Proposal of a Complete Life Cycle In-Process Measurement Model Based on Evaluation of an In-Process Measurement Experiment Using a Standardized Requirement Definition Process

Yoshiki Mitani[1,2], Tomoko Matsumura[2], Mike Barker[2],
Seishiro Tsuruho[1,3], Katsuro Inoue[4], Ken-Ichi Matsumoto[2]
Information Technology Promotion Agency, Japan(IPA) [1],
Nara Institute of Science and Technology(NAIST) [2]
Kochi University of Technology[3], Osaka University[4]
{ymitani|tomoko-m}@empirical.jp, mbarker@MIT.EDU,
tsuruho@ipa.go.jp, inoue@ist.osaka-u.ac.jp, matumoto@is.naist.jp

Abstract

This paper focuses on in-process measurements during requirements definition where measurements of processes and products are relatively difficult. However, development processes in Japan based on the Enterprise Architecture method provide standardized formats for such upstream processes and products, allowing in-process measurements. Based on previous work and on this examination of in-process measurements of requirements definition with the Enterprise Architecture method and previous results of empirical studies of in-process measurements and empirically validates of later development processes, this paper proposes a new measurement model, the "Full In-Process Process and Product (I-PAP) Measurement Model," which includes the complete software development process from requirements to maintenance.

Standardization of the requirements definition phase using the Enterprise Architecture method in Japan allows in-process measurement across the complete development lifecycle. Combining this with collaborative filtering and a project benchmark database will support project evaluation, estimation, and prediction.

Keywords: Empirical software engineering, Software process measurement, In-process measurement, Enterprise Architecture, Requirement definition phase measurement.

1. Introduction

This paper proposes a measurement model for the complete software development process from requirements to maintenance. It is based on previous work with empirical measurement in a midsized multi-vendor distributed software development project and experimental measurement of requirements definition using the Enterprise Architecture method in an ongoing project.

This model is expected to be useful for waterfall type software developments in Japan, and may provide useful information for similar efforts in other countries.

The next section of this paper presents a generalized software process measurement model based on previous measurement experiments with later processes such as detailed design and coding. The third section describes the authors' approach to process measurement during requirements definition. While this is traditionally difficult, the Japanese government has standardized the process and product formats by the Enterprise Architecture (EA) method. This section introduces an outline of the EA method. The fourth section describes a target project using the EA method, outlining experimental measurements, measurement methods, results, and evaluation of the usefulness of these measurements. The fifth section proposes the full process measurement model. The sixth section describes future research and issues. The seventh section is a brief conclusion.

2. The in-process project measurement model based on previous empirical studies

Previous empirical studies on in-process measurement focused on later processes such as detailed design and coding have verified the usefulness of those measurements in software project management [1] [2]. Based on these experiments, a measurement model such as that shown in Fig.1 can be developed. In this model, the software development processes considered as the "requirements definition," "design," "development," "integration test," and "maintenance" phases. The measurement targets are roughly classified into basic measurement targets and extended targets as shown in Fig.1.

The basic measurement targets from the "requirements definition" phase through the "development" phase are the amount of description products for each phase. The basic analysis targets are the number of transitions of description products. Concretely, these are the numbers of additions, eliminations, and modifications of descriptions. In the "testing" phases such as unit tests and integration test, the numbers of detected faults and the numbers of corrections, along with the number of transitions are measured.

As extended measurements, in the "design" phases, the amount of design and design review materials are measured, and transitions in that amount are analyzed. In the "development" phase, the measurement targets are source code entities and their modification histories. In the "testing" phase, various fault reports are measured. Also, fault cause factors and fault related phases are analyzed. In the "maintenance" phases, the extended measurement and analysis approaches are being developed by research on mining software repositories. For example, at the international conference ICSE 2006 in Shanghai, the co-located workshop on mining software repositories had various presentations and discussions [3]. Measurement and analysis in the "maintenance" phase are similar to those in the "development" phase and the "testing" phase. For example, these measurements focus on the source code modification, fault detection, and correction histories.

In a previous report, an in-process measurement experiment targeted a midsized multi-vendor development, focusing on measurements from the basic design phase to the integration test phase. This project included three stages of development in two years, with measurement of the following five items:

1) In the Basic and Detail Design phases, design documents and review reports were collected and analyzed.

2) Detailed attributes of projects were collected and recorded as project benchmark data. Using collaborative filtering technology with in-process measurements of the target project and a database of over 1000 past projects benchmark data, project estimates and predictions were made. Collaborative filtering is one of the estimation techniques using defective data having substantial missing values, in information retrieval research domain [4].

3) Using the automatic measurement platform named "Empirical Project Monitor (EPM)," the program modification history from the configuration management system, faults and maintenance data from the bug tracking system, and project e-mail information from the mailing list management system were analyzed.

4) During development, source code clones (reused segments of code) were analyzed several times.

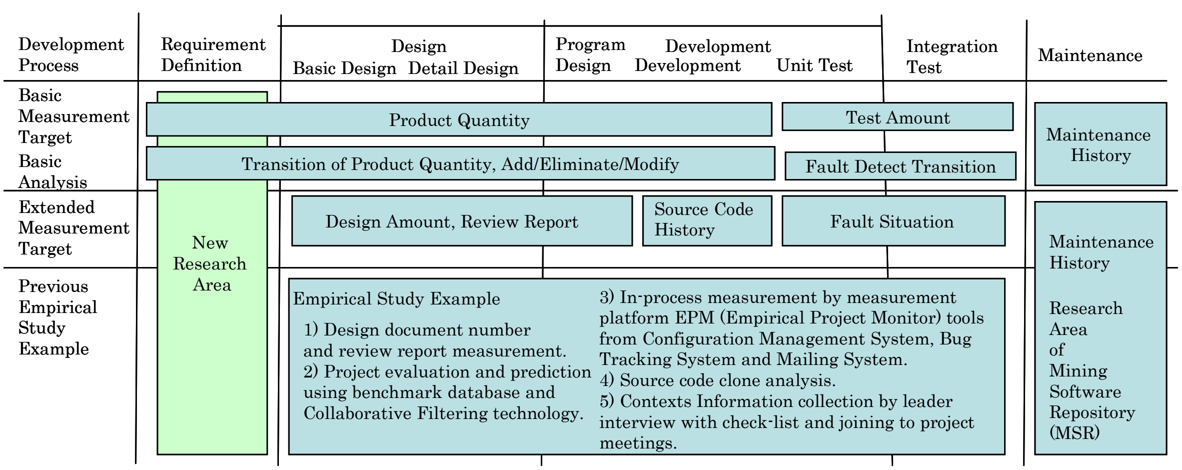

Fig.1 Generalized In-process Project Measurement Model based on Previous Empirical Study

5) Context information about the development phase was collected through interviews with the project leader and sub-leaders using checklists and through attending project meetings.

This empirical study combined the measurements and analysis to provide feedback to the project management. This produced positive effects for the project.

Based on the success of this research, the authors wanted to expand in-process measurement to the "requirements definition" phase and to develop a complete lifecycle measurement method for processes and products across the full development process.

3. Characteristics of the "Requirements Definition" phase

3.1 The Requirements Definition Phase and EA Method

Ordinarily, the "requirements definition" phase precedes the "design" phase, and in most enterprise information system development, this phase is performed as collaborative work involving the system user company and IT vendor company. In many cases, the requirements definition phase and development phase are performed by different companies or divisions, such as a consulting firm with expertise in the target business will perform requirements definition while a software company with professional technology in the information system field will perform the development phases.

Empirical study and measurements of requirements definition phase has been difficult because there are various methods and tools used and there is relatively little standardization. However, the use of the Enterprise Architecture method in Japan makes it easier to measure this phase.

The Enterprise Architecture is a total methodology for enterprise information system development, arranging the organizational business and information systems to provide overall optimizations. It is based on Zachman's framework [5] and constructed from various proposed design and management methods.

Practically, the EA method consists of three phases, the AsIs, ToBe, and policy arrangement phases. In the AsIs phase, the present state of the target system is described. In the ToBe phase, the future ideal system is designed. The policy arrangement phase, between the AsIs and ToBe phases, makes decisions about policies for optimizing the business and systems. A special feature of the EA method is that it defines many standardized hierarchical diagrams for the AsIs and the ToBe phases to increase the mutual understanding between target system stakeholders.

From the viewpoint of project measurement, it is easier to expand project measurement methods from the development phase into the requirements definition phase because the EA method has standardized process and product formats. The in-process measurement can use the amount of descriptions and the transitions in diagrams in the AsIs and ToBe phases, such as the total number of diagrams, along with additions, eliminations, and modifications of diagram elements. Section 4 describes such an empirical study.

3.2 The Japanese Government's EA Guideline

In 2003, the Japanese government provided an EA guideline for the requirements definition phase to reconstruct and optimize government business and information systems [6]. This guideline provides basic architecture, reference models, and EA products as shown in Table 1 for the AsIs and ToBe phases. It provides three phases, AsIs, Optimize, and ToBe, as the project process architecture. It recommends that Earned Value Management (EVM) and Work Breakdown Structure (WBS) method be used as the management architecture. In the AsIs and ToBe phases, the requirements definition uses four kinds of diagrams: The Diamond Mandala Matrix (DMM), Data Flow Diagram (DFD), Work Flow Architecture (WFA), and Entity Relationship Diagram (ERD). These diagrams are illustrated in Fig.2-1~Fig.2-4.

An in-process measurement trial was conducted of a governmental development project using the EA method. The project is now under development, but the interim results indicate that this kind of measurement is useful. The next section provides a report and evaluation of the empirical study.

Table 1 EA Guideline Structure

Basic Architecture	Reference Model	EA Products
Business Architecture	Business Reference Model Performance Reference Model	Business Description Document Diamond Mandala Matrix (DMM) Data Flow Diagram (DFD) Work Flow Architecture (WFA)
Data Architecture	Data Reference Model	UML Class Diagram Entity Relationship Diagram (ERD) Data Definition Table
Application Architecture	Service Component Reference Model	Information System Reference Diagram Information System Function Structure Table
Technology Architecture	Technology Reference Model	Network Structure Diagram Software Structure Diagram Hardware Structure Diagram
Project Process Architecture		AsIs Process, Optimize Process, ToBe Process
Project Management Architecture		Earned Value Management (EVM) method Work Breakdown Structure (WBS) method

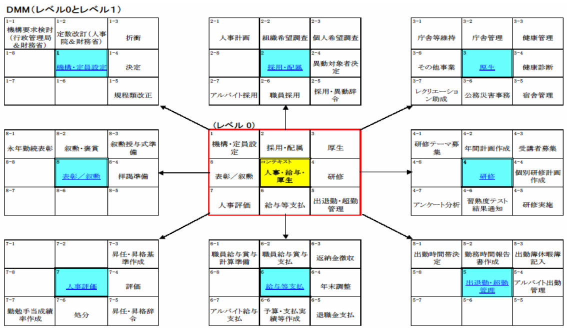

Fig.2-1 Diamond Mandala Matrix (DMM)

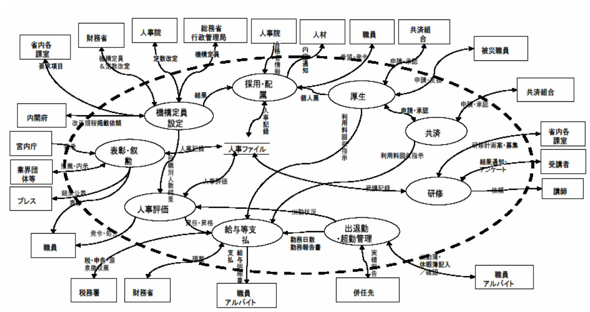

Fig.2-2 Data Flow Diagram (DFD)

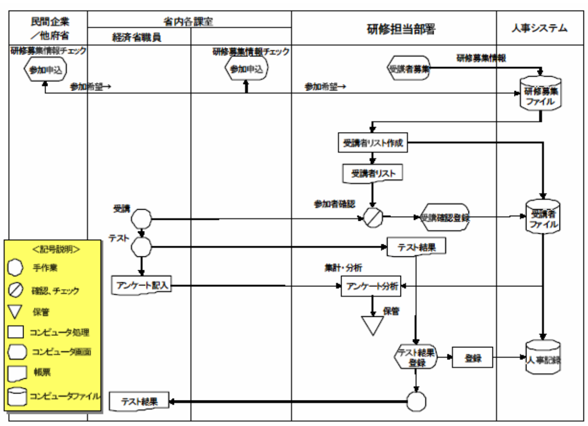

Fig.2-3 Work Flow Architecture (WFA)

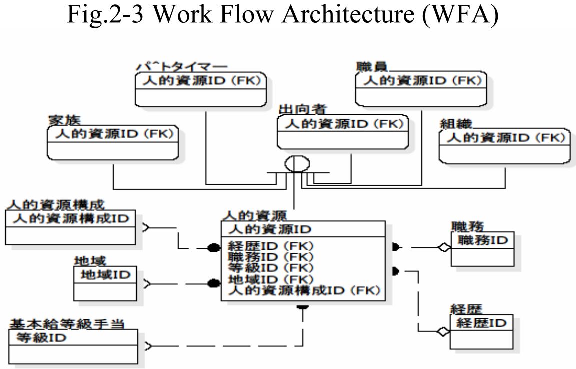

Fig.2-4 Entity Relationship Diagram (ERD)

4. Empirical study and in-process measurements of an Enterprise Architecture project

4.1 Outline of the Target Project

The target project is a plan for business and system optimization of the Information Technology Promotion Agency, Japan (IPA) [7], a subsidiary organization of the Ministry of Economy, Trade and Industry (METI). This project includes four systems (Businesses A to D) such as a total business management system for a business organization of about 200 employees. The requirements definition phase started in August of 2006 and will be completed in March 2007. The requirements definition work is being done by IPA staff and a business consulting company in a collaborative manner. The goal of the project is to provide requirements definition and basic design of optimized systems, cost estimation, and an RFP for development process procurement. By government policy, the development phase should be procured from another software company.

This project produced 34 kinds of documents including 30 diagrams as shown in Table 2. The planned document types are different for each business application because the development phase schedule depends on each business. Fig.3 shows the outline schedule of this project.

Table 2 Planned Documents and Measurement Targets

Target Business	A		B		C		D	
AsIs/ToBe	AsIs	ToBe	AsIs	ToBe	AsIs	ToBe	AsIs	ToBe
Business Description	x		x		x		x	
DMM			**x**	**x**			**x**	**x**
DFD	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
WFA	**x**	**x**	**x**	**x**			**x**	**x**
ERD			**x**	**x**			**x**	**x**
Information System Reference Diagram			x	x			x	x
Network System Diagram			x	x				
Software System Diagram			x					
Hardware Structure Diagram			x					

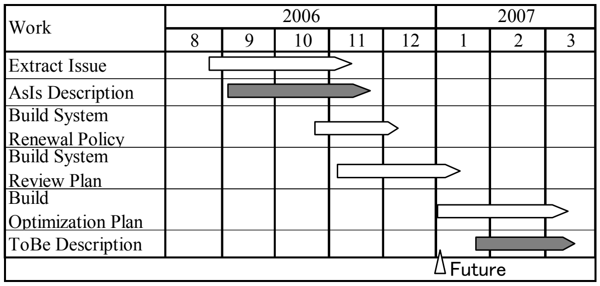

Fig 3 Schedule of Target Project

4.2 Project Measurements

This project used the following in-process project measurements:

1) weekly measurement of four types of diagrams: DMM, DFD, WFA, and ERD (22 diagrams total)
2) interviews of the project leader with a checklist to obtain project context information, a method that was useful in the previous development process measurement experience
3) attending the project meetings to get context information, also useful in the development process study.

The measured diagrams are written with an EA tool constructed on Microsoft's drawing tool Visio. The Visio reporting function was used to measure the number of diagrams, including the number of sheets, diagram elements, and transitions. While the measurement process included some manual work, mainly it was done automatically. The 22 documents were selected for study because there were very little other content and few other documents, suggesting that these 22 documents represent the total process outcome for this project. This is a hypothesis to be verified by the study. Also the reflection of project context information in the measurement is considered very important.

4.3 Project Measurement Results

By December 2006, in about 13 weeks, the description of the AsIs diagrams was completed for three businesses, with 80% of the description for the fourth business also completed. During this process, the following measurements and graphical visualizations were made:

1) the amount of diagram description and transitions measured in number of sheets, both total amounts and for each business
2) the number of diagram elements and diagram connector elements in the diagrams and their transitions, measured as total amounts and for each business
3) changes in the numbers of diagram files through addition, elimination, and modification, both total and for each business
4) for each diagram in each file, measure the addition, elimination, and modifications of elements by counting text strings on the diagram elements.
5) Weekly described amount of diagram transitions by number of sheets, diagram elements, and connector elements, both total and for each business.

Fig.4 to Fig.17 shows examples of these measurement results in graphical form. The following trends are directly read from those graphs.

Fig.4 shows changes in the numbers of diagram sheets during the 13 weeks. In total, about 450 sheets were described. Fig. 5 shows changes in the described diagram elements and connector elements. In total, 20,000 elements were described. About 30% to 50% of those elements were connector elements. Fig.6 shows the number of described elements for business D. In this study, the data for each business could be illustrated. Fig.7 is a graph showing the accumulated data for all businesses. For each business, the starting point of the description work, changes in the amount of description, and the status of the description work are clearly visible. Fig.8 spreads the vertical axis of Fig.6 to clearly show business D. Fig.9 shows the data for business B. it is visible that business D is under development, but business B is already almost finished with the description document in a stable state. Fig.10 shows changes in the number of files. This is not only the accumulative total number, it also includes additions, eliminations, and modifications. From these transitions, the stability of the description work is visible. Fig.11 and Fig.12 shows the status of Business D and Business B. Business D is still active in diagram description while business B is already finished and stable. Fig.13 to Fig.15 focuses on specific diagrams, tracking the addition, elimination, and modification of data elements within the diagram. Perfect tracking is difficult, so the method used is to track modification of text strings on each diagram element. Fig.13 and Fig.14 show modifications in each diagram in business D over six weeks. Fig.15 shows all changes of one diagram file consisting of eight diagrams including the diagrams for Fig.13 and Fig.14. From these figures, the trend appears to show that this area's AsIs description work is gradually stabilizing. Fig.16 and Fig.17 showed changes in the amount of description on a weekly basis. The amount of work performed for each business can be compared from this graph. From this graph, it is clear that the description process for the four businesses were executed shifted a few weeks.

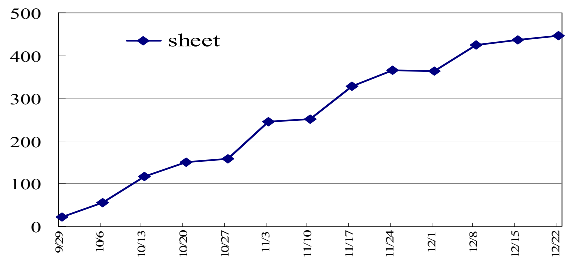

Fig.4 Total Sheet Number

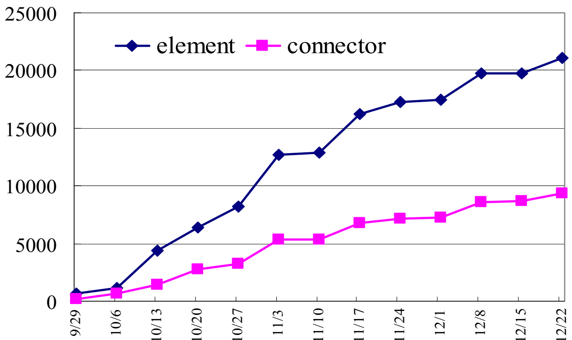

Fig.5 Diagram Element Number

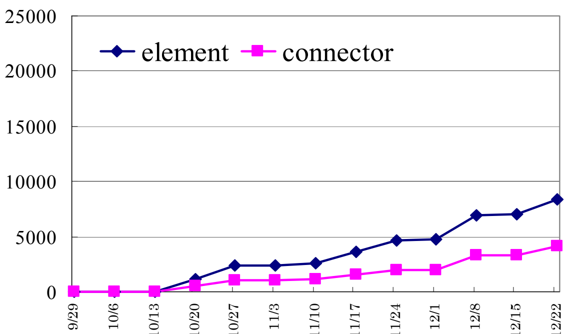

Fig.6 Diagram Element Number in Business D

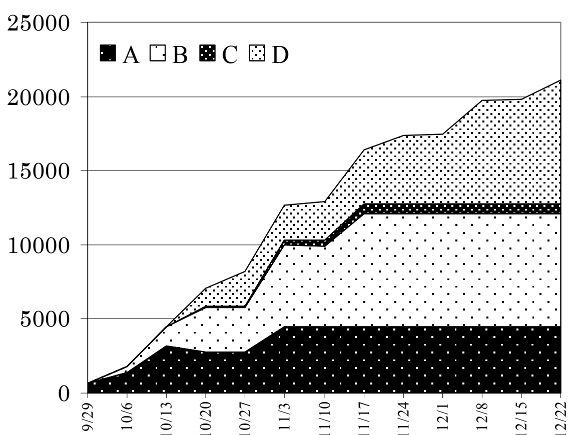

Fig.7 Diagram Element Stuck of all Business

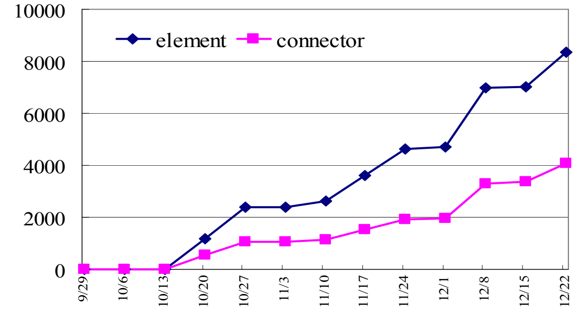

Fig.8 Diagram Element of Business D in full Y scale

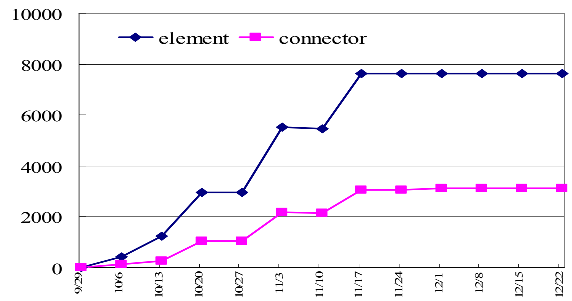

Fig.9 Diagram Element of Business B in full Y scale

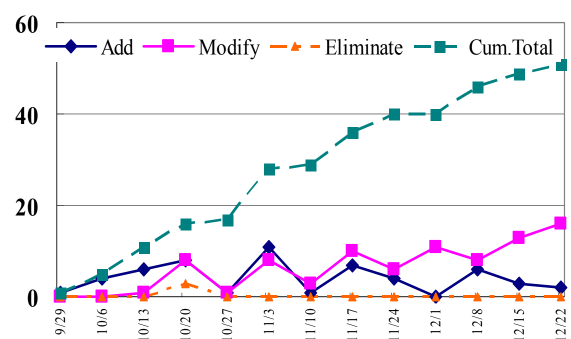

Fig.10 File Number Transition

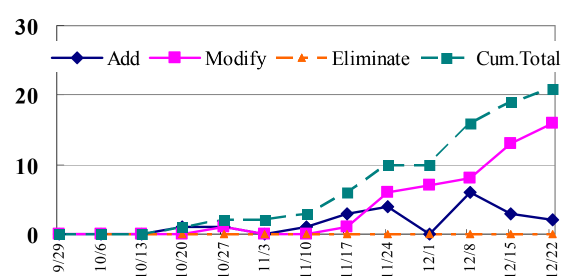

Fig.11 File Number Transition of Business D

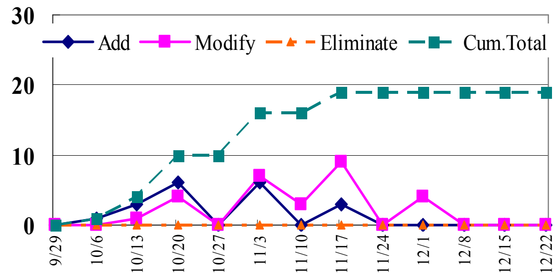

Fig.12 File Number Transition of Business B

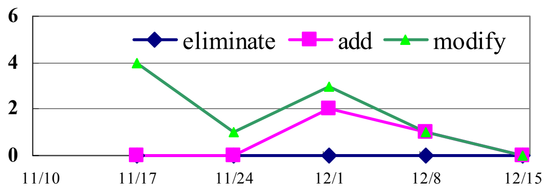

Fig.13 Diagram Modification Transition (Sheet A)

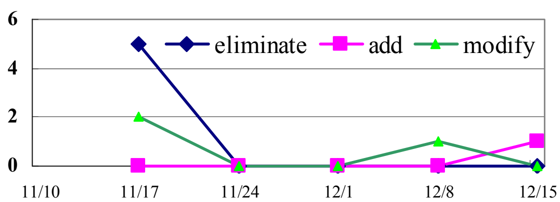

Fig.14 Diagram Modification Transition (Sheet B)

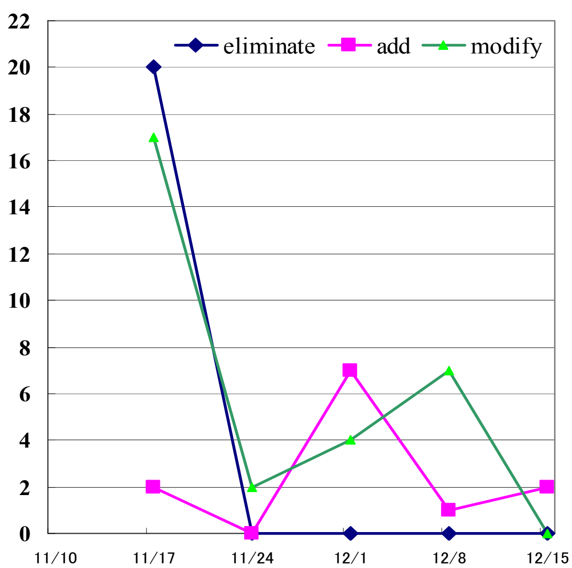

Fig.15 Diagram Modification in File A (8 sheets)

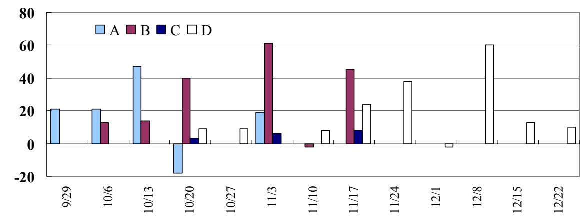

Fig.16 Weekly Addition of Sheet Number

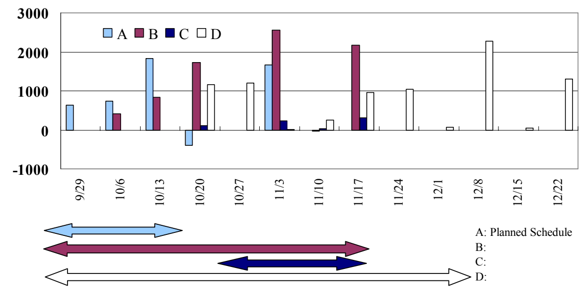

Fig.17 Weekly Addition of Diagram Elements

4.4 Evaluation and Study of the Measurement Results

The measurement results of this trial were evaluated and studied to verify the usefulness of in-process measurement. The results are compared with the official progress report based on declarations. The results also suggest a new metrics development possibility which is described below.

4.4.1 Comparison with official progress report

Based on the governmental EA guideline, the target project is managed with EVM and WBS methods. The official progress report is based on declarations by the participants. Fig.18 is an example of the EVM report. This report shows the consumed human resources, but it is not clear about the situation of the outcome. Table 3 shows an example WBS report. This has been made more visible in Fig.19. In the WBS method, the work in progress is reported through detailed activities. However, the granularity of this report is very rough. There is no information about the amount of outcome produced. The WBS based report is based on declared progress estimation criteria. The criteria are that when an activity is started, 20% of progress is reported. When a working plan is described, 40% is reported. When a basic idea is described, 60% is reported. When the internal review is complete, 80% is reported. And when all work is finished, 100% is reported. This reporting is also limited by self declaration and human intervention.

Compared to these official reports based on self declarations, the product measurement method tried in this study presents detailed information with high granularity based on real amounts of outcome products. This information is based on the raw data of the production, so it is not affected by human intervention. For example, Fig.17 includes the declared schedule of work. From this graph, gaps between the declared schedule and the real work progress based on actual product information are clearly visible.

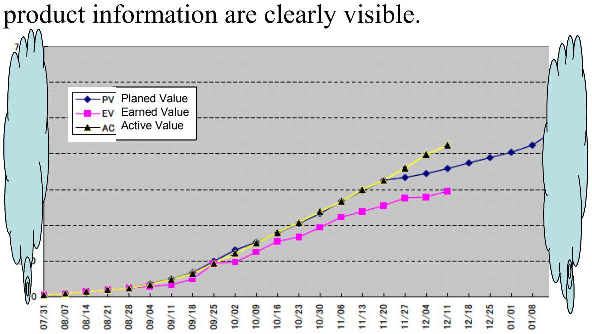

Fig.18 EVM Report Example

Table 3 Declaration Level Progress Report (%)

	Meeting	2	3	4	5	6	7	8	AsIs Schedule	
Business	Report date	10/8	10/22	11/2	11/9	11/17	12/1	12/15	Start	End
A	DMM	40	100						9/25	10/13
	WFA	40	100							
	DFD	0	100							
B	DMM	40	80	100	100				9/25	11/24
	DFD	0	0	50	90	80	100			
	WFA	40	80	100	100					
	ERD	0	0	0	0	80	100			
C	DMM	0	0	60	80	100			10/23	11/10
	DFD	0	0	20	20	100				
D	DMM	70	95						9/25	11/24
	DFD	0	0	0	0	0	20	70		
	WFA	10	20	80	80	70	80	80		
	ERD	0	0	20	20	30	40	70		

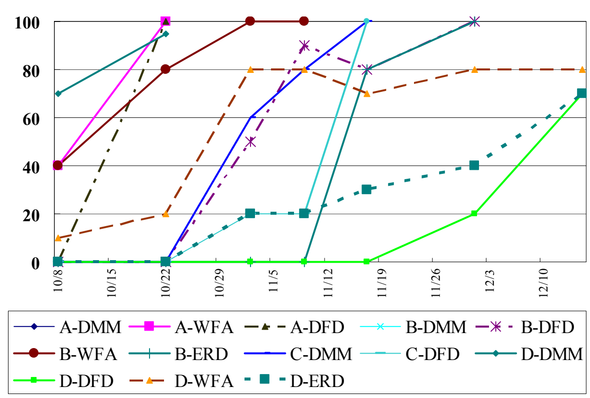

Fig.19 Declaration Level Progress Report

4.4.2 New software metrics possibility for the requirements definition phase

The target system for measurement in this study is not very large, but during the 13 weeks, the AsIs phase included over 400 diagram sheets with over 20,000 diagram elements. Considering analogies between measurement targets such as diagram sheets or numbers of diagram elements and program modules or source lines of code (SLOC), we can propose developing new software metrics for the requirements definition phase. These new metrics are expected to contribute to increasing the productivity and quality of software development processes in the same way as other existing software metrics. For example, it is likely that 20,000 diagram elements are analogous to 20,000 steps of high-level programming language code, the number of diagram sheets can be compared to the number of program modules, and the number of diagram file as corresponds to the number of program files.

To test this, Table 4 shows several new metrics based on three business diagrams already completed. The metrics include working effort per sheet, working effort per diagram element and the inverses, number of sheets per working effort and number of diagram on once per working effort. Working effort can be converted into cost.

Table 4 New Metrics Proposal in Requirements Definition Phase

Busuness	Hours	Sheets	Elements	Hours /Sheets	Hours /Elements	Sheets /Hours	Elements /Hours
A	160	80	4,456	2	0.04	0.5	27.9
B	236	171	7,629	1.38	0.03	0.7	32.3
C	80	17	682	4.7	0.12	0.2	8.5

In this case, only three data from one project were used, but it demonstrates the difference in working density between such works. Collecting and building a database of such data could provide insights in quantitative evaluation of the requirements definition phase and help develop new software metrics.

5. A proposal of a full lifecycle measurement model

5.1 Grand design of in-process measurement

Although this measurement experiment is not complete, it appears to verify the usefulness of in-process measurement for requirements definition where process and product formats are standardized by some method such as the Enterprise Architecture (EA) method. It has high similarity to development phases. Based on this evaluation of the experiment, the authors have extended the concept from Fig. 1 to propose a full lifecycle in-process measurement model for processes and products as shown in Fig.20.

This model combines previous empirical studies in project management and reports in the field of mining software repositories into a full lifecycle measurement model. The basic concept of this model is quite simple, focusing on measurements of the main outcome products of each phase, counting the total amount, including additions, eliminations, and modifications. Under this concept, a project measurement grand design could be developed, combining various measurements fitted to each phase with the basic product measurement approach as the core, utilizing the results in project management, and at the same time recording them in a project benchmark database used for project evaluation, estimation, and prediction.

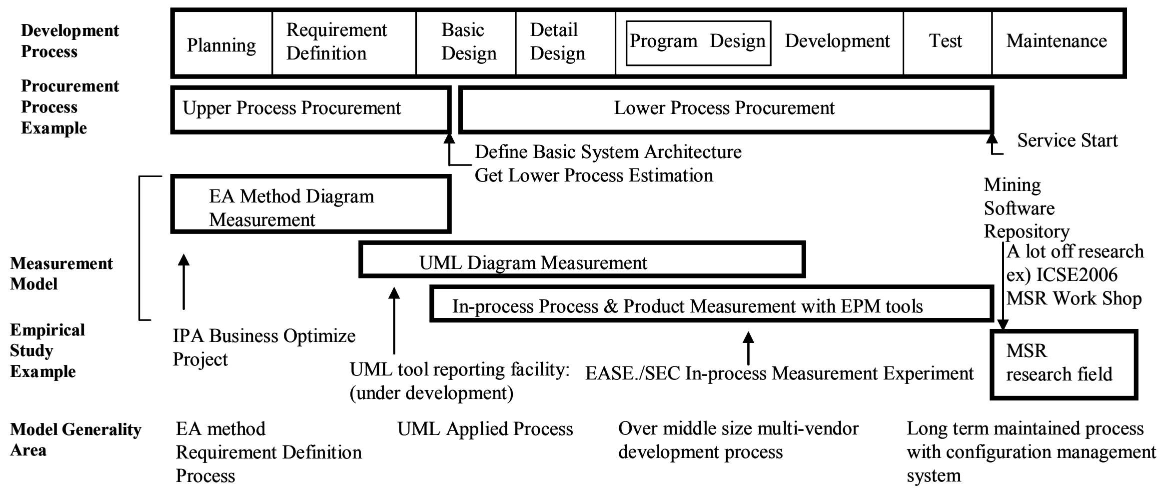

Fig.20 Full In-process Process and Product (I-PAP) Measurement Model

5.2 UML diagram measurement to increase granularity of design phase measurement

As shown in Fig.1, the previous empirical study in the design phase only measured the number of pages of design documents, providing relatively rough granularity for measurement. Based on the current study, it looks useful to measure UML diagrams as part of the design phase. In some approaches to software design, large numbers of UML diagrams will be written, such as use-case diagrams, activity diagrams, sequence diagrams to define system behavior, and class diagrams for system structure definition. In such cases, the granularity of the design phase measurements can be increased by measuring UML diagrams. Table 5 presents example measurement targets for UML diagrams. The authors intend to develop measurement tools for UML diagrams and perform empirical verification on a real project in the future.

Table 5 UML Diagram Measurement Target Example

Description Target	UML Diagram	Measurement Target
System Behavior	Use-case	Use-case, Include, Extend, Actor
	Activity	Action, Flow, Fork, Join, Signal, Decision,
	Sequence	Object, Message
System Structure	Class	Attribute, Operation, Dependence, Generalization

6. Future study

This study looked at measurement and evaluation of the AsIs phase of the requirements definition phase standardized by the EA method. Measurement of the other AsIs phase products and ToBe phase will continue. Measurements of the ToBe phase consider the following five issues:

1) tracking the stability of AsIs documents with completed descriptions

2) tracking behavior of ToBe description documents because such descriptions do not start from zero

3) evaluation of the relationship between quality of work and measurement of quantity of product. This is particularly important because deadline limitations are not favorable for high-quality descriptions of the ToBe

phase. It would be useful if information related to quality can be related to measurements of the process and product.
4) The relationship between current measurements and development phase measurements, in the viewpoints of quality and productivity.
5) Evaluation of positioning of the current measurement target in the entire requirements definition phase. It is important to understand how these measurement targets relate to the complete process. To resolve this, it will be necessary to combine context information and measured quantitative data.

Generally the requirements definition phase is performed by collaboration between the customer company (user company) and development company (consulting company). In this situation, the project measurement method used in this study can create consensus between both companies, for example, by visually displaying effects on the stability and progress of the work caused by the customer's policy decision delays and changes.

Along with this, it is important to include the requirements definition phase in the project evaluation, estimation, and prediction system using collaborative filtering technology, in-process project measurements, and project benchmark database. The requirements definition phase can be a difficult arena to achieve consensus between stakeholders about schedule estimates and cost of work. Extending the database system to include additional requirements phase data and new metrics as presented in this research can help resolve these issues.

Future study will also consider previous Measurement Model Lifecycles, including model validation, acceptance, and accreditation phases. The work will also ensure integration of the measurement model across the development lifecycle. The reliability and significance of the measurements will also be investigated.

7. Conclusion

The standardization of process and product formats in the requirements definition phase using the Enterprise Architecture (EA) method in Japan allows application of in-process measurement in both requirements definition and development phases. This empirical study presents results from measurement of process outcome diagrams, and proposes new candidate metrics for the requirements definition phase. Based on these results, a full lifecycle in-process measurement model for processes and products across the complete development process is proposed. By combining this with collaborative filtering and a project benchmark database, as proposed in previous research by the authors, a grand design of measurement allowing project evaluation, estimation, and prediction can be produced.

8. References

[1] 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) pp.5-8

[2] Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, Ken-ichi Matsumoto: A Proposal for Analysis and Prediction for Software Projects using Collaborative Filtering, In-Process Measurements and a Benchmarks Database. MENSURA 2006, International Conference on Software Process and Product Measurement. Cadiz, Spain, Nov. (2006) pp. 98-107

[3] International Workshop on Mining Software Repository 2006. Collocated ICSE 2006, Shanghai, China May 2006

[4] Naoki Ohsugi, Masateru Tsunoda, Akito Monden, Ken-ichi Matsumoto: Effort Estimation Based on Collaborative Filtering. PROFES 2004, Product Focused Software Process Improvement, 5th International Conference, Kansai Science City, Japan, Apr. (2004) pp.274-286

[5] John A. Zachman: A Framework for Information Systems Architecture. IBM Systems Journal vol.26, no.3 1987

[6] Enterprise Architecture: METI
http://www.meti.go.jp/english/information/data/IT-policy/ea.htm

[7] http://www.ipa.go.jp/index-e.html

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 the SEC and EASE project who kindly support our project.