GR2 estimates PT2 without information
GR2	GR2 estimates PT1 with information	GR1 estimates PT2 without estimates

3.3.4 Hypothesis

The hypotheses evaluated to answer the second research question were:

Null Hypothesis:

There is no significant difference between informed and uninformed estimates by providing information variables explicitly to the experts for making judgment about software estimates.

Alternative Hypothesis:

Informed and uninformed estimates significantly differ by providing information variables explicitly to experts for making judgment about software estimates.

3.3.5 Experiment Elements

The experiment involved two projects as tasks to make judgment estimates. These tasks were assigned to experts to make early judgment estimates. Hence the experiment consists of:

Table 7 - Experiment elements

Elements	**Description**
Object	Judgment to which treatment was applied
Subject	Experts were subjects who apply the treatment
Controlled Object	Time allowed for estimation, Limited communicate with other persons.
Treatment	Explicitly providing information variables needed for more accurate estimation
Dependent Variable	Estimate produced by judgment is the dependent variable
Independent Variable	Quantity of subjective information used

3.3.6 Experiment Execution

Experiment execution started with a 15 minutes discussion period to develop basic understanding regarding experiment and what experts should provide in response.

The purpose of the discussion session was to avoid any concerns of the subjects regarding the experiment or project task and developing a common understanding of different terms. Jorgenson reported that common understanding of the objectives and terms can help to achieve realism in judgment estimation (Grimstad et al. 2005).

Four experts were selected as subjects of the experiment. All of them are working on high position in well reputed software companies in Pakistan and have more than 5 years of estimation related experience. Two balanced groups were formed containing two subjects (called Estimator hereafter) per group. According to study design Group-1 was assigned project task 1 with information variables and Group-2 was assigned project task 2 without information variable. Estimators were allowed to ask any query for their understanding or to clear any ambiguity that they found in project task to make their estimates. Every estimator was given 60 minutes to read and understand the project task and make judgment estimates for given task.

Second phase of the experiment started after a wash away time of one week. The only difference in this phase was the order of project task assigned to estimators. After a 15 minutes discussion GR-1 which had assigned PT1 with information variables was now assigned PT2 without information variable. Similarly GR-2 which had assigned PT2 without information variables was now assigned PT1 with information variables.

The data collected from the each estimator includes assigned project task, efforts estimates in man hours or man months, time estimates in calendar months and cost in Pakistani rupees. The sheets provided to estimators for their calculations were also collected back so that we can analyze and understand the way these estimators calculated the estimates.

3.4 Follow-up Interviews

A reflective follow-up interview was conducted from every estimator. The basic objective of the interview was to understand the way expert judged his software estimates and used the information variables. Participants were asked to recall their experiment work process and information variables used, or assumptions made. Each interview was recorded and a transcript were generated for every interview (*transcripts are given in appendix-I*). The transcripts were textually analyzed for the *steps taken* and *information variables used* to make judgment estimates of given tasks. This analysis also clarified which information variables were used in each step. A textual model is developed for every estimator by categorizing steps taken and information variables used related to *cost judgment, time judgment* and *effort judgment.*

Finally the common steps and the information variable used in these step were organized to construct a descriptive model for understanding the way expert judged effort, time and cost estimates.

CHAPTER 4: RESULTS

4.1 Questionnaire

A summarized detail of questionnaire survey is given below:

Questionnaire sent to:	90 persons
Total responses received from:	56 persons
Completely filled questionnaires:	53
Incomplete questionnaires:	03
Total response rate:	62%

Table-8 blow shows the responses of fifty six respondents against each question.

Table 8 - Questionnaire Results

Information Category	Questions	Very Important	Important	Moderately Important	Unimportant	I'm Uncertain
Environment Data	1- No. of user	28	17	8	3	0
	2- Target platform	27	23	6	0	0
	3-Organizational structure of software development company.	15	12	26	3	0
	4- Communication Interfaces	24	24	8	0	0
	5- Standards and policies followed by Software development company.	15	16	25	0	0
	6- Maturity level of Software development company.	5	34	7	9	0
Project Type	7- Application Type	20	27	9	0	0
	8- Type of target organization	18	7	31	0	0
	9- Non-Analogy Size	19	24	12	0	0
	10- Analogy Categorization	18	15	12	11	0

	11- Algorithmic Complexity	24	23	9	0	0
	12- Defect Rate	21	15	12	8	0
	13- No. of Modules	27	23	6	0	0
Team Productivity	14- Team Size	25	21	5	5	0
	15- Programming Experience	18	27	11	0	0
	16- Tool used	12	32	12	0	0
	17- Customer Participation	20	8	28	0	0
Domain Knowledge	23- Domain Experience	21	23	6	6	0
	24- Development platform maturity	25	22	9	0	0
Technology	25- Technological constraints	33	23	0	0	0
	26- Language Experience	12	38	3	3	0
Development Process	27- Maturity of SPM process	5	33	18	0	0
	28- Development Process Type	5	24	22	5	0
	29- SDLC	5	37	9	5	0
Resource Availability	18- Activities Involved	13	31	0	8	3
	19- Resources Required	16	27	7	3	3
	20- Reusable Component % Exist	15	30	8	3	0
	21- Breakthrough point	0	39	9	3	5
	22- Tools Required	25	14	5	8	0
		Yes		No		
	30- Histogram	26		30		
	31- Resource Assignment Matrix	51		5		
	32- Linear Responsibility Chart	41		14		
Customer	33- Satisfaction(Gant Chart)	51		5		

In the section below every category of information variables is discussed with its respective variable response percentage. Three respondents left question number 6, 9, 18 and 32 blank. So these three respondents were not included to calculate response rate of question number 6, 9, 18 and 32. Finally, analyzing the participants' response rate an ordered list of all the information variables is given with their respective response percentage.

4.2 Survey Results Analysis

4.2.1 Environment Data

The x-axis of the graph shows information variables that constitute development and deployment environment of the software project. The y-axis of the graph represents importance level of information related to x-axis variables on a percentage scale. Graph 1 shows the response rate for information variables categorized under the Environment Data.

Graph 1 - Representing percentage of responses for questions related to Environment Data

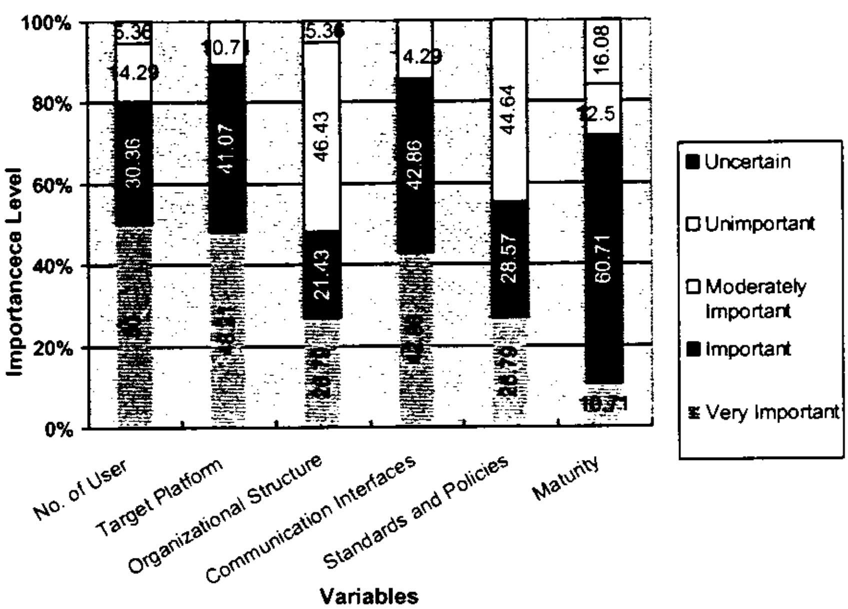


Table-9 below represents the summary of information represented by Graph-1 in descending order based on response percentage.

Table 9- Response Summary of Environment Variables

Environment Variables	Very Important + Important	Moderately Important	Unimportant
Target Platform	89.28	10.71	0.00
Communication Interfaces	85.72	14.29	0.00
No. Of User	80.36	14.29	5.36
Maturity	71.42	12.50	16.08
Standards And Policies	55.36	44.64	0.00
Organizational Structure	48.22	46.43	5.36

The information concerning *Target Platform* for which software should be deployed attained a response of approx 90% (blue part of 2nd vertical bar + plum part of 2nd vertical bar). This high response rate proclaims the information concerning *Target Platform* as mandatory to make judgment about software estimates.

The information related to external and internal Communication Interfaces of software development organization obtained 85.72% response. It shows the tendency of experts considering this information mandatory to make judgment for their software estimate.

The knowledge about how many users will run the application is considered an important factor in software development. More than 80% respondents regarded the information related to No. of Users mandatory to make judgment for their software estimates.

Information about Maturity level and Standards & Polices of software development organization received 71.42% and 55.36% responses respectively. It shows the desirability of expert to have this information for making judgment about their software estimates. Only 48.22% respondents considered the organizational structure of the software company for making judgment about their estimates. This low response rate shows that the information about the Organization Structure is optional for making judgment estimates.

The information related to Maturity level of software development organization is not as compulsory as information about first three variables. The reason might be that the maturity level is more concerned with the development and management processes improvements rather than expert estimation (Dewaynee et al, 1984). It shows that the accuracy of expert estimates in software development organizations, which have achieved a certain maturity level, may not very far from less mature organizations. However, mature development and management processes of prior organizations would help them to meet their estimates (Molokken-Ostvold & Jorgensen, 2005) yet further research is required to validate this assumption.

4.2.2 Project Type

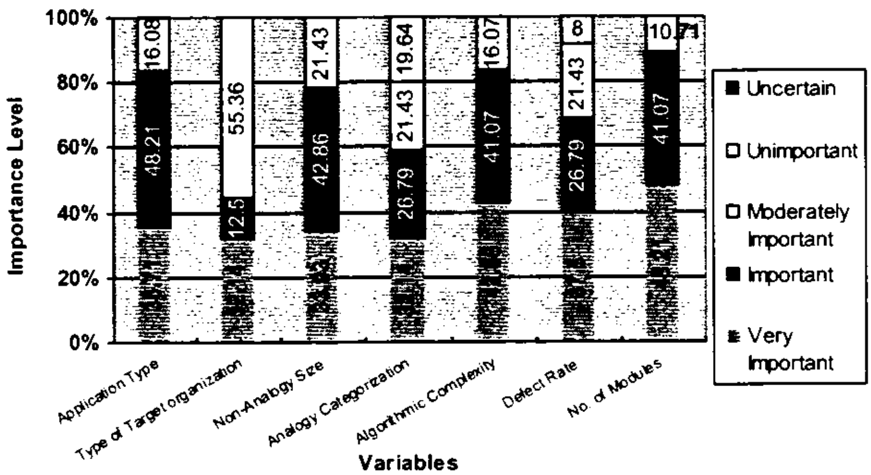
Table-10 shows the response percentage for variable categorized under project type.

Table 10- Response Summary of Project Type Variable

Project Type Variables	Very Important + Important	Moderately Important	Unimportant
No. of Modules	89.28	10.71	0.00

Algorithmic Complexity	83.93	16.07	0.00
Application Type	83.92	16.08	0.00
Non-Analogy Size	76.79	21.43	0.00
Defect Rate	64.29	21.43	8.00
Type Of Target Organization	44.64	55.36	0.00
Analogy Categorization	58.93	21.43	19.64

Graph 2- Representing percentage of responses for questions related to Project Type



The modules development is the realization of thoughts or/and ideas in software development, hence a resource consuming activity. The information related to No. of Modules which should be developed for the software, attained approx 90% response rate. It shows that the experts consider this information mandatory for making judgment about their software estimates.

The information related to complexity level of the algorithms that should be developed for software, attained a response rate of 83.93%. Similarly information regarding type of application software that should be developed attained a response rate of 83.92%. Thus more than 83% respondents considered the information related to these variables mandatory for making judgment about their estimates.

The information related to the *non-analogy* size of software got 76.79% response. This response reveals that experts desire to have the information related to the size of software for making judgment about estimates. Instead of size (FP, OP, USP etc) most probably they visualize through their experience, the *number of modules* and use this information for making judgment about estimates.

The Information related to defect occurrence rate, type of target organization (organization where the software would be deployed and analogy categorization attained response rate of 64.29%, 58.93% and 44.64% respectively. This low response notifies the information related to these three variables as optional for judgment estimates.

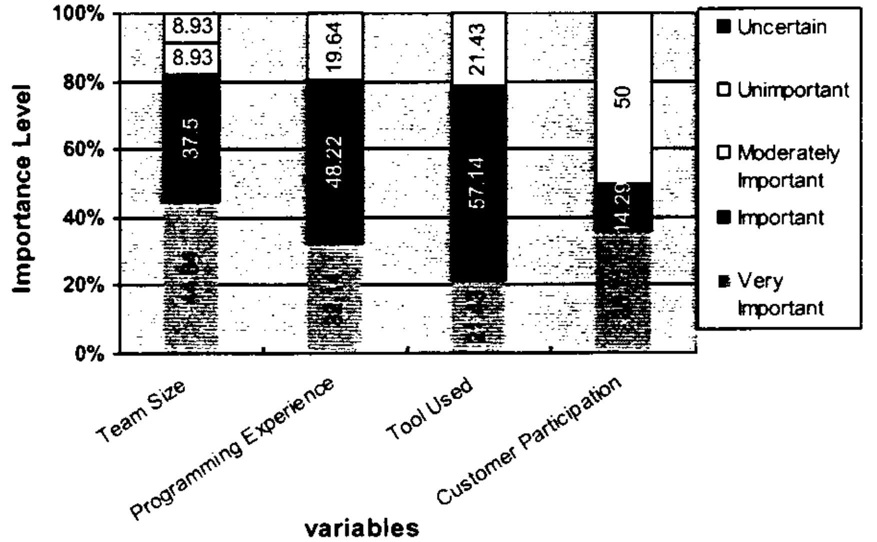
4.2.3 Team Productivity

Table-11 shows the response percentage for variables categorized under the head of Team Productivity.

Table 11- Response Summary of Team Productivity Variables

Team Productivity Variables	Very Important + Important	Moderately Important	Unimportant
Team Size	82.14	8.93	8.93
Programming Experience	80.36	19.64	0.00
Tool Used	78.57	21.43	0.00
Customer Participation	50.00	50.00	0.00

Graph 3- Representing percentage of responses for questions related to Productivity



When asked about the team productivity, 82.14% respondents notified the information regarding team size mandatory for Expert Judgment estimates. It shows that experts mainly consider the size of the team that will be involved in software development to make judgment about the productivity of the team.

It is also considered mandatory to have the information about programming experience of team members. 80.36% respondents use this information for their judgment estimates.

The 78.57% respondents desire to have the information regarding tools used by the team for software development. Although the tools used in software development are considered vital but expert estimators did not considered the information regarding tools used in development as important. The assumed reason for that is the dependency of tool productivity upon the experience and expertise of the team members.

The information regarding customer participation level attained only 50% response rate. This shows that the experts optionally include this information to make judgment estimates. However, customer participation in development of software can clarify the requirements and provides early feedback for better control (Jørgensen & Sjøberg, 2004).

4.2.4 Domain Knowledge

Graph 4 shows the response percentage for variable categorized under the head of Domain Knowledge.

Graph 4- Representing percentage of responses for questions related to Domain Knowledge

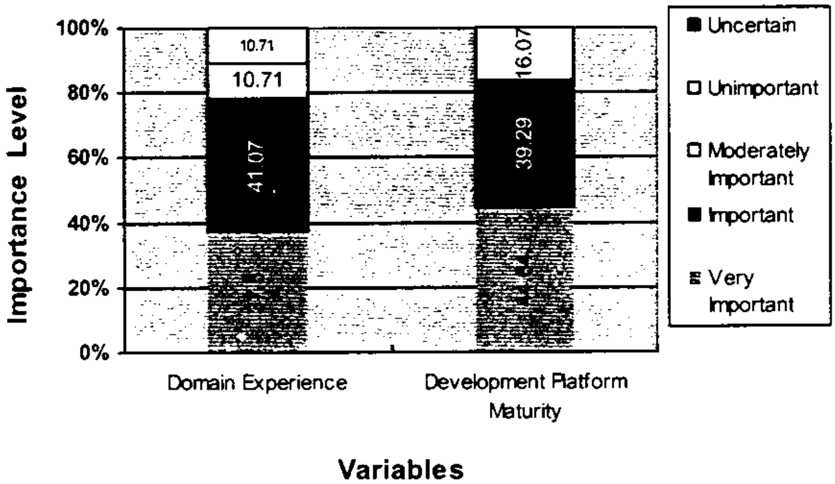


Table 12- Response Summary of Domain Knowledge Variables

Domain Knowledge Variables	Very Important + Important	Moderately Important	Unimportant
Development Platform Maturity	83.93	16.07	0.00
Domain Experience	77.57	10.71	10.71

Questionnaire response in Table-12 reveals that the information related to maturity level of platform used for software development is frequently used by experts for making software estimates. 83% respondents deemed this information necessary for making judgment about their estimates. Mature software development platforms provide many built-in libraries for different tasks such integration, database management etc. which plays an important role to reduce resources consumes for development. Therefore experts consider it mandatory to have the information about the maturity of development platform.

78% response for information related to Domain Experience is not significantly lower than the information regarding Development Platform Maturity except a response rate of 10.71% for *unimportant,* affects its importance negatively as compared to Development Platform Maturity. The deliberate training providing activities which can increase understanding of different concepts of domain are assumed to reduce the importance of information variable.

4.2.5 Technology

Technology used to develop software is considered important by the experts. Responses for Q25 and Q26 confirm the importance of technology related information for Expert Judgment estimate.

Graph 5- Representing percentage of responses for questions related to Technology

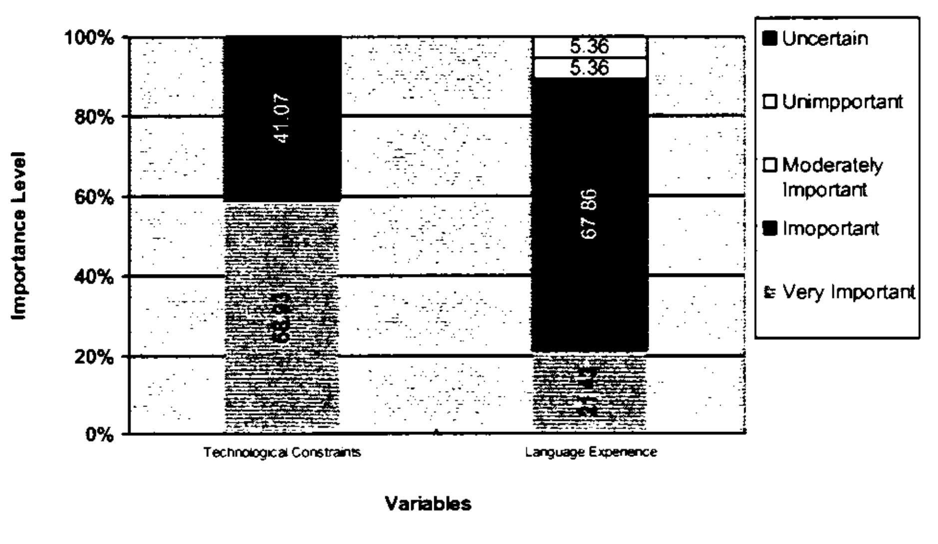


Table 13- Response Summary of Technology Variables

Technology Variables	Very Important + Important	Moderately Important	Unimportant
Technological Constraints	100	0.00	0.00
Language Experience	89.29	5.36	5.36

The technological constraints related information is considered compulsory for Expert Judgment estimation. This is evident by the 100% response in Table -13 that technological constraints and limitation related information play an important role Expert Judgment estimates. The results highlight that the strengths and weaknesses of the technology chosen to develop a project is also considered for estimating that project. We assumed the reason is its relation to risks and unexpected interrupts which dramatically affect the estimates. Secondly, the information related to language experience attained a response rate of 89.29%. This means that experts observe the language experience of the team members as mandatory information for making judgment about software estimates.

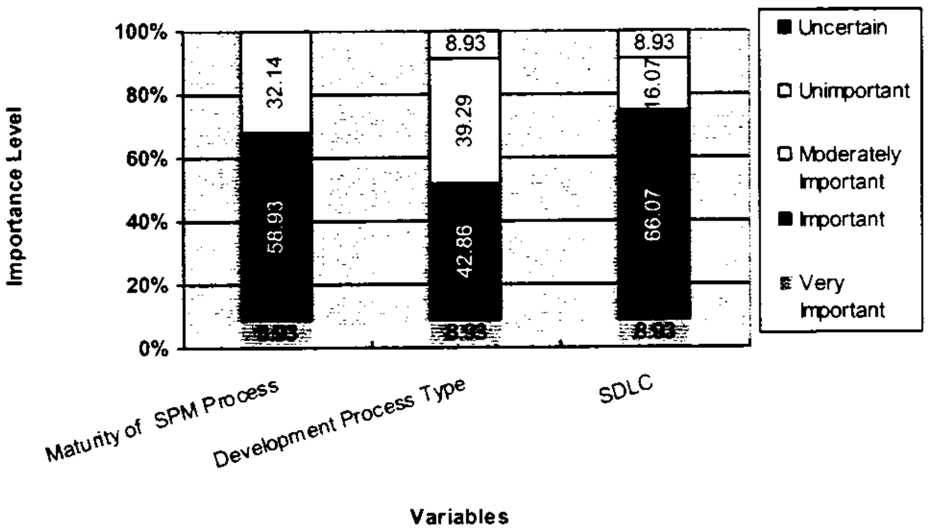
4.2.6 Development Process

Table-14 shows the response percentage for variables categorized under the head of Development Process.

Table 14- Response Summary of Development Process Variables

Development Process Variables	Very Important + Important	Moderately Important	Unimportant
Maturity of SPM process	67.86	32.14	0.00
SDLC	75.00	16.07	8.93
Development Process Type	51.79	39.28	8.93

Graph 6- Representing percentage of responses for questions related to Development Process



The response for Maturity of SPM process denotes it as desirable information for expert estimation. 67.86% respondents consider the maturity of software management process for their estimates judgment.

It is manifest that mature management process can handle the project more effectively that's why the experts consider the information related to maturity of SPM process significant to make judgment about their software estimate.

The Development Process Type and SDLC with a response rate of 51.79% and 75% respectively also have some significance but not as significant as SPM Process. One understanding for that is Development Process Type and SDLC depend on the Software Management Process. Software Management Process lies at the center of software development. It controls and keeps the development process on track. That's why experts consider it more important than development process and SDLC.

It can also be observed that SDLC got higher response as compared to development process. Jorgenson (Molokken-Ostvold & Jorgensen, 2005) notifies the superiority of flexible models over fixed ones to meet estimates. The questionnaire results go in favor of Jorgenson finding (Molokken-Ostvold & Jorgensen, 2005) that light weight process instead of heavy weight process affects the outcome of the project but this is optional for making judgment about estimates of a project.

4.2.7 Resources

Graph-7 shows responses of information variables about resource *required and available.*

Graph 7- Representing percentage of responses for questions related to Resource Availability

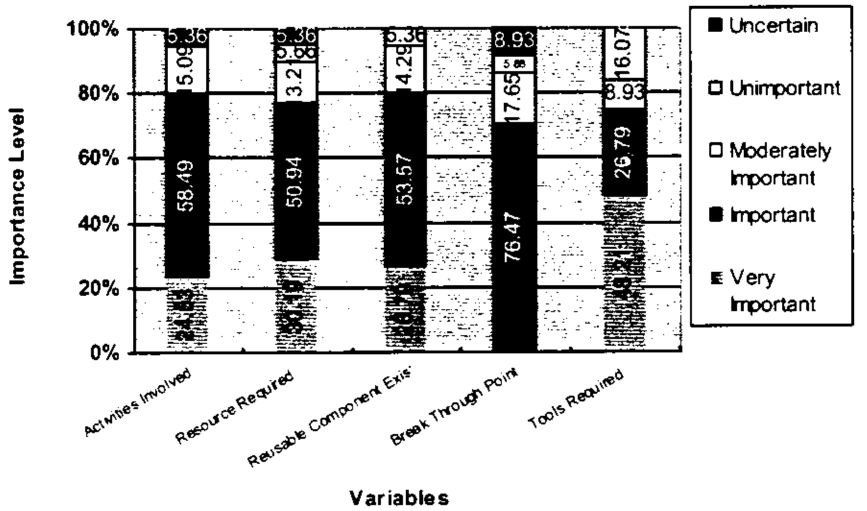


Table 15- Response Summary of Resource Variables

Resource Variables	Very Important + Important	Moderately Important	Unimportant
Resource Required	81.13	13.21	5.66
Activities Involved	83.02	0.00	15.09
Rȩusable Component Exist	80.36	14.29	5.36
Break through point	76.47	17.65	5.88
Tools Required	75.00	8.93	16.07

When asked about resources, 81.13% respondents in Table-15 consider the information regarding resource required to develop the software significant for making software estimates. This information is mandatory especially to make judgment about the time estimates of the software project.

The 80.36% response rate reveals the need to know whether Reusable Components exist. The basic purpose of reusable components is to decrease the software cost by reducing time and efforts required for development. Therefore the experts considered it compulsory to use this information for their judgment estimates.

Breakthrough point is the measure of the rework required prior to use a component in software development. 76.47% respondents desired to use this information to make their judgment estimates. It notifies that expert need to know how much rework would be required in reusable component to use them in software development but this information is rarely managed. The response rate for *uncertain* is 8.93%. A follow-up interview reflects the unfamiliarity of respondents with the term *Breakthrough Point*. Graph 8 shows the artifacts experts describe important to get information about the resource availability.

Graph 8- Representing percentage of responses for questions related Resource Availability

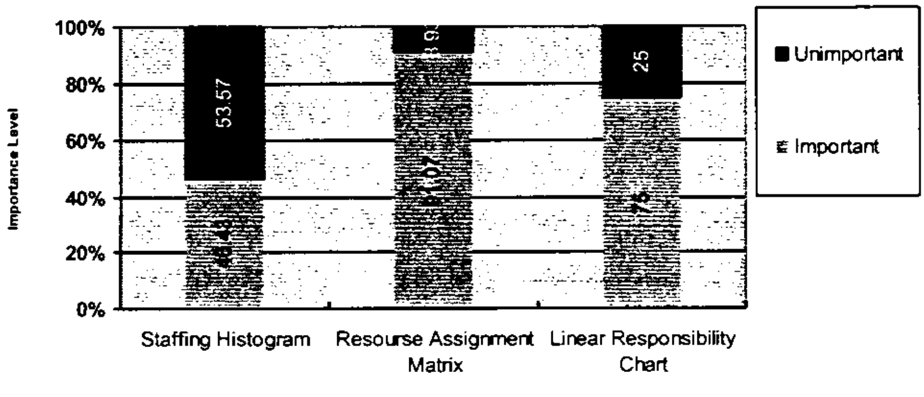


Table 16- Response Summary of Resource Variables

Resource Variables	Important	Unimportant
RAM	91.07	8.93
LRC	75.00	25.00
Histogram	46.43	53.57

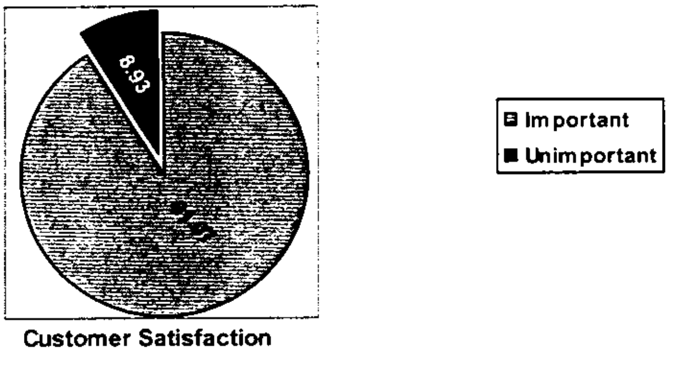
The Resource Assignment Matrix (RAM) is the most important artifact with a response rate of 91.07%. RAM can be used to know which resources are assigned tasks, when they will be available, total resource pool size and currently available resources. This can be the reason that experts consider it compulsory to use RAM for making judgment about their estimates.

Linear Responsibility Chart with 75% response rate and Staffing Histogram which represent the staffing plan with only 46.43% response rate are considered desirable and optional respectively.

4.2.8 Customer

Graph-9 shows the importance of customer satisfaction information for estimation.

Graph 9- Representing percentage of responses for questions related to Customer Satisfaction



91.07% respondents consider using customer satisfaction related information for their judgment estimates. It shows that experts highly desire to meet the customer delivery and budget related expectations. Though customer expectations can ruin the accuracy of estimate (Jorgensen, 2005, 2004), their high demand for estimation work required to uphold these information intact. Hence we recommend taking good care about reality of expectations.

The questionnaire also contained an open ended question. It was assumed that the response to the question will bring some new or missing information but only three (03) out of fifty six (56) participants responded to this question. One participant stated to have additional users and non-functional requirements related information. A couple of participant required security constraints and risk related information to produce more accurate estimates. This low response shows the completeness of the questionnaire that is the questionnaire covered most of the factors considered vital for Expert Judgment.

4.2.9 Questionnaire Result Categorization

Table-17 summarizes all variables in descending order on the basis of their response rate. Furthermore, as per the empirical results, it is assumed that variables with response rate 100% to 80% are more *significant* for Expert Judgment Estimation as compared to other listed variables.

Table 17- Results Summary with response rate

S. No.	Variables	Important	Moderately Important	Unimportant	Information Category
1	Technological Constraints	100	0.00	0.00	Technology
2	Resource Assignment Matrix	91.07	-	8.93	Resource

3	Customer Satisfaction	91.07	-	8.93	Customer
4	No. of Modules	89.28	10.71	0.00	Project Type
5	Target Platform	89.28	10.71	0.00	Environment
6	Language Experience	89.29	5.36	5.36	Technology
7	Communication Interfaces	85.72	14.29	0.00	Environment
8	Development Platform Maturity	83.93	16.07	0.00	Domain Knowledge
9	Algorithmic Complexity	83.93	16.07	0.00	Project Type
10	Application Type	83.92	16.08	0.00	Project Type
11	Activities Involved	83.02	0.00	15.09	Resource
12	Team Size	82.14	8.93	8.93	Team Productivity
13	Resources Required	81.13	13.21	5.66	Resource
14	Reusable Component Exist	80.36	14.29	5.36	Resource
15	No. Of User	80.36	14.29	5.36	Environment
16	Programming Experience	80.36	19.64	0.00	Team Productivity
17	Tool Used	78.57	21.43	0.00	Team Productivity
18	Domain Experience	77.57	10.71	10.71	Domain Knowledge
19	Non-Analogy Size	76.79	21.43	0.00	Project Type
20	Break through point	76.47	17.65	5.88	Resource
21	Linear Responsibility Chart	75.00	-	25.00	Resource
22	SDLC	75.00	16.07	8.93	Development Process
23	Tools Required	75.00	8.93	16.07	Resource
24	Maturity	71.42	12.50	16.08	Environment
25	Maturity of SPM process	67.86	32.14	0.00	Development Process

26	Defect Rate	64.29	21.43	8.00	Project Type
27	Analogy Categorization	58.93	21.43	19.64	Project Type
28	Standards And Policies	55.36	44.64	0.00	Environment
29	Development Process Type	51.79	39.28	8.93	Development Process
30	Customer Participation	50.00	50.00	0.00	Team Productivity
31	Organizational Structure	48.22	46.43	5.36	Environment
32	Staffing Histogram	46.43	-	53.57	Resource
33	Type of target organization	44.64	55.36	0.00	Project Type

4.3 Experiment

Table-18 represents summary of experiment execution.

Table 18- Experiment Execution and Experts Estimates

Phase I					
Groups	**Individuals**	**Task**	**Effort (Man Hours)**	**Time (Calendar Months)**	**Cost (PK RS Millions)**
GR1	Estimator 1	PT1	21000	8	15
	Estimator 2		19712	14	15
GR2	Estimator 3	PT2	2500	4	2
	Estimator 4		1440	2	0.15
Wash Away Time					
Phase II					
GR1	Estimator 1	PT2	3200	3	1.5
	Estimator 2		4400	5	3
GR2	Estimator 3	PT1	19680	10	30
	Estimator 4		27000	6	5.1

4.4 Experiment Analysis

Differences between estimates provided by experts for project tasks and their actual figures calculated and converted to percentage using following formula:

Formula: **% Difference = (Estimate - Actual) /Actual *100**

The purpose of this calculation was to clarify the level of improvement in accuracy of judgment estimates. A comparison of percentage difference between informed and uninformed judgment estimates is shown in Table-19. The (-) sign in Table-19 shows underestimated values.

Table 19- Informed and Uninformed actual, Estimate and Difference%

Effort (Man Hours)					
		Actual	**Estimated**	**Difference form actual**	**Difference%**
PT1	Estimator 1	12936	21000	8064	62.34
	Estimator 2		19712	6776	52.38
	Estimator 3		19680	6744	52.13
	Estimator 4		23000	10064	77.80
PT2	Estimator 1	1050	3200	2150	204.76
	Estimator 2		4400	3350	319.05
	Estimator 3		2500	1450	138.10
	Estimator 4		1800	750	71.43
Time (Months)					
		Actual	**Estimated**	**Difference form actual**	**Difference%**
PT1	Estimator 1	7	8	1	14.29
	Estimator 2		14	7	100.00
	Estimator 3		10	3	42.86

	Estimator 4		6	-1	-14.29
PT2	Estimator 1	2	3	1	50.00
	Estimator 2		5	3	150.00
	Estimator 3		4	2	100.00
	Estimator 4		2	0	0.00
Cost (in Millions)					
		Actual	**Estimated**	**Difference form actual**	**Difference%**
PT1	Estimator 1	4.2	15	10.8	257.14
	Estimator 2		15	10.8	257.14
	Estimator 3		30	25.8	614.29
	Estimator 4		5.1	0.9	21.43
PT2	Estimator 1	0.235	1.5	1.27	538.30
	Estimator 2		3.00	2.77	1176.60
	Estimator 3		2.00	1.77	751.06
	Estimator 4		0.15	-0.09	-36.17

It is found that an expert provided more accurate estimates i.e. less differed from the actual, when he is provided with top *significant information variables* of Table-16. The effort estimate given by experts for PT1 revealed 52 -77 % distance from the actual of PT1, but same experts provided effort estimate for PT2 which decentralized 71-319 % from PT2's actual. It notified that experts utilized the significant information variables in their judgment process and it facilitated them to provide more accurate estimates.

Similar results are found in case of cost estimates. The distance for provided cost estimates of PT1 ranges form 21- 614 % but it ranges from 36% up to 1176% for PT2. Considering these upper and lower limits notifies the importance of provided information variables and estimate's improvement caused by these variables. Although expert mainly based their cost estimate on calculated/estimated time but they used information variables as well. Estimator-1 (E1) and estimator-2 (E2) estimated eight months and fourteen months as required time and twenty-one thousand and nineteen thousand seven hundred twelve man hours as required efforts for PT1. However they provided identical cost estimate i.e. fifteen millions. This finding notifies that experts making judgment about estimates consider explicit information variables which direct the outcome of their judgment.

Finally, the *difference* between actual and estimated time ranges from 14 – 100 % for PT1 and 50 – 150 % for PT2 which again reveals that information variables provided with PT1 have facilitated experts to make better judgment estimate of software project.

Estimator-4(E4) showed a considerable accuracy for estimates of PT1 and PT2. All estimators received consistent information and same project tasks but the work level of E4 was lower than others. He was more experienced to estimate small to medium size projects than large size projects. Since PT1 and PT2 were also medium to small size projects, this might enable E4 to make better judgment compared to other three estimators. Matching work level of estimator can be an important factor to achieve accurate estimates. However further investigation is required to qualify this finding.

An analysis is performed to check whether a significant difference exists between percentage differences of informed and uninformed judgment estimates. The *t-test* is used to assess the validity of null hypothesis. Data used for t-test is given below:

Table 20- Percentage Difference of PT1 and PT2 Estimates

Effort Difference %		
	Uninformed	**Informed**
Estimator 1	204.76	62.34
Estimator 2	319.05	52.38
Estimator 3	138.10	52.13
Estimator 4	71.43	77.80
Time Difference %		
Estimator 1	50.00	14.29
Estimator 2	150.00	100.00
Estimator 3	100.00	42.86
Estimator 4	0.00	14.29
Cost Difference %		
Estimator 1	538.30	257.14
Estimator 2	1176.60	257.14
Estimator 3	751.06	614.29
Estimator 4	36.17	21.43

4.4.1 T-test Analysis

MS-Excel tool is used to perform t-test analysis and 2-Sample Equal Variance t-test is implemented for effort, time and cost estimates difference percentages of PT1 and PT2. Table-21 shows the calculations for t-test values;

Table 21- Statistical t-test implementation

t-test for Effort		
	Uninformed	*Informed*
Mean	183.335	61.1625
Variance	11148.84	145.6371
Observations	4	4

Pooled Variance	5647.24	
Df	6	
t Stat	2.299166	
P(T<=t) two-tail	***0.06117***	
t Critical two-tail	2.446914	

t-test for Time

	Uninformed	*Informed*
Mean	75	42.86
Variance	4166.667	1632.49
Observations	4	4
Pooled Variance	2899.578	
Df	6	
t Stat	0.844099	
P(T<=t) two-tail	***0.430976***	
t Critical two-tail	2.446914	

t-test for Cost

	Uninformed	*Informed*
Mean	625.5325	287.5
Variance	224796.7	59809.47
Observations	4	4
Pooled Variance	142303.1	
Df	6	
t Stat	1.267262	
P(T<=t) two-tail	***0.252025***	
t Critical two-tail	2.446914	

According to t-test, the resulted t-stat values (for effort, time and cost) are less then t-critical values. Similarly the probability that the null hypothesis is true was greater than alpha i.e. p-value > 0.05. Both the finding of t-test analysis provide evidence that a significant difference exist between PT1 and PT2 estimates which rejects the *null hypothesis* and confirmed the *alternative hypothesis* Furthermore, analyzing the deviation of estimates *means and variances* with respect to PT1 and PT2 actual shows that the information provided with PT1 affected Expert Judgment in a positive manner.

4.5 Follow-Up Interview

Figure-4.1 represents the descriptive model constructed to provide better understanding of the Expert Judgment process. Identical judgment process is used for informed and uninformed tasks i.e. PT1 & PT2 but significant improvement in accuracy of judgment estimates is found when experts are provided the required information variables. They used provided information variables in different phases of this descriptive model which enable them to predict the software estimates more accurately as shown in Table-19 & 20. The figure-4.1(page # 75) showing descriptive model for Expert Judgment is given below:

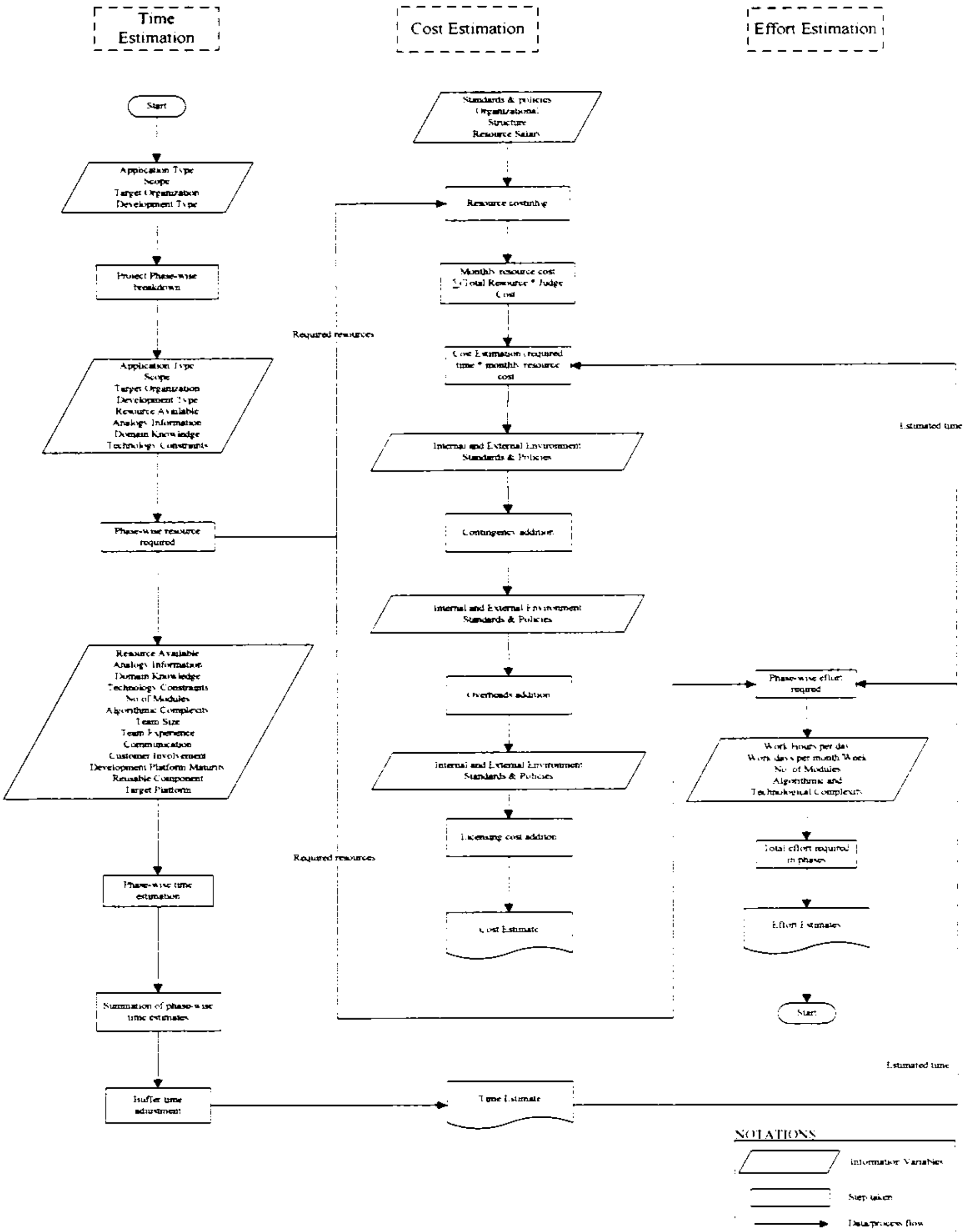
4.5.1 Model Explanation

The descriptive model is divided the in three layers to increase readability and understanding.

- Time Judgment
- Cost Judgment
- Effort Judgment

Multiple steps and information variables are used on every layer. However, time judgment layer occupies the central position. It contains the starting point of judgment and provides *time estimates* to *cost* and *effort judgment* layers. The information variables used in different steps are represented by object and estimates produced are represented by object.

Figure-4.1 Descriptive Model for the way expert judge effort, time and cost estimates



Commonly, the effort anticipated as the basis for time and cost estimates. However, the expert estimators in this study were more conscious about the time required for project. Expert Judgment was started by dividing the project in phases. Then different information variables were used to make judgment about the time required for each phase and then summation was performed to have total time required. Then the total time was used for cost and effort estimation.

Similarly resource costing was done based on monthly salary. Experts in this study first predict the time required completing the project then figure out the monthly cost for resources based on their salary and other standards & policies of organization and finally calculate the cost. Thus information regarding respective salary of resources is found important and in absence of this information experts assumed these figures according to their levels. In our experiment the estimator-4 provided the most accurate cost estimates. He is a senior software engineer with six (6) years of estimation related experiences. The remaining three estimators are working on higher positions with more than ten (10) years of related experience. We assumed that the cost estimates provided by estimator-4 were significantly closer to actual because Estimator-4 figured out the salary of different resources according to his level.

4.6 Discussion

The software development is a knowledge intensive process and multiple type of information is used and generated in (Yenduri et al, 2007). The objective of research work was to determine the information involved in Expert Judgment because maintaining this information has the potential to improve accuracy of judgment estimates. However the research work does not have any intension to give a parametric model.

The study found a set of information variable which are considered mandatory or desirable by the experts to make judgment based software estimates. However these are not the only information variables involved and there could be personalized and tacit information involved in Expert Judgment but eliciting this behavioral information was not in *scope* of our study. A study based on behavioral sciences would be more suitable for eliciting such information.

Although the customer budgetary information is considered important for costing yet this information variable is not directly included in study because the budgetary expectations of customer have the potential to ruin the realism of the estimates (Jorgensen, 2004, 2005). It diverts the attention of experts to meet the budgetary expectation instead of focusing on real estimates.

Simple statistical formulas are used to calculate the estimates deviation from their actual for the experiment results. Tabular comparison of calculated deviation and their means, confirms the *alternative hypothesis*. Although the experiment results report a significant improvement in accuracy of judgment estimates but the achieved accuracy level is not high enough for software industry satisfaction.

A descriptive model is presented for the way experts make judgment which provides a preliminary understanding of the Expert Judgment estimation work. It is not a prescription for the judgment work in any manner.

The study was to elicit the information variables that have the potential to improve the Expert Judgment. Experiment was conducted to check the credibility of information variables and their affect on Expert Judgment.

Although testing a project is a resource consuming activity but Defect Occurrence Rate in a project is not considered important for making judgment about estimates. The defect rate data is not usually maintained due to the cost factors and even if available experts infrequently consider the defect rate during judgment estimation. However, we assume that considering defect interrupts can improve the accuracy of expert estimates.

Finally experts judge the productivity of the team mainly through the team size. Although they realize that large team can cause management and communication issues, based on response rate of team size information variable, it seems that they ignore this fact to make judgment estimates.

CHAPTER 5: CONCLUSION

5.1 Research Summary

Software industry requires accurate software estimates as they play a versatile role in software development. Whether we talk about Bidding, planning budgeting or measuring success, estimates are at the center of all of these activities. Expert Judgment is the most widely used estimation method in industry. However little research is conducted to understand and improve the Expert Judgment estimation.

This study is focused on eliciting the information variables that are considered mandatory or desirable by the experts to make judgment estimates. A questionnaire survey is conducted from experts working in Pakistan software industry and based on the empirical results a list of significant information variables is prepared along with their respective response rates.

In next step a crossover experiment is performed with the objective to check the credibility of elicited information variables i.e. *providing information variable explicitly affect the accuracy of Expert Judgment or not.* Four expert estimators (divided into two groups) alternatively make judgment estimates for project task-1(PT1) and project task-2(PT2). PT1 was provided with information variables and PT2 was without that information. A statistical analysis of estimates as well as means of estimates (see Table-19, 20) reported a positive effect on accuracy of expert estimates.

The final objective of research work was to understand the way experts make judgment estimates in Pakistan software industry. A descriptive model is proposed, based on follow-up interviews, for that purpose. A transcript was generated for each interview and a textual analysis was performed to extract the *steps taken* and *information variables used*. The common steps and information variables for each step are separated and generated transcripts are thoroughly read to check the flow of steps. As a result a descriptive model is constructed to provide a better understanding of the way experts make software estimates.

An unexpected difficulty faced during the transcript generation was the translation of signs and gestures. However analyzing the *context* helped us to overcome this difficulty.

5.2 Research Work Limitations

The experiment tasks 1 and 2 belonged to telecom and accounting domains. However, the experts selected for the experiment did not have thorough experience related to these domains but the discussion session prior to task estimation and their estimation related experience deemed to overcome this limitation.

The constructed model is based on the interviews of only four expert estimators. This small number is insufficient to claim generalization for the model. However, analyzing common *steps taken* and *information variable*s used gives significant credibility to our model to understand Expert Judgment. Further research is required for the validation of that model.

The perceptions and experiences of experts differ significantly based on their environment e.g. culture, geographic location, traditions etc. Model in this study specifically represents the expert's way of judgment estimates in Pakistan software industry. We are uncertain whether the experts in other countries follow the same way.

5.3 Conclusion

Expert estimation is referred to as prediction of the future estimates (Mannhart et al. 2007; Hughes, 1996) but the information required to predict future estimates is unknown. Furthermore huge information is generated during the software development process. This information is rarely maintained for future use. The basic reason is the cost of maintaining this huge information. Yenduri et al (2007) reported that 5% to 10% of total project cost is required to maintain this information.

The basic objective of this study was to elicit real time information used for making judgment estimates. It increases the accuracy of predicted estimates and also reduces the cost of information maintenance by highlighting the top required information variables among less required ones.

The experiment results (Table- 19, 20 & 21) evidence improvement in accuracy of judgment estimates. However, the resulting improvements are not enough to satisfy the needs of software industry, because the accuracy level that the software industry requires is higher than that, which is achieved through this research work. Hence further research is needed to improve the Expert Judgment and to achieve required levels of estimate accuracy.

5.4 Contribution

This study presents a list of information variables with their respective response rate. Empirical evidence is also provided that explicitly providing these information variables to experts improves the accuracy of judgment estimates. The descriptive model, presented in this research study, has potential to serve as a base to understand and improve the way experts make judgment estimates. There exists no model for Expert Judgment in software engineering literature. This descriptive model and the list of information variables is a modest *contribution* towards fulfilling the gap in Expert Judgment estimation literature.

5.5 Future Work

The future work will generalize the model by analyzing the way experts make judgment estimates. We will involve large number of experts from different places and discover their judgment process on different projects. We will also evaluate the accuracy of this model by verifying it in real-time environment.

6. References

APA- Referencing Format

Bekkers, W., Van de Weerd, I., Brinkkemper, S., & Mahieu, A. (2008). *The influence of situational factors in software product management: an empirical study. Software Product Management (ISWPM'08), page 41 - 48, IEEE.*

Boehm, B., & Chulani, S. (1998). *Software development cost estimation approaches-a survey. Annals of Software Engineering, 10, 1-4, 177-205, SpringerLink.*

Boehm, B. (1981). *Software engineering economics. Transaction on Software Engineering, SE-10(1), 4-21, IEEE.*

Collopy, F. (2007). *Difficulty and complexity as factors in software effort estimation. International Journal of Forecasting, 23(3), 469-471, ScienceDirect.*

Dewaynee. P., Nancy, A. S., Lawrenceg, V. (1984). *People, Organizations, and Process improvement. IEEE Software, 1994 IEEE*

Fraser, S., Boehm, B., Erdogmus, H., Jorgensen, M., Rifkin, S., & Ross, M. (2008). *The role of judgment in software estimation. Software Engineering Companion Volume 2009, 13 – 17, IEEE*

Furulund, K. M., & Molkken-Stvold, K. (2007). *Increasing software effort estimation accuracy using experience data, estimation models and checklists. Quality Software, page 342 – 347, IEEE.*

Fairly, R. E. (1992). *Recent advances in software estimation techniques. Software Engineering, 382 – 391, IEEE.*

Ferchichi, A., Bourey, J.P., Bigand, S.M., & Barron, M. (2006). *Design systems engineering of software products: implementation of a software estimation model. Computational Engineering in Systems Applications (CESA), 1181 – 1188, IEEE.*

Grimstad, S., & Jorgensen, M. (2007). *Impact of irrelevant information on estimates of software development effort. Software Engineering, 359 – 368 IEEE.*

Grimstad, S., Jorgensen, M., & Molokken-Ostvold, K. (2005). *Software effort estimation terminology: the tower of babel. Information and Software Technology, 48(4), 302-310, ScienceDirect.*

Hughes, R. T. (1996). *Expert Judgment as an estimating method. Information and Software Technology, 38(2), 67-75, ScienceDirect.*

Jorgensen, M. (2008). *Judgment updating among software professionals.* *www.simula.no.*

Jorgensen, M. (2007). *A preliminary theory of judgment based software project effort predictions. www.simula.no.*

Jorgensen, M., & Boehm, B. (2007). *Viewpoints: software development effort estimation: formal models and experts judgment.* *http://simula.no/research/se/publications/Simula.Se.321*

Jorgensen, M. (2006). *The magic step of judgment based software estimation. International Journal of Project Management, Elsevier.*

Jorgensen, M. (2005). *Practical guidelines for expert-judgment-based software effort estimation. Software and Computer Society, 22(3), 57 - 63, IEEE.*

Jorgensen, M., & Sjoberg, D. (2005). *Expert estimation of software development work: learning through feedback. www.simula.no* .

Jorgensen, M. (2004). *A review of studies on expert estimation of software development effort.* Journal of Systems and Software, 70(1-2), 37-60, ScienceDirect.

Jørgensen, M., Indahl, U., & Sjøberg, D. (2003). *Software effort estimation by analogy and ''regression toward the mean''*. The Journal of Systems and Software, 68(3), 253–262, Elsevier Science Inc.

Jørgensen, M. & D.I.K. Sjøberg.(2004). *The Impact of Customer Expectation on Software Development Effort Estimates*. International Journal of Project Management, 2004. 22: p. 317-325. 1-Oi, 2-Ej, 3-Ex, 4-StPr, 5-On

Jorgensen, M., & Sjoberg, D. I. K. (2002). *Software process improvement and human judgment heuristics*. www.simula.no .

Lionel, C. B., Khaled, E., Dagmar, S., Isabella, W., & Katrina, D. M. (1999). *An assessment and comparison of software cost estimation modeling techniques, Software Engineering, 313 – 323, IEEE.*

Mannhart, A., Bilgic, A., & Bertsche, B. (2007). *Modeling Expert Judgment for reliability prediction - comparison of methods. Reliability and Maintainability Symposium, 1 – 6, IEEE.*

Molokken-Ostvold, K., & Jorgensen, M. (2005). *A comparison of software projects overrun-flexible verses sequential development models. Transactions on Software Engineering, 31(9), 754 – 766, IEEE.*

Nasir, M. (2006). *A survey of software estimation techniques and project planning practices. Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD), 305 – 310, IEEE.*

Senn, S. (2002). *Cross-over trials in clinical research. John Wiley & Sons Ltd.*

Little, T. (2006). *Schedule estimation and uncertainty surrounding the cone of uncertainty, Software, 23(3), 48 - 54, IEEE.*

Uchitel. S., Kramer. J., Magee. J. (2003). *Synthesis of behavioral models from scenarios. Transaction on Software Engineering, 29, IEEE.*

Valerdi, R. (2007). *Cognitive limits of software cost estimation. First International Symposium on Empirical Software Engineering and Measurement, page 117 - 125, IEEE.*

Wu. S. I. K. (2006). *The quality of design team factors on software effort estimation. Service Operations and Logistics, and Informatics, 6 – 11, IEEE.*

Walkerden, F., & Jeffery, R. (1999). *An empirical study of analogy-based software effort estimation. Empirical Software Engineering, 4(2), 135-158. SpringerLink.*

Yenduri, S., Muagala, S., & Perkins, L.A. (2007). *Estimation practices efficiency: a case study. Information Technology (ICIT 2007), 185 – 189, IEEE.*

Yin, M., Peterson, J., & Arellano, R. R. (2004). *Software complexity factor in software reliability assessment. Reliability and Maintainability, 190-194, IEEE.*

7. APPENDIX - I: Associated information

I-a Questionnaire

Name: ______________________ Date: ________________

Designation: __________________ Experience (in years) ________

Organization: ____

Email: ______________________ Contact #. ______________

You have a software project in hand. You are asked to give initial estimates for that project. There exist different information categories that you required to make judgment estimates more accurately. Please tell us that if you have to make judgment estimates:

	How important in your opinion is to know:	Very Important	Important	Moderately Important	Unimportant	I'm Uncertain
1)	No. of users of software system to be developed?	☐	☐	☐	☐	☐
2)	The platform used by the users?	☐	☐	☐	☐	☐
3)	Organizational structure of software development company?	☐	☐	☐	☐	☐
4)	Inter and intra-organization communication interfaces of development company?	☐	☐	☐	☐	☐
5)	The standard and/or policies followed by software developing company?	☐	☐	☐	☐	☐
6)	The maturity level of the software developing company?	☐	☐	☐	☐	☐
7)	The type of software application you are going to develop?	☐	☐	☐	☐	☐
8)	The type of organization for which you are going to develop software system?	☐	☐	☐	☐	☐

9)	The size (LOC, FP, OP, UCP etc) of software to be developed?					
10)	Whether the software to be developed is a small, medium or large project?					
11)	The complexity of algorithm that you need to develop for software system?					
12)	How frequently the defect can occur during development of software system?					
13)	The number of modules has to develop for software system?					
14)	The size of the team involved in software development?					
15)	The programming experience of the team members involved in software development?					
16)	The tool(s) used by team involved in software development?					
17)	Level of customer participation would be available for the team during software development?					
18)	Which activities are involved in developing the software system?					
19)	Which resources are required for developing software system?					
20)	Whether reusable components exist to develop software system?					
21)	The break through point of reusable components that would be used to develop software system?					
22)	Whether the tools required for development are available or not?					
23)	About available domain experience to which the software belong to?					
24)	The maturity of development platform used for developing software system?					
25)	The technological constraints exist for developing software system?					
26)	The level of language experience of programmers developing software system?					
27)	The maturity level of SPM processes of company?					
28)	Whether heavy weight or lightweight development process is followed in company?					
29)	The process use to develop is fixed (waterfall style) or flexible (other than waterfall) development process?					

In your opinion is it important to consult		Yes	No
30)	Staffing plan histogram?	☐	☐
31)	Resource Assignment Matrix (RAM)?	☐	☐
32)	Linear Responsibility Chart (LRC)?	☐	☐
33)	Gant Chart?	☐	☐
34)	What other information do you use to make judgment?(Subjective)		

I-b Project Task 1

Open source Billing and Customer Support software is a suite of applications (Moderator, Rating, Billing applications). It will fulfill all basic requirements of back office Billing and Customer Care of any Wireless or Wire-line Service Providers. This project is about converting our existing Enterprise System to an Open Source, Web2.0 enabled and NGOSS compliant suite of applications that would facilitate the Cellular and Landline Service Providers and Integrated Telecom Service Providers. However the software will develop from scratch i.e. no any existing component or document will be used.

It will perform switch vendor independent, customizable and high-speed rating. It will help the operator in managing post-paid and pre-paid subscribers through a user-friendly interface. It will help the Service Providers in settling their accounts with partner operators through interconnect billing modules. It will also be capable of performing different analysis on data. It will let operator manage their warehouses' inventory and sale of products to customers. This will also provide an opportunity to Billing Service Companies to use part(s) or whole of our system to integrate with other systems to accomplish their objectives.

a) Billing, Customer Care, and OSS applications

CDR is an abbreviation for the term Call Detail Record. It is the data that is submitted by a subscriber's telephone system to a collection device linked to call accounting package. This record contains specific information about every telephone call including date. time. duration. digits dialed and trunk line. Often the term SMDR (Station Message Detail Record is used interchangeably with CDR). The "raw" data is generally interpreted and translated into a database for real time or scheduled processing and reporting. Data Storage software is generally used to collect the data and make it available for processing by call accounting software

b) Billing Systems for Telecom Service Provider

Almost every telecom service provider incorporates a billing system. These systems transform service usage into monetary compensation. They also provide ways to manage customer account information. They perform credit scoring. rate available products, accept payments. and reconcile credit and debit information with third parties. In a nutshell. telecom billing can be defined as the process which refers to how billable charges are processed. collected and invoiced. For most people the interaction with their phone company is infrequent, except for receipt of their phone bill. They typically only contact their provider for a few reasons. Maybe, they want to acquire a new service or product. It's possible that they need to disconnect an existing service. Or third, and most common. is the need to transfer service from one address to another.

For all of these services. the customer receives an invoice. The invoice comes at regular intervals, sometimes referred to as "bill periods". This invoice contains all of their usage. recurring. and one time charges. It includes any taxes and fees associated with services, provided for a given time period. These fees can be anything from hardware fee usage, to local, state, and federal taxes. In order for a provider to be able to generate an invoice, they must collect billable charges from the various telecom network or service elements. The network events which generate these charges are first collected, and then used to rate service usage.

c) Billable Charges

Billable charges may also come from third parties or affiliations which are providing the service. These charges may come in the form of usage charges, recurring charges, or taxes and fees. The charges are associated to particular customer. The customer is associated with their account information and billable charges. Billable charges, typically, come in some type of notification event for the billing system. For example, a third party such as a clearinghouse may send an event. This event indicates what charges are applicable for a customer within a given time period.

The billing system senses this event information to initiate business processes to properly rate and invoice the customer. Typically, the billing system would respond to multiple billing events. It would aggregate and categorize the resultant information. The billing system would use this information to generate the customer's phone bills.

d) Technical Requirements

Table 22- Technical Requirements

1.	Development Tools	J2EE, Java Scripts, XHTML, Flash / Flex frame work, Dream viewer
2.	Methodology	Ajax
3.	Application Server	Apache / Tomcat
4.	System Database	My SQL / Postgres SQL / Max DB / Fire Bird
5.	Operating System Platform	Any flavor of Linux

e) Required Testing

Extensive testing required.

f) Team Composition

Table 23- Team Composition

Designation	**Capability**	**No.**
Project Manager	Domain expert with 11 years of experience	1
Team leads	Good technical skills with 9 years of experience	2
senior developers	2- 5 years of development experience	12
junior developers	6 months – 2 years of development experience	4
Internees	At least CS Graduate	8
QA	2- 5 years of QA experience	3
Technical writers		2
DBA	3~ years of experience	1
Total		33

h) Other Information

Table 24- Other Information

Team Productivity	100-150 LOC/Hr
Domain Knowledge	Average
Size	1617 KLOC (Actual Effort * Avg. Productivity)
Team Communication	Good
Team Size	33 People
Resources Available	Partially
Development Process	XP
Environment Development Type Organizational Structure	 Product Matrix
Target Organizations	Telecom Companies
Innovative Work Level	High

I-c Project Task 2

Zarco Enterprise is a money exchanger and money transfer services providing company. It is working in collaboration with Western Union, a well-known UK based company.

The main services of Zarco Enterprise are to provide easy money transfer in most of the countries of the world. It also deals in money exchange in different leading countries currency. The major portion of their business is on the commission percentage and transfer charges of money. As Zarco Enterprise is foreign currency dealer, it also has affiliation with stock marketing. It deals in sales purchase of different companies shares in context of foreign currency ups and downs. It has to keep regular eye on the prices of foreign currency and shares of foreign currency dependent companies.Currently, my client Shakeel & Co. is not a member of their stock marketing team. However being a member of this team is one of the future concerns of my client.

In Pakistan, Zarco is covering a large area. It has franchisees in different cities of Pakistan. It has a well manage and predefine areas for franchisee offices location to reduce the competition ratio between its own partners as well as to provide easy access for customers. With respect to its functionality, Zarco is working in two basic areas:

a) Local Services

All the services it provide to the countrywide fall under the head of local services. It includes money transfer from one city to another and foreign currency exchange services. Its main source of income is the commission percentage for the transfer of money, it charges. It transfers money via two ways i.e.

i) Via Reference

In this mode, sender office makes a phone call to the receiver office. It gives a reference number, beneficiary name and ID number, amount to be transfer. It's an urgent money transfer mode. Its charges are more then transfer via message mode.

ii) Via Message

It's a regular message-sending mode. More then one messages are sent by fax, containing the required information and a branch specific code. Due to multiple money transfer messages, charges of this mode are less then via reference. This type of message is sent in the end of the day. The beneficiary can get the money on next working day.

Foreign currency exchange is also a local service. Currency of different leading countries sold/purchased. There exists a difference in sell and purchase price, which is recorded in term of income. Selling price normally remains higher from the purchasing price. All the purchased foreign currency can be send to main office to get the local currency with an increased percentage.

b) International Services

All the overseas services fall into the category of International Services. It includes money transfer to out of country and money exchange transfer to the out of country.

International services offer only 'via message 'money transfer. It is because overseas transfer requires proper information about the Zarco office, sender, receiver and amount. This data is submitted to Western Union through email and used for the calculation of loyalty percentage.

The money exchange and transfer is a two step process. First the money is exchanged for the desiring money that the client wants to transfer. This is a local service and the local office takes its charges. Then money is transferred to the desired country office via message.

c) Customer Relation

In customer area, Zarco Enterprise keeps tracking information about the regular customers and always tries to be in touch in them through.

- Greeting cards
- Regards, best wishes and current rates of the foreign currency through the e-mail

d) Currently Maintained Accounts

1) Customer Money Transfer

Managing the customer money transfer information is the basic requirement of Zarco Enterprise. Any mishandling or false information can be cause for a big loss. So all this information should be stored with full assurance and would be quickly accessible on demand.

2) Payable / Receivable Account

3) Currency Management

a) Local currency b) Foreign currency

4) Income/Expense Account

5) Cash Management

I-e Textual Analyzed Transcripts

Interview-1

Different resources are use on every level *(different phases are considered)*. For example 5 resources are used on level 5*(resource required for each phase is considered).* similarly if 20 resources are used then the monthly cost will be 2million, for 10 resources the monthly cost will be 1 million, for 5 resources it will be .5 million *(Monthly resource costing is done).*

In this way our monthly cost will be 2.3 to 2.5 million *(per month cost is estimated).*

Now use this to calculate the cost for 6 to 8 months. So for 6 months our cost will be (23*6) 1.38 million *(total cost is estimated).* Then we round it up to 150 *(a contingency is added).*

Another of our practice is to multiply the figure given by the developer by two. Then the other things will come around *(for time costing).*

Q: have you used the given information in your estimates?

Hmm! *(yes)* First of all, the team size, the team is considerably large (*team size variable information used).* So this information is used. They also have average domain *(Domain knowledge variable information used)* knowledge. Then the costing depends on the technology that you used such as you use java, Ajax etc *(Technology information used).*

Similarly your development become little difficult or more time consuming when you go for open source environment *(Technology variable information used).*

The average domain level increases your time and team communication which is good here will also affect your time *(communication variable information used).*

Q: Extensive testing is required for that project. Have you included the level of testing for your estimates?

The estimates I provided includes the alpha level testing. You can add 2 to 3 months for extensive testing *(testing information used).* You have also learned in SE that the code you write takes 20% of time. 80% of time is consumed to stable the software. Similarly in practice coding takes 20% of time and considerably large time is consumed in stabilizing and testing the software.

Q: One thing that I observed is that you moved up from the last on the form. First you write the cost, then you write time 6 to 8 months and you said that on the basis of these we calculate the efforts. So will you please explain how you calculated the last thing (cost) first?

On the basis of my experience it was my idea those 6 to 8 months. Keeping this 6 to 8 months in mind I develop this formula *(months * monthly cost) (total cost is estimated)*. Such as I have 6 to 8 months and my 1 resource on first level will cost one lack rupees. If 10 resources are there then 1 million *(resource costing)*. So 1 million into 6 months. Then I induct 20 medium level (level of resources is judge) of resources in the project. Each of them cost 50 k (resource cost according to his/her level). Then calculate the cost of 6 months *(resource cost * no. of resources * months)*. This costing is only for human effort. Hardware or any other cost is not included here *(other cost should be added)*. It is totally development cost.

Costing formula will always be around man hours. One of the ways is to multiply the calculated man hours with *(pr hour)* value *(effort used in costing)*. Such as you have 1000 man hours then 1000 * 10 dollars etc. but for that you must know the level of resources you have *(level of resource)*. For example the resources that consumed in the beginning will be of high level resources.

Q: why you consider the level of resources?

For different thing such as their productivity, knowledge, problem solving skills etc *(variable information used)*

Q: Means you consider the productivity of individuals to calculate the man hours?

Hmm! *(Yes)*

Q: you completed a process cycle from document reading to estimates. Will you please recall it?

1. Scope of the project *(Project Type information used).*
2. Complexity of the project *(Complexity information used).*
3. Then based on the experience duration of the project.
4. Monthly value of the resources multiplied by project time *(costing formula).*

Q: what things you consider in your mind thinking about complexity?

For example in call services the call recording, monitoring and backup are significantly complex tasks *(algorithmic complexity information is used).* Then we also need to consider it form technology dimension i.e. how complex is the implementation *(technological complexity information is used).*

Secondly open source has its pros and cons. Open source development normally takes more time *(technological complexity information is used).*

Q: as you refer the man hours. So have you considered the man hours required?

I judge a figure for man hours on the basis of our experience and by breaking the project *(decomposition/WBS).* Such as how much time will be required to build the online customer care module of that project. Similarly how much time will be required to develop the customer complaints module? How much time will be required to develop the database? *(Modules are consider and modules wise time estimated)* You associate them and calculate your cost according to no. of resources *(total no. of modules+ there required development time and no. of resources are used for costing).* For example the no. of men you have. If you come to know the people which will be involved in development then you build teams and assign them modules *(Team information used).*

Even if we estimate for bidding, we consider the included no. of modules too *(No. of modules information used).* For example you can see this file. It contains modules and the time required to develop each module. In this way we estimate *(example of prior estimation work).*

Q: Do you consider the required no. of modules first and resources required developing them?

Yes. *(Confirmation the use of no. of modules information to judge resource required for developing them)*

However I first try to make these modules independent modules. Then I draw these modules on time line to see which can run parallel and which can run independently *(Dependency of modules).* It is a complete process. Then you allocate your team *(Team Information).* Some resources can work in parallel but they can be in one team *(Team /Resource availability information).* So they can not work in parallel. Then you consider it in serial that is second work starts when first work completes. Accordingly you calculate the cost and time for the project.

We use the same formula of Initial level but take inputs on our experience bases. You execute the same process virtually in your mind.

1. Break down of task *(first in phases, then in modules)*
2. How many modules can run in parallel *(Dependency, resource availability)*

3. Then you have time *(time required).*
4. Then you draw your resources *(total resources required)*
5. Then you consider the cost of your resources.*(Resources * Time)*
6. Then you will have your hardware resource, software resource, time and cost. *(cost + hardware cost+ licensing)*

Interview-2

Look… the first variable I used in this calculation is Time. Because through my skills *(experience)* I am judging the months required to complete this project. On the base of this assumed value I performed the remaining calculations.

The way I calculated the effort is eight hours a day and 5 days a week. So it comes to 160 hours a month for a single person *(work hours for a month).* I say this project required six months. The developers will be involved from the start of the project *(Project phases + resource required for each phase).* So their effort will be 26000 hours *(no. of resources * work hours for a month *total months).*

Q: Did you assume the required 6 months for project solely on the basis of no. of people working on this project?

Yes *(confirmation of no. of resources)* and on the basis of my prior experience that if you have this no. of resources then how long this Type of Project *(Project type information used)* will take to complete. Off course you will know that it is must to have an idea about in what time this no. of people can do this task *(Productivity information used).* Using Project development experience this project can be developed in 6 months *(Prior project information- Analogy).*

Q: what information you used from your previous knowledge?

I considered the experience of developers, their productivity *(programming experience and Productivity information used)*...

Q: Can you please recall and explain the steps you take for estimating this task?

First I consider the type of project to be develop *(project type information)*

The no. of people working on it *(team size)*

Experience of people *(programming experience)*

Then assume the required months i.e. 6 *(analogy)*

Monthly effort and Involvement level of people to calculate effort *(effort = Sum of (monthly effort* duration of involvement in project for each resource))*

Base on required months and monthly expense the cost of the project *(required months* monthly expense).*

Interview-3

Q: Kindly explain me step by step how you have calculated these estimates?

First of all I would tell you that the tasks you give for estimation has very less information available for any point of view including estimation. You ask to analyze the cost, effort and time on the base of Expert Judgment. The similar projects form the base for Expert Judgment *(Analogy information used)*. The variations take place on the bases of requirements, scenarios, environment, target audience etc *(modules, Environment, Users information used)*. All these things matter. If you develop the same project for America or any European country then the things becomes different. And if you are developing for Asian country or Pakistan then the things will be different *(environment information used)*. So all these things vary and you have to keep all these things in mind for Expert Judgment. But based upon the requirements provided although very less and based on my experience and judgment I have given some rough cut estimates.

Next as I told you that I don't agree to team composition (*Team productivity information used)*. Because if you put so many resources (*required resources information used)* in situation that you don't properly know the nature of the project*(project type information used)*, don't properly know the requirements, risks are not identified *(application type, organization type, technology information used)* all these things. So based on those things its not good thing to put in so many resources. Your have to utilize resources efficiently. The efficient resource utilization is you consume the minimum required resources (*required resources information used)*. 33 is too high figure *(team size information used)*. Well it can be if you are going for a very high fie product or very high fie quality thing which have a very large scope *(Project Type information used)*. Here you have limited scope. Scope is not very large to estimate that this high no. of resources will be utilized.

Based on my estimation 17 resources will be required *(Team Size information used)*. The team composition you provide did not include analyst and documentation and technical writing which is an important factor that wasn't there *(Team Composition information used)*. So including these I mention you. So I have calculated those estimates that I give you in man months. And also provide you divisions that requirements, design, development and testing *(Phase wise division of project required resources)*. I provide you these 4 division and mention *(men months)* in it. The calculations are shown in front of you, how many calendar months, months and the cost.

I also told you that the cost varies based on the companies i.e. in which way the company do costing *(for its resources) (Standards and policies information used).* I took a rough figure of 270000 per man month *(monthly cost).* And calculated the cost based on the man months required *(cost= monthly cost * man months required).* Plus I added the contingency and inflation *(contingency added).* I also told you that contingency based upon the environment *(environment information used).* Such as if you are developing in Pakistan then the contingency will be different from if you are developing in America or in some other foreign country. Plus the resource cost, means if you are developing in china then you will find resource on lower cost than Pakistan (resource costing based on location). 3rd comes your overhead cost. Overheads cost includes feasibilities, electricity billing and other resources you consume *(Internal environment information used).* So I included 5% for each three of them. It can also vary let say if you are developing in Islamabad then it will be different and if you are developing in Pindi then it will be different. So your overhead cost will vary. These are the basic things on which I judge and provide you the estimates.

Q: first you divide it in the phases and told me the time required for each phase……

Exactly *(confirmation of phases division),* this is what you have to do WBS, till you breakdown your estimates can not be developed. I did a high level breakdown i.e. requirements, design, development and testing *(breakdown in phases, not in activities).* I took these 4 phases and after that test you are ready for beta take. If you go further from beta take then it will take more time *(extensive testing adjustments).* Actually when you launch a product and implement it in real scenario, it comes across a lot of things *(Support and training).* I give you example of a company. They purchase a product. It took them 6 to 8 months so the proper billing can be performed. Now you can see that a developed product, because it has to be changed, it has to be customized, took 6 to 8 months. I am talking about calendar month. I have no idea about the man months. These type of things happens in real scenario which I know and brief you.

Q: First you divide in phases and then you give required months for each phase. On what basis you judge that?

Analysis involves the historical information sort of things and some previous projects you have develop *(Project type- analogy information is used).*

Q: Can you explain what have you consider form that historical information?

Historical information like (1) Type of Project i.e. its billing software, (2) you want to launch as a product because for product you need to include extra information such as *scalability, adaptability, customizable.* You should not target one organization and when you go for other organization it gets fail. So you have to keep these sorts of things in mind.

And it is not a very large level thing. It is targeting only small to medium level *(Target Organization)* companies.

And the things given like customer support, these will be your small modules and you will be integrating these modules *(No. of modules, Integration of modules mostly concern with technology, complexity).*

Q: did you consider what modules or how many modules will be developed?

General will be your billing module *(Modules).* Then your interfaces come in billing that which level of information should be shown to user. Even in your company have different level for information. For example which information should be displayed to Customer support peoples, which information to billing people , which to the people actually receiving payments, which information should be shown to higher management*(complexity of modules)*. These are all sort of levels on which you have to define it. Let say I am your customer. There should be a customer module *(module to be developed and customer satisfaction)* which provide me the login and check his billing history, call log. It will be a module. Except that there will be inter supporting modules for billing *(No. of modules)*. These will be your small size modules based on the way you record call, frequency. There will be criteria defining for recording call of different frequencies. It is package based *(modules complexity/ algorithms)*. All these things will be defined.

Q: what have you consider for the given information?

Mainly I consider the technical requirements *(technology required information)*. I told you my objection about the team *(Team Productivity- team size and composition)*. The database like things does not matter whether you use one db or another. You will use the one which is compatible to your environment *(Environment- target platform, target organization, technology)*.

This information does not include the licensing *(Adjustment factor)*. So if you need licensing you will include it. Even if you are working in Pakistan with a company like Gold partners then you can not go without licensing. It is in company profile *(Standard and policies)*. For example this project uses mySQL. Means it is focusing small to medium organizations *(target organization, project type-scope)*. If you go for large organization you can not use mySQL because it provide limited storage and services *(Technology- constraints and* limitation). Such as .net is a very powerful platform *(Development Platform Maturity)* and if you use SQLServer with it, you can not find a better combination. The reason is that the .net on architectural level has many built in assemblies for its integration (Reusable Component). This reduces your development time, fasten your record fetching. It effect the performance of application and at the end effect the development cost. For example if you have to write a module in Java but .net provide a function to provide all that information then your development cost will be reduced. Its means that here you can do rapid application development but there (java) you can't *(technology limitation).*

The domain knowledge of team, communication, development type is product. Reporting information through organization type, target organization is obviously telecom. The main things are team size and team productivity.

Q: How did you calculate your man-month cost?

It is based mainly on company policy *(Standard and Policy information used)* for example the salary paid to different level of peoples etc. Different companies have different policies which I keep in mind and calculated the cost for a man-month. I have experience of working in different companies and I adjust my judgment according to current company policies. The maturity level of company *(Maturity level Information used)* also influence because you have to maintain proper documents for project and many other thing to meet maturity criteria.

<u>Interview–4</u>

Q: please elaborate how you build these estimates?

Let come from the third point Cost. Ok! Now the person months and the total time span I mentioned. Another thing that I mention is the salary ratio of your employee *(costing for resources)*. So there are different methods for costing. I mentioned the employee's salary according to my organization *(Environment Information used)*. It is not exact.

For this approximation I judge the required person months and the no. of months for this project. Then I multiplied the no. of human resources to resources used *(no of peoples * cost of resources)* and get the total expense of all the employees for one month *(monthly Cost)*. Means if I do this project, what will be my monthly expense.

Then I told you this project required 14 to 15 months. I multiplied the monthly expense to 14 months *(costing= months required * monthly cost)*. It gives the estimated cost of my project that estimated cost will be this much. There will be plus, minus *(contingency buffer)* but the estimated is this. This is the simples and easiest way of costing.

2nd thing is time required. What I did in time required is the modules involved *(no. of modules information used)* in the project. It is a large kind of project (Project type information is used), but it has 3 main modules i.e. moderator, rating and billing. These are the main things which derive my time estimates. And the 4th thing is integration *(modules complexity main driver of time)*. So you have 3 main modules and 4th one is small integration module (no. of modules involved). If you want to do integration then you should know the feasibility. Its means you have to check the feasibility of all the modules to see whether it is possible to develop or impossible *(technology complexity)*. For that I need to consume my senior resources to do feasibility study *(phase division and resource level information)*. According to that I consider the required time *(required time judgment)*.

The same resources will visit the stakeholders *(stakeholder participation)* and perform a complete analysis of the project to know what is required or what will we provide *(customer satisfaction)* and what is the high level features at the end of that project.

In this way I divide in different phases *(decomposition in phases)* and depending upon which resources should be used in which phases (phase wise resource required information used) such as I cannot use interns for feasibility. So I will use same level resource. Similarly I cannot use a DBA in development. for that the high level developers should be used *(experience of resources used)* and team work would be performed *(Team information, communication information used)*. This way I divide it in phases and judge that feasibility will required this and development , testing in alpha and beta release will required this much months *(required time for each phase)*. After doing that I come to an estimated time of 14 to 15 months. It is not accurate it is estimate only.

For effort you have 30 persons *(team size information is used)*. But all of them will not be utilized at the same time *(resource required)*. I have 30 persons but I will not utilize all. I will use only which are required such for QA person for QA with involvement of to 4 developers *(phase wise division of men to calculate effort)*. During development 10 to 12 development persons will be involved. Analyst will be use in analysis. Not all at a time.

I-f Textual Model of Estimators

First person way of judging estimates.

Cost Judgment

1. Resource costing process

1.1 The development level on which resource are used to make judgment about their cost (Salary)

2.2 Total resource on a level multiplied by judged cost

3.3 Sum of resources on each level resources multiplied by their respective cost

2. Monthly Resource Cost Estimates.

3. Cost Estimation Process

1.1 Total months required from *Time Judgment* process multiplied by Monthly resource cost estimates.

2.2 Contingency is added in cost.

3.3 Add your hardware and licensing cost.

4. Cost Estimate.

Time Judgment

1. Phase wise Breakdown of project in phases

1.1 Project Type – E.g. Scope , Application Type, Target Organization information

2.2 Resources required Information

2. Time required to develop these modules

Modules to be develop information

Complexity of modules information

i.Algorithmic Complexity information

ii.Technology- Implementation Difficulty information

Dependencies of modules

Resource Availability information

2.2 Team composition and work assignment information

Team Size information

Team Communication information

Technology- Technological experience information

Domain Knowledge information

3. Time Estimate

Effort Judgment

1. Man hours per month

   i.8 hrs per day * 5 days a week= 40

   ii.40 * 4 weeks a month= 160 hrs.

2. No. of modules information

   i.Complexity – algorithmic, technological information

3. Required Resource (Judge during time estimate)

4. No. of months required (Time Estimate)

5. Add contingency

6. Effort Estimate

2nd person way of judging estimates.

Cost Judgment

1. Required person months

2. Required months

3. Resource Costing

   a. Level based salary package of the organization

i. Standard and policies information

ii. Organization structure information

4. Calculate Monthly Cost
   a. Sum of No. of people on each level multiplied by their cost (Salary)
5. Cost judgment process
   a. Monthly cost is multiplied by the required months of *Time Judgment*
   b. Contingency is added
6. Cost Estimate

Time Judgment

1. Project Type information
2. Phases required for that project
3. Modules involved Information
   a. Complexity of modules Information
4. Technological Complexity Information
5. Level of resource required for each phase
6. Stakeholder involvement Information
7. Customer satisfaction Information
8. Phase-wise resource required
9. Team Experience of Resource used
10. Time Estimate
11. Add buffer
12. Time Estimate

Effort Judgment

1. Team Size
2. Resource required for each phase
3. 176 hrs per month for single resource (8 hrs * 22 days)
4. Phase-wise no. of resource required * 176
5. Sum of 4
6. Effort Estimate

3rd person way of judging estimates.

Effort Judgment

1. 160 hrs for a single person (8 hr * 5 days * 4 weeks a month)
2. No. of resources required information in each phase
3. Multiply 2 and 1 to have phase wise effort required
4. Sum up effort required for each phase
5. Effort Estimate

Time Judgment

1. Project Type Information
2. Resource Available Information
3. Analog Information
4. Phase wise resource division
5. Programming Experience Information
6. Team Productivity Information
7. Technology – Language experience Information
8. Time Estimate

Cost Judgment

1. Required months

2. Monthly Expense of resources based on salary

3. Required Months * Monthly Cost

4. Add overhead cost

5. Cost estimates

4th person way of judging estimates.

Time Judgment

1. Project Type- Product, Scope. Application type

2. Resource Available

3. Target Organization

4. Phase wise division

5. Analogy Information

6. Customer Statistician information

7. No. of modules information

8. Complexity of modules

9. Technology Information- Integration of modules mostly concern with technology

10. Team Size

11. Team Communication

12. Reusable Components information

13. Platform Maturity

14. Technology limitation and constraints

15. Target Platform information

16. Time required for each phase

17. Sum of time required for each phase

18. Testing Level required

19. Time Estimate

Effort Judgment

1. Phases wise division

2. Resource required in each phase

3. Time required for each phase

4. Multiple 2 and 3 to get effort required in a phase

5. Sum up effort of all phases

6. Effort Estimates

Cost Judgment

1. Resource costing

    a. Standard and policies

    b. Environment- where u develop

2. Monthly cost

3. Cost- monthly cost * man months required

4. Add contingency

    a. Environment – where you develop

5. Add inflation

    a. Environment

6. Add overhead expense

    a. Internal Environment

    b. Standards and policies

7. Add hardware Cost

8. Add licensing Cost

9. Cost Estimate