INFORMATION
SCIENCE
TECHNICAL
REPORT

NAIST-IS-TR2010001
ISSN 0919-9527

Standardizing the Software Tag in Japan for Transparency of Development

Masateru Tsunoda, Tomoko Matsumura,
Hajimu Iida, Kozo Kubo, Shinji Kusumoto,
Katsuro Inoue, Ken-ichi Matsumoto

January 2010

NAIST

〒 630-0192
奈良県生駒市高山町 8916-5
奈良先端科学技術大学院大学
情報科学研究科

Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan

Standardizing the *Software Tag* in Japan for Transparency of Development

Masateru Tsunoda†, Tomoko Matsumura†, Hajimu Iida†, Kozo Kubo‡,
Shinji Kusumoto††, Katsuro Inoue††, Ken-ichi Matsumoto†

†Graduate School of Information Science, Nara Institute of Science and Technology
{masate-t@is, tomoko-m@is, iida@itc, matumoto@is}.naist.jp

‡Research Center for Advanced Science and Technology, Nara Institute of Science and Technology; kubo@rsc.naist.jp

††Graduate School of Information Science and Technology, Osaka University; {kusumoto, inoue}@ist.osaka-u.ac.jp

ABSTRACT
In this paper, we describe the *Software Tag* which makes software development visible to software purchasers (users). A software tag is a partial set of empirical data about a software development project shared between the purchaser and developer. The purchaser uses the software tag to evaluate the software project, allowing them to recognize the quality level of the processes and products involved. With Japanese government support, we have successfully standardized the software tag named *Software Tag Standard* 1.0, and have developed various associated tools for tag data collection and visualization. For its initial evaluation, the software tag has been applied to several projects. This paper also presents various activities aimed at promoting the use of the software tag in Japan and the world.

Categories and Subject Descriptors
D.2.8 [**Software Engineering**]: Metrics – *process metrics, product metrics.*

General Terms
Management, Measurement.

Keywords
Information sharing, empirical data, project management, offshore development.

1. INTRODUCTION
Software systems are becoming huge and complex, with our everyday life heavily dependent on such software systems. One of the major concerns of software purchasers (users) in Japan is the quality of the software systems. Japanese society generally demands high-quality software systems with low fault rates and high operability levels.

On the other hand, many software purchasers in Japan are not knowledgeable about the nature of software. It is reported that only 40% of Japanese major companies employ a full-time Chief Information Officer (CIO) and that only 20% of all CIOs are confident of their knowledge about information technologies [7].

Without a sufficient understanding of software quality and software projects, many companies try to purchase software systems from software developers (vendors). This produces a very risky situation. For example, purchasers cannot specify system requirements very well, and they do not oversee the project properly. Such situations often lead to project failures. It is reported that only 31.1% of software projects are recognized as 'successful projects' in Japan [8]. To confront these issues, there is strong demand to provide transparency of software projects to the software purchaser and improve communications between purchaser and developer.

The *Software Tag* is a new scheme to provide information feedback about the project from the developer to the purchaser. It establishes transparency of the software development project by allowing purchasers to view and analyze the elements of the tag. It also provides support for quantitative and qualitative communications between stakeholders. The Software Traceability and Accountability for Global Software Engineering (*StagE*) project [1] is a government-supported project that pursues standardization and promotion of the software tag scheme. In this project, we have defined the detailed structure of the software tag and developed various support tools. The software tag has been applied to real projects of major Japanese organizations. Along with technical development, we have also started various promotion activities, such as formal standardization of the software tag in both domestic and international standards, and exploration of new trade laws for software using the software tag scheme.

An early concept of how software tags could be used for software maintenance was shown in [4]. In this paper, we mainly explain use of the software tag for software development, together with activities and outcomes from the StagE project. In section 2, we describe an overview of the software tag scheme, and in section 3 explain the details of the software tag structure. In section 4, we describe activities of the project. In section 5 we provide some discussion, while in section 6 we outline conclusions and future research topics.

2. OVERVIEW OF THE SOFTWARE TAG SCHEME
A software tag is a packaged data set about a software project. It is currently composed of 41 characteristic elements of project data and progress data, as defined in section 3.1. Figure 1 shows an overview of the software tag scheme.

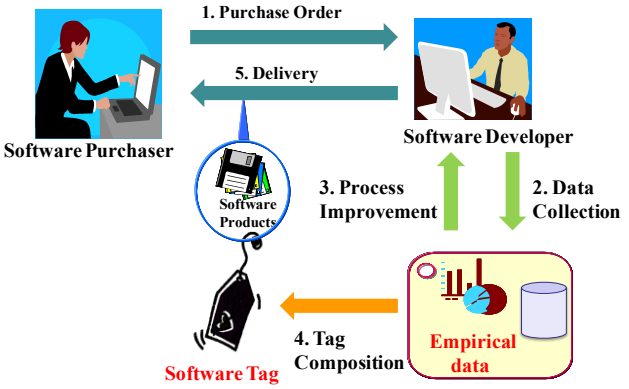

Figure 1. Overview of Software Tag Scheme

1. A software purchaser orders development of a software system. The purchaser includes both the final products and the software tag in their requirements.

2. During software development, various kinds of empirical data are created and generated. For example, requirements documents, software design documents, source code, test cases, issue tracking logs, manual documents, review logs, and quality analysis records may be produced. These are collected and archived. Note that we collect not only the final data at the end, but also interim snapshot data during development.

3. The collected data is analyzed for process improvement of the development organization, as is the usual process improvement scheme for software development organizations.

4. The collected data is used to construct the software tag. Parts of the empirical data are selected and abstracted into the software tag format.

5. The software tag is delivered to the software purchaser periodically during the development and/or finally at the end of the development together with the final software product. The software purchaser evaluates the software development by viewing and analyzing the tag, and accepts the delivered software product.

If a controversy such as a question about the quality of the product occurs between the software purchaser and the developer, the delivered software tag and (if necessary) the empirical data are analyzed, providing a basis for exploring a resolution to the controversy.

The software tag is a key to improving *transparency* of software projects. By examining the software tag, the software purchaser can identify and understand the development process, which has been mostly hidden from the purchaser. The purchaser can evaluate the quality of the processes and products of the project.

For the software developer, the software tag is useful to prove that they have conducted the proper activities in the software project. Also, it can be used to trace the quality of the activities of sub-contractors and sub-sub-contractors... (such contracting chains are very popular in Japan).

This scheme can be very useful for offshore and global development, because transparency and traceability of software development can be established with a fairly low overhead for the developers.

Standardizing the software tag will help to establish a minimum baseline for project quality, and to improve negotiations over software development contracts. Evaluation of software products and projects based on the objective empirical data contained in the software tag will lead to more healthy use of software in society.

3. DEVELOPMENT OF SOFTWARE TAG TECHNOLOGIES

3.1 Software Tag Standard 1.0

We have defined the elements of the software tag as shown in Table 1, named *Software Tag Standard 1.0*. It is composed of 41 tag elements, which are categorized into *project information* and *progress information*. The project information depicts the overall sketch of the project with various basic pieces of information. The progress information provides qualitative and quantitative indices of project achievement with various measures of the development phases. The tag standard provides more precise explanations and example metrics for each tag element which are not presented here.

It is not mandatory to use all 41 elements in the software tag in all cases. The purchaser and the developer can negotiate and select elements to use. Also, they can discuss and determine the details of the metrics. For example, #19, Scale of Programming, might be agreed to be measured by lines of code without comments.

In this standard, we have included various kinds of information that are considered important to the purchasers. The overall structure should be simple for the purchaser to understand, so we have tried to keep it as simple as possible. Also, we have tried to keep in mind the balance of the tag elements. This standard does not include tag elements that are computable from other tag elements. There are a number of standards and reports such as SWEBOK, CMMI, ISO/IEC 15939, and reports by the Software Engineering Center in Japan (SEC) which can help interpret the tag elements.

The definition process was based on discussions with industry and academic collaborators such as:

Purchasers: Tokyo Stock Exchange, Japan Aerospace Exploration Agency, DENSO.
Developers: Fujitsu Lab, Hitachi, NEC, SHARP, SRA Key-Tech Lab, Toshiba, NTT Data.
Others: Information Technology Promotion Agency, Ministry of Economy, Trade and Industry, Japan (IPA), Nara Institute of Science and Technology, Osaka University.

3.2 Support Tools

We are developing various support tools to promote the software tags scheme. In this paper, we introduce two essential tool prototypes that have been created for collection and visualization of the software tag.

(1) Software tag data collection tool (*CollectTag*)

CollectTag supports collection of empirical data from software projects and creation of a software tag. CollectTag uses a wizard as a user interface, allowing the user (developer) to easily input the necessary data for the software tag.

For each project, the purchasers and developers determine the metrics for the tag elements. To provide generality, we implemented CollectTag as a translator that converts a set of

Table 1. Software Tag Standard 1.0

Classification	Category	No.	Tag Element	Explanation
Project Information	Basic Information	1	**Project Name**	Unique name of project
		2	**Organization**	Information of development organization
		3	**Project Information**	Information needed to identify the project characteristics
		4	**Customer Information**	Information identifying the purchaser or owner
	System Information	5	**System Configuration**	Information identifying system configuration to label the type of system
		6	**System Scale**	Development system scale
	Development Information	7	**Development Approach**	Development process type or techniques
		8	**Organizational Structure**	Structure of development organization
		9	**Project Duration**	Information of development length
	Project Organization	10	**Super-Project Information**	Name of super project which creates this project
		11	**Sub-Project Information**	Name of sub projects which is created by this project
	Other	12	**Special Notes**	Other necessary or useful data for interpreting or analyzing tag data
Progress Information	Requirements	13	**User Hearing Information**	Information of user-requirements hearing
		14	**Scale**	Amount of requirements
		15	**Revisions**	Amount of changed requirement
	Design	16	**Scale**	Amount of design products
		17	**Revisions**	Amount of changed design
		18	**Design Coverage by Requirements**	Implementation ratio of design for requirements
	Programming	19	**Scale**	Amount of programming products
		20	**Revisions**	Amount of changed programs
		21	**Complexity**	Complexity of programs
	Test	22	**Scale**	Amount of testing
		23	**Revisions**	Amount of changed test
		24	**Density**	Ratio of test to system size
		25	**Progress Status**	Test progress to plan
	Quality	26	**Review Status**	Quantity information of review
		27	**Review Density**	Ratio of review to system size
		28	**Review Effectiveness**	Ration of found defects to amount of review
		29	**Defect Count**	Number of defects found by test
		30	**Fixed Defect Count**	Number of fixed defects
		31	**Defect Density**	Ratio of defects to system size
		32	**Defect Detection Rate**	Ratio of detected defects to consumed test
		33	**Static Check Results**	Report of static checker
	Development Cost	34	**Overall Cost**	Development and maintenance cost
		35	**Productivity**	Ratio of amount of products to overall cost
	Schedule and Management	36	**Process Management**	Information on management of development process
		37	**Purchaser-Developer Meeting Status**	Amount of user-vendor communication
		38	**Total Risk Item Count**	Number of risk items in the development
		39	**Risk Item Existence Period**	Time length between a risk item creation and deletion
	Other Products	40	**Scale**	Amount of product metrics not listed above
		41	**Revisions**	Amount of change in products not listed above

empirical data provided by the developer into the standard software tag format. That is, the developer periodically inputs values for each tag element, and then CollectTag outputs a software tag.

To reduce the effort of data input, CollectTag provides automatic data collection mechanisms for 11 of the tag elements in the progress information, if the target project uses common software development tools for configuration management and bug tracking. For example, LOC (#19: Scale) and CK (#21: Complexity) metrics [3] can be automatically collected and calculated from configuration management tools such as CVS or Subversion.

Finally, CollectTag generates the software tag elements in XML format (named *standard software tag format*). This makes it easy to provide the output to other visualization and analysis tools for further processing.

(2) Software tag visualization tool (*TagReplayer*)

TagReplayer provides fundamental features for integrated visualization of various historical data included in the software tag. TagReplayer employs the metaphor of video player manipulation for its user interface so that users can replay the progress of the project just like watching video on TV. Users can also instantly recall the details of any points along the timeline based on the software tag as shown in Figure 2. Our experience shows that this feature is very useful for postmortem project reviews.

TagReplayer aligns progress information from the software tag as a series of events. As a project summary, it displays multiple views including line charts, progress bars, and plain lists. For more details, clicking the items or points in the summary view can instantly recall empirical data such as problem descriptions or communicated messages. The tool also provides natural text mining and clustering that is useful for deeper analysis of human activity records, such as the problem reports or developers' communication messages, if the information is associated with the software tag.

3.3 Applications of the Software Tag

We present here three case studies of application of the software tags scheme to real software projects.

(1) A course registration system for a university with 26K LOC in Java was developed for five months by a medium-sized software company in Japan. 32 elements of tag data were collected and used to analyze the project status, *e.g.*, refactoring and the probability of insufficient testing density, by comparing the element values with the publicly available benchmark values from the Software Engineering Center in Japan (SEC).

(2) A medium-sized stock exchange system for a stock market was enhanced by a Japanese major software development company for more than two years. Using tag elements related to the requirements phase such as requirements revisions (#15), defect count (#29), and review status (#26), the purchaser and developer were able to identify problems with the requirements completeness caused by frequent changes.

(3) A Japanese software development company ordered several small-sized projects such as development of a project management support system from various offshore companies in China and Korea. Although remotely located from each other, the purchasers and developers could understand the progress of the specifications using tag elements such as the review status (#26), user hearing information (#13), and defect count (#29).

4. ACTIVITIES FOR PROMOTION AND DIFFUSION

The StagE project is also actively promoting and diffusing the software tags scheme in industry as follows.

(1) International/Domestic Standardization

Interviews with several Japanese software purchasers and developers, along with offshore software developers for Japanese companies in some countries, convinced us that most software purchasers and developers would strongly demand that the software tag and tools should be international and/or domestic technical standards in software engineering. To support this, we

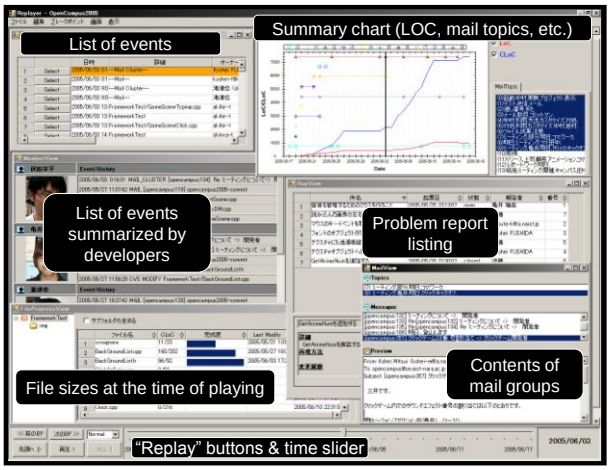

Figure 2. TagReplayer Screenshot

are now serving ISO as a committee member of the working group on process assessment, ISO/ITC JTC1/SC7/WG10. We are also working with some software tool developers to construct a de facto standard software project management system that includes the tag support tools mentioned in Sec. 3.2.

(2) International Collaboration

Offshore software development is one of the most useful application areas of the software tag scheme. To encourage and accelerate international collaboration to have various kinds of case studies and experiments of the software tag in offshore software development, we established Asia-Pacific Software Engineering Research Network (APSERN) in 2008 with software engineering researchers in NICTA (National ICT Australia), ISCAS (The Institute of Software, Chinese Academy of Sciences), and so on.

(3) Professional discussion of Legal issues

In the case of a legal dispute between software purchasers and developers, the software tag can clarify their liabilities and has the potential to help resolve such legal issues in software development. The StagE project has a committee examining the legal issues of software development. Members of this committee include lawyers, patent attorneys, and software engineers. The committee has interviewed many software developers in Japan and China to compile data about troubles between software purchasers and developers. It also distributed questionnaires to more than a hundred software developers in Japan to analyze the trends of such troubles. The software tag provides an opportunity for collaboration between software engineering and software trade law.

5. DISCUSSION

(1) There are metrics repositories aimed at improving and benchmarking development organizations [5], along with some software measurement paradigms [2] [6]. However, the software tag provides a unique approach to involve software purchasers in the quality improvement framework by providing development transparency. As far as we know, there is no similar approach presented in the technical literature.

(2) We believe that the benefits of the software tag scheme for software purchasers will be substantial because the development processes and the developed products become more visible and understandable. However, purchasers will need to collaborate more closely with developers, providing effort and enthusiasm to create successful projects.

(3) We have presented the first standard of the software tag with 41 elements. In some sense, these are very basic data for indicating development quality, and they may be insufficient to perform detailed analysis. However, as a standard used for various software development projects, the set should be minimal and low cost. As presented in Sec. 3.1 and 3.2, our tag standard 1.0 is a lightweight set with low collection and assembly cost. It is important to continue practical applications of the software tag, and to get feedback for further improvement of the standard.

6. CONCLUSIONS

We have introduced our activities for standardization of the software tag in Japan. Through these activities, the concept of the software tag is becoming well understood in Japan.

Our future work will focus on making international/domestic standards of the software tag. With such standardization, the software tag is expected to be used in various software industries, where we think it will strongly promote participation and understanding of software development by purchasers.

7. ACKNOWLEDGMENTS

This work is being conducted as a part of the StagE project, The Development of Next-Generation IT Infrastructure, supported by the Ministry of Education, Culture, Sports, Science and Technology. We are grateful to the members of the StagE project, especially to Michael Barker, Akito Monden, Makoto Matsushita, and Shuji Morisaki.

8. REFERENCES

[1] Barker, M., Matsumoto, M., and Inoue, K. 2008. Putting a TAG on Software: Purchaser-Centered Software Engineering. In Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization, Ramachandran, M., and Carvalho R. A., Eds. Information Science Reference, Hershey, PA, 38-48.

[2] Basili, V.R., and Rombach, H.D. 1988. The TAME Project: Towards Improvement-Oriented Software Environments. IEEE Trans. Softw. Eng. 14, 6 (June 1988), 758-773.

[3] Chidamber, S. R., and Kemerer, C. F. 1994. A Metrics Suite for Object Oriented Design. IEEE Trans. Softw. Eng. 20, 6 (June 1994), 476-493.

[4] Inoue, K. 2008. Software Tag for Traceability and Transparency of Maintenance. In Proceedings of, 24th IEEE International Conference on Software Maintenance (Beijing, China, Sep. 28 - Oct. 4, 2008). ICSM 2008, 476-477.

[5] International Software Benchmarking Standards Group (ISBSG) 2004. ISBSG Estimating: Benchmarking and research suite.

[6] Kitchenham, B. A. Hughes, R. T., and Linkman, S. G. 2001. Modeling Software Measurement Data. IEEE Trans. Softw. Eng. 27, 9 (Sep. 2001), 788-804.

[7] Nikkei Business Publications, Inc. 2008. Survey Report on IT Investment and CTO in Japan. Nikkei Information Strategy. March (in Japanese).

[8] Nikkei Business Publications, Inc. 2008. Second Survey of Japanese Software Projects. Nikkei Computer. Dec. 1, 36-53 (in Japanese).