

EASE Project: Introducing Empirical Software Engineering into Japanese Industry

Naoki Ohsugi

Graduate School of Information Science, Nara Institute of Science and Technology

Kansai Science City, 630-0192 Japan

naoki-o@is.naist.jp

Abstract

In Japanese software development industry, there are few activities to objectively evaluate effectiveness or degree of improvement. The author believes empirical software engineering can solve this problem. This paper introduces an ongoing national project named EASE (Empirical Approach to Software Engineering). EASE project aims to introduce empirical approach into Japanese industry. This paper also explains Empirical Project Monitor, a tool for data collection and analysis.

Discussion Topic: Empirical Software Engineering

1. Introduction

Japanese software development industry has been searching methods and technologies to improve their development productivity and their products' quality. Companies have actively tried to use them in their businesses. For example, process improvement programs such as Capability Maturity Model (CMMI) [1]; agile methods such as eXtreme Programming (XP) [2]; project management methods based on Project Management Body of Knowledge (PMBok) [9]; and new implementation technologies such as Java and C#. Some practitioners voluntarily established user groups of the methods and technologies for learning and introducing them into the industry.

However, there are few activities to objectively evaluate effectiveness or degree of improvement. There are lacks of discussion about the following topics; appropriate quantitative metrics for evaluating the improvement; procedure and tools for measuring these metrics; and, analysis methods of measured metrics. For example, when a company achieves CMMI capability maturity level 3, what are improved things with the process improvement program? If a group employs practices of XP, improvement effect of products' quality is how large? Management skills of PMBoK users and seat-of-the-pants persons is how different? Few people can answer these questions with objective evidences observed in real development field.

The author believes empirical software engineering (SE) can solve this problem. Empirical SE is concerned with the scientific measurement, both quantitative and qualitative, of software engineering process and product [6]. Establishment of framework to collect and analyze measured data (the so-called empirical data) allows practitioners to evaluate their improvement activities. For example, there were some success cases with empirical SE in cost modeling and estimation research

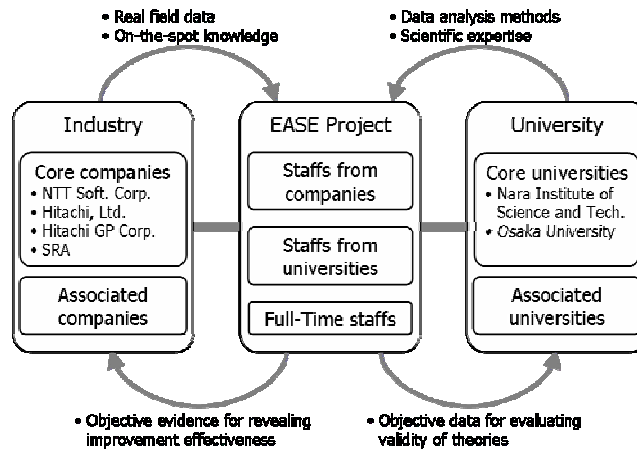


Figure 1. Interactions among stakeholders of EASE project

area [3], [10].

This paper introduces an ongoing Japanese national project named EASE (Empirical Approach to Software Engineering). EASE project aims to introduce empirical SE into Japanese industry. In addition, this paper explains Empirical Project Monitor (EPM). EPM has been developed as a main product of the project, for collecting and analyzing empirical data. Section 2 describes an outline of EASE project. Chapter 3 summarizes features and usage of EPM. Chapter 4 gives outlooks on near future activities.

2. EASE project

EASE project is based on the cooperation among industry, academia and government. MEXT, the Japanese Ministry of Education, Culture, Sports, Science and Technology is funding the project. Four companies and two universities are providing human resources playing the central roles in the project. These staffs share workspace everyday for accelerating discussions and teamwork. In addition, many associated companies and universities have been involved at some parts of the project.

Originally, stakeholders of EASE project were not close to each other. They existed in industry and academia separately as shown in Figure 1. EASE project takes place between industry and academia to act as a bridge between them. Industry side stakeholders provide real field data and on-the-spot knowledge to the project members. They get objective evidence for evaluating effectiveness of their improvement activities, in return for a contribution. On the other hand, academia side stakeholders provide data analysis methods and scientific expertise to the projects. They get empirical data for observing and understanding phenomena.

Activities of the project are classified into the following five categories:

CI. Developing and distributing EPM, a tool for collecting quantitative data.

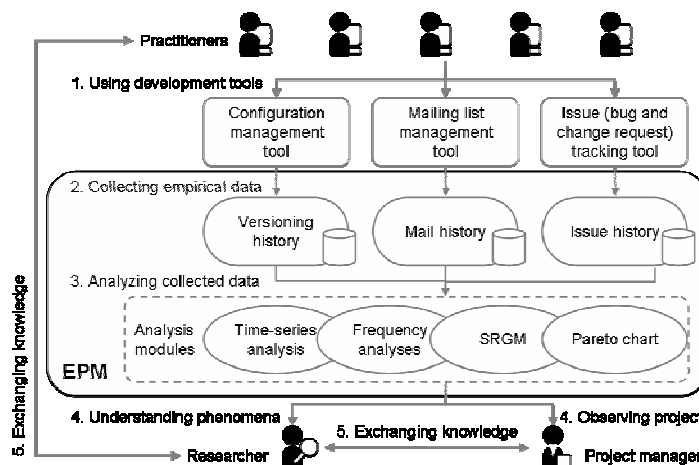


Figure 2. Process of data collection and analysis using EPM

- C2. Collecting empirical data from real software development project.
- C3. Analyzing collected data for observing and understanding phenomena.
- C4. Organizing workshops and study meetings for exchanging knowledge among practitioners and researchers.
- C5. Writing research papers for publishing observed phenomena and lessons learned.

3. EPM: Empirical Project Monitor

EPM is a tool for collecting and analyzing empirical data. Process of data collection and analysis is shown as Figure 2. It is:

1. Practitioners use development tools for configuration management, mailing list management, and issue tracking.
2. EPM collects and stores usage histories of the tools as empirical data.
3. EPM analyzes the collected data with analysis modules such as time-series analysis and frequency analysis, Software Reliability Growth Model (SRGM) and Pareto chart [7].
4. Researchers and project managers see the result of analysis with visualization feature of EPM.
5. Earned knowledge is exchanged among the researchers, project managers and practitioners.

EPM has been developed by industry side staffs of EASE project as open source software^{*1}. EPM developers have been carefully implementing EPM to make it easy to use in real software development. EPM is transparent to the practitioners because it runs as a background process. In fact, some practitioners were not aware of existence of EPM because a project manager did not tell them existence of EPM. Data were automatically collected without any troubles from the practitioners' daily work.

^{*1} <http://empirical.jp/~epm/>

4. Future work

Currently, project members are exploring helpful analysis methods using collected empirical data. The data were collected from two large Japanese software development companies. These companies used EPM on trial in their businesses, and requested to put more many helpful analysis modules to EPM. Academia side staffs of EASE project have developed analysis methods such as logical coupling [4], code clone analysis [5] and collaborative filtering [8]. Usefulness of these methods will be experimentally evaluated using the collected data. Only methods which can pass strict review of all stakeholders will be implemented as analysis modules of EPM. Those modules must make EPM more useful. Of course, those modules will be available as open source software.

Acknowledgement

This work is supported by the EASE project^{*2}, the Comprehensive Development of e-Society Foundation Software program of the Ministry of Education, Culture, Sports, Science and Technology of Japan. I would like to thank to the practitioners of Hitachi Government and Public Corporation System Engineering Ltd. and Mitsubishi Space Software Co., Ltd. for contributing to data collection with EPM.

References

- [1] Ahern, D.M., Clouse, A., and Turner, R., "CMMI Distilled: A Practical Introduction to Integrated Process Improvement," Addison-Wesley, 2nd Edition, 2003.
- [2] Beck, K., Andres, C., "Extreme Programming Explained: Embrace Change (XP)," Addison-Wesley, 2nd Edition, 2004.
- [3] Boehm, B.W., "Software Engineering Economics. Englewood Cliffs," Prentice-Hall, 1981.
- [4] Gall, H., Jazayeri, M., and Krajewski, J., "CVS Release History Data for Detecting Logical Couplings," In *Proc. of the 6th Intl. Workshop on Principals of Software Evolution (IWPSE'03)*, pp.13-23, 2003.
- [5] Higo, Y., Kamiya, T., Kusumoto, S., and Inoue, K., "Refactoring Support Based on Code Clone Analysis," In *Proc. of the 5th Intl. Conf. on Product Focused Soft. Process Improvement (PROFES2004)*, pp.220-233, 2004.
- [6] Jeffery, D. R., and Scott, L., "Has Twenty-five Years of Empirical Software Engineering Made a Difference?," In *Proc. of the 9th Asia-Pacific Soft. Eng. Conf. (APSEC 2002)*, Gold Coast, Queensland, Australia, pp.539-546, December 2002.
- [7] Ohira, M., Yokomori, R., Sakai, M., Matsumoto, K., Inoue, K., and Torii, K. "Empirical Project Monitor: A Tool for Mining Multiple Project Data," In *Proc. of the Intl. Workshop on Mining Software Repositories (MSR2004)*, pp.42-46, 2004.

^{*2} <http://www.empirical.jp/>

- [8] Ohsugi, N., Tsunoda, M., Monden, A., and Matsumoto, K., "Effort Estimation Based on Collaborative Filtering," In *Proc. of the 5th Intl. Conf. on Product Focused Soft. Process Improvement (PROFES2004)*, pp.274-286, 2004.
- [9] PMI Standards Committee, "A Guide to the Project Management Body of Knowledge (PMBOK Guides)," Project Management Institute, 3rd Edition, 2004.
- [10] Walkerden F., and Jeffery, D.R., "Software Cost Estimation: A Review of Models, Process, and Practice," *Advances in Computers*, pp.59-125, 1997.