Pairwise Programming Software Development
Reliability Prediction by using GAM

Somchai Prakancharoen
Department of computer and information science, Faculty of applied science
King Mongkut’s university of technology north bangkok
CountryBangkok, Thailand
spk@kmutnb.ac.th

Abstract—The objective of this research was to construct a method of choosing right two programmers to work together in pairwise programming in order to decrease software development error rate. Independence variable were programmer’s competency and personal type which were considered of suitable with particular software requirement. Thirty finished software projects, pairwise software development were collected then GAM technique was used to construct a prediction software project error rate. R² of GAM equation was 0.616. Accuracy test of prediction was 78.70%

Keywords- Pairwise programming, Generalized additive model

I. INTRODUCTION

Pairwise programming software development is a practical method of increasing the software quality, cross checking, productivity etc.

Most of software project have their specific requirements and constraints. If software project manager could select a right couple of programmers that match to project’s requirement and constraints then software project might be success in purpose. This research suggests a solution of choosing a right two programmers to work in a particular distinct software project. Experiment observations were gathered from thirty finished pair wise software development projects. Prediction equation was constructed from all thirty observations with generalized additive model. Project’ software error rate is a dependent variable that project manager want to know in order to select the right two programmers. Prediction equation fitting to experiment observation was explained in R².

II.RELATED THEORY AND RELATED RESEARCH

A. Independence and dependent variable

Interested independence variable that should predict software development error rate were gathered from many research [1, 2, 3] then these variables were chosen by three experts in three Thailand software houses.

• Independence variable

There are five competencies and one personal type of each programmer.

Programmer#1

X1_1 = How relevance experience to assigned software project - RE (1-100%)
X1_2 = Relevance knowledge-RK(1-100%)
X1_3 = Task availability-TA (1-100%)
X1_4 = Human relationship -HR(1-100%)
X1_5 = Attention-AT (1-100%)
X1_6 = Personal type – PT (Professional oriented-Task oriented-Interaction oriented (Likerts score range 5-5-5))

[represented as X1_6_1 = PO, X1_6_2 =TO, X1_6_3 =IO]

Programmer# 2

X2_1 = How relevance experience to assigned software...
Error rate, Y, is defined as percentage of software error rate, for example Y=10%.

\[ \text{Error rate}(Y) = \frac{\text{# Line -of- error}}{\text{Total line -of-sourcecode}} \]  
(1)

- For individual software project, there have difference in required personal type that should be preferred in the software project difference stage.
- For example, the personal type of human resource that suitable in software requirement engineering stage were professional type =4 of 5, task oriented = 2 of 5 and interaction oriented= 3 of 5.
- This proportion could be represented as (4:2:3).
- The proportion of personal type is not the same amount in difference software development stage.
- However, in the same software development stage under difference infrastructure should have difference personal type proportion.
- It was based on software project manager consideration.
- The required personal type variables (e.g. person#1&#2) were defined as followed.
  - X2_2 = Relevance knowledge-RK (1-100%)
  - X2_3 = Task availability-TA (1-100%)
  - X2_4 = Human relationship-HR (1-100%)
  - X2_5 = Attention-AT (1-100%)
  - X2_6 = Personal type – PT (Professional oriented-Task oriented-Interaction oriented 
    (Likerts score range 5-5-5))

\[ \text{Dependent variable} \]

- \[ Y \text{ (x)} \] of \[ i=1, n \] \[ \beta \text{ (x)} \] \[ j=1, k \] are prediction equation for \[ Y \].
- \[ \beta \text{ (x)} \] \[ j=1, k \] are coefficients (weighting) of each \[ f_j \text{ (x)} \].
- \[ n \] is an number of observations which should be equal or not in each \[ f_j \text{ (x)} \].

\[ Y_{\text{estimate}} = \beta_0 + \sum_{j=1}^{k} (\beta_j \cdot f_j \text{ (x)}) ; i=1, n \]  
(2)

While \[ f_j \text{ (x)} ; j=1,k \] are prediction equation for \[ Y \].
- \[ \beta_0 \] is an intercept to \[ Y \] axis.
- \[ \beta_j; j=1,k \] are coefficients (weighting) of each \[ f_j \text{ (x)} \].

Parameters of (1) should be calculated from least square error method as equation (3) and (4).

\[ e^2 = (Y - Y_{\text{estimate}})^2 \]  
(3)

\[ \frac{\partial e^2}{\partial \beta_0}, \frac{\partial e^2}{\partial \beta_1}, \ldots \frac{\partial e^2}{\partial \beta_k} \Rightarrow 0 \]  
(4)

C. Mean magnitude relative error: MMRE

MRE is a formula of error measurement of estimated value difference from real value. If there are many case then average of MRE (MMRE) was shown in equation (5).

In order to simply understand, multiply MRE with 100 and subtract it from 100 then this result should represent accuracy of estimation in percentage.

\[ \text{MMRE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right| \]  
(5)

\[ \text{Accuracy} = 100 - \text{MMRE} \times 100 \]  
(6)

D. Related research

[1] There are three essential fundamental persona competency of IT such as effectiveness competency, academic competency, workplace competency. [1] has identify that the knowledge, skills, and abilities are needed for workers to perform successfully in the field of IT.

[2] This paper suggest software reliability prediction model on secrecy software development project based on software metrics. Metrics measurement were considered on Defect density (DD), Test coverage (TC), Requirements traceability (RT), Function point analysis (FP) and Bugs per line of code (BLOC)

[3] Software reliability was indirectly measured through software metrics by four categories: Product metrics, Project management metrics, Process metrics and Fault and failure metrics.

III. RESEARCH METHODOLOGY AND RESULT

According to actually working, there are difference manners. Task may be assigned to only one programmer or group of programmers. Therefore in this research, task allocation should be considered in two accounts.

The first is that how to predict efficiency of task perform under working by individual programmer.

The second is that how to predict efficiency of task perform under working together by two programmers.

To solve the first question, regression analysis of each programmer competencies were calculated in order to predict Y, represented as \[ \hat{Y}_i \].

When two selected programmers were work together, their three personal types of both were calculated for average value of PO, TO and IO.

Then, these three average values were subtracted to project’s defined PO, TO and IO respectively. These three
results were summed together which was called as grouping personal type \((XG_{-6,12})\).

To solve the second question, regression analysis of both programmer personal types were calculated in order to predict \(Y\), represented as \(\hat{Y}_2\).

### A. Experimental data

Experiment observations were collected in two periods of time.
- **The first data collection.**
  Task (Program) was assigned Individual programmer, which has his own competency.
  Personal type of each one was not considered in this period since each programmer was work alone. The allocated task was prepared for single person to design, coding and testing until this program was correctly finished. At the finished time, percentage of software’s error rate was computed. Percentage of software error rate was defined as in equation (8).

Thirty software tasks observation were collected, partial of them were shown in table 1.

| TABLE I. Partial of Single Programmer Task Allocation with all Variables |
|------------------|------------------|------------------|------------------|------------------|------------------|
| \(x_{1,1}\) | \(x_{2,1}\) | \(x_{3,1}\) | \(x_{4,1}\) | \(x_{5,1}\) | \(x_{6,1}\) | \(x_{6,2}\) | \(x_{6,3}\) | \(y\) |
| 80 | 80 | 80 | 80 | 80 | 80 | 2 | 2 | 5 |
| 82 | 81 | 88 | 79 | 83 | 3 | 4 | 1 | 5 |
| 78 | 77 | 81 | 86 | 91 | 1 | 1 | 5 | 5 |
| 82 | 81 | 77 | 88 | 81 | 2 | 2 | 4 | 5 |
| 77 | 75 | 80 | 85 | 85 | 4 | 4 | 4 | 5 |
| 80 | 82 | 81 | 75 | 78 | 5 | 1 | 1 | 10 |
| 90 | 95 | 85 | 88 | 80 | 3 | 2 | 1 | 10 |
| 79 | 85 | 91 | 85 | 79 | 4 | 2 | 5 | 10 |

- **Second data collection.**
  This data were collected from a finished software projects that have assigned two programmers to work together. All variables, competencies and personal type, were collected together with “Error rate” \((Y)\) variable.
  Thirty software tasks observation were collected, partial of them were shown in table 2.

| TABLE II. Partial of Single Programmer Task Allocation with all Variables |
|------------------|------------------|------------------|------------------|------------------|------------------|
| \(x_{1,2}\) | \(x_{2,2}\) | \(x_{3,2}\) | \(x_{4,2}\) | \(x_{5,2}\) | \(x_{6,2}\) | \(x_{6,3}\) | \(y\) |
| 90 | 90 | 90 | 90 | 90 | 90 | 5 | 5 | 5 |
| 82 | 81 | 88 | 79 | 83 | 3 | 4 | 1 | 5 |
| 78 | 77 | 81 | 86 | 91 | 1 | 1 | 5 | 5 |
| 82 | 81 | 77 | 88 | 81 | 2 | 2 | 4 | 5 |
| 77 | 75 | 80 | 85 | 85 | 4 | 4 | 4 | 5 |
| 80 | 82 | 81 | 75 | 78 | 5 | 1 | 1 | 10 |
| 90 | 95 | 85 | 88 | 80 | 3 | 2 | 1 | 10 |
| 79 | 85 | 91 | 85 | 79 | 4 | 2 | 5 | 10 |

The personal type variables were explained in table 3. For example, variable \(x_{12,6,1}\) is calculated from \(x_{1,6,1} + x_{2,6,1}/2\). Variable \(xp\) is an expected weight of PO which is subjected to software project manager’s thought. Variable \(xg\) is total mark which calculated from summation of all subraction of each personal type with expected weighting of project as shown in (7).

\[
x_{G_{-6_{,12}}} = [x_{12,6,1} \cdot XG_{-6,12}] + [x_{12,6,2} \cdot XP_{-6,2}] + [x_{12,6,3} \cdot XP_{-6,3}] + 15
\]

### B. Individual programmer Regression analysis in “Error rate”.

The first equation to predict dependence variable error rate \((Y)\) was calculated from 30 observations with \(x_{1,1}, x_{2,1}, x_{3,1}, x_{4,1}, x_{5,1}\) of 30 observations whereas he work alone. Best fitted regression equation was in a linear form as equation (8).

\[
\hat{Y}_1 = 41.606 - 0.864.X4 + 0.374.X5 \cdot R^2 = 0.360
\]

When two programmers work together equation (8) must be prepared average value of significance independence variables as equation (9).

\[
\hat{Y}_2 = 41.606 - 0.864.(X1_4 + X2_4)/2 + 0.374.(X1_5 + X2_5)/2
\]
C. Pair wise programmer Regression analysis in “Error rate”

When two programmers work together in assigned task, regression equation (10) was calculated from independence variable \( x_{g_6} \) and dependence variable error rate \( (Y) \).

\[
\hat{Y}_2 = 27.004 - 1.078 \times x_{g_6}, \quad R^2 = 0.501
\]  

D. Pair wise programmer Generalized additive model in “error rate”.

Equation (9) and (10) were presented in difference independence variables but both were predict error rate therefore GAM was derived by combine them together to be a one GAM regression as equation (11).

\[
\hat{Y}_{\text{Gam}} = -0.746 + 0.472Y_1 + 0.836Y_2, \quad R^2 = 0.616
\]  

E. Evaluation

Ten finished software projects were used to perform cross validation of (11) by MMRE. MMRE value is 0.213 of error (or about 21.30%). Therefore (11) could predict error rate of software development correctly about 78.70%.

II. SUMMARY AND SUGGESTION

Independence variables used in this research were chosen by three software project leaders of Thailand software house. Experiment software project were in field business client server architecture application.

According to restriction of research budget year, thirty sample software project were collected from past – finished software projects which were developed under pairwise programming technique.

For model usage, software project manager could use this model (eq. 11) to support his decision by choosing two programmers then fill in their competency in equation 9 and 10 in order to calculate for \( \hat{Y}_1 \) and \( \hat{Y}_2 \).

After that, \( \hat{Y}_{\text{Gam}} \) could be easily obtained. There are many permuted pairs. The pair of programmer which gave least value \( \hat{Y}_{\text{Gam}} \) (least error rate) was then easily chosen.

REFERENCES


