

# Complying with Coding Standards or Retaining Programming Style: A Quality Outlook at Source Code Level

Yanqing Wang, Bo Zheng, Hujie Huang

School of Software, Harbin Institute of Technology, Harbin 150001, P. R. China  
Email: {yanqing, hjhuang}@hit.edu.cn

Received October 22<sup>nd</sup>, 2008; revised November 10<sup>th</sup>, 2008; accepted November 21<sup>st</sup>, 2008.

## ABSTRACT

In order to make most software engineers and managers pay more attention to software quality at source level, two confusing terms—coding standard and programming style—were reviewed and compared. An evolutionary model of quality assurance at source code level was proposed, which implies that coding standard should be better accepted and more emphasized than programming style. Our current researches on evaluating the compliance with coding standards will likely make the strategy of quality assurance at source code level more operable.

**Keywords:** Coding Standards, Programming Style, Source Code, Quality Assurance (QA), Evaluation Index System

## 1. Introduction

How to raise the international competitive capability of software industry and how to improve software quality are new challenges for software companies and software engineers. The research on *coding standard* and *programming style* presents a quality notion of software engineering at source code level, while writing high quality code is a required skill to a software engineer. In this situation, how to make most software engineers and college students to pay more attention to quality at source level became an important subject in software industry and software engineering education field. In fact, complying with coding standards when programmers are writing computer programs is beyond the readability issue, it actually is an issue of quality assurance at source level. Since about 67% of workload in software lifecycle is at maintenance stage [1], coding standards can indirectly determine the quality of a whole project (see Figure 1).

Publications on this topic often focus on a set of rules on coding standards [2], taxonomy [3] and paradigm of programming style [4], and the approaches to teach coding standards [5]. These researchers upgraded the *coding standard* and *programming style* to high position at source code level. However, *coding standard* and *programming style* are not distinguished clearly in the above researches. Many software practitioners are always taking *coding standard* as the synonym of *programming style* and often use them alternatively. So it is very necessary to review, distinguish and compare these two confusing terms at first.

In this paper, the understanding of *coding standard* and

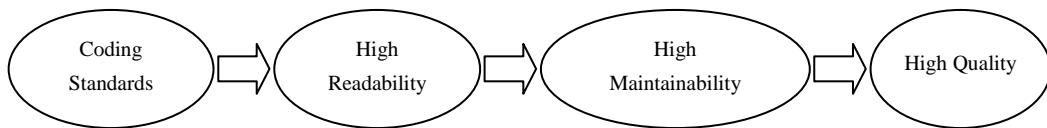
*programming style* was presented in Section 2 by analyzing their difference and relationship. In Section 3, an evolutionary model of quality assurance at source code level was constructed and put forward. The increasing importance of coding standards was emphasized in Section 4. In Section 5, our practices supporting the coding standards strategy were briefly introduced.

## 2. Understanding of Coding Standard and Programming Style

### 2.1 Explanation

(1) *Programming style*. It is also named *code style* or *coding style*. As we know, *programming style* is an intuitive and elusive concept that shows the style of writing code. It's highly individual and easily recognized, yet difficult to define or quantify. The goal of *programming style* is to make a program clear, easy to be understood, and thereby easy to work with [4], but the extreme personal *programming style* which is different from other team members' often degrades the readability of source code.

(2) *Coding standard*. Different from *programming style*, coding standards are a set of industry-recognized best practices that provide a variety of guidelines for developing software code. There is evidence to suggest that compliance with coding standards in software development can enhance team communication, reduce program errors and improve code quality [6]. The coding standards are not only assuring the readability, but also



**Figure 1. The relationship between coding standard and software quality**

creating a helpful circumstance of teamwork. Working under a set of coding standards will make team members to comprehend colleague's programs much more easily and disabuse the injection of unworthy defects.

## 2.2 Difference

*Programming style* is not an equivalent of *coding standard*. Software engineers should understand their difference so that they can really control the quality of the programs they write.

(1) Different times. At early stage of software industry, software size was small, so one programmer can finish a whole project independently. The style of writing programs by individual programmer was called *programming style*. The programs at that time were more like crafts than engineering products, while the programmers were more like artists than engineers. With the rapid development of global IT industry, it was almost impossible for a single programmer to finish a whole software product. The production of software became the activity of teams, organizations or professional software companies. In software companies, various *programming styles* were combed and summarized to build up a set of rules—*coding standards*—to control the maintenance cost and software quality as well.

(2) Different measurability. The compliance with *coding standards* can be measured and evaluated but the *programming style* cannot be. Anyone who has different *programming style* from others can announce that he creates a new *programming style*. The *elegance* or *style* of a program is sometimes considered a nebulous attribute that is somehow unquantifiable; a programmer has an instinctive feel for a *good* or *bad* program [3]. It was believed that *programming style* is a multi-faceted concept that is not captured by a collection of rules or by a single style score [4]. It means that a program cannot be judged *good* or *bad* only by *programming style* or everyone can evaluate a program's style while he or she likes. On the contrast, the coding standards should be industry-recognized and almost software engineers recognized it [5]. Whether programmers complying with coding standards or not can be judged by human or by software programs.

(3) Different objective. The *programming style* focuses on more personal habits than readability. Some programmers retain a certain *programming style* because they are happy to do so. Different programmers may retain different *programming styles*. While coding standards emphasize readability and it prefers teamwork to individual. Writing program with coding standards improves the appearance of source code. Coding standards serves the teams and companies that care about the software quality at source code level.

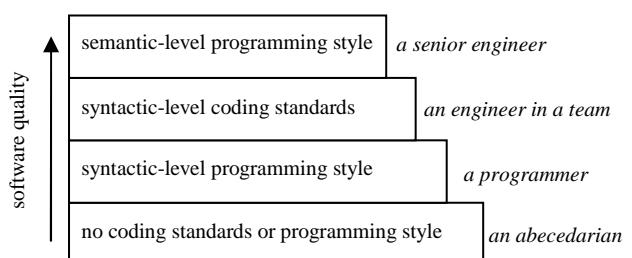
## 2.3 Relationship

*Coding standards* are the evolution of *programming styles*. The *programming style* is an individual concept. As a *programming style* become popular and has been well accepted by many teams, companies, and even software industry, it will be upgraded to *coding standards*. With the development of software industry and outsourcing, the notion of *coding standards* has been taken on by more and more software enterprises. Over the past decades, 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 [7]. As a result, coding standards usually need to be implemented through a formal process in industry [8] to assure the quality of source code.

Meanwhile, some scholars and engineers use these two terms alternatively. In [9], Kernighan and Plauger defined *programming style* as a set of rules or guidelines used when writing the source code for a computer program. They said, "It is often claimed that following a particular programming style will help programmers quickly read and understand source code conforming to the style as well as helping avoid introducing faults." From the above discussions in this paper, it is no doubt that the *programming style* in [9] is equivalent to *coding standard*.

## 3. Evolutionary Model of QA at Source Level

The software quality assurance contains five attributes: functionality, reliability, usability, performance, supportability [1]. Every attribute relates with source code. The quality of source code affects the quality of the software in large measure. To enhance the quality assurance at source level, an evolutionary model was constructed from the viewpoint of *coding standard* and *programming style* (see Figure 2).



**Figure 2. Evolutionary model of coding standard and programming style**

Figure 2 embraces the process from lowest level *no coding standard or programming style* to the highest level *semantic-level programming style*. It also indicates the programmers who have the ability at corresponding level. The evolution is not only the time period concept but also a skill enhancement issue. Software quality will increase while its writer becomes a senior engineer from an abecedarian.

#### (1) Stage 1: no coding standards or programming style

At the very beginning of computer programming age, there was no *coding standards* or *programming style*. Programmers wrote programs as they wished. So did the students in colleges. The source code at this level was always confusing and had low readability. The software quality at this stage was hard to control.

#### (2) Stage 2: syntactic-level programming style

With the development of software engineering, the *programming style* was used to publicize the named good habits [9]. And the *syntactic-level programming style* had been used to assure the readability or efficiency of programs. The *programming style* helped the beginners become programmers. But as it stands, the *programming style* is too individual to be widely used.

#### (3) Stage 3: syntactic-level coding standards

With the industrialization of software development, piles of *programming styles* had to be summarized and upgraded to assure the team working more successful. Then the *syntactic-level coding standards* were implemented in some software companies such as IBM and Sun. The *syntactic-level coding standards* are the *coding standards* we defined above.

#### (4) Stage 4: semantic-level programming style

Moreover, while complying with coding standards, some experienced software engineers might write their programs in their own ways, e.g. inputting from or outputting to files, handling the connection with database and controlling the virtual memory etc., to make programs more efficient, reliable and portable. This level of *programming style* is defined as *semantic-level programming style*, which is based on syntactic-level coding standards and beyond them.

## 4. Role of Coding Standards in Software Engineering

### 4.1 Coding Standards Help Assure the Quality of Software

*Coding standards* can assure the software quality at source code level, and will release testers' workload. A defect in source code level would cost about 4 times money to remove when it remains at testing stage [1]. The coding standards can help the white-box tester read the program easily and save the testing time. The less time and more efficient, the higher quality the software would be. When more engineers concern source code quality and comply with coding standards, the quality of the softwares they produce would be assured.

## 4.2 Advices on Quality Assurance at Source Code Level

When the student starts to learn a new programming language or software engineers does their daily coding work, there are some advices based on the evolutionary model shown as above.

(1) If you are a student or novice software engineer, comply with *coding standards* and not retain *programming style* at syntactic level. Complying with *coding standards* is a basic skill of software engineer, especially working in a big company or developing an outsourcing project.

(2) If you are a teacher of a preliminary programming language, ask students to obey *coding standards*, educate tomorrow's qualified engineers at today's colleges.

(3) At the basis of complying the *coding standards*, develop your own *semantic-level programming style* when you are adequately experienced and you like. That is the highest realm of a programmer.

## 5. Conclusions and Future Work

*Coding standard* and *programming style* is a pair of terms at source code level. With the globalization of software cooperation and popularity of software outsourcing, the quality assurance at source level remains a pressing concern. Quality assurance at source level is much more economical and scientific than testing even though it is hard to operate. In this paper, *coding standard* and *programming style* were compared and an evolutionary model base on their relationship was proposed. *Coding standards* were more recommended than *programming style*. If we would like to retain our *programming styles*, the high level *programming style* beyond *coding standards* was strongly recommended.

To better support our ideas about complying with *coding standards*, we made some preliminary researches such as (1) we proposed an AHP-based evaluation index system on complying with coding standards [10,11]; (2) in order to make sense how many students were ready to write quality programs complying with coding standards, we did a case study and found that the results were not so satisfactory [12] because of the lack of consistent training and timely feedback; (3) a web-based evaluating platform was constructed, with which students can upload their programs anytime and get benchmarking results and detailed shortcomings of their programs on coding standards; (4) based on our previous work, a teaching model of coding standards based on evaluation index system and evaluating platform was proposed [13]. Even though we have made some progress, there are still lots of works to do in this subject. For example, a questionnaire designed for software industry should be delivered to make the evaluation index system more practical. Also, the teaching model should be refined with the accumulation of our experiences on programming languages teaching. Programming is a kind of art [14] so that it takes time for every engineer to reach the apogee of software quality.

## REFERENCES

- [1] R. S. Pressman, Pressman, "Software engineering: A practitioner's approach," (6e), McGraw-Hill, 2005.
- [2] S. Herb and A. Andrei, "C++ coding standards 101 rules, guidelines, and best practices," Pearson Education Asia Ltd, 2006.
- [3] P. W. Oman and C. P. Cook. "A taxonomy for programming style," pp. 244–250, 1990,  
<http://doi.acm.org/10.1145/100348.100385>
- [4] P. W. Oman and C. P. Cook, "A paradigm for programming style research," SINGPLAN notices, Vol. 23, No. 12, 1990.
- [5] P. E. Berry and B. A. E. Meekings, "A style analysis of C programs," Communications of ACM, No. 28, pp. 80, January 1985.
- [6] X. S. Li and C. Prasad, "Effectively teaching coding standards in programming," in Proceedings of SIGITE'05, Newark, New Jersey, USA, October 20–22, 2005.
- [7] W. Kobitzsch, D. Rombach, and R. L. Feldmann, "Outsourcing in India," Software, IEEE, Vol. 18, No. 2, pp. 78–86, March–April 2001.
- [8] TIOBE Software BV, "TIOBE Coding Standard Methodology," 2003, Accessed March 4, 2005.  
<http://www.tiobe.com/standards/tekst.htm>
- [9] B. W. Kernighan and P. J. Plauger, "The Elements of Programming Style (2e)," McGraw Hill, New York, ISBN 0-07-034207-5, 1978.
- [10] Y. Q. Wang, J. Z. Wang, et al., "Quantitative research on how much students comply with coding standard in their programming practices," in the Proceedings of the 3rd China Europe International Symposium on Software Industry Oriented Education (CEIS-SIOE'2007), Dublin, Ireland, pp. 116–119, February 6–7, 2007.
- [11] Y. Q. Wang, J. Z. Wang, et al., "A framework for quantitative evaluation of coding standards in programming language teaching," Journal of Hefei University of Technology (Social Science), Vol. 3, pp. 67–71, 2008. (In Chinese)
- [12] Y. Q. Wang, H. D. Su, et al., "How many students are ready to write quality programs complying with coding standards: A case study," in the Proceedings of the 4th China Europe International Symposium on Software Industry Oriented Education (Guangzhou, China, January 10–11, 2008). CEISIE'2008. Zhongshan Daxue Xuebao/ Acta Scientiarum Naturalium Universitatis Sunyatseni, Vol. 46, No. SUPPL, pp. 93–96, December 2007.
- [13] Y. Q. Wang, L. Lei, et al., "Teaching model of coding standards based on evaluation index system and evaluating platform," in the Proceedings of 2008 International Conference on Computer Science and Software Engineering (Wuhan, China, December 12–14) CSSE'2008. IEEE Computer Society. (to be published)
- [14] D. E. Knuth, "The art of computer programming," Addison Wesley, 1999.