

**博士論文**

**ソフトウェア開発プロジェクトの  
インプロセス計測とフィードバックに関する研究**

神谷 芳樹

2007年8月23日

奈良先端科学技術大学院大学  
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に  
博士（工学）授与の要件として提出した博士論文である。

神谷 芳樹

審査委員： 松本 健一 教授 （主指導教員）  
関 浩之 教授 （指導教員）  
飯田 元 教授 （指導教員）  
門田 暁人 准教授 （指導教員）

# ソフトウェア開発プロジェクトの インプロセス計測とフィードバックに関する研究\*

神谷 芳樹

## 内容梗概

近年、ソフトウェア開発プロセスやプロダクトのデータを計測、分析し、その結果を開発現場へフィードバックするというエンピリカルソフトウェア工学の研究が注目されている。学術界においては、学生実験における小規模開発や、オープンソースソフトウェア開発の分析が盛んである。その一方で、産業界における商用ソフトウェア開発の計測、分析、フィードバックを行った事例報告は極めて少なく、業界全体として知見が蓄積・共有されていないのが現状である。本論文は、産業界における 2 つのソフトウェア開発プロジェクトにおいて、新しい計測手法を実施し、その結果を整理するとともに、計測における課題やフィードバックの効果を明らかにした。

本論文の前半では、政府発注のマルチベンダ中規模情報システム開発プロジェクトの計測機会を得て、設計工程の一部と、製造・試験工程を対象としたインプロセス計測、分析、フィードバックを産学官が連携して実施した結果をまとめ、その有用性を、開発組織やプロジェクトに従事する人間に着目した新しい観察法により明らかにした。本プロジェクトでは、ソースコードの規模、変更頻度、変更回数の推移、障害件数の推移、障害混入工程と発見工程の関係の分析、レビュー記録の分析、コードクローンの分析など、従来個別に提案されてきた計測・分析方法を総合的に実施した。そして、プロジェクト関係者へのアンケートとインタビューを行うことで、プロジェクト管理者、プロジェクトリーダー、サブリーダーなどの各立場における、計測における課題やフィードバックの効果を明らかにした。

本論文の後半では、政府系の業務・システム最適化計画と呼ばれるシステム再構築プロジェクトの計測機会を得て、従来計測の難しかった要求定義工程に対する計測を実施した結果をまとめ、これまでに知られることが稀であった当該工程での成果物の推移を明らかにし、その計測の有用性を示した。本プロジェクトでは、経済産業省により策定された Enterprise Architecture (EA) ガイドラインに沿って開発が進められ、機能構成図、情報機能関連図、業務流れ図、実体関連図の 4 つダイアグラムの作成が義務付けられた。本論文では、これらダイアグラムの構成要素記述の推移をグラフ化することにより、要求定義工程が可視化され、プロジェクト進捗管理に役立つことを示した。また、ダイアグラムの計測値が、要求定義工程の生産性や品質に関する指標値となり得ることを示した。

---

\* 奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 博士論文  
NAIST-IS-DT0661208, 2007 年 8 月 23 日

本論文の最後に、これまでの計測結果を整理し、要求定義工程から保守工程までの全工程にわたる一貫計測モデルとしてまとめるとともに、過去の計測事例を蓄積したデータベースを活用するためのプロジェクト予測支援機構を提案する。また、エンピリカルソフトウェア工学の研究の推進において必須となる、産学連携による研究環境構築における課題をまとめた。

## キーワード

エンピリカル・ソフトウェア・エンジニアリング，ソフトウェア・プロジェクト計測，インプロセス計測，プロジェクト管理，ソフトウェア開発管理，要求定義，エンタープライズ・アーキテクチャ，業務・システム最適化

# Research on In-Process Measurement and Feedback of Software Development Projects<sup>†</sup>

Yoshiki Mitani

## Abstract

In recent years, empirical software engineering research, characterized by measurement and analysis of software development processes and products with analytic results fed back into project operation, has grown remarkably. In academia, such research produces energetic activities with small-scale campus experiments or analysis of open-source software development projects. However, in the industrial world there are only a few research reports about measurement, analysis, and feedback in commercial software development. As a result, there is much less knowledge about commercial software development projects accumulated and shared in the industrial world.

The first half of this thesis describes the outcomes of an in-process experiment with measurement, analysis, and feedback that targeted part of the design, code, and test phases in a government-procured multivendor middle-scale development project. Collaboration between industry, academia, and government created this measurement opportunity. This project integrated measurement, analysis, and feedback of multiple types which are often proposed as separate measures, such as rates of modification of lines of source code, check-in frequency, rates of failure detection, analyses of the relationship between phase of error injection and phase of expected error detection, analyses of review reports, and code-clone analysis. Through questioning development-related individuals, the author investigated the effects of this feedback on in-process software project management as interpreted by project staff such as the project manager, leader, sub-leader, and coder.

The second half of this thesis presents the outcomes of applying project

---

<sup>†</sup> Doctor's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT0661208 Aug. 23, 2007

measurements in the requirements definition phase that has traditionally been difficult to measure. The author had an opportunity to apply measurements to the requirements definition phase of a government-procured middle-scale project called the "business and system optimization project" that aimed at providing a new business management system for the government. For system development such as this, the Japanese Ministry of Economy, Trade, and Industry (METI) established guidelines for the requirements definition phase that define a process standard and formats for output products. METI based the guideline on the Enterprise Architecture (EA) method and it requires specific documents such as the Diamond Mandara Matrix (DMM), Work Flow Architecture (WFA), Data Flow Diagram (DFD), and Entity Relationship Diagram (ERD). Grasping the opportunity provided by this standardization of the requirements definition process in the field of government procurements, this thesis demonstrated that project management found measurements and graphical visualizations of results related to the diagrams such as the rate of change in the number of diagram elements to be useful.

Finally, based on previous research, this thesis proposes a full in-process measurement model for processes and products and a mechanism for predicting project characteristics using a database of accumulated previous project measurements. It also lists issues related to the collaboration between industry and academia that must be resolved to provide an effective research environment for empirical software engineering.

**Keywords:**

Empirical Software Engineering, Software Project Measurement, In-process Measurement, Project Management, Software Development Management, Requirement Definition, Enterprise Architecture, Business System Optimization

## 関連発表論文

### 学術論文誌

1. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, and Ken-ichi Matsumoto : A Proposal for Analysis and Prediction for Software Projects using In-Process Measurements and Collaborative Filtering of a Benchmarks Database. *Journal of Software Measurement*. (2007.7 投稿 , 2007.11 掲載予定) .
2. 神谷芳樹 , マイク・バーカー , 松本健一 , 鳥居宏次 , 井上克郎 , 鶴保証城 : 現場データを産学で共有するソフトウェア工学研究のための枠組み , *産学連携学* , Vol.2.No.2, pp.26-37, 2006.3.

### 国際会議

1. Yoshiki Mitani, Tomoko Matsumura, Mike Barker, Seishiro Tsuruho, Katsuro Inoue, Ken-Ichi Matsumoto: An Empirical Study of Requirement Definition Process Management and Metrics based on an In-process Measurement Experiment of Standardized Requirement Definition Phase. *Proceedings of International Conference on Software Process and Product Measurement (IWSM-Mensura 2007)*, Palma de Mallorca, Spain, November 2007. (2007.8 採録, 2007.11 発表予定) .
2. Yoshiki Mitani, Tomoko Matsumura, Mike Barker, Seishiro Tsuruho, Katsuro Inoue, Ken-Ichi Matsumoto: 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, *Proceedings of International Symposium on Empirical Software Engineering and Metrics 2007 (ESEM 2007)*, Madrid, Spain, September 2007. (2007.3 採録 , 2007.9 発表予定) .
3. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, and Ken-ichi Matsumoto: A Proposal for Analysis and Prediction for Software Projects using Collaborative Filtering, In-Process Measurements and a Benchmarking, *Proceedings of International Conference on Software Process and Product Measurement (MENSURA 2006)*, pp.98-107, Cadiz, Spain, November 2006. (Best Paper Award)
- 4 . Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, and Ken-ichi Matsumoto: A Proposed Method for Building a Database of Project Measurements and Applying it Using

Collaborative Filtering, *Proceedings of 5th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2006)*, Vol.2, Short papers, pp.15-17, Rio de Janeiro, Brazil, October 2006.

5. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Satoshi Iwamura, Yoshiki Higo, Katsuro Inoue, Mike Barker, Ken-ichi Matsumoto: Effect of Software Industry Structure on a Research Framework for Empirical Software Engineering, *Proceedings of 28th International Conference on Software Engineering (ICSE 2006)*, Far East Experience Track, Poster Session, pp.616-619, Shanghai, China, May 2006.
6. 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, *Proceeding of 4th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2005)*, Vol.2, Late Breaking Papers, pp.5-8, Noosa Heads, Australia, November 2005.
7. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Satoshi Iwamura, Mike Barker, Ken-ichi Matsumoto: A Research Framework for Empirical Software Engineering Collaboration and Its Application in a Software Development Project, *Proceedings of International Workshop on Future Software Engineering 2005 (IWFST 2005)*:Shanghai, China, November 2005.
8. Yoshiki Mitani, Mike Barker, Koji Torii, Seishiro Tsuruho: An Experimental Framework for Japanese Academic-Industry Collaboration in Empirical Software Engineering Research, *Proceeding of 3rd ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2004)*, Vol.2, Poster Session, pp.35-36, Redondo Beach, CA., USA, August 2004.

## 国内シンポジウム

1. 神谷芳樹, 菊地奈穂美, 松村知子, 大杉直樹, 門田暁人, 肥後芳樹, 井上克郎, 松本健一: 進行中のプロジェクト計測とフィードバック実験に基づく計測データベース活用方式の提案, *ソフトウェアエンジニアリングシンポジウム 2006 (ソフトウェアエンジニアリング最前線 2006, 近代科学社)*, pp.35-42, 2006.10.

## 解説記事

1. 松浦 清, 神谷芳樹, 樋口 登: Project Report 先進ソフトウェア開発プロジェクト Part II, *SEC journal* no.5, pp.44-49, 2006.2.



2. 神谷芳樹: EASE プロジェクトに見る計測・定量化の実践, *日経 IT プロフェッショナル*, 2005 年 3 月, pp.92-97, 2005.3.

## 共同執筆書籍

1. 情報処理推進機構, ソフトウェア・エンジニアリング・センター: IT プロジェクトの「見える化」, 上流工程編: 日経 B P 社, P.208, 2007.5.
2. 情報処理推進機構, ソフトウェア・エンジニアリング・センター: IT プロジェクトの「見える化」, 下流工程編: 日経 B P 社, P.211, 2006.6.

## その他

1. 伊東洋一, 佐藤匡正, 長野宏宣, 神谷芳樹: 総合ソフトウェア生産システムの実用化, *日本電信電話公社電気通信研究所, 研究実用化報告*, 33 巻 12 号, pp.2879-2893, 1984.12. (花田收悦編: ソフトウェアの計画と管理, 第 5 章に全文再掲, 日科技連, pp.318-326, 1987-2.)
2. Tadamasu Satoh, Hironobu Nagano, Yoshiki Mitani: An Integrated Software Development System for Data Communication Systems Utilizing Various Types of Computers, *Review of Electrical Communications Laboratories (Review of ECL)* Vol.33 No3 1985.

## 参考

1. 神谷芳樹: ソフトウェア計測の国際会議 MENSURA 2006 に参加して(海外レポート), *SEC journal* no.9, pp.18-19, 2007.3.
2. 神谷芳樹: ソフトウェア工学国際会議 ICSE 2006 上海に参加して(海外レポート) *SEC journal* no.7, pp.22-29, 2006.9.

# 目次

1 . はじめに .....	1
1 . 1  研究の背景 .....	1
1 . 2  関連研究 .....	3
1 . 3  研究の位置づけ .....	8
2 . 総合的なソフトウェア開発プロジェクト計測実験（設計工程以降） .....	17
2 . 1  計測対象プロジェクトの概要 .....	17
2 . 2  設計工程以降を対象とした総合的なソフトウェア・プロジェクト計測実験の実施 .....	18
3 . 計測とフィードバックの有用性実証に関する考察 .....	42
3 . 1  有用性実証の視点 .....	42
3 . 2  全工程プロジェクト観察による定性評価 .....	42
3 . 3  リーダー、サブリーダーへの公式アンケート調査 .....	44
3 . 4  ステークホルダの振る舞いの追跡 .....	44
3 . 5  プロジェクト・キーマンへの非公式事後インタビューと産業構造に着目した分析 .....	45
3 . 6  計測とフィードバックのまとめ .....	54
3 . 7  計測実験の成果を反映したプロジェクト計測モデルの提案 .....	57
4 . 要求定義工程のプロジェクト計測実験 .....	59
4 . 1  要求定義工程の課題と研究対象 .....	59
4 . 1 . 1  要求定義工程の課題とEA手法 .....	59
4 . 1 . 2  日本政府のEA策定ガイドラインと課題 .....	60
4 . 2  実証対象プロジェクトとプロジェクト計測 .....	64
4 . 2 . 1  対象プロジェクトの概要 .....	64
4 . 2 . 2  プロジェクト計測方式 .....	66
4 . 2 . 3  プロジェクト計測結果 .....	66
5 . 要求定義工程計測の評価，考察と提案 .....	81
5 . 1  プロジェクト計測結果の評価と考察 .....	81
5 . 1 . 1  自己申告方式の進捗報告とプロジェクト計測の比較 .....	81
5 . 1 . 2  計測範囲に関する考察 .....	82
5 . 1 . 3  要求定義工程の新しいメトリクス領域開拓に関する考察 .....	83
5 . 1 . 4  ベンチマークデータの分析要素としての可能性に関する考察 .....	85
5 . 1 . 5  要求定義工程計測のまとめ .....	89
5 . 2  ソフトウェア開発全工程一貫計測モデルの提案 .....	90
5 . 2 . 1  インプロセス計測のグランドデザイン .....	90

5.2.2	設計工程の計測粒度を高めるUMLダイアグラム計測ツールの開発...	9 1
5.3	計測データの蓄積を活用したプロジェクト予測機構の提案.....	9 4
5.3.1	ベンチマークデータ収集とナショナルデータベースの構築.....	9 4
5.3.2	ベンチマークデータベースの協調フィルタリング.....	9 5
5.3.3	過去プロジェクトデータを活用したプロジェクト予測機構の提案....	9 5
6	おわりに.....	9 9
	謝辞.....	1 0 4
	参考文献一覧.....	1 0 5

## 目次

1 - 1	1984 年実用化の総合ソフトウェア生産システムの構成	4
1 - 2	1984 年提案の開発管理データ自動収集機構	5
1 - 3	1984 年提案の開発管理データ自動収集機構の構成	5
2 - 1	自己申告方式の進捗報告様式例	1 8
2 - 2	設計工程以降の総合的なプロジェクト計測とフィードバックの方式構成	2 0
2 - 3	プロジェクト計測プラットフォーム EPM の構成	2 1
2 - 4	ソースコード規模推移 (プロジェクト立ち上げ時, 全企業重畳)	2 2
2 - 5	ソースコード規模推移 (1 企業例)	2 3
2 - 6	ファイル数と行数の推移 (1 企業例)	2 3
2 - 7	変更頻度・変更規模・変更網羅性 (1 企業例)	2 4
2 - 8	変更者数・変更率・変更回数 (1 企業例)	2 4
2 - 9	ファイル更新履歴分析 (社間比較例)	2 6
2 - 10	障害件数の推移 (全企業重畳)	2 7
2 - 11	障害件数の推移 (X 社例)	2 7
2 - 12	障害件数の推移 (Y 社例)	2 8
2 - 13	障害件数の推移 (X X 社の例)	2 9
2 - 14	障害件数の推移 (Y Y 社の例)	2 9
2 - 15	本来発見すべき工程 - 混入工程 (1 企業例)	3 0
2 - 16	本来発見すべき工程 - 発見工程 (1 企業例)	3 1
2 - 17	コンポーネント毎問題原因分析 (1 企業例)	3 2
2 - 18	社間メール件数の推移計測例	3 2
2 - 19	レビュー記録 (企業別問題記述票) 分析	3 4
2 - 20	コードクローン分析例 (散布図, 6 グループ分)	3 5
2 - 21	コードクローン分析例	3 6
2 - 22	チェックシート分析結果例 (レーダーチャート, 6 グループ分, フェーズ 1)	3 8
3 - 1	プロジェクト管理と計測・フィードバックの論理構造	4 6
3 - 2	論理的な開発構造	4 7
3 - 3	物理的な契約, 管理, マネーフロー構造	4 7
3 - 4	物理的な開発体制の構造	4 8
3 - 5	プロジェクト計測実験の成果を反映したプロジェクト計測モデルの提案	5 7

4 - 1	EA 手法の枠組みの概念図	6 0
4 - 2	機能構成図：Diamond Mandara Matrix (DMM) 記述例	6 2
4 - 3	情報機能関連図：Data Flow Diagram (DFD) 記述例	6 2
4 - 4	業務流れ図：Work Flow Architecture (WFA) 記述例	6 3
4 - 5	実体関連ダイアグラム：Entity Relationship Diagram(ERD) 記述例	6 3
4 - 6	概略作業予定	6 5
4 - 7	記述シート数の推移	6 7
4 - 8	記述ダイアグラム要素数の推移	6 7
4 - 9	AsIs 工程業務全体のシート当たりの要素数推移	6 8
4 - 10	AsIs 工程業務 B のシート当たりの要素数推移	6 8
4 - 11	AsIs 工程業務 D のシート当たりの要素数推移	6 9
4 - 12	ToBe 工程業務全体のシート当たりの要素数推移	6 9
4 - 13	ToBe 工程，業務 A のシート当たり要素数推移	6 9
4 - 14	ToBe 工程，業務 B のシート当たり要素数推移	7 0
4 - 15	ToBe 工程業務 C のシート当たり要素数推移	7 0
4 - 16	AsIs 工程 4 業務全体の記述要素数推移	7 1
4 - 17	AsIs 工程業務 A 記述要素数推移	7 1
4 - 18	AsIs 工程業務 B 記述要素数推移	7 1
4 - 19	AsIs 工程業務 C 記述要素数推移	7 2
4 - 20	AsIs 工程業務 D 記述要素数推移	7 3
4 - 21	AsIs および ToBe 全業務記述要素数推移（全業務積上げ）	7 3
4 - 22	AsIs 工程業務 A 記述要素数推移（フルスケール）	7 4
4 - 23	AsIs 工程業務 B 記述要素数推移（フルスケール）	7 4
4 - 24	AsIs 工程業務 C 記述要素数推移（フルスケール）	7 4
4 - 25	AsIs 工程業務 D 記述要素数推移（フルスケール）	7 5
4 - 26	AsIs 工程の記述ファイル数の推移	7 5
4 - 27	ToBe 工程の記述ファイル数の推移	7 6
4 - 28	AsIs 工程業務 B のファイル数推移	7 6
4 - 29	AsIs 工程業務 D のファイル数推移	7 6
4 - 30	ToBe 工程業務 B のファイル数推移	7 7
4 - 31	AsIs 工程の 1 シートの変化量推移例（A）	7 7
4 - 32	AsIs 工程の 1 シートの変化量推移例（B）	7 8
4 - 33	AsIs 工程の 1 シートの変化量推移例（C）	7 8
4 - 34	AsIs 工程の 1 ファイル（8 シート）の変化量推移例	7 8
4 - 35	AsIs 業務週間記述要素数増加分推移と記述作業予定期間	7 9

4 - 3 6	週間記述要素数増加量の推移	7 9
5 - 1	EVM による進捗報告	8 1
5 - 2	WBS による進捗報告のグラフ化 ( % )( 全工程 )	8 2
5 - 3	作業項目工数比率	8 3
5 - 4	AsIs 作成作業項目工数比率	8 3
5 - 5	ToBe 作成作業工数比率	8 3
5 - 6	業務別生産性 ( ダイアグラム数 / 稼働単位 )	8 4
5 - 7	ダイアグラム要素当たりの単価 ( 相対値 : 全体平均 = 1 )	8 6
5 - 8	ToBe 工程の業務別要素当たりの金額	
	検討稼働を含む場合とダイアグラム作成稼働のみの場合の比較	8 7
5 - 9	ToBe 工程の業務別要素当たりの金額の検討稼働を含む場合と	
	ダイアグラム記述のみの場合の比	8 7
5 - 10	ToBe 工程の業務別要素当たりの金額分布	
	検討稼働を含む場合とダイアグラム作成稼働のみの場合の比較	8 7
5 - 1 1	全工程一貫計測モデルの提案	9 1
5 - 1 2	試作した UML ダイアグラム要素計測ツールの構成	9 2
5 - 1 3	UML 計測ツール出力例 ( ユースケース図設計量推移 )	9 3
5 - 1 4	UML 計測ツール出力例 ( アクティビティ図設計量推移 )	9 3
5 - 1 5	UML 計測ツール出力例 ( ユースケース図設計変更量推移 )	9 4
5 - 1 6	過去プロジェクトデータを活用したプロジェクト予測機構の提案	9 6

## 表目次

1 - 1	開発管理の問題点 (1988 年発表)	6
1 - 2	本論文の目的とアプローチ, 対応策と記述箇所	1 4
3 - 1	リーダー, サブリーダーへのアンケート結果 (部分, n = 26)	4 4
3 - 2	開発体制の階層構造 (1 / 2) (B 社, C 社)	4 9
3 - 3	開発体制の階層構造 (2 / 2) (C 社, D 社, E 社)	5 0
3 - 4	事後のキーマン・インタビュー結果要約 (1 / 2)	5 2
3 - 5	事後のキーマン・インタビュー結果要約 (2 / 2)	5 3
3 - 6	プロジェクト内の立場と計測とフィードバックへの評価, 事後対応	5 6
4 - 1	EA 手法の階層的な表現モデルと EA プロダクト (成果物)	6 1
4 - 2	EA 手法のプロセス体系とマネジメント体系	6 1
4 - 3	対象業務と計画された記述ダイアグラム	6 5
5 - 1	WBS による進捗報告例 (AsIs 工程の部分)	8 2
5 - 2	ソースコードとダイアグラムのメトリクスの視点からの相似性	8 4
5 - 3	データ白書中のデータ分析の基本とする数値 (SLOC と FP)	8 5
5 - 4	UML ダイアグラム計測対象	9 2

## 1. はじめに

### 1.1 研究の背景

はじめに本論文執筆の動機となった背景について述べる[1-1]。本論文の背景に、「計れるものは進歩する」、「計測は何らかの施策の原点」「科学，工学の発展はものごとを定量的に把握することから始まる」[1-2]，あるいは「人は測定しないものを管理することはない」[1-3]，という考え方がある。計測は多くの工学の基礎，あるいは前提条件であり，ISO/IEC15939「ソフトウェア測定プロセス」を紹介した John McGarry らのグループによる「実践的ソフトウェア測定」では「混迷を極めるソフトウェアビジネスの世界に，測定を通じて意思決定のための確かな定量指標を確立しよう！」と訴えている [1-4]。

本論文は，ソフトウェアあるいは広く情報システム構築の生産性と品質の向上を目指すソフトウェア工学の研究の中で，計測をその手段の中心に据えたエンピリカルソフトウェア工学の立場から，この領域の互いに連携した国家計画である文部科学省のリーディングプロジェクト「e-Society 基盤ソフトウェアの総合開発」計画の一つで，EASE (Empirical Approach to Software Engineering: ソフトウェア工学へのエンピリカルアプローチ) と名づけられたプロジェクト[1-5]-[1-7]と，経済産業省の支援で独立行政法人情報処理推進機構 (IPA: Information-Technology Promotion Agency) の中に新規に設立されたソフトウェア・エンジニアリング・センター (SEC: Software Engineering Center) の活動[1-8][1-9]の一環として，両計画の研究員を兼任する立場からすすめた研究成果について述べるものである。

いずれの活動も産学官連携した活動により，実際のフィールドデータを活用しての実践的で実効ある成果と得られる知見，手法，ツールの産業界への普及を目指している。

「ソフトウェア開発に関する定量データの計測」の対象は幅広い。この分野で先駆的な Capers Jones はその対象を 9 項目に整理している[1-2]。

- 1) 運用環境の測定
- 2) 開発中のプロジェクトの測定
- 3) ソフトウェア資産とバックログの測定
- 4) 顧客満足度の測定
- 5) 完了プロジェクトの測定
- 6) ソフト要因の測定
- 7) ソフトウェアの欠陥の測定
- 8) 企業内従業員統計の測定
- 9) 企業内の意識調査

これらの計測結果は分析・評価されて様々な施策の基盤になるが，その目的を粗く分類すると，上記の 5) に代表されるプロジェクトで記録されたデータをプロジェクト終了後に集計するデータ収集，すなわち「ポストプロセス計測」と 2) に相当する進行中のプロジェ



クトのデータの逐次の収集・分析，すなわち「インプロセス計測」がある．定量データ計測を様々な施策の基盤とするために両者は補完関係にあるが，データ収集のための活動のあり方は両者でかなり異なる．

「ポストプロセス計測」ではプロジェクト終了後の集計データを幅広く収集し，一定期間にわたって集積することで，ベースライン分析，ベンチマーク分析ができる．ベースライン分析は，文字通り評価の基準線を設定するもので，様々な施策の効果をプロジェクトごとあるいは経年ごとに評価するのに欠かせない．ベースライン分析が無ければ，生産性一つとっても，様々な施策の効果があつたのか無かつたのか，年々進歩しているのかどうか，進歩の程度はどうか明確にできない．いわゆる「話百篇」のソフトウェア工学になる要因の一つに，この基準線，ベースライン設定の不備があつた．企業内では，この基準線不備のために，生産性や品質向上のための施策への資源投入の正当化が難しく，これらの部門に働く人々が苦戦する状況がしばしば見られた．SEC では，こうしたベースライン分析，ベンチマーク分析のための産業界レベル，国家レベルのデータ収集とその分析を実施している．

一方「インプロセス計測」を主眼とする EASE プロジェクトでは，進行中のプロジェクトの管理，端的に言えば，現場のプロジェクトリーダーとその管理者への直接的な支援を狙って，進行中のプロジェクトの計測とそのダイナミックな分析，プロジェクト進行中の施策への反映に焦点を合わせた．EASE プロジェクトでの「インプロセス計測」のねらいは，現場で工期延伸，予算超過，品質不良など多くの問題に悩むプロジェクトを直接支援したいという動機のほかに，下記のような考えがある．

- 1) 開発終了後の集計データへの不信．開発終了後に人手で集計されたデータには種々の要因で人為的な手が入りやすい．
- 2) 開発終了後の評価では，当該プロジェクトの推進に反映できず，開発担当者のデータ収集への動機づけも難しい．
- 3) 最大の動機として，ソフトウェア開発プロジェクトの実態への一つの認識がある．すなわち，実際のソフトウェア開発プロジェクトは，正常系，すなわち予定通りの進行という部分が極めて少なく，大半が異常系，想定外の事象への対処，それも後手，すなわち後追いの対策に費やされている，という現実である．プロジェクト進行中の計測によって，この現実を少しでも改善するのが「インプロセス計測」に焦点をあわせた基本的な動機である．

製造業では，自動車製造で代表されるようにフロントローディングの考え方が発表されている[1-10]．工程の少しでも前の段階で異常兆候を検出し，先手の対策をとって，できるだけ前の工程で品質を作り込もうという考え方である．先手の対策がコスト的に有利なことは広く認識されている．またその組み立て工程では「インライン計測」と呼ばれる概念が普及し，組み立てライン中の製品，すなわち自動車をミリメートル単位で計測しその場で組み立ての不備を検出，これを組み立てライン中で修正するということが行われるよ

うになった。工業製品としての自動車を、完成してから検査するのではなく、製造工程中から計測して、検出した不備を製造工程中に修正してしまう、という考え方、すなわち完成品検査から製造中チェックへの転換である。こうしたことを踏まえ、筆者および EASE プロジェクトでは、「インプロセス計測」がプロジェクトの異常兆候の早期検出と早期対策に役立つことで現場に支持されると考えた。そして、可視的なプロセスとプロダクトのデータの蓄積を通して、そのプロセス改善に貢献しようと考えた。

本論文はこうした背景のもとに、「ポストプロセス計測」の技術と必要性をふまえつつ、「インプロセス計測」についての新しい試みとその評価・考察について述べるものである。

## 1.2 関連研究

コンピュータのソフトウェアはごくマクロな視点からは、ターゲット・ソフトウェアとサポート・ソフトウェアに分けて考えられる。ターゲット・ソフトウェアは、コンピュータシステムの適用業務、たとえば銀行業務、座席予約業務といった業務を実行するソフトウェアで、サポート・ソフトウェアは、ソフトウェアを開発するソフトウェア、具体的には、ソースエディタ、コンパイラなど意味する。そしてターゲット・ソフトウェアが動く環境を実行環境、サポート・ソフトウェアが動く環境を開発環境と呼ぶようになった。

コンピュータシステムの高度化に伴ってこの開発環境は飛躍的に高度化し、その一部に、ソフトウェア開発プロセスを管理する環境、すなわちソフトウェア開発管理環境という領域が考えられるようになった。ソフトウェア開発プロジェクトが大型化するのに伴い、開発管理に関わる業務が増え、それをコンピュータで支援しようとするものである。本研究はこの開発管理環境の高度化に関するものである。

ソフトウェアの開発管理には、ソフトウェア開発工程の進捗に関する多くの情報を入手し、これを人間に理解できるように可視的に提示してゆく必要がある。その課題は、1) 必要なデータの入手または投入、2) データの分析、そして3) 可視的な提示、のそれぞれにある。そのなかでも開発管理に必要なデータの入手に関して、人手による恣意的な情報の排除、データ入手に関するオーバーヘッドの抑制など課題は大きかった。これはソフトウェア開発管理データの自動収集という課題になったが、コンピュータが紙カード入力を中心とするバッチ処理方式の時代には実現が難しかった。

筆者らは、1980年代前半に我が国のデータ通信用メインフレームコンピュータ開発計画(DIPS 計画)[1-11]の一環として、タイムシェアリング方式(TSS)を用いた会話型のソフトウェア開発環境を実現した際に、ソフトウェア開発管理データの自動収集方式を提案した[1-12]-[1-14]。そしてこの概念は、追って、ソフトウェア・メトリクスを研究するグループによって、ソフトウェア開発管理要領と呼ばれるドキュメントと、これを機械支援する機構として実用化された[1-15]。

図1-1、図1-2に、筆者らの実現した開発環境の構成を示す。コマンドプロセッサ

を介してソフトウェア開発ツール群を起動・操作する会話型の開発環境は実現できたので、コマンドプロセッサの内部からソフトウェア開発プロセスに関する情報を自動収集し、これをデータベースに格納して開発管理に活用しようというものである。図1-3に、開発管理データ自動収集機構の構成を示す。この構成ではプログラムを開発ツールの管理するファイルから取得し、また開発管理データベースには手作業入力データも集めることが考えられていた。

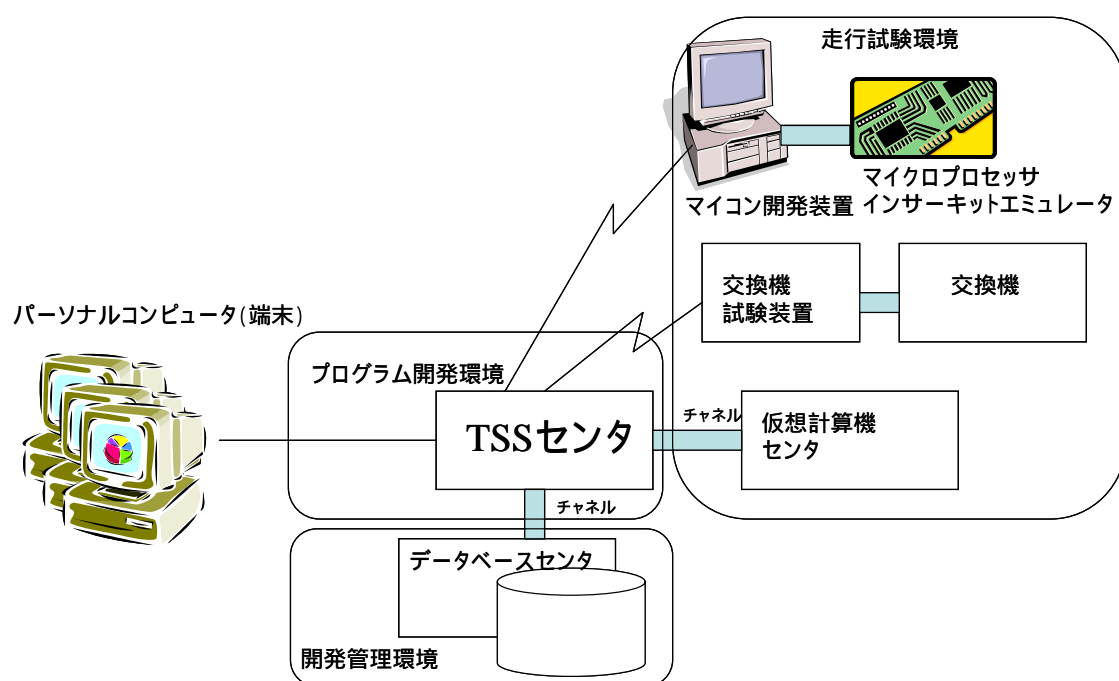


図1-1 1984年実用化の総合ソフトウェア生産システムの構成[1-12]

実用化された PROM(Project Management System)と名づけられた開発管理システムは、開発環境からの自動収集データと手入力のデータを集めて週次でデータを分析、抽出バグ数の推移や残存試験項目数の推移などを20種類を超すグラフや表などの可視的な形で提供し、いわゆる週間進捗状況報告書の作成を機械支援した。また、生産性の年度推移などの分析結果、生産性とバグ数、バグ修正工数などの各種計測要因の相関など統計的手法による分析結果も提供した。

この研究は、会話型のソフトウェア開発環境の実現と同時に開始されている点に歴史的な意義がある。しかしながらこの領域の研究成果はその後高度化する社会のニーズに応えられるような形での普及発展は見られず、20年以上経過した今日でもソフトウェア開発管理を機械支援するという課題は依然として残されたままとなった。一例として、1988年に発表された上述の文献[1-15]に示されている開発管理の問題点という表を表2-1に引用再掲するが、その内容は今日でも全く鮮度を失っていない。

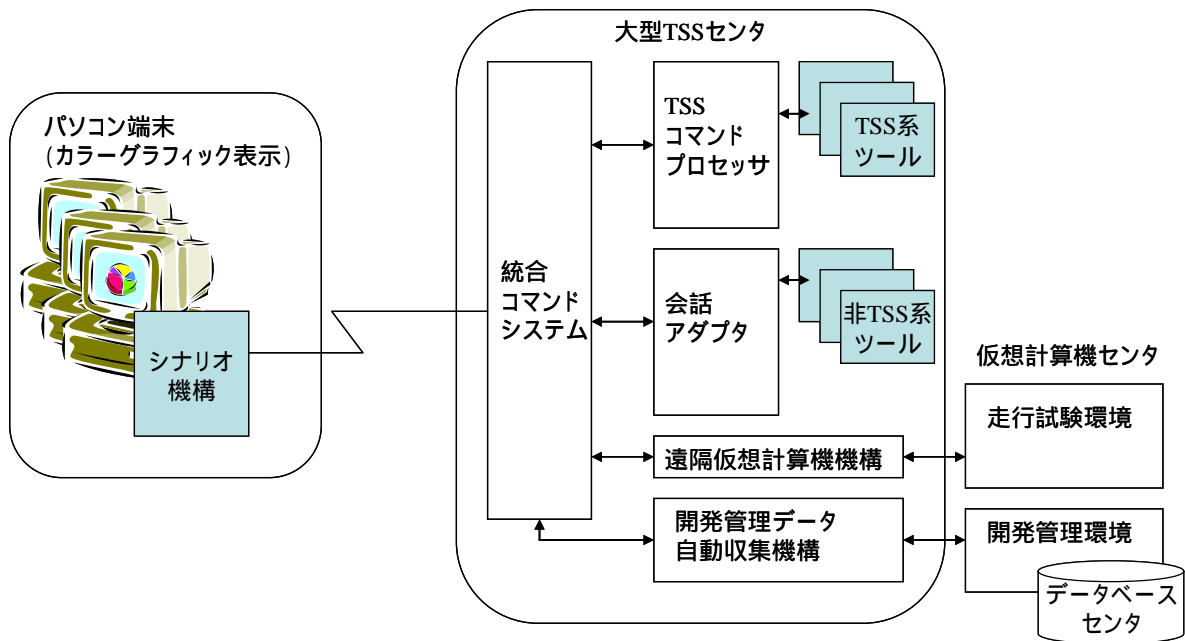


図 1 - 2 1984 年提案の開発管理データ自動収集機構[1-12]

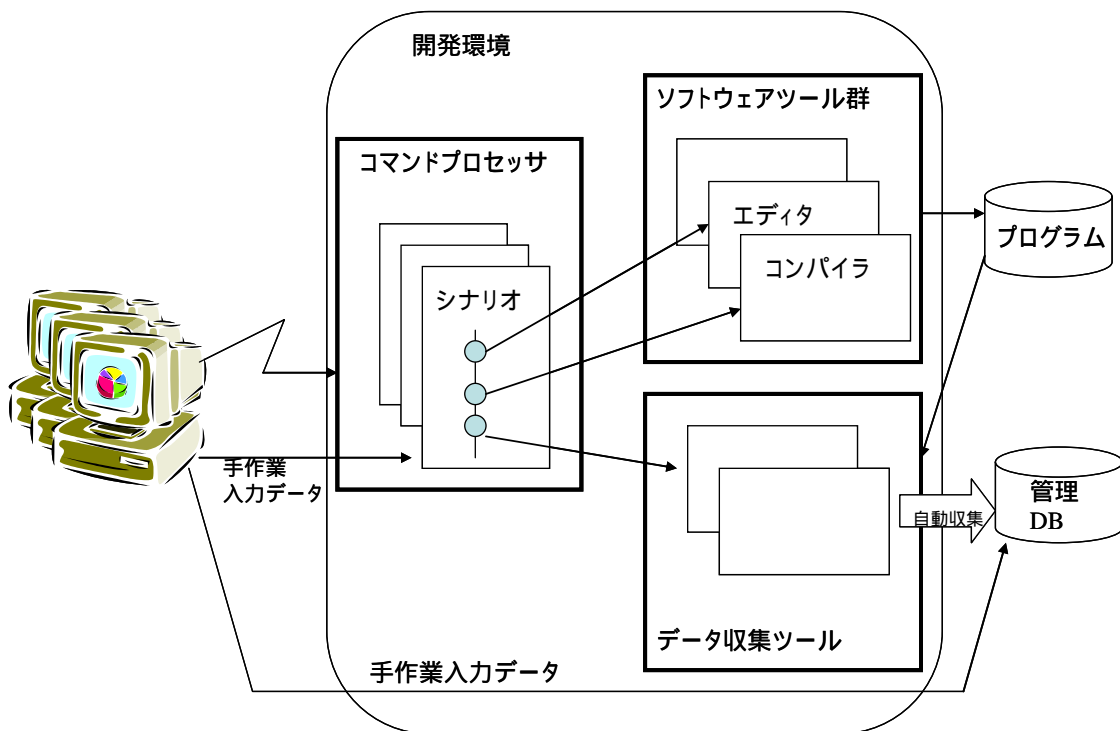


図 1 - 3 1984 年提案の開発管理データ自動収集機構の構成[1-12]

表 1 - 1 開発管理の問題点 (1988 年発表) [1-15]

現象	管理上の問題点	要求される機能
<ul style="list-style-type: none"> <li>・品質不良</li> <li>・納期遅延</li> <li>・コスト増大</li> </ul>	<ul style="list-style-type: none"> <li>・経験と勘に依存した各人各様の管理</li> <li>・計画立案精度が低い</li> <li>・進捗状況把握が不十分</li> <li>・製品品質状況把握が不十分</li> <li>・問題点分析が不十分</li> <li>・フィードバックが不十分</li> </ul>	<ul style="list-style-type: none"> <li>・管理の標準化</li> <li>・精度の高い開発計画立案支援</li> <li>・開発管理データ収集による定量的な作業把握と</li> <li>・管理サイクルの短縮化によるフィードバック</li> <li>・実績データの蓄積・分析支援</li> <li>・組織的な推進</li> </ul>

この領域の実証的な研究発表例は多くはないがそれでも地道な発表努力は続けられており、それらを概観することは意味がある。そこで本研究の位置づけを示すために、以下に1990年代以降の代表的な発表例を追う。

Robert B. Grady は、Hewlett-Packard 社での経験をもとに、ソフトウェアメトリクスへの基本的な考え方とあわせて、豊富な計測例を例示しながらメトリクスの活用法をその狙いごとに示している[1-16]。メトリクスのねらいは戦術的活用と戦略的活用に分けられ、概ね前者がインプロセス計測に、後者がポストプロセス計測に対応している。戦術的活用としては顧客満足 of 極大化、工数・工期の極小化、障害の極小化のためのプロジェクト管理への活用法について述べている。戦略的活用では、障害分析、プロセス改善、ソフトウェア事業の健全性の計測などへの活用について述べている。

John McGarry らは、ISO/IEC 15939 として規格化されている Practical Software Measurement (PSM) と呼ばれる計測プロセスモデルを解説した著書の付録として3つの例を示している[1-4]。1つは測定情報モデルの一般化した実施例で、実測例と推定される豊富な計測例が示されている。2番目は米空軍の人事情報システムプロジェクトにおける適用例、3番目は事務用の複合コピー機のソフトウェア開発事例である。

一般化した計測例では、横軸を時間軸として、10種を超える計測とグラフ化の例が示され、ソフトウェアプロセスの計測に関わる基本的なイメージを提示している。縦軸の項目の主なものを挙げると、試験完了ユニット数、結合試験が終わった統合ユニット数、プロジェクト要員数、コストの予定からの変動率、作成したコード行数、機能要求の要求数、その変更数、試験における欠陥数、ソース行数当たりの欠陥数、人月当たりの作成ソースコード行数、などの推移で、そのうち多くで計画と実際、すなわち予実を見えるようにしている。

米空軍の人事情報システムの例では、対象システムのコンテキストを示し、当該システムが工程のマイルストーンのレビューで不合格になってから回復しリリースするまでの状

況を計測データで追っている。やはり横軸を時間軸とした12種以上のグラフと、いくつかの時間軸以外の分析グラフ例で約3年間のプロジェクト推移を示している。主なものを挙げると、要員数、人月による稼働工数、実装ユニット数、実装済み画面とレポート数、ソースコード行数、問題報告件数、消化したテストケース数、インストール基地数などの推移である。時間軸でない分析として、問題報告件数の分類、構成品目別ファンクションポイント当たりの問題報告件数、おなじく構成品目あたりの手戻り稼働(人月)などがある。可能なものは予実がわかるように表示されている。

それまでのガントチャートによるマイルストーン管理と人手によって収集された不正確なデータによる判断を改め、計測とその分析による的確な判断、処置結果の正確でビジュアルな把握を可能にする新しいプロジェクト管理法によってプロジェクトを成功に導いた課程が計測と分析の実例と合わせてケーススタディとして示されている。

複合コピー機の例は、先端的で大規模な組込みソフトウェアの開発事例である。18ヶ月という短い製品開発期間でラピッドアプリケーション開発(RAD)を採用した。ガントチャートのほかに、横軸を時間軸とする7種類のグラフによってプロジェクト進捗を把握し判断に活用した例が示されている。グラフは、作成したユニット数、統合したユニット数、実行したテスト手順数、要員数、詳細設計完了ユニット数、行数あたりのコードインスペクションで発見した欠陥数、単体テスト完了規模(行数)などの推移である。このほかにモジュール別の欠陥密度なども分析されている。こうした計測と分析がプロジェクトのトラブルの特定に役立ったことが報告されている。

Lawrence H. PutnamとWare Myersは、ソフトウェア開発で基本となる5つのメトリクスについて解説している[1-17]、この中では6,300の計測データが反映されているということである。5つの基本メトリクスとは、時間(スケジュール)、工数(人月)、機能量(プログラムサイズなど)、信頼性(欠陥率)、プロセス生産性である。これらの計測データの活用法について多くの記述がある。その対象にはポストプロセス計測とあわせてインプロセス計測も含まれ、企画段階から機能設計、調達、人員配置、製造などソフトウェア開発の様々な局面でのメトリクスの活用法を解説している。米国陸軍の永年のシステム開発での経験データをもとに時間軸を横軸とした予測グラフを算出し、プロセスの計測値で予実管理をすすめる方式を紹介している。

Linda M. LairdとM.Carol Brennanはソフトウェアの計測と評価というテーマに対して、何を測るか、どうやって測る対象を決めるかなどソフトウェア計測の基礎を紹介したのち、10種類の計測分野について詳しく解説している[1-18]。そして最後に計測結果のプロジェクト管理への効果的な提供法(プレゼンテーション法)について述べている。計測分野として、プログラムのサイズ、複雑度、工数、欠陥、信頼性、レスポンスタイム、プロジェクト進捗、アウトソーシング先の状況、要員の給与状況、ベンチマークを挙げている。プロジェクト進捗の計測では横軸を週次などの時間軸として、ガントチャートによるマイルストーン計測のほか、ソースコード行数、テストの進捗をみるための障害件数、障害修正

件数，障害対応件数，などの推移や予実推移のグラフ化を紹介している．

設計工程での計測に対する研究として松村崇史らは，設計工程の成果物計測の時系列変化について計測実験例を交えて報告している[1-19]．オブジェクト指向設計において，計測ツールを作成して，ひとつの設計ドキュメント中のメトリクス（記述要素数）の改版ごとの推移を13版まで追い，そのうちサイズに関するメトリクスの推移が設計作業進捗のマイルストーン把握に適している可能性を示唆した．

### 1.3 研究の位置づけ

ソフトウェア・プロジェクトのインプロセス計測とフィードバックについては上述のように早くから研究に着手され，少ない中にもいくつかの精力的な実証的報告があるにも関わらず，産業界ではその諸課題を解決する手段として受け入れられて来なかった．

産業界にとってソフトウェア開発管理に関わる施策は直接企業のマネジメントに関わり適用の最初の段階から影響範囲が大きく，影響範囲を限定して試行可能な小規模なソフトウェアツール群の導入と異なり，多くの場合施策を導入する側にとって負荷が重い．小規模な試行では効果を顕在化させるのが難しく，新施策本格導入のコストやリスクが大きい．単に「よさそうだ」という程度での適用は難しく，このコストやリスクの壁を越える説得力ある利便性や利得性のエビデンスが求められる．

筆者らはEASEプロジェクトやSECの活動の中で関連研究にあるようなこれまでの研究成果が産業界に受け入れられない要因を次のように考えた．

- 1) 従来の研究は製造工程中心で設計工程の計測の方法論，事例は貧弱で，さらに要求定義工程，保守工程を含んだ一貫提案になっていない．
- 2) 事例として発表されるプロジェクトのコンテキスト（文脈情報）が産業界の共感と呼ぶほど一般性を持っていない．発表事例はコンテキストや産業構造を反映した分析に欠け，単独では共感を持つほどの一般性がない．また発表事例は少なく散発的なため，一般性を主張できるほどの事例の積上げがない．
- 3) 海外からの発表は特に軍隊や特別な企業の例などプロジェクトのコンテキストが日本と異なり，日本の産業界の共感と呼ぶことが少ない．
- 4) データの分析に重きがあり，必ずしもデータ収集の負荷の軽減，自動収集によるデータ精度向上に焦点があっていない．
- 5) 最新のソフトウェア開発環境，データ分析技術，ツールを反映していない．方式評価が単発的で，最新技術を統合して適用し総合的に評価する試みに欠けている．
- 6) コンテキストやそれぞれの現場に働く人間の振る舞い，そして産業構造を深く分析し，これと結びつけた有用性のエビデンスを提示していない．適用を検討する側にとって現実味，リアリティの高いエビデンスの提示ができていない．
- 7) 少ない計測機会を活用しながら一般性の高い評価結果，エビデンスを提示する工夫に欠

けている。

8) 実証したことの再現性,あるいは普及を図るための工夫が図られていない。実証に用いたツールと手法の提供,普及が考えられていない。

例えば Robert B. Grady の発表[1-16]は Hewlett-Packard 社という組織での経験を例としているが,この大きな機構の中で実際にソフトウェア開発がどのように行われ,ソフトウェア開発プロジェクトにかかわる要員がどのような役割でどのような動機づけで働き,それと計測結果がどのように相互作用しているかは明らかにされていない。実際のプロジェクト運営から少し離れた位置からのマクロなマネジメント法の検討や学術的な観測の位置づけが強い。

John McGarry らの発表[1-4]の付録の最初の例は,計測項目と分析法の一般的な紹介で,プロジェクトのコンテキストとの対応,データ収集方法について触れられていない。2番目の例は空軍の人事情報システムという一般性に欠ける事例で,また,プロジェクトマネジメントの方法について共感を得るような一般的な紹介がない。3番目の例は組み込みソフトウェアの短期開発でマクロなコンテキストは一般性が高いにも関わらず,やはり実際にプロジェクトマネジメント体制との対応の分析・紹介が無い。またデータの収集法,特に自動収集に触れられていない。また,これらの事例で用いられた計測ツール,分析ツールの提供・普及努力,ここで行われたことの再現性については示されていない。

Lawrence H. Putnam と Ware Myers のソフトウェアメトリクスに関する豊富な解説[1-16]は,時間軸を横軸とする多くの計測例が示されているが,その活用対象は人員配置や調達など計画段階やプロジェクト評価段階に重点がありインプロセスのプロジェクト制御については掘り下げた紹介がない。また米国陸軍での永年の経験を元にした予測曲線を用いた予実管理手法も今日のソフトウェア開発現場の状況と対比して現実性(リアリティ)が高いとはいえない。米国陸軍という特定領域の豊富な経験を背景とした一般的管理手法の提示という形になっていて,実際のプロジェクトのコンテキストや人間の振る舞いと対応づけて考えることが難しい。

Linda M. Laird と M.Carol Brennan の発表[1-18]の中では,プロジェクト進捗に関する計測についてインプロセス・メトリクス(in-process metrics)と呼んで,ソフトウェアプロジェクトに有用な下記の5つの項目を挙げている。そして横軸を週次などの時間軸として,ガントチャートによるマイルストーン管理,ソースコード行数,テストの進捗をみるための障害件数,障害修正件数,障害対応件数,テスト項目消化数,障害解決数,障害滞留数などの推移や予実推移のグラフ化など基本的な手法を紹介している。

- 1) プロジェクトのマイルストーン
- 2) コードの集積(ソースコード行数)
- 3) テストの進捗
- 4) 障害の修復と解決



## 5) プロセスの効率

しかしながらその手法は一般化されていて、実際のプロジェクトとの対応、プロジェクト運営への反映、については示されていない。

松村崇史らの設計工程に関する研究[1-19]は設計工程の成果物測定に焦点をあてた先駆的で数少ない発表例の一つであるが、1枚の設計ドキュメントの改版を13版に渡って追った事例で、考え方の検証に留まっている。行われたプロジェクトのコンテキストとの対応、コンテキストの一般性、ツールの提供については触れられていない。

本論文の研究ではその狙いを、産業界にインパクトを与え直接産業力強化に結びつけることに定め、表1-2に示すように下記の目的を設定した。

- 1) 実証する方式の有用性、利便性について、ソフトウェア開発現場の条件、開発に従事する人々の振る舞い、プロジェクトのコンテキスト(文脈)、産業構造にびったり沿った、産業界から共感の得られる現実性(リアリティ)の高いエビデンスを獲得する。
- 2) 実証的な計測実験にもとづいた要求定義工程から製造工程、保守工程まで、一貫した計測とフィードバックの方式を提案する。
- 3) はじめから再現性の高い手法、道具(ツール)を工夫し、1つの実証事例獲得後、手法とツールを普及させ、実証事例を積み重ねてゆける、すなわち実証事例の拡大再生産が可能な手法と道具立てを作り出す。

この目的に沿って本論文では次のアプローチをとった。

- 1) 設計工程以降総合試験工程までのプロジェクト計測機会を得て、ツールを介した計測とあわせてソフトウェア開発現場に密着した継続的なプロジェクト観察を実行し、総合的な計測と計測の有用性に関する多角的なエビデンスを収集した。また要求定義工程でのプロジェクト計測機会を得て、調達側の一員としてプロジェクトに加わり、計測・分析作業を進めながら計測の有用性を明らかにするエビデンスを収集した。
- 2) 設計工程以降の総合的なプロジェクト計測と、要求定義工程のプロジェクト計測の2つの計測機会を得て、その成果から全工程一貫計測とフィードバックのモデルを提案した。
- 3) プロジェクト計測と分析のためにはじめから、普及を考えた再現性の高い道具立てを実現し、最初の実証実験ののちその道具立ての普及とこれを用いた実証実験を拡大再生産してゆく施策を開始した。

本論文の研究ではこのアプローチに沿って、つぎのような対応策ですすめた。

産業界にとって現実性(リアリティ)のあるエビデンスを収集・獲得するために、下記のようにすすめた。

- 1) 実証実験の焦点を日本の典型的なソフトウェア開発環境にあてた。すなわち大手ソフトウェア企業等による中規模のマルチベンダ広域開発プロジェクトと、比較的標準的な

政府系の調達によるプロジェクトである。(前者について2章, 3章で, 後者について4章, 5章で述べる)

- 2) 計測対象プロジェクトにおいて, ほとんどすべての進捗会議, 分析データフィードバックイベントに参加し, 観察型の手法で, 現場の人間の振る舞いや産業構造を反映したエビデンスの獲得につとめた。また政府系プロジェクトでは調達側の一員としてプロジェクトに参加した。
- 3) 対象プロジェクトを深く観察・分析し, 少ない計測機会からより多くの一般性を引き出す努力をした。

いわば発展した産業構造を反映して高度に複雑化したソフトウェア開発体制に「染み渡るような観察(Pervasive observation)」を行い, おなじく「染み渡るような効果(Pervasive effects)」を実証することを心がけた。

全工程一貫計測モデルの提案のために下記のようにすすめた。

- 4) プロジェクト計測の実証実験について製造工程から着手し, 対象を設計工程, 要求定義工程に拡張し, 保守工程も含めた全工程一貫計測モデルを提示した。
- 5) まず製造工程を中心に設計工程以降の下流工程に焦点をあわせ, 一般性の高い中規模プロジェクトでインプロセス計測とフィードバックの実証実験の機会を得て, 産業界の共感を得るエビデンスの獲得を図った。次に要求定義工程を対象とし, わが国情報システム市場の10%を占める政府調達の中で, 「業務・システム最適化計画」と名づけられた工程管理の標準化のすすんだ一般性の高いシステム再構築計画のひとつで計測機会を得た。この2つの計測機会の成果を統合して, 一貫計測モデルを提案した。

実証事例を積上げる再生産プロセス実現のために下記のようにすすめた。

- 6) プロジェクト計測にあたって, 新たに開発されたデータの自動収集機構を適用し, 計測負荷の抑制をはかるとともに, 計測活動の再現性を高めた。
- 7) 設計, 製造工程について, 最新技術, ツールを多数用いた約2年半の総合的な実証実験を組織した。すなわち, オープンシステムによる環境の採用, 新たなプロジェクト計測プラットフォームの開発, 構成管理システム, 障害追跡システム, メーリングリスト管理システムからのプロセスデータの自動収集方式を用いた。これらと, コードクローン分析, 協調フィルタリングといった新しい技術を組み合わせて総合的なプロジェクト計測を実現した。また, SECの活動で収集された1,000プロジェクトを超えるベンチマークデータを活用した。おなじくSECの活動で生み出されたプロジェクト可視化のためのチェックリストを適用した。EASEプロジェクトに組織された高度な分析グループによって計測データの分析を実施した。

また, 要求定義工程について, EA(Enterprise Architecture)ツールと呼ばれるダイアグラム記述ツールと最新の描画システムの計測機能を活用した。さらに, 設計工程につ

いて、新たに UML ダイアグラムの計測ツールを試作した。

- 8) 実証実験にひきつづいてツール群の配布キットを提供するなどツールと手法の普及を図り、実証機会の拡大再生産サイクルを実現するための具体的な事業を開始した。

これらに対し本論文を表 1 - 2 に示すような対応で記述した。2 章，3 章で設計・製造工程での計測実験，4 章，5 章で要求定義工程での計測実験に関して記述した。それぞれ 2 章と，4 章で計測対象と再現性のある計測実験の内容について示した。3 章，3.1 - 3.6 および 5 章で計測実験の評価について述べ、一貫計測モデルについて，3.7 と 5.2 で述べた。そのほか，5.3 で，これらの計測データを過去データとして蓄積し，これを活用したプロジェクト予測機構を提案した。

本論文の意義は、一般性の高い実プロジェクトによるインプロセス計測とフィードバックの環境を再現性の高い形で実現し、実験と評価を工夫することにより、その有用性についてこれまで難しかった産業界の共感の得られる現実性の高いエビデンス獲得し、またこれまで研究の難しかった要求定義工程での計測・評価事例を作り、これらを背景に実証事例の拡大再生産サイクルの糸口を掴んだ点にある。実験においては、計測と分析のために多くの新しい技術を統合し、プロジェクト全工程わたる詳細な観察、産業構造を反映したきめ細かいエビデンスの収集を行った。要求定義工程の計測については政府の業務・システム最適化という新しい枠組みの中で、プロジェクト計測という新しい試みを実現した。

最後にこれらとあわせて、本論文のような研究の実行に必須の研究環境について考察し、必要な産学官連携体制構築上の課題について明らかにした。そして、この環境を実現する仲介者の役割に着目してその条件を明らかにした。

なお、本論文で示す計測実験のなかで筆者は次の役割を果たした。

#### 1) 設計・製造工程での総合的な計測実験

開発対象プロジェクトの調達側のソフトウェアエンジニアリング施策担当

プロジェクト計測とフィードバック計画全体の組織化，組織運営

計測ツールの運用調整（分担）

全プロジェクト会議への参加，フィードバックイベントへの参加（プロジェクト観察）

計測プラットフォーム実用化計画の管理（分担）

共同研究の一員：プロセス計測とフィードバック，協調フィルタリング技術を活用したプロジェクト予測，コードクローン分析

SEC ベンチマークデータの収集に関する部会の一員（SEC 定量データ分析部会）

SEC チェックリスト作成のための部会，およびこれを用いたヒヤリングと分析・フィードバック作業の一員（SEC プロジェクト見える化部会）

#### 2) 要求定義工程

調達側のプロジェクト管理者の一人（IPA 情報化統括責任者（CIO）補佐官）

計測計画の立案と実施，データの収集と分析のすべて

### 3) 設計工程

UML ダイアグラム計測ツールの試作（実装はソフトウェア企業に委嘱）

表 1 - 2 本論文の目的とアプローチ，対応策と記述箇所

本論文の研究の目的	研究のアプローチ	具体的対応策	本論文の記述(注)
産業界の実フィールドに立脚したプロジェクト計測の有用性を実証する現実性(リアリティ)のあるエビデンスの収集.	ソフトウェア開発現場や諸条件, 産業構造を反映し距離なく現実性の感じられるエビデンスを獲得する.	一般性の高い実証プロジェクトの設定. ツールによる計測とプロジェクトに密着した深い観察の統合. 少ない計測機会から一般性を引き出す多面的な評価と考察. (染み渡るような観察と染み渡るような効果の実証: Pervasive observation, Pervasive effect)	(計測・フィードバックの評価・考察)  3章 (設計・製造工程) 3.1 - 3.6 5章 (要求定義工程) 5.1
要求定義工程から製造工程, 保守工程まで, 一貫計測モデルの提案.	設計・製造工程に対し例を見ない総合的な計測実験を実施. これまで難しかった要求定義工程の計測と評価を実施. これらにもとづいた全工程一貫計測モデルを提案する.	計測対象を製造工程から着手し設計工程, 要求定義工程に拡張. 要求定義工程の計測では, 政府系の「業務・システム最適化計画」の一つを対象に設定. 2つの計測実験の成果をもとに一貫モデルを構築.	(モデルの提案)  3章 (設計・製造工程以降) 3.7  5章 (要求定義工程を含む全工程一貫モデル) 5.2
計測とフィードバックの再現性, 実証事例を積上げる拡大再生産プロセスの実現に貢献するプロジェクト計測実験の実施.	当初から再現性の高い道具立てを工夫・適用して実フィールドでの計測実験を実施する.	一般性の高い最新の環境で最新の道具立てを総合的に適用. 計測ツール配布キットの実現と配布のほか, 実証事例積上げ拡大再生産のための事業を開始.	(計測実験の実施)  2章(設計・製造工程)全  4章(要求定義工程)全

注) このほかに, 1章: 研究の背景, 関連研究, 位置づけ, 5.3: で計測データの蓄積を活用したプロジェクト予測機構の提案, 6章: 「おわりに」で本研究のような実践的研究実現に必須の環境である産学官連携の枠組みに関する考察をまとめている.

## 1 章 参考文献

- 1-1. 神谷芳樹：EASE プロジェクトにみる計測・定量化の実践，*日経ITプロフェッショナル*，2005年3月 pp.92-97，日経BP社，2005.3.
- 1-2. Capers Jones 著，鶴保証城，富野壽監訳：ソフトウェア開発の定量化手法 第2版，構造計画研究所，P.546, 1998.4.
- 1-3. Michael A. Cusumano 著，サイコム・インターナショナル監訳：ソフトウェア企業の競争戦略，ダイヤモンド社，P.445, 2004.12.
- 1-4. John McGarry et.al: Practical Software Measurement, Piason Education, 2002.  
( John McGarry ほか著，古山恒夫，富野壽監訳：実践的ソフトウェア測定，構造計画研究所，P.250，2004.)
- 1-5. 井上克郎，松本健一，鶴保証城，鳥居宏次：実証的ソフトウェア工学環境への取り組み，*情報処理*，45巻7号，pp.722-728，2004.7.
- 1-6. EASE プロジェクト <http://www.empirical.jp/>
- 1-7. e-Society 基盤ソフトウェアの総合開発 <http://cif.iis.u-tokyo.ac.jp/e-society/>
- 1-8. 嶋田 隆・祝谷和宏：ソフトウェアエンジニアリングセンター構想について，*情報処理* 45巻4号，pp.367-371，2004.4.
- 1-9. ソフトウェア・エンジニアリング・センター  
<http://www.ipa.go.jp/software/sec/index.php>
- 1-10. 藤本隆宏：能力構築競争，日本の自動車産業はなぜ強いのか：中央公論新社，P.406, 2003.
- 1-11. 戸田 巖，松永俊雄：電電公社のコンピュータ開発，*情報処理*，Vol.44, No.6, pp.631-639, 2003.6.
- 1-12. 伊東洋一，佐藤匡正，長野宏宣，神谷芳樹：総合ソフトウェア生産システムの実用化，*日本電信電話公社電気通信研究所，研究実用化報告* 33巻12号，pp.2879-2893，1984.12.  
(花田收悦編，ソフトウェアの計画と管理，第5章に全文再掲，日科技連，pp.318-326，1987).
- 1-13. Tadamasato Satoh, Hironobu Nagano, Yoshiki Mitani: An Integrated Software Development System for Data Communication Systems Utilizing Various Types of Computers, *Review of Electrical Communications Laboratories (Review of ECL)* Vol.33 No3 1985.
- 1-14. 下田博次：ソフトウェア工場，見えない工業製品の生産と労働，*東洋経済新報社*，P.232, 1986.
- 1-15. 釜范祐治，笠原和男，小田英雄，高橋宗雄：ソフトウェア開発管理方式，*NTT 電気通信研究所，研究実用化報告* 37巻12号，pp.785-794，1988.12.
- 1-16. Robert B. Grady: Practical Software Metrics for Project Management and Process Improvement, Prentice Hall PTR, P.270, 1992.

- 1-17. ローレンス・H・パトナム, ウエア・マイヤーズ, 山浦恒央訳: はじめて学ぶソフトウェアメトリクス, 日経 BP, P.348, 2005.  
(Lawrence H. Putnam, Ware Myer: Five Core Metrics, 2003).
- 1-18. Linda M. Laird, M.Carol Brennan: Software Measurement and Estimation: A Practical Approach, *IEEE CS*, A John Wiley & Sons, Inc., P.257, 2006.
- 1-19. 松村崇史, 島 和之, 松本健一, 鳥居宏次: オブジェクト指向分析・設計ドキュメント計測システムの作成, *信学技報*, SS96-6, pp.9-16, 1996.7.

## 2. 総合的なソフトウェア開発プロジェクト計測実験（設計工程以降）

### 2.1 計測対象プロジェクトの概要

計測対象プロジェクトは、実験用の「プローブ情報プラットフォーム」を開発するもので、SECによってソフトウェアエンジニアリング手法の実証プロジェクトとして選択され、「SEC先進プロジェクト」と名づけられた[2-1][2-2]。これは政府予算による中規模プロジェクトで2005年春に開始された。開発体制として、自動車メーカー、ITベンダなど7社が法律（鉱工業研究組合法）にもとづく研究組合を構成して開発し、実際に完成したシステムを用いてプローブ情報システムとしてのリアルタイムの実験を行う。システムの入力は、乗用車やバス、トラック、タクシーなどの各種車両をプローブカーとした車両位置情報などで、出力はこれらの情報をクレンジング、融合処理したリアルタイムの国民生活に有意な情報である。情報入力側のシステムはタクシー配車システム、バス運行管理システム、物流トラック運行管理システムなど、それぞれの事業者の既存のシステムを用い、これらに新たな手を加えないのが特徴である。開発と実験の期間は2年半で、この間に2フェーズのシステム開発と公開デモと名づけた実証実験が行われた。開発費用は準備期間から実験まで約20億円である。

開発形態は広域マルチベンダ開発で、研究組合を構成する大手ITベンダ6社が分担する。そのうち1社がもっぱらプロジェクトマネージャ（PM）の役割を果たす。また研究組合の1社は大手自動車メーカーでユーザの視点からプロジェクトの要求定義や評価の役割を果たす。研究組合を構成する各企業はこの領域でライバルであり、本計画ではお互いに競争領域と協調領域を峻別し、競争領域では情報を秘匿し、協調領域では情報共有を図りながらすすめた。たとえば、詳細設計書、ソースコード、プログラム出来高（ソースコード行数）は各社間で秘匿される。PMだけは各社の出来高を知ることが出来る。この開発方式は、組合を構成する各企業から内部を公開することの難しい最新技術を持ち寄り、全体で最高水準のシステムを組み上げようと意図された新しい共同開発方式である。このような環境でPMにはある程度のブラインド・マネジメントが強いられる。各社単体開発工程は自己申告ベースの予実管理で進められ、社間結合試験工程になって共通の結合試験環境に各サブシステムが統合されてから開発された各プログラムの状況（主に試験工程の状況）を皆で共有することになる。図2-1に自己申告方式の進捗報告様式例を示す。工程毎に計画時に予定を申告し、週次で実績、累積、消化率を報告する。報告値は設計工程では作成ドキュメントのページ数、製造工程ではモジュール数、試験工程では試験項目数である。申告に対するエビデンスの提示は求められない。

ソフトウェアの開発対象はLinuxサーバ上でRDBを使用するC/C++によるアプリケーションと、情報表示用のPC上のソフトウェアである。



プログラム設計工程進捗報告書						会社名: 報告書																	
NO	タスク名	生産物	管理項目(単位)	総数	終了日	予実値	6月						7月										
							6/1	6/8	6/15	6/22	6/29	7/6	7/13	7/20	7/27	8/3	8/10						
						予定																	
						実績																	
						累積																	
						消化率																	
						予定																	
						実績																	
						累積																	
						消化率																	

コーディング進捗報告書						会社名: 報告書																	
NO	タスク名	生産物	管理項目(単位)	総数	終了日	予実値	6月						7月										
							6/1	6/8	6/15	6/22	6/29	7/6	7/13	7/20	7/27	8/3	8/10						
						予定																	
						実績																	
						累積																	
						消化率																	
						予定																	
						実績																	
						累積																	
						消化率																	

単体試験工程進捗報告書						会社名: 報告書																	
NO	タスク名	生産物	管理項目(単位)	総数	終了日	予実値	6月						7月										
							6/1	6/8	6/15	6/22	6/29	7/6	7/13	7/20	7/27	8/3	8/10						
						予定																	
						実績																	
						累積																	
						消化率																	
						予定																	
						実績																	
						累積																	
						消化率																	

社内結合試験工程進捗報告書						会社名: 報告書																	
NO	タスク名	生産物	管理項目(単位)	総数	終了日	予実値	6月						7月										
							6/1	6/8	6/15	6/22	6/29	7/6	7/13	7/20	7/27	8/3	8/10						
						予定																	
						実績																	
						累積																	
						消化率																	
						予定																	
						実績																	
						累積																	
						消化率																	

図 2 - 1 自己申告方式の進捗報告様式例

2.2 設計工程以降を対象とした総合的なソフトウェア・プロジェクト計測実験の実施  
 実施したプロジェクト計測実験の概要について示す[2-5]-[2-11]．本節は，3章，4章で示す本論文の主題となる考察や実証実験の背景を示すものであるが，この記述に学術的な意義があると考えて示した．これまで学術論文の領域ではここで示すような総合的なプロジェクト計測とフィードバック実験について，その全体を俯瞰できるような形での発表は稀であった．ここで示した実験においても，参加した研究者によって個別の研究テーマに沿った学術発表は多くなされたが，全体像を示す発表はできなかった．エンピリカルソフトウェア工学の立場では，そもそもソフトウェア開発プロジェクトは個別性が高いので，性急に一般化，あるいは平準化した議論を展開するのではなく，個々の事象についてコンテキストやプロジェクトにかかわる人間の振る舞いを明らかにし，その事例を積上げてゆくことの重要性が指摘されている[2-3][2-4]．筆者はこうした研究姿勢を背景に，計測実験の全体を組織した立場からその全体像を示すものである．計測データは各組合企業の開発

現場から各企業単位に収集し研究のために SEC に集めて分析された。分析作業はツールや方法論にあわせて次の分析グループを組織して行われた。

1) EPM 分析グループ (EASE プロジェクト：奈良先端科学技術大学院大学)

EPM およびその拡張ツールをおもな道具として、ファイル更新状況、障害発生状況、メール送信状況に関する計測と分析をおこなった。またレビュー記録を分析した。次のコードクローン分析についても支援した。(筆者は直接分析作業には参加していない)

2) コードクローン分析グループ (EASE プロジェクト：大阪大学)

ソースコードから CCFinder を用いてコードクローン分析を行った。(筆者は直接分析作業には参加していない)

3) チェックシート分析グループ (SEC：IT プロジェクトの見える化部会，筆者参加)

自己診断シートおよびインタビュー用チェックシートにより、インタビューによる Q&A によりプロジェクト状況の可視化を図った。

4) プロジェクト会議参加グループ (SEC：筆者参加)

主要なプロジェクト会議に継続的に参加し、上記では収集できないプロジェクトコンテキスト情報を収集した。公式の進捗会議における自己申告レベルの進捗報告情報を得た。また、工程の区切り毎の PM と各社別の品質評価会議、そしてほとんどすべてのフィードバック会合に参加した。

組合各社へのフィードバックは上記のテーマ毎に行われた。計画 1 年目 (フェーズ 1) では、データは大略週次に収集し、各社へのフィードバックは 2~3 週毎に行い意見交換の会合をもった。分析結果は社間の情報秘匿に配慮して各社毎に、そして PM には全体が俯瞰できる形でフィードバックされた。2005 年 8 月から翌年 3 月末まで 9 ヶ月間に主なフィードバックイベントが 25 回あり、分析グループから組合各社に提供されたグラフ等の分析シートが合計 1,000 シート (1 シート 1 グラフ) を超えた。(PM にはほぼ全社分を提供しているので全体量は物理的にはこの倍となる)

計画 2 年目 (フェーズ 2) では、フェーズ 1 の成果を反映してデータ収集とフィードバックのサイクルを週次とし、この活動をプロジェクト運営に不可欠のものとして組み込んだ。フィードバックは 2006 年 7 月から翌年の 2 月まで 8 ヶ月間で、提供された分析シートは 2,400 シートに達した。このほかに数 100 枚の質問票や指摘事項一覧、これらへの回答がやり取りされた。(おなじく PM 分を考えるとフィードバックドキュメントの量は物理的にはこの 2 倍となる)。

フェーズ 2 に引き続いて約 2 ヶ月の短期開発で公開デモ用プログラムが開発された。ここではそれまで行われていた自己申告レベルの報告は不要とされ、この計測データだけでプロジェクトが運営された。

この計測実験で図 2 - 2 に示す次の 5 種の計測方式を用いた。

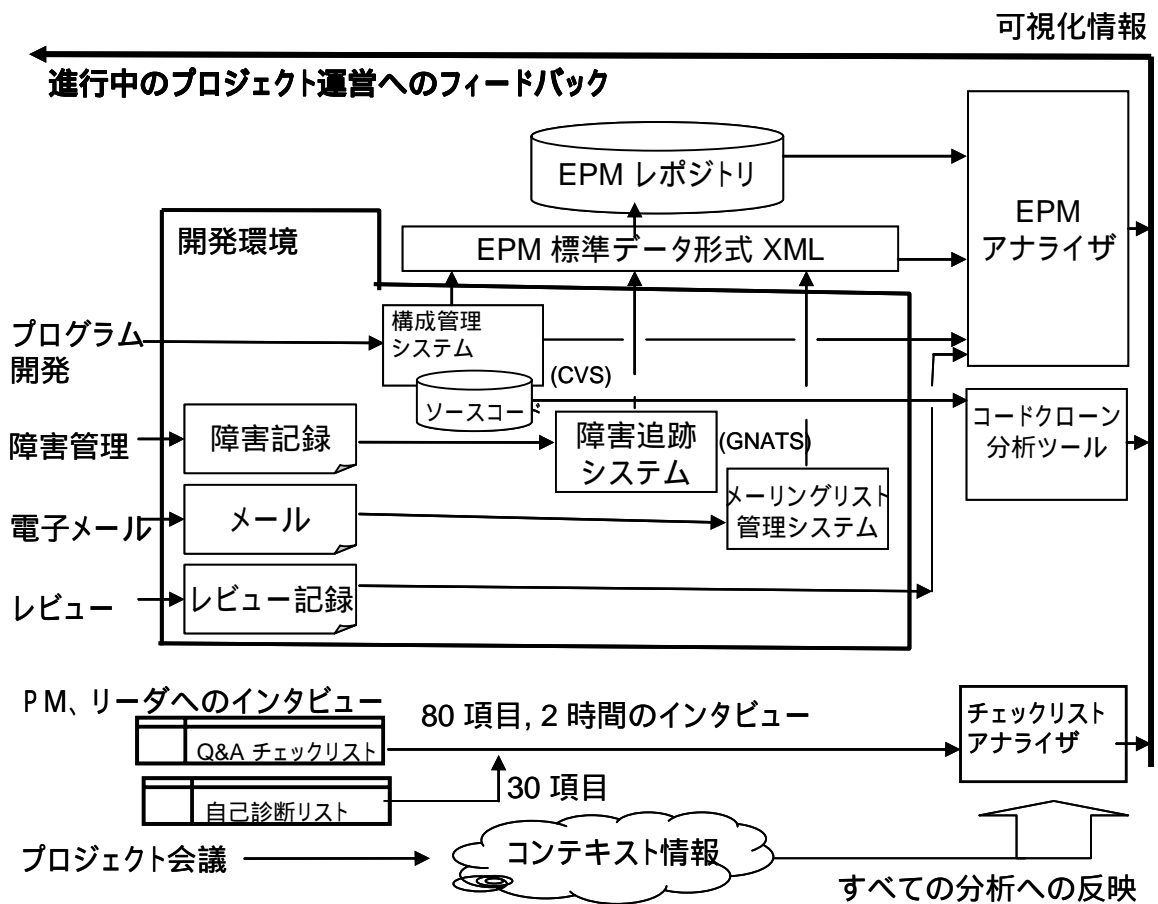


図 2 - 2 設計工程以降の総合的なプロジェクト計測とフィードバックの方式構成

### 1) EPM による計測と分析

EASE プロジェクトで開発したプロジェクト計測プラットフォーム EPM を適用して、開発プロセスとプロダクトに関する情報を取得し、そのアナライザ機能を用いてその推移に関する分析結果を得た。

EPM は図 2 - 3 にその構成を示すように、構成管理ツール、障害追跡ツール、メーリングリスト管理ツールなどの開発管理ツールを介して、ソースコードと、ソースコードに対する多くの操作、すなわちソフトウェア開発のプロセスとプロダクトの推移に関する基本的な情報を自動収集する[2-12]-[2-17]。

EPM は収集した情報を XML の標準データ形式に変換したのち、分析に供するためにリレーショナルデータベースに格納する。そのデータから EPM のアナライザ機能が、横軸を時間軸に、チェックイン・チェックアウトの契機とチェックアウトの頻度、ソースコード行数の推移、試験工程での障害件数の累積、障害の平均滞留時間、残存障害数の推移、そしてメールの件数の推移など基本的な情報を可視的に表示する。この他に信頼度成長曲線

と実際の障害件数の推移を重ねたグラフ表示機能などもある。

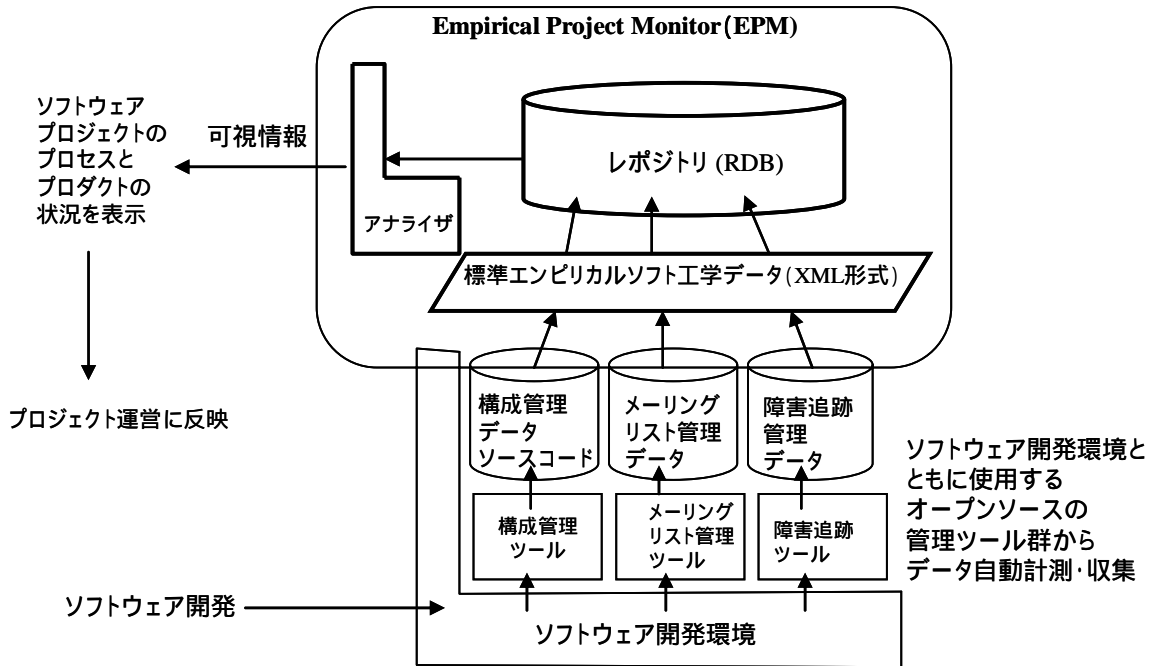


図 2 - 3 プロジェクト計測プラットフォーム EPM の構成

EPM によって開発プロジェクトは、構成管理、障害追跡、メーリングリスト管理のようなソフトウェア開発のための基本的な管理ツールを使用するだけで、特に計測について意識することなく自動的にプロジェクトデータを計測し、その分析結果を可視的に得ることができる。この点が従来の計測法にない優れた特徴であり、またこれらの情報をプロジェクト途中で得て、プロジェクトマネジメントに反映してゆくことができる点が利点である。

フェーズ 2 では、EPM の拡張機能を用いて、分析機能を更に高度化させ、より粒度の細かい分析データの提示をおこなった。

以下に EPM 分析グループによってソースコード行数の推移を計測、可視化された例を示す。[2-20] [2-26]-[2-28] ソースコード推移の計測から、各サブシステムの開発規模感、開発着手時期や立ち上げ時期の相違、さらに各グループでのソースコード管理の粒度について一目に比較、観測することができた。

図 2 - 4 に開発したソースコード規模の推移を全企業俯瞰できる形で示している。プロジェクトの開始時期は各企業同じであるが、各企業の開発活動のスタート時点がばらばらで、五月雨型にスタートしている様子が分かる。また、開発規模がこのスタートから 3 ヶ月弱の時点で大きく異なり、特に 1 社の規模が突出していることが分かる。またこの 1 社は大規模な流用母体を持っていること、ソースファイルに開発コード以外のデータが含まれ、ソースコードの管理方法が他社とは異なることが推測された。説明により、ここには大量のデータ類が含まれていることが判明し、追ってソースコードの管理を他社と同様にすることとした。

またグラフの変化の様子から各開発グループの管理体制を推し量ることができる。すなわち、細かい変化となっている場合は管理の粒度が細かく、現場に近いところできめ細かい管理がなされている可能性が高い。一方粗い階段状のグループは現場から比較的遠い管理階層で、粒度の粗い出荷管理的な管理が行われている可能性が高いことが読み取れた。すなわち、ソースコード推移の階段が細かい場合は、データ提供の元となる構成管理システムが開発現場に存在し、細かい単位でソースコード作成が管理されていることを示唆している。ソースコード推移の階段が粗い場合は、データ提供の元となる構成管理システムが日々の開発現場に無く、ある程度開発が進行したのち一定の単位で出荷管理システムとして機能していて、出荷単位となる開発が終わるまでの日々の開発期間はブラックボックスになっている可能性が示唆される。

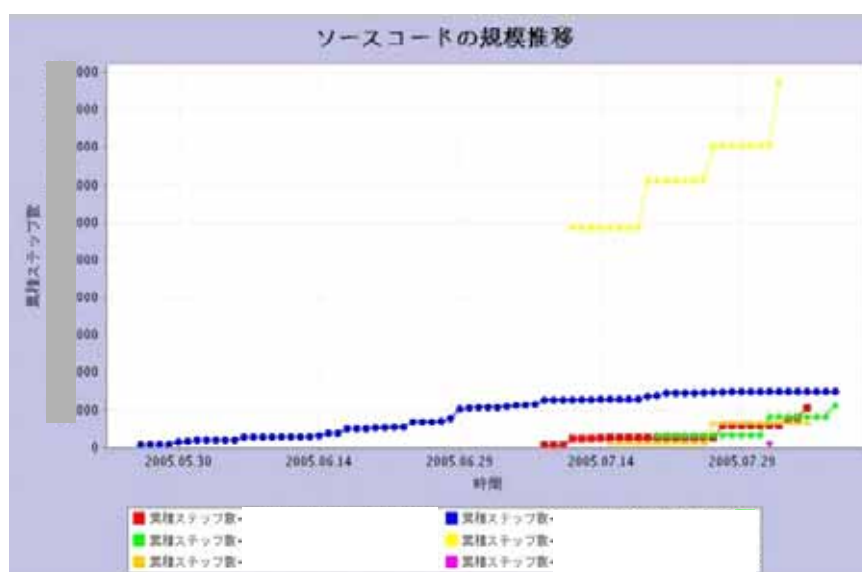


図2 - 4 ソースコード規模推移（プロジェクト立ち上げ時，全企業重畳）  
（秘匿上縦軸数値をマスクしている）（フィードバック資料から，松村知子氏提供）

図2 - 4は全企業を俯瞰して見るためのものであるが，1企業のある期間を拡大して示したのが図2 - 5である。グラフ中縦の緑線はチェックインの契機を示している。当該企業の開発した機能のリリースとその規模感を把握することができる。ここでも当該企業のソースコード管理の粒度（きめ細かさ）を把握することができる。図2 - 5は粗い階段状で、リリースとリリースの間は現場の管理に任されていることが推測できる。

EPMの拡張分析機能を用いることで、ソースファイルの更新履歴を追うことができる。ソースファイル更新履歴の計測から、開発グループごとに、ウォーターフォール型の開発、方式テストを含んだ試行錯誤型開発などの状況を明示的に観測できた。また、工程後半でのソースコードの安定度を把握できた。たとえば、仕様変更や修正のインパクト、工程途中でのコーディング標準の変更などのインパクトを明示的に把握できた。図2 - 6～図2 - 13にファイル更新履歴に関する分析グラフの例を示す。



図 2 - 5 ソースコード規模推移 ( 1 企業例 )( 縦の緑線 : チェックイン契機 )  
 ( 秘匿上縦軸数値をマスクしている )( フィードバック資料から , 松村知子氏提供 )

図 2 - 6 は 1 企業のある期間のファイル数とソースコード行数の推移を示している . この期間 , 順調に開発量を増やしている様子が見える . そして開発はまだ安定状態には入っておらず , この期間ではまだ終了していないことが分かる .

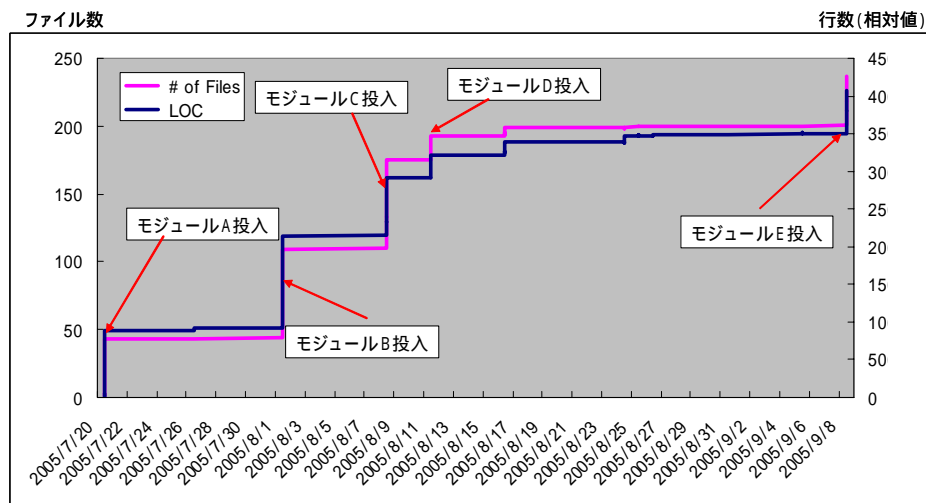


図 2 - 6 ファイル数と行数の推移 ( 1 企業例 ) [2-27]

( 秘匿上縦軸数値をマスクしている )( フィードバック資料から , 松村知子氏提供 )

図 2 - 7 はファイルやソースコードの変更量に着目した推移を示している . 比較的大きな変更と考えられる 5 行以上削除された変更の割合 ( 青 ) , 追加・削除された行数のうち 5 行以上削除された行数の割合 ( 緑 ) , 5 行以上削除されたファイルの割合 ( 桃 ) , の遷移をプロットしている . グラフはこの期間 , 大幅な修正のあとに , もう 2 回障害対応の修正があったことと対応している .

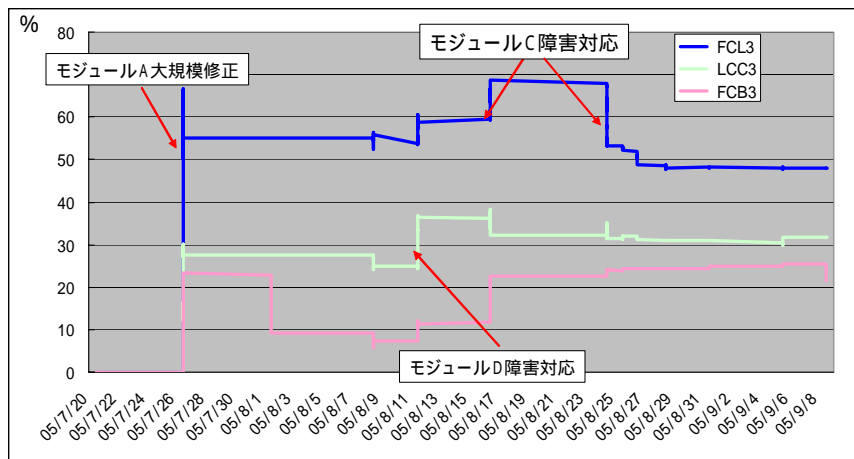


図 2 - 7 変更頻度・変更規模・変更網羅性 ( 1 企業例 ) [2-27]

LCC : 全追加 + 削除行数のうち, 5 行以上削除された行数の割合

FCL : 5 行以上削除された変更の割合

FCB : 5 行以上削除されたファイルの割合

( フィードバック資料から, 松村知子氏, 森崎修司氏, 玉田春昭氏提供 )

図 2 - 8 はファル更新の別の側面に着目している . 1 企業のある期間の , 2 回以上更新されたファイルの割合 ( 水色 ) , 5 % 以上更新されたファイルの割合 ( 紫 ) , 2 人以上で変更されたファイルの割合 ( 茶 ) の推移をプロットしている . 障害修理等インパクトのあるとき跳ね上がり , 開発にともなう全体量 ( 母体 ) の増加にともなう減少するパターンを観測できる . この例では修正と新規モジュール投入の動きを追うことができる . そしてまだ安定期ではないことがわかる .

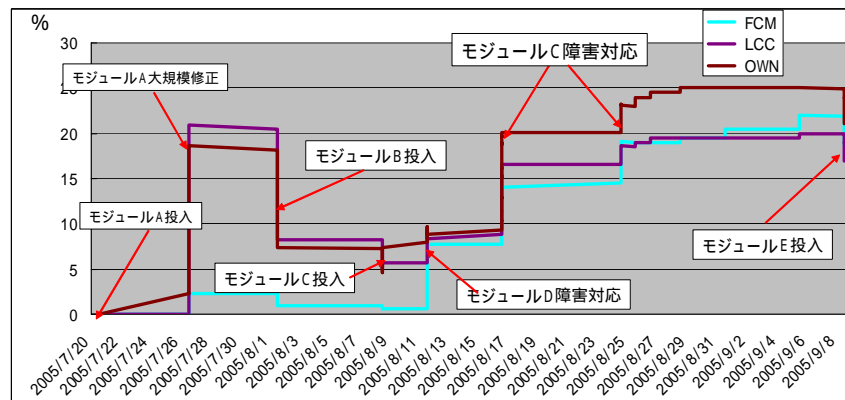


図 2 - 8 変更者数・変更率・変更回数 ( 1 企業例 ) [2-27]

FCM : 2 回以上変更されたファイルの割合

LCC : 5 % 以上変更されたファイルの割合

OWN : 二人以上で変更されたファイルの割合

( フィードバック資料から, 松村知子氏, 森崎修司氏, 玉田春昭氏提供 )

図2 - 9はこのようなファイル更新履歴に関する分析グラフを企業毎に並べて俯瞰するイメージを示している。図2 - 9はA社とB社の一部を並べた例である。A社の例は図2 - 6から図2 - 8のものである。B社は前半に試行錯誤(カット&トライ)型の開発を行っている。これはB社が今回の業務領域の開発について深い経験が無く、新しい方式を確認しながら開発したことを反映している。B社ではこうした開発方法に対して、開発期間や専門家の確保など一定のリスク管理を行って開発した。またB社のデータは開発現場から直接提供されたもので、開発管理の粒度が細かいことが推測される。B社に比しA社のデータは1段上の管理階層から提供されたもので、B社に比し、管理の粒度が粗いことが想定される。

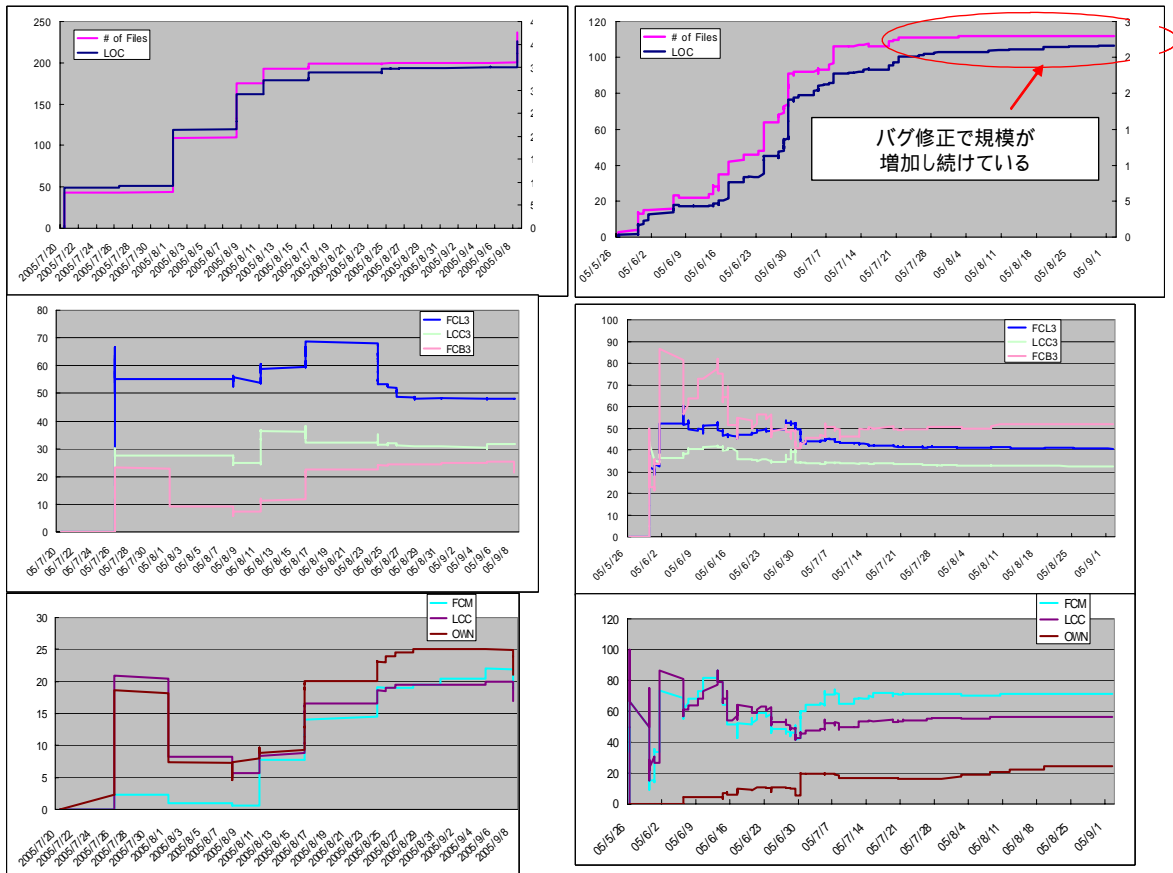
このように、各サブシステムを担当し並行に開発を進めている各社の分析グラフを俯瞰することで、プロジェクト全体の進捗をある程度推し量ることが出来る。

図2 - 9に示すようなファイルの更新状況の推移は、必ずしもそのパターンだけから特別なリスクやアクションを必要とする契機を捕捉することを目的としたものではない。これらのグラフに示されるような開発プロセスに対して、それぞれにふさわしい品質や工期が管理されていることが確かめられることが重要である。たとえばA社の場合は、これらのグラフの管理対象外の開発階層において作られたモジュールが順次積上げられ、その都度試験を実施し、検出された障害の対応を進めるというプロセスとなっている。この場合、各モジュールの品質が確保され、内部試験での障害対応が円滑に進められていることが確認できれば良い。B社の場合は、試行錯誤型開発を行うための開発期間がきちんと確保され、当該分野の専門家や経験者へのアクセスが確保されるなどのリスク対策が採られていれば良い。

また、これらの計測と可視化した情報のフィードバックの効果は、可視化された情報から直接読み取れる事象への対応だけではない。ブラックボックスの多い管理階層の中での、ある視点からの抽象化された情報を開発者や管理者の間で共有することによって、皆のモチベーションの向上、見られていることへの肯定的なプレッシャー、見えていることによる心理的な安心感などがもたらされ、これらがプロジェクト運営全体に肯定的な作用をあたえる。この効果は、今回の試みを通して定性的には観測され、また開発リーダーへのアンケート調査によっても、こうした計測とフィードバックの効果に肯定的な回答が集まったことである程度実証された。

EPMによって障害追跡システムから自動収集される情報の分析で、障害の発生状況、その種別や原因、修正状況を可視的に把握できた。また障害に関する混入工程・発見工程・発見すべき工程の分析から問題を潜在させている工程の推定ができた。障害修正に掛かっている時間の推移の分析からプロジェクトの状況を推し量ることができた。図2 - 10～図2 - 14はEPMと関連ツールによる障害関連の分析・グラフ化例である。





A社

B社 (秘匿上縦軸数値をマスクしている)

図2-9 ファイル更新履歴分析(社間比較例) [2-27]

(フィードバック資料から, 松村知子氏, 森崎修司氏, 玉田春昭氏提供)

図2-10は, 累積の障害発生件数を全社分重畳させてグラフ化したものである。ソースコード行数の推移グラフと同様, 各社のスタートと工程の立ち上がりの相違が分かる。また, 発生件数から試験の進捗の度合いの相違が分かる, またこのグラフでは, まだどの社も安定期に入っておらず開発と試験の途中であることが分かる。

図2-11, 図2-12はこのうちのそれぞれある企業について, ある時期の累積障害件数(赤), 残留障害件数(青), 障害平均滞留時間(滞留時間)(緑)(\*)をプロットしたものである。試験の進行に伴って障害が検出され, 未解決障害が残り, 平均滞留時間が伸びた, そして収束に向かっている様子が見える。( \*: 障害発生時から障害解決時までの時間の総和を障害数で除した値。)

図2-11は, テストの開始に伴って累積障害, 滞留時間とも伸びが見られ, 一旦試験作業が休止状態となった後, 障害対応によって残留障害が減少し収束に向かい, 再び次の試験工程が始まっている様子が見える。

図2-12は当初いくつかの障害対応をしていない模様で障害滞留時間が伸びている。

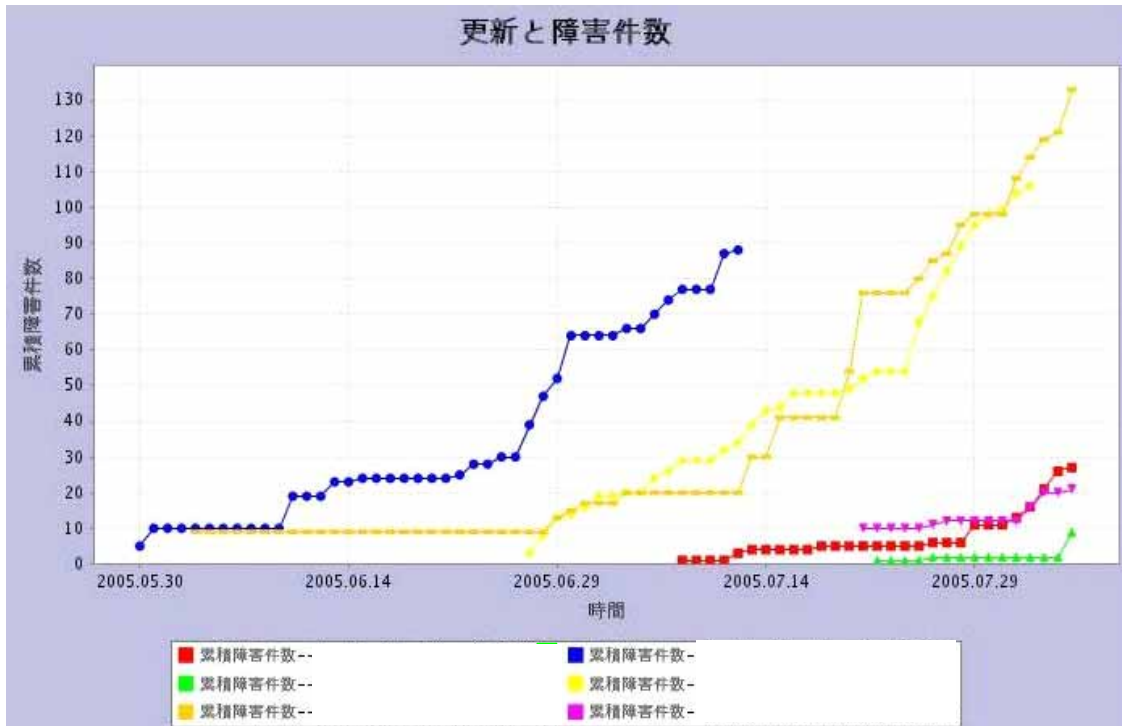


図 2 - 1 0 障害件数の推移（全企業重畳）（フィードバック資料から，松村知子氏提供）

その後障害対応を行って，滞留時間が減少に転じるが，一方試験作業が本格化し，累積障害の増加とともに残留障害も増えている．しかしながら，そうした中で障害対応が行われ滞留時間の伸びが抑制され，試験も障害発生も収束に向かっている．しかしながらこの報告期間には夏休みが含まれ，まだ残留障害があって完全には収束せず，休み後次の試験工程が開始されている様子である．

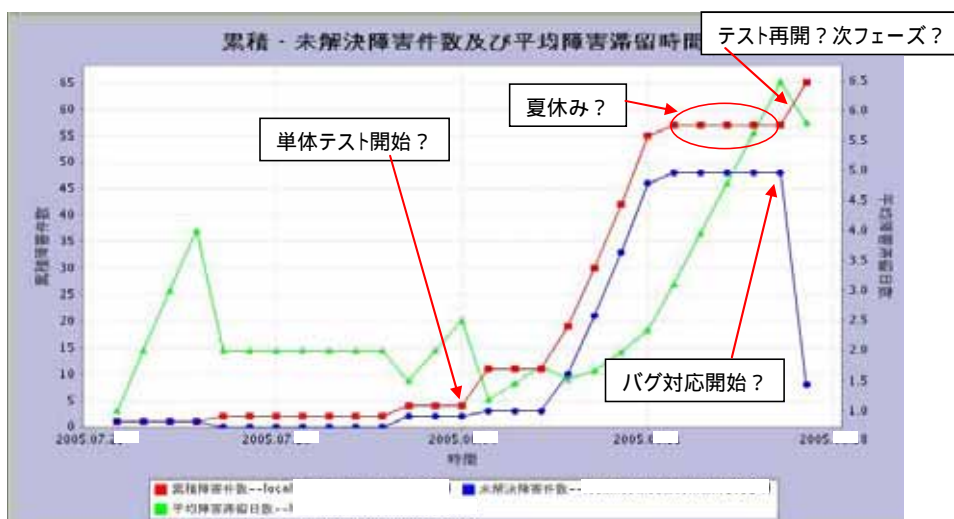


図 2 - 1 1 障害件数の推移（X社例）（フィードバック資料から，松村知子氏提供）

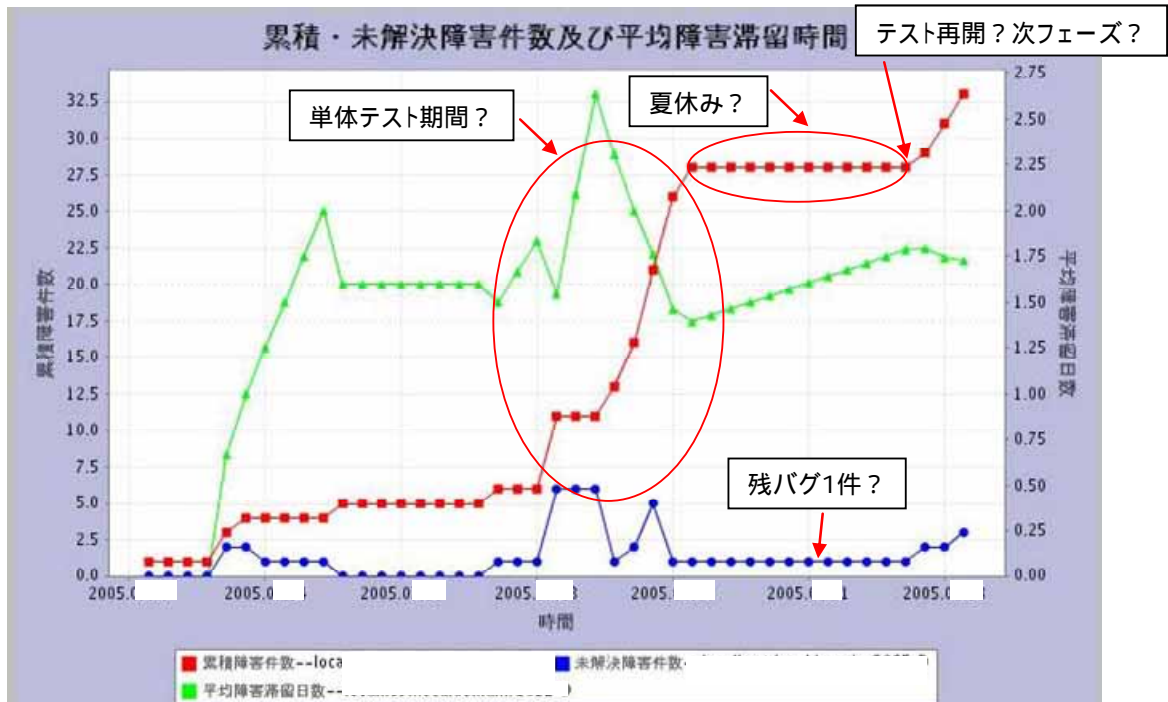


図 2 - 1 2 障害件数の推移 (Y社例) (フィードバック資料から, 松村知子氏提供)

図 2 - 1 3, 図 2 - 1 4 はこのうちのそれぞれある企業について, ある時期の累積障害件数 (赤), そのうち重要な累積障害件数 (青), 重要な残留障害件数 (緑), 重要な障害の平均滞留時間 (黄) を EPM の拡張機能によってプロットしたものである。試験の進行に伴って障害が検出され, 未解決障害が残り, 平均滞留時間が伸び, そして収束に向かっている様子が見える。PM と開発グループの間でのこのようなグラフを共有しての質疑が, それぞれの企業のテストの進行状況, 作業の休止, 障害対応などの作業状況を把握する支援になり, 各社各様の状況を把握しての深い議論に役立つ。

図 2 - 1 3 は, テストの開始に伴って累積障害, 滞留時間も伸びが見られ, 一旦試験作業が休止状態となった後, 障害対応によって残留障害が減少し収束に向かう様子がわかる。滞留時間が波打っている期間, 試験に伴う障害の発生とその解決という作業が続き, 平均滞留時間が延びるのを抑制している様子がよく分かる。後段, 残留障害が解消し試験も収束した様子が伺える。

図 2 - 1 4 は, 試験の規模 (あるいは開発規模) が図 2 - 1 3 の例より小さい様子が伺え, 障害をある程度ためてからその解消を図る作業スタイルが見て取れる。この場合も残留障害が解消し試験作業が収束している。

図 2 - 1 5 と図 2 - 1 6 は, 障害に関する工程別の分析を行ったものである。図 2 - 1 5 は障害の混入工程と本来発見すべき工程の関係, 図 2 - 1 6 は発見工程と本来発見すべき工程の関係を示している。表中の数字は, 該当する障害件数を示している。図 2 - 1 5

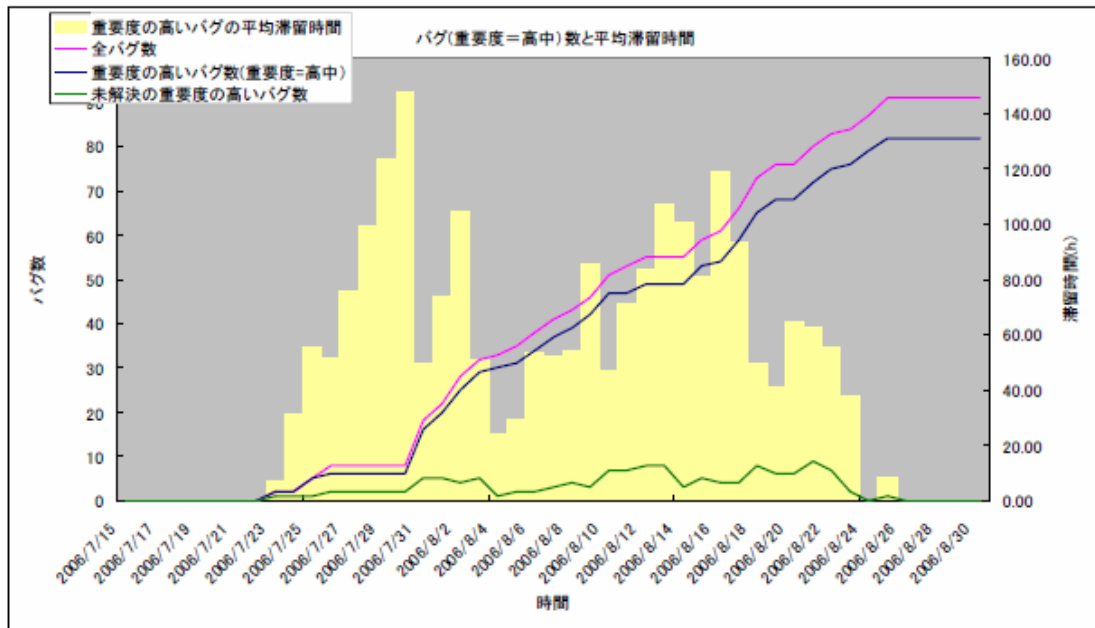


図 2 - 1 3 障害件数の推移 ( X X 社の例 ) [2-27]

( フィードバック資料から , 松村知子氏 , 森崎修司氏 , 玉田春昭氏提供 )

全障害と , 重要度の高い障害 , 重要度の高い障害の平均滞留時間 , 未解決の重要度の高い障害

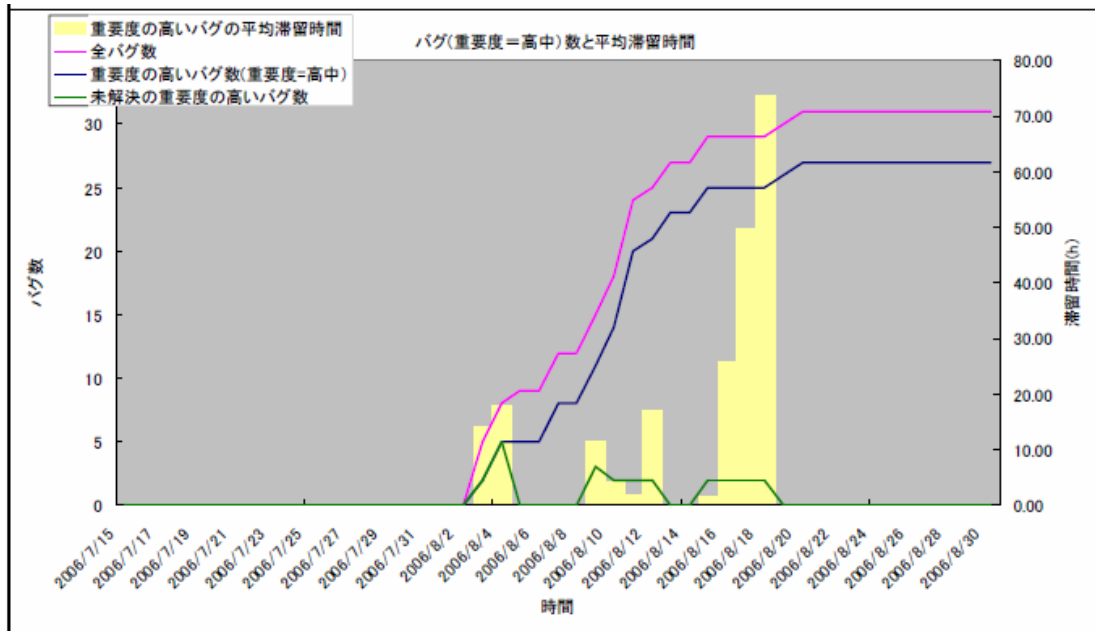


図 2 - 1 4 障害件数の推移 ( Y Y 社の例 ) [2-27]

( フィードバック資料から , 松村知子氏 , 森崎修司氏 , 玉田春昭氏提供 )

全障害と , 重要度の高い障害 , 重要度の高い障害の平均滞留時間 , 未解決の重要度の高い障害

では注釈で指摘するように、発見すべき工程として混入工程より後ろの工程を示している障害がある。これは設計時に混入した障害で、設計時あるいは設計レビュー時の検出は難しく、コーディング時にはじめて検出可能な論理構成上の障害、あるいは、コーディング時に混入したがコーディングやコーディングレビュー時の検出は難しく、試験工程にならないと検出できないと申告された障害の意味である。また、図2-16では発見すべき工程として、混入工程よりも前の工程を指定している例が2件ある。これは障害情報の投入誤りであるが、工程途中でのこうした管理上の誤りの発覚とその修正は、プロジェクト運営の状況把握に役立つだけでなく、規律の維持に肯定的な影響があると推定される。

図2-16では、注釈でいくつかの箇所について修正にかかった平均工数を示している。この分析から、発見工程が混入工程と同じ障害、1工程遅れの障害、2工程遅れの障害の修正工数の増加傾向を見れる。また障害発見工程が後ろになるほど工数を要していることもわかる。

PREFERRED-PHASE	割合	合計 PROJECT-ID	混入工程							
			00 未入力	10 要件定義	20 基本設計	30 詳細設計	コーディング (製造) / 単体テスト	50 結合テスト	60 総合テスト	90 運用
00未入力	2.67%	12	12							
10要件定義	0.00%	0								
20基本設計	0.00%	0								
30詳細設計	10.91%	49				48	1			
40コーディング(製造) / 単体テスト	84.91%	379	4			2	372	1		
50結合テスト	2.00%	9					5	3		
60総合テスト	0.00%	0								
90運用	0.00%	0								
合計		449	17	0	0	50	378	4	0	0
割合			3.79%	0.00%	0.00%	11.14%	84.19%	0.89%	0.00%	0.00%

図2-15 本来発見すべき工程 - 混入工程 (1企業例)  
(フィードバック資料から、松村知子氏提供)

図2-17は1企業の担当分に着目しモジュール毎に障害原因をグラフ化したものである。ここでは複数のコンポーネントで発生している機能強化・拡張に起因する問題と1つのコンポーネントで大量発生した操作ミスに起因する問題について問いが提示された。フィードバックの場での協議の結果、複数のモジュールで発生した機能強化・拡張に関する障害は、外部のシステムからデータを取得する方式に関する共通の要因による障害で、す

で解決しており、また操作ミスとしてカウントされた障害は、試験の操作に関するもので、プログラムそのものの障害ではないことが判明した。いずれもこの時点で特別なアクションに繋がる障害ではなかったが、障害によっては、社内のプロセスの改善や社間調整など再発防止策、抑制策を必要とする場合もある。[2-22][2-24]

P R E F E R R E D - P H A S E	割合	合計 P R O J E C T - I D	発見工程							
			00 未入力	10 要件定義	20 基本設計	30 詳細設計	コーディング (製造) / 単体テスト	50 結合テスト	60 総合テスト	90 運用
				平均工数 1.9人時					平均工数 5.5人時	
00.未入力	2.67%	12					7	5		
10.要件定義	0.00%	0								
20.基本設計	0.00%	0								
30.詳細設計	10.91%	49					22	27	平均工数 3.3人時	
40.コーディング(製造)	84.41%	379	2				335	42		
50.結合テスト	2.00%	9						9	平均工数 1.7人時	
60.総合テスト	0.00%	0								
90.運用	0.00%	0								
合計		449	2					83	平均工数 3.0人時	0
割合			0.45%	0.00%	0.00%	0.00%	81.07%	18.49%	0.00%	0.00%

図2 - 16 本来発見すべき工程 - 発見工程 (1企業例)  
(フィードバック資料から、松村知子氏提供)

EPMは上記で示してきたようにソフトウェア開発のプロセスとプロダクトの推移を視覚的に捉えることの支援になるが、もうひとつソフトウェア開発の重要な要素として、開発要員、グループ間のコミュニケーションの課題がある。

EPMにはメールの頻度を計測してグラフ化する機能がある。対象プロジェクトでは開発者個人のメール計測は開示されず、社間メールのみの計測となった。対象プロジェクトでは、社間調整は週次の進捗会議、開発事務所内の各社常駐メンバによる打ち合わせ、そして社間結合試験体制では、開発事務所での毎朝会議などですすめられることが多く、社間メールの利用は事務的な連絡、周知に使用されることが多かった。図2 - 18はEPMによるメール投稿数のグラフ化である。あえて見ると、やはりプロジェクト立ち上がり時が急峻で活発なメールによる調整が進められたことが伺え、中段以降、参加人数が増えているにもかかわらず、幾分安定したメール頻度となっていることが読み取れる。

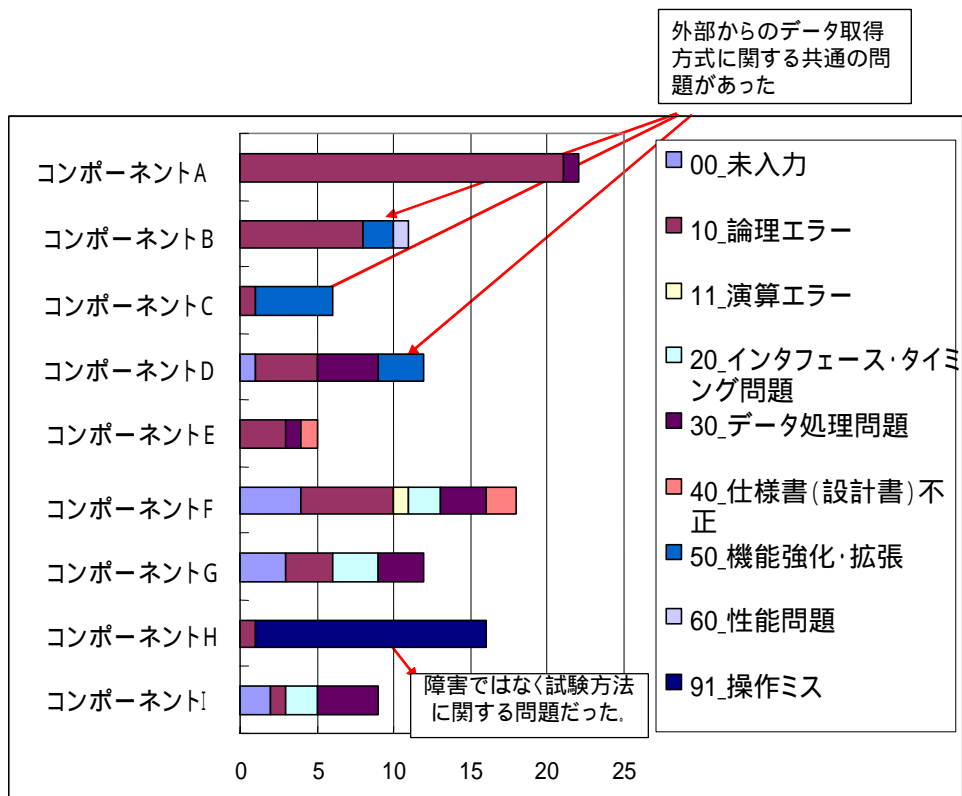


図2 - 17 コンポーネント毎問題原因分析 (1企業例)  
(フィードバック資料から, 松村知子氏提供)



図2 - 18 社間メール件数の推移計測例 (フィードバック資料から, 松村知子氏提供)

これらの計測・分析結果は、社間の秘匿条件に沿って、研究組合の各社別にフィードバックされた。その役割はフィードバック先によって異なることが明確になった。対象プロジェクトでは競争領域を設定しているため、社間秘匿情報が存在し、全体のPM（プロジェクト・マネージャ）には一種のブラインド・マネジメントが要請される。そうした中で、EPM ツール群による計測・分析データのフィードバックは、ある程度プロジェクトの透明性を確保するのに役立った。開発担当の各サブグループでは、開発現場に接する機会の少ない上位のマネージャにとっては、ある程度開発状況を定量的に把握する支援となり、また、品質管理部門など並行して多数のプロジェクトを書面上で管理している部門にとっては、新しい気づき材料を与えることがあった。また、現場に接している、もしくは接する機会の多いマネージャにとっても、新しい定量的な材料を提供することがあった。

これらのデータ提供は工程区切りの品質評価会議などでの議論を深め、関係者間でプロジェクト運営への共通の認識を醸成した。これらの活動は従来の管理手法で得る状況認識にエビデンスを提供した。また、計測の自動化により現場レポート業務の省力化に貢献した。さらこうした現場データの捕捉は、感覚的に把握している現場状況を上位管理組織や他の関係者に早期にエスカレーションするのに役立つという指摘もあった。全般に従来の自己申告方式による進捗報告に比して、自動収集により省力化され、かつ人手介入による不正確さのない報告データを提供することができ、円滑なプロジェクト運営を支援する効果があった。また各グループからはこれらの計測とフィードバック手法に接することで啓発される側面もあったと申告された。

## 2) 設計工程レビュー記録の収集と分析

設計工程のレビュー記録が、問題記述票と呼ぶ約30項目からなる電子フォームを用いて、基本設計と詳細設計に対して収集、分析された。主な項目として、レビューで指摘事項のあったドキュメントの位置（ドキュメント種別、ページ、行など）、問題点、修正内容または検討結果、エラーの現象、原因や混入工程、修正工数、それにレビュー人数や工数の情報が記録・収集された。フェーズ2では、設計ドキュメントも構成管理システムに登録し、ドキュメント加筆量などを測定して、さらに深い分析が行われた[2-23]。

設計レビュー記録の分析から、各社（各サブグループ）ごとのレビュー密度、レビュー稼働などを見ることができた。各グループのレビューへの姿勢をある程度読み取ることができる。EPM分析グループによって計測された基本設計工程と詳細設計工程のレビュー記録の例を図2-19に示す。図では、各企業別にレビュー対象としたドキュメントのファイル数、ページ数、レビューに稼働した延べ人数、延べ時間数、レビューで指摘された欠陥数を比較している。レビュー対象としたドキュメントの分量に大きなバラツキがあることなどがわかる。[2-21][2-23]



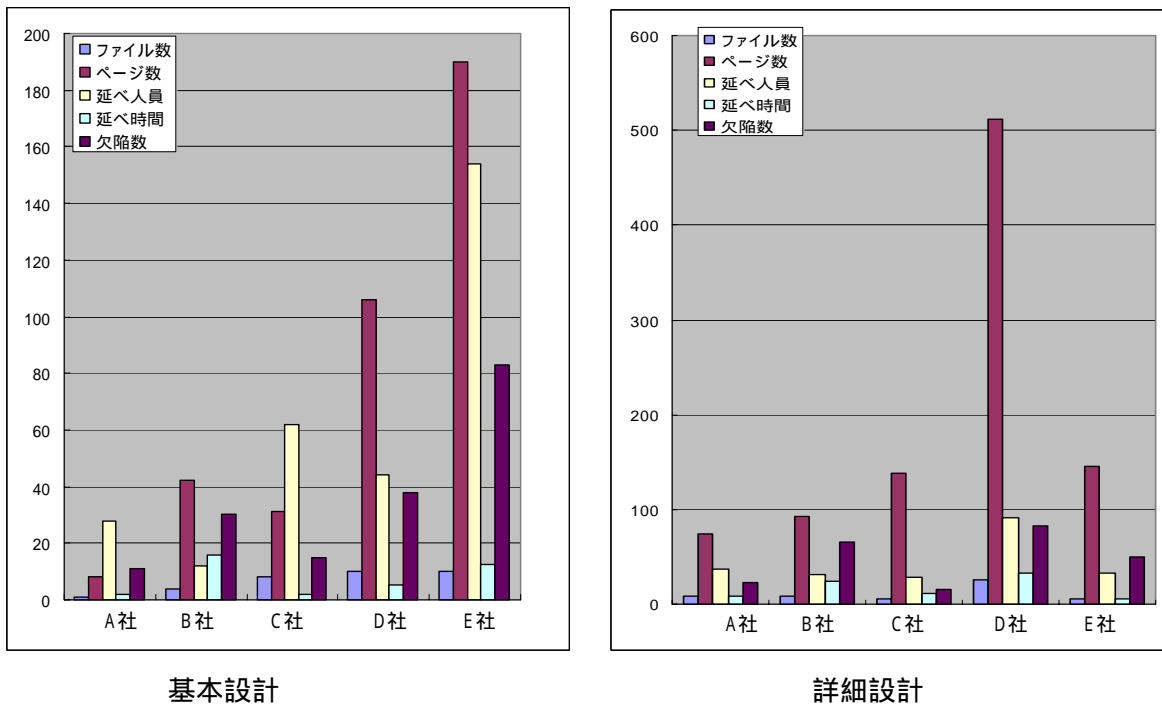


図 2 - 1 9 レビュー記録（企業別問題記述票）分析  
（フィードバック資料から，松村知子氏提供）

レビュー対象ファイル数，ページ数，延べレビュー人数，延べレビュー時間，抽出欠陥数

### 3) コードクローン分析

コードクローンはソースコード中の類似するコード片のことで，その分布状況，含有率などからプロダクトの性質を推し量ることができる．対象プロジェクトではソースプログラムの提供を受け，CCFinder と名づけられたツール[2-18]を用いて，ソースコードからコードクローンの含有状況，含有率が分析された[2-25]．クローン含有状況は，各種のクローン関係メトリクスと合わせてスキャターチャートと呼ぶダイアグラムによって，全体を鳥瞰できるビジュアルな形で表示される．プロジェクトの途中で複数回計測された．フェーズ 2 では，2 年目だけの分析ではなく，フェーズ 1 とフェーズ 2 の間のクローン分析も実施し，さらにモジュール構成の推移の観測や，プログラム複雑度と組み合わせた分析を試みた．

ソースコードが社間秘匿されている研究組合において，クローン分析はブラックボックス中のソースコードの特性を推測する材料を与えた．またクローンの観測を実施していることが，コードのモチベーションやモラルを高める役割を果たすことが申告された．

これらのことから，クローン分析のプロジェクト運営への支援効果を認識できた．

コードクローン分析グループによって示された 6 社分のある時点の散布図を図 2 - 2 0 示す[2-25]．クローン分析ツール CCFinder はこれらの散布図と，クローンの存在箇所，クローン含有率などのクローンに関するメトリクスを表示できる．散布図中の赤丸と A，B，C 等の文字は，分析者が気のついた大きなクローンの存在場所を示す．この部分に該当す

るモジュール名などをもとに分析者と開発者の間で、当該クローンの存在が既知であったかどうか、既知の場合その生成理由、今後の方針などについて協議することが出来た。このチャートを共有してクローン分析者と開発者が会話することで、他の方法では明らかにならない開発状況や課題を明らかにすることができた。また、散布図を比較することで、複雑なクローン分布を内包する内部流用の多い開発、クローンの少ないスクラッチ型の比較的新規の開発、開発者の保守に対する姿勢などを推し量ることができた。

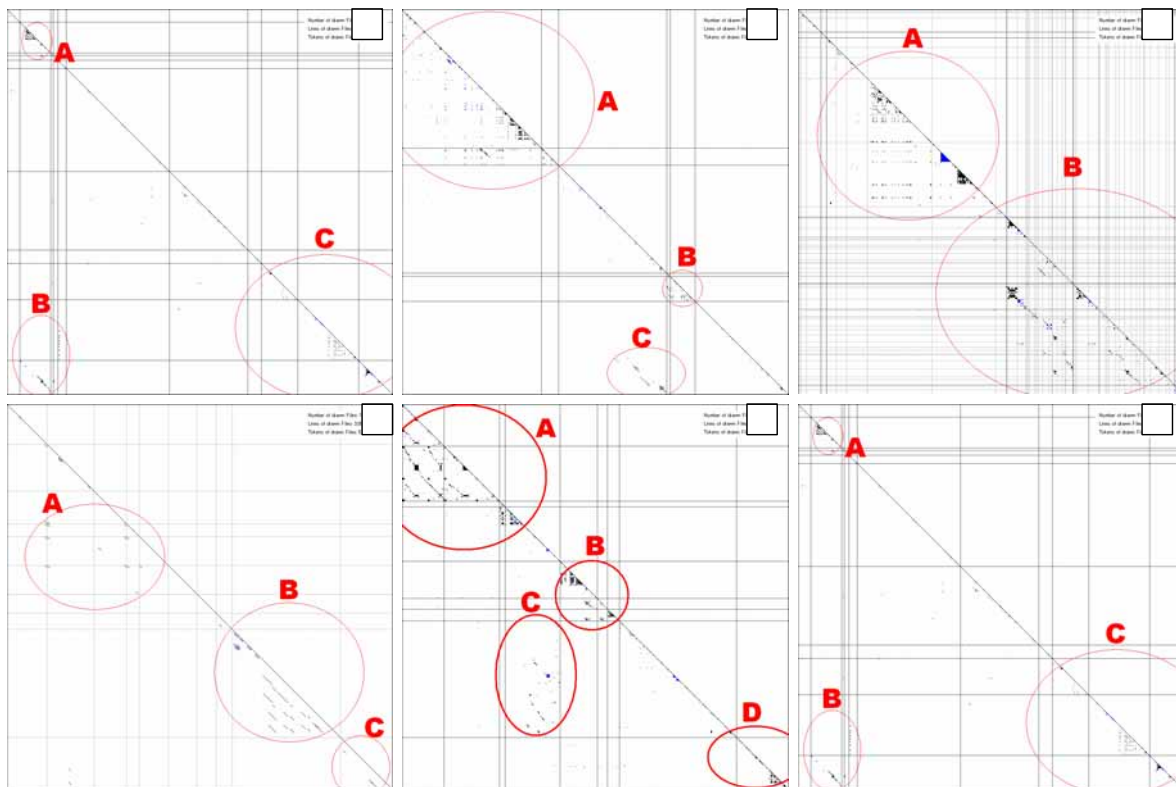


図 2 - 2 0 コードクローン分析例（散布図，6グループ分）[2-25]  
（フィードバック資料から，肥後芳樹氏提供）

たとえば，図 2 - 2 1 はあるグループのある時点の散布図である。分析者によって気につく顕著なクローンとして A から H まで 8 個のクローンの存在の指摘があり，それぞれについて開発者と議論があった。たとえば，A のクローンは，入力属性の異なる類似した 2 つの処理があり，ロジックとしては 1 本化可能であるが，片方の処理が将来大きく進化する可能性があり，それを見越して意図的にクローンにしてあるという設計思想が示された。工程中数回コードクローン分析を行い，結果を社間秘匿の条件にそって組合会社にフィードバックした。全体の PM には全社分の結果を提供した。ソースコードの秘匿された PM の視点からは，各サブグループの開発したコードの属性をコードクローン分布の視点からマクロに把握するのを支援できた。内部に大きな流用部分を持つ開発，内部にほとんどクローンを持たない，スクラッチ開発に近いコード，単純なクローンが散在し，経験の浅い

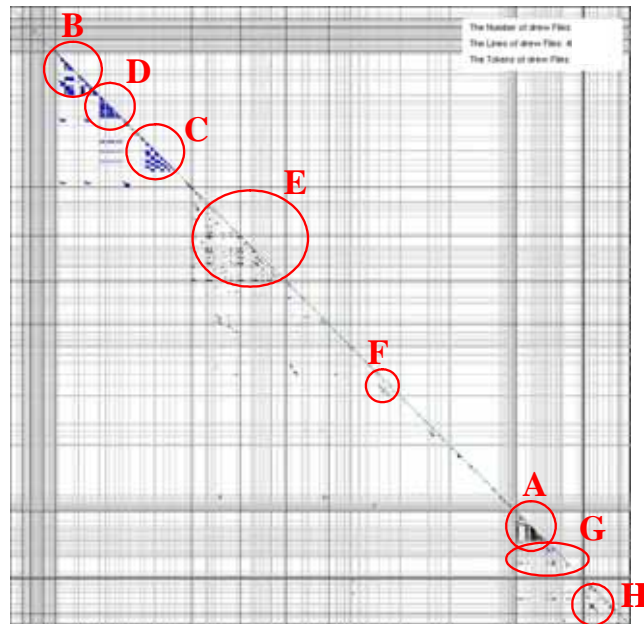


図 2 - 2 1 コードクローン分析例 [2-25] (フィードバック資料から, 吉田則裕氏提供)

コーダ(新人)を投入していると想定される開発, 全体の規模感, グループ間や社間での流用状況を把握するのを支援できた。

各開発グループの, 現場との接点の少ないマネージャには, 自社のコードのクローン分布から管理上の「気づき」を得ることがあった。たとえば新人の投入など把握していないことの指摘を受けることがあった。現場に接しているリーダーにとっては, 開発対象の機能充足だけでなく構造に関する深い議論ができ, モチベーションとモラルの向上に貢献した。

コードクローン分析を中心としたソースコード分析はフェーズ 2 でも行い, さらに 2 年間にわたる分析をすすめた。たとえば, フェーズ 1 とフェーズ 2 におけるクローン含有状況の変化, フェーズ 2 開発におけるフェーズ 1 資産の流用, 改造の状況, フェーズ 1 のクローンのフェーズ 2 への持ち込み状況, フェーズ 1 とフェーズ 2 でのプログラムの複雑さの変化とクローン含有状況の関係などの分析が行われた。

#### 4) チェックリストを用いた PM, リーダー向けインタビューによるプロジェクト情報の収集

筆者も参加するチェックリスト分析グループでは, プロジェクトのリスク, プロジェクト管理上の問題などをあぶりだすことを目的に研究組合の各企業を訪問し, 企業内の PM や開発リーダーに, SEC の部会活動(「プロジェクト見える化部会」)で作成した約 80 項目のチェックリストによる約 2 時間のインタビュー調査を実施した。このインタビューに先立ち 30 項目のチェックリストによる自己診断も行い, インタビュー結果と比較した。このチェックリストは PMBOK の知識領域を参考に SEC の部会活動の中で部会に集まる各企業の提供するノウハウをもとに作成したものであり, 上流工程用と下流工程用がある

[2-19][2-20] .この調査を通して、EPMのような自動収集ツールでは収集できない多くのコンテキスト情報を得た。

設問への回答は、自己診断の場合は3段階、専門家によるヒヤリングの場合は5段階で採点集計され、図2-22に示すような15個のパラメータからなるレーダーチャートに、自己診断結果とヒヤリング結果が比較可能なかたちで可視的に表現される。

フェーズ1で全社に下流工程編を適用し、プロジェクトリスクの抽出を図り、同時にこの実績をチェックリストの向上に反映した。また、公開デモにおいて、上流工程編を3社に適用し、おなじくチェックリストの向上に反映した。チェックリスト分析の結果はインタビュー対象の各社別にレーダーチャートと簡単なメッセージの形で示した。

本分析活動で、自動計測では収集できないプロジェクト情報を得ることで、これらの情報をプロジェクト全体の計測とフィードバック活動に役立てることが出来、チェックリストの効用を認識することができた。公式的な報告・申告だけで把握できない各社、各開発グループの開発と管理の体制、管理手法についてもその概要が把握できたのはこのインタビューによる。得られた情報は、そのほかの計測データの分析にも役立った。

レーダーチャート上の分析では、重要なことは自己診断と専門家診断に乖離がない（認識があっている）ことで、次に点数が低い部分に対し、専門家のサポートが可能かどうかという点である。インタビューの中でリスク管理に関して点数の低い社があり、これは注意喚起になった。自己評価結果とヒヤリング結果の比較分析を行うことで、プロジェクト・マネージャのプロジェクトリスクに関する認識についてある程度把握できることを確認した。また、ヒヤリングに参加した外部専門家の意見を集約してプロジェクト・マネージャに説明することにより、プロジェクトを成功に導くための対策やヒントを提示できた。

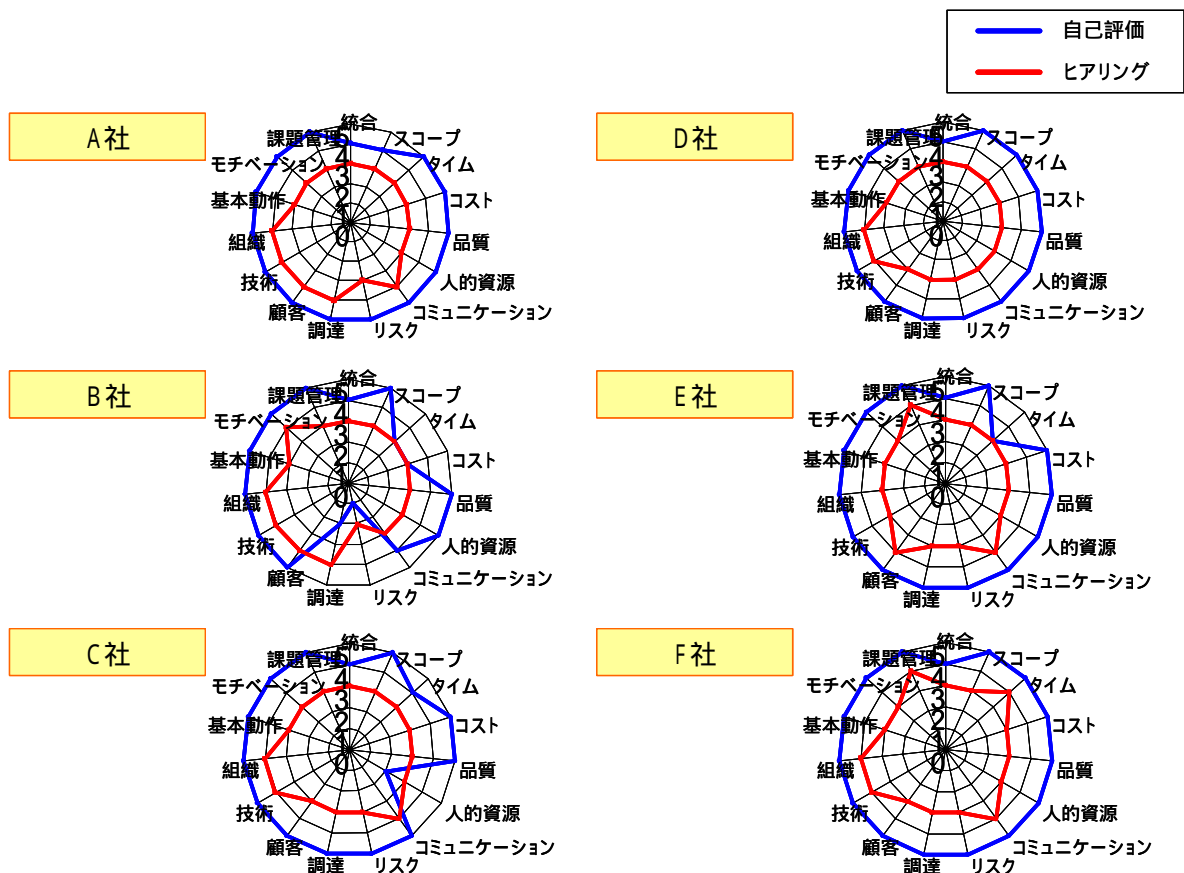


図 2 - 2 2 チェックシート分析結果例 (レーダーチャート, 6 グループ分, フェーズ 1)  
(フィードバック資料から)

### 5) プロジェクト会議への継続参加によるコンテキスト情報の収集と分析への反映

さらに各種のコンテキスト情報を得るために、筆者を含むメンバがプロジェクトを通してプロジェクト会議およびフィードバックの会合に参加し、自動収集では得られない情報を取得した。これらの活動で得られた情報をすべてのデータ分析に反映した。

これらの計測手段は、可能な道具立てを組み合わせたものではあるが、ソフトウェア開発のプロセスとプロダクトの状態推移を多角的な視点で効率的にモニタできるよう工夫している、自動収集できない情報についても収集努力をしている、製造工程以降に重点を置いているが設計工程に対してもレビュー記録という形で対象範囲を広げている、などの特徴をもち、これまでの関連研究では例を見ない総合的なものとなっている。

## 2章 参考文献

- 2-1. 樋口 登：Project Report 先進ソフトウェア開発プロジェクト；*SEC journal* no.2，pp.56-57, 2005.4
- 2-2. 松浦 清，神谷芳樹，樋口 登：Project Report 先進ソフトウェア開発プロジェクト Part II；*SEC journal* no.5, pp.44-49, 2006.2,
- 2-3. Victor Basili: Empirically Driven SE Research: State of the Art and Required Maturity, *Proceedings of 28th International Conference on Software Engineering (ICSE 2006)*, Invited Talk, pp.32, Shanghai, China, May 2006.  
(<http://www.cs.umd.edu/~basili/presentations/2006%20ICSE%20with%20Elbaum.pdf>)
- 2-4. Victor R. Basili: Is There a Future for Empirical Software Engineering?, *Proceedings of 5th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2006)*, Keynote, pp.1, Rio de Janeiro, Brazil, October 2006.  
(<http://www.cs.umd.edu/~basili/presentations/ISESE%202006%20Keynote%20Final.pdf>)
- 2-5. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, and Ken-ichi Matsumoto: A Proposal for Analysis and Prediction for Software Projects using Collaborative Filtering, In-Process Measurements and a Benchmarking, *Proceedings of International Conference on Software Process and Product Measurement (MENSURA 2006)*, pp.98-107, Cadiz, Spain, November 2006.
- 2-6. 神谷芳樹，菊地奈穂美，松村知子，大杉直樹，門田暁人，肥後芳樹，井上克郎，松本健一：進行中のプロジェクト計測とフィードバック実験に基づく計測データベース活用方式の提案，ソフトウェアエンジニアリングシンポジウム2006（ソフトウェアエンジニアリング最前線2006，近代科学社），pp.35-42, 2006.10.
- 2-7 . Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, and Ken-ichi Matsumoto: A Proposed Method for Building a Database of Project Measurements and Applying it Using Collaborative Filtering, *Proceedings of 5th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2006)*, Vol.2, Short papers, pp.15-17, Rio de Janeiro, Brazil, October 2006.
- 2-8 . Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Satoshi Iwamura, Yoshiki Higo, Katsuro Inoue, Mike Barker, Ken-ichi Matsumoto: Effect of Software Industry Structure on a Research Framework for Empirical Software Engineering, *Proceedings of 28th International Conference on Software Engineering (ICSE 2006)*, Far East Experience Track, Poster Session, pp.616-619, Shanghai, China, May 2006.
- 2-9. 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, *Proceeding of 4th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2005)*, Vol.2, Late Breaking Papers, pp.5-8, Noosa Heads, Australia, November 2005.
- 2-10. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Satoshi Iwamura, Mike Barker, Ken-ichi Matsumoto: A Research Framework for Empirical Software Engineering Collaboration and Its Application in a Software Development Project, *Proceedings of International Workshop on Future Software Engineering 2005 (IWFST 2005)*: Shanghai, China, November 2005.
- 2-11. 情報処理推進機構「柔の力，剛の技」第3部：競争力；ソフトウェア開発プロセスの「見える化」：アスキー出版，pp.160-176, 2006.5.
- 2-12. 神谷芳樹：EASE プロジェクトに見る計測・定量化の実践，*日経 IT プロフェッショナル*，2005年3月，pp.92-97, 2005.3.
- 2-13. 大平政雄，横森励士，阪井 誠，松本健一，井上克郎，鳥居宏次：Empirical Project Monitor:プロセス改善を目的とした定量的開発データの自動収集・分析システムの試作，*電子情報通信学会技術報告 SIGSS*, Vol.103, No708, SS2003-48, pp.13-18, Mar.2004.3
- 2-14. 阪井 誠，大平雅雄，横森励士，松本健一，井上克郎，鳥居宏次：EPM，導入の容易な開発データ自動収集・分析支援システム - お手軽にリアルタイムの生データ - ，*ソフトウェアシンポジウム 2004*，2004.
- 2-15. 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 (MSR2004)*, pp.42-46, Edinbargh Scotland, UK, May 2004.
- 2-16. Masao Ohira, Reishi Yokomori, Makoto Sakai, Ken-ichi Matsumoto, Katsuro Inoue, Michael Barker, Koji Tori, Empirical Project Monitor: A System for Managing Software Development Projects in Real Time, *Proceeding of 3rd ACM-IEEE International Symposium on Empirical Software Engineering (ISESE2004)* ,Vol.2, Poster Session, pp.37-38, Redondo Beach,CA, USA, ,August 2004.
- 2-17. 大平 雅雄，横森 励士，阪井 誠，岩村 聡，小野 英治，新海 平，横川 智教：ソフトウェア開発プロジェクトのリアルタイム管理を目的とした支援システム，*電子情報通信学会論文誌 D-I*, VolJ88-D-I, No.2, pp228-239, 2005.
- 2-18. Toshihiro.Kamiya., Shinji.Kusumoto, Katsuro.Inoue: CCFinder: A Multi-Linguistic Token-based Code Clone Detection System for Large Scale Source Code. *IEEE Transactions on Software Engineering (TSE)* 28, pp.654-670, 2002.
- 2-19. 情報処理推進機構，ソフトウェア・エンジニアリング・センター：ITプロジェクト

- の「見える化」, 上流工程編 : 日経 B P 社, P.208, 2007.5.
- 2-20. 情報処理推進機構, ソフトウェア・エンジニアリング・センター : ITプロジェクトの「見える化」, 下流工程編 : 日経 B P 社, P.211, 2006.6.
- 2-21. 松村知子, 森崎修司, 玉田春昭, 大杉直樹, 門田暁人, 松本健一 : GQM モデルに基づく設計工程完成度計測手法の提案, 奈良先端科学技術大学院大学, テクニカルレポート P.22, 2007-3.
- 2-22. 松村知子, 森崎修司, 玉田春昭, 大杉直樹, 門田暁人, 松本健一 : プロセス改善を目的とする ODC を用いた欠陥修正工数分析, 奈良先端科学技術大学院大学, テクニカルレポート, P.11, 2007.3.
- 2-23. 松村知子, 森崎修司, 玉田春昭, 大杉直樹, 門田暁人, 松本健一 : 設計書の再利用を考慮したレビュー効率の比較方法の提案と事例紹介, 奈良先端科学技術大学院大学, テクニカルレポート, P.16, 2007.3.
- 2-24. 松村知子, 門田暁人, 森崎修司, 松本健一 : マルチベンダ情報システム開発における障害修正工数の要因分析, *情報処理学会論文誌*, Vol.48, No.5, pp.1926-1935, 2007.5.
- 2-25. 肥後芳樹, 吉田則裕, 楠本真二, 井上克郎 : 産学連携に基づいたコードクローン可視化手法の改良と実装, *情報処理学会論文誌*, Vol.48, No.2, pp.811-822, 2007.2.
- 2-26. 玉田春昭, 松村知子, 森崎修司, 松本健一 : プロジェクト遅延リスク検出を目的とするソフトウェア開発プロセス可視化ツール ProStar, 奈良先端科学技術大学院大学, テクニカルレポート, P.11, 2007.2.
- 2-27. 松村知子, 勝又敏次, 森崎修司, 玉田春昭, 大杉直樹, 門田暁人, 楠本真二, 松本健一 : 自動データ収集・可視化ツールを用いたリアルタイムフィードバックシステムの構築と試行, 奈良先端科学技術大学院大学, テクニカルレポート, P.23, 2007.2.
- 2-28. 大杉直樹, 松村知子, 森崎修司 : ソフトウェア開発の「見える化」を支援するデータ分析力, ~ エンピリカルアプローチによる既存データの有効活用 ~ : *J I S A 会報* 80 2006 年 1 月, pp.13-29, 2006.1.



### 3. 計測とフィードバックの有用性実証に関する考察

#### 3.1 有用性実証の視点

2章で述べた計測実験では、データの計測、分析、フィードバック方式およびその実行とあわせて、その計測効果の有用性の実証が課題となる。繰り返しの出来ない、そして稀にしか得ることのできないたった一回の計測事例から、いかにしてその有用性を一般性をもって実証するかが大きな課題である。

プロジェクト計測そのものが研究対象であり、新方式適用以前の計測データは無く、計測前、計測後の比較は極めて難しい。また類似の条件のプロジェクトを2つ用意し、片方に新手法を適用、もう片方に適用せずに双方を比較する、いわゆる「制御された実験 (Controlled Experiment)」という手法も実現は難しい。仮にこのような環境を1対実現できても、その評価は1例による実証の主張と大差はない。プロジェクト計測のような新しい試みでは、多くの事例を重ねて、統計的データ処理によってその有用性を立証する手法を求めることは現状では非現実的である。そこで現実的な実証手法として、獲得できた1つの事例をできるだけ多面的な視点から観察、分析し、その中に可能な限り多くの一般性を見いだすことを考えた。

本論文の研究で計測機会を得ることのできたプロジェクトは、政府発注の中規模のマルチベンダ広域開発プロジェクトであり、その中にはソフトウェア開発プロジェクトとして一般性をもった多くの要素、属性を含んでいる。それらの属性に着目して、1つの事例での実証に可能な限り一般性をもたせることを試みた。

一つの事例に一般性を持たせる手法の一つとして、ある視点を設定して一つの事例の中に複数の事例を見る方法がある。今回の計測対象プロジェクトは6つのサブシステムから構成され、各サブシステムは5社6グループに任されて開発され、それぞれを持ち寄って結合試験に臨んでいる。そこでこの6グループを比較することで6例の事例を得ている。しかしながらこの分析はサブシステム開発、言うなれば単体開発の事例を重ねたことに相当し、プロジェクトの全体の運営に関して事例を重ねたことにはならない。

そこで、具体的に次の4つの視点からの観察と分析を試みた。

- 1) プロジェクト全体の進行を俯瞰しての観察記録としての定性的評価
- 2) リーダ、サブリーダーへの公式のアンケート調査
- 3) プロフィットを追求するステークホルダのプロジェクト前後の振る舞いの追跡
- 4) プロジェクト終了後の落ち着いた環境でのキーマンへの非公式なインタビューによる評価の獲得とそのソフトウェア開発組織、および産業構造に着目した分析

#### 3.2 全工程プロジェクト観察による定性評価

開発グループからの自己申告レベルの報告に依存する開発体制において、今回試みたプロジェクト計測はブラックボックスの多い開発現場の基礎情報を得ることができ、具体的

なマネジメントに反映された。たとえば、開発担当各社の以下のような状況を、データに基づいて認識することができた。

1) ソースコード行数の推移，障害件数の推移から，プロジェクトの進捗，開発量。

これらのデータの推移を追うことで，各社の開発の進捗状況，規模感を手にとるように俯瞰することができた。

2) コードクローン含有率，分布から，流用ソフトウェアの状況。

出来てきたソースコードのコードクローンを分析することで，ソースコードの素性，開発グループの特性を推し量ることができた。スクラッチで開発されたコード，大規模な流用部分を持つコード，経験の浅い要員によるコード，コード全体の保守性に関する懸念事項，リファクタリングの状況など，クローン含有という視点からコードの素性を知ることができた。

ソースコードの秘匿された PM と各開発企業がコードクローンの分析データを共有して議論することは非常に有用だった。クローン分析は PM のマネジメントに役立つだけでなく，開発企業にとって PM がクローン分析を参照しているということで肯定的な効果があった。たとえば，ある社は計測されたクローンに対して，将来の機能の進化を考えてあえてクローンにしてある，という設計思想を自信をもって主張することができた。

また別の例では，1 本化可能な類似の機能が 2 つあったが，片方の機能の社間での仕様調整が遅れ，これが決定したときにはもう一つの機能の製造と試験が終了していた。そこで作業効率上コピーを作って遅れていた機能を実現した，というようにプロジェクト全体のマネジメント上の状況を説明できた。このようにコードクローン分析を介したコミュニケーションは開発企業にとって設計書に示された機能の充足以上の作業品質に対する動機付けを生み，高いモラルをもたらすことが期待できた。

3) CVS レポジトリの分析から，試行錯誤型開発などソースコードへのアクションの状況。

ファイル更新状況を様々な角度から分析することで，開発の推移を推し量ることが出来た。順調なウォーターフォール型の作業による開発，カットアンドトライのある開発，ファイル更新の安定度，後工程での設計変更や障害発覚などのインパクト，など手にとるように見ることが出来た。

4) 障害分析により，障害の要因や計測した各種の要因と障害の関係を分析できた。特に，障害の混入工程など工程品質に関する分析は，開発工程全体の評価に役立った。

5) レビュー記録の分析から，レビューに取り組む各社の姿勢の粗密。

ウォーターフォール型工程を組みレビューを重視している企業と，後工程の試験を重視している企業の差が見えた。レビューに関し異なる文化を持つ企業によるパーツが社間結合試験で出会うときのコンフリクトへの備えをすることができた。

6) チェックリストにもとづいた各企業のリーダへのインタビューにより，一般には秘匿されている各企業内の開発体制に関する情報を得ることができた。これらの情報はプロジェクトのコンテキスト情報として，ツールで自動計測される情報の分析に反映させる

ことができた。

これらの情報を PM に与えることで、PM は各企業の進捗状況などプロジェクト運営の基礎的な認識を得ることができた。

- 1) たとえば、ある社は順調な進捗に見え、クローン含有率が低く、スクラッチ開発の割合が多く、CVS のレポジトリ分析から、ロジック部分の試行錯誤開発が行われた、と推定された。
- 2) また別の社はクローン分析とコード量の推移から大きな流用部分をもち、ここの改造の良否が開発を左右すること、全体規模の膨張から将来のメンテナンスに課題が残ることが推定された。
- 3) また別の社は、レビューよりも走行試験を重視していると推定された。

### 3.3 リーダ、サブリーダーへの公式アンケート調査

計測とフィードバックの効果を直接開発グループに問うアンケートが実施された[3-1]。アンケートは、1年目、フェーズ1の開発の終了後に、全体のPMと各サブグループのリーダー、サブリーダー5名程度、合計26名に行われた。回答は社名は特定できるが個人は特定できない限定的な匿名方式で行われた。

回答の中でフィードバックの有用性に関する部分を表3-1に示す。肯定的に考えれば、もとより本プロジェクトは従来方式のマネジメントだけで推進可能なように計画されていて、今回の実験の計測とフィードバックは開発担当にとっては冗長な作業という側面もあったにもかかわらず、そのような条件下でも各リーダーからプロジェクト運営上有用だったとの回答を得ることができたことを評価できる。

しかしながら批判的にとらえれば、この計測実験は発注者の意向で予算措置をとまなげて行われたもので、このような公式的な、しかも計画途中の調査ではある程度発注者の意向を汲んだバイアスのかかった回答となっている可能性がある。

表3-1 リーダ、サブリーダーへのアンケート結果(部分, n=26)

	規模 推移	障害 進捗	障害工程 分析	障害原因 分析	コードクローン 分析	設計レビュー 分析
有用/少しは有用	20	21	16	17	18	11
あまり/全く有用でない	2	1	2	1	0	4
どちらとも言えない	4	4	8	8	8	11

### 3.4 ステークホルダの振る舞いの追跡

実際に開発プロジェクトを運営する開発グループは学術的な意義よりも開発プロジェクトの円滑な成功を第一義にする。そこで、実際にソフトウェアを開発する利益追求集団であるプロジェクトの各ステークホルダの計測とフィードバックに対する振る舞いを観測す

ることで、その有用性評価につながるエビデンスを得ることが期待できる。

計画当初開発グループは、プロジェクトへの擾乱が無ければ発注者の意向に沿って学術的な研究に協力するという姿勢で、計測とフィードバックへの期待は持っていなかった。しかしながらプロジェクトの進行にともなって、具体的に下記のような推移となった。

- 1) 1年目、フェーズ1での計測とフィードバックは、それ無しでもプロジェクト運営が進行する体制の中で、プロジェクト運営に擾乱を与えないモニタ型の方式で行われた。モニタしたデータの分析結果は開発グループに参考情報として提示された。
- 2) 2年目、フェーズ2の開発では、フェーズ1の成果を肯定的に積極的に評価し、計測とフィードバックをプロジェクト運営に組み込んだ。(各開発グループにとって、計測とフィードバックは当初の想定以上に効果的との印象を与えた。)具体的には計測とフィードバックの周期を正確に週次とし、毎週の進捗会議に計測・分析データが反映できるようにした。工程区切りの品質検討会議ではそれまでの計測・分析データを共有しながら工程評価について協議する方式となった。
- 3) 2年目の後段、フェーズ2につづいて行われた公開デモ用開発では、短期開発の事情も反映して自己申告レベルの進捗報告を廃止し計測とフィードバックを一步進めこのデータだけでプロジェクト運営がすすめられた。

プロジェクトのこの推移は、開発グループから計測とフィードバックへの一定の信頼が得られたことを意味する。

次に、プロジェクト終了後の各開発担当社の振る舞いに着目すると、下記の結果となった。(社名は匿名性確保のため以降の説明とは対応させていない)

- A社：開発をすすめたグループ企業がIPA/SECの企画したEPM等の計測ツール群検証計画に参加。オフショア開発を含めた協力会社体制の運営への適用を検討。本体企業においても適用を決定。
- B社：既存の社内ツール群を活用し、考え方のみ社内に反映してゆく。
- C社：チェックリストを活用、社内管理システムに組み込む。別の顧客の依頼でEPMツール群検証計画への参加を表明。
- D社：コードクローン分析に関する共同研究を実施。EPMツール群検証計画への参加を検討。
- E社：EPMツール群検証計画への参加を検討。
- F社：EPMツール群検証計画への参加を検討。

このように、プロフィットを追求する企業が学術目的でなく実際のフィールドへの適用を検討し、そのために自社の資源を割り当ててゆくということは、試みた計測とフィードバックへの評価のエビデンスと考えることができる。

- 3.5 プロジェクト・キーマンへの非公式事後インタビューと産業構造に着目した分析  
開発プロジェクトが成功裏に終了した時点で対象プロジェクトに参加した各社のキーマ

ン各1名から、当該プロジェクトでのソフトウェアエンジニアリングに関する施策への評価を聞く機会をつくった。このインタビューの目的は学術調査ではなく、当該計画のソフトウェアエンジニアリング施策について、読み物として出版する活動の一環として行われ、プロジェクト終了後という時期もあり、公式的立場にこだわらない、いわば本音を聞ける機会であった。

インタビューは各社1名、合計8名から行ったが、それぞれプロジェクトの中で多様な位置、立場を持っている。その発言はそれぞれの位置づけの中で評価する必要があるが、同時に、産業構造を構成する多様な立場からのソフトウェアエンジニアリング施策への評価が得られ、本研究でのプロジェクト計測の意義、有用性を実証する高い現実性をもったエビデンスを得ることが出来た。

はじめに、対象としたプロジェクトの開発体制について概観する[3-2]。次にインタビュー対象のこの体制の中の位置づけについて明確にしながらその発言を評価することを試みる。

図3-1にプロジェクト管理と計測・フィードバックの論理的な構造を示す。ソフトウェア開発は発注者から研究組合へ行われ、研究組合の中ではプロジェクト・マネージャ(PM)を担当する企業(A)と複数の研究組合企業(B~F)で分担して開発する。分担開発されたサブシステムは統合され社間結合試験、総合試験を経て1つのシステムとして目的の実験に供された。組合企業はそれぞれグループ企業や協力会社と開発グループを構成し自社担当分を開発した。プロジェクト管理のフローはPMから開発担当の組合企業への流れとなる。プロジェクト計測データは各開発企業から直接、発注者の位置づけの計測グループ

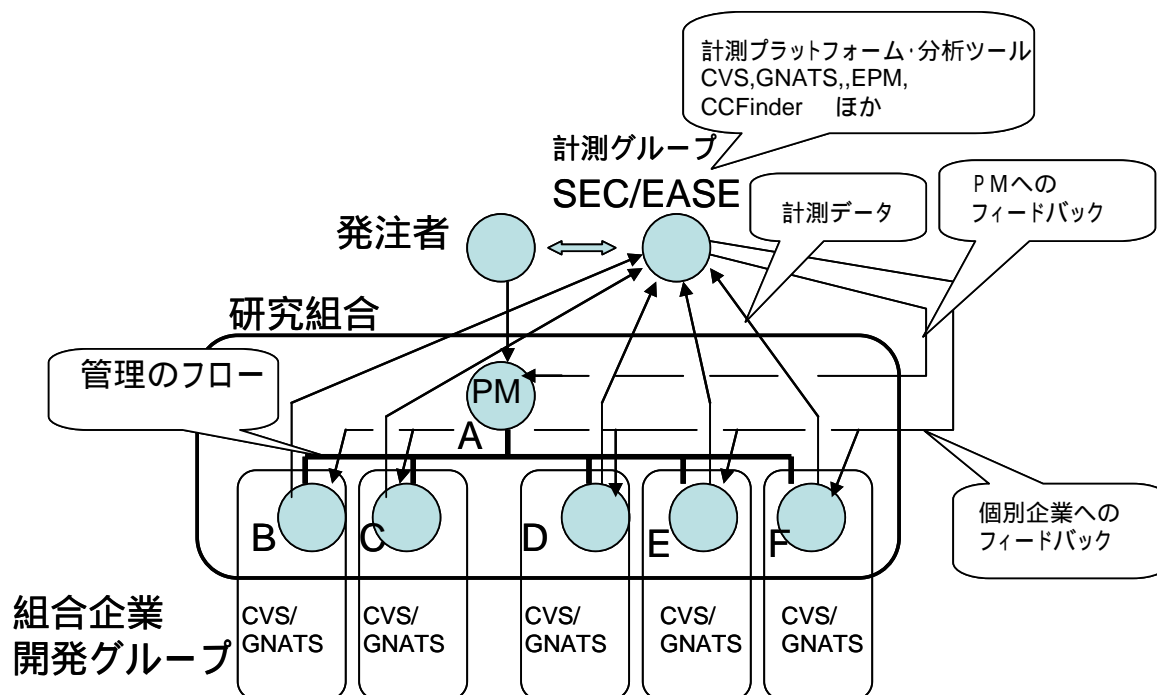


図3-1 プロジェクト管理と計測・フィードバックの論理構造

ループ(すなわち IPA/SEC および連携した EASE プロジェクトの分析グループ)へ集められる。分析結果のフィードバックは PM へは全体が俯瞰できる形で、各開発担当企業へは他社分を秘匿して個別に行われた。

図 3 - 2 に論理的な開発構造を示す。発注契約は発注者から研究組合との間で結ばれた。

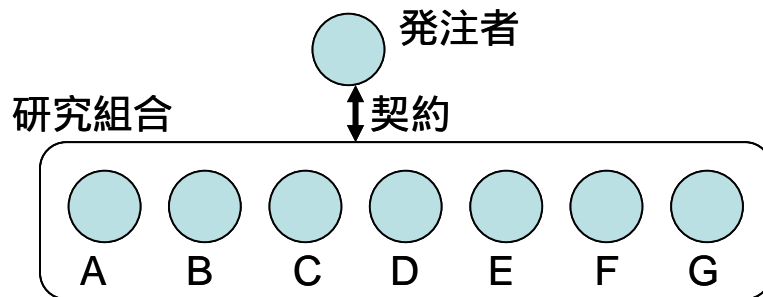


図 3 - 2 論理的な開発構造

しかしながら物理的な契約とマネーフローは実質的に図 3 - 3 に示すように発注者と組合企業個別に行われるのと等価である。管理の流れのみ PM 企業を経由した形となっている。この点は、研究組合方式と元請け方式の異なる点である。PM が、契約とマネーフローを管理しない分だけ、PM にとってブラックボックスの多い形態となっている。このほかに、本プロジェクトではユーザの視点でもうひとつの企業 ( G ) が加わり、要求定義とターゲットシステムの実験を指揮した。

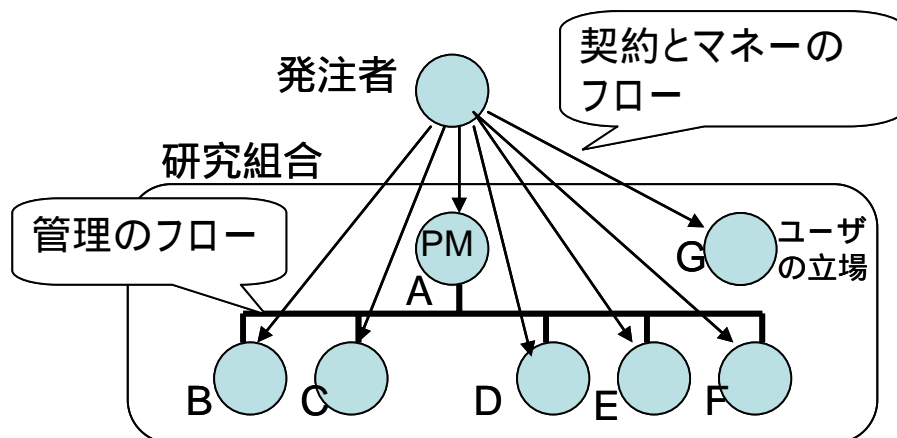


図 3 - 3 物理的な契約，管理，マネーフロー構造

( , , は表 3 - 4 , 5 のインタビュー先に対応)

図 3 - 4 に物理的な開発体制の構造を示す。各開発担当組合企業は産業構造を反映した奥の深い協力会社体制で開発をすすめるが、発注者からはその様子は公式には見えない。

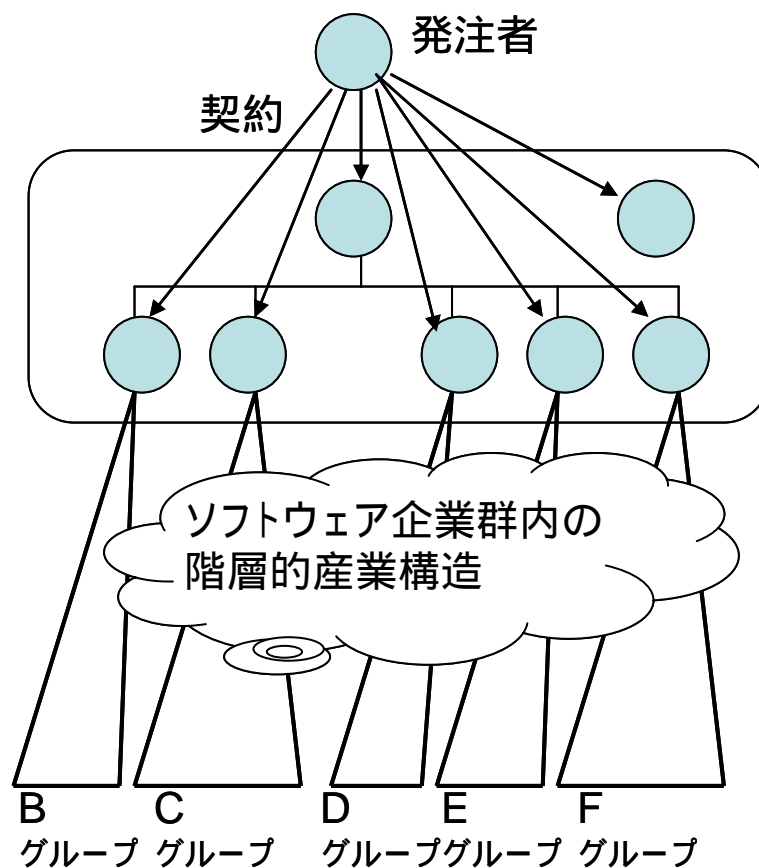


図3 - 4 物理的な開発体制の構造

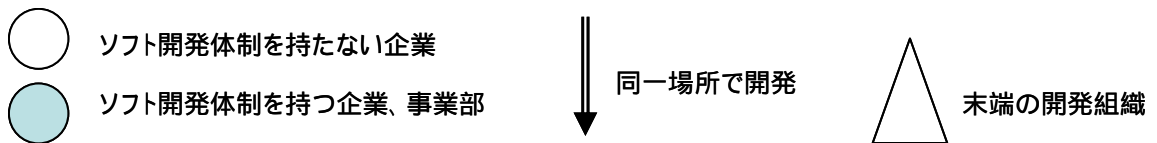
表3 - 2 , 表3 - 3 に開発体制の階層構造を示す . 各開発グループで多様な構成がとられている .

このような構成は公式には明らかにされていないものであるが , 今回のプロジェクト計測ではその一環として行ったチェックリストによるヒヤリング活動の中で明らかにすることができた . 図中白丸で示した組合企業3社はソフトウェア開発要員 , 開発体制をもたず , グループのソフトウェア企業に開発を委ねている . グループのソフトウェア企業もソフトウェア開発環境を持つ場合と持たない場合がある . 構成管理システムなど開発管理環境のみを持つ場合もある . 開発グループ内で企業が階層を構成しても , 物理的に一箇所で開発する場合もあれば , それぞれが異なる場所で開発する場合もある . また企業階層の中ではネットワークアクセスが政策上禁止されている場合もある .

ソフトウェア開発環境およびプロジェクト計測の中心となる構成管理システム ( CVS ) , 障害追跡システム ( GNATS ) の位置は各組合企業によって異なる . 計測データはここから直接収集されるが , 分析データのフィードバック先は階層の頂上または1段下のグループ企業どまりである . ここにはプロジェクト計測に関する情報の片方向性が認められる .

表 3 - 2 開発体制の階層構造 ( 1 / 2 ) ( B 社 , C 社 )

開発グループ	B	C
階層的な産業構造		
↑ CVS/GNATS 提供元	B1	C1
↓ フィードバック先	B1	C,C1
階層構造の概要	<p>グループ企業 B1 がグループの PM で CVS や GNATS を持つ。いくつかの協力会社 B<sub>n</sub> が開発。協力会社は小さく、CVS や GNATS を装備しない。開発ソフトは B1 へ集められる。協力会社 B<sub>n</sub> はアクセス政策で B1 内のサーバにアクセスできない。</p>	<p>組合企業 C がグループの PM でグループ企業 C1, 協力会社 C<sub>n</sub> と一体になって開発。C1 に開発環境、CVS, GNATS が装備されている。</p>



、 …… 表 3 - 4 , 5 のインタビュー先に対応



表3 - 3 開発体制の階層構造 ( 2 / 2 )( C社 , D社 , E社 )

開発グループ	D	E	F
階層的な産業構造			
↑ CVS/GNATS 提供元	D1	Ea, Eb1-->Eb	F2-->F1, Fn-->F1
↓ フィードバック先	D, D1	Ea, Eb	F, ( F1 )
階層構造の概要	<p>グループのソフトウェア企業D1がグループのPM. 協力会社のDnとともに開発。開発環境、CVS, GNATSはD1に装備。D1はアクセス政策でD内のサーバにアクセスできない。</p>	<p>E社の2つの事業部EaとEbが別々にサブシステムを担当。EaはEaグループのPM. 開発環境、CVS, GNATSを装備、一体となって同一場所で開発。EbはEbグループのPM. グループ企業のEb1に開発環境、CVS, GNATSを装備。このほかに、Ea, Eb共通組織があり、技術課題を担当。</p>	<p>FがグループのPM. グループのソフトウェア企業F1が実質的に開発を管理、協力会社のFnが開発。開発環境、CVS, GNATSはFnに装備。F1に開発環境は無い。当初フィードバックがF1に届かなかったが、後にF1へもフィードバックする体制となった。</p>

プロジェクト全体で直接のソフトウェア開発要員は 60 名程度であった。これは開発体制の大規模な構えに比して少なく、対象プロジェクトは大規模開発体制による中規模開発とすることができる。

キーマン・インタビューはこのような階層構造の中の、各組合企業の活動を代表する人物に対して行われた。その主要な発言を、階層構造中の位置と合わせて、表 3 - 4、表 3 - 5 に示す。

概括的に評価すれば、まず発注者はプロジェクトの成功に満足した。

次にシステムのユーザの立場で参画したターゲット領域の技術的リーダー、実験リーダーは、ソフトウェアエンジニアリングの施策を肯定しながらも、ソフトウェア開発には全く煩わされることがなかったことを言明している。

開発担当の組合企業のリーダーにはソフトウェア開発にかかわらない企業があり、それぞれソフトウェア開発に煩わされることなく、ターゲットの実験に専念できたこと、また開発担当からソフトウェアエンジニアリング施策への高い評価を聞いていることが言明された。そしていずれも自社の施策への組み入れを検討している。

ソフトウェア開発にかかわったリーダーからはそれぞれ深いコメントが得られた。それまでソフトウェアエンジニアリングの手法に馴染みの薄かった企業は触発されるところが大きく、自社の施策に組み込んでゆくことを検討している。開発現場に接していたリーダーは、EPM のようなプロセス管理機能よりも、個別の分析機能に高い評価を与えた。特に相関分析ルールを用いた高度な分析法に高い評価が示された[3-3][3-4]。現場から遠い位置づけだったリーダーは、EPM などの遠隔管理機能、協力会社などのモニタリング機能を高く評価した。全体の PM からは、ブラックボックスの多い開発体制の中での、計測フィードバックの意義を評価し、特に EPM のようなプロセスのモニタリング機能の意義を認めた。また、プロジェクトの中で計測とフィードバックの方法、運用の改善に自ら関わって来たことを述べた。そして、個別の分析機能については、現場適用へのブレークダウン不足、全体が調和していない、全体像が見えない、などまだ途上にあるとの評価であった。

このように、開発体制上の様々な位置づけのリーダーへのインタビューを俯瞰すると、本論文の研究で試みたプロジェクト計測とフィードバックがそれぞれの立場に有効に機能していることがわかる。また同時にいくつかの課題を明らかにすることができた。

たとえば、全体の開発 PM の提言にもあるように、フィードバック情報は均一ではなく、フィードバック先の立場に合わせて選択、調整されるべきだ、という指摘である。先に述べたように、この計測とフィードバックには情報の片方向性の問題もあり、こうした産業構造の研究、産業構造にあわせたフィードバック情報とフィードバックの方式を案出してゆくことが今後の課題である。

表3 - 4 事後のキーマン・インタビュー結果要約(1/2)

インタビュー先	立場	開発体制中の位置づけ	意見	マクロな分析
1	発注者	発注者	プロジェクトの完遂を評価。成果は予想以上。満足感、達成感がある。 評価点： EPMの効果が確認できたこと。 プロジェクトの時系列の「見える化」。 プロジェクト外部の人間が見ても、プロジェクト内部で起こっていることが非常に見やすくなった。 マルチベンダー開発環境下での潜在的リスクの回避。 様々なソフトウェアエンジニアリング手法の実践（コードクローン分析、人材スキル分析、類似プロジェクトの蓄積データからの工数見積もりや障害の予測など）	全体のオーガナイゼーションとして満足
2	全体技術リーダー 要求定義・仕様管理のリーダー 実験リーダー 対象領域の専門家	ユーザの視点で参画 図3-3 G	ソフトウェアエンジニアリングにはタッチしなかった。 V字モデルに沿って2サイクルの開発を運用した。 今までノー管理だったものを「見える化」することはよいことと考えている。	ソフトウェア開発にはノータッチで、対象システムの仕様管理・実証実験に没頭できた
3	B社リーダー 対象領域の専門家	表3-2 B社B	ソフトウェアエンジニアリングにはタッチしなかった。 担当の100%グループ企業から、非常に良い刺激になった、という評価を聞いた。 自社の開発方法、ツールを見直すきっかけになった。 特に評価できる点：遠隔地の開発をサポートする環境。 海外まで含んだソフトウェア構築をサポートできる環境として非常な可能性があるという意見が上がっている。	同上 開発現場から良い評価を聞いた
4	E社リーダー 対象領域の専門家	表3-3 E社共通	直接ソフトウェア開発に携わらなかったが、開発グループから役に立った、参考になったと聞いた。 ツール導入も円滑だったと聞いた。 自分のセクションでも手法を取り入れて、従来のやり方を変えてゆきたい。個人の裁量に依存するところが大きく、折角のアイデアもソフトウェアの完成度が低くて展開しづらいことがあるが、ソフトウェアエンジニアリング手法で改善してゆきたい。	同上 開発現場から良い評価を聞いた
5	D社リーダー 対象領域の専門家 開発リーダー	表3-3 D社D	ソフトウェアエンジニアリングの経験に乏しく、苦労したが、良い経験になった。社としてソフトウェアエンジニアリングで得たものは大きかった。 是非、手法やツールを自社に展開してゆきたい。 これまでソフトウェアエンジニアリングの文化が無かったので、自社独自の「見える化ツール」を作るぐらいの気持ちで、時間とお金をかけて粘り強く展開してゆきたい。	ソフトウェアエンジニアリングの勉強になった。自社展開を計画

表 3 - 5 事後のキーマン・インタビュー結果要約 ( 2 / 2 )

インタ ビュー 先	立場	開発体制中 の位置づけ	意 見	マクロな分析
6	C社サブリーダー	表 3 - 2 C社C 開発現場に 接している リーダー	<p>今回の計画でソフトウェアエンジニアリングの実践に大きな関心を抱いていた。</p> <p>一番関心を持ったのはコードクローン分析。コードの客観的評価が得られたことは大きな意義があった。</p> <p>相関ルール分析によるモジュール間の類似性分析と、スキル分析には大きな可能性を感じた。</p> <p>相関ルール分析はプロジェクトメンバー内の障害傾向がわかり、プロジェクトメンバーの配置に反映できる可能性が感じられた。</p> <p>コードクローン分析と、相関分析による指摘は、プロジェクトの中で反映した。指摘事項はやはり新人の仕事に集まった。</p> <p>スキル分析は体制作りの指針になった。</p> <p>1年目の分析結果を2年目に反映し、開発に措置し、ステップ数の削減などに結びついた。品質向上に貢献したと考えている。</p> <p>ロジカルカップリング分析、協調フィルタリングはまだ研究段階で現場作業ですぐ役立つまでにはブレークダウンされていないと感じた。</p>	<p>全体に高い評価。</p> <p>個別分析手法に高い評価。プロジェクト内で運用に反映したことを明言。</p>
7	F社リーダー	表 3 - 3 F社F 開発現場から 遠いリーダー	<p>EPMでソフトウェアエンジニアリングの適用範囲が広がったことを実感した。</p> <p>ソースコードの類似性という客観的観点からの指摘が、バグに関して担当者に対するネガティブな指摘ではなく、モチベーション維持に非常に役立った。また、網羅性、処置の即応性の観点からも非常に効果があった。</p> <p>想定していたよりも、作る側の立場で分析されていた。フィードバック情報と現場の実際との乖離がなく、素晴らしいと感じた。</p> <p>分析者との開発中のダイナミックなやりとりが有益だった。分析者の指摘に、さらに次の行動への指針が含まれるともっと良かったが、そこまでは踏み込まなかった。</p> <p>自社内、協力会社へも展開を考えている。大規模プロジェクトだけでなく、ソフトウェアエンジニアリング手法の適用の難しい中小規模プロジェクトへも適用可能との感触を得た。</p> <p>SECで中小規模のプロジェクトにおけるソフトウェアエンジニアリング適用データを沢山収集し、その効果を発表してほしい。業改全体の改善につながると考えられる。</p>	<p>管理機能を評価。</p> <p>協力会社体制への展開を検討。</p>
8	全体開発PM	図 3 - 3 A 全体のソフトウェア 開発リーダー	<p>研究組合方式で、ブラックボックス部分が多く、またマネージャーにタッチしない環境でマネジメントは難しかった。</p> <p>EPMには一定の効果があった。仕組みとして、リアルタイムの分析データとフィードバックには意義があったと評価できる。</p> <p>EPM以外のツールは単発の印象。ツールの関連、意図するところの全体像が見えなかった。</p> <p>計測とフィードバックには注文を出し、フェーズ1、フェーズ2と進化させた。</p> <p>リアルタイム分析、現場に負担をかけない計測の実現に努力した。</p> <p>現場から離れているマネージャと現場に接しているサブマネージャで求められる情報が異なる。情報過多にも配慮が必要。</p> <p>今後、新しいプロジェクトで、プロジェクトでモニターしているぞ、というプレッシャーに使いたい。</p>	<p>ブラックボックスの多い環境下での機能を評価。</p> <p>個別分析機能よりEPMのプロジェクトマネジメント機能を評価。</p>

### 3.6 計測とフィードバックのまとめ

適用前、適用後あるいは適用、非適用を比較するような「制御された実験」ができない、統計処理を行うような多数の事例収集が出来ないような条件下での新手法の有用性を実証するエビデンス獲得のために3.2～3.5節で述べた4種の観測と分析を行った。

プロジェクト全体を俯瞰した観測では、試みた計測とフィードバックの有用性を認識できる事象が定性的に積み重ねられた(3.2節)。

リーダー、サブリーダーへの公式のインタビューでは、フィードバック効果を肯定する回答があった(3.3節)。

開発プロジェクトの成功を第一義に考えるプロジェクト運営の中では、当初モニタ型の観測が許されたただだったが、その効果への肯定的な評価が得られたことにより、最終的には計測とフィードバックを全面的にプロジェクト運営に組み込むことが実現した。

また、プロジェクト後の各企業の振る舞いを追うと、特に義務の無い状況下で、自らの資源を割り振りながら経験した新しい手法を導入しようと具体的な努力を始めたところが多数観測された(3.4節)。

さらに、開発プロジェクト終了後、拘束の少ない条件下で、プロジェクトを構成した様々なキーマンから本音を聞く機会をつくったところ、開発プロジェクトの産業構造上の役割に沿って、高い信頼度で計測とフィードバックを評価する意見が集積された(3.5節)。

全体をプロジェクト参加者のプロジェクト中の立場、そのプロジェクト計測とフィードバックへの評価、その後の振る舞いと言う視点で俯瞰すると、この計測の効果が浮き彫りになる。これを表3-6に整理して示した。プロジェクト参加者には大別して、発注者、ターゲットシステム(この場合、プローブ情報システムプラットフォーム)実現の責任者、各サブシステム担当企業(この場合、研究組合企業)のターゲットシステム実現の責任者がいる。そしてソフトウェア・システムの開発には、全体のPMと各サブシステム担当企業の開発リーダーがいる。サブシステム担当の開発リーダーには、本来プロジェクトマネジメントを行うことがミッションであるが、下部機構まかせで、現場を見ていないリーダー、現場を見ていたいが見れないリーダー、現場に接し現場を把握しているリーダーがいる。そしてプロジェクト参加者として実際にソフトウェアを開発する担当者がある。今回試みた計測とフィードバック施策はこのようなプロジェクト参加者の組織階層の中に、それぞれの役割に沿って効果があったことが認められる。ターゲットシステムに責任を持つ人々には、ソフトウェア開発に煩わされることなく、ターゲットシステムの実現に専念でき、計測とフィードバックについては下部組織から肯定的な評価が上がった。現場の見れないリーダーにはEPMツールのようなプロジェクト管理系のツールが高く評価された。一方、現場に接しているリーダーには個別分析ツールのいくつかが高く評価された。そして全体PMの提言にあるようにフィードバック情報は均一ではなく、情報過多にならないように、それぞれの人間の役割に沿った選択的な情報提供がもとめられることが課題として明らかに

なった。

本論文の研究では、インプロセスのプロジェクト計測とフィードバックの実証実験を通して、様々な立場の人間によって構成されるソフトウェア開発の高度な組織構成に対し、染み渡り渡るようなきめ細かい観測をプロジェクト期間中継続し、おなじく組織構成に染み渡るような計測とフィードバックの効果、そして課題を明らかにすることができた。あえて言えば、ソフトウェア・プロジェクトの Pervasive Observation 手法による Pervasive Effect の捕捉（染み渡るような観察法による、染み渡るような効果の捕捉）である。研究手法として、経営学や産業研究で用いられている「フィールド・リサーチ」と呼ばれる手法をソフトウェア工学の領域に適用したことに相当する[3-5]。このような研究はこれまでの関連研究では発表されることが稀だったものである。

表3 - 6 プロジェクト内の立場と計測とフィードバックへの評価，事後対応

立場	評価		事後対応
	積極的評価	消極的評価	
発注者	ターゲットシステムの実現に満足。 ソフトウェア開発に満足。		次のプロジェクトを企画。 さらに実証事例を積み重ねる施策を進める。
ターゲットシステム 実現の責任者	ソフトウェア開発に煩わされず、ターゲットシステムに専念できた。		開発したシステムの応用、展開を検討する。
ソフト開発PM	<p>ブラインドマネジメントが強いられる中で、プロジェクト管理に役立った。 特にEPM系ツール活用に一定の成果を認める。 開発リーダーとの進捗に関する深い協議、課題発掘時間の短縮に役立つ。 個別分析ツールは未完成でばらばらの印象。現場で使うにはこなれていない。 フィードバック情報は情報過多を避け、均質でなく、管理階層に沿って選択提供することを提言。</p>		<p>新プロジェクトで試行してゆきたい。  「見られている」プレッシャーも活用したい。</p>
サブシステム責任者	ソフトウェア開発に煩わされず、ターゲットシステムに専念できた。 開発現場からは計測とフィードバック施策を評価する声を聞いた。		
開発現場を見ない サブリーダー	下部組織任せで計測施策を評価できない。 結果としてプロジェクト進行に満足。		<p>自社展開を検討。  SECが計画した計測ツール検証計画への参加を検討。</p>
開発現場を見れない サブリーダー	プロジェクト管理に役立った。 EPM系ツールを評価する		
開発現場に接している サブリーダー	<p>個別分析ツールを積極評価する。 特に相関ルール分析とコードクローン分析、スキル分析の可能性を評価。 こなれてない分析ツールもある。 ロジカルカップリング分析、協調フィルタリングによるプロジェクト予測などはまだアカデミック段階。</p>		個別分析技術について大学と共同研究を開始。
開発担当	<p>見られていることをやりがいにつなげ、積極評価。現場プロセスの改良にも反映。 特にコードクローン分析をよいコード作りに反映。プレッシャーにもなった。</p>		上位管理機構へ肯定的なレポートを上げた。

### 3.7 計測実験の成果を反映したプロジェクト計測モデルの提案

ここまで示してきた設計工程以降を対象にした、インプロセス計測と、フィードバックの成果から、今後の課題としてソフトウェア開発工程全体にわたるプロジェクト計測モデルを導き提案した[3-6]。これを図3-5に示す。

開発工程	要求定義	設計		製造			結合試験	保守
		基本設計	詳細設計	プログラム設計	製造	単体試験		
基本計測対象		成果物の記述量				試験の実施量		障害発生量 / 修正量
基本分析		成果物記述量の推移、追加量 / 削除量 / 変更量				障害検出・修正量の推移		修正量の推移
拡張計測対象	新しい研究領域	設計量の計測 設計レビュー記録		ソースコード実体 ソースコード 操作履歴	障害要因 障害関連工程 障害修正状況		保守履歴	
拡張分析		設計量の分析 設計レビュー量、属性、品質		コード分析 履歴分析	障害分析 障害工程分析		保守履歴分析	
		<b>実証実験例</b> 1) 設計書枚数の計測と分析 設計レビュー記録の収集と分析 2) 工程途中までのベンチマークデータと過去のベンチマークデータベースによる協調フィルタリング技術を用いた類似プロジェクト抽出とプロジェクト評価と予測 3) 計測プラットフォーム(EPM: Empirical Project Monitor)による計測と分析。 構成管理システムの履歴情報分析 障害追跡システムのデータ分析 メールリスト管理システムのメール量分析 4) ソースコードのクローン分析 5) コンテキスト情報の収集と計測への反映 チェックリストを用いたリーダーへの面接調査、プロジェクト会議への継続参加					レポジトリ活用 例: ICSE2006併設ワークショップ Mining Software Repository (MSR)の研究領域	

図3-5 プロジェクト計測実験の成果を反映したプロジェクト計測モデルの提案

まずソフトウェア・プロジェクトの全体工程を簡略化して、要求定義，設計，製造，結合試験，保守の工程に分けて考え，そしてインプロセスの計測対象を大別して基本計測対象と，拡張計測対象に分けて考えた。

要求定義から製造工程までのもっとも基本的な計測対象は各工程の成果物の記述量であり，その分析は記述量の推移である。より具体的には，追加量，削除量，変更量の推移である。単体試験，結合試験などの試験工程では何らかの試験の実施量を計測し，検出した障害やその修正量の推移を分析する。保守工程では稼働後の障害や修正の量を計測し，その推移を分析する。

拡張した計測としては，設計工程では設計量と設計レビュー記録が計測対象で，設計量の推移やレビューの量や属性，品質を分析する。製造工程ではソースコードの実体とその更新履歴を対象として，実体とその推移を分析する。試験工程では各種の障害情報にもとづいて障害原因とあわせて，障害の混入工程，発見工程，検出すべき工程などの工程分析を行う。保守工程での拡張した計測と分析は，レポジトリを活用した分析技術が該当する。例えば Source Forge と呼ぶデータベースに蓄積されたオープンソースのソースコードとレポジトリを活用して多くの研究が発表されている。一例として，Mining Software Repository (MSR)と名づけられたワークショップではこの領域の多くの研究が発表されている[3-7]。この工程での計測と分析技術は，ソースコードの更新履歴や障害管理に着目す



るなど、製造工程や試験工程との共通性が高い。

このモデルでは要求定義工程に触れていないが、今後の課題として、この未完の部分埋め、モデルに沿った一貫した計測とフィードバックの環境を実現し、その効果を実証しながら手法とツールの普及を図ってゆくことが課題である。本論文の4章、5章はこの課題に応えるものである。

### 3章 参考文献

- 3-1. 神谷芳樹, 菊地奈穂美, 松村知子, 大杉直樹, 門田暁人, 肥後芳樹, 井上克郎, 松本健一: 進行中のプロジェクト計測とフィードバック実験に基づく計測データベース活用方式の提案, ソフトウェアエンジニアリングシンポジウム2006(ソフトウェアエンジニアリング最前線2006, 近代科学社), pp.35-42, 2006.10.
- 3-2. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Satoshi Iwamura, Yoshiki Higo, Katsuro Inoue, Mike Barker, Ken-ichi Matsumoto: Effect of Software Industry Structure on a Research Framework for Empirical Software Engineering, *Proceedings of 28th International Conference on Software Engineering (ICSE 2006)*, Far East Experience Track, Poster Session, pp.616-619, Shanghai, China, May 2006.
- 3-3. Shuji Morisaki, Akito Monden, Tomoko Matsumura, Haruaki Tamada and Ken-ichi Matsumoto: Defect Data Analysis on Extended Association Rule Mining, *International Workshop on Mining Software Repository 2007 (MSR2007)*, 2007.5.
- 3-4. 森崎修司, 松村知子, 玉田春昭, 大杉直樹, 門田暁人, 松本健一: 相関ルール分析を用いた障害対応データの特徴分析, 奈良先端科学技術大学院大学, テクニカルレポート P.15, 2007.2.
- 3-5. 小池和男・洞口治夫(編): 経営学のフィールド・リサーチ, 「現場の達人」の実践的調査手法, 日本経済新聞社, P.264, 2006.1.
- 3-6. Yoshiki Mitani, Tomoko Matsumura, Mike Barker, Seishiro Tsuruho, Katsuro Inoue, Ken-Ichi Matsumoto: 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, *Proceedings of International Symposium on Empirical Software Engineering and Metrics 2007 (ESEM 2007)*, Madrid, Spain, September 2007 (2007.3 採録, 2007.9 発表予定).
- 3-7. International Workshop on Mining Software Repository 2006. Collocated ICSE 2006, Shanghai, China May 2006.

## 4．要求定義工程のプロジェクト計測実験

これまで述べてきた様々なプロジェクト計測とフィードバックは組み合わせられて考察され、そのプロジェクト運営への反映では、すでに示したように非常に有用性の高い肯定的な結果を得て来た。そこで本論文では、本章と次章で次の段階としてこのインプロセス計測を設計工程より上流の要求定義工程に拡張し、全工程一貫した計測とフィードバック手法の確立を目指す試みとその結果および評価について述べる[4-1][4-2]。

### 4．1 要求定義工程の課題と研究対象

#### 4．1．1 要求定義工程の課題と EA 手法

要求定義工程はいうまでもなく設計工程に先んずる工程である。システムを構築する IT ベンダでは、要求定義のような上流工程と基本設計から製造・試験までの下流工程を別の企業、あるいは別の部門が担当することが多い。上流工程は開発対象業務に通じたコンサルタント系の企業や部門が実施し、下流工程は IT システム技術に通じたソフトウェア企業や部門が実施するのが一般的である。

これまで要求定義工程の計測の研究が進まなかったのは、この工程で用いられる手法が様々で、下流工程に比してそのプロセスや使用されるツールが定まらなかったことによる。そして、この工程における一般性の高い計測がすすまないために、定量データにもとづく議論を進める実践的ソフトウェア工学の立場からの研究も容易でなかった。

このため、要求定義工程については、製造工程になど下流工程に比して、次のような問題があった。

- 1．作業や成果物にかかわるメトリクスが不明確なため、作業の見積もりのよりどころがない。同じ RFP (Request For Proposal: 提案依頼書) に対して桁違いの見積もりが提示されることがある。
- 2．作業の進捗把握が自己申告に基づくものが多く、精度の悪いプロジェクト管理が強いられる。
- 3．工程中あるいは工程終了時に作業の品質を推し量ることが難しい。下流工程を進める中で要件定義工程の品質が明らかになることが多い。上下分離方式の調達では、上流工程担当企業のモラル、モチベーションを高めることが難しい。
- 4．作業の生産性を推し量るのが難しい。

しかしながら近年、分野を絞ることによって、要求定義工程のプロセスや用いられるツールの標準化が進み、下流工程と一貫性のある計測と分析を可能にする条件が整ってきた。そのひとつが政府調達を中心に進められている Enterprise Architecture(EA)と呼ぶ手法を用いた業務システム再構築の分野である。

EA は、企業などの情報システム構築の領域で、組織全体の業務と情報システムを全体最適化の観点から体系的に整理するための方法論である。ザックマンの提唱したフレームワーク[4-3]などを背景として、従来からの手法を再構成したものである。

具体的には AsIs と称して現状を把握しドキュメント化する工程，ToBe としてあるべき理想を描いてドキュメント化する工程，その間に理想に近づくステップを整理し比較的簡潔な文書で表現する工程の3つで構成する．その手法の特徴は AsIs と ToBe の検討成果を階層的に記述した豊富なダイアグラムによって「可視化」し，経営者など調達側を含む関係者の理解性を高める努力をしている点にある．図4 - 1に，我が国政府の省庁システムを中心とした業務システム再構築における EA 手法の概念図を示す[4-4]．AsIs 記述，最適化検討，ToBe 記述の3つの工程と，AsIs と ToBe における階層的表現の概念を表現している．

筆者はプロジェクト計測の視点から，EA 手法による要求定義工程では，そのプロセスと目に見える成果物の様式が標準化されることで製造工程の計測で案出されて来た基本的な手法を応用することが可能と考えた．すなわち，AsIs と ToBe の検討成果を表現するダイアグラムの作成量，追加，削除，変更量の推移をインプロセスで追う方式である．以下にこの考え方に基づくこの領域ではこれまでに例を見ない実証的な試みについて述べる．

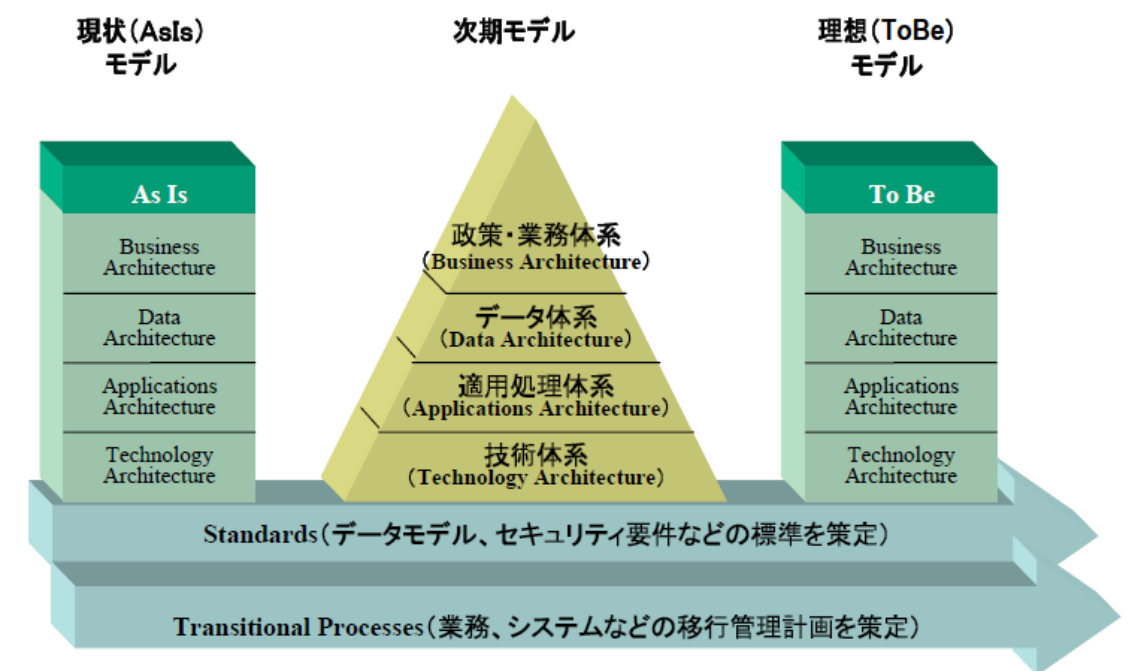


図4 - 1 EA 手法の枠組みの概念図 [4-4]

#### 4.1.2 日本政府の EA 策定ガイドラインと課題

政府は 2003 年に，政府機関の情報システム再構築のために，要求定義工程の進め方として EA 策定ガイドラインを定め，これに沿って政府機関の業務と情報システムの最適化を進めつつある[4-4]-[4-8]．

このガイドラインは、AsIs と ToBe の検討成果の表現に関し、表 4 - 1 に示すような基本体系、参照モデル、EA プロダクト（成果物）を定めている。そして表 4 - 2 に示すようにプロセス体系として、全工程を AsIs 工程、最適化検討工程、ToBe 工程の 3 つに分けて考え、マネジメント体系として EVM（Earned Value Management：出来高管理法）と WBS（Work Breakdown Structure：作業分割構成法）の手法を用いることを推奨している。特に AsIs 工程および ToBe 工程でそれぞれ、機能構成図（DMM：Diamond Mandara Matrix）、情報機能関連図（DFD：Data Flow Diagram）、業務流れ図（WFA：Work Flow Architecture）、実体関連ダイアグラム（ERD：Entity Relationship Diagram）と呼ばれるダイアグラムの大量記述によって要求定義が進められるのが特徴である。図 4 - 2 ~ 図 4 - 5 に 4 つのダイアグラム記述例を示す[4-8]。

表 4 - 1 EA 手法の階層的な表現モデルと EA プロダクト（成果物）[4-4]

基本体系 Basic Architecture	参照モデル Reference Model	EA プロダクト (成果物) EA Products
政策・業務体系 Business Architecture	政策・業務参照モデル Business Reference Model 性能参照モデル Performance Reference Model	業務説明書: Business Description Document 機能構成図: Diamond Mandara Matrix (DMM) 情報機能関連図: Data Flow Diagram (DFD) 業務流れ図: Work Flow Architecture (WFA)
データ体系 Data Architecture	データ参照モデル Data Reference Model	情報体系整理図; クラス図: UML Class Diagram 実体関連図: Entity Relationship Diagram (ERD) データ定義表: Data Definition Table
適用処理体系 Application Architecture	サービス要素参照モデル Service Component Reference Model	情報システム関連図 Information System Reference Diagram 情報システム機能構成図 Information System Function Structure Table
技術体系 Technology Architecture	技術参照モデル Technology Reference Model	ネットワーク構成図: Network Structure Diagram ソフトウェア構成図: Software Structure Diagram ハードウェア構成図: Hardware Structure Diagram

表 4 - 2 EA 手法のプロセス体系とマネジメント体系[4-4]

プロセス体系 Project Process Architecture	AsIs工程、最適化検討工程、ToBe工程 AsIs Process, Optimize Process, ToBe Process
マネジメント体系 Project Management Architecture	Earned Value Management (EVM) method Work Breakdown Structure (WBS) method

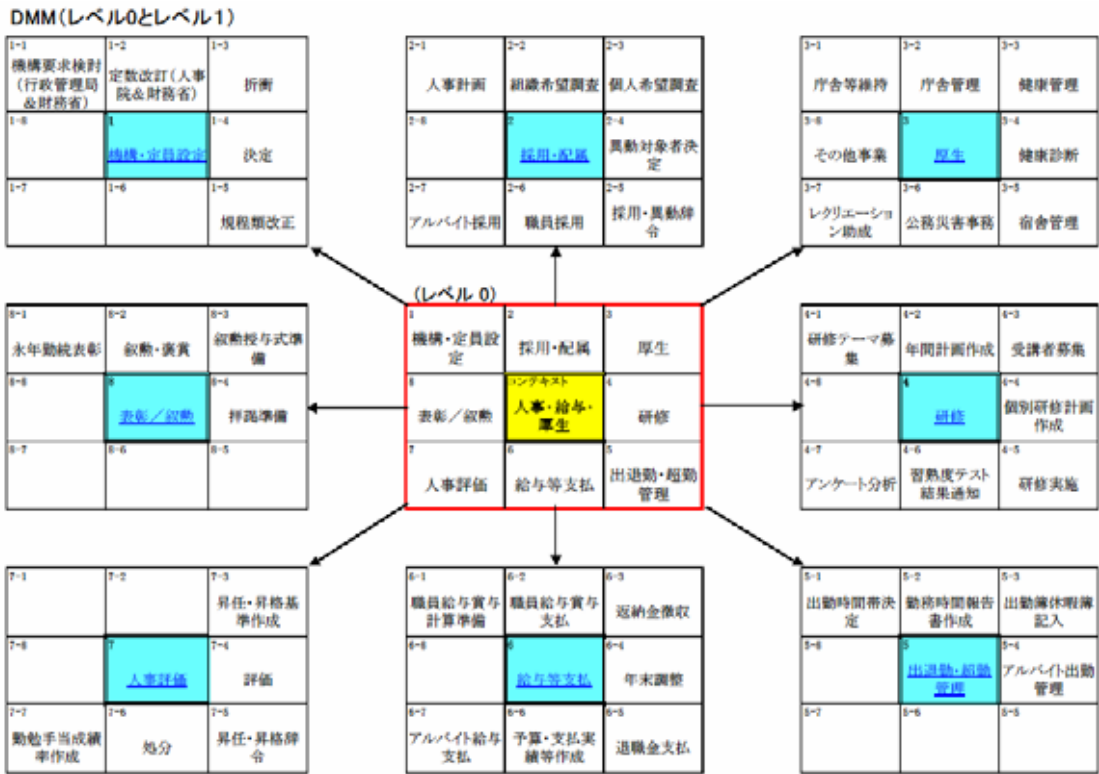


図 4 - 2 機能構成図 : Diamond Mandara Matrix (DMM) 記述例 [4-8]

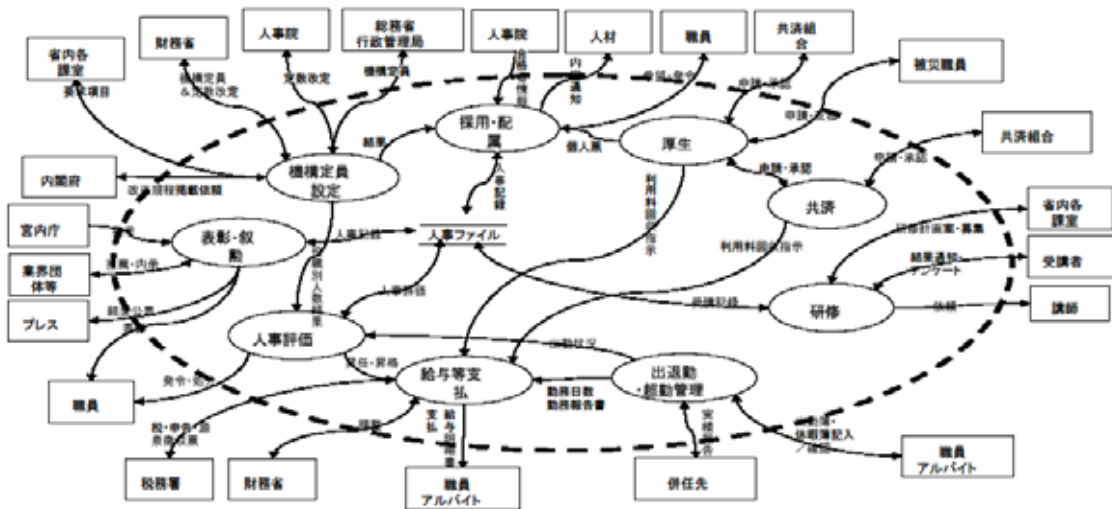


図 4 - 3 情報機能関連図 : Data Flow Diagram (DFD) 記述例 [4-8]

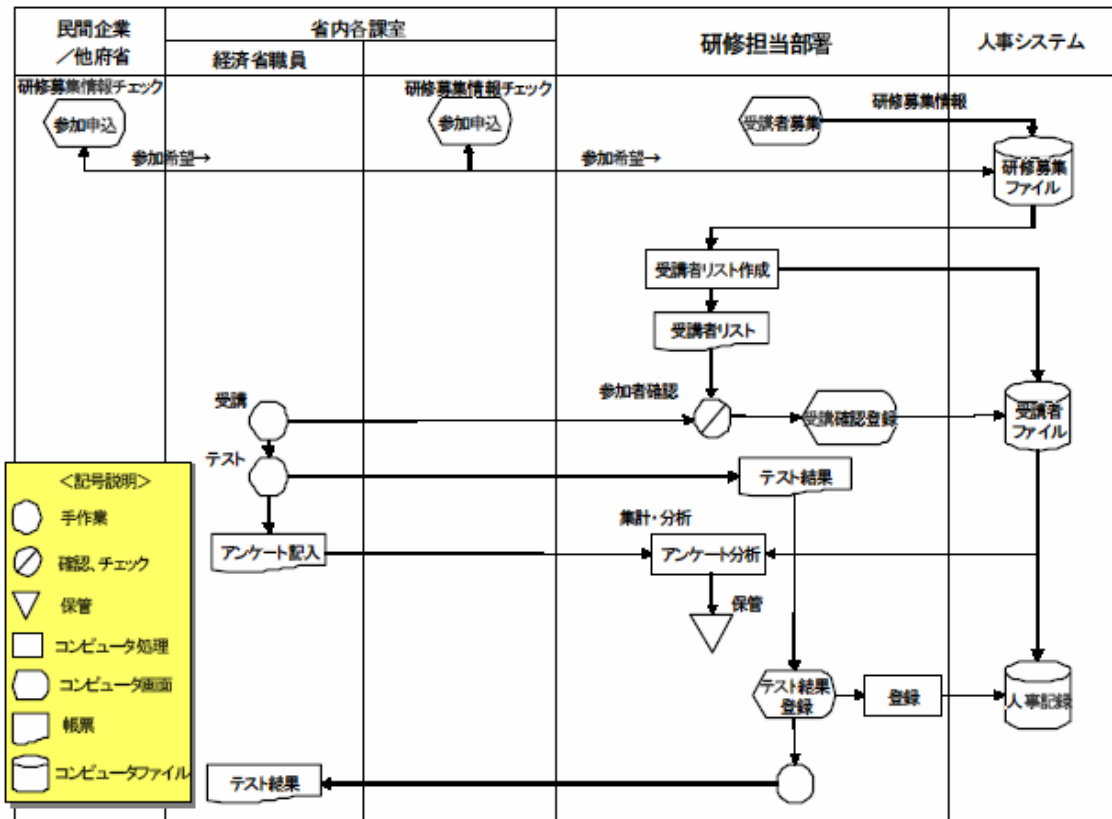


図 4 - 4 業務流れ図：Work Flow Architecture (WFA) 記述例 [4-8]

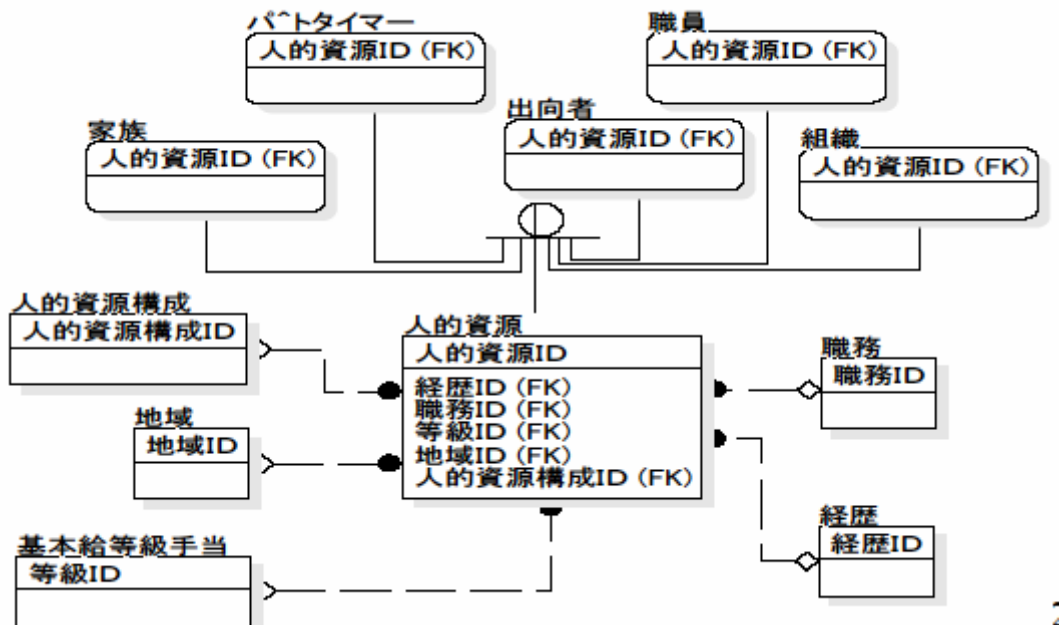


図 4 - 5 実体関連ダイアグラム：Entity Relationship Diagram(ERD) 記述例 [4-8]

政府のこの施策は中央省庁を対象とした試みが一巡し、独立行政法人などの外郭団体および地方自治体への展開が始まり、さらに省庁では方式を見直しつつ二巡目に入ろうとしている。一巡した段階での課題として、政府の連絡会議で、このガイドラインの課題と対策の提示を目的としたワーキンググループから次のような報告が行われている[4-9]。

- 1) プロジェクト推進体制について。
  - 合議制のため、推進力がない。
  - 組織の役割が建前と実際が異なっている。
  - 発注者側として持つべきスキルが不足している。
- 2) プロジェクト管理方法について
  - EVM の導入である程度の可視化が実現した。
  - プロジェクトのリスクを理解している人が少ない。
  - 受注側も進捗をよく見せようとする傾向がある。
  - 調達側に何を管理してよいか理解できていない人が多い。
  - 管理資料の書式が千差万別で、各要件の比較がしにくい。
- 3) 情報共有について
  - 情報共有がほとんどできていない。

これらの報告を概観すると、その課題は政府組織に限定することなく一般的で、下流工程と合わせて上流工程にも政府のガイドラインを適用してもなお多くの課題が残っていることがわかる。特に、プロジェクト管理や情報共有の課題はプロジェクト計測とその可視化による効果が期待できる領域である。

筆者は研究対象として、この EA 手法にもとづく 1 つの政府系プロジェクトについて調達側の一員として計測機会を得た。

## 4.2 実証対象プロジェクトとプロジェクト計測

### 4.2.1 対象プロジェクトの概要

計測対象プロジェクトは独立行政法人情報処理推進機構 (IPA) の業務システム最適化計画である。職員約 200 人の独立行政法人の 4 つの業務システム、すなわち財務管理システム、情報処理技術者試験管理システム、提案公募業務管理システム、セキュリティ認証業務管理システムを最適化するための要求条件を定義した。作業期間は 2006 年 8 月から翌年の 3 月末まで、専門のコンサルティング企業と IPA の共同作業で進められた。政府の上下分離の調達政策に沿って、後ろ工程は別のソフトウェア企業から調達することを前提としている。

本プロジェクトでは各業務の説明書と合わせて、表 4 - 3 に示す 34 種類のダイアグラムの作成が計画された。機能構成図 (DMM) など 9 種類のダイアグラムなどの文書を 4 業務の AsIs 用と ToBe 用に作成する。このうちハッチングで示した 5 種については外部条件の

変化により計画途中で記述が中止された。業務によって記述ダイアグラムの種類が異なるのは、業務によって後ろ工程にすすむ予定が異なるためである。作業は概略図4-6に示す計画に沿って進められた。課題抽出ののち、AsIs 図面作成工程と、見直し方針作成、見直し計画作成、最適化計画作成といった検討工程、次いで最後の ToBe 図面作成工程の3つの工程がある。

表4-3 対象業務と計画された記述ダイアグラム

対象業務	業務A		業務B		業務C		業務D	
	AsIs	ToBe	AsIs	ToBe	AsIs	ToBe	AsIs	ToBe
業務説明書								
機能構成図(DMM)								
情報機能関連図(DFD)								
業務流れ図(WFA)								
実体関連ダイアグラム(ERD)								
情報システム関連図								
ネットワーク構成図								
ソフトウェア構成図								
ハードウェア構成図								

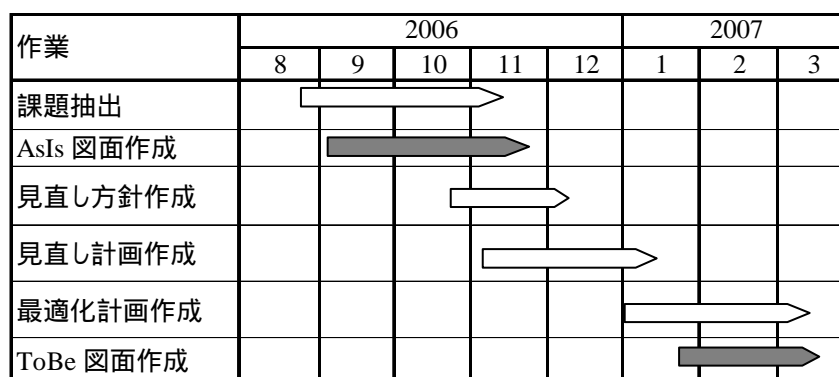


図4-6 概略作業予定

対象プロジェクトは具体的に次のような手順で進められた。

- 1) 調達側(業務担当)と公募により選択されたコンサルタント企業の共同作業体制を組む。
- 2) 既存システムに関するドキュメントと業務手順書をもとに、コンサルタント企業が業務担当へのインタビューによって、既存システムの実態を AsIs ドキュメントとして書き出す。
- 3) 調達側(業務担当)とのレビューと承認作業によってドキュメントを確定する。
- 4) 見直し方針、最適化計画作成として、コンサルタント企業のリーダーシップで調達側内部の TOP までのコンセンサスを作り出し、短い文章にまとめる。



- 5) 定まった方針・計画をもとに, AsIs ドキュメントから ToBe ドキュメントを作り出す。
- 6) 調達側 (業務担当) とのレビュー, 承認作業によりドキュメントを確定する。

#### 4.2.2 プロジェクト計測方式

本プロジェクトでは次の計測を実施した。

- 1) 大量記述される DMM, DFD, WFA, ERD の 4 分類 18 種のダイアグラムの記述量の週次の計測
- 2) 先に述べた製造工程以降の計測で成果のあったプロジェクト初期の段階でのチェックリストによるリーダへのインタビュー調査, およびプロジェクト終了直後のインタビュー調査。
- 3) 同じく先の実証実験で有用だった計測メンバー (すなわち筆者) のプロジェクト会議への継続参加。(筆者は IPA 情報化統括責任者 (CIO) 補佐官として, 調達側の一員としてプロジェクトのマネジメントに参加した)。

1) の各種のダイアグラムは Microsoft 社の汎用的な描画ツール VISIO 上に構成された EA ダイアグラム用のテンプレートを定義した EA 用のツールで記述された。記述量の計測は VISIO のレポート機能と VISIO がデータを出力する先の Microsoft 社の EXCEL を活用し, 一部手作業を含むが極力自動化して集計した。

2), 3) はプロジェクトのコンテキスト情報を得て計測と分析に反映するためのものである。

#### 4.2.3 プロジェクト計測結果

AsIs 工程のダイアグラム記述が開始されて 13 週間に, 3 つの業務についてその記述が終わり, 残り 1 つの業務については, ToBe ダイアグラムを作成しないことになったため工期最後の 2007 年 3 月までに AsIs ダイアグラムの記述が完了した。

3 つの業務の ToBe ダイアグラムについて 2007 年 1 月に記述が始まり, 11 週をかけてすべてのダイアグラムの記述が完了した。

この工程で次の計測とグラフによる可視化を実施した。

- 1) ダイアグラムの記述枚数 (シート数) の推移。全体量と, 業務別。さらに業務別の積み上げ。
- 2) ダイアグラム中のダイアグラム要素数の推移。さらに要素中のコネクタ (フロー, すなわち矢印) の数の推移。全体量と, 業務別, さらに業務別積み上げ。
- 3) ファイル数の推移。ファイル数の削除数, 変更のあったファイル数, 追加ファイル数の推移。全体と業務別。業務別のファイル数の積み上げ。
- 4) サンプルとして, 特定のファイル中のダイアグラム (8 シート) に着目し, その中のテキスト要素に着目した追加, 更新, 削除数の推移。それらのファイル全体の集計。

- 5) 記述枚数の週間増減の推移．全体と業務別推移．
- 6) ダイアグラムシートあたりのダイアグラム要素数の推移．
- 7) ダイアグラム要素数（全体，およびコネクタ数）の週間増減の推移．全体と業務別推移．

これらのグラフの代表例を図4 - 7～図4 - 36に示す．以下，グラフから直接読み取れる事項のいくつかについて述べる．括弧 [ ] 内に計測・分析の項目を示す．

[シート数の推移]

図4 - 7は24週間にわたるダイアグラムの記述シート数の推移を示している．730枚のダイアグラムが記述されている．図4 - 8はダイアグラム中の要素の数，また要素中のコネクタ（フロー，矢印）要素の数の推移を示している．全体で約34,000個の要素が記述されている，いずれもAsIs工程が急速に立ち上がり，ダイアグラム大量記述のあと，検討工程で記述ペースを落とし，そのあとToBe工程で最後まで大量記述が続いたことを示している．

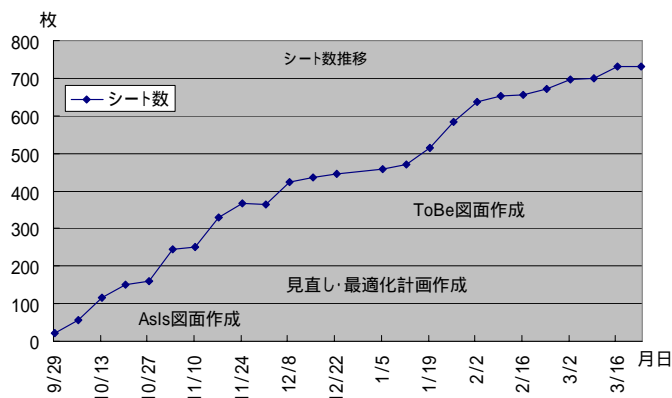


図4 - 7 記述シート数の推移

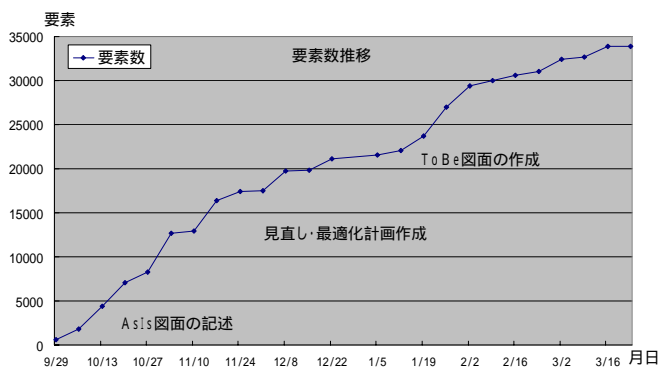


図4 - 8 記述ダイアグラム要素数の推移

[ シート数当たりの要素数の推移 ]

図4 - 9 ~ 図4 - 15は、AsIs 工程と ToBe 工程でのシートあたりの要素数の推移の一部を示している。AsIs 工程では立ち上がり時期にはダイアグラムの密度が一定しない。いずれも一定期間を経たのち安定点を迎え、以降は記述密度としては安定する。業務Dにおいては特異な高密度記述から始まり、転換点を経て記述密度を下げ安定点に至っている。これは同業務の既存のシステムドキュメント記述状況を反映したものと推定される。また ToBe 工程ではいずれも最後まで安定せず、わずかに業務Bにおいて期限ぎりぎりに安定状態を実現しているように見える。これらのことから、進捗把握に記述シート枚数のみを指標としていては不十分であるということが見て取れる。

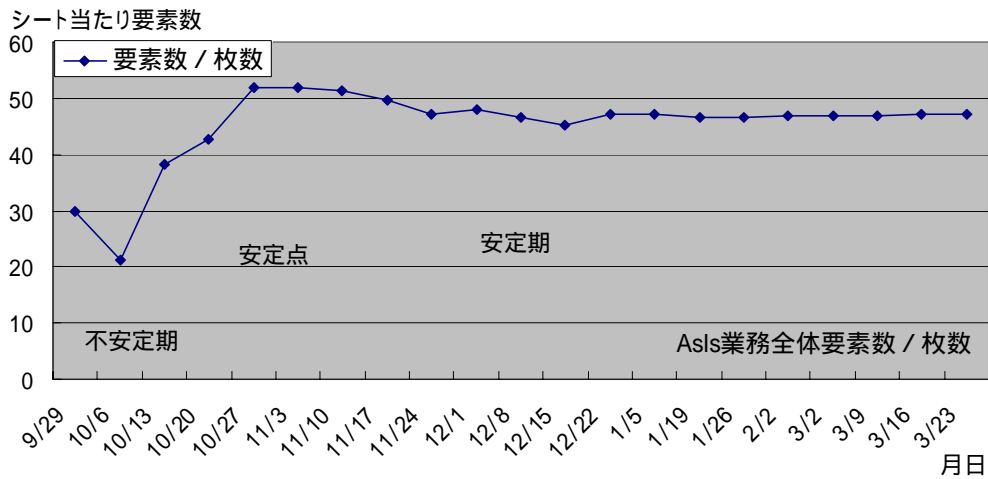


図4 - 9 AsIs 工程業務全体のシート当たりの要素数推移

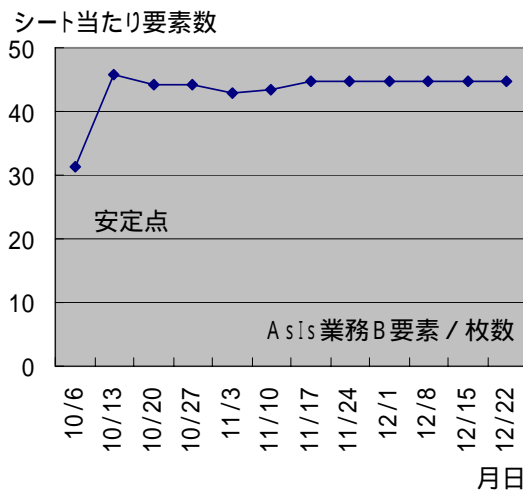


図4 - 10 AsIs 工程業務Bのシート当たりの要素数推移

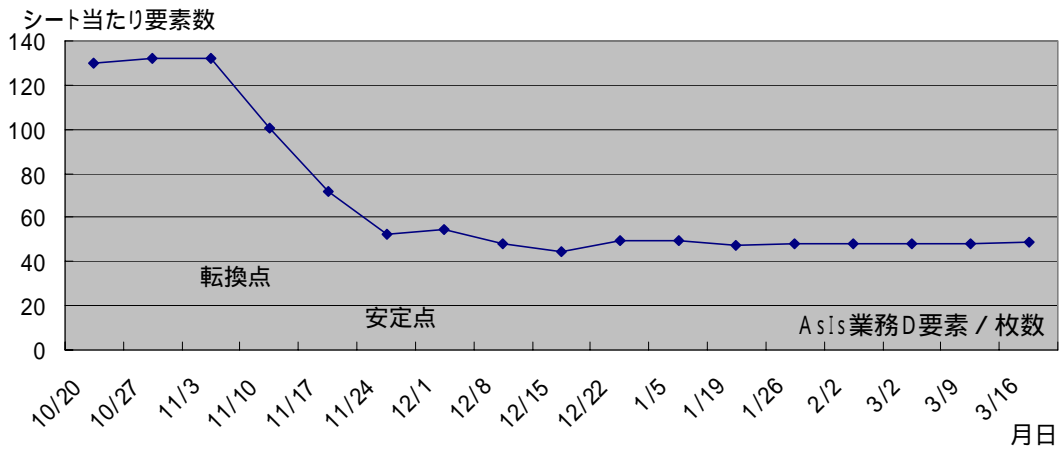


図 4 - 1 1 AsIs 工程業務Dのシート当たりの要素数推移

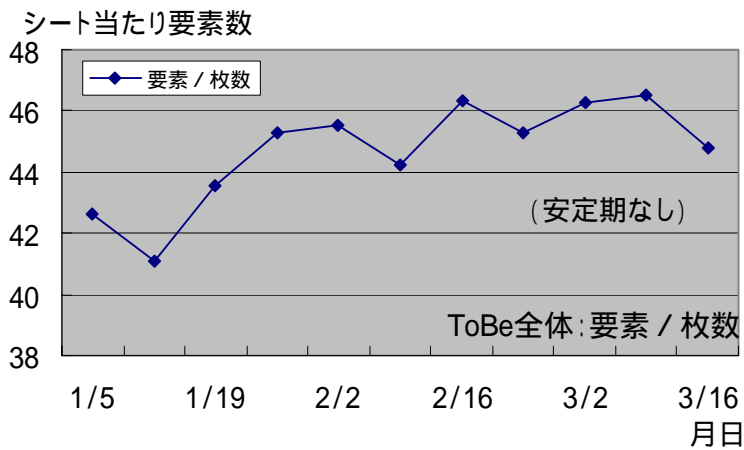


図 4 - 1 2 ToBe 工程業務全体のシート当たりの要素数推移

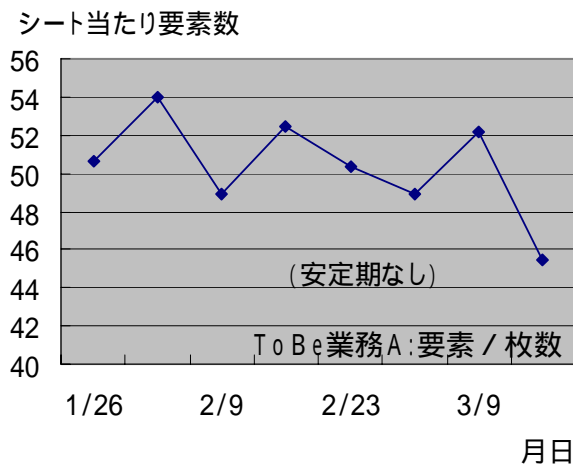


図 4 - 1 3 ToBe 工程, 業務Aのシート当たり要素数推移

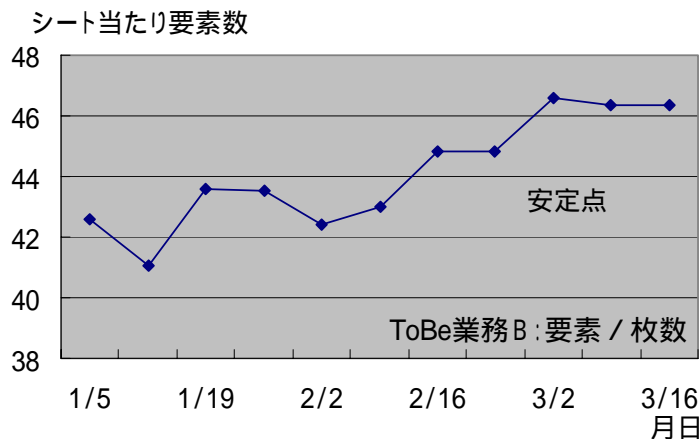


図4 - 14 ToBe 工程，業務Bのシート当たり要素数推移

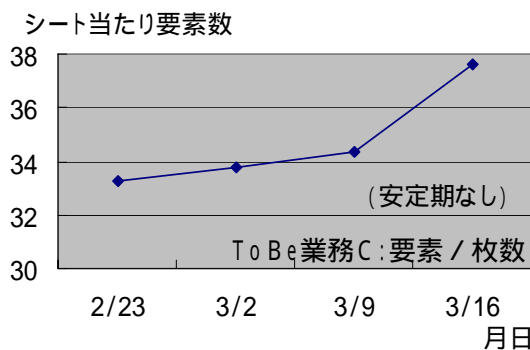


図4 - 15 ToBe 工程業務Cのシート当たり要素数推移

[ 要素数の推移 ]

図4 - 16 ~ 図4 - 20は，AsIs 工程の全体と4つの業務それぞれのダイアグラム要素数とコネクタの推移を示す．コネクタはダイアグラム要素の中で要素を結ぶ要素，具体的には矢印で，その数は総要素数の1/3から1/2を占めている．AsIs 工程では大量記述が続いた後3ヶ月程度で転換点を迎えたが，その後も記述はつづき安定期とはいえない経過である．業務別では，業務A，B，Cが数ヶ月で安定期を迎えているのに対して業務Dは最後まで記述作業が続いている様子がわかる．

図4 - 21はAsIs 工程およびToBe 工程の業務別の要素数を積み上げたものである．これらのグラフから各業務別の記述規模や，開始時期，ほぼ終了し安定期に入った時期を読み取れる．図4 - 16 ~ 図4 - 20のグラフは縦軸のスケールを揃えてあるので相対的な規模を把握できるが，業務固有の傾向を読み取りにくい．そこで図4 - 22 ~ 図4 - 25に図の縦軸をフルスケールにしてこの点を読みやすくしたものを示した．これらから，業務A，B，Cは明確に安定点を迎え工程が収束しているのに対し，業務Dは最後まで工程が続いていることがわかる．

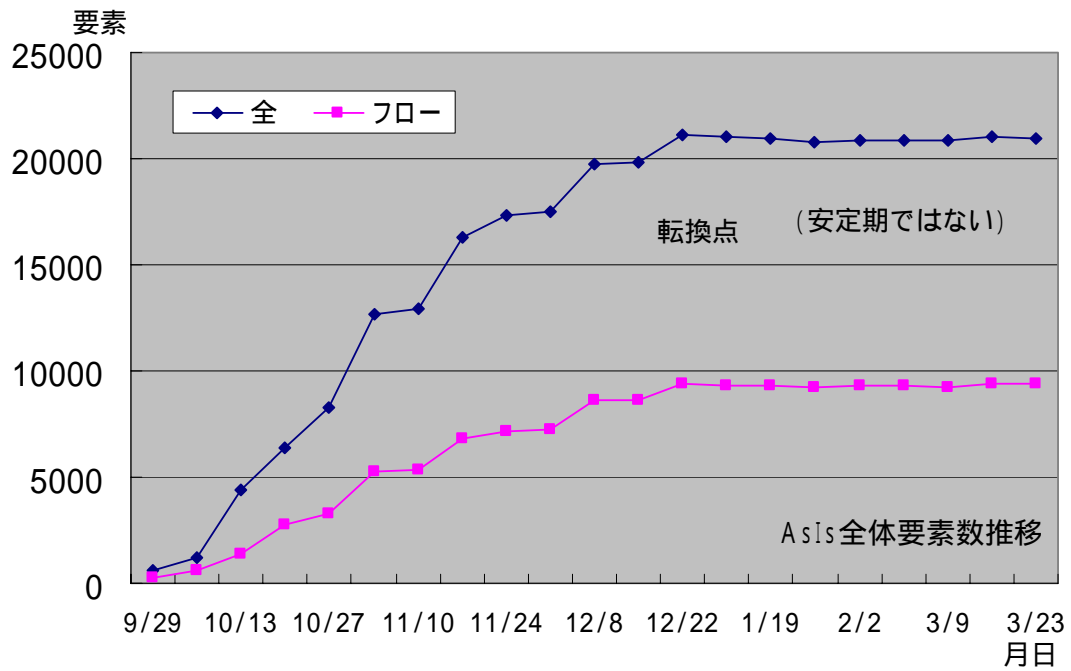


図4 - 16 AsIs 工程4 業務全体の記述要素数推移（フロー：コネクタ（矢印）の意）

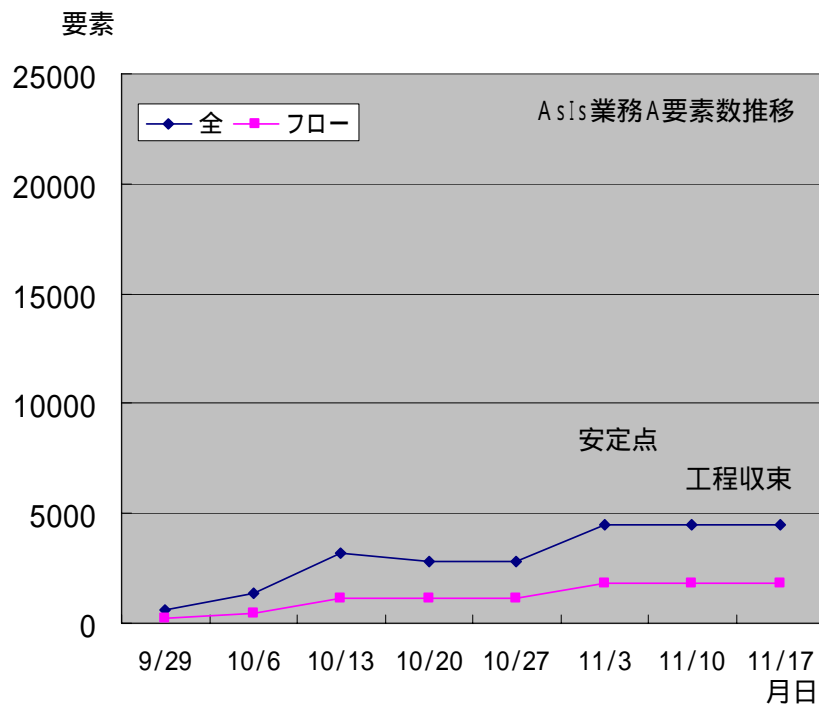


図4 - 17 AsIs 工程業務A 記述要素数推移（縦スケール共通）  
（フロー：コネクタ（矢印）の意）

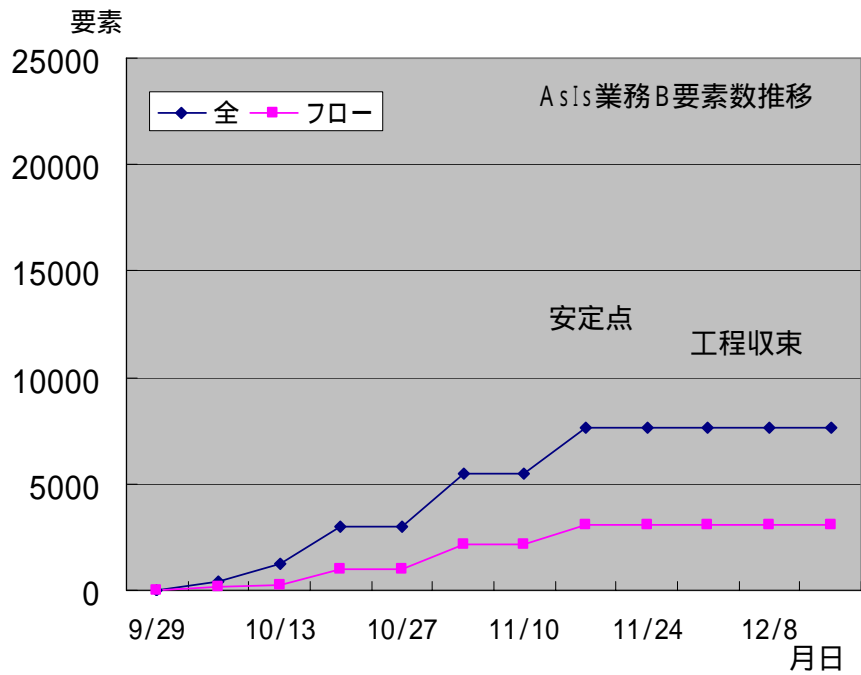


図4 - 18 AsIs 工程業務B 記述要素数推移 (縦スケール共通)  
(フロー：コネクタ(矢印)の意)

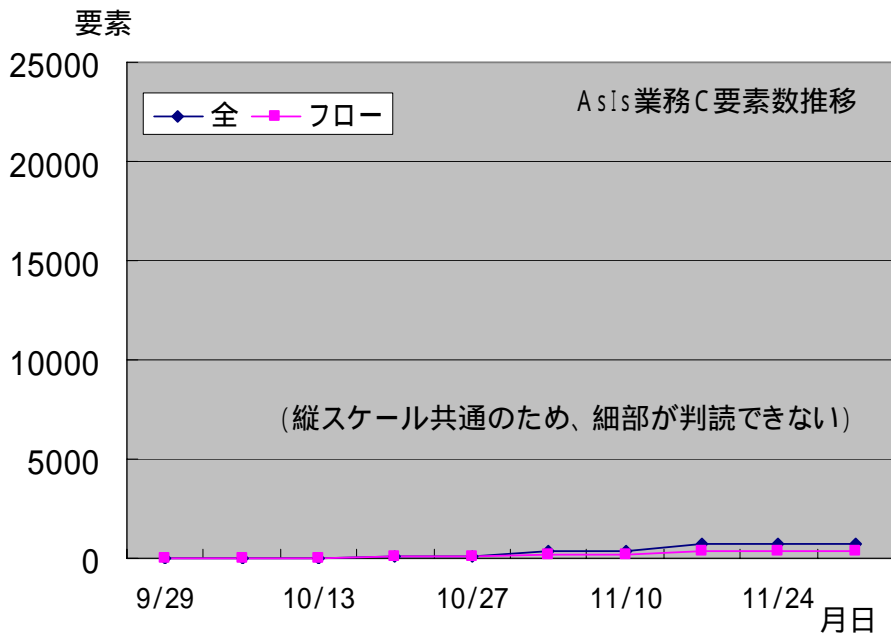


図4 - 19 AsIs 工程業務C 記述要素数推移 (縦スケール共通)  
(フロー：コネクタ(矢印)の意)

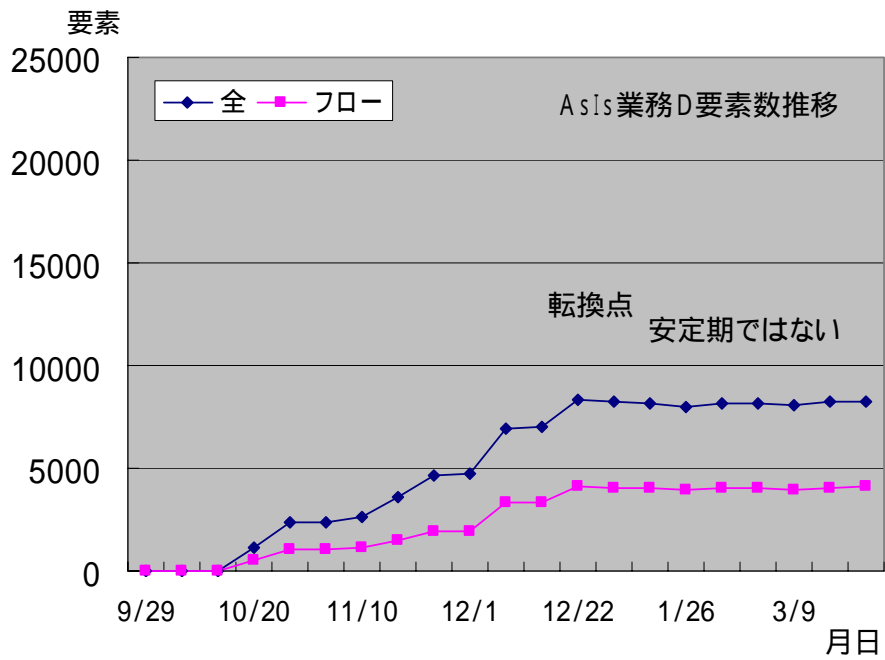


図4 - 20 AsIs 工程業務D記述要素数推移 (縦スケール共通)  
(フロー：コネクタ(矢印)の意)

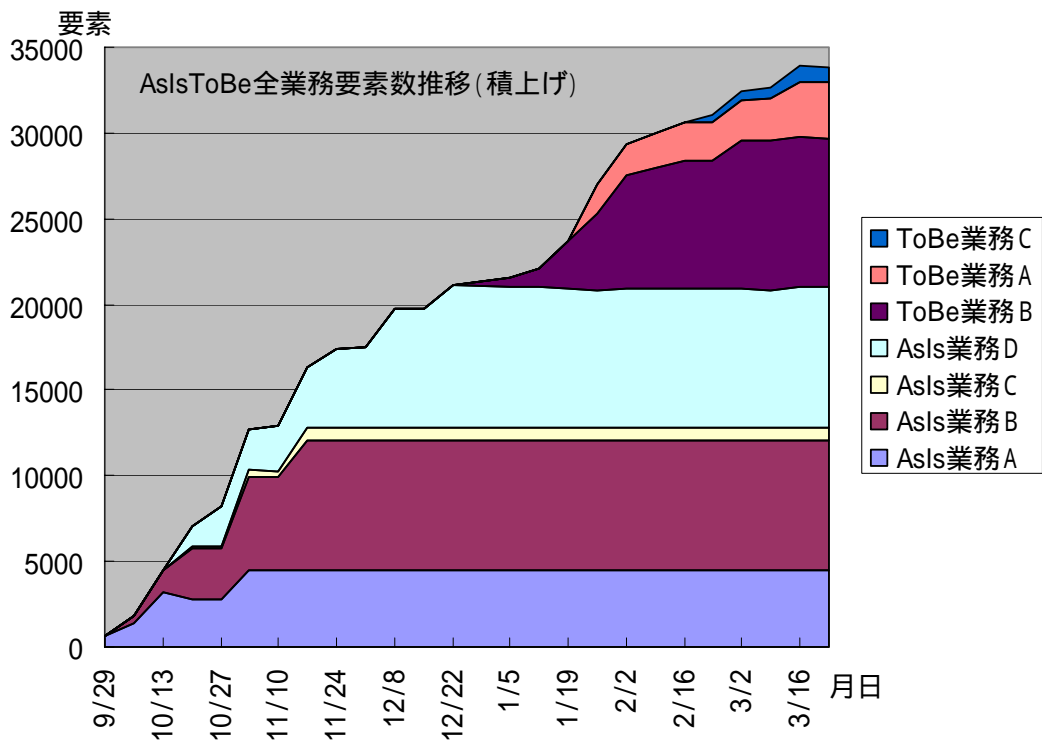


図4 - 21 AsIs および ToBe 全業務記述要素数推移 (全業務積上げ)



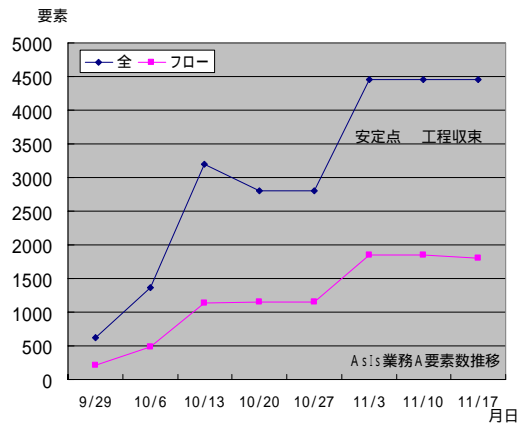


図 4 - 2 2 AsIs 工程業務 A 記述要素数推移 (フルスケール)  
(フロー：コネクタ (矢印) の意)

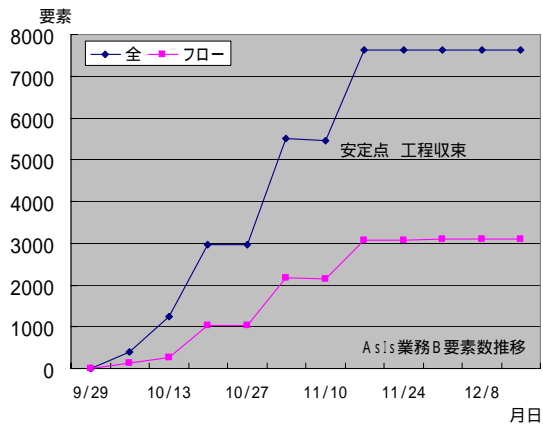


図 4 - 2 3 AsIs 工程業務 B 記述要素数推移 (フルスケール)  
(フロー：コネクタ (矢印) の意)

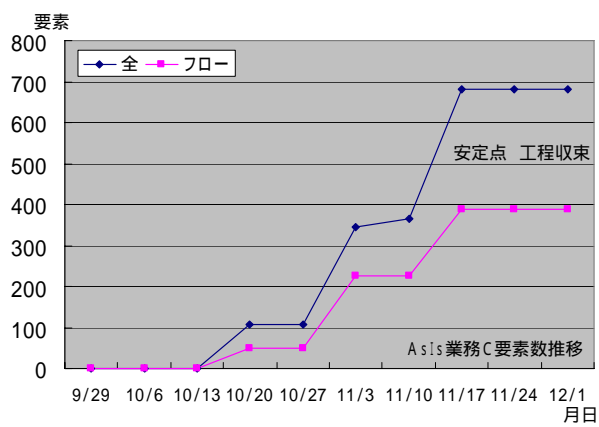


図 4 - 2 4 AsIs 工程業務 C 記述要素数推移 (フルスケール)  
(フロー：コネクタ (矢印) の意)

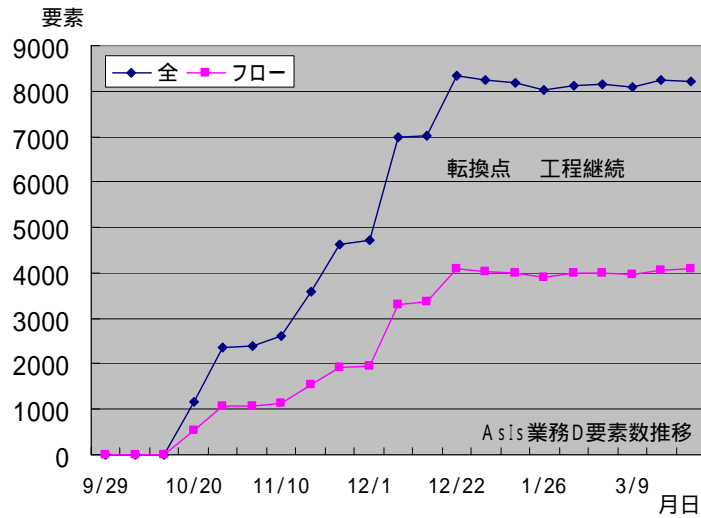


図4 - 25 AsIs 工程業務D 記述要素数推移 (フルスケール)  
(フロー：コネクタ(矢印)の意)

[ファイル数の推移, 追加, 変更, 削除数の推移]

図4 - 26, 図4 - 27は AsIs 工程, ToBe 工程それぞれのファイル数の推移を示している。全体の記述量だけでなく, 追加数, 削除数, 変更数を計測することで作業の進捗と安定度を推し量ることが出来る。AsIs 工程と ToBe 工程で時間軸が異なり, AsIs 工程は期間にゆとりを持って作業が収束したが, 転換点以降も安定状態とは言えず, 最後に修正が行われ, ToBe 工程は時間的に余裕がなく, 期限にあわせて作業を終了させた状況が見える。

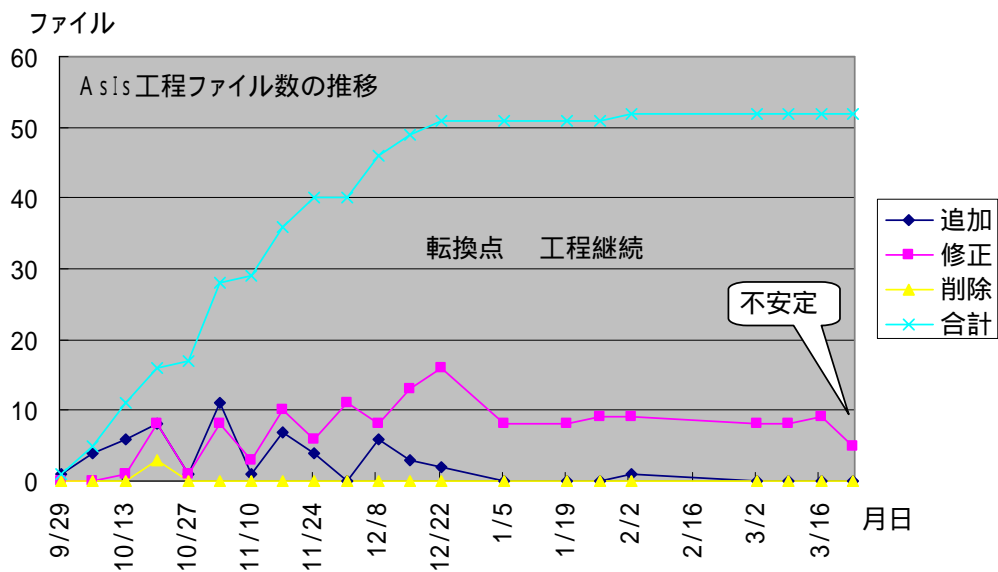


図4 - 26 AsIs 工程の記述ファイル数の推移

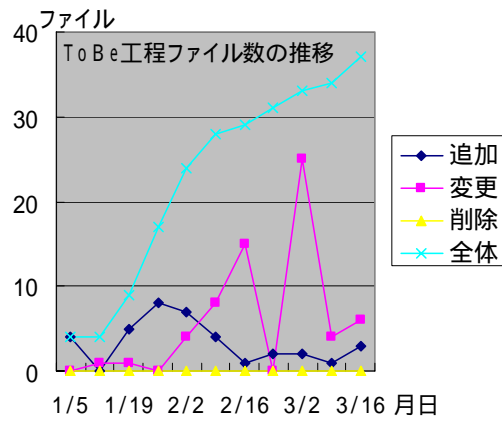


図4 - 27 ToBe 工程の記述ファイル数の推移

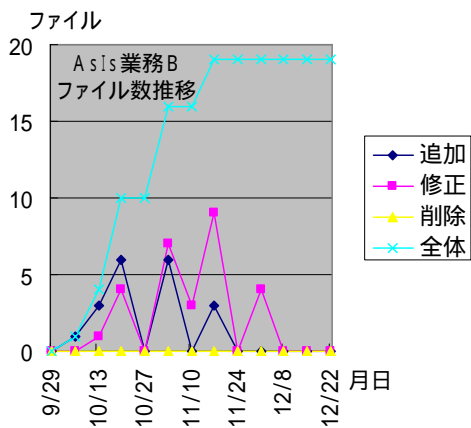


図4 - 28 AsIs 工程業務Bのファイル数推移

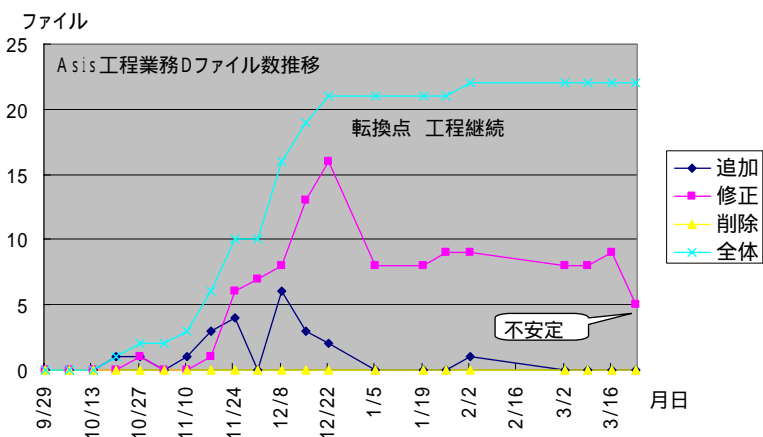


図4 - 29 AsIs 工程業務Dのファイル数推移

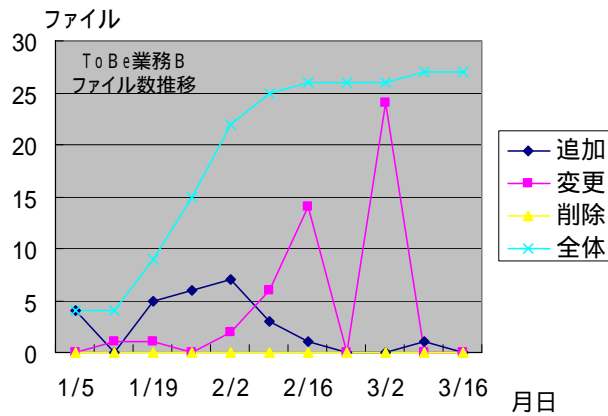


図4 - 30 ToBe 工程業務Bのファイル数推移

図4 - 28 , 図4 - 29は AsIs 工程の業務Bと業務Dについてファイル数の推移をグラフ化したもので、それぞれの業務の進捗と安定度が見て取れる。業務Bは約3ヶ月経過の時点でほぼ終了し安定状態なのに対し、業務Dは同じ時点では記述作業の最中、完了は工期末まで延び、急速に収束させた様子がわかる。図4 - 30は業務BのToBeについてファイル数の推移を比較表示したもので、図4 - 28のAsIsの例と比較することで、それぞれの安定度を押し量れる。すなわち、AsIsは順調に作業が収束し安定状態にあり、ToBeは期限を目指して精力的に作業が進められ収束した様子が伺える。

[シート中の要素の追加，変更，削除数の推移]

図4 - 31 ~ 図4 - 33は AsIs 工程の業務D中の1枚のシートに着目して、その追加，削除，変更を6週間にわたって追ったものの3例を示している。ダイアグラムの完全な更新状況を追うのは難しいが、ここではダイアグラム要素中に記述されるテキスト表示列の追加，削除，変更数を計測している。図4 - 34は、この3シートを含む同一ファイル内の全シート、この例では8シート分の要素更新量を追ったものである。この例では、ダイアグラム作成がダイアグラム要素の追加や修正を通して収束してゆく様子が見える。

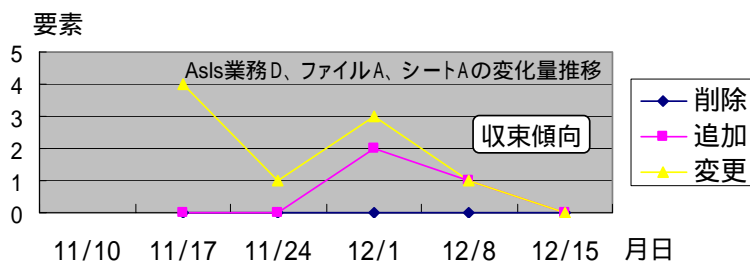


図4 - 31 AsIs 工程の1シートの変化量推移例 (A)

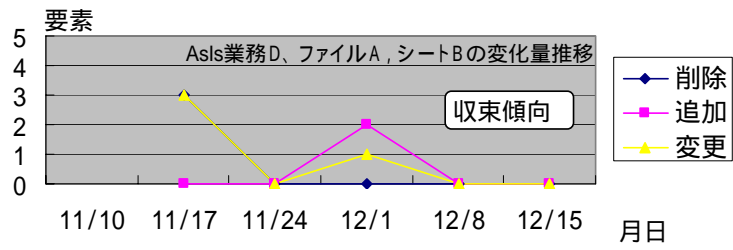


図 4 - 3 2 AsIs 工程の 1 シートの変化量推移例 ( B )

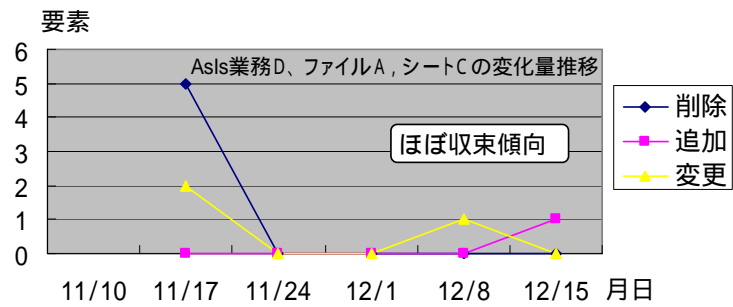


図 4 - 3 3 AsIs 工程の 1 シートの変化量推移例 ( C )

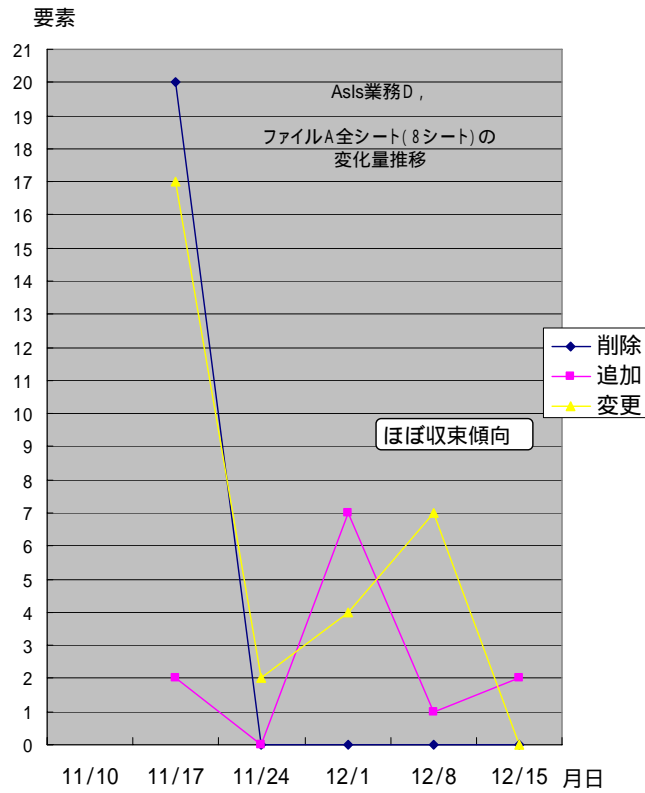


図 4 - 3 4 AsIs 工程の 1 ファイル ( 8 シート ) の変化量推移例

[ 週間記述量の推移 ]

図4 - 35は、AsIs工程で4つの業務別の週間の記述量を要素数で計測したものである。4業務のAsIs作成工程が数週間ずつ時期をずらして進められた様子が、記述量と合わせて見て取れる。図4 - 36はAsIs、ToBe合わせて全期間の週間記述要素数をグラフ化した。AsIsとToBeの2つの山が見え全体の作業進捗状況を俯瞰することができ、表4 - 6で示した計画と対比しながら、作業全体の進捗状況を見通すことができる。後半のToBe工程の期間にも一部のAsIs工程の記述が続き、またToBe工程が急速に収束されている様子が見える。

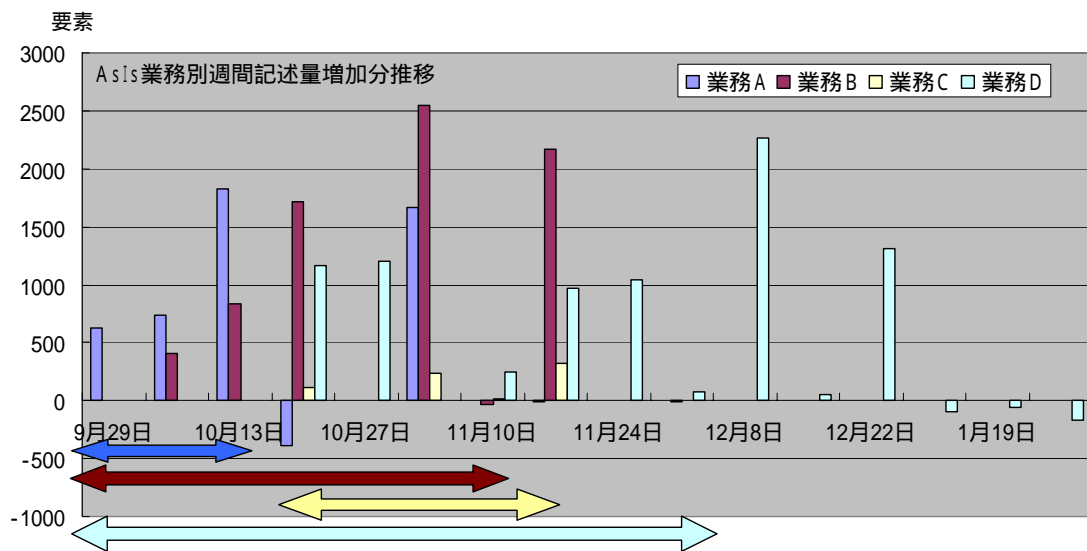


図4 - 35 AsIs 業務週間記述要素数増加分推移と記述作業予定期間

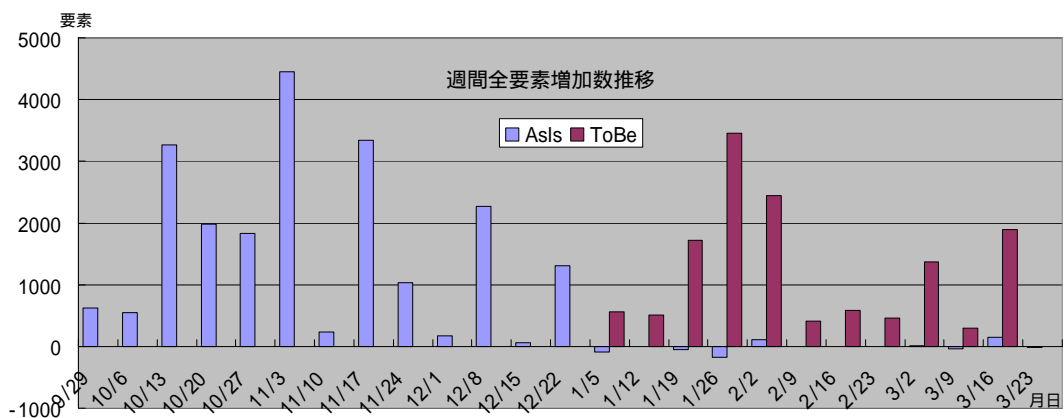


図4 - 36 週間記述要素数増加量の推移

#### 4 章 参考文献

- 4-1. Yoshiki Mitani, Tomoko Matsumura, Mike Barker, Seishiro Tsuruho, Katsuro Inoue, Ken-Ichi Matsumoto: An Empirical Study of Requirement Definition Process Management and Metrics based on an In-process Measurement Experiment of Standardized Requirement Definition Phase. *Proceedings of International Conference on Software Process and Product Measurement (IWSM-Mensura 2007)*, Palma de Mallorca, Spain, November 2007. (2007.8 採録, 2007.11 発表予定) .
- 4-2. Yoshiki Mitani, Tomoko Matsumura, Mike Barker, Seishiro Tsuruho, Katsuro Inoue, Ken-Ichi Matsumoto: 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, *Proceedings of International Symposium on Empirical Software Engineering and Metrics 2007 (ESEM 2007)*, Madrid, Spain, September 2007. ( 2007.3 採録 , 2007.9 発表予定 ) .
- 4-3. John A. Zachman: A Framework for Information Systems Architecture. *IBM Systems Journal* vol.26, no.3. 1987.
- 4-4. 村上敬亮 : Enterprise Architecture について , ~ これからの I T ガバナンスと情報サービス産業 ~ , 経済産業省 , 2004.1.19.  
[http://www.meti.go.jp/policy/it\\_policy/ea/data/report/p1/index.html](http://www.meti.go.jp/policy/it_policy/ea/data/report/p1/index.html)
- 4-5. 経済産業省 , 商務情報政策局 , 情報政策課 : EA ポータル ,  
[http://www.meti.go.jp/policy/it\\_policy/ea/index.html](http://www.meti.go.jp/policy/it_policy/ea/index.html)
- 4-6. 経済産業省 , 商務情報政策局 , 情報政策課 : 業務・システム最適化計画について( Ver.1.1 )  
~ Enterprise Architecture 策定ガイドライン ~  
[http://www.meti.go.jp/policy/it\\_policy/ea/data/report/r2/index.html](http://www.meti.go.jp/policy/it_policy/ea/data/report/r2/index.html)
- 4-7. 村上敬亮 : EA 策定に必須の「 参照モデル 」標準化や調達効率化に寄与 , EA 大全 , Part5 , 参照モデル編 , pp.184-185 , 日経 BP 社 , 2004-5
- 4-8. 村上敬亮 : Enterprise Architecture について , ~ EA の策定方法 ~ , 経済産業省 , 2004-1-19.  
[http://www.meti.go.jp/policy/it\\_policy/ea/data/report/p2/index.html](http://www.meti.go.jp/policy/it_policy/ea/data/report/p2/index.html)
- 4-9. 各省庁情報化統括責任者 ( C I O ) 補佐官等連絡会議 , ワーキンググループ 3 最終報告 , 首相官邸ホームページ ( I T 戦略本部 ) , 2007.4.19.  
<http://www.kantei.go.jp/jp/singi/it2/cio/hosakan/dai36/36w3houkoku.pdf>

## 5. 要求定義工程計測の評価、考察と提案

### 5.1 プロジェクト計測結果の評価と考察

この計測のプロジェクトマネジメントへの有用性を確認する視点から計測結果について評価、考察した。はじめに、EA ガイドラインで推奨されている自己申告方式の進捗報告によるプロジェクトマネジメントと比較した。ついで、計測値に基づいた要求定義工程の新しいメトリクス開発の可能性について考察した。

#### 5.1.1 自己申告方式の進捗報告とプロジェクト計測の比較

対象プロジェクトはEAのガイドラインに沿ってEVMとWBSの手法によって管理され、公式の進捗報告は自己申告方式である。図5-1にEVMによる報告例を示す。EVM報告では、費用に換算した稼働の消化状況の予実管理が可能であるが、成果物の作成状況については表現されていない。表5-1にWBSによるAsIs工程の報告例を示す。ToBe工程を含めて実績報告を筆者がグラフ化したものを図5-2に示す。作業はWBSによって細分化され作業毎にあらかじめ宣言された進捗管理基準に沿って報告がなされるが、その粒度は極めて粗く、成果物の量的な情報は得られない。宣言された進捗管理基準は、一番細かいもので着手時20%、作業計画立案時40%、素案作成時60%、内部レビュー終了時80%、完了時100%というものである。またこの報告には人手作業による自己申告の限界がある。WBSの進捗報告では図5-2中に示したように進捗率が急速に80%程度に立ち上がった後、進捗しないまま相当期間停滞する例などが見受けられる。EVMでは図5-1中に示したように長期にわたって遅れていた進捗が納期を目指して急速に遅れを回復した様子が見える。WBSやEVMの報告からは作業や成果物の品質を推し量ることはできない。

これらの公式の報告に比して、本研究で試みた成果物の計測では、作業進捗をきめ細かく把握でき、業務ごとの作業に反映することができる。先に示した図4-35中に、AsIsの例で、WBSで宣言された作業予定期間を併記したが、予定と実際の乖離を成果物の量と合わせて明瞭に読み取ることができる。このように人手介入の余地の無いフィールドデータの集計とその可視化からEVMやWBSにない明瞭な情報を得ることが出来ることを明らかにすることができた。

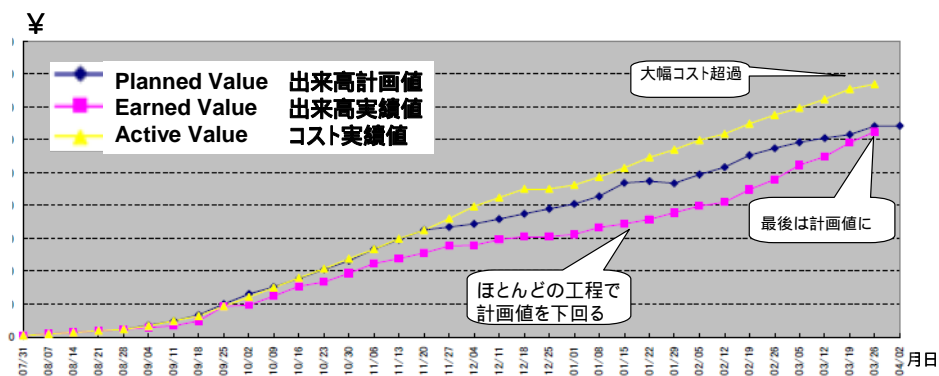


図5-1 EVMによる進捗報告（縦軸の金額をマスクしている）



表 5 - 1 WBS による進捗報告例 (単位: %, AsIs 工程の部分)

進捗打合せ	報告期間	進捗管理基準											
		2回	3回	4回	5回	6回	7回	8回	着手	作業計画立案	素案作成	内部レビュー	完了
業務A	A - DMM	40	100							20			100
	A - WFA	40	100							20		60	80
	A - DFD	0	100							20			100
業務B	B - DMM	40	80	100	100					20		60	80
	B - DFD	0	0	50	90	80	100			20		60	80
	B - WFA	40	80	100	100					20	40	60	80
	B - ERD	0	0	0	0	80	100			20			100
業務C	C - DMM	0	0	60	80	100				20		60	80
	C - DFD	0	0	20	20	100				20		60	80
業務D	D - DMM	70	95							20		60	80
	D - DFD	0	0	0	0	0	20	70	20			60	80
	D - WFA	10	20	80	80	70	80	80	20	40	60	80	100
	D - ERD	0	0	20	20	30	40	70	20				100

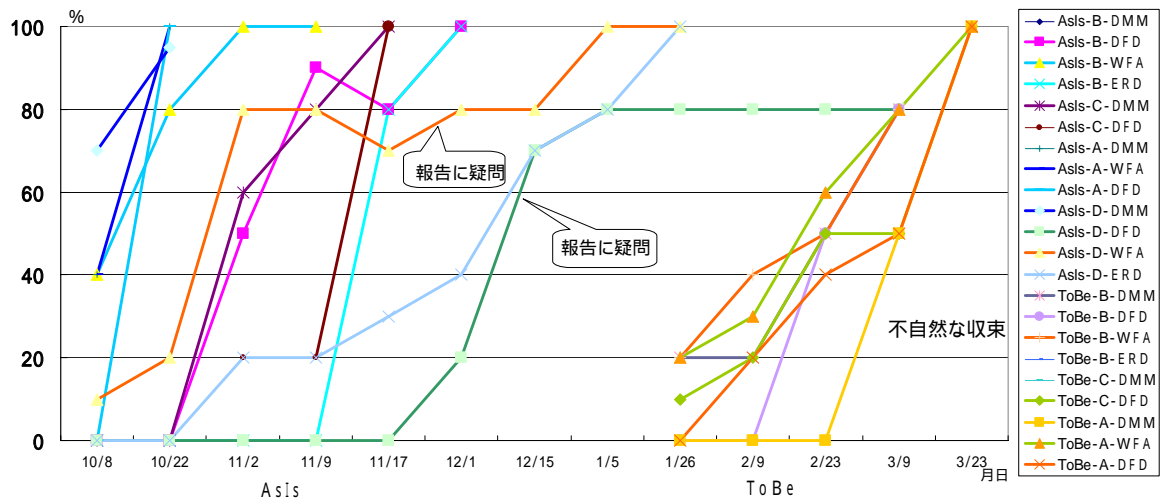


図 5 - 2 WBS による進捗報告のグラフ化 (%) (全工程)

### 5.1.2 計測範囲に関する考察

本論文でのプロジェクト計測は工程中のすべての作業あるいはすべての成果物の量を計測したものではない。4つのダイアグラムだけを計測したのは計測の容易さとあわせて、それが、製造工程での SLOC と同様に、計測対象工程の作業を代表しているという想定による。

その裏づけとして、要求定義工程の作業別稼働比を求めた。全体の作業は図 5 - 3 に示すように 13 種類の作業として管理されていて、このうち約 35% が AsIs および ToBe 作成の直接的な稼働として管理されている。しかしながら全体の 23% を占めるプロジェクト管理作業は全体の作業に按分して考えられ、また約 33% を占める見直し方針作成、最適化計画策定作業は本来 ToBe 作成のための作業と考えられる。こう考えると、計測対象とした要求定義工程の 90% 以上を AsIs および ToBe 作成作業と考えることが出来る。

次に、AsIs および ToBe 作成作業の中で計測対象とした DMM, DFD, WFA, ERD の 4 種のダイアグラム作成作業の稼働比率を求めたところ、図 5 - 4, 図 5 - 5 に示すように、それぞれ 87%, 94% であった。これらのことから、計測した 4 つのダイアグラム作成が、

SLOC が製造工程を代表していると考えられることと相似的に、プロジェクト管理の視点から、この工程を代表していると考えても良い、ということの意味していることができる。

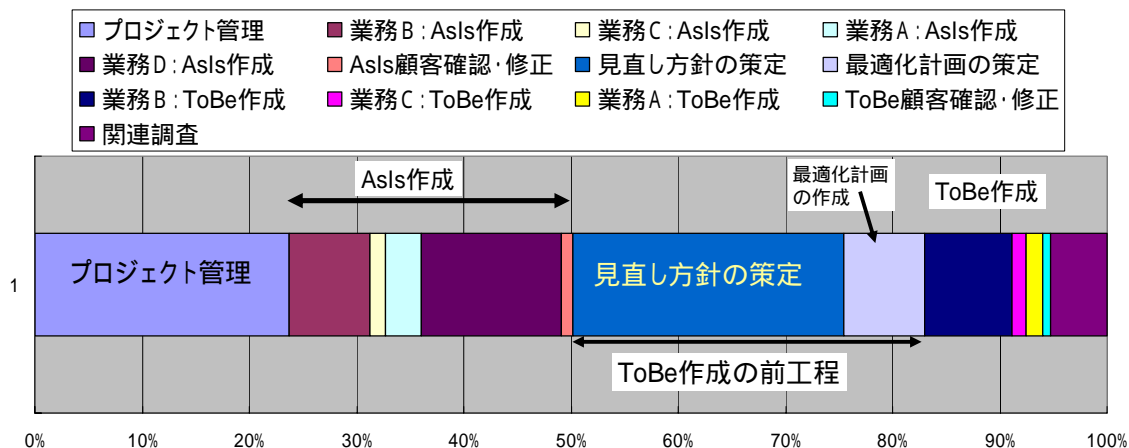


図 5 - 3 作業項目工数比率

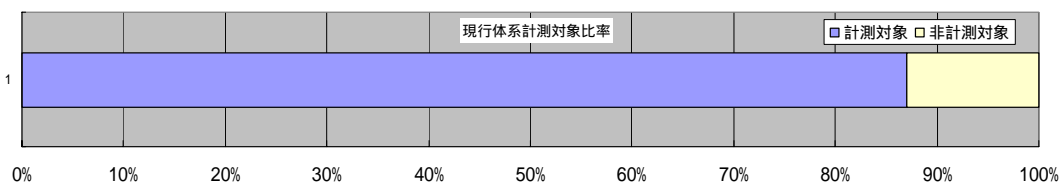


図 5 - 4 AsIs 作成作業項目工数比率

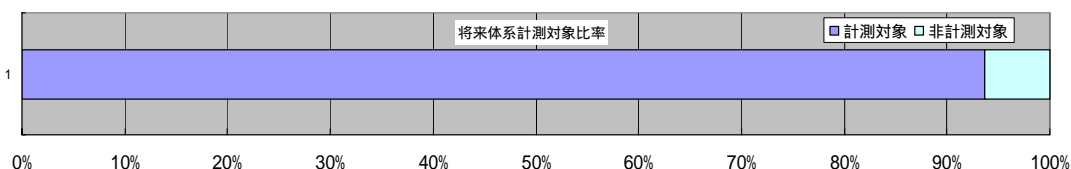


図 5 - 5 ToBe 作成作業工数比率

### 5.1.3 要求定義工程の新しいメトリクス領域開拓に関する考察

本論文における計測対象は大規模開発ではないが、それでもシート数は24週間に700枚を超え、ダイアグラム要素にして30,000要素を超える記述が行われている。この枚数や要素数に対し製造工程で各種メトリクスの基礎となっているソースコード行数(SLOC)やモジュール数の概念から類推することで生産性や品質を図る新しいメトリクスの開発を展望することができる。各ダイアグラムの要素を高水準言語の1ステップと考えれば30,000要素の記述は30KStepのソースコーディングに、またダイアグラムシート数はある種のプログラムモジュール数に、それをいくつか集めたファイル数はプログラムファイル数と相似性

が考えられる。表5 - 2に、ダイアグラムとソースコード行数を基盤としたソフトウェア・メトリクスの対比を示す。

試みに、対象プロジェクトの AsIs, ToBe 工程で記述されたそれぞれ 4 種と 3 種の業務のダイアグラムについてメトリクスの例として、稼働時間当たりの要素数を算出し、AsIs, ToBe それぞれの平均、全体の平均とあわせて図5 - 6に示した。このほかに、枚数あたりの稼働時間、要素数あたりの稼働時間、あるいはその逆数に相当する、時間当たりの枚数や要素数などを考えることもできる。稼働時間はコストに換算することもできる。ここでは 1 つのプロジェクトの 7 つの業務に関する指標が得られたが、この範囲だけでも業務間の作業密度の違いを見通すことが出来る。

たとえば生産性には業務によって 2 倍以上の開きがある。業務 A と業務 C はともに成果物の量が少ないが生産性は異なる傾向を示している。業務 A は、業務内容が容易で成果物の量も少なく生産性が高い結果となり、一方業務 C は、成果物の量は少なくても業務の把握が難しいので低い生産性となったものと推定された。SLOC や Function Point(FP)で計られる製造工程のように、規模の大きい開発ほど生産性が低いといった一般則は適用できない結果となった。

このような指標を多くのプロジェクトから得て蓄積すれば、これまで評価の難しかった要求定義工程の定量的な評価が進展し、新しいソフトウェア・メトリクスの領域の開拓が期待できる。さらにデータの蓄積によって、製造工程のソースコード行数を基礎としたメトリクスと比較することも意味を持つと考えられる。

表5 - 2 ソースコードとダイアグラムのメトリクスの視点からの相似性

	プログラムソースコード	要求定義ダイアグラム
基本計測 単位	ソースコード行数 (SLOC)	ダイアグラム要素数
	(分岐数など)	(ダイアグラムの複雑さ)
	モジュール数	ダイアグラム・シート数
	プログラム・ファイル数	シート・ファイル数
生産性	稼働 / SLOC	稼働 / ダイアグラム要素数
	SLOC / 単位稼働	ダイアグラム要素数 / 単位稼働

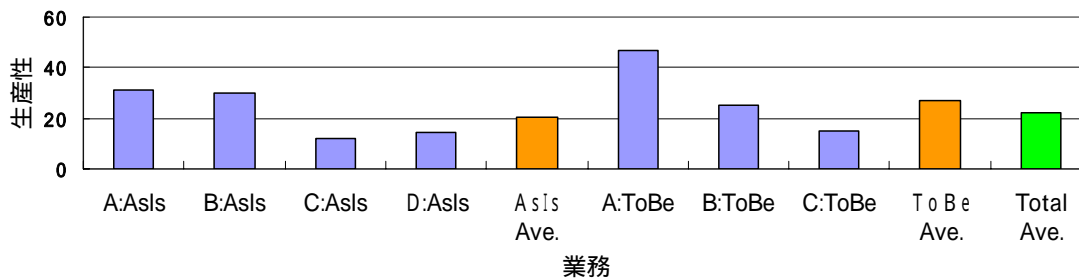


図5 - 6 業務別生産性 (ダイアグラム要素数 / 稼働単位)

#### 5.1.4 ベンチマークデータの分析要素としての可能性に関する考察

##### (1) ダイアグラム要素数と SLOC の相似性からの示唆

ソフトウェア・メトリクスの視点からは、要求定義工程のダイアグラム要素数は、製造工程中のその記述量の推移を追うことが意味を持つ SLOC との類似性を考えることができる。

ソースコードに含まれる意味は多様で、SLOC の定義も厳密ではないが、今日でも SLOC は重要なメトリクスとして意味をもち、使われている。たとえば、IPA/ソフトウェア・エンジニアリング・センターでは定量的なデータに基づくソフトウェア工学施策の一環として、2005 年から毎年、日本の 15 社を超える企業から 1,000 を超すプロジェクトデータを収集し、ベンチマークデータベースを構築している。そして、このデータの分析結果を全 11 章、200 ページを超えるデータ白書として毎年出版している[5-1][5-2]。この白書では、2006 年版の例で表 5 - 3 に示すように、そのデータ分析の中心となる品質や生産性を分析した 4 つの章に 213 個の分析が示されており、そのうち FP に関するものが 75 個、SLOC に関する分析が 70 個収録されている。

FP は SLOC に比して定義が厳格であるが、ソフトウェア開発において静的なデータであって、その量の推移を追うインプロセスの計測に不向きである。一方 SLOC はその計測単位の意味は多様であるが、動的なデータのため製造工程でのインプロセス計測の対象として扱える。こうしたことから、要求定義工程でのダイアグラムの要素数計測の有用性について、ソフトウェア・メトリクスにおける SLOC の位置づけから推定することができる。

表 5 - 3 データ白書中のデータ分析の基本とする数値 (SLOC と FP)

章	分析回数	FP	SLOC	章の内容
5	64	13	13	プロジェクトの主要要素の統計
6	106	48	47	工数、工期、規模の関係の分析
7	18	8	8	信頼性の分析
8	25	6	2	工程別分析、予実分析
合計	213	75	70	

##### (2) 1 プロジェクトのデータでの生産性評価と AsIs/ToBe 比較

今回計測機会を得たのは 1 プロジェクトに過ぎないが、その制限のなかでもこの分野の計測の将来の可能性を示すいくつかの分析が可能であった。

その例として、図 5 - 7 に、1 ダイアグラム要素あたりの単価の相対値を示す。ダイアグラム要素あたりの単価は業務によって大きく異なる。さらに、AsIs と ToBe の差も認められる。2 つの業務では ToBe のほうが生産性が高く、1 つの業務 (業務 B) では逆だった。そして全体では ToBe のほうが生産性が高いという結果になった。これは、一般的な考え方

とは異なる。AsIs は現行業務の状況を書き出すため、将来のあるべき姿を設計する ToBe より作業が容易で生産性が高いはずである。この測定結果から、今回の ToBe 工程の品質が懸念される。また業務 B に関しては、一般の考え方に沿った結果となり、この業務の ToBe 作業が他の業務より丁寧に行われたことが推測される。このことは図 5 - 1、図 5 - 2 の進捗報告や EVM から読み取れる傾向と一致する。すなわち ToBe 工程は、その前の最適化検討工程に時間を要したことから着手が遅れ、スケジュール上の終了時期に合わせるために急いで進められ、ただし業務 B の ToBe 工程は相対的に丁寧に進められたことが見て取れる。

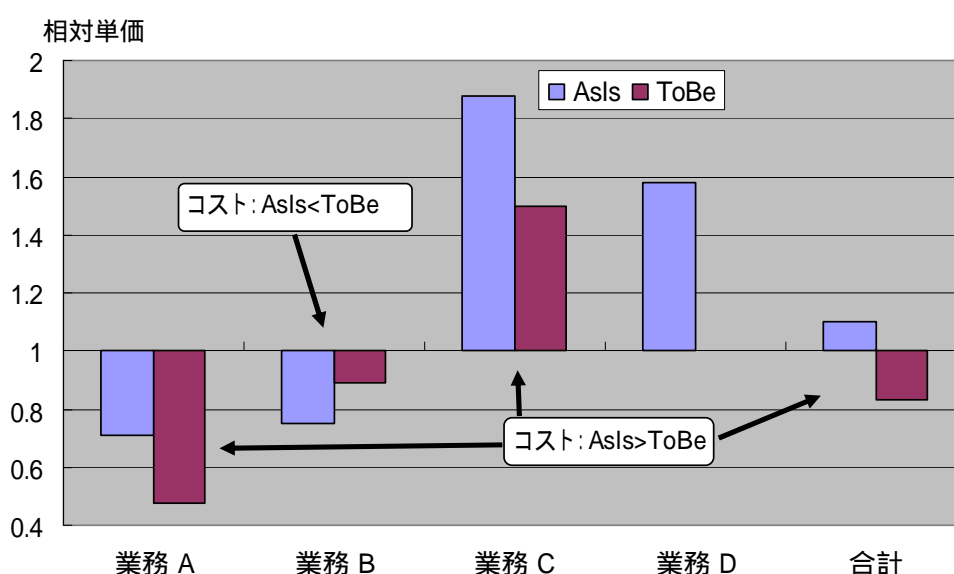


図 5 - 7 ダイアグラム要素当たりの単価 (相対値: 全体平均 = 1)

### (3) ダイアグラム記述の前工程に配慮した分析

さらに ToBe 工程については与えられた条件の中でも、もうすこし深い分析が可能である。図 5 - 3 でも示したように、WBS と EVM による稼働報告では、「AsIs を作成したあとの将来系の作業に関して」、「見直し方針の策定」、「最適化計画の策定」、「ToBe ダイアグラムの記述」という項目で業務別に報告されている。ここまで、図 5 - 6、図 5 - 7 で示してきた業務別の生産性データは直接ダイアグラム記述に要した稼働のみに着目してきた。将来系の作業全体を考えれば、「ダイアグラム記述」のみでなく、それに先立つ「見直し方針の策定」、「最適化計画の作成」も含めて考えることもできる。図 5 - 8 に、図 5 - 3 で示したこれらの見直し方針策定・最適化計画の作成作業を含めた場合と ToBe ダイアグラム作成だけの場合の生産性を比較している。図 5 - 9 は両者の比を示した。また、図 5 - 10 には分布の形で示した。

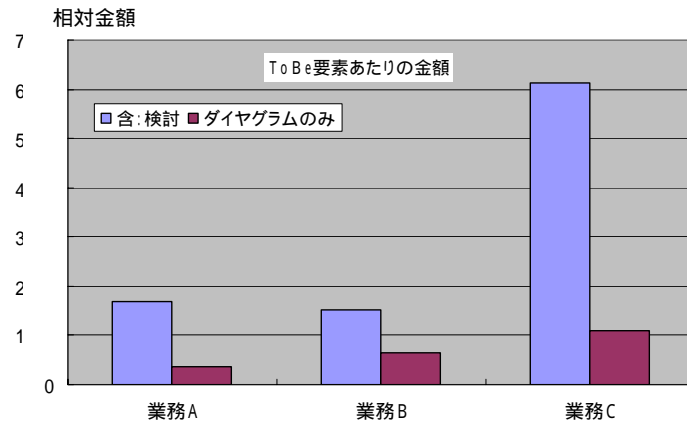


図5 - 8 ToBe 工程の業務別要素当たりの金額  
 検査稼動を含む場合とダイアグラム作成稼動のみの場合の比較  
 (縦軸金額相対値)

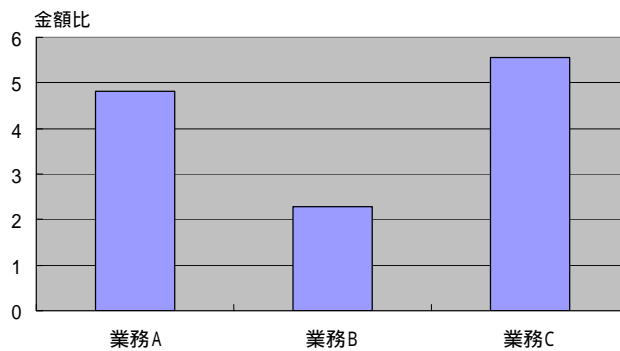


図5 - 9 ToBe 工程の業務別要素当たりの金額の検査稼動を含む場合と  
 ダイアグラム記述のみの場合の比

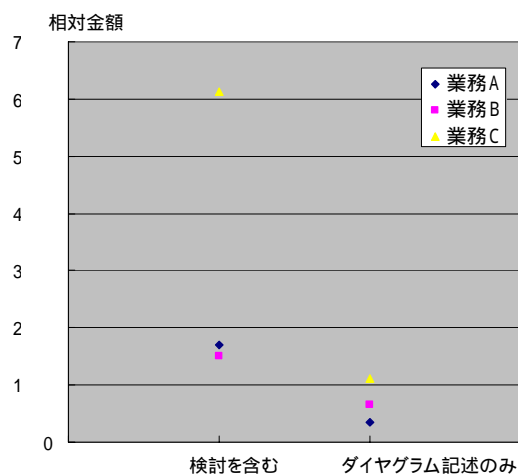


図5 - 10 ToBe 工程の業務別要素当たりの金額分布  
 検査稼動を含む場合とダイアグラム作成稼動のみの場合の比較 (縦軸金額相対値)

ダイアグラム記述のみに着目すれば、業務 A、B、C の順にダイアグラム記述量当たりの生産性が下がるが、検討稼働を含めて考えると、業務 A と B の関係が逆転し、また業務 C が突出して生産性が低いことが見える。検討・策定稼働を含めると業務 A と B の関係が逆転したことは図 5 - 9 から読み取れる。業務 A はダイアグラム記述稼働に対し検討稼働の比が 2 倍以上高く、突出しているように見えた業務 C に近い特性を持っている。図 5 - 8 に示した関係は図 5 - 10 のように分布にして示すとより明確になる。業務 A は絶対的なダイアグラム記述量が少なく、記述されたダイアグラム当たりの生産性は悪くなる。業務を把握し、その将来を検討し方針を策定するには一定の稼働が必要で、これはそのあとに記述するダイアグラムの量とは必ずしも比例関係にない、ということを示しているように見える。この特性は、製造工程と異なる可能性があるが、結論を導くにはさらなるデータと分析の積み重ねが必要である。

検討・策定稼働を含めれば、ダイアグラム当たりの稼働は 2 倍から 5 倍以上増加し、この点を考慮すれば、AsIs 工程よりも ToBe 工程のほうが、生産性が大幅に低いと考えることが出来る。AsIs 工程には先立つ工程、先立つ稼働は計上されておらず、AsIs ダイアグラム記述に必要な稼働はプロジェクト管理以外すべて含めて集計・報告されている。このことから検討・策定作業につぎの 3 つのケースを考える必要がある。

- 1) 検討・策定作業が稼働報告に見合う充実したものであった場合。
- 2) 検討・策定作業が、調達側の意思決定の遅れなどで、多くの待ち時間のような空費された稼働を含む場合。
- 3) 検討・策定作業が、稼働のみ消費し、実りの少ない作業だった場合。

これらについてはダイアグラムのような成果物計測だけでは推し量ることができず、ToBe 工程の生産性に関するメトリクスの精度を上げるためにはさらに観察法や検討会の議事記録の収集などの手段を組み合わせることが必要なことが明らかになった。

#### (4) まとめ

ToBe 工程の 3 つの業務に見られた生産性のばらつきは、このような要求定義工程の見積もりや計画策定の難しさを示している。このような難しいなかで立案された計画に沿って、単純な自己申告方式の予実管理でプロジェクトマネジメントをすすめることの限界も想定できる。こうした視点からもこの領域でのメトリクスの進化、データの蓄積が求められる。本論文では唯一のプロジェクトの 4 つの業務の比較を行ったが、こうした分析から要求定義工程の作業の状況を推し量ることが出来るということを示すことができた。しかしながら、この計測で読み取れた事項は唯一の計測例からの判断であり、この結果だけで断定的な判断は適当でなく、傾向が示唆されたという程度に考えるのが妥当である。計測の有意義なことは明らかに出来たので、今後の課題として事例を積み上げて行くことが重要である。

これらのことから、このような計測データを多数のプロジェクトについて収集、蓄積す

ることで、要求定義工程の生産性や品質に関してより有用な分析結果が得られ、新たなソフトウェア・メトリクスが育つことが期待できる。

#### 5.1.5 要求定義工程計測のまとめ

##### (1) 計測結果と実際のプロジェクト運営の関係

対象プロジェクトはこの計測に依存せず既存の政府ガイドラインに沿って WBS と EVM による管理で進行するように組織されていた。プロジェクト計測はモニター型で計画され可視化したデータは参考資料に留まってプロジェクト運営への直接の反映、影響は認められなかった。

具体的には、プロジェクト管理は自己申告方式の WBS と EVM による報告で進められ、計測データによる管理は行われなかった。前述したように、短期に 80% の進捗率となり、その後何週間もそのままの状態では停滞する報告や作業が容認され、EVM の報告においてほとんどすべての期間で、実績が予定を大幅に下回っていながら、計画期限近くになると急速に予定に近づき、最終的に計画値を満足するという経過であった。政府の方針で上下分離の調達が行われるため、継続する工程は別の企業から調達され、要求定義担当のベンダは後ろ工程に責任を持たない。当初予定されたレビュー記録が全く作られず、作業の品質を推測する手段が無い。もともと要求定義工程はベンダとユーザの共同作業の要素が強く、不安定なユーザ側の条件によって当初計画に沿った進捗が難しく、自己申告方式の予実管理によるプロジェクト進行には限界がある。このような中でプロジェクト計測なしに進め、高品質、高生産性を得ることには無理がある。

このような課題の多い進捗であったが、研究目的のプロジェクト計測の基礎的な作業を進めることが出来、次の段階への基礎的なデータを得ることができた。分析結果は作業現場を知るマネージャにはインパクトは少なく、逆に筆者のように現場に接することのない管理者にとってはプロジェクトの進行を把握する有力な手段となった。

##### (2) 今後の課題とまとめ

本論文でのプロジェクト計測は基本的なもので、3.7 で述べた下流工程でのプロジェクト計測の「基本計測」に対応する。「拡張計測」に対応する高度な計測領域はまだ今後の課題として残されている。

残された大きな課題として次の課題が明らかになった。今後、さらに精度の高い要求定義工程の計測をすすめるにはこれらの課題を解決してゆく必要がある。

- 1) 作業品質について推し量るデータが得られなかった。そのためにはレビュー記録を残し、分析する必要があるが、今回はレビュー記録そのものが作られなかった。
- 2) 要求定義工程はユーザ、ベンダの共同作業で進められるが、ユーザ側の稼働に関するデータを得ていない。したがって、全体の稼働量、その推移、生産性が計測されていない。この課題は多くのソフトウェア計測に共通した課題である。



まとめると、これまで計測の難しかった要求定義工程についてインプロセスの成果物計測を試み、試みた方式が進行中のプロジェクト把握と、製造工程での SLOC に匹敵する新しいメトリクス領域の開拓に高い可能性を持っていることを示すことができた。この計測が可能になったのは、政府主導で、政府計画を対象とした EA 手法の考えに沿った要求定義工程の標準化がすすんだことによる。標準化された要求定義工程で一定期間に大量に記述されるダイアグラムの記述量を計測することは、工程の進捗把握に役立つ。またその分析は当該工程の生産性や品質を推し量る新しいメトリクスを生み出す可能性がある。

先に示した政府の連絡会での指摘事項に照らし合わせると、本論文で示したプロジェクト計測とその可視化は問題解決とはゆかないまでも、次のような貢献が考えられる。

#### 1) プロジェクト推進体制について。

可視化されたプロジェクト計測データの提示はファクトデータを共有することで合議制による作業遅延を抑制する。

組織の役割の論実不一致による根回し時間の長期化も、計測データの共有によって抑制できる。

発注者のプロジェクト推進に関するスキル不足を計測データによって補うことができる。

#### 2) プロジェクト管理方法について。

可視化されたプロジェクト計測データは EVM による報告の限界を克服し、遥かに見通しのよい進捗把握を可能にする。

計測データにより、プロジェクトリスクを把握しやすくなる。

計測データは受注側がプロジェクト進捗を実状より良く見せようとするのを抑制する。

計測データは調達側の管理能力の不足を補う。WBS では把握できない進捗状況を容易に把握できる。

本論文のようなプロジェクト計測の普及は、プロジェクト管理法に関して一定の標準化効果がある。

#### 3) 情報共有について

可視化された計測データの提供はプロジェクト進捗状況の情報共有に貢献する。

### 5.2 ソフトウェア開発全工程一貫計測モデルの提案

#### 5.2.1 インプロセス計測のグランドデザイン

本論文の計測実験により、要求定義工程においても政府の EA 手法のようにプロセスとプロダクトの表現形式を一定の範囲で標準化すれば、2章、3章で示したような下流工程で効果の見られたインプロセスの計測が同様の有用性を持つと考えることができる。

そこでこの成果を背景に 3.7 の図 3 - 5 で示したモデルを発展させて、ソフトウェア

開発の全工程を包む一貫したインプロセス計測モデルを描き図5 - 1 1として提案した [5-3] .

モデルとして、企画・要求定義から保守にいたるソフトウェア開発の全工程を視野に入れ、これに対する計測モデル、それを実証する研究例、そしてこのモデルの適用範囲を示した。企画・要求定義工程では、本研究の4章、5章で示した計測実験を背景にEA方式ダイアグラムの計測方式を、設計・製造工程では2章、3章で示したEPMを中心とした計測プラットフォーム、分析ツール群を駆使した計測とフィードバックを、そして保守工程では、Source Forgeのようなレポジトリを活用した継続的な観測・分析をつないで一貫モデルとした。そのモデルを貫く基本的な考え方は簡素なもので、各工程の主たる成果物の生産量、追加・変更・削除量の推移を追うというものである。要求定義においてはダイアグラムを、設計においては設計ドキュメントを、製造、試験においてはソースコードと試験による障害発生・修復を、保守においては、レポジトリの推移を、と言うものである。設計工程をより強力に計測するために、UMLを活用した方式について次節で述べる。

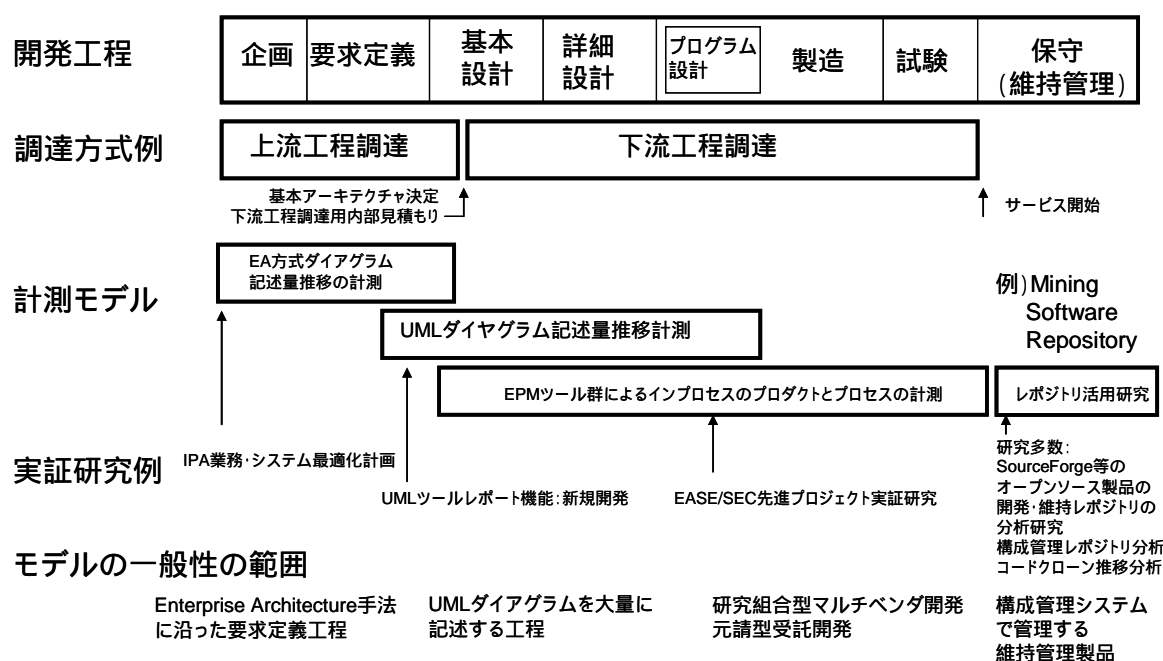


図5 - 1 1 全工程一貫計測モデルの提案

### 5.2.2 設計工程の計測粒度を高めるUMLダイアグラム計測ツールの開発

3章、図3 - 5で示したこれまでの実証実験では、要求定義につづく設計工程での計測対象が、設計書のドキュメント枚数と設計レビュー記録ということで製造工程に比して粒度が粗かった。そこで設計工程に対してここまで述べた要求定義工程での成果物計測成果から、設計工程で大量記述されることの多いUMLダイアグラムの記述量の計測を考えた。

ソフトウェア開発のいくつかの分野では、UML の数多くのダイアグラムの中で、システムの振る舞いを表すユースケース図、アクティビティ図、シーケンス図、システムの構造を表すクラス図を大量に記述する場合がある。このような工程では、要求定義工程の計測と同様に各ダイアグラムの記述量の推移を計測することで、計測の粒度を上げることができると想定される。

表 5 - 4 にこれらのダイアグラムの計測対象を示す。筆者はこの計測のためのツールを開発した。(ツール開発は発注による。) 図 5 - 1 2 に開発したツールの構成を示す。ファイルにバックアップされたダイアグラム群、または構成管理システム(具体的には Subversion) に格納された UML ダイアグラム群を指定して、前章で示した EA ダイアグラム群の分析と同様に横軸を時間軸として設計量の推移、設計量の変化量の推移を可視化表示する。(JUDE は対象とした UML 描画ツールである)。図 5 - 1 3 ~ 図 5 - 1 5 にその出力例を示す。図 5 - 1 3 はユースケース図 3 種の設計量の推移をバックアップファイル群を指定する方式で表示した例、図 5 - 1 4 は同様に 4 種のアクティビティ図の設計量の推移を、構成管理システムのレポジトリを指定する方式で表示した例、図 5 - 1 5 はユースケース図の設計量の変更量の推移を表示した例である。(データはいずれも機能表示用のものである)。このツールを用いた実証実験の実現とともに、プロジェクト計測ツールとして EPM との接続方式を検討中である。

表 5 - 4 UML ダイアグラム計測対象

記述対象	UMLダイアグラム	計測対象
システムの振る舞い	ユースケース図	ユースケース、包含、拡張、アクター
	アクティビティ図	アクション、フロー、フォーク、ジョイン、シグナル、判断・マージ
	シーケンス図	オブジェクト、メッセージ
システムの構造	クラス図	依存、汎化、属性、操作

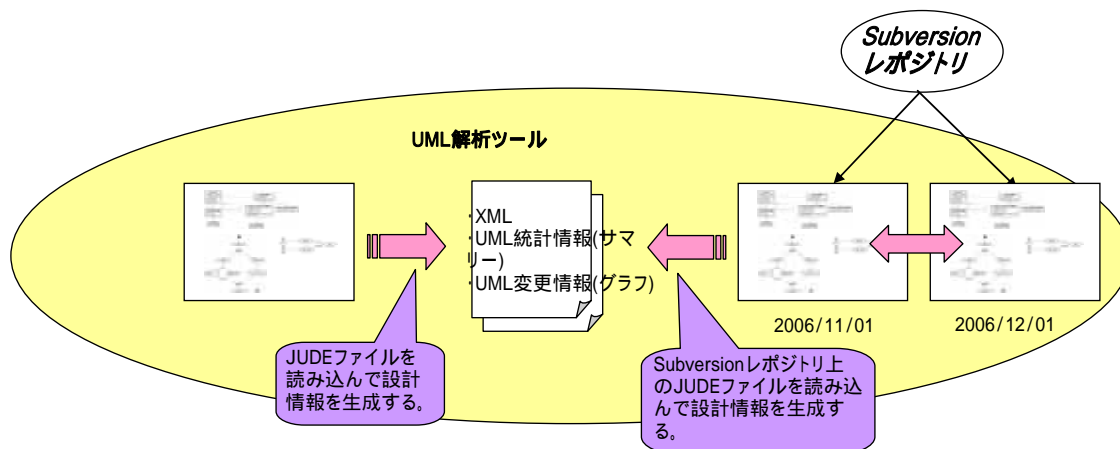


図 5 - 1 2 試作した UML ダイアグラム要素計測ツールの構成

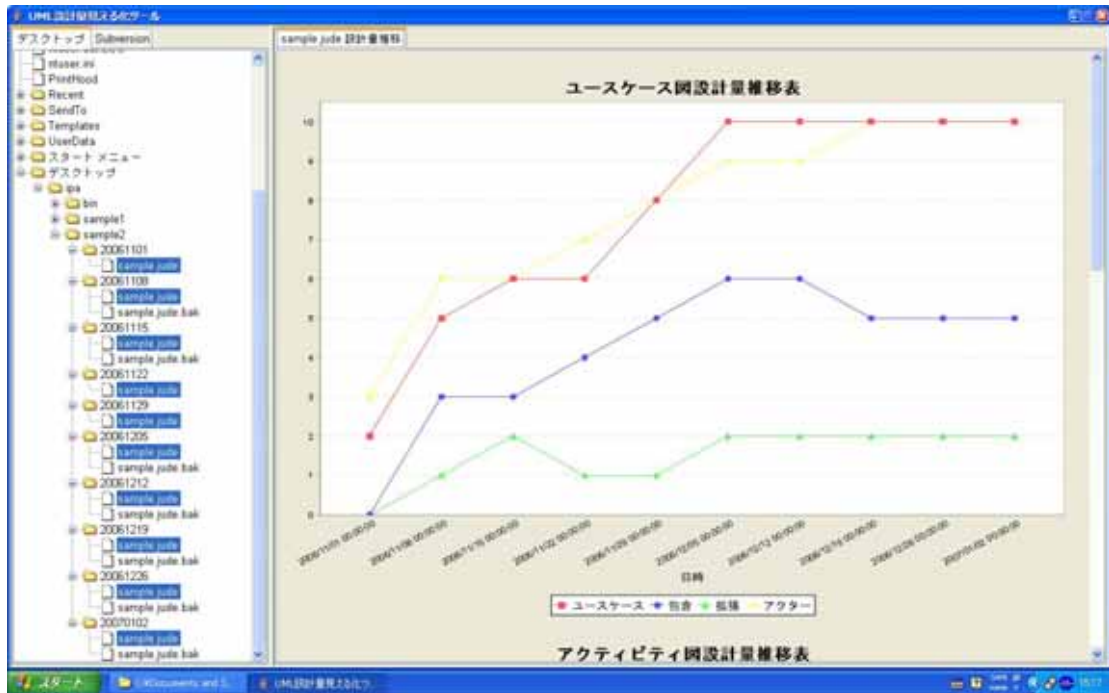


図5 - 13 UML 計測ツール出力例 (ユースケース図設計量推移)

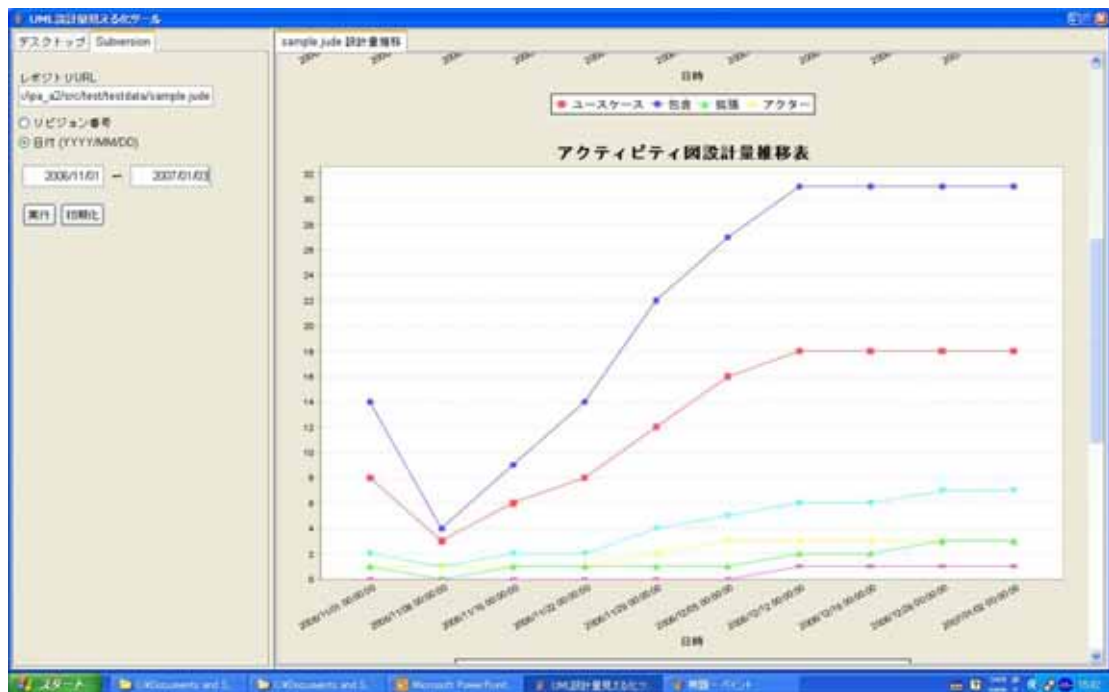


図5 - 14 UML 計測ツール出力例 (アクティビティ図設計量推移)

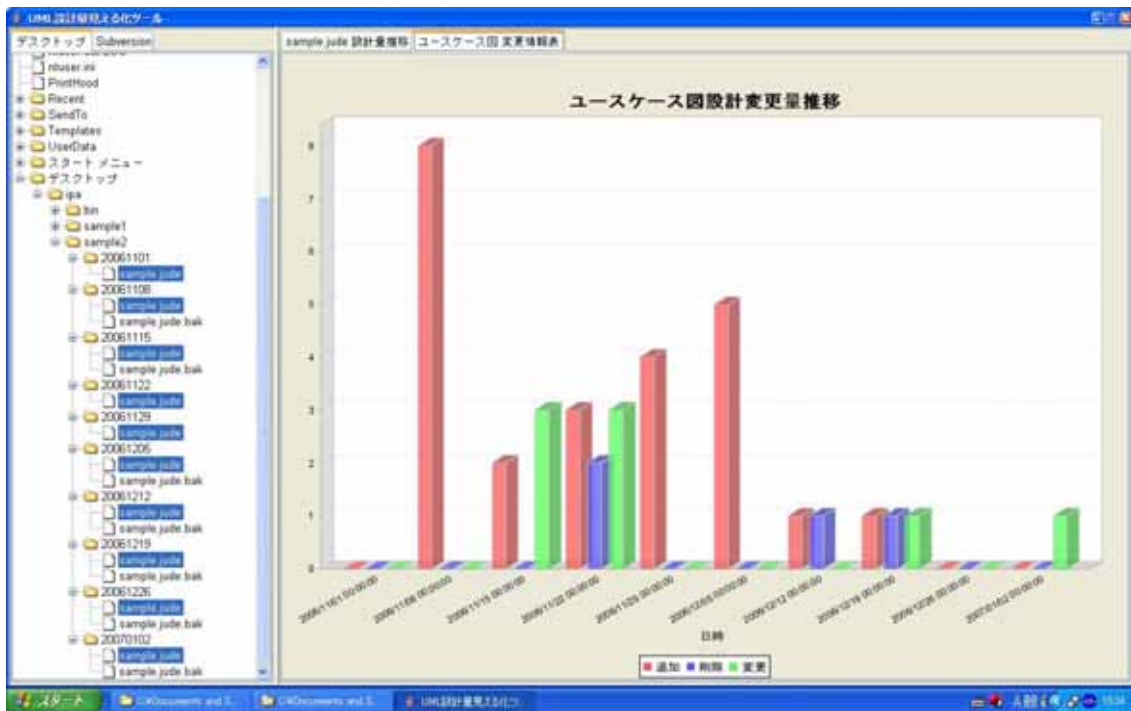


図 5 - 15 UML 計測ツール出力例 (ユースケース図設計変更量推移)

### 5.3 計測データの蓄積を活用したプロジェクト予測機構の提案

本論文で述べてきた進行中のプロジェクトのプロセスとプロダクトの計測データを、プロジェクト終了後システムに蓄積すれば、有益な過去データベースになる。しかしながらこのデータは静的なベンチマークデータと異なり、その活用は蓄積しただけでは検索手段が無く容易でない。

進行中のプロジェクトの計測データは、一般に時間軸を横軸とする様々な指標の推移を見るものが多く、その指標は絶対的な数値よりも、マクロなトレンドに意味を持つものが多い。多くはグラフなど人間の視覚に訴える物理的なパターンがプロジェクト運営に関する判断材料になる。このような計測データを蓄積し、新しいプロジェクトの途中までに得たデータから過去の類似プロジェクトの計測データを抽出しプロジェクト予測に役立てる機構を提案した[5-4]-[5-7]。

#### 5.3.1 ベンチマークデータ収集とナショナルデータベースの構築

IPAのSECでは先に述べたように産業界からプロジェクトのベンチマークデータを収集しデータベースを構築して、その分析結果をデータ白書として出版してきた[5-1][5-2]。

収集データ項目は、収集対象の企業がそれまでに社内で蓄積していた項目、International Software Benchmark Standard Group(ISBSG)[5-8]での項目などをもとに、SECの活動の中で調整・設定した。これらは10分類、約490項目のデータから構成されている。データ

は専用の入力フォームによって収集された。

収集したデータの分析から、規模、工期、工数、生産性、信頼性などの要素別のプロフィールや、基本的なデータ項目間の相関が明らかにされて来た。

### 5.3.2 ベンチマークデータベースの協調フィルタリング

収集したベンチマークデータには、この種の収集データの特徴として多くの欠損データを含み、また多彩なプロジェクトのものが含まれ、その分析には欠損データへの工夫と、層別分類などのカテゴリズが必要である。そこで、人手による分類のほかに、欠損のあるデータセットから類似度の高いデータをグルーピングする手法である協調フィルタリングが用いられた。協調フィルタリングの特徴は、データセット中の欠損値に対して特別な操作を行わずに、また特別な変数選択をおこなわずに、そのまま分析の入力とする点である。

この手法によるグルーピングの可能性を計る為に、プロジェクトデータの中から1つを取り出し、そのデータに対してたとえば最終的な工数など一部の値を隠したまま全体のデータセットから類似度の高いデータセットグループを取り出し、そのグループの隠した値に対応する値、たとえば総工数と隠した値を比較する試みが行われた。すなわち、過去の類似プロジェクトのデータに基づいた類推見積もりである。大杉らの研究では、一つのケーススタディではあるが、この手法の見積もり予測手法としての可能性を示している[5-9]-[5-12]。

先に述べたSEC先進プロジェクトでは、ベンチマークデータに関して基本設計終了時に開発グループごとに全体のプロジェクト計画値と基本設計終了時までの実績値を集め、プロジェクト終了時に残りの実績値を収集した。基本設計終了時のデータを用いて、過去のプロジェクトデータベースから協調フィルタリングにより類似プロジェクトグループを抽出し、そこからプロジェクト予測が試みられた。この研究からSECのベンチマークデータのように収集された多量のデータセットに対して、類似プロジェクトのグルーピング手段として協調フィルタリング技術が有効なこと、さらにこの技術が進行中の新規プロジェクトから得られる部分的なベンチマークデータをキーとして過去データから類似プロジェクト群を抽出し、プロジェクト予測に役立てることが出来ることが示された。

### 5.3.3 過去プロジェクトデータを活用したプロジェクト予測機構の提案

そこで、蓄積したプロセス計測データの活用法の一つとして、過去プロジェクトのベンチマークデータベースによる類似プロジェクトのグルーピング手法を用いることを提案した[5-4]-[5-7]。

図5-16にその概念を、以下に、その手順を示す。括弧( )に図5-16中の番号を示す。

1) まず、各プロジェクトに対し、5.3.1で述べたベンチマークデータと2章で述べ

たプロセスとプロダクトの計測データを対にして収集し(1)(2), データベースに蓄積する(3)(4).

2) 次に, 新規のプロジェクトに対して, プロジェクト途中までに収集可能なベンチマークデータを収集し(6), これをキーとして協調フィルタリングによりベンチマークデータベースから類似プロジェクトをグルーピングして抽出する(7).

3) 次に, この類似プロジェクト群をキーにして(8), プロセスとプロダクトの計測データベースからそれぞれに対応するプロセスとプロダクトの計測データを参照し, プロジェクト予測に反映する(9). これらと, ベンチマークデータからの予測(8), 進行中プロジェクトの計測データ(5)を, 進行中のプロジェクト運営に反映する.

ここで提案した方式は, プロジェクト運営に活用したプロセスとプロダクトの推移に関する計測データをそのまま蓄積し, あとから過去データ中の類似プロジェクトを検索する方法で活用することを提案するものである. 過去データのなかから類似プロジェクトを検索する方法として, 計測データと対にして蓄積したベンチマークデータと協調フィルタリング技術を用いる方式を提案した.

本提案の実現・実証の鍵は EPM のようなインプロセスの計測ツールとそれによる計測の普及である. そして, これとあわせてすでにスタートしているベンチマークデータの収集と紐づけるかたちでその計測データの蓄積が進むことが必要である. それまではここで主張する予測機構の実現は提案に留まらざるを得ない. そしてすでに筆者らは EPM ツール群の配布キットを実現し, その配布活動を開始している.

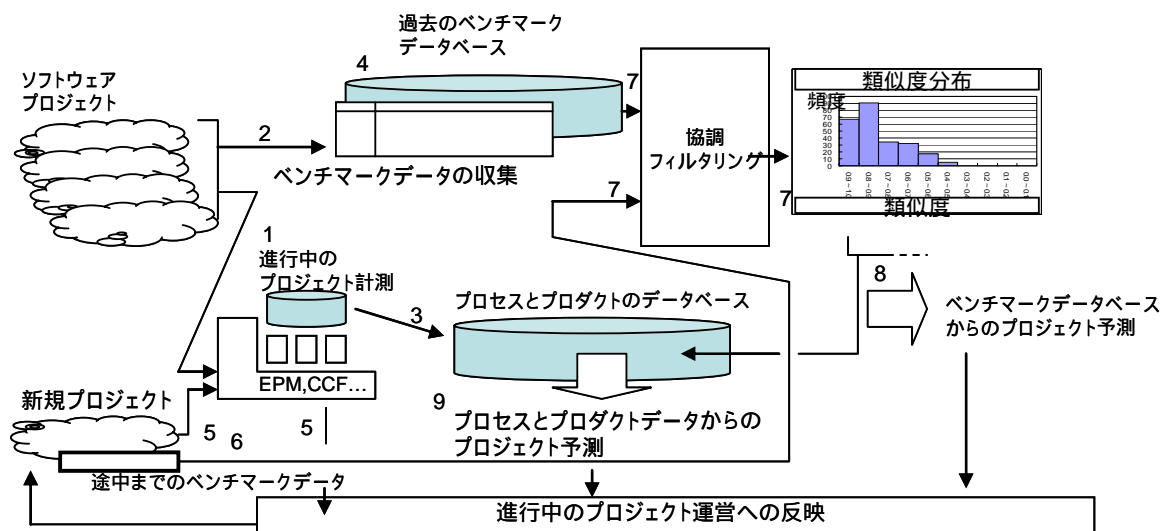


図5 - 16 過去プロジェクトデータを活用したプロジェクト予測機構の提案

## 5章 参考文献

- 5-1. 情報処理推進機構，ソフトウェア・エンジニアリング・センター：ソフトウェア開発データ白書 2005，日経 BP 社，2005-5，P.137，2005.
- 5-2. 情報処理推進機構，ソフトウェア・エンジニアリング・センター：ソフトウェア開発データ白書 2006，日経 BP 社，2006-6，P.213，2006.
- 5-3. Yoshiki Mitani, Tomoko Matsumura, Mike Barker, Seishiro Tsuruho, Katsuro Inoue, Ken-Ichi Matsumoto: 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, *Proceedings of International Symposium on Empirical Software Engineering and Metrics 2007 (ESEM 2007)*, Madrid, Spain, September 2007. (2007.3 採録，2007.9 発表予定) .
- 5-4. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, and Ken-ichi Matsumoto : A Proposal for Analysis and Prediction for Software Projects using In-Process Measurements and Collaborative Filtering of a Benchmarks Database. *Journal of Software Measurement*.(2007.7 投稿，2007.11 掲載予定) .
- 5-5. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, and Ken-ichi Matsumoto: A Proposal for Analysis and Prediction for Software Projects using Collaborative Filtering, In-Process Measurements' and a Benchmarking, *Proceedings of International Conference on Software Process and Product Measurement (MENSURA 2006)*, pp.98-107, Cadiz, Spain, November 2006.
- 5-6. 神谷芳樹，菊地奈穂美，松村知子，大杉直樹，門田暁人，肥後芳樹，井上克郎，松本健一：進行中のプロジェクト計測とフィードバック実験に基づく計測データベース活用方式の提案，ソフトウェアエンジニアリングシンポジウム2006（ソフトウェアエンジニアリング最前線2006，近代科学社），pp.35-42，2006.10.
- 5-7. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, and Ken-ichi Matsumoto: A Proposed Method for Building a Database of Project Measurements and Applying it Using Collaborative Filtering, *Proceedings of 5th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2006)*, Vol.2, Short papers, pp.15-17, Rio de Janeiro, Brazil, October 2006.
- 5-8. ISBSG, "The Benchmark Release 6." <http://www.isbsg.org.au>
- 5-9. Naoki Ohsugi, Akito.Monden, Shuji.Morisaki: Collaborative Filtering Approach for Software Function Discovery: *Proceedings of 1 st. ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2002)*, vol.2, Nara, Japan,



pp.45-46, 2002.

5-10. Naoki.Ohsugi, Masateru.Tsunoda, Akito.Monden, and Ken-ichi.Matsumoto: Effort Estimation Based on Collaborative Filtering, *Proceedings of 5th International Conference on Product Focused Software Process Improvement (PROFES 2004)*, pp.274-286, 2004.

5-11. 大杉直樹, 角田雅照, 門田暁人, 松村知子, 松本健一, 菊地奈穂美: 企業横断的収集データに基づくソフトウェア開発プロジェクトの工数見積り, *SEC journal* No.5 2006-2, pp.16-25, 2006.

5-12. 情報処理推進機構, ソフトウェア・エンジニアリング・センター: ソフトウェア開発見積もりハンドブック, EASE 協調フィルタリング法, 2006-4, pp.190-201, オーム社, 2006.

## 6. おわりに

### [ ソフトウェアプロジェクト計測実験のまとめ ]

ソフトウェア開発管理を支援するための実証的な研究はこれまでも試みられて来たが、その研究成果が産業界の共感を呼ぶことは少なく、提唱される手法やツールが普及せず、事例が積み上がる事がなかった。

課題の一つとして、プロジェクト計測とフィードバックの効果について、実際にソフトウェアを開発する人間の立場や振る舞い、開発組織や産業構造との対応が分析されておらず、誰にどんなフィードバックがどのように効果があったのか明らかでなく、手法普及のための共感が得にくい、という問題があった。

本研究では、設計工程以降を対象とした総合的なインプロセスのプロジェクト計測実験において、計測とフィードバックの効果を開発体制と個々の人間の役割に深く立ち入って観察し、誰にどんなフィードバックがどのように効果があったのか、産業界の共感を得られるようなエビデンスの獲得を試みた。

また従来の研究は、必ずしも事例の文脈情報が明らかでなく、また海外の軍隊や特別な大企業の内部計画など、産業界で共感を呼ぶような一般性がなかった。また、発表事例は少なく、一般性を主張できるほどの事例積上げがなく、日本の産業界の共感を呼ぶことが少なかった。

本研究では、一般性の高いソフトウェア・プロジェクトを計測実験の場を選び、文脈情報を明らかにしながらその効果の実証を狙った。事例として、2つの計測実験の事例を積むことになった。

さらに従来の研究は、データ収集と分析に最新技術を適用し総合的に評価する試みに欠け、またデータ収集の負荷軽減、自動収集によるデータ精度向上に焦点があてられていない問題があった。さらにデータ収集・分析ツールの普及など実証したことの再現のための工夫が図られていなかった。

本研究では、プロジェクト計測プラットフォーム EPM,をはじめ、最新の計測、分析技術を動員し、どのようなフィードバックが誰にどのように有用か観察した。はじめから自動収集による計測負荷の軽減、そして実験の再現性、手法とツールの普及を狙った。

最後に、従来の研究はその範囲が製造工程中心でソフトウェア開発の幅広い工程を対象としていない問題があった。

本研究では2つの計測実験によって研究の範囲を要求定義工程から製造、試験工程まで幅広いものとし、その成果から全工程一貫計測モデルを提案した。

本論文で述べた研究の成果が、従来の問題のすべてを一気に解決したわけではないが、計測実験に参加したステークホルダが実験終了後 SEC の企画した EPM ツールの検証プロジェクトに参加したり、プロジェクト計測に関して大学との新たな共同研究を行うなど、自己の資源を投入する形で次の段階へ活動を進めたこと、これらの報告を得た一般企業が、SEC の企画した手法とツール普及のためのプロジェクトに徐々に参加し始めていることな

ど、手法とツールの普及が始まり、あたらしい段階へ一歩を進めることが出来たことを評価できる。

これらの成果が得られた要因は、産学官連携により新しい研究の枠組みが実現できたことと、この環境において、ソフトウェア・プロジェクトの構造や実際に働く人間の役割に着目した新しい視点をもったことによる。

#### [ソフトウェア工学研究のための産学連携の仲介者の役割に関する考察]

おわりにあたって、本論文の研究を通して得られた経験をもとに、実践的なソフトウェア工学の研究環境を実現する産学連携の仲介者の役割に関する考察について述べる[6-1]。

本論文の研究をすすめることで、このような実践的なソフトウェア工学研究の枠組み実現における課題が明らかになった。本論文で述べた研究は、研究環境が実現できなければ実施できず、実践的なソフトウェア工学研究においてこのような研究環境の獲得・確保はその死命を制する重要課題である。本論文の研究ははじめに述べたように文部科学省の委託事業の一つである e-Society 基盤ソフトウェアの総合開発計画、データ収集に基づくソフトウェア開発支援システム（略称 EASE プロジェクト）および経済産業省の施策の一つとして独立行政法人情報処理推進機構（IPA）に新設されたソフトウェア・エンジニアリング・センター（SEC）の施策の一環として実施された。これらの施策は国の施策をトリガとして産学官連携の新しい研究の枠組みを実現し、当初の狙いである産学でのフィールドデータの共有環境を実現することができた。そしてこの場で、実践的ソフトウェア工学の考え方を反映したインプロセスのプロジェクト計測と分析・フィードバック実験を実現し、その有用性を一定の範囲で主張できたことになる。この試みで明らかになった研究環境構築に関する主な課題をまとめると次のようになる[6-1]-[6-3]。

#### 1) 産学連携の研究の枠組み構築における政府施策の重要性

ソフトウェア工学の領域での産学でのフィールドデータの共有は、それぞれの努力だけでは難しく政府施策のトリガが重要で有効である。

#### 2) 計測ツールへの考え方

フィールドデータを取得するためのインプロセスの計測ツールには、構成要素としてオープンソース・ツールを組み合わせた経済性や親しみやすさ、導入や運用に特別な負荷のかからない自動収集機能や全体構成、利便性の明確な分析・表示機能などが有効である。

この計測ツールは同時にフィールドデータを産学で共有する研究の枠組み実現のための重要なツールの役割も担っている。

#### 3) 周到な計測計画の立案と実行、関係者間での協議

計測計画実行にあたっては、特に関係者間での十分な協議と調整、共通認識の醸成が重要である。

#### 4) 全体のコーディネーション

産学官を連携させ円滑に計画を進め発展させるためには、多くの利害関係者のベクトルを揃える計画全体のコーディネーションが重要である。

そして、このようなソフトウェア工学研究環境実現のための産学連携に関して考えると、まず、産業界と学界のように距離のある組織間でフィールドデータの共有のような密な共同作業を実現するために、何らかの仲介者の役割が不可欠であることがわかる。本論文の研究では、EASE プロジェクトの中でエンピリカルラボという場に集められた産学二つの世界からの混成グループ、あるいは産学連携の場を作ることをミッションの一つとするSEC、そして全体を指導する政府組織（総合科学技術会議、文部科学省、経済産業省）がその役割を果たしている。

複雑化する現代社会でこのような仲介者の役割が高まることは以前から指摘されている。たとえば、米国労働長官を務めた経済学者のロバート B・ライシュは、かつて現代の新しい職種としてシンボリック・アナリストやストラテジック・ブローカという考え方を示し、新しい価値創出の潮流を分析した[6-4]。その分析は 1990 年代後半に加熱した各種の金融サービスや経営コンサルタント、リーガルビジネス、メディアや情報産業、そして企業経営者そのものの振る舞いに説得力をもってあてはまった。本論文の基礎にあるソフトウェア工学分野での産学連携の実現においても今回の試みを俯瞰すると、従来の社会の枠組みに無い新しい推進主体の役割が求められていることがわかる。

産学連携の仲介を経済原則のなかで考察すると、投資銀行家の神谷秀樹が提唱しているテクノロジー・アービトラージ（技術による鞘取り）という考え方から示唆を受ける[6-5]。産学連携の仲介とは、産業界と学界の間に存在する技術やデータの落差を捉えて自らの存在価値を作りだす一種の鞘取りサービスである。ソフトウェア開発のフィールドデータおよびデータ源は産業界に自らはその価値を感じないほど大量に死蔵といっても良い状況で存在するが学界では稀少である。一方ソフトウェア工学に関する技術的な潜在力は学界に学界自身が必ずしもその価値の高さを認識していないほど日常的に存在しているが、産業界からはアクセスしにくい。この両者の落差を埋めるところにアービトラージが存在し得る。

また産学二つの世界の連携実現を商取引になぞらえると、二つの取引方式が考えられる。資本市場にあるオークション方式とマーケットメイク方式である。

オークション方式は学界から産業界への技術移転や各種のコンサルティングサービス、学界に起源をもつソフトウェア製品事業、あるいは逆の流れで、産業界からは学界へのデータや一種のコンテンツの有償提供といったことには当てはまるが、本論で対象とするソフトウェア工学の分野にはあてはまらない。

ソフトウェア工学の分野での産学連携に必要な取引形態は、マーケットメイク方式である。学界、産業界ともそのままの状態では互いに相手方への期待は薄く、いうなれば双方に売り気配、買い気配が希薄な状態となっている。ここで仲介者が双方に相手に対する期

待を醸成し、一時的なコミットを引き受けてそれぞれの潜在力を引き出すことに成功すれば取引を成立させることができる。この取引が活発化し市場（取引所）が活況を呈するようになってはじめて産業力強化への貢献が期待できるようになる。

まとめると、エンピリカルソフトウェア工学の考え方を反映して、産業界と学界の間で比喩的にマーケットメイク方式の取引形態で、テクノロジー・アービトラージを行う仲介者が働いて、その取引市場を活発化させることが我が国のソフトウェア産業力強化の一つの鍵である、ということになる。EASE プロジェクトと SEC における産学連携活動は、これまでのプロセスの中で、新しい時代に求められるこのようなメカニズムの一端を実現しその有用性を実証しつつあると位置づけることができる。互いに相手に期待していない溝のある組織間で、双方に期待感を醸成するような企画を提示し、ツールを提供し実際にこの期待に応える双方に実利もたらされる活動を進めることによって本来の目的が達成されるということを実証しつつある。

産学連携では、学界側に蓄積された発明などの知的財産や技術をいかにして産業界に移転し事業化するか、あるいは、産業界が抱える問題をいかにして学界側の力を活用して解決するか、という視点からの取り組みが一般的であるが、ソフトウェア工学研究あるいはソフトウェア産業の産業力強化という領域ではこうした取り組みの前に、学界と産業界の溝を埋め、両方でソフトウェア開発のフィールドデータの共有を実現することが大きな課題となる。本論文では、この課題へ取り組むため実現した新しい産学官連携の研究の枠組みによる成果の具体例を示した。

## 6章 参考文献

- 6-1. 神谷芳樹, マイク・バーカー, 松本健一, 鳥居宏次, 井上克郎, 鶴保証城: 現場データを産学で共有するソフトウェア工学研究のための枠組み, *産学連携学*, Vol.2.No.2, pp.26-37, 2006.3.
- 6-2. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Satoshi Iwamura, Yoshiki Higo, Katsuro Inoue, Mike Barker, Ken-ichi Matsumoto: Effect of Software Industry Structure on a Research Framework for Empirical Software Engineering, *Proceedings of 28th International Conference on Software Engineering (ICSE 2006)*, Far East Experience Track, Poster Session, pp.616-619, Shanghai, China, May 2006.
- 6-3. Yoshiki Mitani, Mike Barker, Koji Torii, Seishiro Tsuruho: An Experimental Framework for Japanese Academic-Industry Collaboration in Empirical Software Engineering Research, *Proceeding of 3rd ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2004)*, Vol.2, Poster Session, pp.35-36, Redondo Beach, CA., USA, August 2004.

6-4. ロバート B・ライシュ, 中谷 巖訳: ザ・ワーク・オブ・ネーションズ, 21世紀資本主義のイメージ, ダイヤモンド社, P.445, 1991.

6-5. 神谷秀樹: ニューヨーク流たった5人の「大きな会社」, 亜紀書房, P.260, 2001.

## 謝辞

本研究を終始ご指導いただいた，  
奈良先端科学技術大学院大学 松本 健一 教授 に心より感謝いたします．  
本研究，特に論文についてこころのこもったご指導いただいた，  
奈良先端科学技術大学院大学 関 浩之 教授 に心より感謝いたします．  
本研究，また参加する EASE プロジェクトを通して終始ご指導いただいた，  
奈良先端科学技術大学院大学 飯田 元 教授に心より感謝いたします．  
本研究を終始ご指導いただいた，  
奈良先端科学技術大学院大学 門田 暁人 准教授 に心より感謝いたします．  
本研究を EASE プロジェクトを通して終始ご指導いただいた，  
大阪大学大学院 井上 克郎 教授， 楠本 真二 教授  
奈良先端科学技術大学院大学 マイク・バーカ 客員教授  
奈良先端科学技術大学院大学 小山 正樹 名誉教授 に心より感謝いたします．  
本研究を EASE プロジェクトを通して終始ご支援いただいた，  
(元)奈良先端科学技術大学院大学 松村 知子 特任助教  
(元)奈良先端科学技術大学院大学 大杉 直樹 特任助教  
奈良先端科学技術大学院大学 森崎 修司 特任助教  
奈良先端科学技術大学院大学 玉田 春昭 特任助教  
大阪大学大学院 肥後 芳樹 氏， 吉田 則裕 氏， 馬場慎太郎 氏  
に心より感謝いたします．  
本研究を EASE プロジェクト，EPM の開発を通して終始ご支援いただいた，  
日立システムアンドサービス 津田 道夫 氏， 十九川博幸 氏， 新海 平 氏  
日立公共システムエンジニアリング 小野 英治 氏， 泉原 健史 氏  
SRA 先端技術研究所 岸田 孝一 氏， 松村 好高 氏， 阪井 誠 氏  
NTT ソフトウェア 岩村 聡 氏 に心より感謝いたします．  
本研究をソフトウェアエンジニアリングセンターの活動から終始ご支援いただいた  
IPA/SEC 樋口 登 研究員， (元)IPA/SEC 松浦 清 所長補佐  
IPA/SEC 菊地奈穂美 研究員 およびSEC 研究員各位に心より感謝いたします．  
本研究をご支援いただいた，ソフトウェアエンジニアリング技術研究組合の方々，  
NTT データ 勝又 敏次 氏  
経済産業省 安田 篤 氏 に心より感謝いたします．  
最後に，すべてにわたって終始こころのこもったご指導を頂いた，  
奈良先端科学技術大学院大学 鳥居 宏次 特任教授  
IPA ソフトウェア・エンジニアリング・センター 鶴保 征城 所長  
に心より感謝いたします．

## 参考文献一覧

(章別再掲)

### 1章

- 1-1. 神谷芳樹：EASE プロジェクトにみる計測・定量化の実践，*日経ITプロフェッショナル*，2005年3月 pp.92-97，日経BP社，2005.3.
- 1-2. Capers Jones 著，鶴保証城，富野壽監訳：ソフトウェア開発の定量化手法 第2版，構造計画研究所，P.546, 1998.4.
- 1-3. Michael A. Cusumano 著，サイコム・インターナショナル監訳：ソフトウェア企業の競争戦略，ダイヤモンド社，P.445, 2004.12.
- 1-4. John McGarry et.al: Practical Software Measurement, Piason Education, 2002.  
(John McGarry ほか著，古山恒夫，富野壽監訳：実践的ソフトウェア測定，構造計画研究所，P.250，2004.)
- 1-5. 井上克郎，松本健一，鶴保証城，鳥居宏次：実証的ソフトウェア工学環境への取り組み，*情報処理*，45巻7号，pp.722-728，2004.7.
- 1-6. EASE プロジェクト <http://www.empirical.jp/>
- 1-7. e-Society 基盤ソフトウェアの総合開発 <http://cif.iis.u-tokyo.ac.jp/e-society/>
- 1-8. 嶋田 隆・祝谷和宏：ソフトウェアエンジニアリングセンター構想について，*情報処理* 45巻4号，pp.367-371，2004.4.
- 1-9. ソフトウェア・エンジニアリング・センター  
<http://www.ipa.go.jp/software/sec/index.php>
- 1-10. 藤本隆宏：能力構築競争，日本の自動車産業はなぜ強いのか：中央公論新社，P.406, 2003.
- 1-11. 戸田 巖，松永俊雄：電電公社のコンピュータ開発，*情報処理*，Vol.44, No.6, pp.631-639, 2003.6.
- 1-12. 伊東洋一，佐藤匡正，長野宏宣，神谷芳樹：総合ソフトウェア生産システムの実用化，*日本電信電話公社電気通信研究所，研究実用化報告* 33巻12号，pp.2879-2893，1984.12.  
(花田收悦編，ソフトウェアの計画と管理，第5章に全文再掲，日科技連，pp.318-326，1987)
- 1-13. Tadamasato Satoh, Hironobu Nagano, Yoshiki Mitani: An Integrated Software Development System for Data Communication Systems Utilizing Various Types of Computers, *Review of Electrical Communications Laboratories (Review of ECL)* Vol.33 No3 1985.
- 1-14. 下田博次：ソフトウェア工場，見えない工業製品の生産と労働，東洋経済新報社，P.232, 1986.
- 1-15. 釜范祐治，笠原和男，小田英雄，高橋宗雄：ソフトウェア開発管理方式，*NTT 電気*



通信研究所, 研究実用化報告 37 巻 12 号, pp.785-794, 1988.12.

- 1-16. Robert B. Grady: Practical Software Metrics for Project Management and Process Improvement, Prentice Hall PTR, P.270, 1992.
- 1-17. ローレンス・H・パトナム, ウエア・マイヤーズ, 山浦恒央訳: はじめて学ぶソフトウェアメトリクス, 日経 BP, P.348, 2005.  
(Lawrence H. Putnam, Ware Myer: Five Core Metrics, 2003).
- 1-18. Linda M. Laird, M.Carol Brennan: Software Measurement and Estimation: A Practical Approach, *IEEE CS*, A John Wiley & Sons, Inc., P.257, 2006.
- 1-19. 松村崇史, 島 和之, 松本健一, 鳥居宏次: オブジェクト指向分析・設計ドキュメント計測システムの作成, *信学技報*, SS96-6, pp.9-16, 1996.7.

## 2 章

- 2-1. 樋口 登: Project Report 先進ソフトウェア開発プロジェクト; *SEC journal* no.2, pp.56-57, 2005.4
- 2-2. 松浦 清, 神谷芳樹, 樋口 登: Project Report 先進ソフトウェア開発プロジェクト Part II; *SEC journal* no.5, pp.44-49, 2006.2,
- 2-3. Victor Basili: Empirically Driven SE Research: State of the Art and Required Maturity, *Proceedings of 28th International Conference on Software Engineering (ICSE 2006)*, Invited Talk, pp.32, Shanghai, China, May 2006.  
(<http://www.cs.umd.edu/~basili/presentations/2006%20ICSE%20with%20Elbaum.pdf>)
- 2-4. Victor R. Basili: Is There a Future for Empirical Software Engineering?, *Proceedings of 5th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2006)*, Keynote, pp.1, Rio de Janeiro, Brazil, October 2006.  
(<http://www.cs.umd.edu/~basili/presentations/ISESE%202006%20Keynote%20Final.pdf>)
- 2-5. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, and Ken-ichi Matsumoto: A Proposal for Analysis and Prediction for Software Projects using Collaborative Filtering, In-Process Measurements and a Benchmarking, *Proceedings of International Conference on Software Process and Product Measurement (MENSURA 2006)*, pp.98-107, Cadiz, Spain, November 2006.
- 2-6. 神谷芳樹, 菊地奈穂美, 松村知子, 大杉直樹, 門田暁人, 肥後芳樹, 井上克郎, 松本健一: 進行中のプロジェクト計測とフィードバック実験に基づく計測データベース活用方式の提案, *ソフトウェアエンジニアリングシンポジウム 2006 (ソフトウェアエンジニアリング最前線 2006)*, 近代科学社, pp.35-42, 2006.10.
- 2-7. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito

- Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, and Ken-ichi Matsumoto: A Proposed Method for Building a Database of Project Measurements and Applying it Using Collaborative Filtering, *Proceedings of 5th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2006)*, Vol.2, Short papers, pp.15-17, Rio de Janeiro, Brazil, October 2006.
- 2-8 . Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Satoshi Iwamura, Yoshiki Higo, Katsuro Inoue, Mike Barker, Ken-ichi Matsumoto: Effect of Software Industry Structure on a Research Framework for Empirical Software Engineering, *Proceedings of 28th International Conference on Software Engineering (ICSE 2006)*, Far East Experience Track, Poster Session, pp.616-619, Shanghai, China, May 2006.
- 2-9. 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, *Proceeding of 4th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2005)*, Vol.2, Late Breaking Papers, pp.5-8, Noosa Heads, Australia, November 2005.
- 2-10. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Satoshi Iwamura, Mike Barker, Ken-ichi Matsumoto: A Research Framework for Empirical Software Engineering Collaboration and Its Application in a Software Development Project, *Proceedings of International Workshop on Future Software Engineering 2005 (IWFST 2005)*: Shanghai, China, November 2005.
- 2-11. 情報処理推進機構「柔の力，剛の技」第3部：競争力；ソフトウェア開発プロセスの「見える化」：アスキー出版，pp.160-176，2006.5.
- 2-12. 神谷芳樹：EASE プロジェクトに見る計測・定量化の実践，*日経 IT プロフェッショナル*，2005年3月，pp.92-97，2005.3.
- 2-13. 大平政雄，横森励士，阪井 誠，松本健一，井上克郎，鳥居宏次：Empirical Project Monitor:プロセス改善を目的とした定量的開発データの自動収集・分析システムの試作，*電子情報通信学会技術報告SIGSS*, Vol.103, No708, SS2003-48, pp.13-18, Mar.2004.3
- 2-14. 阪井 誠，大平雅雄，横森励士，松本健一，井上克郎，鳥居宏次：EPM，導入の容易な開発データ自動収集・分析支援システム - お手軽にリアルタイムの生データ - ，*ソフトウェアシンポジウム2004*，2004.
- 2-15. 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 (MSR2004)*, pp.42-46, Edinbargh Scotland, UK, May 2004.
- 2-16. Masao Ohira, Reishi Yokomori, Makoto Sakai, Ken-ichi Matsumoto, Katsuro

- Inoue, Michael Barker, Koji Tori, Empirical Project Monitor: A System for Managing Software Development Projects in Real Time, *Proceeding of 3rd ACM-IEEE International Symposium on Empirical Software Engineering ( ISESE2004 )*, Vol.2, Poster Session, pp.37-38, Redondo Beach, CA, USA, August 2004.
- 2-17. 大平 雅雄, 横森 励士, 阪井 誠, 岩村 聡, 小野 英治, 新海 平, 横川 智教: ソフトウェア開発プロジェクトのリアルタイム管理を目的とした支援システム, *電子情報通信学会論文誌 D-I*, Vol.J88-D-I, No.2, pp228-239, 2005.
- 2-18. Toshihiro.Kamiya., Shinji.Kusumoto, Katsuro.Inoue: CCFinder: A Multi-Linguistic Token-based Code Clone Detection System for Large Scale Source Code. *IEEE Transactions on Software Engineering (TSE)* 28, pp.654-670, 2002.
- 2-19. 情報処理推進機構, ソフトウェア・エンジニアリング・センター: ITプロジェクトの「見える化」, 上流工程編: 日経BP社, P.208, 2007.5.
- 2-20. 情報処理推進機構, ソフトウェア・エンジニアリング・センター: ITプロジェクトの「見える化」, 下流工程編: 日経BP社, P.211, 2006.6.
- 2-21. 松村知子, 森崎修司, 玉田春昭, 大杉直樹, 門田暁人, 松本健一: GQM モデルに基づく設計工程完成度計測手法の提案, 奈良先端科学技術大学院大学, テクニカルレポート P.22, 2007-3.
- 2-22. 松村知子, 森崎修司, 玉田春昭, 大杉直樹, 門田暁人, 松本健一: プロセス改善を目的とする ODC を用いた欠陥修正工数分析, 奈良先端科学技術大学院大学, テクニカルレポート, P.11, 2007.3.
- 2-23. 松村知子, 森崎修司, 玉田春昭, 大杉直樹, 門田暁人, 松本健一: 設計書の再利用を考慮したレビュー効率の比較方法の提案と事例紹介, 奈良先端科学技術大学院大学, テクニカルレポート, P.16, 2007.3.
- 2-24. 松村知子, 門田暁人, 森崎修司, 松本健一: マルチベンダ情報システム開発における障害修正工数の要因分析, *情報処理学会論文誌*, Vol.48, No.5, pp.1926-1935, 2007.5.
- 2-25. 肥後芳樹, 吉田則裕, 楠本真二, 井上克郎: 産学連携に基づいたコードクローン可視化手法の改良と実装, *情報処理学会論文誌*, Vol.48, No.2, pp.811-822, 2007.2.
- 2-26. 玉田春昭, 松村知子, 森崎修司, 松本健一: プロジェクト遅延リスク検出を目的とするソフトウェア開発プロセス可視化ツール ProStar, 奈良先端科学技術大学院大学, テクニカルレポート, P.11, 2007.2.
- 2-27. 松村知子, 勝又敏次, 森崎修司, 玉田春昭, 大杉直樹, 門田暁人, 楠本真二, 松本健一: 自動データ収集・可視化ツールを用いたリアルタイムフィードバックシステムの構築と試行, 奈良先端科学技術大学院大学, テクニカルレポート, P.23, 2007.2.
- 2-28. 大杉直樹, 松村知子, 森崎修司: ソフトウェア開発の「見える化」を支援するデータ分析力, ~エンピリカルアプローチによる既存データの有効活用~: *JISA 会報* 80 2006年1月, pp.13-29, 2006.1.

### 3 章

- 3-1. 神谷芳樹, 菊地奈穂美, 松村知子, 大杉直樹, 門田暁人, 肥後芳樹, 井上克郎, 松本健一: 進行中のプロジェクト計測とフィードバック実験に基づく計測データベース活用方式の提案, *ソフトウェアエンジニアリングシンポジウム 2006* (ソフトウェアエンジニアリング最前線 2006, 近代科学社), pp.35-42, 2006.10.
- 3-2. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Satoshi Iwamura, Yoshiki Higo, Katsuro Inoue, Mike Barker, Ken-ichi Matsumoto: Effect of Software Industry Structure on a Research Framework for Empirical Software Engineering, *Proceedings of 28th International Conference on Software Engineering (ICSE 2006)*, Far East Experience Track, Poster Session, pp.616-619, Shanghai, China, May 2006.
- 3-3. Shuji Morisaki, Akito Monden, Tomoko Matsumura, Haruaki Tamada and Ken-ichi Matsumoto: Defect Data Analysis on Extended Association Rule Mining, *International Workshop on Mining Software Repository 2007 (MSR2007)*, 2007.5.
- 3-4. 森崎修司, 松村知子, 玉田春昭, 大杉直樹, 門田暁人, 松本健一: 相関ルール分析を用いた障害対応データの特徴分析, 奈良先端科学技術大学院大学, テクニカルレポート P.15, 2007.2.
- 3-5. 小池和男・洞口治夫 (編): 経営学のフィールド・リサーチ, 「現場の達人」の実践的調査手法, 日本経済新聞社, P.264, 2006.1.
- 3-6. Yoshiki Mitani, Tomoko Matsumura, Mike Barker, Seishiro Tsuruho, Katsuro Inoue, Ken-Ichi Matsumoto: 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, *Proceedings of International Symposium on Empirical Software Engineering and Metrics 2007 (ESEM 2007)*, Madrid, Spain, September 2007 (2007.3 採録, 2007.9 発表予定).
- 3-7. International Workshop on Mining Software Repository 2006. Collocated ICSE 2006, Shanghai, China May 2006.

### 4 章

- 4-1. Yoshiki Mitani, Tomoko Matsumura, Mike Barker, Seishiro Tsuruho, Katsuro Inoue, Ken-Ichi Matsumoto: An Empirical Study of Requirement Definition Process Management and Metrics based on an In-process Measurement Experiment of Standardized Requirement Definition Phase. *Proceedings of International Conference on Software Process and Product Measurement (IWSP-Mensura 2007)*, Palma de Mallorca, Spain, November 2007. (2007.8 採録, 2007.11 発表予定).
- 4-2. Yoshiki Mitani, Tomoko Matsumura, Mike Barker, Seishiro Tsuruho, Katsuro Inoue,

- Ken-Ichi Matsumoto: 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, *Proceedings of International Symposium on Empirical Software Engineering and Metrics 2007 (ESEM 2007)*, Madrid, Spain, September 2007. (2007.3 採録, 2007.9 発表予定).
- 4-3. John A. Zachman: A Framework for Information Systems Architecture. *IBM Systems Journal* vol.26, no.3. 1987.
- 4-4. 村上敬亮: Enterprise Architecture について, ~これからのITガバナンスと情報サービス産業~, 経済産業省, 2004.1.19.  
[http://www.meti.go.jp/policy/it\\_policy/ea/data/report/p1/index.html](http://www.meti.go.jp/policy/it_policy/ea/data/report/p1/index.html)
- 4-5. 経済産業省, 商務情報政策局, 情報政策課: EA ポータル,  
[http://www.meti.go.jp/policy/it\\_policy/ea/index.html](http://www.meti.go.jp/policy/it_policy/ea/index.html)
- 4-6. 経済産業省, 商務情報政策局, 情報政策課: 業務・システム最適化計画について(Ver.1.1)  
~ Enterprise Architecture策定ガイドライン ~  
[http://www.meti.go.jp/policy/it\\_policy/ea/data/report/r2/index.html](http://www.meti.go.jp/policy/it_policy/ea/data/report/r2/index.html)
- 4-7. 村上敬亮: EA策定に必須の「参照モデル」標準化や調達効率化に寄与, EA大全 Part5, 参照モデル編, pp.184-185, 日経BP社, 2004-5
- 4-8. 村上敬亮: Enterprise Architecture について, ~EAの策定方法~, 経済産業省, 2004-1-19.  
[http://www.meti.go.jp/policy/it\\_policy/ea/data/report/p2/index.html](http://www.meti.go.jp/policy/it_policy/ea/data/report/p2/index.html)
- 4-9. 各省庁情報化統括責任者(CIO)補佐官等連絡会議, ワーキンググループ3最終報告, 首相官邸ホームページ(IT戦略本部), 2007.4.19.  
<http://www.kantei.go.jp/jp/singi/it2/cio/hosakan/dai36/36w3houkoku.pdf>

## 5章

- 5-1. 情報処理推進機構, ソフトウェア・エンジニアリング・センター: ソフトウェア開発データ白書 2005, 日経BP社, 2005-5, P.137, 2005.
- 5-2. 情報処理推進機構, ソフトウェア・エンジニアリング・センター: ソフトウェア開発データ白書 2006, 日経BP社, 2006-6, P.213, 2006.
- 5-3. Yoshiki Mitani, Tomoko Matsumura, Mike Barker, Seishiro Tsuruho, Katsuro Inoue, Ken-Ichi Matsumoto: 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, *Proceedings of International Symposium on Empirical Software Engineering and Metrics 2007 (ESEM 2007)*, Madrid, Spain, September 2007. (2007.3 採録, 2007.9 発表予定).
- 5-4. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito

- Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, and Ken-ichi Matsumoto : A Proposal for Analysis and Prediction for Software Projects using In-Process Measurements and Collaborative Filtering of a Benchmarks Database. *Journal of Software Measurement*. (2007.7 投稿 , 2007.11 掲載予定) .
- 5-5. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, and Ken-ichi Matsumoto: A Proposal for Analysis and Prediction for Software Projects using Collaborative Filtering, In-Process Measurements' and a Benchmarking, *Proceedings of International Conference on Software Process and Product Measurement (MENSURA 2006)*, pp.98-107, Cadiz, Spain, November 2006.
- 5-6. 神谷芳樹, 菊地奈穂美, 松村知子, 大杉直樹, 門田暁人, 肥後芳樹, 井上克郎, 松本健一: 進行中のプロジェクト計測とフィードバック実験に基づく計測データベース活用方式の提案, *ソフトウェアエンジニアリングシンポジウム 2006 (ソフトウェアエンジニアリング最前線 2006)*, pp.35-42, 2006.10.
- 5-7. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Naoki Ohsugi, Akito Monden, Yoshiki Higo, Katsuro Inoue, Mike Barker, and Ken-ichi Matsumoto: A Proposed Method for Building a Database of Project Measurements and Applying it Using Collaborative Filtering, *Proceedings of 5th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2006)*, Vol.2, Short papers, pp.15-17, Rio de Janeiro, Brazil, October 2006.
- 5-8. ISBSG, "The Benchmark Release 6." <http://www.isbsg.org.au>
- 5-9. Naoki Ohsugi, Akito.Monden, Shuji.Morisaki: Collaborative Filtering Approach for Software Function Discovery: *Proceedings of 1 st. ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2002)*, vol.2, Nara, Japan, pp.45-46, 2002.
- 5-10. Naoki.Ohsugi, Masateru.Tsunoda, Akito.Monden, and Ken-ichi.Matsumoto: Effort Estimation Based on Collaborative Filtering, *Proceedings of 5th International Conference on Product Focused Software Process Improvement (PROFES 2004)*, pp.274-286, 2004.
- 5-11. 大杉直樹, 角田雅照, 門田暁人, 松村知子, 松本健一, 菊地奈穂美: 企業横断的収集データに基づくソフトウェア開発プロジェクトの工数見積り, *SEC journal* No.5 2006-2, pp.16-25, 2006.
- 5-12. 情報処理推進機構, ソフトウェア・エンジニアリング・センター: ソフトウェア開発見積もりハンドブック, *EASE 協調フィルタリング法*, 2006-4, pp.190-201, オーム社, 2006.

## 6 章

- 6-1. 神谷芳樹, マイク・バーカー, 松本健一, 鳥居宏次, 井上克郎, 鶴保証城: 現場データを産学で共有するソフトウェア工学研究のための枠組み, *産学連携学*, Vol.2.No.2, pp.26-37, 2006.3.
- 6-2. Yoshiki Mitani, Nahomi Kikuchi, Tomoko Matsumura, Satoshi Iwamura, Yoshiki Higo, Katsuro Inoue, Mike Barker, Ken-ichi Matsumoto: Effect of Software Industry Structure on a Research Framework for Empirical Software Engineering, *Proceedings of 28th International Conference on Software Engineering (ICSE 2006)*, Far East Experience Track, Poster Session, pp.616-619, Shanghai, China, May 2006.
- 6-3. Yoshiki Mitani, Mike Barker, Koji Torii, Seishiro Tsuruho: An Experimental Framework for Japanese Academic-Industry Collaboration in Empirical Software Engineering Research, *Proceeding of 3rd ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2004)*, Vol.2, Poster Session, pp.35-36, Redondo Beach, CA., USA, August 2004.
- 6-4. ロバート B・ライシュ, 中谷 巖訳: ザ・ワーク・オブ・ネーションズ, 21世紀資本主義のイメージ, ダイヤモンド社, P.445, 1991.
- 6-5. 神谷秀樹: ニューヨーク流たった5人の「大きな会社」, 亜紀書房, P.260, 2001.