Analysis on Diversity and Similarity among Software Development Projects Using Empirical Project Monitor

Masao Ohira, Ken-ichi Matsumoto
Nara Institute of Science and Technology,
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
ohira@empirical.jp, matumoto@is.naist.jp

Abstract

We introduce the Empirical Project Monitor (EPM) which automatically collects and measures history data accumulated during everyday software development activities. EPM helps users (developers and managers) keep their projects under control in real time without additional work. By providing means for analyzing not only single project data but also multiple projects data, users are able to identify diversity and similarity among projects in order to reuse artifacts and knowledge across projects.

1. Introduction

In software development in recent years, the improvement of software process has increasingly gained attention. Project management for achieving effective software process improvement should be based on quantitative data. In general, data collection for software measurement requires high costs and additional effort from developers and managers [1].

This paper introduces the Empirical Project Monitor (EPM) [2, 3] which currently automatically collects and measures quantitative data from three kinds of repositories widely used in software development support systems such as configuration management systems, mailing list managers and issue tracking systems. By providing integrated measurement results graphically, EPM helps users (developers and managers) keep their projects under control in real time.

One of the features is that EPM dose not only deal with single project data but also multiple projects data. Changing parameters extracted from repositories, users are able to compare a target project with past projects from various aspects. This would be useful for reusing artifacts and knowledge from past projects and probably detecting a project which indicates unusual values.

2. Empirical Project Monitor (EPM)

Figure 1 shows an overview of EPM. EPM consists of four components: data collection, format translation, data store, and data analysis/visualization.

First, EPM automatically collects versioning histories from configuration management systems, mail archives from mailing list managers, and issue tracking records from issue tracking systems. Because these data are accumulated through everyday development activities, developers and managers do not need to do additional work for data collection. Second, EPM translates the collected data into the XML format, so that EPM can deal with not only the above three kinds of history data but also other kinds of data according to the purposes for measurement. Data from other systems can be included by small adjustments of data collector and format translator. Third, the data is imported into a PostgreSQL database. Finally, EPM analyzes the data stored in the database and then visualizes various measurement results. Developers and managers can view the results through common web browsers.

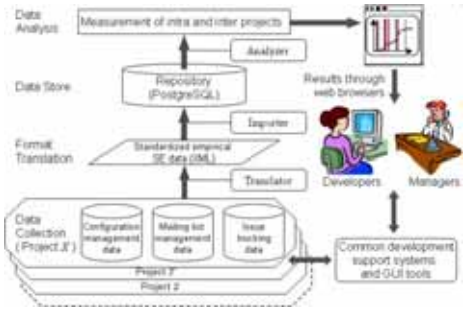

Figure 1. Empirical Project Monitor

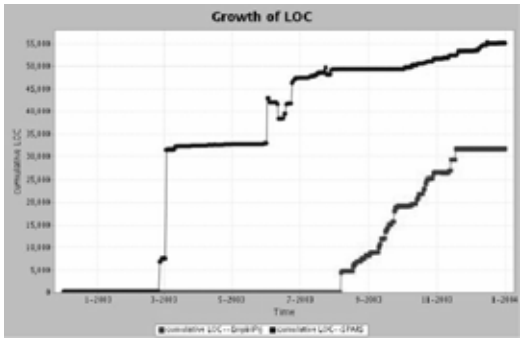

Figure 2. Comparison of two projects

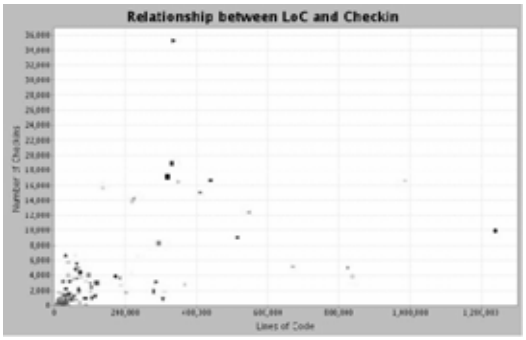

Figure 3. Distribution map

3. Measurement using EPM

EPM provides users with a variety of visualizations of measurement results based on the three kinds of history data currently available [2]. The features of EPM are to deal with data from multiple projects simultaneously and to visualize combinations of measurement results [3].

For instance, Figure 2 represents the relationship of the growth of lines of code between two projects[1]. Because a person who manages the both projects can grossly estimate the progress of the later project based on her/his experience from the earlier project, she/he would be able to quickly take steps to cope with problems in the later project if the graph is not up to her/his expectations. Such the graph could help people who have to manage multiple projects at the same time or for comparing productivity.

EPM can also generate a distribution map such as in Figure 5. The graph shows a distribution map of 100 projects[2], which represents the relationship between lines of code (X-axis) and number of checkins (Y-axis). Such a graph can help managers objectively identify "unusual" projects with extremely high or low values.

[1] Figure 2 shows the SPARS project (http://iip-lab.ics.es.osaka-u.ac.jp/SPARS/) and the EASE project (http://www.empirical.jp/).

[2] We selected the 100 projects in Figure 3 randomly from the list of "most active projects" in SourceForge.net.

4. Future work

Currently we are exploring to find metrics for identifying diversity and similarity among multiple software projects based on history data accumulated during everyday software development activities. Although the graphs mentioned above are calculated using data only from configuration management systems, EPM can visualize any combinations of three kinds of software repositories (e.g. using data from mailing list managers and issue trucking systems, EPM can also visualized the relationship between the number of bugs and mails).

Our study dose not focus on "micro" understanding of software development activities as seeing a particular project in detail, but focuses on "macro" understanding of activities of a number of projects in a large software company or a huge community such as SourceForge.net. We believe that this approach would be useful for efficiently reusing knowledge and software artifacts in a large organization and for managing software development process which is apt to rely on managers' experience and "intuition".

Acknowledgement

This work is supported by the Ministry of Education, Culture, Sports, Science and Technology of Japan, the Comprehensive Development of e-Society Foundation Software program and Grant 15103 of the Open Competition for the Development of Innovative Technology program.

Reference

[1] V. R. Basili and D. M. Weiss, "A methodology for collecting valid software engineering data", IEEE Transactions on Software Engineering, Vol.SE–10, No.6, pp.728–738, 1984.

[2] M. Ohira, R. Yokomori, M. Sakai, K. Matsumoto, K. Inoue, and K. Torii, "Empirical project monitor: Automatic data collection and analysis toward software process improvement", In Proceedings of 1st Workshop on Dependable Software System, pp.141–150, Tokyo, Japan, February, 2004.

[3] M. Ohira, R. Yokomori, M. Sakai, K. Matsumoto, K. Inoue, and K. Torii, "Empirical Project Monitor: A Tool for Mining Multiple Project Data", In Proceedings of International Workshop on Mining Software Repositories (MSR2004), pp.42–46, Scotland, UK, May, 2004.