

Modeling Software Project Defects With Fuzzy Logic Maps

Dr. Chandrakanth.G. Pujari
Associate Professor,
MCA Dept. Dr.AIT, Bangalore
varuncp@rediffmail.com

Abstract: I propose a defect determining the quality model for software projects. The proposed model is based on fuzzy logic concept. The knowledge behind fuzzy logic process is built with a cognitive map of software defects. This map is developed for software projects, taken into account their characteristic lifecycle. The model was used to test the software project; the probability of not obtaining quality outputs was calculated considering the quality level idea and the over-budget sum. The calculated defect is situated on the maximum level in the defects map. A software system for determining defects in software testing project was also developed.

Keywords: software defects management; software projects; Quality model; Fuzzy logic; Defects map

I. Introduction

Recognizing, ranking and treating defects represent management activities. For a long period, danger defects, and financial problems have been managed [1][2]. But, the different, number and interactions between defects are spontaneously increasing. The defects have been increased due to mechanisms failure. According to these reasons, the industries admit the importance of managing all types of defects [5][6].

Various industries induced an environmental pressure on the organizational management for maintaining defects. One idea is to adopt the portfolio approach, the organization management considering the portfolio defect as the defect to the whole industry. The defects are managed in a holistic approach [11][12].

There is a upcoming inclination to quantify defects. The defect quantification allows managers to develop “what-if” pointed views and make decisions. Modern technologies have made the quantification possible. There is some effort to quantify the portfolio defects, based on the individual defects and the interactions quantification. But, this is not always possible [3][4].

Hard work and experience, industries become more capable of managing defects, and even searching out opportunities to assume defects. Industries understood that

informed defect-taking is a means to competitive advantage [7][8][9].

II. Software project management

The software project management is having maximum uncertainty. This study has component parts of innovation and correct prediction of the study result is therefore very complicated. It is the software project manager work to manage both the studies and the uncertainties with study outcome [13].

The software project managers should make quotation to software project team members: “In software development if you do not have maximum defects, you are not doing good work”. Innovative workers doing safe are most likely to deliver expected outputs. In order to obtain innovative outputs, innovative people should have a loss taking in nature, maximizing the probability of failure. Innovative people will avoid failure [14].

III. Defects model techniques

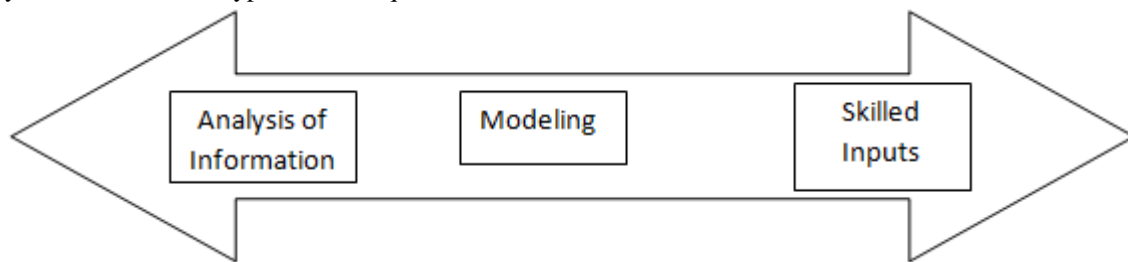
According a person’s behaviour towards in which the computations are carried out, analytical and simulated techniques for defect modelling. The analytical techniques need a group of assumptions, particularly connected to the probability distributions. Simulated techniques need a maximum number of trials to give nearest answer. The following diagram shows the defect model classes with their advantages.

Computation method Model of the relationships	Analysis (formula and its solutions)	Imitation (important solution for distribution)
Statistical Techniques if required	Analytical and statistical methods	
Designed qualities for Outputs not for Inputs		Designed Simulated Models

Figure 1: Classes of Defect model

The techniques used to model the defects are modified according to the clear defects which happened in the software Industry. There are various types of techniques that

can be applied to model defects. The following diagram lies in a continuous sequence of the data sources.



Practical evaluate From historical Information	Diff. Equ. Stochastic nature	Dynamics Simulation System	Diagram of Influence	Direct evaluation
Fit variables From distribution	Networks of Neural		Networks of Belief Bayesian	Preference of of quiz, lotteries
Maximum theory	Parameters that Affect defect		fuzzy logic	Technique of Delphi

Figure 2: The Continuous techniques for software defects modelling

Various methods for defect modelling, maximum experts are frightened to use them for making precise defect calculations. Fuzzy logic helps to obtain best solution and assessments in software projects management, within specific limits.

IV. Defects computation in software development using fuzzy model

Fuzzy logic contributed to defects management by using some methods in defects computation to software developments. This has two important stages: defect invention from experienced information, using defect maps. This allows defect quantification by classical set theory values of defect sources. The output has a maximum computation values.

V. Software Defects Identification Using Effect of Cognition Map

The logic behind software defects is not a correlated with its sources, but with software research programs, “eliminate

software defects, and also try to reduce it, it must required software defects is considered to be correct software defects”. Managing software defects is helpful to develop an effective and cognition fuzzy logic map of software defects. It is depends on software champion’s knowledge with his experience. The cognitive fuzzy logic map gives details of software defects in software research projects. Software defects occurred in software research projects as shown in following figure3.

Software research program introduce new activities, software research project implementation and closing down project. Figure3 will create different software defects in the following situations. Software defects are happened in all steps of a software research projects. Such as software research project opportunity, project proposals, contract of research projects, management plan, status of software projects, software project execution.

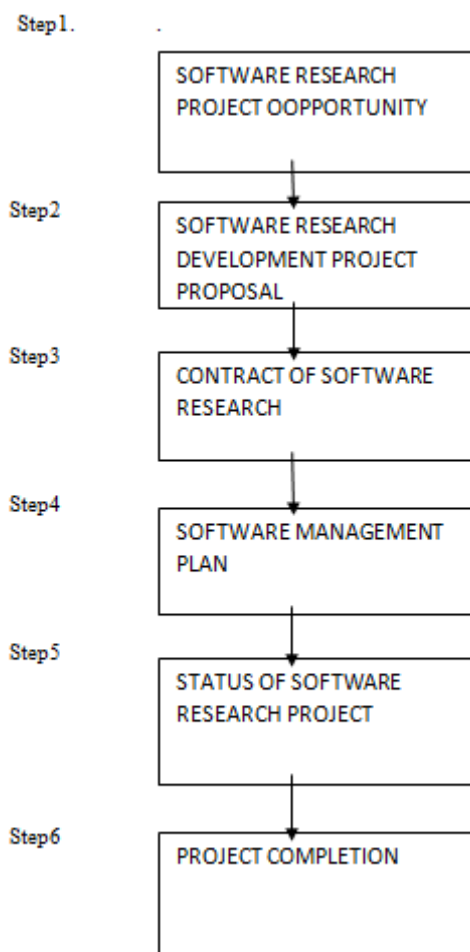


Figure 3. Software development lifecycle

Starting from the software development lifecycle diagram following defects are identified:

1. Defects in environment
2. Defects in management
3. Defects in finance
4. Technology and production

After creating the software defects fuzzy map, in the map every step identified defects, these entire defects name, description, and types are shown in table1. Identified software defects are fuzzy logic model law.

Defect Code	Defect Name	Defect Description	Defect Type
D1	Low quality idea	Accepted idea is not developed properly	Technology and Production
D2	Budget defect	The software projects schedule and cost is for completing the research projects	Finance

Figure 4. A small part of Software defects register in Research Programs

Software defects fuzzy Map in Software development

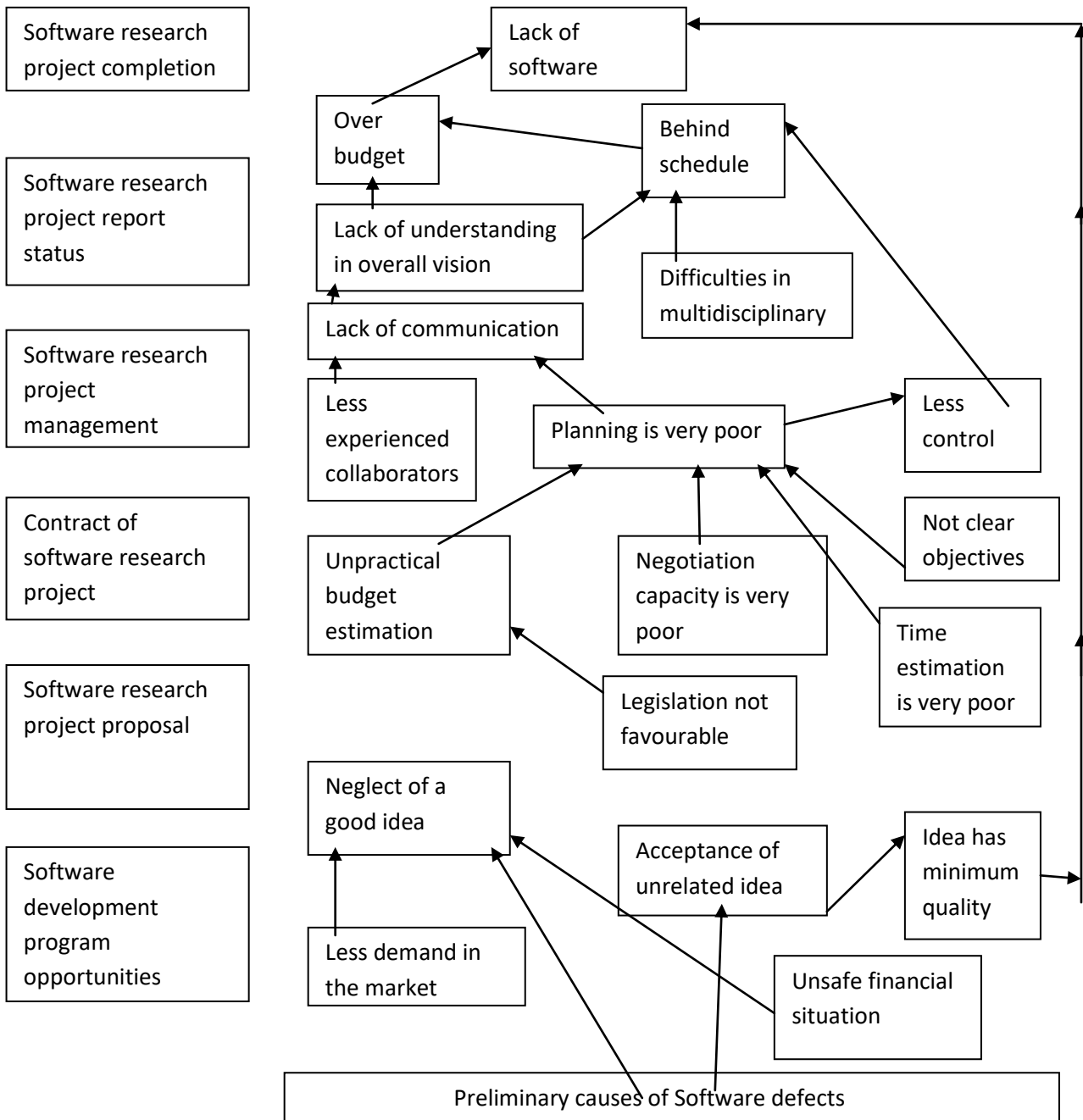


Figure 5. Software defects fuzzy MAP in software development

In software development research projects have typical software components of fuzzy logic model such as input / output parameters, and fuzzy logic rules. These rules are used in software defects modelling, and these are built on the familiar concepts from software defects. All these are recognised in software defects as shown in fig 4 of fuzzy logic map. In the software research development project “the maximum budget and maximum embedded software and research idea [15].

VI. Input/output Parameters

The fuzzy logic model has two types such as Constants and functions

Probe (Def) = Probability of software defects occurrence

Imp (Def) = impact of software defect on software research program

Here Def is represented as defect code they are shown in following table

Fuzzy parameter name	Scale in %	Performance Grade
Probe (Def)	[0,100]	VP (very poor), P(poor), M(medium), H(high), VH(very high)
Imp (Def)	[0,50]	VP (very poor), P(poor), M(medium), H(high), VH(very high)

Figure 6. Input parameters description in software defect analysis model

The output parameter is identified software defect and is represented by

Val (Def) = Quantitative value for software defects, Def = software defect code as shown in table3.

Fuzzy parameter name	Scales	Value Grades
Val (Def)	[0,10]	VP (very poor), P(poor), M(medium), H(high), VH(very high)

Figure 7. Output parameters Description of software defect analysis model

VII. Conclusions

This study offers on friendly usable tools for software defects computation in software research programs. This study is based on fuzzy inference theory and its rules, these rules are built on fuzzy maps of software defects. And also software research programs are well known high level software defects. Very few dedicated software defect systems are developed. Therefore fuzzy model is for software defect computation in software research programs is in innovative tools. This can be further developed to calculate all types' software defects from fuzzy map.

VIII. References

- [1]. Sommerville; Software Engineering, Pearson Education, 2007.
- [2]. K.O.Elish and M.O.Elish; Predicting defect-prone software modules using support vector machines, The Journal of system and software, Vol. 81, 2008, pp. 649-660.
- [3]. C.Catal, and B.Diri.; A Systematic review of software fault prediction studies, Expert Systems with Applications, Vol. 36, 2009, pp. 7354-7364.
- [4]. Capers. Jones; Applied Software Measurement: 3rd edition; McGraw-Hill, New York, NY, 2008.
- [5]. The Wall Street Journal, Nov 13, 1987.
- [6]. Carson, "Scrap, Defect Rates Cut by 50%," In Quality, Vol. 30, No. 4, April 1996.
- [7]. Humphrey, Watts, "Making Software Manageable." In CrossTalk, 1996,pp. 3-6.
- [8]. Teodorescu, H.N. Zbancioc Metal. "Knowledge Based Systems Applications, performantica, Lassy, 2004.
- [9]. Hussain, O.K.Chang. etal. "A fuzzy Approach to Risk Based Decision Making", Lecture Notes in Computer Science, Springer Berlin/Heidelberg, Vol. 4278, 2006, PP. 1765-1775.
- [10]. Kearney, J.K., R.L.Sedlmeyer, W.B.Thompson, etal, "Software Complexity measurement," Communications of the ACM, Vol.29, No. 11, Nov 1986.
- [11]. A.D. Oral and A.B. Bener; "Defect Prediction for Embedded Software", In 22nd International Symposium on Computer and Information Science, 2007, PP. 1-6.
- [12]. B. Turhan and A. Bener; "Analysis of Naive Bayes assumptions on software fault data"; An empirical study Data & Knowledge Engineering, Vol. 68, 2009, PP. 278-290.
- [13]. N.E.Fenton and M.Neil; "A Critique of Software Defect Prediction Models", IEEE Transactions on Software Engineering, Vol. 25, Issue 5, 1995, PP. 675-689.
- [14]. Garmus. David & Herron. David; "Measuring the Software Process"; A Practical Guide to Functional Measurements; Prentice Hall, Englewood Cliffs, NJ; 1995.
- [15]. Kan. Stephen H.; "Metrics and Models in Software"; Quality Engineering, 2nd edition: Addison Wesley Longman, Boston, MA; ISBN 0-201-72915-6; 2003; 528 pages.