An Analytic Study for Software Quality Management Based on Software Process Improvement

Noriaki Izumi†
Intelligent Systems Research Institute,
National Institute of Advanced Industrial Science and Technology (AIST),
Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568 JAPAN
+81-29-861-5201, Email: n.izumi@aist.go.jp

Takahiro Seino
Faculty of International Social Studies, Maebashi Kyoai Gakuen College.
1154-4 Koyaharamachi, Maebashi, Gunma, 370-2192 JAPAN
+81- 27-266-7575, seino@c.kyoai.ac.jp

Motoi Yamane, Koichi Hayashi
Peacemeal Technology Inc.
Triton Square, Office Tower X 14F 1-8-10 Harumi Chuo Tokyo 104-6014 JAPAN
+81-3-6220-1471, {motoi.yamane, koichi.hayashi}@pmtech.co.jp

Abstract

The development of high-quality software has been recognized as a significant issue of the software industry.. In a development project of a large-scale information system, it is important to anticipate, respond and react to the problems of software design and implementation. In fact, it is difficult to identify the root causes of poor software quality due to the quantity-aspects in finding software bugs including errors and other defects. As a result, software quality management has been focused on software testing as the process of validating and verifying that the implemented system works as expected. In order for the development project to improve its product quality, we propose a methodology for measurement and improvement of software quality according to the standardized processes and products based on a reference framework. In order to identify a root cause of poor software quality, a real development project in million-user service of Japanese public sector has been investigated from a standpoint of the software process. Through the detailed investigation with our proposed framework of traceability analysis, eight primary causes in three subject areas have been identified as patterns that result in poor software quality. Furthermore, in order to mitigate the damaging effects on the poor quality, we propose how to reduce misunderstandings of the project's standards and references, as a framework of software process improvement, according to the identified patterns. For the further discussion of our proposed methodology, an experimental study has been provided with fifty software engineers. In the case study, quality measures are collected and improving patterns are applied to keep a consistency among the developed product and the project's standards. In the consequence of the case study, we have confirmed that our proposed framework contribute to keep a quality controlled and an efficiency increased in a large-scale software development.

Keywords: Software quality management, Software process, Process improvement, Traceability analysis, Quality measure.

1. INTRODUCTION

Generally, a project of a system development is performed based on various criterions for processes, qualities, costs, and so on. Although the cost injection becomes a major strategy for the quality improvement in many cases, it is difficult to identify the root cause of poor software quality. In fact, quality control operations are concentrated on the product modification with respect to reviews, inspections and walkthroughs, as the response of th e various management tools. In order to improve a software process for software quali ty, it is very important to measure the product status from the viewpoint of the qualit y issues.

From the above standpoint, this paper describes a strategy for the software quality manageme nt based on a standardized software process. We have demonstrated the proposed methodo logy applying the actual project as a case study. The project adopts a large-scale syste m development by employing a reference architecture of Java framework as a process and product standard.

2. REFERENCE ARCHITECTURE FOR DEVELOPMENT CASE

In order to make exact assessment of software product quality, reference architecture of Java development framework is employed as a set of a project's standards and references for a development project. Process standard of the framework defines requirement analysis, design, implementation and administration processes. In this paper, we will address issues and process improvements on the requirement analysis process.

3. REQUIREMENT DEFINITION BASED ON REFERENCE ARCHITEC TURE

Requirement analysis is a typical process of a system development. Thus, it is th e key process of software quality. In order to measure the software quality including medium products, almost all the processes contain the following four tasks for comple ting the products.

Task 1: Drafting Products

First, the development team generates the draft version of a product set based on all the information obtained from the client such as operational manuals, described business rules, and so on. Then, the team holds a session to make an interview with users of the system. They review their products and make a list of the clarified point based on the project standards including the reference architecture.

Task 2: Interviewing Each Person in Charge

When the drafts are completed, developers start to interview by sharing the draft products and questionnaires with each individual person in charge. The purpose of this task is to check whether operating procedures and rules are correctly described, and to extract issues in current operations.

Task 3: Product Correction

When the interview is finished, developers modify and correct the drafted products according to the comments obtained.

The developers had better perform internal reviews for the updated products and move to the next task when the review of the products is completed.

Task 4: Inspection

Inspection is defined as a task to check whether the product satisfies the quality for providing sufficient specification to the following processes. Inspection should be performed not only about individual products, but also about the consistency among the related products. This task is carried out as a meeting of a formal software inspection.

4. ANALYTIC EXPREIMENT OF PROCESS IMPROVEMENT BASED ON PRODUCT QUALITY

In order to investigate a relationship between software process and product quality, a real project of a large-scale system development is focused, which employs Java framework as a reference architecture. This project has 4 subject areas and the scale of the development is approximately 1,500 million yen. We have observed the activities of this project and show the detailed record of requirement analysis process as follows.

Table 1: Products and their quantities in the case study.

Product Name	Quantity
Scope of Business Activities	1
A List of Actors	1
A List of Workflow Diagrams	1
Workflow Dialog	37
A List of Business Rules	1
Business Rule Descriptions	65
A List of Business Forms	1
Glossary	1
A List of Problems	1
Records of Interview Meeting	32

Table 1 shows a list of the generated products created in the first subject area and their volumes. Workflow diagrams and business rule descriptions play the main roles in the requirement analysis. So, quantity of these products will commensurately increase with respect to a number of business activities. In this case study, we have 37 workflow diagrams and 65 business rule descriptions in the first subject.

We have examined these products with software inspection as the final task of the requirement analysis, the result of this is that we have obtained 629 findings as comments about quality issues of the above products.

4.1 Classification of the Quality Issues

We will summarize the quality issues raised in the above observation. We classify the issues the following three aspects: a) process, b) format and c) content. We show more detailed classification for each aspect in Table 2.

Table 2: Classification of quality issues.

classification	Description
process	
process accordance	violation of process.
product accordance	missing product.
formatting	
format	standard formats are not used.
incomplete form	missing required information,
traceability	wrong traceability information,
style	violation of style guidelines.
contents	
wrong description	wrong description about business activities (differ from the explained by person in charge.)
incomplete description	imcomplete description about business activities (differ from the explained by person in charge.)
granularity	various granularity.
or consistency	inconsistent between other products.

a) process aspect issue points a difference between are difference on a progress between how developers should proceed their task defined by the process standard and they had actually performed. In this case study, since developers performed the task in accordance with the process standards, there are no comments about this aspect issues. We consider that whether they have carried out the task as defined and whether they have created the all products as the standard defined.

b) format aspect issues are non-conformance of product standard. The product standard employed defines products' format, required items, traceability constraints and style guidelines. There are totally 123 findings about this aspect issues.

c) content aspect issues are misunderstanding business activity explained by each person in charge. There are totally 506 findings about this aspect issues.

These three aspects aim to make analyzing causes and planning improvements easier. The result of this analysis is shown in Table 3. Table 3 shows us the following two points.

First, most quality issues are in business process diagram and business rule descriptions. Because these products concretely describe business activity users explained, this result is by necessity.

Second, we need to pay our attention to the distribution in raised issues. Although there are no process aspect issues, there are many format aspect issues and there are still more content aspect issues. This result shows us the followings:

Table 3: Raised issues in our case study.

classification	Scope of Business Activities	A List of Actors	A List of Workflow Diagrams	Workflow Diagrams	A List of Business Rules	Business Rule Descriptions	A List of Business Forms	Glossary	A List of Problems	Records of Interview Meetings	Other comments	Subtotal	Total
process													
process accordance												0	
product accordance												0	0
formatting													
format							1					1	
incomplete form												0	
traceability				35	1	41	3	4	1			85	
style	5	6	3	13	2	3	2	1			2	37	123
contents													
wrong description	24	1	3	172	8	72	10	4	7	6		307	
incomplete description				19		4						23	
granualiry of consistency	3	1	2	123	2	29	2	2	4	8		176	506
total	32	8	8	362	13	149	18	11	12	14	2	629	629

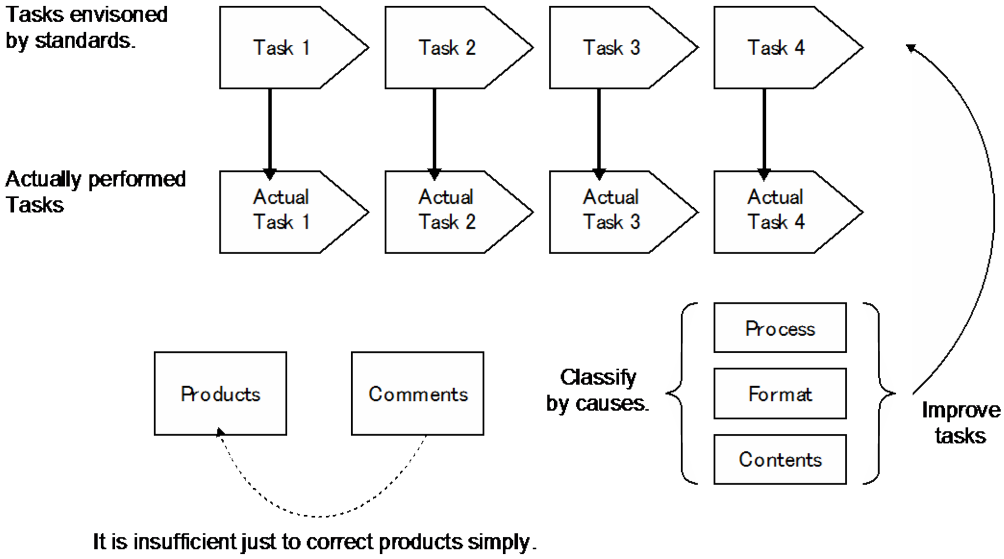

Figure 1: Improvement Guideline.

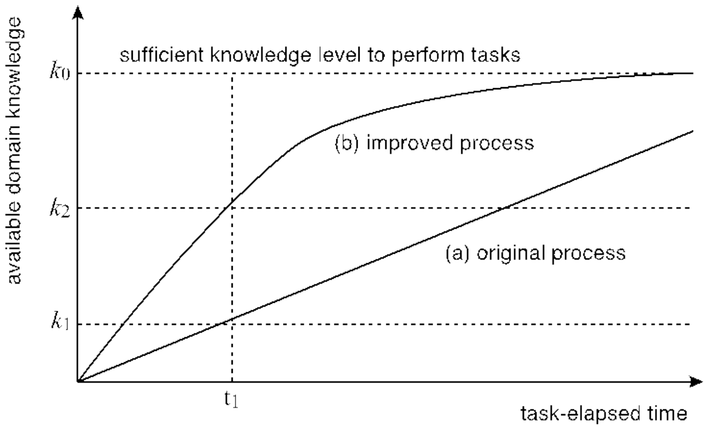

Figure 2: Acquisition of knowledge.

- The team has performed their tasks accordingly to the process standard and has created the products accordingly to the product standard.
- Content aspect issues shown in Table 3 are four or more times as more as than format aspect issues. We have examined comments which are classified as content aspect issues, and then we have concluded that most comments were indication that the told business activity was not correctly described by the products.

Figure 1 shows how the process improvement is performed when comments on related products are filled with findings.

5. DISCUSSION

5.1 Analytic Investigation of Process and Quality

Each quality issues shown in Table 3 can be solved to correct its related products and we can improve the quality of the whole products. However, the effect of the improvement a bout the fixing work is limited to the quality in the related subject area. If we carry out of this process to the other subject areas, same kinds of issues will be occurred. In order to prevent these repetitions, we need to take step toward process improvement of the originally adopted process standard shown in Figure 2.

A developer team needs a certain time to learn the expertise as well as enough skill to perform the analysis. Note that the knowledge is not only the business activity, but also the standards of processes and products. The team should share the expertise and each member performs his works according to the shared knowledge.

We show that time series transition of the knowledge in Figure 2. The developer team gradually stores the domain knowledge through learning documents about the business activities and interviewing or observing the users. Similarly, they gradually acquires the knowledge about the standards through their working or learning.

We think that the reason why raising quality issues in Table 4 because there are gaps of the expertise and the skills between developers and users in a moment (t_1 in Figure 2). We need to bridge these gaps, by understanding correctly the information about the users' state, expressing them correctly in products and maintaining the consistency between documents.

Table 4: The number of quality issues before and behind the process improvement.

team	number of workflow diagrams	total of pages	number of detected issues	average number per page
A	7	25	60	3.6
B	7	22	20	0.91

In order to acquire knowledge quickly and to propagate them on the team, it is necessary to improve how to advance. Note that we should consider some assumptions as variation factors such as in case of absorbing the business activities beforehand or the business activities are well-documented.

In this case study, we find out the reason why quality issues of the product such as incorrectness. It is because developers did not have sufficient domain knowledge, and learned slowly. The granularity and consistency issues have not been clarified because of ambiguity about boundaries. They did not share what are similar activities and descriptive policies in the team.

5.2 Process Improvement and Its Effects

We change the requirement analysis process shown in Figure 4. We add a preliminary interview task before an interview task. This new task aims that the development team grasps the overview of business activities and tasks over the gaps about the business domain. Therefore, this interviewee should be a domain expert who knows a whole of the business activities such as a manager.

Another development team performs requirement analysis for the other subject area adopting the new process shown in Figure 4. In this team, raised quality issues on the software inspection are significantly reduced. As shown in Table 4 the first team has created 7 workflow diagrams which amount to a total of 25 pages and has been pointed out 60 quality issues, that is, the products contained 3.6 issues per page. While on the other hand, the second team have created 7 workflow diagrams which amount to a total of 22 pages, and has been pointed out 20 issues, that is, the products contained 0.91 issues per page.

Moreover, we have observed the following three additional improvements. First, this team has completed the products with efforts fewer than the first team adopting the original process. This result has been clearly grasped to schedule management. In the first subject area, the team has many delays, which the biggest one is ten days behind. Because each subject is different from its difficulty and/or its amount of the business activities, we should compare carefully. In the second subject area, almost tasks have been completed on schedules; the remaining ones have been completed few days behind or ahead of the schedule.

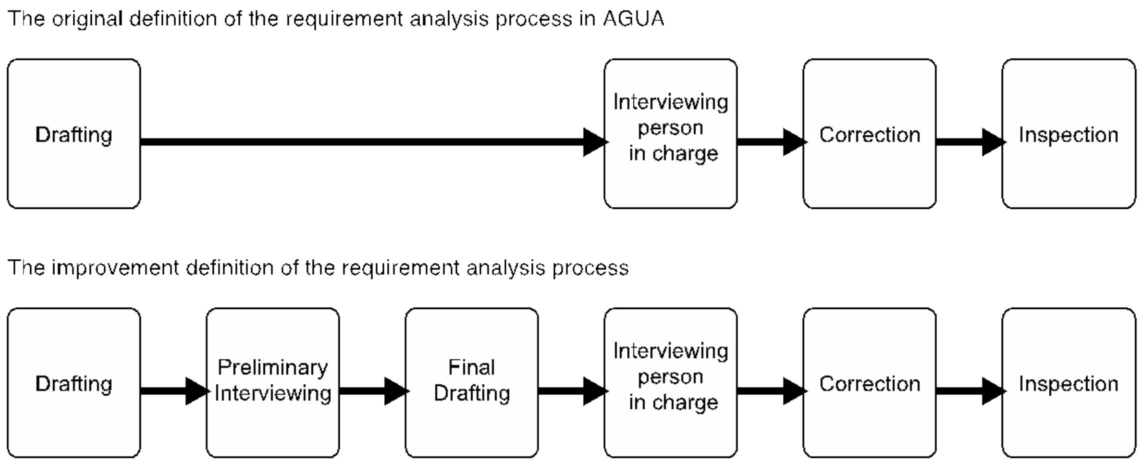

Figure 4: An Example of the Improvement Process.

Second, the new process allows the team to understand the details of the individual business activity quickly. The main reason is that the team has grasped some certain patterns peculiarly to a business organization in the preliminary interview. This patterning drives forward to improve the quality of the draft version of products and narrow down beforehand the questions about details of the individual business activity for the successive interviews.

Finally, the team has mitigated the difference of notations such as granularity of descriptions, terminological representation with they have independently introduced some notational rules for this project through drafting. By the favor of such efforts, the style of the products has become well-choreographed.

Accordingly, the process improvement have not only improved the quality of products, but also turned into more productive works.

6. RELATED WORKS

The lessons learned from experiences with goal-oriented measurement have been structured into practical guidelines (Briand et al. 1996) for efficient and useful software measurement aimed at process improvement in industry.

As the following research, many efforts, in the Software Process Improvement (SPI) literature and an empirical study, have been forcused on capability maturity model (CMM) (Paulk 2002).

The current problem with SPI is not a lack of standard or model, but rather a lack of an effective strategy to successfully implement these standards or models. In the design of the maturity model, the concept of critical success factors (CSFs) have been proposed (Niazi et al. 2005).

The problem of optimally allocating effort between software construction and debugging has been studied (Yonghua et al. 2005).

Although these research results have many advantages in the detailed model, they focused on long terms aspect on abstract activity.

In the area of management research, the relationship between life-cycle productivity and conformance quality in software products (Krishnan et al. 2000) has been investigated. This investigation is expanded into an analysis between quality improvement and infrastructure cost activity (Donald et al. 2003).

From the standpoint of releasing the new version software on time at minimum cost in the response to the change of business environment, deterministic models has been investigated (Iravani et al. 2012).

In order to make the above research detailed in the software production activity, a systematic literature review has been performed (Unterkalmsteiner et al. 2012) to identify and characterize evaluation strategies and measurements. Furthermore, as a latest research focused on real development activity, various metrics have been measured about time and quantity(Kohichi et al. 2012).

These efforts provide us some lessons about partitioned business forms into development groups and determined staffing levels for each group. Unfortunately, the lessons is concentrated on organizational aspects and far from software development activity. Our proposed framework enables us to improve software process and activity focusing on product quality as real time feedback.

7. CONCLUSION

It is important for the large-scale system development to improve task of software process as well as the measurement and a correction of the product quality. From a standpoint of quality driven SPI, we have proposed a methodology that enables a project improves its software process from an indication about quality inspection. In order to remove the root cause in actual tasks, we provide the detailed aspects about quality inspections of products based on a software process of reference architecture.

In this paper, we mentioned about the concrete improvement study in the particular project. However, there are many different conditions depending on each project such as project's domains and scales, or experiences and skills of a staff. Therefore, such conditions may have some effect on what to be improved, and we are required to establish generic methodology to improve processes based on standardized processes and measured indexes of the product qualities. We leave these matters as the future works.

REFERENCES

Boehm, B.W. (1976) Software engineering. *IEEE Transactions on Computers*, **25(12)**. 1226-1241,.

Briand, L.C., Differding, C.M., and Rombach, H.D. (1996) Practical guidelines for measurement-based process improvement. *Software Process: Improvement and Practice*, 2.

Harter, D.E. and Slaughter, S.A. (2003) Quality improvement and infrastructure activity costs in software development: A longitudinal analysis. *Management Science*, **49**.

Iravani, F., Dasu, S., and Ahmadi, R. (2012) A hierarchical framework for organizing a software development process. *Operations Research*, **60**.

Ji, Y., Mookerjee, V.S., and Sethi, S.P. (2005) Optimal software development: A control theoretic approach. *Information Systems Research*, **16**.

Kohichi, U., Igaki, H., Higo, Y., and Kusumoto, S. (2012) A study of student experience metrics for software development pbl. In *2012 13th ACIS International Conference on SNPD*, 465-469.

Krishnan, M.S., Kriebel, C.H., Kekre, S., and Mukhopadhyay, T. (2000) An empirical analysis of productivity and quality in software products. *Management Science*, **46**.

Niazi, M., Wilson, D., and Zowghi, D. (2005) A maturity model for the implementation of software process improvement: an empirical study. *Journal of Systems and Software*, **74**.

Paulk, M. (2002) *Capability Maturity Model for Software, Encyclopedia of Software Engineering*. John Wiley & Sons.

Unterkalmsteiner, M., Gorschek T., Islam, A.K.M.M., Cheng, C.K., Permadi, R.B., and Feldt, R. (2012) Evaluation and measurement of software process improvement? a systematic literature review. *IEEE Transactions on Software Engineering*, **38**.