

問題の早期発見・改善を支援するインプロセスプロジェクト管理手法 の実プロジェクトへの適用

松村 知子[†] 森崎 修司[†] 勝又 敏次^{††} 玉田 春昭^{†††}
吉田 則裕^{†††} 楠本 真二^{†††} 松本 健一[†]

Applying In-Process Project Management Method to Real Software Development
Project for Early Detection and Handling of Problems

Tomoko MATSUMURA[†], Shuji MORISAKI[†], Toshitsugu KATSUMATA^{††},
Haruaki TAMADA^{†††}, Norihiro YOSHIDA^{†††}, Shinji KUSUMOTO^{†††},
and Ken-ichi MATSUMOTO[†]

あらまし 本論文では、複雑化するソフトウェア開発プロジェクトの組織・体制に適したプロジェクト管理を支援する手法、EASE インプロセスメソッドを実プロジェクトに適用し、その有用性を評価した結果について述べる。本手法は、CASE ツールからの自動データ収集・分析・可視化ツールを包含し、複数組織による複雑な体制のプロジェクトでも組織間の機密を保持しながらリアルタイムでの詳細なデータ収集・分析が可能で、プロジェクトを統括するマネージャが対策を必要とする問題を早期発見・対処することを支援する。我々は、マルチベンダ開発プロジェクトのプロジェクト管理に本手法を適用し、プロジェクトで発生する変更管理や要員管理などに関する問題の発見に有用であることを確認した。

キーワード プロジェクト管理, 定量的データ, ソフトウェアメトリックス, 構成管理, 障害管理

1. ま え が き

近年のソフトウェアの高度化、大規模化によって、ソフトウェア開発プロジェクト管理やソフトウェア開発プロセス改善の重要性が指摘され、PMBOK [1] や CMM [2], [3]/CMMI [4], [5], ISO9000 など、プロジェクト管理やプロセス改善、品質管理の枠組みが注目されている。また、多くの企業がこれらの枠組みを導入しつつある。これらの枠組みはいずれも、プロセスやプロダクトの計測による現状把握・改善を要求し

ている。しかし、ソフトウェア開発は、アプリケーションドメインや開発環境等多くの属性の影響を受けるため多種多様なものであり、特徴に応じた管理が要求されるが、これらの枠組みでは具体的な計測方法やそれによる開発管理手法は示されていない。

ソフトウェア工学におけるエンピリカルアプローチが、プロセスやプロダクトの計測・定量化と評価、フィードバックによるプロジェクト管理やプロセス改善を具現する有効な手法として、注目されている。産業界でも、エンピリカルアプローチを用いた開発プロジェクトの「見える化」が推進されている [6]。Rombachらは、エンピリカルアプローチによって発見されたソフトウェア工学上の法則や理論をまとめ、実用的なプロジェクト管理手法にフィードバックしていく具体的な手順を示している [7]。

しかし、最近の複数組織に分散した開発体制や外部組織への開発委託の増加により、エンピリカルアプローチに基づくプロジェクト管理が困難になり、ソフトウェア開発プロセスや品質に関する情報の透明性が

[†] 奈良先端科学技術大学院大学情報科学研究科, 生駒市
Graduate School of Information Science, Nara Institute of
Science and Technology, Ikoma-shi, 630-0192 Japan

^{††} (株) NTT データ・アイ, 東京都
NTT DATA i Corporation, Chiyoda-ku, Tokyo, 100-0005
Japan

^{†††} 京都産業大学コンピュータ理工学部, 京都市
Faculty of Computer Science and Engineering, Kyoto
Sangyo University, Kita-ku, Kyoto-shi, 603-8555 Japan

^{††††} 大阪大学大学院情報科学研究科, 豊中市
Graduate School of Information Science and Technology,
Osaka University, Toyonaka-shi, 560-8531 Japan

低下している。組織間の機密保持要求や物理的な距離によって、成果物や開発現場に直接アクセスすることができない場合が多く、データの計測や分析が困難という理由による。このような状況下では、開発中に発生する問題発見が遅れ、プロジェクトの遅延や大幅なコスト超過を招く危険性が高くなっている。対策として、詳細な進捗報告や品質評価を要求すると、開発者への負担が増加するとともに、管理者自身の作業も増大する。また、関連組織内部の開発プロセスや進捗管理方法の変更も必要となり、その結果、開発コスト・管理コストの増大などの問題が発生する。

本研究は、エンピリカルアプローチによる進行中のプロジェクト管理支援手法、EASE インプロセスメソッドの構築を目的とする。本手法は、EASE (Empirical Approach to Software Engineering)^(注1)プロジェクトにより開発されたデータ収集・分析ツールを用い、具体的なプロジェクト管理支援プロセスモデルを示す。これによって、進行中のプロジェクトから定量的なデータを収集・分析し、開発作業を透明化することによってプロジェクトで発生した問題の早期発見・解決を支援する。本論文では、まず同手法について説明し、マルチベンダによる実プロジェクトに適用し、有効性を検証した結果について報告する。

EASE インプロセスメソッドは、具体的に以下の四つの特徴をもつ。

- (A) 実プロジェクトへの適用のための具体的なプロジェクト管理支援プロセスモデル
- (B) プロジェクトで一般的に用いられる CASE ツールからの自動データ収集・分析・可視化ツールの導入
- (C) 計測目的を明確にしたデータ分析モデルに基づいたメトリックス群
- (D) データ種類や目的に応じた異なるフィードバック方法の活用

本手法は、データ計測によるプロジェクト管理の基盤となる技術として、Basili らの提唱するプロセス改善のための QIP (Quality Improvement Paradigm) フレームワーク [8] を用いるが、QIP フレームワークでは「計測プロセスの計画」に関して「プロジェクトの目的を達成するための適切な実施モデル、手法、ツールなどを選択する」とのみ述べ、具体的な構築方法を示していない。本手法は、まず (A) で、開発現場で「インプロセス」での実運用を想定したプロセスモデルを示し、運用時の定義項目を明らかにする。

(B) は、インプロセスでの運用を容易にするための自動データ収集・分析ツール EPM (Empirical Project Monitor) [9] の導入や EPM-ProStar [11] の開発を示す。これらのツールによって収集されるデータをベースに、対象プロジェクトの特性を考慮した目的指向のデータ分析モデルを GQM (Goal-Question-Metric) パラダイム [10] に従って作成し、(C) にあたるメトリックス群を抽出する。更に、(D) では (C) のメトリックス群を 2 種類に分け、効果的なフィードバック方法を構築する。このように、プロジェクト全体の開発作業の透明化を実現するため、既存技術を組み合わせることで実運用が可能なプロセスモデルを定義し、実プロジェクトに適用・評価を行った点に、本研究の新規性がある。

以降、2. では EASE インプロセスメソッドのアプローチと具体的な手法について述べ、3. では適用対象としたプロジェクトと本手法の適用手順について説明する。4. では本手法の適用により発見された問題について報告し、5. では開発者へのインタビューや適用コストから本手法の評価を行う。最後に 6. でまとめと今後の課題について述べる。

2. EASE インプロセスメソッド

2.1 EASE インプロセスメソッドの概要

図 1 は、EASE インプロセスメソッドのプロジェクト管理における位置付けと、その概要を示している。本手法は、点線で示されている従来のプロジェクト管理での管理者・開発者の作業を変更するものではなく、図中の実線で示されている (1) ~ (3) のプロセスを追加

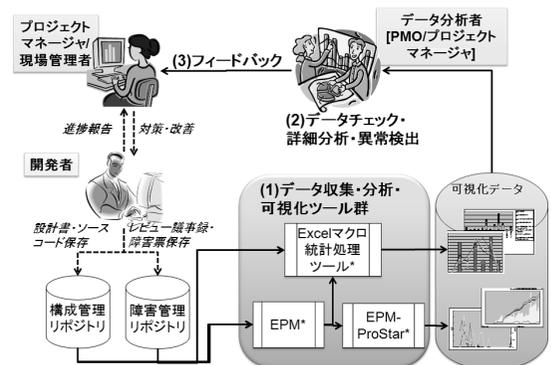


図 1 EASE インプロセスメソッドの概要 (*付きのツールの詳細は、2.3 参照)

Fig.1 Overview of EASE in-process method.

(注1): EASE: Empirical Approach to Software Engineering <http://www.empirical.jp/top.html>

することにより、より詳細で速報性の高い情報をプロジェクトマネージャや開発者に提供可能にし、問題の早期発見・改善を支援する。

図中の(1)~(3)は、1.の(B)~(D)の特徴に対応し、(A)のプロセスモデルは、(1)~(3)の各プロセスについて定義すべき項目を示す。(1)では、開発者の作業履歴が蓄積された構成管理、障害管理などのCASEツールのリポジトリから、自動的にツール群がデータ収集・分析・可視化を行う。この際に、(2)のデータ分析者に対して開示できないデータを排除し、分析者にとって有用なデータのみを適切な形式に加工・可視化する。(2)では、対象プロジェクトのプロジェクトマネージャやPMOなどの分析者が、(1)の出力結果をチェックし、更に詳細な検証したり、異常検出や対策を検討する。その結果が、プロジェクトマネージャやリーダなどの現場管理者にフィードバックされ、(3)で現場での対策・改善が行われる。

2.2 プロジェクト管理支援プロセスモデル

プロジェクト管理支援プロセスモデルは、データ計測に基づくプロジェクトの管理を容易にするために、プロジェクト計画時に定義すべき項目を明確にする。本モデルによって、計測手順や必要な要員・ツール等を明確にし、プロジェクトへの導入を容易にする。

定義項目は、「タイミング」、「関係者」、「対象」、「方法」の4項目で構成される。「タイミング」は、開発プロセスの中でそれぞれの作業が行われる時期・間隔である。「関係者」は、その作業にかかわる人材を示し、データの入力者・抽出者・受領者・中継者などが考えられる。「対象」は、具体的なデータの種類やデータの出力形式、作業の結果、プロセス改善などのアクションを含む。「方法」は、各作業を行うために必要な手段や手順で、ツールやアプリケーション、環境などを示す。

表1に図1の各プロセスに対する定義項目のパラメータ例を示す。

2.3 データ収集・分析・可視化ツール群

本節では、図1の(1)に対応するデータ収集・分析・可視化を行うツール群について説明する。EASEインプロセスメソッドでは、設計書やプログラムコード、レビューによる指摘データ、障害票データを分析対象とする。

EASEプロジェクトでは、特に下流工程(コーディング工程以降)のデータ収集を支援するEPMを開発した。本ツールは、開発者が用いるCASEツール(構成管

表1 プロジェクト管理支援プロセスの定義項目とパラメータ例

Table 1 Definitions and their samples for project management support process.

プロセス	定義項目	パラメータ例
(1) データ収集・分析・可視化	タイミング	日次, 週次, 工程区切り, 開発(改造)前, 開発完了時
	関係者	PM, PL, 開発者, ツール管理者
	対象	構成管理情報, 障害情報, 進捗報告, 現場状況
	方法	ツール, レポート, 口頭報告
(2) データチェック・詳細分析・異常検出	タイミング	(1)と同じ
	関係者	品質保証担当者, PMO, PM, SEPG
	対象	メトリックス, 可視化情報(図表)
	方法	指標値/予定値との比較, 機能間比較, インタビュー
(3) フィードバック	タイミング	(1)と同じ
	関係者	経営者, PM, PL, 開発者
	対象	考察/判定, 対策案
	方法	レポート, メール, 口頭(会議など)

理ツール: CVS など, 障害管理ツール: GNATS^(注2) など, メールリスト管理ツール: mailman など) から直接プロジェクト管理に有用な情報を抽出する機能をもつ^(注3)。

以下に、本論文の適用事例で用いる主な抽出データを示す。

- 構成管理: ファイル名, 変更者, 変更日時, 変更量 [追加・削除], 総行数など
- 障害管理: 発見日, 完了日, 発生原因, 混入要因, 重要度, 発見工程, 混入工程, 修正工数など

EASEインプロセスメソッドでは、EPMのデータ収集機能を用いて、CASEツールのリポジトリからの自動データ収集を行い、プログラムコードや障害情報に直接アクセスすることなく、必要なときに最新のデータを抽出する。

また、我々は、プロジェクト管理やプロセス分析研究などで用いられているメトリックスを調査し、独自の視点からのデータも含めて、EPMで収集できるデータから計測できるメトリックス群を抽出した[13]。抽出されたメトリックスは、八つの組織・プロジェクトに適用し、計測目的との適合性、開発者に理解容易な可視化方法、データ収集上の留意点、データ収集・

(注2): GNATS (GNU Bug Tracking System)

<http://www.gnu.org/software/gnats/>

(注3): 詳細は、「EPM 標準エンピリカルデータ」を参照。

<http://empirical.jp/EPM/Release/help/index-j.htm>

分析・フィードバックのシステムについて評価し、改善を行った [14] ~ [17]。本手法で用いるメトリックスは、これらのメトリックスの一部であり、詳細は 3.5 で述べる。

次に、上記のメトリックス群の分析と可視化を支援するツール EPM-ProStar [11] を開発した。EPM-ProStar は、EPM の収集データから計測されるメトリックスを、時系列で可視化することによって、プログラム行数の大幅な変動や障害対応の長期停滞、開発要員の交代など、開発遅延や低品質の原因や、その結果発生する現象の検出を容易にする。EPM-ProStar は、プロジェクトの管理方法の相違（日次・週次、コンポーネント単位・チーム単位など）による分析のカスタマイズ機能をもつ。また、累計値の時系列表示だけではなく、週次での集計値、モジュール別の分析値、複数モジュールの同一グラフ表示、更に時間的にずれたデータの重ね合わせ表示なども可能である。このような可視化によって、プロジェクト管理者はプロジェクト全体を俯瞰することも、モジュール単位で詳細にチェックすることもできる。更に、Microsoft Excel/Access の VBA で開発され、Windows 上で操作できるため、出力データのレポートなどへの流用が容易になる。

図 2 に EASE インプロセスメソッドにおけるデータ収集・分析ツールのデータ入出力関係を示す。EPM-ProStar が分析・可視化するメトリックスの一部は、3.5 の図 3 ~ 図 5 に示す。その他のメトリックスや可視化方法については、[11], [13] を参照されたい。

そのほか、設計書など Microsoft Office 等で作成される文書は、構成管理システムを用いてバージョン管理され、バージョンごとの規模や差分を抽出し計測するツールを用いて計測を行う [12]。レビューによって指摘された欠陥は、Excel のマクロ付きファイルに記

入され、自動的にデータの抽出・集計を行うツールを用いる。

EPM, EPM-ProStar は、CASE ツールのデータリポジトリから直接データ収集するため、開発者・管理者にとって図 1 の (1) の段階での追加作業はほとんど発生しない。各ツールは、簡易なユーザインタフェースを備えており、専門知識のない作業でも容易に操作することが可能である。

2.4 データの解釈と詳細分析

図 1 の (2) は、2.3 のツールからの出力である可視化データに基づいて行われる。分析者は、通常プロジェクトマネジメントオフィス（図中、PMO とする）やプロジェクトマネージャを想定しているので、プロジェクト管理に関する一般的知識と、対象プロジェクトの管理に必要なシステム構成、体制、進捗状況などの情報をもっている。データ解釈は、これらのプロジェクトの背景となる情報と分析者の過去の経験に基づく判断によって行われる。時系列メトリックスを用いた分析では、その変動の増減傾向、増減量、変動間隔などから、異常を検出する [13]。大量のデータ集合の分析では、特に顕著な数値（極大、極小）、あるいは複数の数値間の不整合に着目する。例えば、障害発生数が多いのに修正工数が少ない、などである。それらのデータの示す状況と、工程や作業の進捗状況、システムの特徴（再利用率、複雑度など）から、異常を検出する。

2.5 フィードバック

図 1 の (3) は、(2) の作業結果のフィードバックプロセスである。本手法は、プロジェクト進行中のプロジェクト管理支援に焦点を当て、事後分析ではなく、適切な情報を適切なタイミングで提供するための実施サイクルを定義する。メトリックスは、2 種類に分け、一つは時系列でメトリックスの推移・変化量を見て異常を検出する「リアルタイム型」、他方は一定量のデータ集合を分析することにより傾向を把握するもので「データ集約型」と呼ぶ。それぞれについて、フィードバック方法とタイミングを定義する（具体的なデータについては、3.5 参照）。

リアルタイム型データ分析は、日次や週次などの一定時間ごとに行われ、異常を検出した場合、現場管理者や開発者に電子メールや定期会議などで確認をとることで、プロジェクト内で発生している問題を早期に発見し、解決のための対策や改善を行うことが可能になる。開発者からの進捗報告からは読み取れない詳細

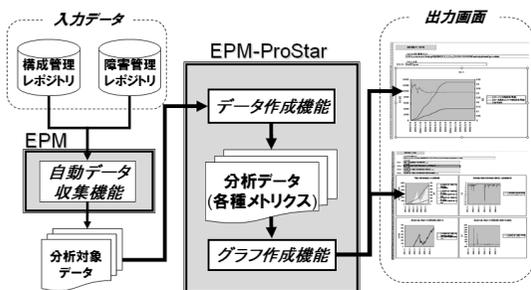


図 2 EASE インプロセスメソッドにおけるデータ収集・分析ツールの入出力

Fig. 2 Input and output for EASE in-process method tools.

な現場の状況を、リアルタイムに読み取ることができるといのが特徴である。データ集約型データ分析は、工程区切りなどの作業の区切りで行われ、会議形式でフィードバックを行う。タイミングはリアルタイム型より遅れるが、その場で問題の指摘や報告・確認が行われ、次工程への留意事項の確認、前工程の見直しや再レビューなどによって、プロジェクトでの問題の潜在化や拡大を防止することが可能になる。

3. ケーススタディ

3.1 対象プロジェクトの概要

EASE インプロセスメソッドの適用事例対象プロジェクト「プローブ情報システム開発プロジェクト」は、今日の日本における情報システム開発の典型ともいえる“マルチベンダ開発”である。本プロジェクトは、3年にわたって、基本構想の構築（要件定義）、新規開発（第1フェーズ）、処理の共通化と機能追加（第2フェーズ）が行われた[18], [19]。

このプロジェクトはユーザの立場の会社とプロジェクト管理担当会社、開発担当5社からなり、ユーザの立場の会社が要求を出し、開発担当5社が分担してサブシステムの開発を行い、プロジェクト管理担当会社が5社を統括した全体のプロジェクト管理を行う。5社はそれぞれ得意とする開発分野が異なり、それぞれの分野に応じた独自の開発プロセスをもつ。更に、各社内での開発体制も大きく異なり、外部企業・系列会社への委託、社内開発と様々な体制が見られた。このプロジェクトでは、社間のみならずプロジェクト管理担当会社に対してもプロダクト（プログラム等）や社内情報（工数、外部委託先、要員等）に関する機密保持が求められ、詳細設計以降、社内結合試験までの成果物、障害票、社内管理情報なども、関連各社間で開示されない。基本設計書と社間結合試験以降のテスト仕様書、障害票などは共有されるが、それ以外の工程では、各社から週次で示される進捗報告、工程区切りで報告される品質評価情報以外は、関連各社間で共有されることはない。

3.2 EASE インプロセスメソッドの導入ステップ

本手法は、対象プロジェクトの第2フェーズ開発に適用された。適用は、以下のような手順で実施された。

- (1) 目的の設定
- (2) 目的に応じたメトリックスの選択とフィードバック方法の検討
- (3) 計画の実施

(4) 開発関係者からの評価とその分析

- (1),(2),(3)については、次節以降で説明する。
- (4)については、4.で述べる。

本手法の適用にあたって、EASE プロジェクトは、対象プロジェクトの開発担当企業に対して EASE インプロセスメソッドについて説明し、一部企業に対しては EPM の提供・導入支援を行った。

3.3 プロジェクト管理支援の計測目的

プロジェクト管理支援の計測目的は、対象プロジェクトの特性を考慮して、5社を統括管理するプロジェクトマネージャ（以降、エグゼクティブ PM とする）と EASE 研究員で以下の四つを設定した。

- 要員に起因するリスクを管理する：エグゼクティブ PM に対して、各社内の要員体制は機密事項である。しかし、第1フェーズの開発中も、要員交代や不適切な作業配分は、社間にまたがる大きな問題を引き起こしている。機密を保持しつつも要員に関する情報を得られれば、エグゼクティブ PM にとって有効なリスク管理を行うことが可能になる。

- 組織内外のコミュニケーションを円滑に行う：各社間のコミュニケーションは、エグゼクティブ PM を交えた場で行われるが、各社担当者によって受け取り方が異なったり、漏れが発生したりする可能性が高い。また、社内の開発体制（複数組織）が複雑で、社間での取決めが直接担当者に伝達されない場合があるため、コミュニケーションミスによる問題が発生しやすい。

- 変更（要求・設計・プログラム）の安全性を確保する：マルチベンダによる共同開発であり、第2フェーズの開発では、一部機能の共通化など社間の調整が必要な状況が多数発生すると考えられる。機能の変更や追加に対する変更管理が適正に行われていないと、プロジェクト全体に影響する問題が発生する可能性がある。

- 既存コードの効率的・効果的活用を行う：第2フェーズの開発は、第1フェーズの開発に対する改造・追加であることから、既存コードへの改造や再利用が主である。各社内でコードの再利用のプロセスが適正に行われているか、再利用に際しての問題が発生していないかをチェックする必要がある。

3.4 適用プロセス

本節では、本手法のプロセスモデルの各定義項目について本事例での具体的内容を表2に示し、詳細について説明する。

表 2 COSE におけるプロジェクト管理支援プロセスの具体例 (“WebDAV” については (注 4) 参照)

Table 2 Example of Project Management support process on COSE project.

プロセス	定義項目	リアルタイム型	データ集約型
(1) データ 収集・分析・ 可視化	タイミング	週次	工程区切り
	関係者	ツール運用者 (各社内)	ツール運用者 (各社内)
	内容	構成管理情報, 障害情報	障害情報
	方法	EPM, EPM-ProStar, WebDAV	EPM, 市販ツール (Excel/Access/SPSS 等), WebDAV
(2) データ チェック・ 詳細分析・ 異常検出	タイミング	週次	工程区切り
	関係者	EASE スタッフ	EASE スタッフ
	内容	表 1 の*付きメトリックス	表 1 の*なしメトリックス
	方法	時間推移の変化量を目視確認, 複数メ トリックスによる事象確認	前年度の結果との比較, 機能間の比較,
(3) フィード バック	タイミング	分析時リアルタイム (週次)	品質評価会議 (工程区切り)
	関係者	エグゼクティブ PM, 各社開発関係者	エグゼクティブ PM, 各社 PM/PL, 品質保証担当者
	内容	図表, 質問/考察	図表, 質問/考察
	方法	メール	会議形式

表 2 の (1) では, 各社からのプロジェクトデータは, 毎週 WebDAV^{注4)}経由で提出され, 機密室内のスタンドアローンマシンに保存され, そのマシン上で EPM・EPM-ProStar 等を用いた分析が行われた。(2) では, 出力された図表とプロジェクトの進捗状況から, 問題の発生が推測されるポイントが抽出され, (3) で図表とともに各社に状況を問い合わせ, それに対して, 各社担当者からの回答を得た。具体的な抽出例は, 3.5 で述べる。これらの問い合わせ・回答は, エグゼクティブ PM にも伝えられ, 必要な場合はエグゼクティブ PM や各社 PM が何らかの対策を行う。

各プロセスのタイミングは, リアルタイム型メトリックスについては, 水曜日の対象プロジェクトの定期会議に合わせて, 金曜日までのデータを月曜日朝までに提出, 分析結果を火曜日にエグゼクティブ PM と各社開発者にフィードバックし, 問い合わせや回答を電子メールで行った。データ集約型については, 設計, 社内試験, 社間結合試験などの工程終了のタイミングで, 社別の会議形式でフィードバックし, エグゼクティブ PM と各社開発者及びデータ分析者が分析データをもとに質疑応答や議論を行った。

また, 電子メールでのデータ送付については, パスワードつきの暗号化ファイルで行い, 送付ミスや機密保持に配慮した。

3.5 計測メトリックス

設定した目的に対して, 我々は有効と思われるメトリックスをプロジェクトマネージャとともに選択した。表 3 に 3.3 の各目的に対するメトリックスとそれを用いる背景を説明する。(*) 付きは, リアルタイム型メトリックスを示し, その他はデータ集約型メトリックスである。なお, 表中で「欠陥」とは, 上流工程の

レビューで検出される指摘とテスト工程で発見される障害を含む。

以降, 一部のメトリックスの可視化例とその解釈, 検出された異常例を示す。

(1) 障害の対応状況の可視化

図 3 は障害対応の進捗を示している。総発生障害数だけでなく, 重要度・優先度の高い障害数, 未解決の障害数, 未解決の障害の平均滞留時間を可視化している。図中の未解決の障害数や平均滞留時間の増加傾向が続く場合, 深刻な障害の発生による作業遅延や更に詳細な調査から障害対応に対するリソース不足などを検出することができる。

(2) プログラミング・コード変更作業の可視化

図 4 のグラフ 1 はファイル数とステップ数の推移を時系列で示している。グラフ 2 と 3 は 1 週間単位で, その週に構成管理システム上で行われた更新回数と, それぞれの更新における追加行数/削除行数の合計を示している。分析者は, まず, グラフ 1 から規模の異常な変動を確認する。コーディング工程初期ならば, 行数の変動がない状態や大幅な減少を「異常」とするが, テスト工程では大幅な増減は「異常」である可能性が高い。また, グラフ 1 で異常がない場合でも, グラフ 2 で多数の更新が行われている場合「異常」として検出する。特に, 結合試験以降での変更回数増加は, 大きな問題の兆候と考えられる。グラフ 3 では変更の規模を確認できる。グラフ 1 は変更結果の規模を示し, 追加・削除それぞれの規模を確認できない。グラフ 3 では, コーディング工程初期の大規模な削除や

(注 4): WWW でファイルの転送に使われる HTTP を拡張し, クライアント (Web ブラウザ) から Web サーバ上のファイルやフォルダを管理できるようにした仕様。http://e-words.jp/w/WebDAV.html

表 3 目的別計測メトリックス
Table 3 Metrics for each goal.

目的	メトリックス	背景説明
要員に起因するリスクを管理する	欠陥密度 (*)	特定のコンポーネントの欠陥密度が高い場合、担当者の知識・技術レベルに問題がある可能性がある。
	要因別欠陥数	設計理解不足による欠陥が多い場合、担当者の知識・技術レベルに問題がある可能性がある。
	欠陥修正工数	欠陥修正にかかる工数が大きい場合、担当する要員が不適切ということが考えられる。修正量が多いが修正工数が小さい場合、言語知識不足などによる単純ミスが多いことが考えられる。
	平均変更者数 (*)	1 ファイルを複数名で変更している場合、作業ごとの要員配置が不適切、あるいは要員変更などのリスクを伴う状況が考えられる。
組織内外のコミュニケーションを円滑に行う	障害平均滞留時間 (*)	テスト工程での障害解決までの滞留時間が長い場合、対応要員の配置や分担に問題があることが考えられる。
	要因別欠陥数	仕様伝達ミスや周知連絡の不徹底などによる欠陥が多い場合、組織内外のコミュニケーションプロセスに問題がある可能性がある。
	モジュールタイプ別欠陥数	異なるコンポーネントとのインタフェース関連のモジュールに欠陥が多い場合、仕様の確定・レビュープロセスに問題がある可能性がある。
変更の安全性を確保する	欠陥モジュールのインタフェース構成	複数のコンポーネントにまたがるインタフェースもつモジュールの欠陥の場合、組織間・開発者間のコミュニケーション不足による修正漏れや二次不具合発生危険性が高い。
	変更履歴と欠陥情報の不整合 (*)	欠陥として修正完了しているが、構成管理上の更新漏れなどによる。変更管理プロセスが確立していない可能性がある。
	変更モジュールと欠陥情報の不整合 (*)	欠陥として登録されているコンポーネント名と変更ファイルが一致しないなど、変更ミスや欠陥情報の入力ミスなどが考えられる。
	平均変更者数 (*)	1 ファイルを複数名で変更している。変更の上書きなどによるデグレードの危険がある。
	混入工程別欠陥数	テスト工程で混入する欠陥が多い場合、影響範囲の確認など変更管理不足によるデグレードが考えられる。
	変更ファイル数 (*)	開発後半にファイル数が増減する場合、構成管理への登録漏れ、登録ミスなどが考えられる。
既存コードの効率的・効果的活用を行う	追加行数・削除行数 (*)	変更量が多い場合、変更影響による二次不具合の危険性が高くなる。
	要因別欠陥数	修正ミス、再利用ミスによる欠陥が多い場合、改造・流用コードの変更管理プロセスが確立していない可能性がある。
	コンポーネント別流用率	流用率の高いコンポーネントでの改造や修正で、流用元の欠陥が多い場合、流用先でも欠陥が多いことが考えられる。
	コンポーネント別バグ発生数 (*)	類似機能で流用したコンポーネント間でバグの欠陥件数が大きく異なる場合、流用が適切でなかった可能性がある。
	発生箇所別欠陥数	流用・改造箇所での欠陥が多い場合、改造・流用コードの変更管理プロセスが確立していない、若しくは流用元のプログラムの品質の問題がある可能性がある。
	追加行数・削除行数 (*)	流用コンポーネントで、一つの欠陥修正や製造工程前半での削除・変更量(行数)が多い場合、流用前の検討不足や流用に伴う危険性が高いと考えられる。
	変更ファイル数 (*)	流用コンポーネントで変更ファイル数が多い場合、流用に伴う危険性が高いと考えられる。
	ファイル数 (*)	流用コンポーネントでファイル数が大きく増減する場合、構成管理上のファイルの再構成などコストやミスの危険性が高いと考えられる。
	修正工数	流用箇所での修正工数が大きい場合、流用機能が対象システムに不適合である、あるいは流用コードが複雑で、修正が困難などの問題が考えられる。
	工程別欠陥数 (*)	流用箇所での欠陥が後工程で多く発生する場合、流用前のレビュー不足や対象システムへの流用機能の不適合が考えられる。

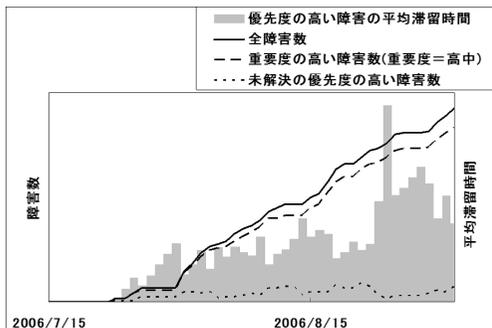


図 3 障害対応進捗グラフ
Fig. 3 Graph of fault management.

テスト工程での大規模な追加/削除をチェックし、「異常」として検出する。

(3) 開発者の作業範囲の可視化

図 5 は、2 人以上で変更されたファイル数と 1 ファイル当りの平均変更者数を示している。構成管理ツールへの更新を開発者自身が直接行っている場合、上記のデータが要員の状況を把握するために有効である。これらのメトリックスは、各開発者の担当範囲をまたがるような広範囲若しくは複雑な作業が発生すると増加する。これらが増加し続けると、要員配置や作業分担などを見直す必要がある可能性がある。図 5 のよう

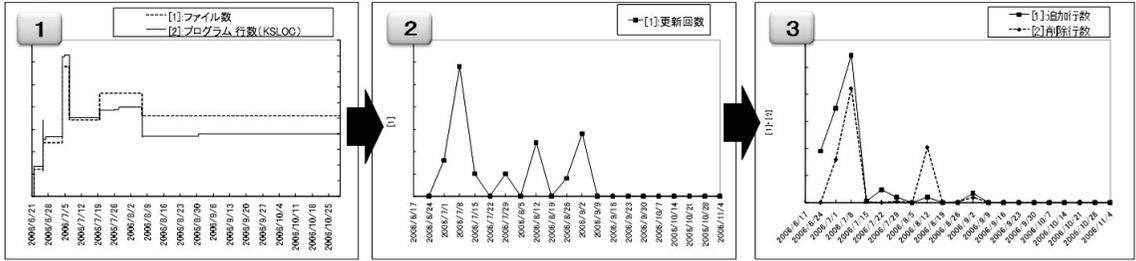


図 4 コンポーネント別規模・変更回数・変更規模推移グラフ

Fig. 4 Graphs of size/number of file changes/number of change volume for a certain component.

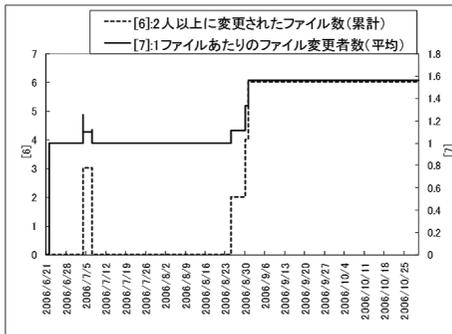


図 5 ファイル別変更者分析グラフ

Fig. 5 Graph for analysis of members who have changed program files.

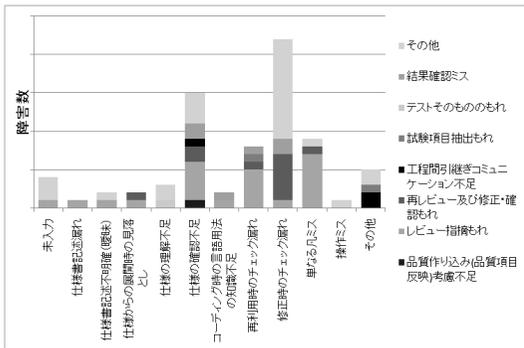


図 6 障害の発生原因と混入要因による分析

Fig. 6 Analysis of defect causes and causes of detection delay.

に一時期に急増し安定する場合、要員の一時的な追加や要員変更が考えられる。要員変更の場合、対象コンポーネントでの作業遅延リスクなどを考慮しなければならない。

(4) 障害発生原因と混入要因の関連の可視化

図 6 は、ある拠点での社内結合試験までに発生した

障害の発生原因の種類別件数の棒グラフで、「問題を抽出できなかった要因」（以降、混入要因とする）別に分類している。この図から、既存コードの流用（『再利用時のチェック漏れ』）や変更・バグ修正時のデグレード（『修正時のチェック漏れ』）の危険性を判定することができる。また、その「混入要因」を精査することにより、プロセス改善の指針を導出することができる。

このほか、設計工程でのレビュー欠陥データ分析や設計ドキュメントの変更頻度・変更量の分析も行われた [12]。

4. 適用結果

4.1 分析データの概要と分析単位

5社6拠点（1社は2拠点[二つの異なる事業部]で開発）から毎週提出される成果物の構成管理履歴データ及び障害管理データに対して分析を行った。

第1フェーズは、主に拠点別の分析を行い、一部の特徴的なコンポーネントに対しては対象コンポーネントのデータのみを抽出して分析を行ったが、第2フェーズは、コンポーネント単位で分析を行った。これは、改造開発と新規開発が混在し、モジュールの共通化なども行われたため、拠点単位の分析では状況把握が困難であったためである。ただし、障害情報に関しては、一つひとつの障害をコンポーネント別に分類するのが困難であることと、開発管理体制や要員体制にかかわる問題の検出が目的であることから、拠点別の分析が妥当と判断し、拠点別に分析を行った。本章では、第2フェーズでの適用結果について述べる。

4.2 指摘事項の分類

リアルタイム型のメトリクスについては、問題発生が推定されるポイントを抽出後、必要ならば詳細調査を行った。グラフ間（障害グラフと規模推移グラフなど）を対応させ、異常の原因が分かればそれを社別

のメーリングリストを通して指摘した。また、原因が分からない異常については、同じメーリングリストを通して質問し、回答を求めた。データ集約型のメトリックスについては、各社がテスト密度・バグ密度などを分析した品質評価報告書と本手法による分析結果と併せて、エグゼクティブPM、各組織のPM若しくはプロジェクトリーダーや開発者、分析者（本事例ではEASE プロジェクト研究員）が参加する会議に計られ、問題点に関して質疑応答や議論が行われた。質問と回答の例は、文献 [12], [20] に示している。

本手法の適用によって第2フェーズの対象プロジェクトの開発中に各社に問い合わせた質問や指摘事項は、その回答内容によって以下の観点で分類し、集計した。各指摘の件数を表4に示す。この分類によって、EASEインプロセスメソッドが目的とする問題検出に対する有効性と、運用上の難易度を知ることができる。

- 異常指摘（問題発生が推定される状況の指摘）：例）異常な欠陥密度、開発完了（進捗100%）でのコード更新など
- 入力データ品質確認（入力データの不整合・ミス）：例）単体試験工程で発見された障害の混入工程が結合試験など
- データ収集プロセス確認（ツール運用上のミス・データ提出漏れ）：例）障害の登録漏れ（遅延）、完了（更新）漏れなど

表4から、問題検出につながる可能性がある異常指摘は83件で、全指摘の約61%を占める。残り約39%が、本手法の適用目的を阻害する運用上の問題である。データ収集プロセスについては全指摘の約7%で、割合は低い。入力データ品質についての指摘は、手入力データのタイプミスなどで、特に障害管理データに多かった。未回答/不明など事実背景が確認できなかった指摘は、開発が佳境に入っていた時期のメールによる指摘が多かった。また、会議での指摘に

ついては、実際の開発者が会議に出席していなかったため、状況が把握できなかったものがあった。

5. EASEインプロセスメソッドの適用評価

本章では、適用事例からEASEインプロセスメソッドの実プロジェクトにおける有効性を評価する。

5.1 評価の観点

EASEインプロセスメソッドの実プロジェクトにおける有効性を評価する観点として、以下の点を挙げる。

- (1) 本手法の導入目的に対する対象プロジェクトでの達成度
 - (a) プロジェクトの管理支援の計測目的に対して、計測したメトリックスとその可視化情報は適切だったか？
 - (b) 組織間の機密を保持できたか？
 - (c) リアルタイムの分析データフィードバックと、それによる改善・対策が実施されたか？
- (2) 本手法の他のプロジェクトへの適用可能性
 - (a) 検出された問題の一般性
 - (b) 分散開発・マルチベンダ開発における有効性
- (3) 本手法の実プロジェクトへの適用コスト
 - (a) ツール導入
 - (b) 運用環境の整備
 - (c) ツールを用いたデータ分析
 - (d) データ分析結果から異常検出

5.2 評価方法

本手法の評価のため、以下の二つの情報を用いる。

- プロジェクト関係者へのインタビュー
- プロジェクト進行中に観察できた事実

インタビューは、対象プロジェクトにおける「新しいソフトウェア工学的計測手法の導入の有効性」の評価のために行われたもので、直接的に本手法の評価を行うものではない。そのため、実際の評価結果から、本手法の評価につながるもののみをここで取り上げる。

インタビューは、表1の異常指摘83件のうち28件について、エグゼクティブPMと各社開発担当者と対面形式で行った。インタビュー時間に限りがあるため、各社4~9件を抽出した。また、28件のうち1件は、4件の類似した異常指摘を集約している。したがって、インタビューを行った異常指摘の割合は、全83件中31件で約37%であった。インタビューへの回答結果の詳細は、付録に掲載する。文中の「問題」とは、プロセス改善など何らかの対応を必要とする状況、若し

表4 指摘に対する回答内容の分類表
Table 4 Classification of answers for each indicated issue.

指摘 タイミング	異常 指摘	入力 データ 品質確認	データ 収集プロ セス確認	未回 答/ 不明	合計
設計工程 後会議	34	7	3	1	45
週次	35	11	3	10	59
社内試験 工程後会議	14	11	3	4	32
合計	83	29	9	15	136

くはその状況を引き起こしている原因を意味する。

プロジェクトの観察は、メールと工程区切りでの品質評価報告会議を通して行った。EASE プロジェクト研究員は、対象プロジェクトの開発・管理者用メーリングリストに参加し、メールで配布される会議議事録や検討課題についてのやり取りを随時観察していた。また、工程区切りでの品質評価報告会議では、各社開発担当者の品質評価報告やエグゼクティブ PM との質疑応答に直接参加し、プロジェクト中に発生した問題や対策などの情報を収集した。

5.3 評価結果と考察

本節では、5.1 の各観点に対する評価結果について述べ、考察する。

(1) 本手法の導入目的に対する対象プロジェクトでの達成度

(a) プロジェクトの管理支援の計測目的に対して、計測したメトリックスとその可視化情報は適切だったか

インタビューした 28 件の異常指摘が、3.3 のどの目的に関連するかを分類した結果を以下に示す。

- 要員に起因するリスクを管理する：6 件
- 組織内外のコミュニケーションを円滑に行う：2 件
- 変更（要求・設計・プログラム）の安全性を確保する：13 件
- 既存コードの効率的・効果的活用を行う：5 件
- その他：2 件

選択したメトリックスを用いて、28 件中 26 件は意図した問題を検出し、異常指摘 83 件中 26 件、約 31% の異常指摘が対象プロジェクトで有効な指摘であった。したがって、設定した目的に対して選択したメトリックスが適切であったと考えられる。インタビューできなかった異常指摘を考慮すると、この割合は更に高いと予想される。中でも、要員に関する異常指摘が 6 件と 2 番目に多い。これらは、平均変更者数、欠陥修正工数、変更行数と修正工数のアンバランス（変更行数が膨大なのに、修正工数が少ない）、設計工程の欠陥密度などのメトリックスから、判明している（表 3 参照）。開発体制が不透明なプロジェクトにおいて、要員に起因する問題の検出は有効である。また、2 件あったその他の問題は、品質管理プロセスの不備などだった。

(b) 組織間の機密を保持できたか

本手法では、CASE ツールから直接データ収集し、

設計書・プログラムコードや障害票などの内容に触れることなく、定量的に計測・分析されたデータをもとにプロジェクト管理が行われた。したがって、開発組織外部の管理者・開発者とデータ共有にあたって、重要な機密（処理のアルゴリズム、要員情報）の流出はなかった。また、すべて電子データでネットワークを介して授受されたが、送受信過程でのデータ流出も報告されていない。

(c) リアルタイムの分析データフィードバックと、それによる改善・対策が実施されたか

コーディング工程以降では、金曜日までの作業が月曜日に分析され、問題はその日のうちに発見された。提案手法によるフィードバックで、各社からの進捗報告（毎週水曜日）に先立って作業状況が把握でき、プロジェクト管理に有効に用いられた。インタビュー結果から、指摘された問題若しくは問題原因のほとんどが、プロセス改善などの対処が必要で、かつ対処が行われたことから、本手法は開発中の問題の早期解決や問題発生の予防に有効であると考えられる。

(2) 本手法の他のプロジェクトへの適用可能性

(a) 検出された問題の一般性

インタビューに回答した開発者は、指摘された問題のほとんどが他のプロジェクトでも発生する問題と認識している。このことから、本手法で用いられたメトリックスは、他のプロジェクトでも問題の検出に有効であると考えられる。

(b) 分散開発・マルチベンダ開発における有効性

指摘した問題は、組織内のプロジェクト管理者によって指摘前に把握されていたが、その手段は日常的な進捗会議や障害票のチェックである。本事例では、組織外にいるエグゼクティブ PM は提案手法による指摘で問題を知った。各組織からの報告に依存する分散開発やマルチベンダ開発におけるエグゼクティブ PM にとって、本手法はリアルタイムに現場の情報を得る有効な手段であるといえる。また、ほとんどの収集・分析データは電子データでネットワークを介して授受され、物理的に距離のある 6 地点間で容易に実施できた。このことから、更に複雑で規模の大きな複数組織での開発や海外委託開発でも、本手法の適用が可能であると考えられる。

その他の有益な点として、週次で各社とエグゼクティブ PM が行っていた進捗会議での進捗報告を、本手法の導入によって省略することができたことが挙げ

られる。エグゼクティブPMは、各社からの進捗報告レポートをメールで受け取り、更に本手法による分析報告を参照することによって、進捗会議での報告に代わる詳細な情報を入手できる、と判断した。進捗会議での報告の省略は、エグゼクティブPM、及び各社プロジェクトマネージャの時間的コストを大幅に低減したといえることができる。

(3) 本手法の実プロジェクトへの適用コスト

対象プロジェクトにおける本手法の導入・実施コストは、以下のとおりである。EASE プロジェクトという第三者組織によるデータ分析であるため、通常のプロジェクトで必要となるコストとは同等とはいえないが、分散拠点の複数組織による開発のプロジェクトマネジメントとしては、類似する点多いと考えている。

(a) ツールの導入

今回は、実質各社内でもEPM及びEPM-ProStarを導入してはいないが、EPMの導入については2人日くらいのコストがかかる、と報告されている[9]。EPM-ProStarについては、インストールはCDからのファイルコピーのみであるが、データ分析の粒度やタイミング、可視化の形態等によって、ツールの初期設定などは必要となる。これらについても、5社のデータ構造などの調査やフィードバック方法の検討などを含めて、1人日程度の作業が必要であった。

(b) 運用環境の整備

ネットワークを用いたので、データ授受のWebDAV環境の整備、メーリングリストの作成などの作業が発生した。対象プロジェクトでは、他の用途でネットワーク環境は整備されていたので、特に多大な作業が発生したことはない。

(c) 週次・工程区切りでのデータ分析

EPM、EPM-ProStarを用いた週次データ分析に関しては、入手データの構造に対するツールの初期設定や調整が必要であったが、2週目以降はデータの入手から分析結果の出力・レポート作成まで、2~3時間の単純作業(6組織・34コンポーネント分)で完了した。上流工程での工程区切りの分析には、手作業によるデータ加工を行ったため、2人日程度必要であった。工程区切りに、データ集約型の分析を行う際にも、市販ツールを用いた手作業で行われたが、実質1人日程度であった。これらの作業は、ソフトウェア管理に関する専門知識のない、一般事務作業員によって実施さ

れた。

(d) データ分析結果からの異常検出

週次の分析で約130くらいの出力グラフは、実際の管理者若しくはプロジェクト管理経験者によって手作業で検証され、異常が検出されない場合は、1人時程度の作業であった。異常が検出された場合は、詳細なデータ検証やフィードバックのメールの作成・送信などの作業に2~3人時必要な場合もあった。

本手法の導入、すなわちツールのインストールやツールの運用、分析・フィードバックなどの作業は、通常の進捗管理と比較して著しく大きいわけではない。初期のネットワークやツール運用環境整備には、各組織の状況に応じて作業が発生するが、これも初期のみと考えてよい。また、最後の作業以外は、プロジェクト管理者やプロジェクトリーダー自身が行う必要がないので、コストの低減を図ることも可能と考えられる。

6. む す び

本論文では、複雑化するプロジェクトの組織・体制に適した進行中のプロジェクト管理を支援する手法、EASEインプロセスメソッドについて説明し、実プロジェクトに適用して有効性を評価した結果について報告した。本手法は、EASEプロジェクトにより開発された自動データ収集・分析・可視化ツールを用い、それらを進行中のプロジェクトに適用するための具体的な管理支援プロセスの定義方法を包含する。本手法により、分散・遠隔環境での開発プロセスの透明化を図り、問題の早期発見を可能にすることで、プロジェクトの遅延リスクを低減できる。

我々は、マルチベンダ開発プロジェクトのプロジェクト管理に本手法を適用し、その有効性を検証した結果、以下のことが分かった。

- EPM収集データから計測されるメトリックスは、プロジェクト進行中に対処が必要とされる問題状況の顕在化に寄与している。

- 指摘された問題のほとんどが、一般的なプロジェクトでも発生する可能性があるもので、本手法は一般的なソフトウェア開発プロジェクトに適用可能である。

- 問題を指摘前に把握していたケースでは、ほとんど日常的な進捗会議や障害票のチェックによって検出されており、日常的に現場とコンタクトがとれない環境(外部組織への開発委託・オフショア開発など)では、本手法はより有効であると考えられる。

● 本手法自体の導入・運用コストは小さく、本手法の導入によるプロジェクト管理者や進捗報告のための作業コストを削減することができた。全体的には、プロジェクト管理コストの低減が可能になる、と考えられる。

今後の課題としては、より多くのプロジェクトへの適用による目的に応じたメトリックスモデルの確立と、その事例検証による問題状況・原因の自動検出方法の確立が挙げられる。

謝辞 本研究の一部は、文部科学省「次世代 IT 基盤構築のための研究開発」の委託に基づいて行われた。本研究にあたり多くの御協力を頂いたソフトウェアエンジニアリング技術研究組合参加企業、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、EASE プロジェクトの関係諸氏に感謝します。

文 献

[1] P.M. Institute, A Guide to the Project Management Body of Knowledge (PMBOK Guides), Project Management Institute, 2004.

[2] W.S. Humphrey, Managing the Software Process, Addison-Wesley, 1989.

[3] M.C. Paulk, B. Curtis, M.B. Chrissis, and C.V. Weber, "Capability maturity model, version 1.1," IEEE Software, vol.10, no.4, pp.18-27, 1993.

[4] C.P. Team, "CMMI for development, version 1.2," 2006.

[5] M.B. Chrissis, M. Konrad, and S. Shrum, CMMI: Guidelines for Process Integration and Product Improvement, 2nd ed., Addison-Wesley, 2006.

[6] 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, 2006.

[7] A. Endres and D. Rombach, A Handbook of Software and Systems Engineering, Empirical Observations, Laws and Theories, Pearson Education Limited, UK, 2003.

[8] V.R. Basili and H.D. Rombach, "The TAME project: Towards improvement-oriented software environment," IEEE Trans. Softw. Eng., vol.14, no.6, pp.758-773, 1988.

[9] 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," 信学論 (D-I), vol.J88-D-I, no.2, pp.228-239, Feb. 2005.

[10] V.R. Basili, "Goal question metric paradigm," in Encyclopedia of Software Engineering, ed. J.J. Marciniak, pp.528-532, 1994.

[11] 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," Technical Report NAIST-IS-TR2007002, 2007.

[12] 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," Technical Report NAIST-IS-TR2007005, 2007.

[13] 松村知子, 門田暁人, "ソフトウェア開発プロジェクト管理支援のための構成管理・障害管理データの活用," Technical Report NAIST-IS-TR2008002, 2008.

[14] 大杉直樹, 松村知子, 森崎修司, "ソフトウェアの見える化を支援するデータ分析力—エンビリカルアプローチによる既存データの有効活用," JISA 会報, vol.80, pp.13-29, Jan. 2006.

[15] EASE プロジェクト, "データ収集に基づくソフトウェア開発支援システム," e-Society 文部科学省リーディングプロジェクト基盤ソフトウェアの総合開発平成 17 年度報告書, pp.37-44, Dec. 2006.

[16] 楠本真二, "日立公共システムエンジニアリング株式会社様での EPM 収集データの詳細分析状況を報告," EASE PROJECT NEWS LETTER, vol.4, p.8, Dec. 2005.

[17] 卯川悟史, "自社の開発プロジェクトへの EPM の適用結果を紹介," EASE PROJECT NEWS LETTER, vol.4, pp.8-9, Dec. 2005.

[18] 樋口 登, "先進ソフトウェア開発プロジェクト," SEC J., vol.2, pp.56-57, 2005.

[19] 松浦 清, 神谷芳樹, 樋口 登, "先進ソフトウェア開発プロジェクト PARTII," SEC J., vol.5, pp.44-49, 2006.

[20] 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," Technical Report NAIST-IS-TR2007001, 2007.

付 録

開発者へのインタビュー集計結果

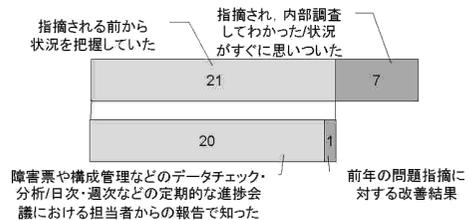


図 A.1 指摘事実の認識

Fig. A.1 Recognition of the indicated issues.

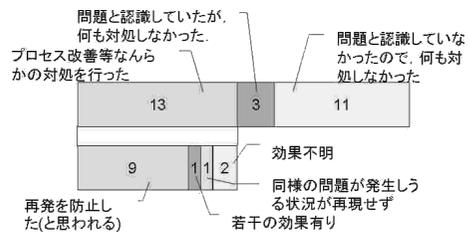


図 A.2 指摘された問題への対処

Fig. A.2 Action and result against the indicated issues.

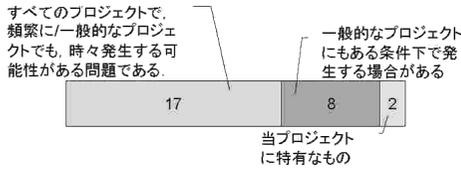


図 A・3 問題の一般性

Fig. A・3 Generality of the found issues.

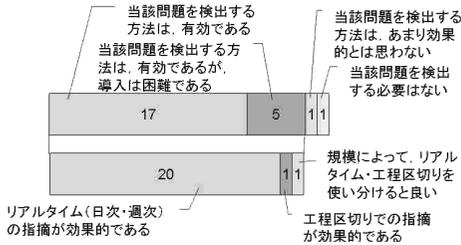


図 A・4 EASE インプロセス手法の有効性と改善方法

Fig. A・4 Effectiveness and improvement of EASE in-process method.

(平成 21 年 3 月 19 日受付, 6 月 15 日再受付)



松村 知子 (正員)

平 16 奈良先端大・情報科学・博士後期課程了。同年より同大学産学連携研究員。博士(工学)。ソフトウェアのフォールト検出技術, ソフトウェア開発プロジェクト管理, エンピリカルソフトウェア工学の研究に従事。IEEE 会員。



森崎 修司

平 13 奈良先端大・情報科学・博士後期課程了。同年(株)インターネットインシアティブ入社。オンラインストレージサービスの立ち上げ/企画/開発, RFID ソフトウェアの国際標準策定活動に従事。平 17 奈良先端大・情報科学・産学連携研究員。平 19 同大助教。エンピリカルソフトウェア工学の研究に従事。博士(工学)。IEEE, IPSJ 各会員。



勝又 敏次

昭 48 幾徳工業高等専門学校・電気工学科卒。同年日本電信電話公社(現 NTT データ)入社。中央省庁系中大規模システムの開発・プロジェクト管理に従事。平 20 NTT データアイ入社。中央省庁系システムの開発・プロジェクト管理に従事。IPSJ, プロジェクトマネジメント学会各会員。



玉田 春昭 (正員)

平 11 京都産業大・工・情報通信卒。平 13 同大大学院博士前期課程了。同年住商エレクトロニクス(株)入社。平 18 奈良先端大・情報科学・博士工期課程了。同年同大学産学連携研究員。平 19 同大・情報科学・特任助教。平 20 京都産業大・コンピュータ理工・助教。博士(工学)。ソフトウェアセキュリティ, エンピリカルソフトウェア工学の研究に従事。IEEE, IPSJ 各会員。



吉田 則裕 (学生員)

平 16 九工大・情報工・知能情報卒。平 18 阪大・情報科学・博士前期課程了。現在同大大学院博士後期課程在学中。コードクローン分析やリファクタリング支援の研究に従事。IPSJ, 人工知能学会, IEEE 各会員。



楠本 真二 (正員)

昭 63 阪大・基礎工・情報工学卒。平 3 同大大学院博士課程中退。同年同大学基礎工学部情報工学科助手。平 8 同学科講師。平 11 同学科助教授。平 14 大阪大学情報学部コンピュータサイエンス学科助教授。平 17 同学科教授。博士(工学)。ソフトウェアの生産性や品質の定量的評価, プロジェクト管理に関する研究に従事。IEEE, JFPUG 各会員。



楠本 健一 (正員)

昭 60 阪大・基礎工・情報工学卒。平元同大大学院博士課程中退。同年同大学基礎工学部情報工学科助手。平 5 奈良先端科学技術大学院大学情報科学研究科助教授。平 13 同大学情報科学研究科教授。博士(工学)。ソフトウェア品質保証, ユーザインタフェース, ソフトウェアプロセス等の研究に従事。IPSJ, 日本ソフトウェア科学会, IEEE, ACM 各会員。