Effects of Software Industry Structure on a Research Framework for Empirical Software Engineering

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

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

Tomoko Matsumura
NAIST (EASE)
tomoko-m@empirical.jp

Satoshi Iwamura
NTT Software (EASE)
iwanof@empirical.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 (EASE)
mbarker@is.naist.jp

Ken-ichi Matsumoto
NAIST (EASE)
matumoto@is.naist.jp

ABSTRACT
The authors describe a new research framework for applying empirical software engineering methods in industrial practice and accomplishments in using it. The selected target for applying the framework is a governmentally funded software development project involving multiple vendors. This project involved in-process project data measurement in real time, data sharing with industry and academia, data analysis, and feedback to the project members. Today the project is in the system integration process. This paper shows the value of this research framework and describes issues of empirical data sharing between industry and academia which have emerged while using it. This experiment raised two major issues. One is the necessity of a new research framework for project measurement called the "Macro Measurement Tool". The other is effects of the software industry structure on this framework.

Categories and Subject Descriptors
D.2.9 [Management] Software process models.

General Terms
Experimentation

Keywords
Empirical Software Engineering. Software Process Measurements.

1. INTRODUCTION
To firmly support empirical software engineering research based on measurement, Industry and Academia (I&A) must share field data and knowledge and also exchange feedback about accomplishments.

Such collaboration has been difficult in Japan [1]. To overcome these limitations, we defined a new research framework supported by government policy [2]. Then practically we constructed such a framework and applied it to an actual software development project to make sure it was effective. Also we tried to illustrate the requirements for effective use of the framework.

ICSE'06, May 20-28, 2006, Shanghai, China.
ACM 1-59593-085-X/06/0005.

2. The Background
Under the government's political leadership, two new software engineering research frameworks for collaboration between I&A were developed by two ministries, MEXT (Ministry of Education, Culture, Sports, Science and Technology) and METI (Ministry of Economy, Trade and Industry). They are EASE (Empirical Approach to Software Engineering) project and SEC (Software Engineering Center)[3]. Each project has connections with both industry and academia, and both promote collaboration between I&A. With the commonality of interests, these two projects decided to work in collaboration.

The EASE project is an I&A collaboration research project and has a small research laboratory at a convenient location. Some post-doctoral researchers from local universities and software development practitioners from the software industry were gathered there for software engineering investigation. This research team developed a software project measurement platform named EPM (Empirical Project Monitor). EPM was distributed to industry as open source to provide a communication media for I&A. Then EASE started I&A collaboration activity of sharing data from software developments. EPM has been distributed to several software companies and adopted to real development projects, with some collaboration research projects sharing field data[4] [5][6].

SEC is a new investigation and research organization, organizing various software engineering professionals and practitioners into several task forces to publish practical working reference models, engineering standards and white papers. In this activity, SEC gathered and analyzed project benchmark data from over 1000 projects and published them as a software development data white paper.

In spring 2005, EASE and SEC started a new collaborative activity to apply their accomplishments to a real software development project and to feedback results to industry. The target project is named the Advanced Software Development Project (ASDP).

3. Features of target project
ASDP is a middle scale project, selected under METI leadership.

This project is in the field of Public Information Systems. It will develop an experimental Probe Information Platform, using a special research consortium organized with seven companies including an automobile manufacturer, an automobile component manufacturer and five major IT vendors. System inputs are location and speed information from various cars such as automobile, bus, truck and taxi, which are called probe cars. System outputs are publicly useful real time information generated from those data by cleansing and assimilation processing. This system development expense is supported by government. The experimental period is two years with two stages of development.

The project will develop software through wide area distributed development across the IT vendors in the consortium. These IT vendors compete with each other, so the development project clearly distinguishes between collaboration and competition areas. In the collaboration area, the vendors share information, while in the competition area information is confidential.

Developing software is mainly application programs on a Linux server, coded in C/C++ with access to RDB. Some personal computer programs for displaying results are also included.

4. Reach of the Empirical Environment

4.1 Project measurement framework

Fig.1 shows the in-process project measurement and feedback structure. The measured project data are gathered to SEC.

Key points are minimal added load to measurement for development worker, analysis tools for fast feedback to project operation and visual presentation of analyzed data.

When a software developer uses the configuration management tool, bug tracking tool or mail system without thinking about measurement, EPM collects development process and product data in the background. Then the EPM analyzer analyzes and displays the data visually. In addition, data analysis also uses source code and review records reported using electronic data sheets.

Electronic data sheets collect project attribute records along with 400 SEC defined benchmark data items. For example, these records include source lines of code, function points, development effort, duration and project attributes. Along with these records, initial planning data and actual records are collected at the end of basic design. At the end of the project, final actual records are collected.

Additionally, a 110 item questionnaire checklist and interview collected project context information from the project manager and leaders.

4.2 Data sharing and analysis

When taking measurements, SEC, EASE project analyst group and software development group held discussions and negotiations. Consequently we got consensus about measurement items, data input forms, data collection methods, measurement tools and tool operation rules. Such negotiation was a key factor for success in this measurement project.

The consensus on basic analysis items includes items to estimate the daily project progress, to support combination analyses and for project benchmark analyses. Source code is analyzed using the source code clone analyzer [7]. Benchmark data are analyzed by applying collaborative filtering technology to retrieve similar projects from the benchmark data about 1000 past projects in the SEC database and predict future project estimates from the group of similar projects [8].

4.3 Measurements effects

Analysis of review reports showed each company's attitude to the review process. Some companies were faithful to the waterfall process and respected the review process. But another company respected the testing process and rather slighted the review process. We could expect some problems when software parts from different cultural companies were submitted to the system integration test process.

We produced a bird's-eye view of each company's project based on the source line of code transitions and bug number transitions.

Code clone analysis showed the origin of the source code and background of the development group. For example, code developed from scratch,

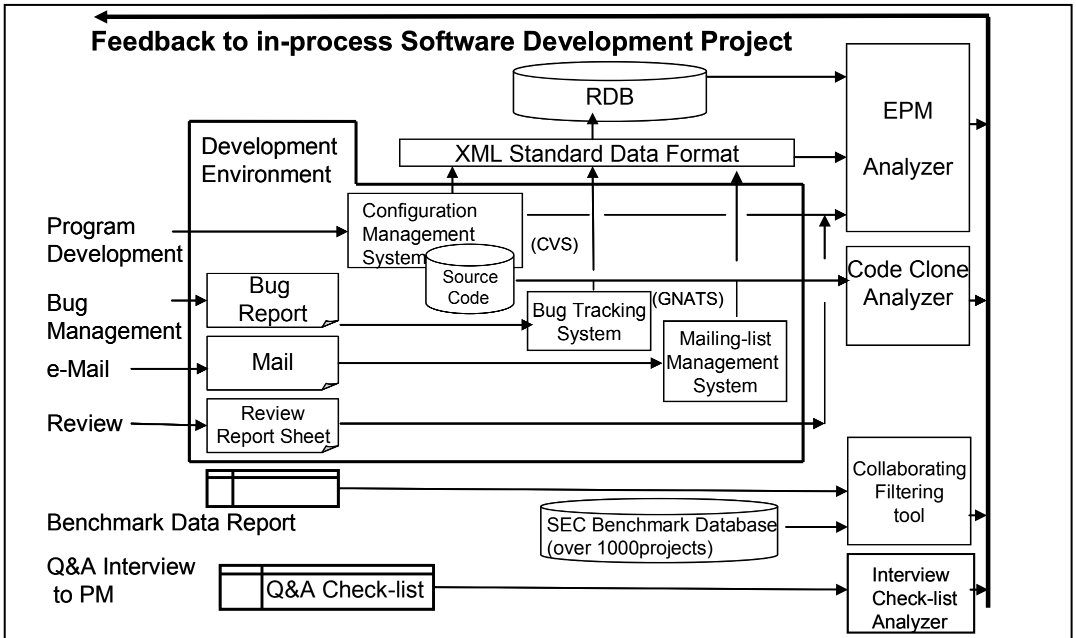

Fig.1 In-process Project Measurement and Feedback Structure

by cut-and-try approach, or through reuse. The code clone content ratio also helped identify characteristics of source code such as whether a less experienced coder or not wrote it, and whether there are concerns about future code maintenance and re-factoring.

Analysis of file renewal transitions indicated the type of development process, such as a waterfall type development or cut-and-try development. This analysis showed the file renewal stability, impacts of design changes and attention to serious bug detection in the late development process.

Benchmark data analysis using collaborative filtering technology provided a way of projecting the project future characteristics.

5. Revealed point for empirical environment

5.1 Necessity of "Macro Measurement Tool"

In the case of ASDP, with preparation and agreement between all parties concerned in the project, it was possible to measure the whole process from the viewpoint of software engineering and to share analyzed data between I&A.

The hierarchical structure shown in Fig. 2 strongly affected data collection. From the view point of tools, there are three layers of "Macro Measurement Tool" hierarchy. First layer is the software tools layer such as CVS and GNATS, 2 nd layer is the measurement platform layer such as EPM, and then 3 rd layer is the organization structure layer such as SEC and EASE project. By tailoring tools to these conditions, we could develop an effective measurement and feedback environment, in other words empirical environment. Followings are key factors of this "Macro Measurement Tool" realization in the case of ASDP.

1) Governmental leadership for I&A collaboration framework
2) Lightweight concept of measurement platform
3) Thoughtful measurement planning and execution
4) Aggressive total coordination

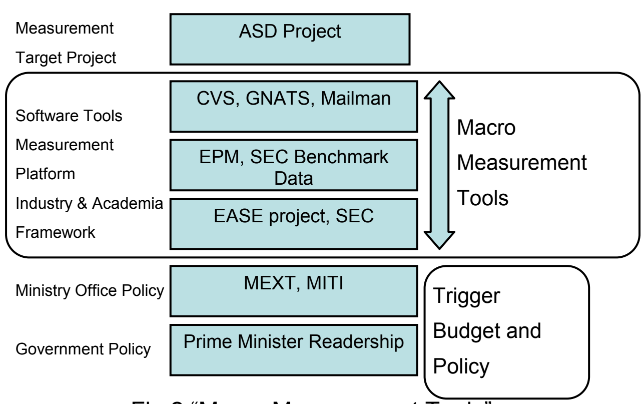

Fig.2 "Macro Measurement Tools" for Software Project Measurement

5.2 Necessity to correspond with the industrial hierarchical structure

5.2.1 Actual development hierarchical structure

Generally Japanese software industry has a complicated hierarchical structure. Fig.3 roughly shows the development organization hierarchy in the case of ASDP.

ASDP owner is METI. Assigned by METI, SEC experiments with empirical software engineering trials. Company A to F practically have contracts with METI and share development with each other.

Company A plays a role of PM partially. Company A's role is development management but human resources management and money flow control are entrusted to each consortium company. Company A is not a main contractor so accessibility to development management information is limited.

Company B to F develop the probe system under the project management of Company A. After each company's internal development and test, all software is gathered to the integration test environment, and integration and total test is executed in a collaborative manner.

Company B to F construct various structured partner company groups for development. There are various group structures and it is difficult to find patterns in them. Generally those structures are confidential except to the owner, METI. Company B to F act as the main contractor for each group and manage inside their own company group. Also company B to F contribute requirements definition and basic design and control for inter group issues. Properly there are contracts and money flow between each company and the individual developers belong to each company. Quality assurance and shipment are controlled hierarchically by each company along with each company's contract. The company locations are various. At last, developed software are shipped to the owner under responsibility of company B to F.

5.2.2 Project measurement granularity and feedback targets

In this complex development organization structure, the project was measured at the point of the CVS and GNATS servers. In most company groups, CVS was used for shipping management for each company group and not for daily configuration management.

The targets for analyzed data feedback were various companies in each company group, as shown in Fig.3. In some cases the feedback target was a top company of each company group, and in other cases the target was a middle software company, or both. But they had never directly provided feedback to the lower layer development company.

The feedback report was edited for each individual company and other company's data was confidential. For the PM, a special report was edited to give a project total view, but source code and file names were confidential.

5.2.3 Future issue for project measurement and feedback

This trial was revolutionary in dealing with the black-box structure in the software industry, however the effect of analyzed data feedback was limited by the industry structure shown in Fig.3.

As a future issue, it is necessary to provide software development management environment such as CVS and GNATS to the tail end development field and measure data from them, and to share analyzed data with the tail end developers. This is an evolutionary issue in the software industry to conquer contract hierarchy and create a logically flat and transparent development organization.

6. CONCLUSIONS

This experiment demonstrated the effectiveness of in-process project measurement and feedback in the black-box environment of a governmental consortium-type software project without a main contractor.

This experiment has given us two key points from the process innovation view.

One point is the necessity of a national or industry level "Macro Measurement Tool." Only applying measurement and analysis tools as software tools cannot accomplish process innovation based on measurement and information sharing. It is essential to create a fundamental shared platform for measurement and feedback with special organizations such as the SEC and EASE projects by powerful governmental leadership.

Another point is the effects of the hierarchical software industrial structure. Software development organizations in Japan are not a simple flat structure such as owner, leader and developers. Generally they are complicated hierarchical structures with various ad-hoc additions. Measurement and feedback are useful if they reflect that structure.

Finally, the point-of-view discussed here is from the owner's view rather than that of the developers. This experiment showed that the software process innovation effort was insufficient when it was approached from the point-of-view of a small software development group or single software company. It was meaningful that we could have a wide view for software process innovation from this experiment.

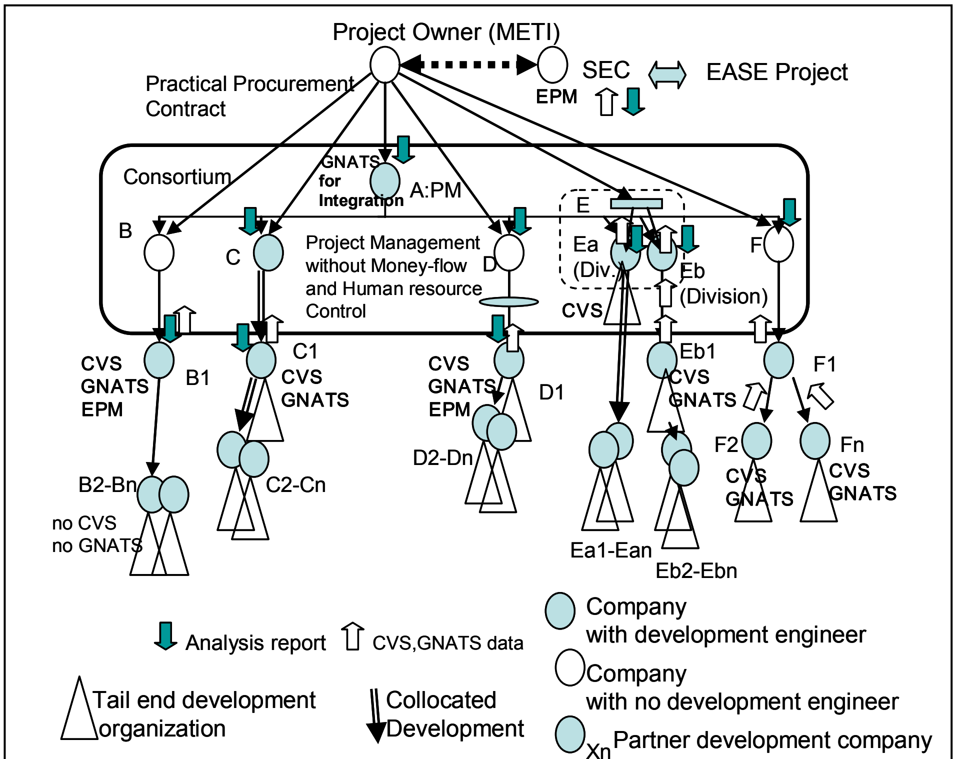

Fig.3 Development Organization

7. ACKNOWLEDGMENTS

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 SEC and EASE project who kindly support our project.

8. REFERENCES

[1]Collins, S. & Wakoh, H. :Universities and technology transfer in Japan: Recent reforms in historical perspective., *Journal of Technology Transfer 25* ,(2000) 213-222

[2]Yoshiki Mitani, Mike Barker, Koji Torii, Seishiro Tsuruho: An Experimental Framework for Japanese Academic-Industry Collaboration in Empirical Software Engineering Research, *International Symposium on Empirical Software Engineering (ISESE)2004*,Redond Beach, USA,Aug.(2004).

[3]Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Satoshi Iwamura, Mike Barker, Ken-ichi Matsumoto, An empirical trial of multi-dimensional in-process measurement and feedback on a governmental multi-vendor software project. *International Symposium on Empirical Software Engineering (ISESE) 2005, vol.2,*Noosa Heads, Australia,Nov.(2005)5-8

[4]Masao Ohira, Reishi Yokomori, Makoto Sakai, Ken-ichi Matsumoto, Katsuro Inoue, Koji Torii: "Empirical Project Monitor: A Tool for Mining Multiple Project Data", *International Workshop on Mining Software Repositories (MSR)2004*, Scotland, UK, (2004.)

[5]Masao Ohira, Reishi Yokomori, Makoto Sakai, Ken-ichi Matsumoto, Katsuro Inoue, Michael Barker, Koji Torii: Empirical Project Monitor: A System for Managing Software Development Projects in Real Time , *International Symposium on Empirical Software Engineering (ISESE) 2004*, Redond Beach, USA,Aug (2004).

[6]EASE project http://www.empirical.jp/English/index.html

[7]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) 654-670

[8] N.Ohsugi, A.Monden, S.Morisaki: Collaborative Filtering Approach for Software Function Discovery. In: *Int. Symp. Empirical SE (ISESE)2002*, *vol.2*, Nara, Japan (2002) 45-46

IPA: Information-Technology Promotion Agency, Japan

NAIST: Nara Institute of Science and Technology