

Software Tag: Empirical Software Engineering Data for Traceability and Transparency of Software Project

Katsuro Inoue

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

Abstract

In this paper, we propose a scheme, named Software Tag, of software trade and development for improvement of traceability and transparency. Empirical data is collected during development, and processed into two types, open tag and secret tag, composing a software tag which is finally delivered to software purchaser.

1. Introduction

Software systems are becoming indispensable infrastructure, and failures of software systems cause serious damage to our everyday life. *Empirical Software Engineering* is an area of Software Engineering researches and practices, in which empirical data on the software development are collected and evaluated to get a feedback and to improve efficiency and quality of the project. Using the notion of empirical software engineering, we are going to construct a framework of *Software Tag* which is a package of information of software development and software products.

In this paper, we will explore the notion of software tag, and propose a framework of the software tag where software purchasers can measure and evaluate quality of software product and development process.

2. Overview of Software Tag

Figure 1 shows an overview of the concept of software tag. A software tag is a collection of empirical data and their abstracted data. It is mainly composed of "open tag" and "secret tag".

The open tag contains the data directly presented to the software purchaser. The software purchaser can evaluate the data, and know the quality of the process and product.

The secret tag mainly contains raw empirical data which is proprietary to the software developers. It may

contain development secret and/or know-how; therefore, the data is encrypted not to be seen by many other people. The decryption key would be kept at the developer side until a controversy arises between the software purchaser and developer.

2. Usecase of Software Tag

A scenario of using the software tag is summarized 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. The software developer agrees with this contract.
3. Software developer starts developing software product. During 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 of such empirical data. Not only the final status of those data at the end of the project, but also periodical snapshots of those data during development are collected and archived as the process and progress data of the project.
4. Some of the collected data are selected and transformed into the open tag, so that the software purchaser can easily understand and evaluate the project quality. Also, by this transformation, proprietary development know-how is eliminated.
5. Most of the archived empirical data are encrypted as the secret tag.
6. By merging the open tag and secret tag, a software tag is composed. This software tag is delivered to the software purchaser associated with the final software product. Also the secret tag is kept at a software liability mediation center where the controversy of software product quality is resolved in case.

7. The software purchaser evaluates the achievement of the software developer by viewing the open tag, and accepts the delivered software product.
8. When a controversy such as quality issue of the product happens between the software purchaser and developer, the secret tag kept at the software liability mediation center is decrypted, and the original empirical data is restored. By analyzing the empirical data, resolution of the issue is explored. Also correctness of the open tag is validated.

4. Applications of Software Tag

Software tag is a key vehicle to improve "visibility" and "transparency" of software project. This is very useful both for the software purchaser and the software developer.

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.

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.

5. Conclusion

We have proposed notion of software tag, and showed the overview of trade framework using the software tag. Now we are going to define the scheme of software tag in a more rigorous way, together 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, Shuji Morisaki, and many other project members.

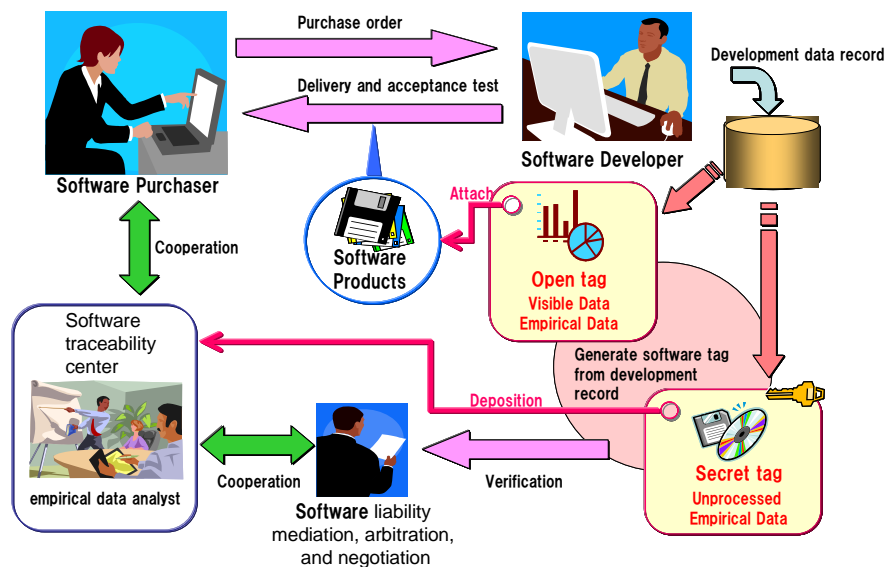


Fig. 1 Overview of Software Tag Scheme