Continuous Improvement Approach of Software Process Based on Product Quality Inspection

Takahiro Seino† 
Motoi Yamane‡ 
Koichi Hayashi‡ 
and Noriaki Izumi*†

†Faculty of International Social Studies 
Maebashi Kyoai Gakuen College, Japan
‡Piecemeal Technology Inc., Japan
*National Institute of Advanced Industrial Science and Technology (AIST), Japan

seino@c.kyoai.ac.jp, 
motoi.yamane, koichi.hayashi@pmtech.co.jp, 
n.izumi@aist.go.jp

Abstract

We propose a methodology for the 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 quality, a real development project has been investigated from the standpoint of the software process. Through the detailed analysis with our proposed framework of traceability inspection, eight primary causes in three subject areas have been identified as patterns that result in poor quality. Furthermore, in order to mitigate the damaging effects on the poor quality, we suggest ways of reducing misunderstandings of the project’s standards and references according to the identified patterns.

Keywords: Reference framework, development processes, quality control, requirement definition

1 Introduction

Generally, a project manager of development projects manages his/her project based on various indices for qualities, processes, effort, and so on. It is difficult to show whether products of the requirement analysis convey requirements correctly and exhaustively. Most projects carry out their own quality control, such as issue management. Using such traditional activities, to ensure quality does not only require additional effort, but also hardly predicts how much effort is needed.

The qualities which should be managed in such projects can be roughly divided into the following three aspects: whether the project proceeds according to the process standards, whether the products are created according to the product standard such as syntactical rules, and whether the documents convey users’ requirements exactly and exhaustively. Traditionally, most projects have such standards, and measure only such surface-level indices, and expend most effort on corrections. However, efforts just increase and quality does not improve.

By exploring the root causes of these quality problems, we should be able to solve them by changing how the work is done. From the above viewpoint, this paper describes what project improvements can be obtained from comments in software reviews about the quality of products based on the software process, while analysing the quality subject which occurred to the product, based on the analysis results. We have demonstrated the proposed methodology by examining a real project as a case study. This project engages a reference architecture called AGUA[1; 2].

2 Reference Architecture for User-centered Development

In order to make an exact assessment of software product quality, a reference architecture is engaged as a set of project’s standards and references for user-centered developments[1; 2]. The framework supposes that there are business experts who need to develop an information system, and ITC experts who can implement such a system. We call the former user and the latter developer in this paper.

The framework is designed with a glass box approach. The main concerns of users are that the system should provide functions and performances, not in technical details. On the other hand, for the developer, all required design infor-
mation needs to be disclosed. Therefore, AGUA provides the benefits of the black box for users, and the white box for developers simultaneously.

Process standard defines requirement analysis, design, implementation and administration processes. In this paper, we address issues and process improvements in the requirement analysis. As is known, the quality of the created products in requirement analysis is important and its poor quality carries high risks. If errors of requirements are discovered in the subsequent process, corrections will take time and effort compared with correcting in earlier process[3].

3 Requirement Definition Based on Reference Architecture

Requirement analysis process is the first process of a development, and its products are subsequent inputs. Investigating the current business activities consists of these four tasks:

Task 1: Drafting Products

Developer drafts a set of products based on the best explanations with all the information obtained from users, such as operational manuals.

Task 2: Interviewing Each Person in Charge

When drafts are completed, developers interview each individual person in charge. The purposes of these interviews are to check whether operating procedures and rules are correctly and exhaustively described in their products, and extraction of issues in the current operations. Developers keep track of interview.

Task 3: Modification products

Developers correct their products according to the comments of users. Developers perform internal reviews and proceed to the next task if the products pass the review.

Task 4: Inspection

Users check whether the products describe the current business activities correctly and exhaustively. Users also checks that there is a consistency between products. This task is performed in a meeting of formal inspections.

4 Analytic Experiment of Process Improvement Based on Product Quality

In order to investigate relationships between software process and product quality, a real project of a large-scale development is focused on, which engages AGUA framework. This project has over 4 subject areas and its budget is approximately 1.5 billion yen.

We describe the details of the requirement analysis process. We show the types of products created in the first subject area and their volumes in the bottom of Table 1. Now that workflow diagrams and business rule descriptions play the main roles in the requirement analysis, the quantity of them commensurately increases with business activities.

We have examined these products with an inspection, as a result we have 629 comments about quality issues of the above products.

4.1 Classification of the quality issues

We summarize the quality issues raised in the above observation. To make analyzing causes and planning improvements easier, we classify the issues as the following three aspects.

Process aspect issues are the difference between project reference and actual activities or products. We classify them into the following two sub-aspects. A process accordance is an issue with the lack of tasks to be performed or out of order. A product accordance is an issue with the lack of products to be produced on the basis of the references.

Format aspect issues are the non-conformance of syntactic aspects of the product standard. We classify them into the following four sub-aspects. Format is an issue where standard formats are not used. Incomplete form is an issue where missing information is required. Traceability is an issue where there is a lack of traceability information, or the information is wrong. Style is an issue where there is a violation of style guidelines.

Content aspect issues are misunderstanding the business activity explained by each person in charge. We classify them into the following four sub-aspects. Wrong description is an issue where the description about business activities differs from the explanation by the person in charge. Incomplete description is an issue where there is an incomplete description about business activities by the person in charge. Granularity or consistency is an issue that there are various granularity or inconsistencies with other products.

The result of this classification is shown in Table 1. It implies the following two points.
First, most issues are in the business process diagram and business rule descriptions. This is a natural result because these products concretely described the business activities explained by users.

Second, we need to pay attention to the distribution of issues. This is because there are no process aspect issues, the developer has complied with the process and product standards. Contrarily, there are many format aspect issues and there are still more content aspect issues. Content aspect issues shown in Table 1 are four or more times than format aspect. Through an examination of comments that are classified as content aspect issues, we conclude that most of the comments point out that the business activity was not described correctly.

5 Discussion

5.1 Analytic Investigation of Process and Quality

Each quality issue shown in Table 1 can be solved to correct each product; however this repair can improve quality only in the targeted subject. If we carry out the same process with another subject, the same kinds of issues will arise. In order to prevent these repetitions, we need a process improvement.

Developers need a certain time to learn the knowledge as well as have enough time to carry out the analysis. The developer gradually stores the domain knowledge through reading and understanding documents about the business activities and interviewing or observing the users. Similarly, they gradually acquire knowledge about standards.

We think that the reason why raising quality issues are important is because there are gaps in the knowledge between developers and users. We need to fill in these gaps, in order to understand correctly the information, which users give, to express them correctly in products and maintain consistency between documents. In order to acquire knowledge quickly and to disseminate it to the team, we need to improve techniques.

5.2 Process improvement and its effects

We add a preliminary interview task before the interview task of the requirement analysis. The new task aims at enabling developers to grasp the overview of business activities and fill in the gaps in their business knowledge. Therefore, this interviewee should be a business expert such as a manager.

Two other teams carried out the requirement analysis for the other two subject areas engaging the new process. In these teams, questions raised about quality issues on the inspections were significantly reduced. Moreover, we have observed the following three additional improvements.

First, both teams completed the task with less effort than the team engaged in the original process. This result has clearly shown up on schedule management. The original team had many delays, the longest one of which was ten days. Of course, we should take into account that each
Table 2. The number of quality issues before and behind the process improvement

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of workflow diagrams</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>total of pages</td>
<td>25</td>
<td>22</td>
</tr>
<tr>
<td>number of issues</td>
<td>60</td>
<td>20</td>
</tr>
<tr>
<td>average issue/page</td>
<td>3.6</td>
<td>0.91</td>
</tr>
</tbody>
</table>

The subject has its own difficulty and/or its own volumes, but in the subjects performed by both teams, most tasks were completed on schedule, the remaining being completed a few days behind or ahead of the schedule.

Second, the new process allows the team to understand the details of each business activity quickly. The main reason for this is that the team had grasped certain patterns peculiar to the business organization in the preliminary interview. This led to improving the quality of the draft version of products and narrowed down beforehand the questions about details of the individual business activity for the successive interviews.

Finally, the team mitigated the difference of notations, such as granularity or terminology, because they had introduced some notational rules. Through such efforts, the style of the product have became well-choreographed.

6 Related Works

The lessons learned from experiences with goal-oriented measurement have been structured into practical guidelines[4] 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 focused on the capability maturity model (CMM)[5].

In order to make the above research detailed in the software production activity, a systematic literature review has been performed[6] to identify and characterize evaluation strategies and measurements. Furthermore, as the latest research focused on real development, various metrics have been measured about time and quantity[7].

These efforts provide us with some lessons about partitioning business forms into development groups and determining staffing levels for each group. Unfortunately, the lessons are only concentrated on organizational aspects. 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 development to improve the task of software process as well as the measurement and correction of the product quality. From the standpoint of quality driven SPI, we have proposed a methodology that enables a project to improve its software process through 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 a reference architecture.

References