Software Quality Prediction Model Research Based on Object-Oriented Petri Nets

Fen. He, Aihua. Ren, and Ziran. Ding
Beihang University, Beijing, China
Email: {fenheHelen, dingziranrr}@163.com, renah@buaa.edu.cn

***Abstract*—With the increasing complexity and size of software system, the difficulty of managing software quality is growing rapidly. How to ensure software quality has always been the important issue that needs to be solved. This paper builds a software quality predicting model for solving this issue. In order to realize this, it establishes OOPN (Object-oriented Petri nets) model to describe software development process which are organized as waterfall order. The stages are divided into three parts: previous stages, current stage and following stages. For the previous stages, the defects number could obtained by respective review activities. For the current and following stages, the defects number could predicted by the software quality regression model of each stage, which is built by using multiple linear regression method based on history data. At last, simulating the whole OOPN model and the Rayleigh function could be achieved by using the defects number of each stage. The Rayleigh function could be used to predict the defects number of software product.**

***Index Terms*—software quality predcting, OOPN model, multiple linear regression, rayleigh model**

I. Introduction

From the late 1960s, the development of software engineering has gone through a very long process, which from basic period, development period, the consideration period to the quality and efficiency period. For decades, software development has been fraught with problems of discipline, such as: cost overruns, schedule delays, difficult to maintain and lower customer satisfaction. These problems are closely related to the software development process. In 1989, Watts S. Humphrey, the founder of CMM (Capability Maturity Model), proposed that software development tasks can be viewed as a series of controllable, measurable and improved processes [1]. Following, CMMI is proposed by Software Engineering Institute of Carnegie Mellon University. According to CMMI, it is available to establish software quality prediction model based on analyzing the current situation and state of software organizations and the software quality goal could be achieved by continuous improvement of the software development process. Since software quality prediction model is hard expressed by mathematical formula, so it needs process simulation to describe. Petri net is a kind of graphical tool for mathematical modeling and simulation by using token to describe the flow of the system dynamic process [2]. For the basic Petri net may produce state explode issue while modeling middle-sized system [3], [4], OOPN produced by combining the basic Petri net with object-oriented technology. By using OOPN, the software development process is visualized. By simulating the OOPN model the Rayleigh function of whole model can obtained, which could be used to predict the number of defects in product delivery. For the above reasons, it is natural to use OOPN to establish the software development process model and then build software quality predicting model.

Releted Theory

A. Object-Oriented Petri Nets

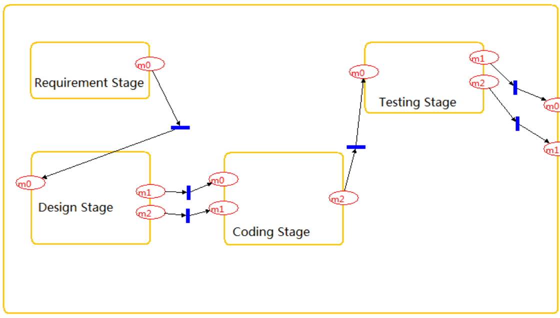

Figure 1. A sample of OOPN

The Basic Petri net is a mathematical modeling tool with rigorous mathematical foundation and graphical depiction. The Basic Petri net can use network graphic to represent system model, and its constituent elements are: place, transition, directed arc and token, which respectively represents the actual system state, action, relationships and flow of objects [2]. The Basic Petri net is suitable for asynchronous and concurrent system modeling and analysis. The circle represents place, solid dot indicates token, rectangle indicates transition and arrow indicates the flow of relationship. Compared to the Basic Petri net, Object-Oriented Petri Nets (OOPN) has combined the feature of object-oriented, such as inheritance, polymorphism, and permission rules. In OOPN, predicate network subnet describes the object template, namely model class and the rectangular box enclosed subnets represents encapsulation and abstraction [3]. Model classes can be divided into composite class and simple class. The composite class is constructed of simple class or composite class. The model classes

Manuscript received April 24, 2014; revised June 27, 2014.

doi: 10.12720/ijeee.3.3.225-229

communicate with each other through the "message queuing" and "gate", which can effectively solve the state explosion problem of the Basic Petri net. Fig. 1 shows an OOPN model that describes software development process, which is consist of requirement stage model, design stage model, coding stage model and testing stage model.

B. Rayleigh Model

In 1978, Putnam proposed that software project followed the life cycle model which described by the Rayleigh density curve [5]. Recent scientific studies show that many software projects defects removal model follow the Rayleigh model, which is shown in Fig. 2. And the number of defects in maintenance is the estimated target of Rayleigh model, which can be predicted by establishing appropriate Rayleigh model.

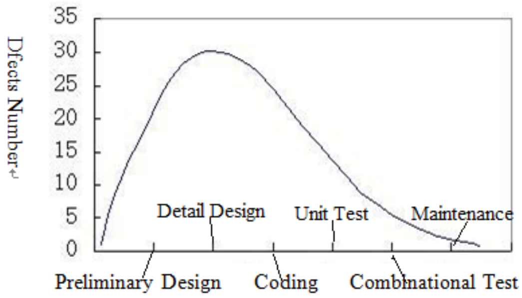

Figure 2. Rayleigh model of defect removal model

Rayleigh model is a reliability model to manage software quality. Its core idea expresses the key points of early defect prevention and defect removal. In Rayleigh model, if the rate of implanting error is lower, then the area of entire area under the curve and the potential defects of product reduce. On the other hand, if the error as much as possible discover and remove in an early stage of software development, then the number of defects of later stages will greatly reduce. The framework of Rayleigh model can be used as the foundation of software quality improvement strategy, in particular defect prevention and early defect removal. These two aspects are the main direction to improve software quality. As Fig. 3 shows, the goal of software quality management is to reduce as much as possible the height of Rayleigh curve.

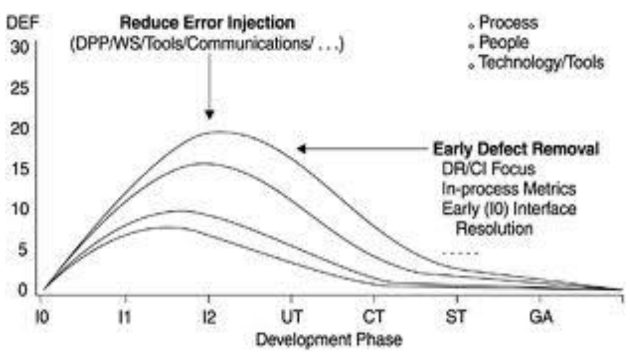

Figure 3. The direction of software quality improvement

II. The Research of Software Quality Prediction Model Based on OOPN

Before building the software quality prediction model, it needs to be solved how to model the software development process by using OOPN. According to the meaning of OOPN elements, the place can represents the project state, transition indicates the actions of software development, directed arc means the relationship between states and transitions, token means the flow of objects and message queuing indicates the output and input. By using this method, OOPN model of each stage in software development can be built. These stages are requirement analysis, preliminary design, detail design, coding, unit testing, integration testing and system testing. At last, the software development process OOPN model can be built by the gate to connect OOPN models as the development order.

Rayleigh function indicates the relationship between the defects number and time. So it can be used to predict the final defects number in product delivery stage. Rayleigh function can be generated when the number of defects in each development stage is obtained. First, the stages are divided into three parts: previous stages, current stage and following stages. For the previous stages of software development, the number of defect is obtained by the actual data from related review activity. While the number of defects in current and following stages, they are calculated by respective software quality regression model which is built based on the history data. Quality regression model of each development phase is built by multiple regression technology. The dependent variable of each model is the number of defects in respective stage, and independent variables are the quality control factors of each stage. Thus, the factors that affect the quality of software need to be identified and controlled to ensure predictable and measurable software [6]. For general software development projects, the relationship between defects number and quality control factors is complex non-linear rather than simple linear. Thus, it is reasonable to adopt Multivariate nonlinear regression equation fitting data. For reduce the complexity of building and solving process of non-linear regression equation, the logarithm operation is used, such as (1).

$$\ln Y = \beta 0 + \beta 1 \ln X1 + \beta 2 \ln X2 + \ldots + \beta m \ln X m \quad (1)$$

After the defects number of each stage is obtained, then stimulate the OOPN model to generate Rayleigh function, which indicates the relationship between the defects number and the time. And Rayleigh function can be used to predict the defects number in product delivery stage.

To sum up, the process of establishing software quality predicting model based on OOPN could divided into the following seven steps: First, building software development process model by using OOPN; Second, input the defects number of previous stages, which are obtained by the review activities, Third, importing project history data for the current and following stages, which could be used to build the quality regression model, Fourth, establishing software quality regression model for the current and following stages, where in a software quality regression model the dependent variable is the defects number of stage and the independent variables are the controllable quality factors; Fifth, generating OOPN simulation model and make it instantiated; Sixth, simulating OOPN simulation model, and obtain the Rayleigh function; Seventh, using the Rayleigh function to predict the defects number of product delivery stage. The flow chart of generating software quality prediction model is shown in Fig. 4.

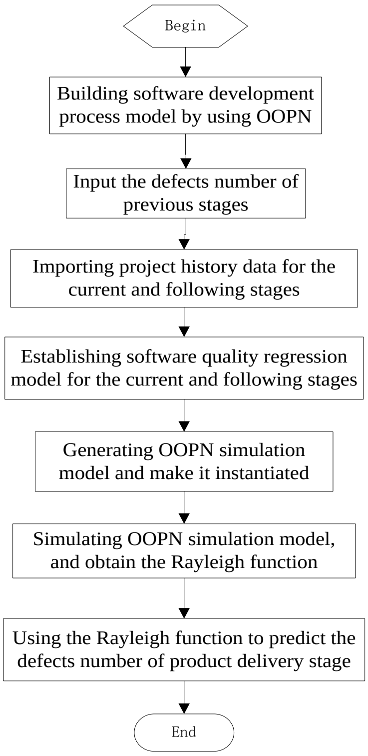

Figure 4. The flowchart of software quality predicting model based OOPN

III. An Application of Building Software Quality Pridecting Model

There are many software development models recently. But waterfall model is the most wide used in software development. And many other models are based on the waterfall model, such as iterative development model. Therefore, this paper will establish an OOPN model of waterfall model, which is the foundation of software quality predicting model. In general, waterfall model includes the following stages: requirement analysis, preliminary design, detail design, coding, unit testing, integration testing, system testing and product delivery. The current stage of project can be described by the state element of OOPN, which is represented by a circle in graph. And output of each stage can be described by the message queue element of OOPN, which is represented by an ellipse in graph. There is a transition element between every two stages, which is used for triggering the beginning of next stage. In this way, the OOPN model of waterfall model is established. For simplicity, here the waterfall model adopted simple form. And the graph of this model is as Fig. 5.

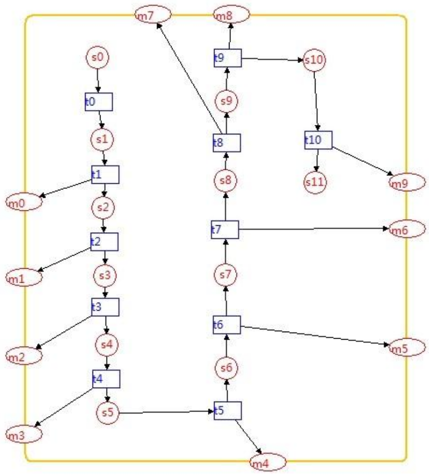

Figure 5. OOPN model of waterfall

In Fig. 5, so represents the initial state, t0 indicates the event of triggering the requirement analysis activity, s1 represents the ongoing state of requirement analysis, t1 indicates the events of ending the requirement analysis, and triggering the generation of requirement specification and requirement review activity, m0 delegates the requirement specification, s2 represents ongoing state of requirement review, t2 indicates the events of ending the requirement review, and triggering the generation of requirement review report and preliminary design activity, s3 represents the ongoing state of preliminary design, t3 indicates the events of ending the preliminary design and triggering the generation of preliminary design specification and preliminary design review activity, m2 delegates the preliminary design specification, s4 represents the ongoing state of preliminary design review, t4 indicates the events of ending of preliminary design review, and triggering the generation of preliminary design review report and detail design activity, m3 delegates the preliminary design review report, s5 represents the ongoing of the detail design, t5 indicates the events of ending the detail design and triggering the generation of detail design specification and detail design review, m5 delegates the detail design specification, s6 represents the ongoing of detail design review, t6 indicate the events of ending the detail design review , and triggering the generation of design review report and coding activity, m6 delegates the detail design review report, s7 represents the ongoing state of coding activity, t7 indicates the events of ending the coding activity, and triggering the generation of product code and the unit testing activity, m7 delegates the product code, s8 represents the ongoing state of unit testing, t8 indicates the events of ending unit testing activity, and triggering the generation of unit testing report and integration testing activity, m8 delegates the unit testing report, s9 represents the ongoing state of integration testing activity, t9 indicates the events of ending the integration testing activity, and triggering the generation of integration testing report and the system testing activity, m8 delegates the integration testing report, s10 represents the ongoing state of system testing, t10 indicates the events of ending system testing activity, and triggering the generation of system test report and Rayleigh function and beginning of product delivery activity, m9 delegates the system testing report, s11 represents the state of product delivery.

Assume the current stage is coding stage. So for requirement analysis, preliminary design, detail design stages, the defects numbers can be obtained by review activity. The defects number information of requirement analysis stage could be added in transition t2, which could be set as Fig. 6 show. And the defects number information of preliminary design, detail design stages could also added in t4 and t6.

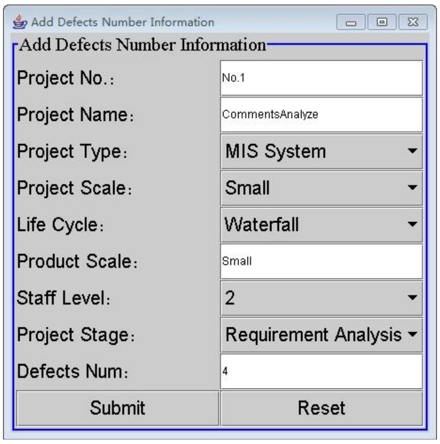

Figure 6. Set defects number of requirement analysis

For the coding, unit testing, integration testing, system testing and product delivery stages, the defects number information should gained by the quality regression model. Before building the quality regression model of each stage, the history quality controllable factors and defects number data in each stages of the organization must be added. The data could be added as Fig. 7 show.

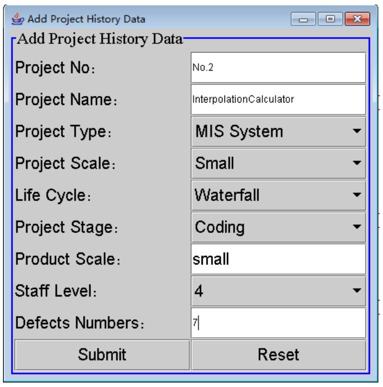

Figure 7. Add project history data of coding stage

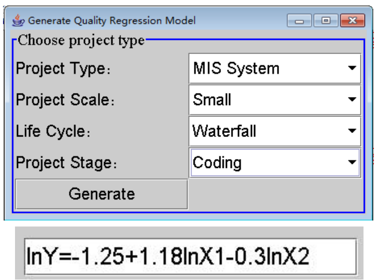

Figure 8. Generate quality regression model and result of coding stage

After import the history data of Coding stage, then the data could be used to building the quality regression model of Coding stage by multiple regression technology. The generation of quality regression model and the result is show in Fig. 8.

For the other stages, such as unit testing, integration test, system test, and product delivery stages, the regression model could be gained as the coding stage. After building the current and following stages quality regression model, then generate the OOPN simulation model. The OOPN simulation model is show in Fig. 9.

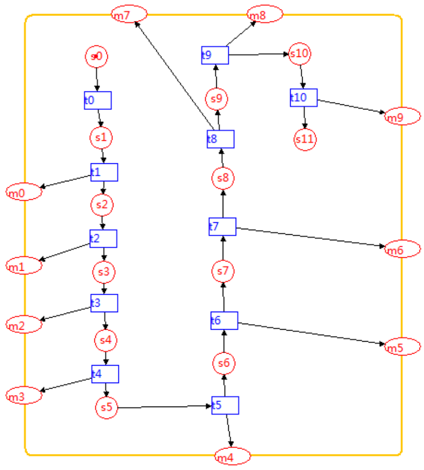

Figure 9. OOPN simulation model

For simulating the OOPN model of waterfall model, it needs to create the simulation instance of the model. At the beginning, there is a token in initial state s0. When it starts to stimulate the OOPN model, then the token will flow into the transition t0 which is connected with s0 and s1 with different direction. After t0 received the token, then it will trigger the start of requirement analysis activity and meantime the token will flow into state s1. When requirement analysis finished, then the token will flow into t1, and trigger the events of generating the requirement specification and the start of requirement review. After that, m0 will receive a token which means the generation of requirement specification. And the token flow into s2, which means the ongoing state of requirement review. When requirement review is ended, then the token will flow into t2, which means trigger the events of generating requirement review report, ending the requirement review, passing the defects number of requirement analysis to the preliminary design activity and starting preliminary design activity. For the rest previous stages, such as preliminary design (s3 to t4), detail design (s5 to t6), the simulation process is the same as requirement analysis (s1 to t2). For the current coding stage (s7 to t7) and the following stages, such as unit test (s8 to t8), integration test (s9 to t9), the simulation process is the same as coding stage (s7 to t7). When s7 receive the token, it means the project is ongoing of coding. After coding ends, the t7 receive a token, which triggers the collection of the quality controllable factors data, here is project scale and staff level, using the regression model to calculator the defects number of coding stage, then ending coding stage, passing the defects number to product unit system and start the unit system activity. For system test stage (s10 to t10), when s10 receive a token, it means the ongoing of system test. When system test ended, the token will flow into t10, it trigger the generation of system test report, the collection of quality controllable factors data, here is system test scale and system tester level, using the system test quality regression model to calculate the defects number of system test, using the defect number data of previous stages before product delivery to generate the Rayleigh function of whole model, and then use it to predict the defects number of product delivery stage, which would passed to s11, the product delivery state. The Rayleigh function generated as show in Fig. 10.

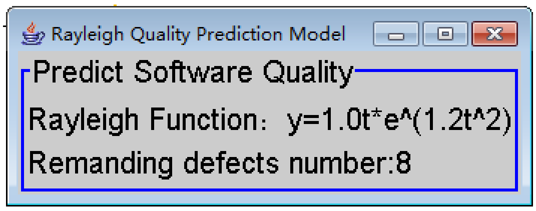

Figure 10. Generate rayleigh function and the prediction result

IV. CONCLUSION

Software quality predicting model mainly used to predict the quality of software products. Modeling the software quality predict model based on OOPN involves the following steps: First, building software development process model by using OOPN; Second, input the defects number of previous stages, which are obtained by the review activities, Third, importing project history data for the current and following stages, which could be used to build the quality regression model, Fourth, establishing software quality regression model for the current and following stages, where in a software quality regression model the dependent variable is the defects number of stage and the independent variables are the controllable quality factors; Fifth, generating OOPN simulation model and make it instantiated; Sixth, simulating OOPN simulation model, and obtain the Rayleigh function; Seventh, using the Rayleigh function to predict the defects number of product delivery stage.

ACKNOWLEDGMENT

Thanks to the reviewers of the conference. Thanks for my teacher and my classmates in our lab. They help and support me all the time.

REFERENCES

[1] Y. Chong-Yi, *Principle and Application of Petri Nets*, Publishing House of Electronics Industry, 2005.
[2] Z. Yu and Y. Cai, "Object-Oriented Petri nets based architecture description language for multi-agent systems," *IJCSNS*, vol. 6, no. 1, 2006
[3] Y. K. Lee and S. J. Park, "OPNets: An object-oriented high-level petri net model for real-time system modeling," *J. of Systems and Software*, vol. 20, no. 1, pp. 69-86, Jan 1993.
[4] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in *Magnetism*, vol. III, G. T. Rado and H. Suhl, Ed. New York: Academic, 1963, pp. 271-350.
[5] S. H. Kan, *Metrics and Models in Software Quality Engineering*, Addison-Wesley Longman Publishing Co., Inc., 2002.
[6] G. Denaro and M. Pezze, "Petri nets and software engineering," in *Lectures on Concurrency and Petri Nets*, Springer Berlin Heidelberg, 2004, 439-466.

Aihua Ren got Ph.D. degree from Beihang University in 2001 and MS degree from the State University of New York at Buffalo in 1988.
She is the Professor of Beihang University and SEI certificated SCAMPI Lead Appraiser. Aihua Ren's Interested Areas are: Software Engineering, Software Process Improvements and Appraisals

Fen. He is a master student in the Department of Computer Sicence and Engineering, Beihang University, China. She got her Bachelor Degree in Computer Science and Technology from China University of Petroleum (Beijing) in 2012. Her research field is Application of Object-Oriented Petri Nets.
She is very interested in C# development, process improvement and software quality.
Ms. He have achieved scholarship every year and competition in her college stage. She became a member of China Computer Federation in 2014.

Ding Ziran is a master student in the Department of Computer Sicence and Engineering, Beihang University, China. He received his B.E. in Computer Science from Beihang University. He is studying for master's degree at Beihang University.
His research interests are in Software engineering: process improvement, agile method, functional programming, agile project management, empirical software engineering, software architecture and software measurement.