CHAPTER 3

PROBLEM DEFINITION

This chapter describes about the various challenges and issues related to software effort estimation research studies, and it also derives the problem statement with respect to the study carried out.

3.1 INTRODUCTION

A feast of investigations has been made to configure a relationship between the size and cost of software. There is a vast potential for cost evaluation in view of the uncertainty with reference to the input data. These may relate to the various expenses and in addition to the overheads of personnel like the accountants, administrators, system managers, technicians cleaners coupled with outlays for the networking and communication.

Thus, the software cost evaluation emerges as a complicated science. Therefore, a multitude of functional sizing methods, sizing metrics, cost and effort models has been launched in bits and pieces, only the regular methods employed in the Software Cost Estimation (SCE) in common are discussed, thus throwing light on the most significant developments in this domain trends and focusing mainly on the vital subject that is to be investigated, and the main issues in SCE process are added to it.
3.2 CHALLENGES OF SOFTWARE COST ESTIMATION

The software cost estimation task encounters several challenges for attaining good precision for number of reasons. As software is intangible and the evaluation task in respect of an intangible product is cumbersome, the software estimation is entirely different to any evaluation function in a different domain. In this regard, the following are the biggest challenges at the time of software estimation,

- It is not possible to result any evaluation of the unidentified subject
- The evaluations are generally positive
- In most of the cases, the evaluated period is optimally utilized.
- The organization depends excessively on estimations.
- Additional duration of time is generally consumed on things which are not evaluated
- The estimates are not deemed as estimates
- The estimates fail to account for the productivity changes between programmers
- Any changes in requirements are not accounted in the estimations
- The actual state of affairs is revealed after the damage is done.
- The estimates are not revisited when it is impossible to achieve them.
- The programmers are blamed for the failure of estimates.
- The estimates are carried out by inefficient and improper personnel.

One of the vital challenges in the software cost estimation task is the lack of data which is essential to authenticate the suitability of any recommended models, metrics and functional sizing methods. In fact, a large chunk of the models and sizing methods launched are based on insufficient quantity of data. Further, certain models evaluating the size, estimates the
cost based on 30 UML files and the resultant suffers from lack of reliability there by making it unfit for generalization and application in a growth environment. Moreover, the data also tends to be exceedingly susceptible as one sizing technique may generate an excellent estimate in a particular software development company in a specific country. But, may find it incapable of achieving the identical precision rate in a different software development company in another country. Another difficulty emerges in the form of lack of sufficient data prior to the software process, though this can be overwhelmed by gathering data from a collection of projects and generalizing one specific rule for the purpose of estimation. Incidentally, the cost estimation is habitually labeled as the effort estimation, though they are not exactly synonymous because in the case of effort estimation, there are ancillary project costs such as the licensing, travel, hosting, training of personnel etc. Further, the staff expenses relating to a project are certainly the leading segment of the overall expenditure. Therefore the cost estimate bears different nomenclatures. The effort estimation is achieved by means of parametric equations, prior experience or expert judgment. As a rule, they are categorized in to three different methods such as the algorithmic models, expert judgment and machine learning.

3.3 CHALLENGES IN USE OF TARGET ESTIMATES

A decision document generally contains only the anticipated expenses. However, with the intention of attaining cost aggressiveness a supplementary cost estimate is a must, which takes the name “target estimate”. A Target estimate, in turn, is nothing but an estimate with an extended target, which is viable based on a list of opportunities:
• Optimization of the implementation plan.
• Risk based staffing.
• Enhanced team configuration and assimilation.
• Estimation of fundamental cost propellers.
• Deployment of standard curves to assist the debate on targets
• Risk cut-down steps and
• Opportunity register – price

It is unrealistic to anticipate precise cost estimate of any software because of the distinct ambiguity in software growth. The challenging and robust interaction of factors has a vital effect on the software development cost. However, it is possible that the estimates can be fine-tuned as the software development cost estimates are systematic and very incoherent. A vital goal of the software engineer is to design functional models which are competent to precisely estimate the software cost.

3.4 CHALLENGES FOR SOFTWARE DEVELOPER

The software development business has bumped into preparing gigantic cost and time estimates. The programming estimation is based on the review of programming qualities which are commonly identified with the item, the process and the assets of programming development.

• To predict the enhancement effort for a product structure centered around a certain dimension for the last 10 years.
• Travel and training expenses.
• The salary of software engineers.
• Maintenance and repairs of hardware & software.
The software development effort estimation represents the function of forecasting the most reasonable employment of effort needed for building up the software on the basis of certain constraints. It has emerged as one of the grave challenges in Computer Science field for the last several years. This is because it is difficult to get the time and cost estimate at the initial phases of the software development and they also are defective because of lack of accuracy.

A specific model or any general way to estimate the software costs estimation of object-oriented software is by means of component based system engineering. Thus, it is possible to locate the important parameter for component based cost evaluation and also the level of their relevance for the relative cost estimation technique.

Conventional algorithmic methods like the regression models, Software Life Cycle Management (SLIM), COCOMO II model and function points invariably need an evaluation procedure on a permanent basis. However, at present, it is not suitable for software developers and companies.

Innovative soft computing methods for the purpose of effort estimation are based on non-algorithmic approaches like the Fuzzy Logic (FL) provides an appealing option for tackling the challenge. This investigation is targeted at designing a novel fuzzy logic realistic technique to attain further precision in software effort estimation. The important motive of this investigation is to explore the function of fuzzy logic method in fine-tuning the effort estimation precision by characterizing the input constraints, by employing the twin-fold Gaussian function which furnishes better transition from one interval to another.
Software development effort estimates are likely to be highly inaccurate and systematically over optimistic due to the various effect of prediction, anchoring, planning misconception and cognitive effects. Empirical evidence suggests that the causes of the problem, to some extent, were due to the influence of irrelevant and misleading information. The input data are further compounded and it is difficult to obtain them, especially in early stages in a program development. The size must be estimated early in a project using one or more sizing models. Some sensitive inputs, such as analyst and programmer capability in cost drivers, are based on individual and are often difficult to determine.

3.5 PROBLEM OF SOFTWARE COST ESTIMATION USING COCOMO

It has to be overemphasized that ambiguity at the input level of the COCOMO model leads to ambiguity at the output level. This is easy to observe, and it significantly holds a realistic effort. The Fuzzy logic-based cost estimation models are very suitable in cases where indistinct and inaccurate data are to be accounted. The cost drivers are generally expressed by means of uncertainty, which necessitates subjective evaluation. The effort multipliers and scale factors of the COCOMO were mentioned in natural language as very low, low, nominal, high, very high and extra high and these were characterized by permanent numerical values. More typically, the challenge of software cost estimation employing COCOMO invariably relies on a single (numeric) value of cost driver of a specified software project to forecast the effort. However, this is not an appropriate way to fix numerical number to each of these scales. But, still there is certain linearity when these functions are employed. The overlapped symmetrical triangles or trapezoids decrease the fuzzy systems to precise linear systems. Moreover, it is likely
that while employing a trapezoidal function certain attributes are allocated to the maximum degree of compatibility when they have to be assigned lower degrees. With an eye on getting rid of this linearity it is proposed to employ more permanent Gaussian function to characterize the cost drivers.

A host of debates appear in the literature illustrating the efficiency of the COCOMO techniques in ascertaining the precision of cost estimation of both models when applied on a particular project. Moreover, it is also effective to determine the impact of cost drivers in the course of the system development life cycle phases. Further, it is capable of authenticating the precision of the cost estimations of COCOMO models for projects which are created with the help of the Hyper Text Markup Language (HTML) and JavaScript. Thus, it is competent to ascertain the level of precision and reliability they have, and also their suitability for evaluating the HTML and JavaScript Code.

3.6 CHALLENGES IN WEB COST ESTIMATION

The appropriate evaluation of size and duration is the vital concern in this regard. A Novel modified size metrics are preferred to tackle the Web objects such as the shopping carts, Java scripts, and building blocks like the Cookies, ActiveX controls, and Object Model components. Further, innovative modified duration-estimating equations are needed to tackle the issue that the cube root laws employed by several evaluation techniques do not appear to work for the Web.

- Everyone is eager to ascertain the probable cost.
- The requirements are not distinctively listed out.
- The requirements have to be altered for business purposes.
• The requirements need appropriate change further solution.
• It involves a lot of time to start the different site.
• A site is not actually finished when it is started.
• Design and financial plan for upgrades.
• The up-front costs are very high.

It is very difficult for new companies to conduct evaluations for a multitude of reasons. Usually, some of them may be start-up and idea businesses, which have negligible revenues and perhaps suffer from operating losses. Even in the case of the new profitable companies they have insufficient experience. Several standard methods intended for estimating the cash flows, growth rates and discount rates usually fail to perform efficiently. The integration of data on several grown-up companies in the business and the company’s unique traits are essential to predict the revenues, earnings and cash flows. Suitable processes have to be setup for determining discount rates for private capital and for adapting the present value considering probability of failure. In the process, the venture capital method of valuation which is at present extensively employed is defective and hence has to be substituted.

3.7 ISSUES IN THE SOFTWARE COST ESTIMATION

The software cost estimation has emerged as a complicate science and hence there is a host of functional sizing methods, sizing metrics, cost and effort models that have been launched. Some of the issues in cost estimation process are described below,
o **Metrics conversion**: The behaviour of estimation process in the software industry is quite different to those in the rest of other fields, it gives rise to a multitude of sizing methods, LOC, function points, object points, application points, use case points, though there exists no clear-cut rules to toggle between these metrics.

o **Software sizing from requirement and design**: Software estimation is performed from the infant stages of software development, and hence the capture of the estimation from the identified traits of software represents a functional method for estimation. In this regard, several gadgets are employed to characterize the prerequisites and the blueprint of any software like the UML diagrams, Z specification language, state transition diagrams, and the data flow diagrams.

o **Complexity analysis**: The complexity represents a variable and subjective expression which can be observed in several ways. In a nutshell, if the software is more complex, it turns out added deficiencies, and is very difficult to sustain and modernize. With reference to the software industry the extensively employed formula used to evaluate the complexity of software code is the unique Cyclomatic and Halasted formula.

o **Risk and ROI analysis**: In the software development process, it is observed that several projects have failed to deliver or exceeded its budget and resource allocated for identical reasons in almost all the development companies. Hence, it is essential that further investigation on the subject has to be taken into account for these common factors, so as to incorporate them in the estimation process
with a margin of occurrences probability to keep away the overrun in the estimation.

- **Investigating OSS**: As the wheel of Open Source Software (OSS) technology run, reveals that, there is an urgent requirement to design a novel model which is competent to estimate the effort expenses for the OSS. Further, as the normal cost-effort models are not suitable for the purpose, these models have to take extreme care for the nature of OSS setting in order that the advancement of OSS can be realized at any time between the volunteers around the world, in addition to finding the way to evaluate the effect of the community adjacent to the core development team, without caring for the normal effects of software required to be estimated.

- **Revaluation of function point**: The function points represents one of the well-acknowledged and dynamic sizing methods employed in the software cost estimation process. The function points over a period of time do not seem to have been renovated so as to reflect the modern status of technology. However, several investigations have been conducted to standardize the weights of function points, though such calibrations have failed to fine-tune the precision of function points, thereby necessitating an innovative valuation model for the function points.

### 3.8 EXPERIMENTAL METHODOLOGY

Cost estimation process involves a number of organized steps that provide valuation with reduced error. However, the procedure used by
the existing estimation analogy techniques is not yet able to correctly handle the categorical data.

Fuzzy logic is a methodology to solve problems which are too complex to be understood quantitatively, based on fuzzy set theory. In this research, a novel approach has been proposed which is based on reasoning by analogy, fuzzy logic and linguistic quantifiers to estimate the effort when the software project is described either by categorical or numerical data. In order to overcome the disadvantages of using Fuzzy and Analogy individually, a new method - Fuzzy Analogy, has been proposed to estimate the effort. A set of candidate measures for software projects similarity has been developed and validated. These measures are based on fuzzy sets, fuzzy reasoning and linguistic quantifiers. Use of fuzzy logic-based cost estimation models is more suitable if unclear or inaccurate information are considered. Fuzzy systems attempt to imitate the processes of the brain through a rule base.

This research introduced an optimized fuzzy logic based framework to handle the imprecision and uncertainty in the data at early stages of the project, to foresee the effort more exactly. The said framework is built upon an existing cost estimation model - Cost Constructive Model (COCOMO). The COCOMO model is an empirical model that was derived by collecting data from a large number of software projects. These data were analyzed to discover formula that was the best fit to the observations. These formulas link the size of the system and product, project and team factors to the effort to develop the system.

Fuzzy logic-based cost estimation models have proved their mettle as the most efficient model for the cost estimation in software
engineering. It is also resorted to the inclusion of the optimization algorithm in fuzzy which goes a long way in further boosting the estimation process. The optimization algorithm deployed in the novel technique is the mighty Particle Swarm Optimization (PSO) algorithm. The effort of the software is deemed as the fitness for the optimization function. In accordance with these fitness values, the optimization of the fuzzy rule is effectively carried in this study.

The major problem that exists in the software cost estimation is the error rate that is estimated finally. The increase in error value leads to increased amount of cost in terms of software production which will remain as the major drawback in software cost estimation process. This has motivated the research to develop an alternate technique which can improve the cost estimation by reducing the error value.