Chapter 1

Introduction

1.1 Motivation

“Estimating size and cost of a software system is one of the biggest challenges in software project management”.[1]

Software cost and effort estimation is the most challenging task in software engineering. Accuracy in estimation is still a development area for software developers. It is admitted by many academicians and researchers that size and cost estimation is the most difficult task in the software industry. The company’s goodwill, to a large extent, depends on how accurately it estimates the cost. For wise decision making, it is essential to minimize the uncertainty and maximize the accuracy related to effort, development time and the cost of software development.

According to Standish group chaos report [2], “The Standish Group research shows that 31.1% of total projects get canceled before they even get completed. 52.7% of projects exceed their cost by 189% of their original estimate”.

Standish Group report 2009[3]: "This year's results show a marked decrease in the project success rate. Only 32% of all successful projects were delivered on time, on budget, with required features and functions" says Jim Johnson, chairman of The Standish Group, "44% of the projects were considered challenged as they were are late, over budget, and/or with less than the required features and functions while 24% were considered failed as those were cancelled prior to completion or delivered and never used."

Therefore, as discussed by the Standish chaos group report, cost overruns and late delivery of the projects are the main reasons for software failure. This result in
customer’s dissatisfaction, reduces profits, destroys the company’s reputation and demotivates the employees.

In last 50 years, many software cost estimation techniques have evolved in the market to address the different challenges faced by the software industry in software cost estimation. The different estimation techniques/models are divided into two categories i.e. algorithmic and non-algorithmic models. Algorithmic models use mathematical formulae where parameters are based on industrial experience or on historical information that relates some software metric (i.e. usually size) to the project cost. Algorithmic models are also known as parametric model. They include COCOMO I, COCOMO II, Putnam resource allocation model etc.

In non-algorithmic models, no mathematical formulae are used for cost estimation. Expert estimation, estimation by analogy, top-down estimation, bottom up estimation, Parkinson’s Law and price to win are some examples of non-algorithmic models.

Both these models have their pros and cons. In case of non-algorithmic methods, it is necessary to have enough information about previous projects of similar type, because these methods heavily rely on historical data to perform estimation. Additionally, non-algorithmic methods are easy to learn as compared to the algorithmic ones. Algorithmic methods are based on mathematics and some experimental equations. They operate on extensive data related to the current state of the project. The results obtained using algorithmic methods, however, are more accurate and reliable.

From a research standpoint, it is of great interest to analyze which cost estimation method is suitable for the specified requirements, functionality or other project characteristics. To start with, it is important to identify potential factors and the impact of those factors on estimation of effort and cost. In my thesis, the potential factor would be software quality, as this factor plays a critical role in the development of any software.
There are many factors that influence the cost of software. Some of the factors are number of developers required, complexity of the project, functionality required and some other quality attributes. There are many quality models available in the market like Boehm software quality model, McCall quality model, ISO 9126 etc. The software engineering community standardized the ISO 9126 as a universal model for comparing one product with another. The quality attributes specified in this model are functionality, reliability, usability, efficiency, maintainability and portability. These quality attributes are further sub-divided into more quality factors.

1.2 Objective

The objective of this research work is to analyze different estimation techniques conforming to quality requirements.

The following research questions were defined in the beginning of this research:

Q1. Which cost estimation technique is more relevant - parametric or non-parametric?

Q2. What are the advantages and disadvantages of different cost estimation techniques?

Q3. What all techniques are prevailing in the market?

Q4. Is there any gap between the literature and practicality?

Q5. What are the challenges and barriers organizations face during the estimation process?

Q6. What impact do specific quality factors have on software estimation methods?

The key objectives of this thesis are:

- **To provide new software cost estimation technique**: There are many software cost estimation techniques available in the market. Software engineers, however, continue to struggle in estimating costs that are near accurate. This new technique will help them to achieve results that are very close to actual cost.
- **To establish a new framework for software quality assessment**: Software quality is an integral part of the software development lifecycle. The software industry, however, lacks in quantifying and estimating software quality requirements. This new framework will help them to measure the quality requirements. Through this new technique they would be able to combine various quality attributes and evaluate a single value for the whole process instead of finding separate values for each. This will not only reduce the effort but also give crisp value for multiple attributes. The best part of this tool is that it can be used with other estimation approaches as well.

- **To combine the quality framework with software cost estimation in order to predict the software cost conforming to quality requirements**: The proposed software quality framework can easily be used with any software cost estimation in order to predict the cost of development of the software.

1.3 **Scope of This Thesis**:

This thesis supports the development of a new software cost estimation model considering the advantages and disadvantages of the existing software cost estimation models. The proposed method also incorporates expert suggestions and quality attributes.

This thesis also provides a new technique for quality assessment wherein it can be used with cost estimation in order to estimate the software cost conforming to quality requirements.

1.4 **Structure of the Thesis**

This section is an overview of all the chapters of this thesis:

*Chapter 2* covers different research concepts and methodologies
Chapter 3 will throw light on the background studies required to fully understand the methods and concepts like: different estimation techniques, group consensus methods, and discussion on quality in general and software quality in particular, the concept of cost of quality

Chapter 4 will showcase literature review on the selection and performance of cost estimation techniques with regard to software quality. Here I will discuss the performance and applicability in practice (based on the results taken from the academic literature)

Chapter 5 contains information derived from the survey and interviews conducted with the experts for this research. It will reflect details of experts interviewed, the questions, the topics of discussion, important findings, best practices and comments of the experts.

Chapter 6 will analyze some important software cost estimation factors and their impact on final estimation like training, nature of project, review, risk buffer and so on.

Chapter 7 will discuss a proposed software cost estimation technique as a result of data that I gathered from survey and interview process.

Chapter 8 will throw some light on software quality attributes associated with software estimation process and its impact.

Chapter 9 mentions the conclusion of this work and provides an outlook for possible future work.
Figure 1.1: Structure of the Thesis

Chap-3: Software estimation Techniques

Chap-4: Literature Review

Chap-5: Interview data

Chap-5: Survey

Chap-6: Analyze the data

Chap-7: New Software Estimation Techniques

Chap-8: Evaluate quality parameters with new approach

Chap-9: Conclusion

DATA

Chap-3: Software Quality Techniques

Chap-3: Literature Review

Chap-3: Survey