IJCSC International Journal of Computer Science & Communication Vol. 1, No. 2, July-December 2010, pp. 391-394

A Fuzzy Approach for Successful Reuse Metrics

Jyoti Mahajan[1], V. Mansotra[2] & Devanand[3]

[1]Govt. College of Engg. & Technology, Jammu
[2,3]Department of Computer Science & I.T., University of Jammu
Email: [1]jyoti_1972@sify.com

ABSTRACT

Software Development has never been a perfect process and software projects have often failed or face the challenges because of cost overrun, poor quality, incomplete assessment of requirements, and delays in schedule. Therefore, the biggest issue of software development management is to improve the development process to avoid or minimize the levels of these pitfalls. That's why developers create reusable assets that can be used in future projects. Reusability not only decreases costs, it also improves the quality of projects as reusable assets were tested components[1]. The purpose of this research is to identify and provide insight into the critical factors that are helpful in maintaining discipline in agile projects concerning reusability. This study is significant in the sense that the agile methodology has not been considered disciplined, so we have worked on 14 metrics taken from morisio dataset and further reduced them to six affecting reusability the most.

Keywords: Fuzzy Logic, Software Reuse, Linguistic Variables, Defuzzification, Reusability

1. INTRODUCTION

Software becomes increasingly expensive to develop and is a major cost factor in any information system budget. Software development costs often get out of control due to lack of measurement and estimation methodologies. Software cost estimation or software effort estimation is the process of predicting the effort required to develop a software system.A key factor in selecting a cost estimation model is the accuracy of its metrics, since these models rely on metrics as their input. Metric can be defined as a quantitative measure of the degree to which a system, component, or process possesses a given attribute. Previous studies show that only 15% of the software development effort is devoted to programming, and reusability is one of the major contributors. This study suggests that such a view of reusability factor represents a gross oversimplification. The variance in lines of code alone is not an adequate measure of reusability factor in this context, and development effort will generally not vary with reusability in a linear fashion. In fact, this study will show that reusability depends on many other factors which are quite essential in determining the software development effort.

2. RESEARCH METHODOLOGY

The main objective of this study is to predict the impact of reusability on estimation of software development effort.

1. Selecting Fuzzy Variables

To use fuzzy logic, all input and output variables that define the behavior of new metric set are needed to be defined or identified. For this purpose, we have started from the sub metrics identified in the new metric set. Fuzzy Linguistic variables[2] are used as indicators of these sub metrics. The sub metrics will serve as input parameter while the Reusability component is the output parameter.

Table 1
Fuzzy Variables for New Model

Category	*Fuzzy Variable*		
A Input Variables	1	MCOM	Management Commitment
	2	SEXP	Staff experience
	3	INTM	Independent team
	4	RWPY	Rewards Policy
	5	DMAS	Domain Analysis
	6	SFSZ	Software size
	7	DVAP	Effectiveness of Development Approach
	8	SPMT	Software Process Maturity
	9	RUPE	No. of Reuse Process Introduced
	10	NRPE	No. of Non-Reuse Processes Introduced
	11	RUIN	Reusability Integration
	12	RUOG	Scratch Development – Origin
	13	RUEG	Engineering approach
	14	CNMT	Configuration management
B Output Variables	15	JRUSE	Resultant Reusability metric

2. FUZZY SET DEFINITIONS

To assess each of the sub metric, Fuzzy JRUSE uses three fuzzy sets LOW, MEDIUM, HIGH, to describe each of the input sub metric and the output adjustment factor as they relate to the overall project effort. The concept of a sub metric being MEDIUM means that the attributes being measured by the sub metric are normal and will have little or no impact on metric output. The concept of a sub metric being either LOW or HIGH has the meaning that the output will be affected by the attribute represented by the sub metric.

$$low(x) = \begin{cases} 1 & x < .8 \\ -5x + 5 & x \in [.8.1] \\ 0 & x > 1 \end{cases}$$

$$\text{medium}(x) = \begin{cases} 0 & x < .8 \\ 5x - 4 & x \in [.8.1] \\ -5x + 6 & x \in (1.1.2] \\ 0 & x > 1.2 \end{cases}$$

$$\text{high}(x) = \begin{cases} 0 & x < 1 \\ 5x - 5 & x \in [1.1.2] \\ 1 & x > 1.2 \end{cases}$$

3. RULE IDENTIFICATION

A predictive function in the form of fuzzy expert system is used to calculate the adjustment factor. The collection of rules forms the knowledge bank for fuzzy expert systems. Each rule was developed by using the relationship of sub metric to adjustment factor. Three rules are used to relate the significance of each sub metric to adjustment factor. One rule was used for each possible state of each sub metric. Therefore, the rule base for fuzzy JRUSE contains 45 rules.

4. MODEL OPTIMIZATION

Fuzzy JRUSE was optimized by selecting the best performing model over a variety of configurations. For each configuration tested, the sum of Chi square error was calculated as follows:

$$Chi^2(x_i) = \sum_i \frac{(x_i - e_i)^2}{x_i}$$

Where x_i is actual value and e_i is calculated value for project $_i$. As measured by least Chi Square sum, the fuzzy JRUSE configuration with best performance was selected for testing.

4.1. Fuzzy Input Set Configuration

Fuzzy set definitions were adjusted over the range of 0.7 – 1.66 [4] using increments of 0.1 for MEDIUM. As described earlier, LOW and HIGH fuzzy set definitions are left and right shouldered functions. The same configuration is used for all 15 fuzzy input variables. Table 2 shows the resulting three configurations.

Table 2
Fuzzy Input Set Configuration

Configuration	*Fuzzy Input Set Definitions*		
	Low	*Medium*	*High*
1	.7/1, .9/1, 1/0	.9/0, 1/1, 1.1/0	1/0, 1.1/1, 1.66/1
2	.7/1, .8/1, 1/0	.8/0, 1/1, 1.2/0	1/0, 1.2/1, 1.66/1
3*	.7/1, .7/1, 1/0	.7/0, 1/1, 1.3/0	1/0, 1.3/1, 1.66/1

*Best Configuration

4.2. Fuzzy Output Set Configuration

Fuzzy set definitions were adjusted over the range of 0 – 4.0 using increments of 0.1 for MEDIUM. As described earlier, LOW and HIGH fuzzy set definitions are left and right shouldered functions. Table 3 shows the resulting five configurations.

Table 3
Fuzzy Output Set Configuration

Configuration	*Fuzzy Output Set Definitions*		
	Low	*Medium*	*High*
1	.25/1, .9/1, 1/0	.9/0, 1/1, 1.1/0	1/0, 1.1/1, 4/1
2	.25/1, .8/1, 1/0	.8/0, 1/1, 1.2/0	1/0, 1.2/1, 4/1
3	.25/1, .5/1, 1/0	.25/0, 1/1, 1.5/0	1/0, 1.5/1, 4/1
4	.25/1, .3/1, 1/0	.3/0, 1/1, 1.7/0	1/0, 1.7/1, 4/1
5*	.25/1, .25/1, 1/0	.25/0, 1/1, 1.75/0	1/0, 1.75/1, 4/1

*Best Configuration

5. DATA ANALYSIS

5.1. Multiple Linear Regression (MLR) Model

Multiple Linear regression (MLR)[8][3] is a method used to model the linear relationship between a dependent variable and one or more independent variable. MLR is probably the most widely used method in statistical analysis. The model expresses the value of a predictand variable as a linear function of one or more predictor values and an error term:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots.. + \beta_k x_k + \varepsilon$$

where

y = dependent variable

x = Independent variable(s)

ε = Statistical error

β = Regression Coefficient

All 14 independent variables were entered for this model. Following is the analysis and the results.

5.2. Regression Analysis

Table 4 shows the R and R^2 values of the regression model for output variable JRUSE (the Metric concerning reusability in Software Development Projects). R is the multiple correlation coefficient, i.e. the linear correlation between the observed and model-predicted value of JRUSE. The R value was above 0.500 mark, and through the R^2 value, this model explains 98% of the total variance.

Table 4
Summary of the Regression Model for Output Variable JRUSE

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.9962	0.9925	0.9808	0.2860

Table 5 shows the detail of the coefficients of all 14 independent variables. This table shows the t-value, which signified the probability of sample occurrence in testing the null hypothesis. The table shows the un-standardized β coefficient which explains how much the variable contributed to the prediction of JRUSE having all other variables remain constant. There were three variables with relatively larger values for the coefficient β as well as higher *t* values.

- Origin of Reusable Assets (β = 1.1252, *t* = 3.8717).
- Engineering Approach (β = 0.8947, *t* = 2.9117).
- Management Commitment (β = 0.6223, *t* = 10.2508).

There were three variables having significance value at below 0.1.

Table 5
Regression Coefficient for all Input Variables in the Final Metric Set for JRUSE

Model	*Variable*	*Unstandardized Coefficients*			
		β	*Std. Error*	*T*	*Sig.*
1	(Constant)	-6.5202	2.1867	-2.9817	0.0154
	Management Commitment	0.6223	2.0607	10.2508	0.0000
	Staff experience	-0.0480	0.0609	-0.7875	0.4512
	Independent team	-0.5005	0.1633	-3.0645	0.0135
	Rewards Policy	0.1030	0.0668	1.5416	0.1576
	Domain Analysis	0.0010	0.1866	0.0053	0.9959
	Software Size	-0.1357	0.0901	-1.5049	0.1666
	Effectiveness of Development Approach	-0.2569	0.1124	-2.2856	0.0481
	Software Process Maturity	-0.1901	0.0768	-2.4753	0.0353
	No. of Reuse Processes Introduced	0.0739	0.1761	0.4196	0.6846
	No. of Non-Reuse Processes Introduced	-0.0670	0.1283	-0.5224	0.6140
	Reusability Integration	-0.1283	0.0990	-1.2955	0.2274
	Scratch Develop-ment – Origin	1.1252	0.2906	3.8717	0.0038
	Engineering Approach	0.8947	0.3073	2.9117	0.0173
	Configuration Management	-0.1647	0.1021	-1.6127	0.1413

Figure 5.1 displays the histogram of the residual in the regression model for output variable JRUSE. Residual represents unexplained variation after fitting a regression model. It is the difference between the observed value of the variable and the value suggested by the regression model, and the histogram helps determine the assumption of normality of the residual distribution. The shape of the histogram in Figure 5.1 is acceptably close to the curve.

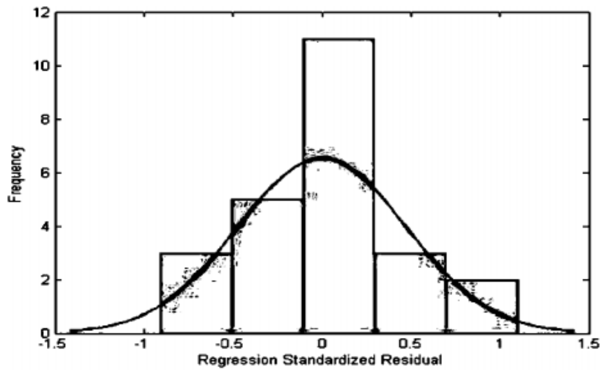

Fig. 5.1 : Histogram of the Residual in the Regression Model

Figure 5.2 shows the Residual case order plot of the regression model for JRUSE. The interval around some of the residual, shown in red when plotted, do not contain zero. This indicates that the residual is larger than expected in 95% of new observations, and suggests that the data point is an outlier.

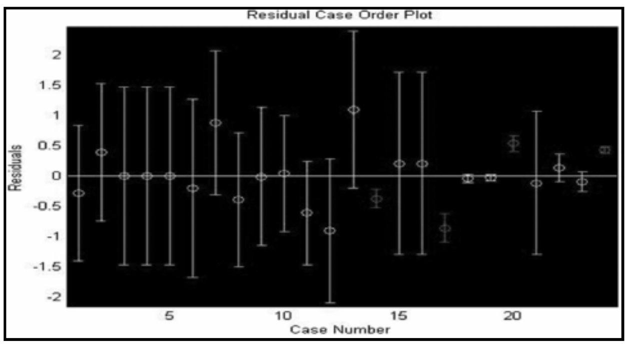

Fig. 5.2 : Residual Case Order Plot

6. RESULT

This research study is set out to explore the critical factors affecting reusability in a software development project using quantitative methods. The data collected from 24 projects from a diverse group of organizations of various sizes, and industries provided enough empirical information for statistical analysis to arrive at a number of conclusions.

Although a large number of factors, affecting reusability in software projects, are discovered; the multiple regression analysis revealed that the following factors contribute successfully in estimating the effect of reusability on software development effort.(based on positive β value):

- Scratch Development – Origin,
- Engineering Approach,
- Management Commitment,
- Rewards Policy,
- No. of Reuse Processes introduced,
- Domain Analysis.

7. FUTURE SCOPE

This exploratory study is among the first in the academic area to search for empirical evidence to the role of reusability in software development project. As such it provides a starting point for further research related to the importance of reusability in estimating the effort of software development, something that so far has been comprised of only anecdotal evidence in the practitioner literature. Such an approach would allow a researcher to discover the other aspects of software development process such as human, organizational, economical, and societal dimensions.

REFERENCES

[1] Basili, Victor R., and David M. Weiss, "A Methodology for Collecting Valid Software Engineering Data", *IEEE Transactions on Software Engineering*, 10.6 (1984): 728-738. ABI/INFORM Global, ProQuest. Web. 04 Nov, 2009.

[2] Pedrycz, Witold and Fernando Gomide, *An Introduction to Fuzzy Sets: Analysis and Design*, Massachusetts: MIT Press, 1998.

[3] Briand, Lionel, Sandro Morasca, and Victor R. Basili, "An Operational Process for Goal-driven Definition of Measures", *IEEE Transactions on Software Engineering* 28.12 (2002): 1106. ABI/INFORM Global, ProQuest. Web. 04 Nov. 2009.

[4] Idri, Ali, Alain Abran, "COCOMO Cost Model Using Fuzzy Logic" *7th International Conference on Fuzzy Theory & Technology*, Atlantic City, New Jersey, March 2000.

[5] Morisio, Maurizo, Michael Ezran, and Colin Tully. 2002, "Success and Failure Factors in Software Reuse", *IEEE Transactions on Software Engineering*, **28**, No. 4, (April 1): 340-357.

[6] Bidoit, Michael, *Algebraic System Specification and Development: a Survey and Annotated Bibliography*, New York: Springer, 1991.

[7] Perry, D., A. Porte, L. Votta, "Empirical Studies of Software Engineering: A Roadmap", *Futures of Software Engineering*, Ed. Anthony Finkelstein, ACM. 2000.

[8] Wohlin, Claes, Per Runeson, Martin Host, Magnus C. Ohlsson, Bjorn Regnell, Anders Wesslen, *Experimentation in Software Engineering: An Introduction*. London: Kluwer Academic Publishers, 2000.

[9] Zadeh, L. A. *Fuzzy Sets*, Victoria: Abe Books, 1965.