In Proceedings of 5th ACM-IEEE International Symposium on Empirical Software Engineering, Vol. 2, pp. 15-17, Sep. 2006.

A Proposed Method for Building a Database of Project Measurements and Applying it Using Collaborative Filtering

Yoshiki Mitani
IPA/SEC, NAIST
ymitani@empirical.jp

Nahomi Kikuchi
IPA/SEC
n-kiku@ipa.go.jp

Tomoko Matsumura
NAIST
tomoko-m@empirical.jp

Naoki Ohsugi
NAIST
naoki-o@is.naist.jp

Akito Monden
NAIST
akito-m@is.naist.jp

Yoshiki Higo
Osaka Univ.
y-higo@ist.osaka-u.ac.jp

Katsuro Inoue
Osaka Univ.
inoue@ist.osaka-u.ac.jp

Mike Barker
NAIST
mbaker@is.naist.jp

Ken-ichi Matsumoto
NAIST
matumoto@is.naist.jp

ABSTRACT

This paper proposes a new method for developing predictions and estimates for on-going projects by comparing in-process measurements of the current project with benchmark data from previous projects. The method uses collaborative filtering to identify groups of similar projects in the benchmark database and then to develop predictions and estimates based on the in-process measurements of the current project and the comparison data from the similar projects.

1. INTRODUCTION

Missing or incomplete data often complicates analysis and predictions based on comparing measurements of software development projects with collected data from other projects. This report explores this problem and a potential solution from the authors' research.

The paper describes the Empirical Project Monitor (EPM), a project measurement system [1]. It is developed by the Empirical Approach to Software Engineering (EASE) project, an academic based project for collaboration between industry and academia. It presents experimental results from use of EPM and related tools in a governmental project with collaborative multi-vendor development [2]. It also describes a project aimed at collecting benchmark data from software projects that has collected data from over 1000 projects in 15 software projects. The Software Engineering Center (SEC) Japan, a new Japanese organization for collaborative industry-academia research and investigation, conducted this project. The paper also introduces a method for data analysis using collaborative filtering technology that has proven effective for data sets with missing elements [3][4].

This paper describes a general method for performing analysis and predictions of software development projects. Based on the described research, this method uses dynamic measurements of software process and a database of project measurements, along with collaborative filtering technology.

2. EPM: the in-process project measurement platform

EPM automatically collects software development management data from development tools such as a configuration management system, bug tracking system, and mailing list management system. Drawing especially from the configuration management system, EPM automatically collects source code and operational histories of source code development, the basic information concerning transitions that occur in the software development process. Fig. 1 shows an example of the EPM displays.

The EPM analysis functions display information in visual formats. The information includes changes in the source lines of code, timing analysis of check-in and check-out, changes in bug numbers, analysis of inter-company mail volumes, and the Software Reliability Growth Model (SRGM) curve.

By using EPM, a software project can receive the benefits of automatic measurements and visually presented analyses of project data without the burden of intrusive manual tracking.

3. Experience with in-process project measurement

3.1 Project description

The target project is a government funded middle scale experimental development project.

The project was organized as a development consortium, composed of seven companies, including six major software development companies. One of the six companies acts as Project Manager (PM). The six companies are rivals in target system field, so the project clearly distinguishes between collaboration and competitive materials. Information in the collaborative field is shared and in the competitive field is confidential. For example, detail design, source code, and source line of code (SLOC) productivity are confidential. However, the PM needs SLOC information for meaningful project management. Normally this situation would force the PM into a kind of blind management. During the companies' individual development phase, management would be based on declarations. Only in the inter-company integration test phase would all members share the real situation of the developed software.

Each consortium company measured project data, which was collected by SEC and analyzed for software engineering research. Analyzed data was fed back to the individual companies with respect for confidentiality. The PM was provided with a bird's-eye view of the total information, again with respect for confidentiality. This allowed more than the blind management that had been expected.

For this project, the following four methods of measurement were used:

1) EPM Measurement and Analysis

EPM collected development process and product information, and produced analysis results.

2) Collection and analysis of review reports

An electronic data form with 30 items was used to collect information concerning basic and detailed design reviews.

3) Code Clone Analysis

A code clone is a code fragment in source code which is identical or similar to each other. Code clone fingerprints such as code clone distribution or content ratio represent software product characteristics. In this trial, we used CCFinder [5], which is a code clone detection tool.

4) Collect project context information by participation in project meetings

To collect more data about the project context, research staff attended all the project meetings. This was very useful in collecting information that could not be collected in other ways.

3.2 Results of Project

Generally in consortium development projects such as this, the status of internal development for the individual companies is kept in confidence except for inclusive reports based on company declarations until the inter company integration test.

In this project, the measurement effort brought development out of the black box into the daylight and helped to form a consensus about how to handle project management. For example, the following characteristics of this project were identified:

- Measurement provided a bird's-eye view of each company's project based on transitions in source line of code count and transitions in bug numbers.
- Code clone analysis helped identify the origin of source code and the development approach taken by different groups. For example, this analysis helped identify whether they mostly used code developed from scratch, by cut and try methods, or appropriated and reused code. The analysis also suggested characteristics about the source code such as whether it was produced by a less experienced coder or not, possible issues with future code maintenance, and the status of the re-factoring process.
- Analysis of file renewal suggested differences in the development process, such as use of waterfall type process or cut-and-try development. This analysis clearly showed the stability of file renewal, the impact of design changes, and attention to bug detection in the late developed process.
- Analysis of bug reports showed clear relationships between various bug factors. In particular, analysis of relationships between the bug injection process and the bug detection process, along with consideration of when bugs should ideally be detected, were particularly useful in evaluating the early development process.
- The analysis of review reports clearly indicated the different attitudes towards the review process. Some companies invested significant effort in the review process, reducing problems in later stages of their waterfall development process. Other companies slighted the review process, expecting problems to be caught by later testing instead. We could expect some conflict on software parts from these different companies will be combined at the system integration test phase.

4. Post-process benchmark data collection

4.1 Benchmark data collection from over 1000 projects and building a national database

The SEC has started to collect software project benchmark data from industry to build a national level database. As the first step in this process, the SEC has collected data from 1009 projects.

The data items collected were defined by the SEC in reference to data items previously collected by Japanese software industries. The list contains about 490 items in 10 categories. Preliminary analysis has identified some useful database attributes such as program size, total effort, productivity, reliability, and some correlations between basic data items.

4.2 Collaborative filtering of the benchmark database

The collected database includes many data sets with missing elements, and various kinds of projects. To analyze them required a technology that can handle missing elements and perform grouping and categorization of the projects. Collaborative filtering technology was applied to group similar projects from data sets with missing elements. A key feature of collaborative filtering for this application is that it can analyze data sets directly without any special operation for missing elements in included data sets or any special variable selection.

For example, the collaborative filtering tool calculated a project similarity grade indicating project similarity in 10 steps from 0.0 to 1.0, which was used to illustrate a similarity distribution graph. In the target project trial, collaborative filtering using the partial benchmark data from five consortium companies at the end of the basic design phase retrieved about 70 similar projects in the similarity range from 0.9 to 1.0 from the 1009 projects in the SEC database. Fig. 2 shows a part of this graph as an example. Both manual review and some statistical processing allow extraction of useful information for project operation from the characteristics of the retrieved group of similar projects.

5. A proposed method for using in-process measurements

Accumulated data concerning the process and product form a valuable database after completion of projects. However, it is difficult for a project to use this information while conducting in-process measurements. Missing data and the difficulty of identifying similar projects make such databases hard to compare with the in-process measurements.

The authors propose to use in-process measurements of projects along with information from groups of similar projects extracted from the project benchmark database as described in 4.2. Fig. 2 illustrates this method, while the following describes the procedure. The numbers in parentheses correspond to those in Fig. 2:

1. Benchmark data as described in 4.1 and measurement data about the process and product as described in 3.1 are collected in a dataset (1)(2). This data is accumulated in a database (3)(4).
2. For a new project, interim benchmark data is collected and collaborative filtering used to identify a group of similar projects from the benchmark database (7)
3. Process and product measurements from similar projects (8) are used to generate estimates for the new project (9). The data, predictions based on the benchmark database (8) and in-process data measurements (5) are referenced to project operation in all (10).

One area for further research and investigation concerns the best way to use the process and product data from the projects identified as similar by the collaborative filtering tool. For example, when 70 projects of the 1009 projects in the SEC database were identified as similar above, the data included 70

sets of EPM collected and analyzed data graphs plus the output charts and metrics from the code clone analysis tool CCFinder. Some of the benchmark data can be utilized with simple statistical methods. For example, with 70 sets of comparison data, a weighted average calculation can provide a rough estimate of the project effort. However, information such as the EPM charts of process transitions and the CCFinder output plot are not easily aggregated using statistical techniques. In such cases, human review and observation may be useful in suggesting abstractions to guide project operation. As discussed in 3.2, in-process measurement of the project supports project operation by providing information about the project status and transitions. Comparison data from similar projects can provide additional understanding for project operation.

6. Conclusion

Software project measurement often provides two broad kinds of measurements, post-process collections of benchmark data and in-process measurements of process and product. However, combining the in-process measurements with benchmark data to provide estimates and guidance for projects has been difficult due to problems identifying similar projects and handling missing data.

The authors' research has demonstrated one method for combining the in-process measurements and the benchmark database. The method provides predictions or estimates for projects by applying collaborative filtering to retrieve groups of similar projects from the benchmark database using the interim measurements of the current project as the retrieval key.

7. Acknowledgements

This work is supported by IPA/SEC, METI and the MEXT of Japan, the Comprehensive Development of e-Society Foundation Software program. We thank researchers in the SEC and EASE project who kindly support our project.

8. REFERENCES

[1] M. Ohira, R. Yokomori, M. Sakai, K. Matsumoto, K. Inoue, M. Barker, K. Torii: Empirical Project Monitor: A System for Managing Software Development Projects in Real Time, *3rd Intl. Symposium on Empirical Soft. Engineering (ISESE) 2004*, *Vol.2*, pp.37-38

[2] Y. Mitani, N. Kikuchi, T. Matsumura, S. Iwamura, M. Barker, K. Matsumoto: An empirical trial of multi-dimensional in-process measurement and feedback on a governmental multi-vendor software project. *4 th(ISESE) 2005*, *Vol.2*, pp.5-8

[3] N. Kikuchi: Experience from Analysis of 1000 Collected Project Data, *International Workshop on Future Software Technology (IWFST) 2005.*

[4] N. Ohsugi, M. Tsunoda, A. Monden, and K. Matsumoto: Effort Estimation Based on Collaborative Filtering, *Proc. of the 5th Intl. Conf. on Product Focused Soft. Process Improvement (PROFES) 2004*, pp.274-286.

[5] T. Kamiya, S. Kusumoto, K. Inoue: CCFinder: A Multi-Linguistic Token-based Code Clone Detection System for Large Scale Source Code. IEEE TSE 28 (2002)pp.654-670

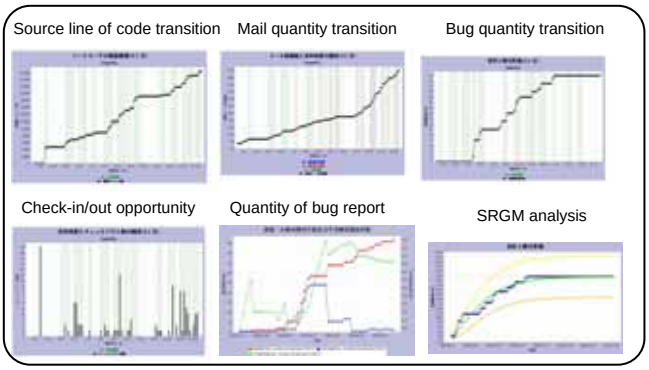

Fig.1 Empirical Project Monitor (EPM) Display

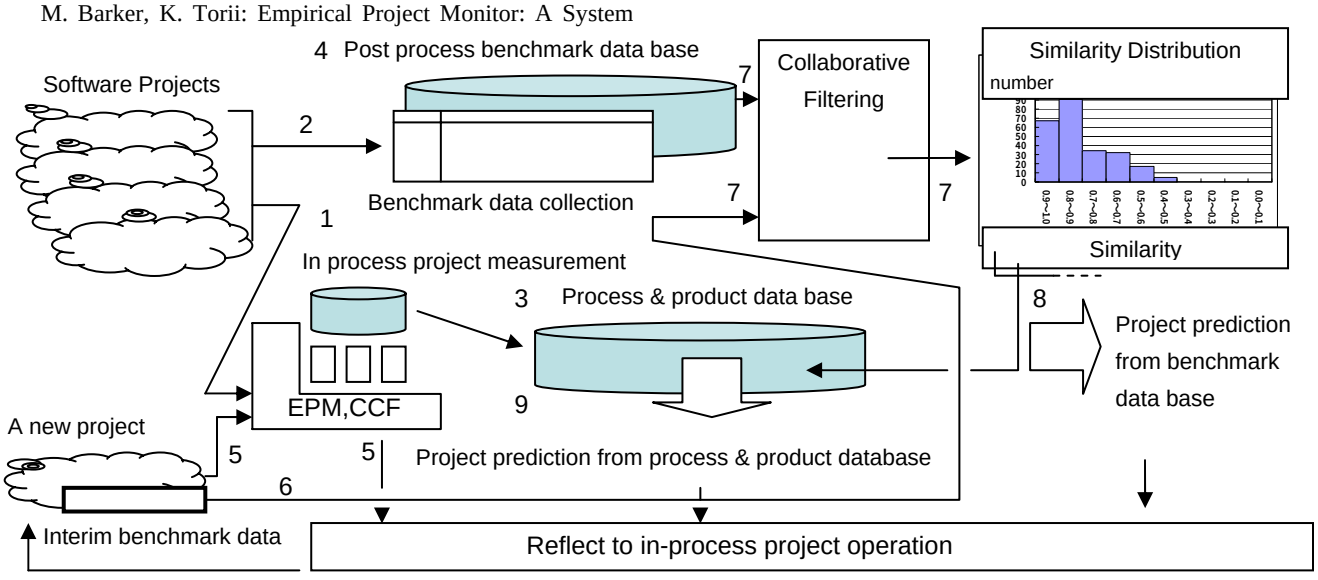

Fig.2 Project prediction by collaborative filtering with two kind of project database