GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

SOFTWARE COST ESTIMATION USING MODEL BASED TECHNIQUES: AN OVERVIEW

Javaid Iqbal Bhat*[1] & Zahid Hussain Wani[2]

*[1]Assistant Professor, Department of Computer Science, Islamic University of Science & Technology, Kashmir, INDIA

[2]Assistant Professor, Department of Computer Science, Islamic University of Science & Technology, Kashmir, INDIA

ABSTRACT

Software cost estimation is the forecasting of development effort and development time needed to develop a software project. It is considered to be the primary step of software development process and at the same time considered to be the key task as accurate assessments of growth of the current project, its delivery exactness and its cost control can only be achieved once calculated estimation is accurate. And at broader perspective an accurate estimation of a currently developing software project will result in landing the organization in a better schedule of its futuristic software projects too. With due above reason, software effort estimation has received a considerable amount of attention of many researchers for past so many decades. And accordingly, a good number of software cost estimation techniques from last so many decades have been proposed which differ from model based techniques, also called as algorithmic techniques to Non algorithmic to Data mining and to metaheuristics based. Besides all these, a varying range of artificial neural network based models along with their hybrids have also been developed. In this research study, a thorough review of model based techniques has been carried out for the purpose of getting details vis-à-vis strengths and weaknesses in these model based software cost estimation techniques.

***Keywords**: Algorithmic Techniques, Model Based Techniques, Software Cost Estimation.*

I. INTRODUCTION

Software cost estimation is the summation of predictions of both building effort and calendar time used to develop a software project. The building effort includes the summation of working hours and the total of workers included in the process of soft project development. Just from the inception of software project development, organizations of this nature came across to the problem of poor estimations of development effort and development time of software projects. A good reason for this was and which is persistent even this time is the availability of vague information about the software project to be developed at the time of its estimation process. A better estimate of software product is the only thing that can let any software development project manager to evaluate the project progress, gives him / her good track of potential cost control and accuracy in delivery time. This in widespread, however, gives the organization a better insight of resource utilization and consequently will land the organization in a better schedule of its futuristic projects. For this purpose, a good number of software cost estimation techniques from last so many decades have been proposed which differ from algorithmic [1] [2] to Non algorithmic to Data mining and to metaheuristics algorithmic based, but unfortunately non among these satisfy the acceptance standard of accuracy in estimation of software development projects. However, a varying range of artificial neural network based models along with their hybrids have also been developed and have shown improvements in the said. The most commonly used method for predicting software development effort as the case with COCOMO [1] [3] [4] are based on linear-least-squares regression. Being extreme susceptible to local variations in data points [5], the model have failed in dealing with implied Non-linearities and interactions between the characteristics of the project and effort [6]. Moreover, software cost estimation models if yield estimates with 25% of mean relative error to the actual and if follows for at least 75% of the time, are believed to be in acceptable accuracy. However, there remains an open space for developing effort estimation models with better predictive accuracy [7] too.