Toward Optimized Collection and Visualization of Software Metrics for Progress Sharing in Offshore Software Development Project

Shuji Morisaki and Ken-ichi Matsumoto
Graduate School of Information Science, Nara Institute of Science and Technology Japan
smrs@is.naist.jp, matumoto@is.naist.jp

Abstract

In order to share progress of software development between software supplier and purchaser in ordered and dedicated software development such as system integration, time series of software metrics are often included in progress report. However, desirable granularity and tradeoffs between cost and benefit of measurement needs more discussion. The authors would like to discuss cost and merit of collection and sharing time series transition of software metrics including size metrics in order to share progress of software development.

1. Introduction

In distributed software development, low visibility of development progress easily leads to delivery slippage or low software quality. A number of methods and tools to collect and visualize product metrics [1], change histories [4], time series software size metrics [3] and process quality [2] in distant site have been proposed. However, collection and visualization granularity of those metrics and information is not sufficiently discussed especially distributed and offshore software development project.

In this paper, we show a visualization of time series of software size metric in different granularity as an example. Based on the example, the authors would like to discuss desirable software metrics granularity of collection, visualization and analysis toward optimized granularity.

2. Effort for progress sharing and visibility

Sharing progress of software development between purchaser and supplier enables to take actions for avoiding risk of low quality or delivery delay. However, in most cases, progress sharing doesn't move forward software development. In other words, effort of progress sharing should be minimized and visibility should be maximized. Figure 1 depicts a relationship between visibility of progress and progress sharing cost under assumption of manual or semi-automatic progress reporting. Progress report is usually prepared by software developers (supplier). Prepared progress report is understood by software user (purchaser). Too much progress information may cost high because increased information requires more effort for preparing and understanding. Contrary, with insufficient information, purchaser and supplier can not share progress and can not take actions avoiding risk and issues.

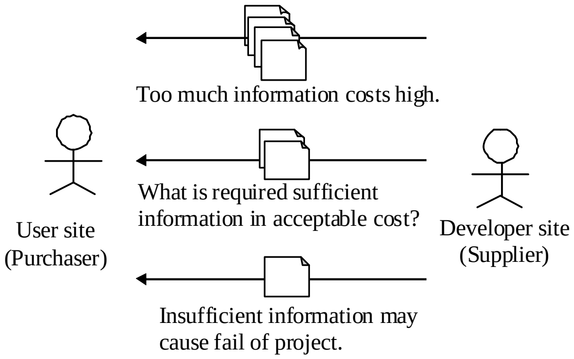
Too much information costs high.
What is required sufficient information in acceptable cost?
User site (Purchaser)
Developer site (Supplier)
Insufficient information may cause fail of project.

Fig. 1 Cost and benefit of progress information

3. An example of progress sharing

The authors select time series transition of software size metrics (LOC) as progress sharing information. The authors assume that progress report includes the transition and supplemental descriptions such as explanation in written or verbal natural language.

Fig. 2 and Fig. 3 shows time series transition of software size metrics collected during a software development project. In Fig. 2 and 3, horizontal axis represents date and vertical axis represents software size metrics (LOC). Triangles on line plot represent inspection meeting at supplier site. Squares on line plot represent testing progress meeting at supplier site. Fig. 2 visualizes measured LOC of entire software. Fig. 3 visualizes measured LOC of each functions of the software. In Fig. 3 each line corresponds to a function of the software.

The effort for preparing Fig. 2 and its supplemental description is smaller than the effort for Fig. 3. On the other hand, the visibility of progress provided by Fig. 2 is smaller than that of Fig. 3.

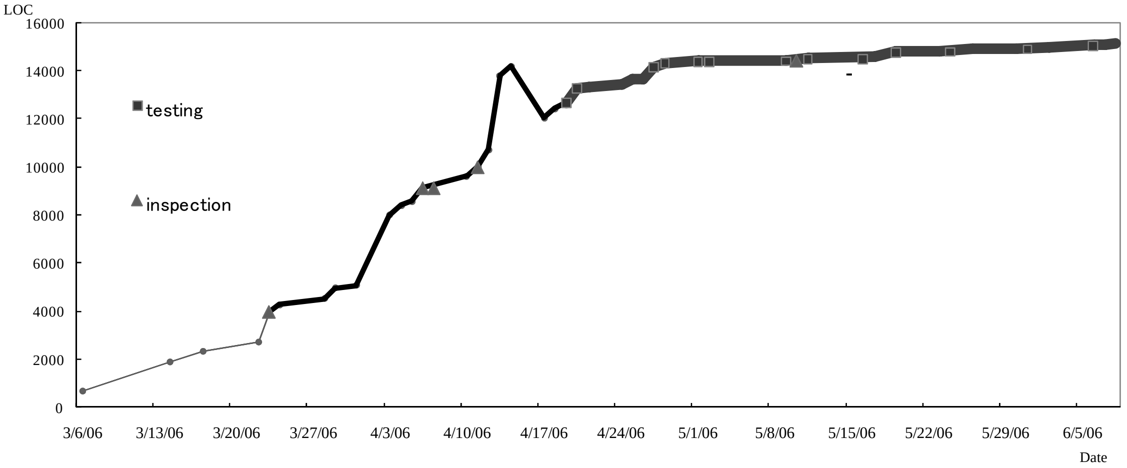

Fig. 2 Time series transition of source lines of code of entire software

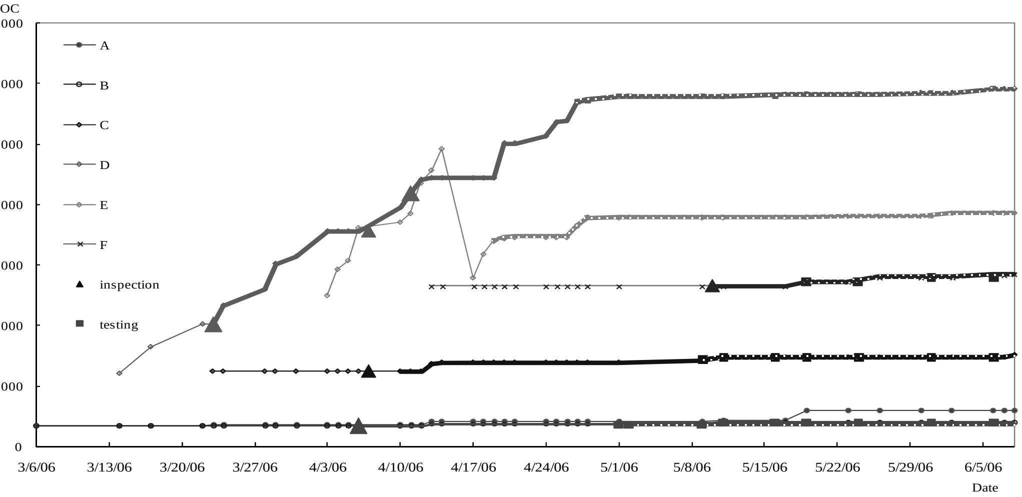

Fig. 3 Time series transition of source lines of code of 6 sub-functions

4. Conclusion

In order to share progress of software development between software supplier and purchaser in ordered and dedicated software development such as system integration, time series transition of software metrics are often included in progress report. We stated problem of preparing and understanding effort and visibility of progress are problem. in such progress report. As an example, we show two granularity of time series transition of LOC.

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 of Japan.

1. Froehlich J. and Dourish P., “Unifying Artifacts and Activities in a Visual Tool for Distributed Software Development Teams,” Proceedings of International Conference on Software Engineering, pp.387-396 (2004)
2. Morisaki S. et al., “Fine-grained Software Process Analysis to Ongoing Distributed Software Development,” Proceedings of Workshop on Measurement-based Cockpits for Distributed Software and Systems Engineering, pp. 26-30. (Aug. 2007)
3. Ohira M., et al., “Empirical Project Monitor: A Tool for Mining Multiple Project Data,” Proceedings of International Workshop on Mining Software Repositories, pp. 42-46, (2004)
4. Zimmermann T., et al.., “Mining Version Histories to Guide Software Changes,” Proceedings of International Conference on Software Engineering, pp.563-572, (2004)