AN ESTIMATIVE MODEL OF THE POINTED DEFECTS RATE IN SOFTWARE PRE-REVIEW FOR NOVICE ENGINEERS IN CHINESE OFFSHORE COMPANY

Zuoqi Wang, Yixiao Qu
Jinan Ryouka Science & Technology Co., Ltd.
Pioneer Square, Qilusoft Park, Jinan Hi-tech Zone, No.1 Shunhua Road, Shandong 250101, Republic of China

Masanori Akiyoshi, Norihisa Komoda
Department of Multimedia Engineering, Graduate School of Information Science and Technology
Osaka University, Yamada-oka, 2-1, Suita, Osaka, 565-0871, Japan

Keywords: Pre-review, bug rate, offshore development, peer review, estimating model.

Abstract: This paper quantitatively discusses the pre-review effectiveness of software developed by novice software engineers in Chinese offshore company. Pre-review process is applied to the software product developed by novice engineers before the normal test process for keeping the quality of product. We extract the factors that influence the number of defects. Then, the collected data of the pointed defects rate and the factors in 27 pre-review are analysed by using the “quantification theory type I” to create a mathematical model for estimating the pointed defects rate. The coefficient of determination R of the obtained estimative model is 0.86. The model provides sufficient accuracy. In the model, “ difficulty of task” is the most effective factor.

1 INTRODUCTION

For the rapid increase of offshore software development (Gold, T., 2005)(Aspray, W., et al. (eds.), 2006), the number of the novice software engineers (SEs) employed by the offshore company is drastically increased in China. For example, the offshore software development company for Japan, which the authors belong to, is planning to employ 120 novice engineers in 2008 fiscal year and 180 novice engineers in 2009 fiscal year. Half of the employees are less than 2 years work experience. Therefore, the offshore company has to assign the novice engineers to the real projects under keeping the quality of products.

One of the schemes for quality guarantee is pre-review applied to the software product developed by novice engineers before the normal test process using test tools. By the pre-review to the novice engineers, man-hour of rework can be reduced, and productivity and quality of software can be improved (Wang, Z., et al., 2008). Moreover, it can make the novice engineers use programming language correctly and improve their quality consciousness. However, it is difficult to judge whether the pre-review is appropriately performed or not.

To judge the appropriateness, the model, which estimates the standard pointed defects rate under the various situations, is necessary. Statistic techniques such as the regression analysis (Gelman, A. and Hill J., 2006) are frequently used for similar estimation purpose. However, the influence factors in the pointed defects rate estimation problem are not only the quantitative factors such as experiment year but also the qualitative factors such as “skill level” and “difficulty of task”. Furthermore, the quantitative affect factors are ambiguous. Therefore, it is difficult to apply the traditional regression analysis or similar methods to the pointed defects rate estimation problem.

In this paper, to cope with this problem, we propose a quantitative estimating method of the pointed defects rate at pre-review process. The estimating model is developed by using “the

Wang Z., Qu Y., Akiyoshi M. and Komoda N. (2008).
AN ESTIMATIVE MODEL OF THE POINTED DEFECTS RATE IN SOFTWARE PRE-REVIEW FOR NOVICE ENGINEERS IN CHINESE OFFSHORE COMPANY.
In *Proceedings of the Third International Conference on Software and Data Technologies - SE/GSDCA/MUSE*, pages 228-233
DOI: 10.5220/0001899002280233

quantification theory type I"(Hayashi, C., 1952.) based on the two year real data collected in the authors' company, Jinan Ryouka Science & Technology Co., Ltd.(RYOUKA).

2 FACTORS INFLUENCING THE POINTED DEFECTS RATE

2.1 Pre-review Process

In RYOUKA, after three-month introductive education, the novice engineers are assigned to development teams. In each team, they take charge of manufacturing processes (coding and unit test specifications making) mainly. For the program developed by a novice engineer (entering the company less than one year), an expert engineer carries out pre-review before a normal unit test process so that it is necessary to bring them up by a real development project. By the pre-review, pointing the defects before unit test process, man-hour of rework can be reduced, and productivity and quality of software can be improved. Moreover, the novice engineers can use programming language correctly and improve their quality consciousness while they carry out real software development.

In the similar way of the peer review (Humphrey, W.S., 2005)(Nonaka, M., 2004), experienced engineers in a same project team entirely review the results (sources code and unit test specifications (test cases)) of novice engineers. The reviewers fill review scale, review time, review-discovery number, review item (it classifies into high, middle, and low according to the importance) into review sheets.

Using CBR (Checklist-Based Reading) and TCBR (Test Case Based Reading) which were used as review technique well by downstream process, a project leader and sub-leaders may add check items peculiar to a project suitably based on the standard check list (an inspection standard of source code and a unit test specifications inspecting standard) as shown in Table 1. If there is a shortage in the way of reviewing, the reviewers can add them into the checklist at any time. The review is based on detailed specifications. It is checked whether test cases written in the unit test specifications are proper and enough. A source code review is performed in whether there are both a grammar check and a simple mistake, whether the program conforms to the coding standard, and whether business logics are correctly installed. The review time of a source code takes about twice of unit test specifications. Practically, the review speed of a source code is about 0.6 hour/Ksteps. For the purpose of novice engineers' education, the review speed is kept slower than the normal.

Table 1: Review Check List (a part).

The whole	Make a coding standard and whether the coding is according to it?
	Are the standard checklists used?
	Is it considered as a structured programming?
	Have all the functions that described in DS (Design Specification) been confirmed? (Does C0/C1 measure become 100 %?)
	Has a comparison been taken between source version and completed version?
	Do you set an appropriate input condition, concrete confirmation content for a check item?
Processing	Do you confirm input number about zero case, one case, or n cases (n>3)?
	When you used the following for a judgment condition, divide it into three conditions of `<', `=', and `>'. Do you confirm it?
	In the case of an OR condition, do you confirm it about all conditions?
	Do you confirm contents that set to the interface every item?
	Do you confirm it about an initial value, the initialisation contents of the item?
	Do the contents of various messages have been confirmed?
	For each item, zero, minus, biggest have been confirmed?
	In the case of an output position / editing contents confirmation, assume layouts attached document and do confirm it every item?

When the review was over, the pointed defects are corrected. The reviewers confirm whether they are correctly revised. If there are many problems remained, the review process is gone over.

In addition, when a project was completed, the review situation about novice engineers is summarized in a completion report and is submitted to the company knowledge system (Cai, L., et al., 2007) so that all employees can share knowledge in the company. Novice engineers summarize quality problems and technical points, and present them at the project reflection meeting.

2.2 Candidate of Influence Factors

In order to evaluate the appropriateness of the pre-review process, estimating the pointed defects rate is considered. Estimating the pointed defects rate requires determining the factors that have an influence on the pointed defects rate. From our experience, there are several possible factors such as skill level, difficulty of task, year of experience, and change of language.

(1) Skill Level. When company's in-house education is over, the educators rank the novice engineers into three skill levels of A (high), B (middle), and C (low). Evaluation factors are the university score, the Japanese-language ability, development languages skill, the understanding ability of the Japan-oriented project development, and the development technique. The engineers of the skill level "A" have passed not only "JLPT (Japanese Language Proficiency Test) level 2" but also FE (Fundamental Information Technology Engineer Examination). The engineers of the skill level "B" have passed "JLPT level 3" and their in-house education results are excellent levels. And, the engineers of skill level "C" are the persons whose Japanese-language level is less than "JLPT level 3". In offshore software development, ability of Japanese language is as important as programming skill. For the record, the distribution of 56 novice engineers in the past three years, there were 3 persons in skill level "A". 31 persons are in skill level "B", and 22 persons are in skill level "C".

(2) Difficulty of Task. The "difficulty" is a difficult degree divided by A (many update and retrieval processing are included), B (many update processing are included), and C (few update processing is included). SEs assigned to difficult tasks are considered to make mistakes compared with easy projects.

(3) Years of Experience. "years of experience" is the period after starting the development as at the project started. Longer experienced SEs are considered to write more correct program.

(4) Change of Language. Programming language which SEs use may be changed during SEs' carrier. In the case, the value of development experience may decline. Therefore, "change of language" is a candidate of the influential factors.

3 SAMPLES OF REAL CASES

We checked out 27 novice engineers' review sheets in recent years. Table 2 lists the results of the influential factors in each case. The unit of the pointed defects rate is defects/Ksteps. All tasks are on the manufacturing phase. Programming languages are ".NET framework", C and Visual Basic. The average period of projects is about six months. Target projects are a human salary information system and a credit card system. In the data, there are no cases of "skill level of A". Since the projects contents were relatively difficult, the number of bugs was more than other projects.

Table 2: Collected data.

(unit: Experience: year,
Pointed defects rate: defects/Ksteps)

Case No	Skill level	Difficulty	Experience	Language change	Pointed defects rate
1	C	B	0.58	Change	4.09
2	C	C	0.42	Change	1.32
3	B	B	0.92	Same	3.29
4	B	C	0.75	Same	1.13
5	C	B	0.08	Same	0.86
6	C	C	0.08	Same	1.36
7	C	B	0.08	Same	3.82
8	C	B	0.08	Same	2.00
9	B	B	0.67	Same	5.33
10	C	A	0.67	Same	18.21
11	C	B	0.67	Same	2.14
12	B	B	0.67	Same	3.62
13	B	B	0.67	Same	2.63
14	C	A	0.67	Change	9.43
15	C	A	0.67	Change	8.68
16	B	C	0.50	Same	1.13
17	B	C	0.50	Change	1.67
18	C	C	0.50	Change	3.46
19	C	C	0.50	Change	2.90
20	B	B	0.25	Change	2.50
21	B	B	0.42	Same	3.75
22	C	B	0.50	Same	5.69
23	A	B	0.92	Change	8.00
24	A	B	0.25	Same	6.79
25	A	B	1.08	Same	5.67
26	C	B	0.25	Same	6.67
27	C	B	0.25	Same	6.00

4 ANALYSIS USING "QUANTIFICATION THEORY TYPE I"

4.1 How to Estimate the Pointed Defects Rate

As discussed in Section 2.2, the influence factors are mixed data of qualitative and quantitative. The "quantification theory type I" (Hayashi, C., 1952.) is an efficient method for analysing multivariate data for which variables are qualitative data. We use the “quantification theory type I” to estimate the pointed defects rate. We associate the terms for the “quantification theory type I” with the influence factors as follows.

We define the four influence factors of the pointed defects rate (those are, skill level, difficulty of project, year of experience, and change of language) as items. Here, an item of “quantification theory type I” is called FACTOR.

The definition of FACTOR (=item) j is:

j = 1: skill level
j = 2: difficulty of task
j = 3: year of experience
j = 4: change of language

Category k for FACTOR j is:

j=1: (k=1: C, k=2: B, k=3: A)
j=2: (k=1: C, k=2: B, k=3: A)
j=3: (k=1: <0.2, k=2: “>0.2 & <0.4”,
k=3: “>0.4 & <0.6”, k=4: “>0.6 & <0.8”,
k=5: “>0.8”)
j=4: (k=1: change, k=2: same)

For the model of the “quantification theory type I”, if the number of categories for FACTOR j is K_j, then the estimator i of the pointed defects rate (y_i) for case i is given as follows.

$$y_i = \sum_{j=1}^{4} \sum_{k=1}^{K_j} b_{jk} \delta_i(jk) \qquad (1)$$

Here, δ_i (jk) is defined as follows:

$$\delta_i(jk) = \begin{cases} 1 & \text{Item } j \text{ of case } i \text{ belongs to category } k \\ 0 & \text{Otherwise} \end{cases}$$

b_{jk} is called a category score: the quantity given to category k of FACTOR j. For “quantification theory type I”, we determine the value of b_{jk} that minimizes the sum of squares of the deviation of y_i from.

$$\sum_{i=1}^{n} (y_i - \sum_{j=1}^{4} \sum_{k=1}^{K_i} b_{jk} \delta_i(jk))^2 \rightarrow \text{Minimum}$$

n indicates the total number of cases. Determining b_{jk} involves solving M simultaneous equations, where M is equal to:

$$(\sum_{j=1}^{4} K_j + 1 - (\text{number of items} - 1))$$

Here, at least 11 cases are required.

4.2 Analysing the Pointed Defects Rate using Quantification Theory

Table 3 shows the results of analysing the evaluation data in Table 2 by the “quantification theory type I” for the 4 influence.

Table 3: Analysis of pointed defects rate by “quantification theory type I”.

FACTOR	Category	Category score	Range	Partial correlation coefficient
skill level	1. C	1.223	2.649	0.237
	2. B	- 0.426		
	3. A	0.668		
difficulty of task	1. C	-2.732	12.210	0.916
	2. B	-0.256		
	3. A	9.478		
year of experience	1. <0.2	- 3.046	5.399	0.734
	2. >0.2 & <0.4	0.435		
	3. >0.4 & <0.6	2.353		
	4. >0.6 & <0.8	-0.989		
	5. >0.8	-1.836		
change of language	1. change	-2.184	3.469	0.699
	2. same	1.285		
Constant term		4.524		

coefficient of determination: 0.861, multiple correlation coefficient: 0.928, standard error: 1.68

The estimation expression of the pointed defects rate y is denoted as follows:

$$\begin{aligned} y = {} & 1.22\, x_{11} - 0.43\, x_{12} + 0.67\, x_{13} - 2.73\, x_{21} \\ & - 0.26\, x_{22} + 9.48\, x_{23} - 3.05\, x_{31} + 0.44\, x_{32} \\ & + 2.35\, x_{33} - 0.99\, x_{34} - 1.84\, x_{35} - 2.18\, x_{41} \\ & + 1.29\, x_{42} + 4.52 \end{aligned} \qquad (2)$$

Here, x_{jk} is defined as follows.

$$x_{jk} = \begin{cases} 1 & \text{Item } j \text{ of the case belongs to category } k \\ 0 & \text{Otherwise} \end{cases}$$

The range R_j as a scale to measure the contribution rate of each influence factor or item (=FACTOR) to the pointed defects rate is calculated as the difference between maximum and minimum category scores.

Table 4 shows the residual between the estimated results and the measured pointed defects rates.

Table 4: Residual of estimated and measured pointed defects rate.

(unit: defects/Ksteps)

Case No	Pointed defects rate	Estimate	Residual
1	4.09	4.56	-0.47
2	1.32	2.08	-0.76
3	3.29	3.29	0.00
4	1.13	1.66	-0.53
5	0.86	2.63	-1.77
6	1.36	0.15	1.21
7	3.82	2.63	1.19
8	2.00	2.63	-0.63
9	5.33	4.14	1.20
10	18.21	14.14	3.79
11	2.14	4.69	-2.55
12	3.62	4.14	-0.52
13	2.63	0.67	1.96
14	9.43	10.95	-1.52
15	8.68	10.95	-2.27
16	1.13	1.53	-0.40
17	1.67	1.53	0.14
18	3.46	2.08	1.38
19	2.90	2.08	0.82
20	2.50	3.09	-0.59
21	3.75	5.00	-1.25
22	5.69	4.56	1.13
23	8.00	8.57	-0.57
24	6.79	6.66	0.13
25	5.67	5.23	0.44
26	6.67	6.11	0.56
27	6.00	6.11	-0.11

4.3 Consideration of Estimation Model

For the estimation model of Expression (2), the multiple correlation coefficient R is 0.928, and the contribution rate (R^2), the square of R, is 86%. It is generally said that a practical model must have a multiple correlation coefficient of at least 0.85; therefore, the estimation model of Expression (2) provides sufficient accuracy.

As shown in Table 3, "difficulty of task", which has the maximum range and partial correlation coefficient, has a most affect the pointed defects rate. The category scores of "difficulty of task" increase in the degree of difficulty. This means that the novice SEs assigned easy tasks make few mistakes and SEs assigned difficult tasks make lots mistakes. It is a common-sense result.

The second effective factor is the "year of experience." The category scores of "year of experience" roughly decrease year by year. This means that inexperienced SEs make few mistakes and richly experienced SEs make lots of mistakes. However, the category score of the most inexperienced SEs (less than 0.2 experience) is the smallest. It means the most inexperienced SEs generated most correct programs. The reason of this counterintuitive result is considered that SEs less than 0.2 year experience are still involved in in-house education and are assigned to the quite easy tasks such as extension of the programs generated by skilled SE's.

The "change of language" and the "skill level" is little influence to the pointed defects rate. The category scores of the "change of language" are strange. The SEs changed the programming language make less mistakes. And, the level "B" persons make lots of mistakes compared with the level "A" persons. These out-of-line results may be considered as sample noise. To clarify the accurate effect of these two factors, it is necessary to gather more real data.

Finally, we argue the cases whose residual are relatively large. The case 10, the largest residual in the cases, is the case that since only one novice SE was involved in the project, the review was done very carefully. Therefore, so many defects were pointed. In contrast, on the case 11, the largest negative residual in the cases, the pre-review was done with corner cutting, because a novice SE was engaged in the previous phase development of the same project. So, less pointed defects than estimation was measured.

We consider the proposed model can be used to check whether the pre-review is appropriately performed or not. As evidenced above discussion, less pointed defects than the estimated value means imperfect review activity or simplistic task assignment. On the other hand, more pointed defects than the estimated value means existence of special situation or extremely incapable novice SEs.

5 CONCLUSIONS

By analysing cases of novice software engineers in real projects, we extracted the factors that influence the pointed defects rate at pre-review. And, by using the "quantification theory type I", a sufficiently accurate mathematical model for estimating the pointed defects rate was generated. This model can be used for the management of pre-review process.

As future works, firstly, the validation of the proposed model is listed up. And, by gathering more data and adding the factors such as the characteristic of the projects, the skill level of review persons, programming languages, development environment, and so on, we think that we can improve the estimation model. Another research direction is a support of accurate classification of factors. Imprecise qualitative value depresses the model accuracy in a grand deal.

REFERENCES

Aspray, W., Mayadas, F., and Vardi, M. Y. (eds.), 2006. Globalization and Offshoring of Software, Report of the ACM Job Migration Task Force, Association for Computing Machinery

Cai, L., Wang, Z., Jial, Y., Akiyoshi, M., and Komoda, N. , 2007. A knowledge sharing and managing system for offshore software development company, in Proc. of the 7th WSEAS International Conference on Applied Computer Science (ACS'07), pp. 352-355.

Gelman, A. and Hill J., 2006. Data analysis using regression and multilevel/hierarchical models, Cambridge University Press.

Gold, T., 2005. Outsourcing software development offshore - Making it work, Auerbach Publications.

Hayashi, C., 1952. On the prediction of phenomena from qualitative data from the mathematic-statistical point of view," Annals of the Institute of Statistical Mathematics, 3, pp. 69-98.

Humphrey, W.S., 2005. PSP: a self-inprovement process for software engineers, Pearson Education, Inc.

Nonaka, M., 2004. An experimental comparison of individual review methods for design and source code, 2004-SE-146-(4) IPSJ.

Wang, Z., Cai, L., Qu, Y., Akiyoshi, M., and Komoda N., 2008. Effectiveness evaluation of pre-review of software developed by novice engineers in China, " in Proc. of the Japan-China Joint Symposium on Information Systems, pp.25-29.