How Many Students are Ready to Write Quality Programs Complying with Coding Standards: A Case Study *

WANG Yan-qing[1], SU Hui-dong[2], WANG Jian-zheng[1], WANG Bao[3]
(1. School of Software, Harbin Institute of Technology, Harbin 150005, China;
2. Easycon Electronics Co., Ltd., Beijing 100101, China;
3. Faculty of Computer Science, Heilongjiang East College, Harbin 150086, China)

Abstract: The compliance with coding standards is becoming an evaluating index for a software engineer's teamwork capability or international cooperation capability. The best place to solve this problem should be in an educational environment such as college where students begin to write their introductory programs. After constructing a web-based evaluating system, a case was studied in our school of software and some important conclusions were made such as (1) the evaluating results were not so satisfactory because of the absence of necessary training and (2) students' ability to comply with coding standards did not improve automatically with the assignments one by one without timely feedback. In short, there is a long way to run before the students can write quality code meeting the requirement of software industry.

Key words: coding standards; quality assurance; quantitative evaluation; on-line evaluation
CLC number: TP31 Document code: A Article ID: 0529-6579 (2007) S2-0093-04

1 Introduction

The rapid development of the Internet economics and global integration process are bringing every nation a new challenge - how to raise the international competitive capability of software industry. Over the past decade, with the international competition and the growing popularity of software outsourcing, international and industrial programming capabilities also have the unprecedented requirements for programmers. This was proved by the successful experience of software outsourcing to India[1, 2]. In fact, quite a few graduates from computer science faculties at Chinese universities can not reach the international and industrial requirements in programming. In this situation, how to educate the students to become qualified programmers and to adapt to the software industry demands will become a very important subject of college software education now.

In essence, the software design is rather a project than a personal handicraft[3]. With many people being involved in product development, if the programmers do not comply with unified coding standards, the ultimate integration code will have very low readability. Not only does it create obstacles of understanding code and increase the maintenance workload, but the possible defects in poorly written programs might be considerable. Therefore, in the software development process, strictly complying with coding standards is of great significance[4]. College students should be given high standard training in coding standards and they should accept the guidelines of a series of scientific methods in their introductory programming.

In this paper, the research on coding standards was continued based on[5], and some progress was achieved in many aspects. Utilizing the developed evaluation system and the collected data, a case in our School of Software was studied. Some obvious problems were found through the whole process of case study, which is hoped to be a reference or guidance for other educators' teaching activity.

2 Evaluation Index System

In the software industry, most of the coding standards manuals contain at least 100 standards or guidelines[6-8], which are impossible to apply for university students who are just beginning to learn introductory programming language. Because (1) these students do not understand what the coding standards are, and (2) it is hard to strictly comply with the well-accepted coding standards even for professional programmers, let a-

*收稿日期: 2007-10-28
作者简介: 王延青 (1968年生), 男, 讲师; E-mail: yanqing@hit.edu.cn

bone the students who have just started to learn programming.

From the survey, we found that college students should pay more attention to at least four sections - Layout, Naming, Comment, Coding in today's highly competitive era[5]. Layout includes proper layout, code indentation, the usage of spaces, etc.; Naming requires students to give variables and functions meaningful names; Comment asks students to write sufficient comments to explain the purpose of the programs regardless whether they are large or small; and Coding demands students to properly use the control statements, such as if, switch, do...while and so on.

In summary, a simplified version of coding standards was determined as Appendix 1. These coding standards indicate the basic requirements in the software industry and are easy for college students to understand. Furthermore, these standards can be taken as solid cornerstones for the students to practice more professional coding standards.

With the simplified coding standards, the hierarchical structure of evaluation index system was constructed which consists of three layers: target level, criterion level, and sub-criterion level. After that, by AHP (Analytic Hierarchy Process) together with the questionnaire survey[9], the weight distribution of all evaluating indices was achieved. Eventually, a web-based evaluation system was built up relying on all the work mentioned above.

3 Case Study

So as to find the current status for our students to comply with coding standards, a case study was designed and brought into operation.

3.1 Data collection

All the 159 Year one students were involved in our research. The course Personal Software Process (PSP) was determined to be the research environment. PSP course takes nine weeks, in which teachers require students to write and submit a program with about 100 LOC at a certain interval. Before programming, the teachers or teaching assistants do some necessary analysis about the programming assignments in the class. There are five assignment programs totally, so there should be $159 \times 5 = 795$ programs theoretically.

However, although the teacher announced the result of these programs would be a part of the final scores of PSP, there were still a few students who did not complete all five programs. By analyzing all the submitted programs, it was concluded that the main reason was due to low programming capability of minority students. As for some students who did the repeated submission of one program, the teaching assistants should find time to check and pick up the latest ones. Through the above process, 675 available samples were collected eventually.

After that, the developed evaluating system was applied to check each assignment (source program files) written by each student and all the evaluation results were stored into a MySQL database.

3.2 Analysis of mark distribution

With the data in database, the mark distribution was calculated and summarized. The Fig. 1 to Fig. 5 denotes the mark distribution of students' assignments from 1 to 5. In the figures, the height of blue bars means the count of students who got the corresponding marks. For example, in Fig. 1, the highest bar indicates there were 84 students who turned in mark from 60 to 69 (in the form of "60—").

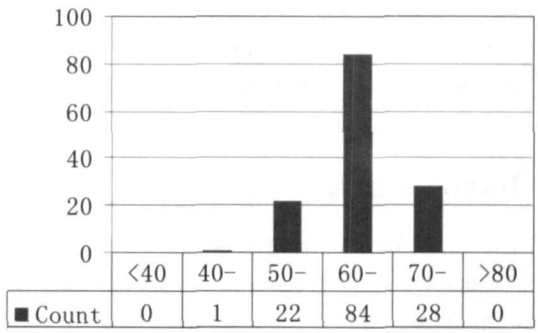

Fig. 1 Mark distribution of Assignment 1

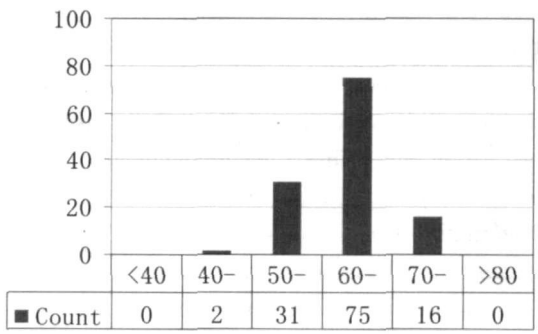

Fig. 2 Mark distribution of Assignment 2

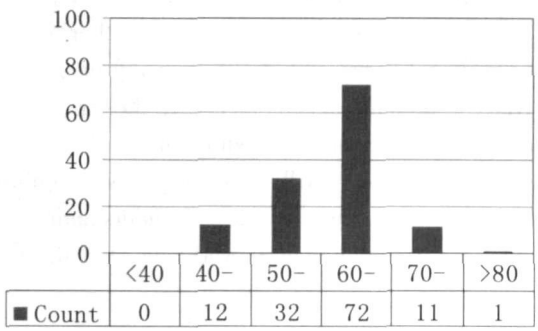

Fig. 3 Mark distribution of Assignment 3

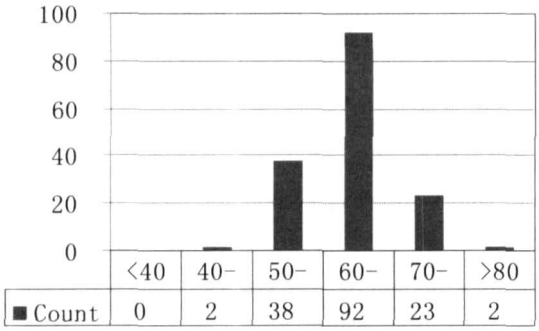

Fig. 4　Mark distribution of Assignment 4

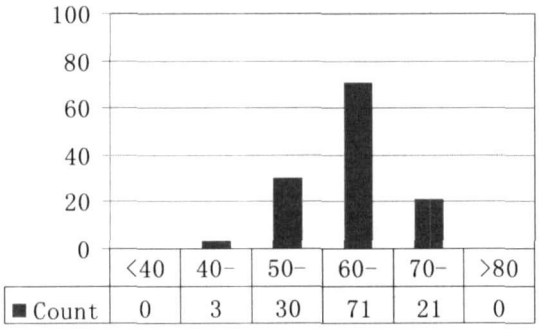

Fig. 5　Mark distribution of Assignment 5

From the mark distribution, it can be found that the evaluating result of each assignment is a normal distribution. It proves to some extent that the evaluation is valid. However, the evaluation results are not so satisfactory. Although only a few students got marks less than fifty, the number of the students who achieved marks greater than eighty is likewise small. The marks of the most students are standing in the range of 60 to 70.

3.3　Problems finding

Through analyzing the score of all students in the database, three major problems as following were found.

(1) Because of the delay with the evaluation and feedback, the average scores of the five assignments from 1 to 5 have no exciting difference (see Fig. 6). Before the case study, it was predicted that there was little probability for the extent of compliance with coding standard by students to increase automatically in the entire nine weeks.

(2) Because of no consistent training and timely feedback, the average scores of the four sections are comparatively low (see Figure 7). This fact has also been estimated before the case study.

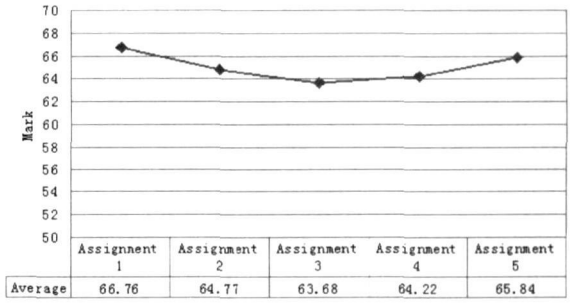

Fig. 6　Average marks distribution of 5 assignments

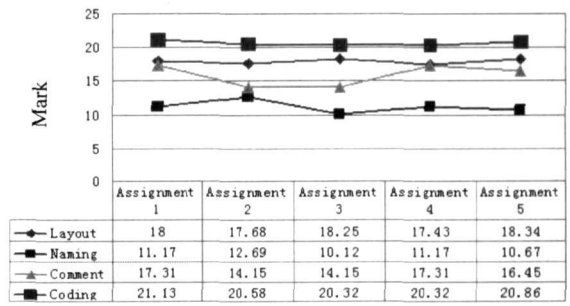

Fig. 7　Average marks of each section

(3) The analysis of the final average scores of four sections shows that the sorting result is Coding, Layout, Comment, Naming in descending order (as shown in Fig. 7). That means the future training emphasis should be focused on Naming and Comment so as to improve the extent to which programmers comply with coding standards.

4　Contrast Experiment

In order to further discover the difference between those students and senior programmers on complying with coding standards, a comparative analysis was also implemented. Specifically referring to the three randomly selected examples in textbook [10] p10.cpp[10]: 193 ~ 196, p7.cpp[10]: 106 ~ 107 and p6.cpp[10]: 88 ~ 89, the same evaluating system was used to do the same test, the results are shown in Table 1.

Tab. 1　Evaluating score for textbook examples

	Total	Layout	Naming	Comment	Coding
p10.cpp	88.80	17.94	26.38	21.18	23.30
p7.cpp	91.06	20.30	25.12	22.34	23.30
p6.cpp	92.04	21.68	24.59	23.31	22.46

Table 1 shows that (1) the example programs written by senior programmers or experienced teachers got higher evaluation scores, which will become the next teaching goal for us to cultivate our students; (2)

Even being taken as examples, the three programs are still not perfect. That shows that the problem with not complying with coding standards is even prevalent among the experienced teachers and excellent software engineers.

5 Summary

It has an extremely practical value to research how students recognize coding standards and how many of them are ready to write quality programs complying with coding standards. The qualitative measurement for these educational results will become an effective way to measure and evaluate teaching quality. Without this measure, it is difficult for teachers to know whether their methods are scientific and effective or not. Under such circumstances, the paper launched a research method on evaluate our Year one students' readiness for writing quality programs.

Through the case study and analysis, the following conclusions were drawn. Even though a beginner programmer has known the importance of coding standards, the majority of students find it difficult to apply coding standards in programming without systematic training and guidance. If the students do not receive consistent training and timely feedback, the extent of students complying with coding standards will not automatically increase. As a result, to college students, the education and training on compliance with coding standards is difficult and should not be skipped.

As much as the conclusions were drawn above, there are still some topics to be explored.

(1) To enable students to fully understand and grasp coding standards, an effective teaching method should be carried out with systematic and continual training to students.

(2) An open web-based evaluation system should be developed, with which the students can upload their programs to the website and get detailed benchmarking results. With the timely evaluation and feedback, students can be motivated to observe the coding standards and ultimately retain an ideal learning curve (evaluating scores keeps going up).

(3) By monitoring the students' scores, teachers can assist lower scoring students separately.

(4) The evaluation index system of coding standards suitable for high grade students and software engineers needs to be constructed, while the training scheme and evaluating system for them should be prepared in advance.

References:

[1] Kobitzsch W., Rombach D., Feldmann R. L. Outsourcing in India[J]. Software, 2001, (2): 78—86.

[2] Burger Werner. Offshoring and Outsourcing to INDIA, Second[J]. IEEE International Conference on Global Software Engineering (ICGSE 2007), 27—30 Aug. 2007: 173—176.

[3] Steve M. Code Complete 2[M]. Beijing: Electronics Industry Publishing House, 2006: 9—23, 729—768.

[4] Fang Xue-fen. Using a coding standard to improve program quality[C]. Proceedings of Second Asia—Pacific Conference on Quality Software, 2001: 73—78.

[5] WANG Yan-qing, WANG Jian-zheng, SUI Xiang-nan, MA Pei-jun. Quantitative Research on How Much Students Comply with Coding Standard in Their Programming Practices[C]. Proceedings of the 3rd China Europe International Symposium on Software Industry Oriented Education (CEIS-SIOE 2007), Dublin, Ireland, Feb. 6—7, 2007: 116—119.

[6] Herb S., Andrei A. C++ Coding Standards 101 Rules, Guidelines, and Best Practices[M]. Pearson Education Asia Ltd., 2006.

[7] Frank V. <http://corelinux.sourceforge.net/cppstnd/cppstnd.html>[EB]. Date Access: December 20 2006.

[8] Carl R. D. Code-Style Prescriptions, 1991, 9(9): 41—45.

[9] WANG Yan-qing, WANG Jian-zheng, ZHANG Li-jie, WANG Bao. A Framework for Quantitative Evaluation of Coding Standards in Programming Language Teaching[C]. The Proceedings of the 2nd Symposium on the Teaching of Computer Programming Courses, Hefei, China, Oct. 2007: 29—31. (In Chinese)

[10] Paul Kelly. A Guide to C++ Programming[M]. Gill & Macmillan, 2001: 88—196.