

Software Tag for Traceability and Transparency of Maintenance

Katsuro Inoue

Department of Computer Science,
Graduate School of Information Science and Technology, Osaka University
inoue@ist.osaka-u.ac.jp

Abstract

We have proposed the notion of software tag, which is a complex of various characteristic elements of software development project. Empirical data for the project is collected and abstracted into the tag, and the resulting tag is given back to the software purchaser who can evaluate the quality of software product and development process. In this paper, we will discuss an extension of the application area of the software tag into the software maintenance activities.

1. Introduction

In these days, software maintenance becomes one of important business resources for many software companies in Japan. It is very common that a software company who has developed a system continues to maintain the system and to get stable income from the software purchaser (user) more than 10 or 20 years. Also, many software companies recently start so-called maintenance business such that they take care of large systems with less maintenance cost than the original developer's cost.

In such circumstance, quality and cost of the maintenance are key factors for the software purchasers. They would concern the maintenance quality against the money they pay; however, the process and quality of the maintenance activities are generally unclear. Therefore, method to have software maintenance more transparent and traceable is strongly desired.

In this paper, we will explore the application of *software tag* to software maintenance, to clarify the software purchaser's concern.

2. Overview of Software Tag

A software tag is a package data set of a software development project[1]. It is currently composed of 41 characteristic elements of the project data and process data, as the software tag elements.

Fig. 1 shows an overview of the concept of software tag. A software tag is an abstract data set of the project's empirical data, and it is given to the software purchaser for the quality indicator of the project. The software purchaser can evaluate the tag, and know the quality of the process and product. A scenario of using the software tag is as follows.

1. A software purchaser orders development of a software system. The purchaser requires not only the final products, but also a software tag.
2. During software development, various kinds of empirical data are collected. Requirement documents, software design documents, source code, test cases, issue tracing logs, manual documents, review logs, quality analysis recodes, are examples. These data could be collected easily and effectively by a project support system such as *Empirical Project Monitor*[2]. We collect not only the final data at the end, but also interim snapshots during development.
3. Some of the collected data are selected and abstracted into the tag form. By this transformation, proprietary development know-how is eliminated from the tag, and it is hidden from the purchaser.
4. Most of the archived empirical data are encrypted for the further validation in case.

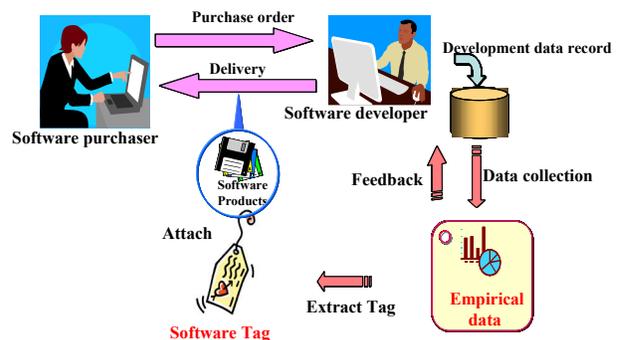


Fig. 1 Overview of software tag scheme

5. This software tag is delivered to the software purchaser associated with the final software product.
6. The software purchaser evaluates the achievement of the software developer by viewing the tag, and accepts the delivered software product.
7. When a controversy such as quality issue of the product happens between the software purchaser and developer, the encrypted empirical data is deciphered, and the original empirical data is restored. By analyzing the empirical data, correctness of the tag is validated, and a resolution for the controversy is explored.

3. Applications of Software Tag

Software tag is a key vehicle to improve "visibility" and "transparency" of software project. By software tag, the software purchaser can identify and understand the development process which was mostly secret to the developer. The purchaser can evaluate the quality of the processes and also products.

For the software developer, software tag is used to prove their proper activities in the software project. Also, it is used for tracing the quality of the activities of sub-contractors. Fig. 2 shows a hierarchy of software development in a sub-contract chain with software tags associated with each sub products.

Standardizing software tags will help to improve trade custom of software development contract. Evaluation of the software products and project based on subjective empirical data in the software tag will lead the society more healthy way.

The scheme is very useful for offshore and global development in the sense that the traceability and transparency of software development are established with fairly low overhead of the developers.

4. Extension to Software Maintenance

We have discussed mainly on software tag for the software development phase. However, the notion of software tag can be applicable to software maintenance activities although there is no explicit maintenance quality indicator in the 41 pre-defined elements.

The software maintenance process is generally composed of various activities such as debugging, designing, coding, testing, documenting, and so on. The empirical data for these activities are easily captured as the newly development projects with support systems. Some of the elements which are mainly used for the newly development projects, say, requirement, may be omitted as the tag for the

maintenance, but many other elements are used and filled with the abstracted data from the empirical data.

Software purchaser receives software tag with updated software products, and evaluates the activities and quality of the developer's maintenance work. Concern for the quality such as insufficient test efforts can be identified easily by checking the data in the tag. In the case of controversy, the raw empirical data would be used for the validation of the software tag.

5. Conclusion

We have shown the notion of software tag, and discussed on its application to software maintenance process.

Now we are finalizing the scheme of software tag Version 1.0 with various industrial collaborators of both software purchaser and developer sides. Also, tools to support empirical data collection, analysis, and abstraction are under designing.

Acknowledgement

This work is being conducted as a part of Stage Project, the Development of Next Generation IT Infrastructure, supported by Ministry of Education, Culture, Sports, Science and Technology. This work is the result of discussions with Koji Torii, Kenichi Matsumoto, Hajimu Iida, Shinji Kusumoto, Kazuko Matsumura, Shuji Morisaki, and many other project members.

References

- [1] Inoue, K., Software Tag: Empirical Software Engineering Data for Traceability and Transparency of Software Project, Workshop on Accountability and Traceability in Global Software Engineering (ATGSE2007), pp.35-36, Nagoya, Japan, 2007.
- [2] Ohira, M., et al., Empirical Project Monitor: A System for Managing Software Development Projects in Real Time, Proceedings of 3rd International Symposium on Empirical Software Engineering (ISESE2004), Vol.2, pp.37-38, Redondo Beach, CA, 2004.

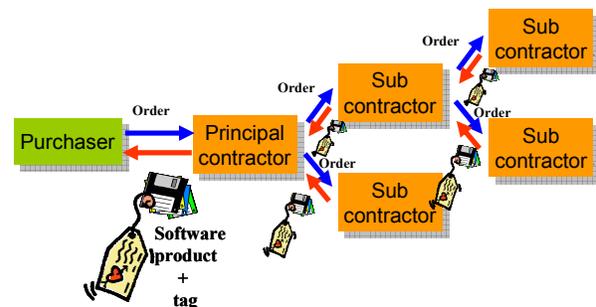


Fig.2 Hierarchy of sub-contract and tag chain