Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 7, July 2015, pg.464 – 475

RESEARCH ARTICLE

Risk Assessment of Multiple Factors using Fuzzy Logic

Anjali Kinra

Department of Computer Sciences, ITM University Gurgaon, India

Kinra.anjali@yahoo.in

Abstract -- Software Cost Estimation is the challenging factor in project management. To complete the project with in time and budget, we need exact cost estimation. But the most demanding aspects of managing project today are the encompassing uncertainties in our estimates. Uncertainties like project durations, project efforts, and project costs with agenda are some concerns. Due to this behavior of the project it is considered as a risky project. Project planning phase includes two basic activities, which measure effort estimation and risk management. These have to be run together because the accuracy of the effort appreciation is highly pendent on the size and nature of the project risks. Risk is one of major factor to identify the criticality of a software project. The risk analysis is based on the evaluation of different factors related to the software product. These factors are divided in several categories. These risk factors include the process risk, product risk, platform risk, schedule risk etc. The proposed work will be defined for some open source software in which dataset is given in terms of module information complimented with risk factors. Large number of effort estimation methodologies has been proposed, which include the COCOMO model. COCOMO provides a complete value rather than an approximation value. Consequently, the existing effort estimation model decline to produce absolute reference for project manager due to its lack of accuracy. Most of the existing research studies based on risk analysis use inefficient technologies which produces unreliable results. It was not able to identify risks in terms of risk factors and criticality using fuzzy logic. Fuzzy Logic was primarily used to check rule based system can solve the software effort estimation drawback. By applying rule based system, we can analyze the process, platform and product based attribute.. Analysis is divided in to two stages; the fuzzification of individual risk and a collaborative analysis with fuzzy modeling. These will be performed to conclude the software quality.

Key Words -- COCOMO, Risk Management, Fuzzy, Cost Estimation, Reliability, effort estimation

I. INTRODUCTION

Software Cost estimation is a divination of the prize of the resources which is required to complete all of the work of the software project. Irregularities are considered as a risk .Risk is to calculate future irregularities in executing performance goals and facts within defined prize, list and performance obligation. Risk can also be joined with various aspects which recite across the Work Breakdown Structure (WBS) and Integrated Master Schedule (IMS). It invokes the potential changes in the planned approach and its result [1]. Whereas these changes may consist favorable as well as against effects, but this approach will only address against future effects since programs have figuratively experienced problem in this region during the acquirement process

This research combines fuzzy logic, which has the potential to contend with situation ambiguity and linguistic variables. Fuzzy Logic used to improve the sensitivity of software project risk assessments using COCOMO. The proposed methodology provides a better result that can be used as a decision support system for an individual project manager or a top management team in making project comparisons based on their risks or to prepare better project risk mitigation approaches.

II. SOFTWARE RISK MANAGEMENT

Risk management is a continued process that is completed through the whole life span of a system. It is a proved strategy for continuously finding and calculating the unascertained; exhibiting moderation options; choosing, planning, and applying suitable risk mitigations; and tracing the implementation to reduce the risk[2]. There are several factors which depends upon which are risk management planning; timely identification and analyses of faults; timely implementation of alternative actions; regular monitoring and reassessment; and conversation, documentation, and coordination.

This model (as shown in fig 2.1) consist the following actions, performed on endless basis:

- Risk Identification
- Risk Analysis
- Risk Mitigation Planning
- Risk Mitigation Plan
- Implementation
- Risk Tracking.

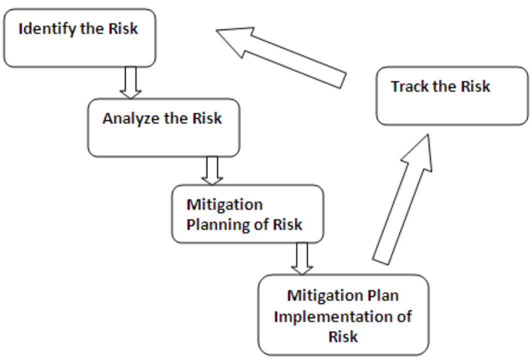

Fig. 2.1 Risk Management Cycle

III. RISK CATEGORIZATION

Risk invariably involves uncertainty and also the potential for loss. Occurrence of faults during development of software project is additionally known as “software risk” [3] and is defined as “a live of probability of an unacceptable outcome affecting the software project, procedure, and item”. Risk administration also plays an important role here, it acknowledging the fact that a software project will be used in an environment where the results are intangible and subject to a higher level of uncertainty compared to the other types of projects. In the Project Planning stage, risk administration activities converges mostly on risk assessment, which is a discovery method of identifying the potential faults, examining or assessing their risk effects, and prioritizing the risks. An activity to identify the risk basically focuses on determining the doable risks, creating a risk statement, and establishing the context of the possible risks as deliverables. On the basis of risk statement and the risk context, all aspects of the risks will be analyzed and prioritized, so that a project manager can determine where action should be taken to manage such risks . Hence, the risk assessment phase will provide the information about the number of risks and an estimate of the risk-exposure relationship of each. In order to support project managers in a development of software project, various models have been developed to help in the Effort Estimation and Software Risk Assessment. The most significant effort estimation models that have been used in software development projects are the Constructive Cost Model (COCOMO), the System Evaluation and Estimation of Resource Software Evaluation Model (SEER-SEM) ,and the Software Life Cycle Management (SLIM) model. Barry Boehm developed COCOMO model in 1980's, which is the most widely used estimation model for software project. Even though the risk assessment (identification and analysis) is conducted together with the software estimation in the Project Planning phase, analysis and identification of risk is usually done separately from cost estimation. The E-COCOMO model improves this process by utilizing the information taken from the effort estimation step to establish a risk assessment for a particular software project. The cost factors including scale factors and cost drivers in the COCOMO model become the inputs for the E-COCOMO model. The result for the model is a list of software risks that are related to the COCOMO price factors, such as: Schedule Risk, Product Risk, Platform Risk, Personnel Risk, Process Risk, and Reuse Risk. The software risk taxonomy within the Expert-COCOMO model is presented in Fig 3.1below:

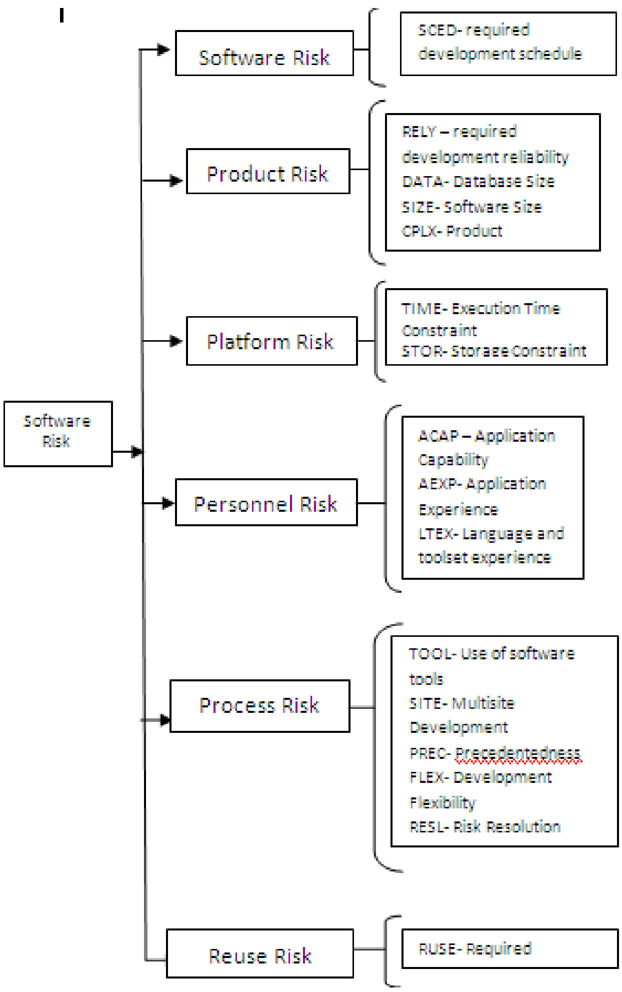

Fig 3.1 Risk Categorization

The term fuzzy logic (FL) is not a balanced system; however it could be a processed data by allowing partial set membership rather than crisp set membership [3]. It supports fuzzy set theory and introduced in 1965 by Prof. L.A Zadeh[6] in the paper fuzzy sets. If feedback controllers are well programmed just to accept clamorous and exact input. Controllers may be easier to apply and adequate. FL is a critical-solving control framework that contributes to perform implementation within the system ranges from simple, small, embedded micro-controllers to extensive, networked, or workstation-based knowledge acquirement and management systems. This may be enforced in both hardware and software and also individually.

IV. FUZZY SYSTEMS

Fuzzy systems are knowledge based or rule based system [4]. Fuzzy systems is a knowledge base consisting so called as Fuzzy "If Then rules" in which few words are characterized by continuous member functions. Fuzzy logic systems would be classified into three types of systems: pure fuzzy logic system, Takagi and Sugeno's fuzzy system and fuzzy logic system with fuzzifier and defuzzifier. Since large portion of the designing applications generate crisp data as input and rest is taken as output. The end point is the last type is the most broadly used fuzzy logic system with fuzzifier and defuzzifier. This system applied to a various industrial processes and consumer products as shown in below Fig:

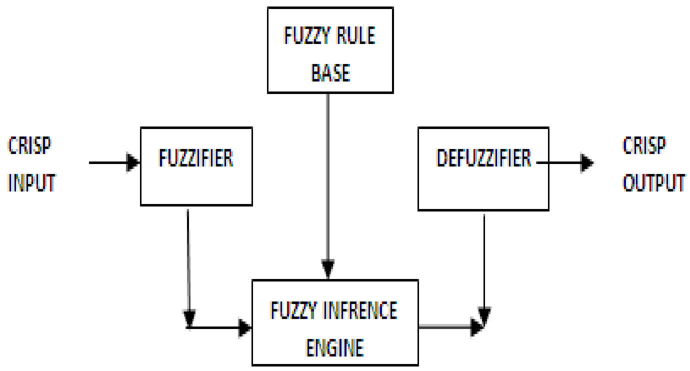

Fig. 4.1 Fuzzy System

A. *Fuzzifier*

It converts the crisp input into a fuzzy set, and Fuzzifier used to describe situation graphically, to transform the crisp input in to fuzzy set. Membership functions are used for transformation.

A. *Fuzzy Rule Base*

It uses "if-then rules" formulae.

B. *Fuzzy Inference Engine*

A collection of if -then rules stored in fuzzy rule base is known as inference engine. It does two basic functions i.e., collection and composition.

C. *Defuzzification* – Defuzzification transforms fuzzy output into crisp output.

V. PROPOSED MODEL

A. Flow of Work

The proposed Fuzzy based model will estimate software quality using Risk Parameter. Initially, collect the database from PROMISE repository [5]. This repository data set made openly accessible keeping in mind the end goal to support repeatable, verifiable, refutable and improvable predictive models of software engineering. It includes large number of risk attributes. Every attribute is described according to their category. Generally, it has four risk category, which are personal, platform, process and reuse. These categories are further split into different sub categories as shown in Fig 4.1 Now apply the fuzzification on each risk attribute. Assign weight to each sub category. At last apply the rule based system on each software risk attribute. It saves information set made openly accessible keeping in mind the end goal to support repeatable.

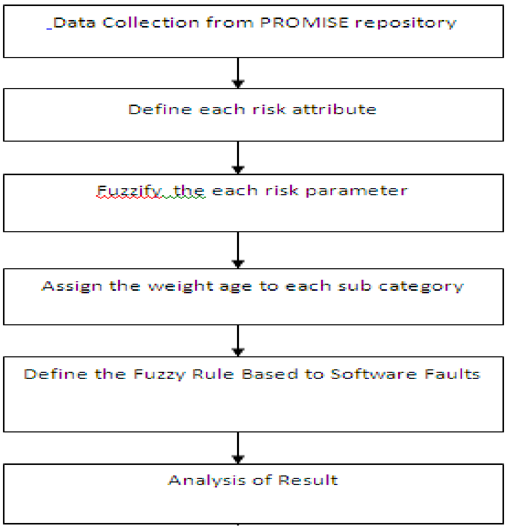

Fig 5.1 Flow of Work

B. Data Collection

To create an appropriate model, one extracts the data from the PROMISE repository[7]. This repository dataset made publically available to encourage repeatable, verifiable, refutable and improvable predictive models of software engineering. This dataset contains various risk attributes.

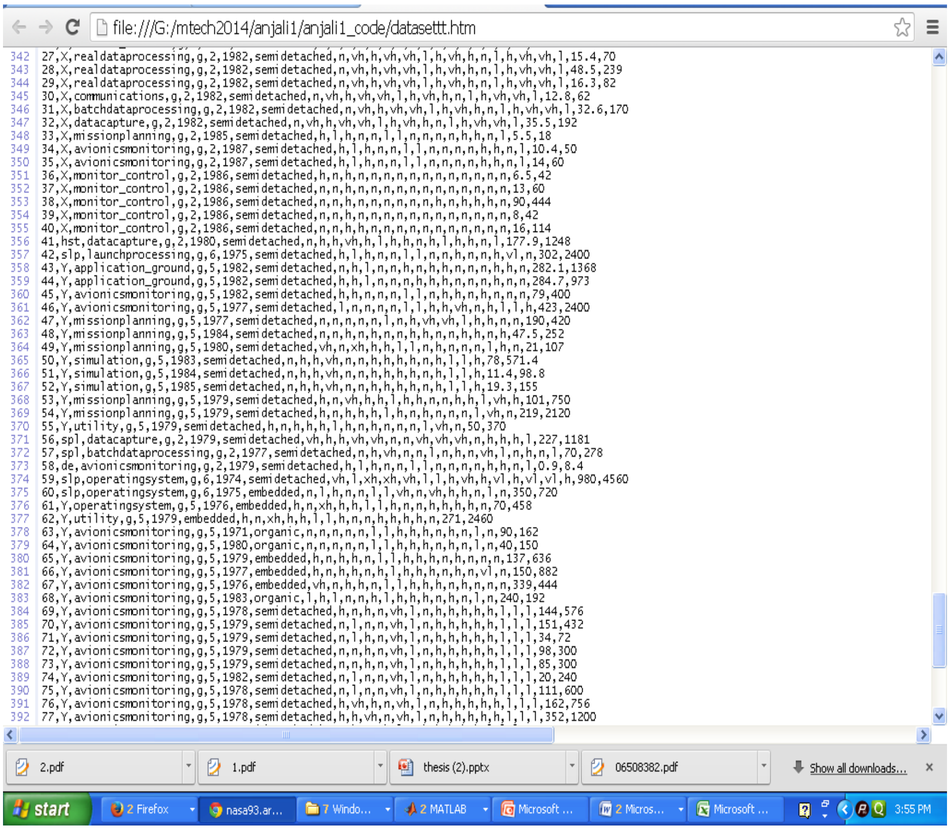

Fig 5.2 Data Collection

C. Categorization of Risk

Risks are categorized in 5 basic types of risk which are Product, Process, Personal, Platform and Reuse risk. Categorization of risk attributes are given in table below:

TABLE 5.1

CATEGORIZATION OF RISK

S.No	Risk Occurred	ID	Attribute Name	Purpose
1	Personal Risk	ACAP	Analyst Capability	It describes the capability of the analyst. Its rating should not consider the level of experience
2		PCAP	Programmer Capability	Evaluation depends upon the capability of the programmer as a team not individuals.
3		AEXP	Analyst Experience	Rating should be done on the level of application experience of project team
4		VEXP	Virtual Machine Experience	It describes how efficiently programmer can use the hardware.
5		LEXP	Language Experience	It measures the level of programming language and software tool experience of project team developing the software system
6		MODP	Modern Programming Practices	It describes the programming knowledge of the team member and rate them according to their levels.
7	Process Risk	TOOL	Use of Software Tools	Tool rating ranges from simple edit and code.
8		TIME	Time Constraint	percentage of available execution time expected to be used by the system or subsystem consuming the execution time resource.
9		TURN	Turnaround Time	This attribute defines how much time it will take to complete the project
10		DATA	Data base size	It captures the influence test data requirements have on program development
11		SCED	Schedule Constraint	It specifies a rating that represents the degree of schedule constraint imposed on a system software
12		STOR	Main Memory Constraint	It specifies a rating that represents thedegree of main storage constraint imposed on a software system or subsystem.
13	Reuse Risk	VIRT	Machine Volatility	,, describes the volatility of the machines ,how it works efficiently
14		CPLX	Process Complexity	It describes the rating of complex processes of system software.
15		RELY	Required software Reliability	Software must perform its intended function over a period of time.

Now every parameter is distributed according to their category. One risk attribute may belong to two different categories. Weight age would be assign to every risk attribute.

D. Weight age Assignment

Risk attributes are categorized in 3 basic categories: low risk, medium risk and high risk. Weight age would be assigned randomly to the risk attribute. Here, we took 10 risk attributes with their category and assigned weight to them. Weight age assessment is shown in below table:

TABLE 5.2

WEIGHT AGE ASSIGNMENT

Type of Risk	Category	Weight age
R1	Low Risk	.1
R2	Low Risk	.2
R3	Low Risk	.3
R4	Medium Risk	.4
R5	Medium Risk	.5
R6	Medium Risk	.6
R7	Critical	.7
R8	Critical	.8
R9	Critical	.9
R10	Critical	1

E. Module Assessment

Once the risks are categorized, the next work is to categorize the risk under different vectors. The available risk-module vectors are listed here:

TABLE 5.3

MODULE ASSESSMENT

Module	Risks	Weights
M1	R1,R2,R3	.1+.2+.3=.6
M2	R1,R2,R3,R4	.1+.2+.3+.4=1
M3	R1,R2,R3,R6,R7	.1+.2+.3+.6+.7=1.9
M4	R1,R6,R9	.1+.6+.9=1.6
M5	R1,R2,R3	.1+.2+.3=.6
M6	R1,R2,R6,R9	.1+.2+.6+.9=1.8
M7	R1,R3,R4,R8,R10	.1+.3+.4+.8+1=2.6
M8	R1,R3,R4,R5,R8	.1+.3+.4+.5+.8=2.1
M9	R1,R3,R10	.1+.3+1=1.4
M10	R8,R9,R10	.8+.9+1=2.7

F. Table of Efforts

The presented research work is defined as the effort table is given in fig. It defines the effort multiplier with their ranges from “very low to extra high”. This dataset of effort multiplier is extracted from NASA repository[5]. It shows the range of each and every multiplier.

TABLE 5.4

EFFORT MULTIPLIER

Range of Efforts	VL	L	N	H	VH	EH
ACAP	1.46	1.19	1.00	0.86	0.71	
PCAP	1.42	1.17	1.00	0.86	0.70	
AEXP	1.29	1.13	1.00	0.91	0.82	
MODP	1.24	1.10	1.00	0.91	0.82	
TOOL	1.24	1.10	1.00	0.91	0.83	
VEXP	1.21	1.10	1.00	0.90		
LEXP	1.14	1.07	1.00	0.95		
SCED	1.23	1.08	1.00	1.04	1.10	
STOR			1.00	1.06	1.21	1.56
DATA		0.94	1.00	1.08	1.16	
TIME			1.00	1.11	1.30	1.66
TURN		0.87	1.00	1.07	1.15	
VIRT		0.87	1.00	1.15	1.30	
RELY	0.75	0.88	1.00	1.15	1.40	
CPLX	0.70	0.85	1.00	1.15	1.30	

G. Risk Assessment using Fuzzy Rule Based Approach

1) Fuzzy Rule Based System

Considering the input parameter x from the universe X, and the output parameter y from the universe Y, the statement of a system can be described with a rule base (RB) system in the following form:

Rule1: IF x=A1 THEN y= B_1

Rule2: IF x=A2 THEN y= B_2

Rule n: IF x=An THEN y= B_n

This is denoted as a single input, single output(SISO) system

If there is more than one rule proposition, i.e. the i^{th} rule has the following form

Rule i: IF x_1=A_1i AND x_2= A_2i…THEN y=bi

Then, this is denoted as a multi input, single output (MISO) system.

H. Research Design

The present work has defined a two layer fuzzy approach to analyze the software risk under different vectors. Some of the specific features of using fuzzy logic approaches in software reliability prediction are:

• It is easy to design and construct models for reliability growth of varying complexity at different points of time for a given data set.

• Fuzzy Logic based models are easily adaptable in complex operational phase for different failure datasets.

I. Code Modules

The description of the code module is given here in Table 5.5 below

Table 5.5

Code Modules

File Name	Description
AnalystCapabilityF.m	Convert the analyst nominal values to numerical continuous values.
ApplicationExpF.m	Convert the Application Experience nominal values to numerical continuous values.
DatabaseSizeF.m	Convert the Database Size nominal values to numerical continuous values.
languageExpF.m	Convert the Language Experience nominal values to numerical continuous values.
MachineVolatalityF.m	Convert the Machine Volatility nominal values to numerical continuous values.
MainMemoryConstraintF.m	Convert the Main Memory nominal values to numerical continuous values.
MordrenProgrammingPracF.m	Convert the Modern Programming Practice nominal values to numerical continuous values.
ProcessComplexityF.m	Convert the Process Complexity nominal values to numerical continuous values.
ProgrammerCapabilityF.m	Convert the Programmer Capability nominal values to numerical continuous values.
ReqSoftwareRelF.m	Convert the Software Release nominal values to numerical continuous values.
ScheduleConsF.m	Convert the Schedule Constraint nominal values to numerical continuous values.
TimeConstraintF.m	Convert the Time Constraint nominal values to numerical continuous values.
TurnAroundTimeF.m	Convert the Turn Around Time nominal values to numerical continuous values.
UseOfToolF.m	Convert the Use of Tool nominal values to numerical continuous values.
VmachineexpF.m	Convert the Virtual Machine Experience nominal values to numerical continuous values.
Main.m	This is the main program file that will accept the data values and process it under different weight age functions

J. Analysis of Result

Here, we apply the fuzzy rule based system to the risk attributes and their results are categorized in Very Low, Low, Medium, High and Very High.

Result has shown in below Fig5.3 describes the category of risk occurred. It shows that risk is of very low category. Middle straight line denotes that attribute is not risky.

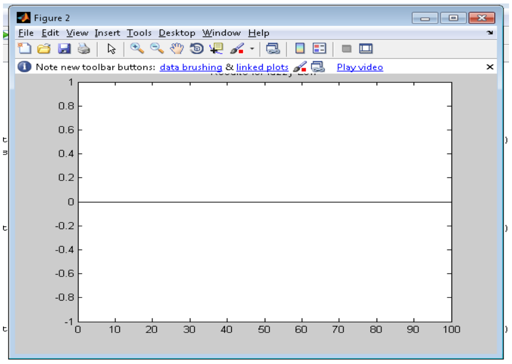

Fig 5.3 Very Low Category Risk

Result has shown in below Fig 5.4 defines that risk is of Low category. It means Probability of occurrence of risk is very less.

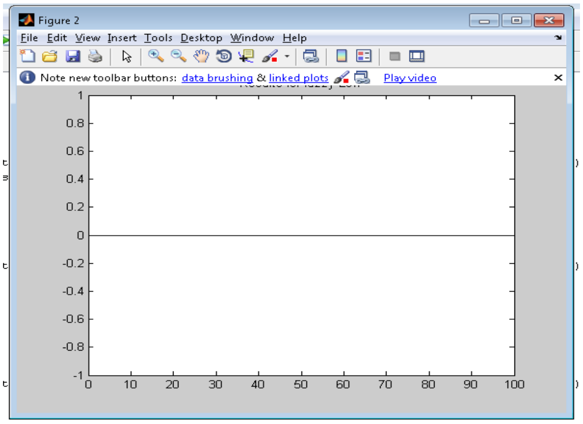

Fig 5.4 Low Category Risk

MATLAB result shows the Medium Category of risk in Fig 5.5. It shows that risk is of medium quality.

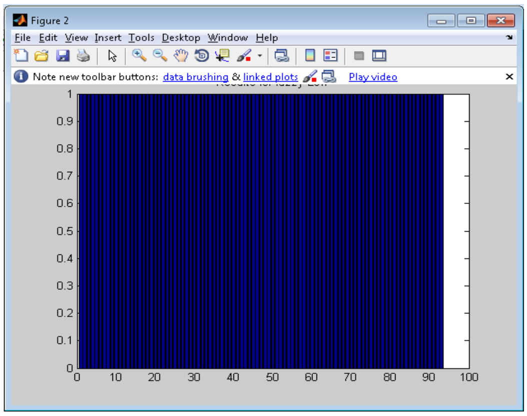

Fig 5.5 Medium Category Risk

Result has shown in below Fig 5.6 defines that risk is of Low category. It means Probability of occurrence of risk is high. It increase the criticality level of the attributes.

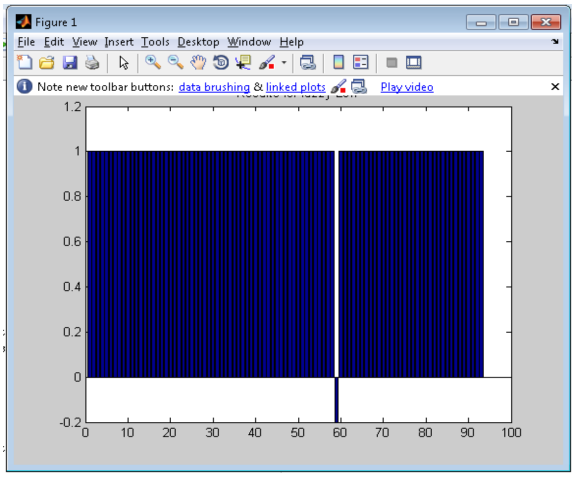

Fig 5.6 High Category Risk

Result has shown in below Fig 5.7 describes that risk is of very high category. It means Probability of occurrence of risk is very high. This type of risk increases the critical level. It directly affects the quality of the software.

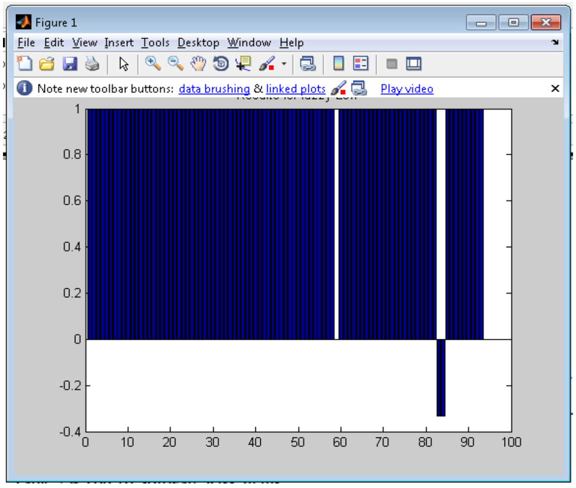

Fig 5.7 Very High Category Risk

VI. CONCLUSION & FUTURE SCOPE

Above research describes in this thesis introduces a new Model, which has the following characteristics and capabilities

The proposed work would be defined for some open source software in which the dataset is given in terms of module information correlated with the risk factors. The work is to categorize the large number of risks. After the categorization, work is to perform the distribution over these dataset respective to risk category. Once the distribution is done, it is required to identify the risky modules in the software system. Once the statistics are obtained, the next work is to perform the aggregation on these risk vectors so that the combined software risk will be identified.

To perform this risk analysis, find out the different kinds of risks which can occur in any module of a project. Analyze the software risk in terms of criticality and categorize them. Now the fuzzy logic is applied for the aggregation of risk.

By applying the fuzzy rule base to reduce the uncertainty produces better results rather than the existing COCOMO Model. It can take multiple attributes at a time and calculate the criticality of each parameter. It reduces the average time to calculate the occurrence of risk and their criticality level.

We will explore the applicability of fuzzy logic models for better prediction of reliability in a realistic environment and present an assessment method of software reliability growth using connectionist model

VII. ACKNOWLEDGEMENT

I take this opportunity to express my sincere thanks and deep gratitude to those people who extended their wholehearted co-operation and have helped me in completing this paper.

My deepest thanks to **Dr. SUPRIYA P.PANDA**, for guiding and correcting various documents of mine with attention and care. She has taken the pain to go through the complete paper again and again and make necessary correction as and when needed.

I also extended my heartfelt thanks to my family and well wishers.

REFERENCES

[1] T.N. Sharma, "*Analysis of Software Cost +Estimation using COCOMO II*", International Journal of Scientific & Engineering Research Volume 2, Issue 6, ISSN 2229-5518, pp. 1-5., June-2011

[2] Laurie Williams, "Risk Management", © Laurie Williams 2004

[3] Anjali Kinra, - "A Fuzzy Based Model for Software Quality Estimation Using Risk Parameter Assessment", International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 3, Issue. 3, pp. 807 – 814, ISSN 2320–088, March 2014.

[4] Lucas Layman," Mining Software EffortData: Preliminary Analysis of Visual Studio Team System Data", MSR'08, Leipzig, Germany. ACM 978-1-60558-024-1/08/05, May 10–11, 2008.

[5] Https://code.google.com/p/promisedata/source/checkout

[6] Zadeh. L. A., Fuzzy Sets, Information and Control, Vol. 8, pp. 338-353, 1965

