International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 21 Issue 2 – APRIL 2016.

MULTITIER DIVERSIFICATION IN SOFTWARECOST ESTIMATION

T.Sarada Kiranmayee[#1], Srijith S, Kaushik P, Rahul Kumar P, Madhanagopal D

[#1]*M.TECH, Asst Professor, Department of Computer Science and Engineering, SRM University, Ramapuram, Chennai, Tamil Nadu*
Student, Department of Computer Science and Engineering, SRM University, Ramapuram, Chennai, Tamil Nadu

tskiranmayee@gmail.com
shrijithsridhar@gmail.com
Kaushikchn185@gmail.com
Rahulmarlecha6@gmail.com
Gdmathan13@gmail.com

***Abstract*- Software cost estimation, is one of the important factors which affect the success of a project development. There areseveral area of the software engineering in which we can use the function point analysis like project planning, projectconstruction, software implementation etc. In softwaredevelopment, accuracy and efficiency of cost estimation methodology for a web based application is very important. Accurate project estimation such as cost estimation, qualityestimation and risk estimation is a major role in softwareproject management. In this paper, we present a softcomputing framework to tackle this challenging problem. Wefirst neuro-fuzzy inference system tohandle the dependencies among contributing factors anddecouple the effects of the contributing factors into individuals. Then we use software cost estimator to calibrate the parameters of contributing factors. In order to extend our frame- work into fields that lack of an appropriate algorithmic model of their own, we propose a default algorithmic model that can be replaced when a better model is available. Validation using industry project data shows that the framework produces good results when used to predict software cost.**

I. INTRODUCTION

As software development has become an essential investment
for many organizations, software cost estimation isgaining an ever-increasing importance in effectivesoftware project management. In practice, softwareestimation includes cost estimation, quality estimation,
risk analysis, etc. Accurate software estimation canprovide powerful assistance for software management
The concept of software cost estimation has been growing rapidly due to practically and demand for it. Today people are expecting high quality of software with a low cost which is goal of software engineering. So many popular cost estimation models like COCOMO81, COCOMOII, SLIM, FP and Delphi. Today most of the software companies follow COCOMOII for estimating the cost of products, this model is also not giving an accurate results.

PROPOSED SYSTEM

A NOVEL NEURO-FUZZY MODEL

The input for this model is the software size and ratings of 22 cost drivers including 5 scale factors (SFR_i) and 17 effort multipliers (EMR_i). The output is the software development effort estimation. Ratings of cost drivers can be continuous numerical values or linguistic terms such as ‘‘low’’, ‘‘nominal’’ and ‘‘high’’. The parameters in this model are calibrated by learning from industry project data.
There are two major components in our neuro fuzzy model:

- Twenty-two sub-models NF_i: for each sub-model, the input is the rating value of a cost driver, and the output is the corresponding multiplier value, which is used as the input of the COCOMO model.
- COCOMO model: the input is the size of software and the output of NF_i. The output is software effort estimation.

Sub-model NF_i:

There are 22 cost drivers in our neuro-fuzzy model. Each cost driver represents one factor that contributes to the development effort, such as application domain experience and product complexity. We use six qualitative rating levels to evaluate the contribution. When expressed in linguistic terms, these six rating levels are verylow (VL), low (L), nominal (N), high (H), very high (VH) and extra high (XH). Each rating level of every cost driver relates to a value called a multiplier value, which is a quantitative value used in the COCOMO model. Sub-model NF_i is used to translate the qualitative rating of a cost driver into a quantitative multiplier value and to calibrate these relations using industry project data. It should be noted that not all six rating levels are valid for all cost drivers. A natural way to represent linguistic terms is to use fuzzy sets. We define a fuzzy set for each linguistic term of every cost driver, i.e. ‘‘very low’’, ‘‘low’’, ‘‘nominal’’, ‘‘high’’, ‘‘very high’’, ‘‘extra high’’. The membership functions are either triangular functions or other functions, and the universe of discourse is the interval. We use fuzzy numbers ‘‘about 1’’, ‘‘about 2’’... ‘‘About 6’’ to represent linguistic terms ‘‘very low’’, ‘‘low’’, ‘‘nominal’’, ‘‘high’’, ‘‘very high’’, ‘‘extra high’’, respectively.

II. ARCHITECTURE

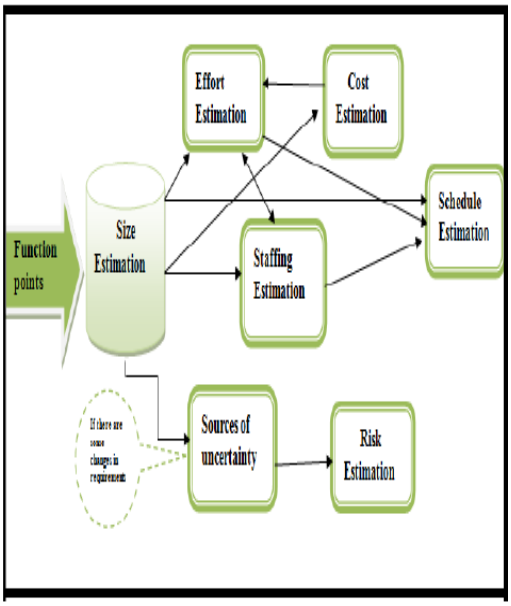

Figure 1: Overall Architecture

Function point is a measure of software size that useslogical functional terms inputs and outputs. A risk is inexposure to loss or injury or a factor, thing, element orcourse that involves uncertain danger. Risk assessmentinvolves risk identification, risk analysis, risk planning &risk controlling. Software risk can be internal or external; the internal risks come from risk factor within theorganization. The external risks come from out of theorganization & are difficult to control. Software risks canbe grouped into project risks, process risks & productrisks

III. CONCLUSION

. This paper has presented a general framework for software estimation. The framework concentrates on thepre-processing neuro- fuzzy inference system, the neurofuzzybank and the algorithmic model. We consider as the rating value of contributing factor as input and produce software metric as output. This framework has been validated with project data from the industry. The main benefit of this approach is its good interpretability, that is, by using the fuzzy rules, the approach tries to simulate the software engineers' line of thought when they are doing software estimation process of software engineers software estimation. An- other great advantage of this research is that we could put together expert knowledge(fuzzy rules), project data and the traditional algorithmic model into one general framework that can have a widerange of applicability in software cost estimation, quality estimation and risk analysis. In this proposed Expert estimator focuses on developing estimation tool for web based application. This tool namely (i.e. Expert estimator) is developing by using java as development language & java eclipse as the development tool. Proposed tool easily estimate the risk in software & also to estimate cost of the software. The cost estimation depends on the calculation of function points, cost adjustment factors & reuse. Function point approach as an input parameter into the "Expert estimator". This information is needed in the calculation of effort, schedule & total cost for the project. The risk estimation based on the risk assessment of software projects. Risk identification, risk analysis & risk prioritization are the main subparts of risk assessment. From proposed model it is easy to calculate the risk at different phase as the software projects progresses from phase to phase. From further research, it is highly recommended that other cost estimating method is considered such as Price-to-win as an added method to cost estimation for web based application & software requirements after adding the thread in to it & then we

will prioritize it using analytic hierarchy process & quality function deployment, & after this we will generate the results of that software using the proposed Expert estimator tool.

REFERENCES

[1] Khalid nadan, UAE University" Practical software project total cost estimation method, MCIT 2010 IEEE, page 7-10.

[2] Daya Gupta, MohdSadiq "Software Risk Assessment and Estimation Model", International Conference on Computer Science and Information Technology, IEEE Computer Society, Singapore, 2008. pp 963-967

[3] 3) Low G.C. and Jeffery D.R 1990, Function Point in the Estimation and Evaluation of the Softwa11re Process, IEEE Trans Software Engineering, Vol.16, no.1.

[4] 4) http://www.aw.com/cseng

[5] 5) Improving Estimation Practices by Applying Use Case Models, BenteAnda ,EndreAngelvik , and Kirsten RibuSimula Research Laboratory, Norway M. Oivo and S. Komi-Sirviö (Eds.): PROFES 2002, LNCS 2559, pp. 383-397, 2002. Springer-Verlag Berlin Heidelberg 2002

[6] 6) Improving Cost Estimation with Quantitative Risk Analysis Be More Precise by Employing Uncertainty by Gregory Nolder, Vose Consulting, www.voseconsulting.com.

[7] 7) "Classification and Analysis of Risks in Software Engineering" HoomanHoodat, and Hassan Rashidi, World Academy of Science, Engineering and Technology 56 2009 .

[8] 8) "Function point estimation methods:A comparative overview", Roberto Meli, Luca Santillo Data Processing Organization.