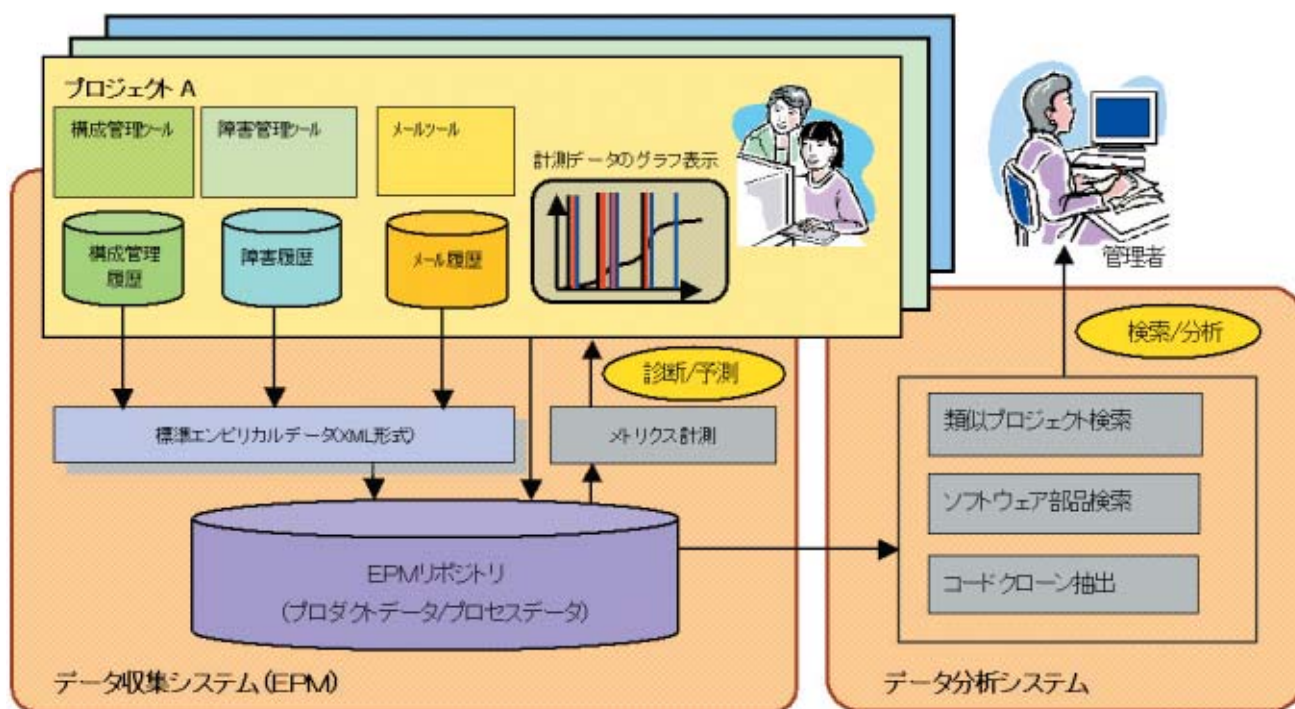


産学官連携によるエンピリカルソフトウェア工学 プロジェクトEASE

EASE Project: Empirical Software Engineering under a Collaboration of Industry, Academia, and Government

井上 克郎* *Katsuro Inoue* 津田 道夫*** *Michio Tsuda*
松本 健一** *Ken-ichi Matsumoto* 新海 平*** *Taira Shinkai*



EASEによるソフトウェア開発プロセスの改善

EASEは、データ収集システムとデータ分析システムから構成され、ソフトウェア開発のプロジェクトデータを計測・定量化して、評価・フィードバックにより開発プロセスを改善する支援環境を提供する。

EASEプロジェクト(以下、本プロジェクトと言う)は、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」のひとつとして、2003年5月に開始した産学官連携プロジェクトである。

ソフトウェア開発における生産性と品質の確保は、ソフトウェア産業界の重要なテーマである。しかしながら過去に多くの支援技術が開発されたが、いまだに不十分なのが実状である。

本プロジェクトは、「定量データを計測し、科学的根拠により分析・フィードバックしてソフトウェア開発の生産性と品質を継続的に改善する」エンピリカルソフトウェア工学

の実用化を目的にしている(EASE ;Empirical Approach to Software Engineering)。

EASEは、データ収集システム(EPM; Empirical Project Monitor)とデータ分析システムからなるプロセス改善支援システムである。EPMは、オープンソースの管理ツールからデータを自動収集してソースコードの規模推移などをグラフで表示する支援システムで、現在評価版がリリースされている。

株式会社 日立システムアンドサービス(以下、日立システムと言う)は産業界メンバの一員として、本プロジェクトの開発と評価に参画している。

*大阪大学 **奈良先端科学技術大学院大学 ***株式会社 日立システムアンドサービス



1 はじめに

日本のソフトウェア産業は多くの課題を抱えている。主要な情報サービス企業600社で構成する(社)情報サービス産業協会(JISA)のソフトウェア開発実態調査¹⁾では、売上金額(売上総額16,600億円)比でプロジェクトの納期遅れ9%(納期遅れプロジェクトの売上額1,542億円)、低品質による手直し発生60%、コスト超過49%の報告がある。同じくJISAの基本統計調査²⁾では、30%の企業が売上高経常利益率2%未満で、厳しい業績結果が出ている(加重平均4.85%、中央値3.10%、回答368社)。

またコンサルタントの松原友夫氏は、わが国のソフトウェア産業の実態とインド、中国など海外企業との国際比較において、その脆弱性に警告をしている³⁾。

情報サービス企業が、企業体質を強化して業績確保を図るには「ソフトウェア開発プロセスの計測と改善を継続して開発組織の成熟度を上げる」施策が有効である。開発対象物を計測して分析、改善する科学的なアプローチをエンピリカル(実証的)ソフトウェア工学(Empirical Software Engineering)⁴⁾と呼び、近年注目を集めている技術である。

本プロジェクトは、ソフトウェア開発データを自動的に収集・分析してプロセス改善を図る実証的ソフトウェア工学環境(EASE)を構築する実用化プロジェクトである。

本稿では産学官プロジェクトの特徴を述べ、次にEASEのデータ収集システム(EPM)の機能を述べる。

2 産学官プロジェクト

文部科学省は、ライフサイエンス、情報通信、環境、エネルギー、ナノテクノロジーなどの科学技術振興を目的にした支援事業を実施している。本プロジェクトは情報通信分野の施策である「e-Society基盤ソフトウェアの総合開発」のひとつとして採択され、2003年より5ヵ年計画で開始した。テーマ名は「データ収集に基づくソフトウェア開発支援システム」である。本プロジェクトでは、大学から大阪大学と奈良先端科学技術大学院大学の研究者、企業から日立システムなどの開発要員が大阪に設置したEASEラボに集結して研究と開発をしており、従来にない産学連携体制を作っている。

EASEラボを中心にして、研究・開発、実践と実データの収集、分析による新たな知見を提供するエンピリカルモデルを実現している(図1参照)。

また中堅企業を中心にEPM産業部会を作り、企画提案、設計レビュー、成果の評価と普及を推進しており、利用者と開発者が協調してプロジェクトを推進している。

この他、本プロジェクトは米国メリーランド大学Basili教授、ドイツ・カイザースロータン大学Rombach教授、米国サザンカリフォルニア大学Boehm教授、オーストラリア・ニューサウスウェールズ大学Jeffery教授をアドバイザーとして国際連携を図っている。特にRombach教授はドイツの産学連携研究機関フラウンホーファ(Fraunhofer)実験的ソフトウェア工学研究所(Institute for Experimental Software Engineering)の所長でもあり、100

名余りの研究員がソフトウェア開発技術の評価やソフトウェア品質向上システムの設計に従事している。

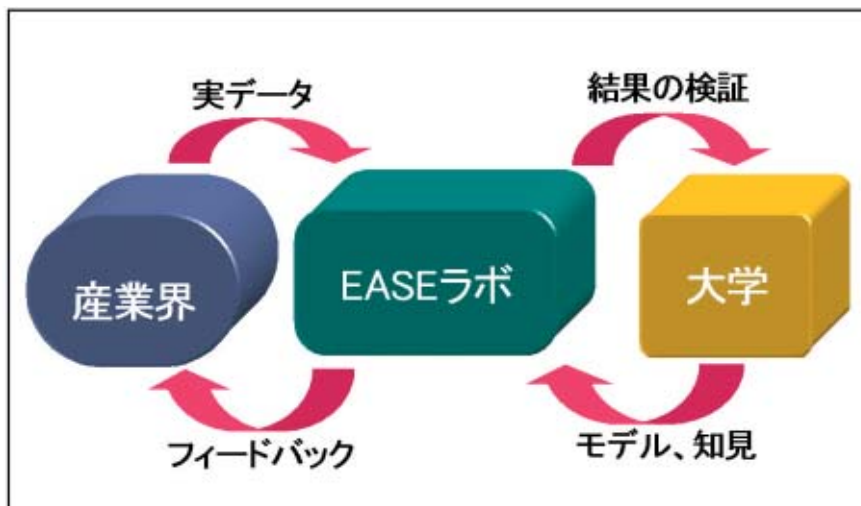


図1 エンピリカルモデル

EASEラボを中心にエンピリカル工学を実践して企業のソフトウェア開発プロセスを改善する。

3 エンピリカルソフトウェア工学の実用化

EASEの目的は、エンピリカルソフトウェア工学を実現する支援環境の構築である。ソフトウェア開発データの計測・定量化、データの分析・評価とフィードバックによる継続的なプロセス改善により生産性と品質を向上させる。

EASEの利用者は、ソフトウェア開発を主たる業務にしている情報サービス企業である。このためEASEでは、各企業の組織、開発環境に依存しない柔軟な支援システムの実現を図っている。

3.1 ソフトウェア開発データの計測・定量化

プロジェクトデータの効率的収集を目的に以下の方針で開発する。

- (1) プロジェクトの負荷をかけないデータ自動収集
- (2) リアルタイムのデータ収集
- (3) 標準エンピリカルデータの公開

プロジェクトで使われている管理ツールからデータを自動収集する。管理ツールとして、工程管理、構成管理、障害管理、原価管理などのツールがあり、オフィスツールとしてドキュメント作成ツール、メールツールなどがある。EASEでは、開発現場で普及しているオープンソースの管理ツールとオフィスツールをデータ計測対象ツールにしている。

自動収集により、加工していない生のデータを収集する。プロジェクトメンバーに負担をかけずにデータを収集するのは、EASEの利用者である多くの中堅情報サービス企業には特に効果的である。また、人の判断の入らない生のデータ収集により、分析の精度確保と知識の蓄積が図れる。データは、主にプロダクトデータを主体に収集し、標準ルールによりプロセスデータに変換する。例えば、「組合せテスト中にソースコードの量が10%増加したなら、まだプログラム仕様が安定していないと判定して、設計レビューを再度実施する」などである。

標準エンピリカルデータは広く公開して、支援ツールベンダやオープンソースコミュニティでの利用拡大を推進する。

3.2 ソフトウェア開発データの分析・評価

データの評価方法についてZelkowitzとWallaceは以下の3つに分類している⁵⁾。

- (1) 観測型 (Observational); 実際に行われているプロジェクトを横から観測して評価する方法。
- (2) 履歴型 (Historical); 既に終了した工程やプロジェクトのデータを評価する方法。
- (3) 制御型 (Controlled); 目的のデータを得るために環境を作って評価する方法。

EASEは観測型の評価を実現する。ソフトウェア規模や障害などのプロダクトデータを適宜観測する。

分析結果は速やかにプロジェクトにフィードバックする。計測データはEPMリポジトリに蓄積して履歴型の評価により、ガイドラインや基準値の作成に反映させる。

EASEでは、分析・評価ツールとして、類似プロジェクト検索ツール⁶⁾、ソフトウェア部品検索ツール⁷⁾、コードクローン検出ツール⁸⁾を開発している。類似プロジェクト検索ツールは、協調フィルタリング技術を用いて過去のプロジェクトデータから類似度の高いn個のプロジェクトを検索して、規模、品質、リスクなどを予測する。開発プロジェクト毎に計算するので高い予測精度が期待できる。ソフトウェア部品検索ツール(対象言語;Java)は、部品間の類似度(Similarity)を計算して類似した部品の集合を部品群に分ける。次に入力した検索キーに近い部品から順次検索する。要求と部品の適合度、部品利用実績順位のふたつの観点から検索する。コードクローン検出ツール(対象言語;C++,Java,COBOL)は、ソースコードに存在するコードクローンを抽出するツールである。コードクローンとは同一または類似したコードの断片で、多くは他のプログラムから「コピー」されて組み込まれている。コードクローン量の計測で、ソフトウェアの部品化や最適なプログラム規模の管理が図れる。特に情報システムの保守コスト適正化に対象プログラム規模の最小化は有効である。



4 データ収集システムEPM

EPMは、プロジェクトデータを自動収集して、メトリクス(指標)を計測・表示するシステムである。

4.1 データの収集

EPMは、オープンソースの管理/オフィスツールからデータを収集する。現在、EPMがサポートしているツールは、構成管理ツールとしてCVS、障害管理ツールとしてGNATS、Bugzilla、メールツールとしてmailman、Major-domo、fmlである。

収集したデータは標準エンピリカルデータ(XML形式)に変換して、EPMリポジトリ(DB:PostgreSQL)に格納する。標準エンピリカルデータはインタフェース情報として公開されている。EPMリポジトリには、プロダクトデータとプロセスデータを格納している。

例えば、構成管理ツールCVSからはプロダクトデータとしてソースコードの規模(ステップ数)と、バージョン情報を収集し、プロセスデータとして更新、チェックイン/チェックアウトの履歴情報を収集する。同じく障害管理ツールとメールツールからは障害報告、未解決障害報告等と共にメール数の推移、障害発生/解決時期等の時間軸による履歴情報を収集する。

4.2 データの測定と表示

EPMは、収集したデータを計測してWebでグラフ表示する。ソースコードの規模や障害報告と履歴情報との関連図を作成する。EPMが作成する関連図は次に示す6種類のグラフである。

- (1) ソースコードの規模推移グラフ(図3)
- (2) チェックインとチェックアウトの関連グラフ
- (3) メール数の推移とチェックイン/障害発生時期/障害解決時期との関連グラフ
- (4) チェックインと障害件数との関連グラフ
- (5) 累積/未解決障害件数と平均障害滞留時間との関連グラフ

(6) 障害件数の推移と発生予測グラフ

例えば、メール数の推移と障害発生時期の関連を見て、プロジェクト管理者は、「障害の発生が増加したにもかかわらず、開発者間のメールによる議論が増加しないのはコミュニケーションに問題があるか、他に重要な問題が発生している」と判断する。

チェックインと障害件数との関連では、障害が増加しているにもかかわらず、チェックインが増加していなければ対策が進んでいないと推測される。

このようにプロジェクト管理者は、さまざまな角度からカレントなデータを見てプロジェクトの状況を推測する。

障害件数の推移と発生予測グラフは、発生した障害件数からソフトウェア信頼性評価モデルSRGM(Software Reliability Growth Model)を用いて予測した総障害件数を表示するグラフである。

EPMはオープンソースプラットフォームのソフトウェアアーキテクチャで稼動する(図2)。

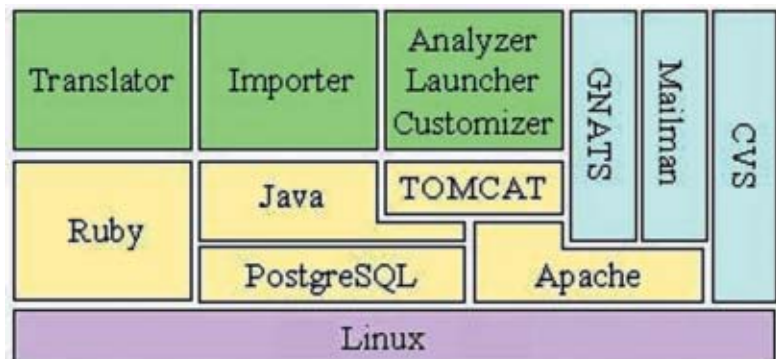


図2 EPMの実行環境

EPMはオープンソースプラットフォームで稼動する。

EPMはデータ収集・変換のTranslator, Importerとデータ測定・表示のAnalyzer, Launcher, Customizerの5モジュールから構成されている。データ変換機能(言語: Rubyスクリプト)は拡張性があるので、容易に未サポートの管理ツールデータを標準エンピリカルデータに変換できる。また測定機能(言語: Java)は、関連図の追加・修正が容易なプラグイン構造になっている。

5 EPMによるプロセス改善

ソースコードの規模推移グラフを利用したプログラム製造工程のプロセス改善の例を述べる。図3はプログラム製造工程(8月と9月)と組合せテスト工程(10月)の3ヶ月間のソースコードの規模推移である。プログラム製造工程は、プログラム設計、コーディング、単体テストの作業を経て9月29日に終了している(規模19KS)。作業の終了条件は、プロジェクトで決めてある。例えばテストケース100%終了やカバレッジ100%などである。続いて組合せテストを開始したが、10月中旬から2回にわたりソースコードの増加があった(増加規模7KS)。

プロジェクト管理者は、単体テスト終了後のソースコードと比較して増加率が37%あることから、「プログラムの品質は悪い」と判定して原因を分析する。原因が設計仕様の変更・追加なら設計ガイドの見直しと再レビューを実施すると共に、設計工程での作業内容、レビュー・承認のプロセス改善を図る。また原因がプログラム製造工程での不良なら、再度単体テストによる品質向上作業を実施すると共に、プログラム仕様レビューやコーディングチェック、単体テスト方式などの改善を図る。

6 評価

当社では、EPMを社内ツール修正作業で試行・評価をした。評価の目的は、EPMの導入親和性の評価である。適用工程は、詳細設計からプログラム製造工程である。

EPMの適用で得られたデータを表1に示す。

ツール	データの種類	データ量
CVS	ソースコードの規模	29,000ステップ
Mailman	設計情報	57件
Gnats	不良情報	21件

表1 EPM適用によるデータ件数

ソースコードの規模に比べて不良件数が少ないのは、適用対象作業がプログラム修正作業であることによる。

EPMの適用で得られた評価を以下に述べる。

- (1) プロジェクトデータの自動収集は、開発要員の報告作業を軽減させるので効果的である。
- (2) 管理ツールがオープンソースなので導入しやすい。特に海外を含む分散開発拠点での活用が期待できる。
- (3) ツール間の連携は良い。

今回は初期評価のためプロセス改善までは評価できなかった。今後は、データ分析システムも含めた評価を継続していく。

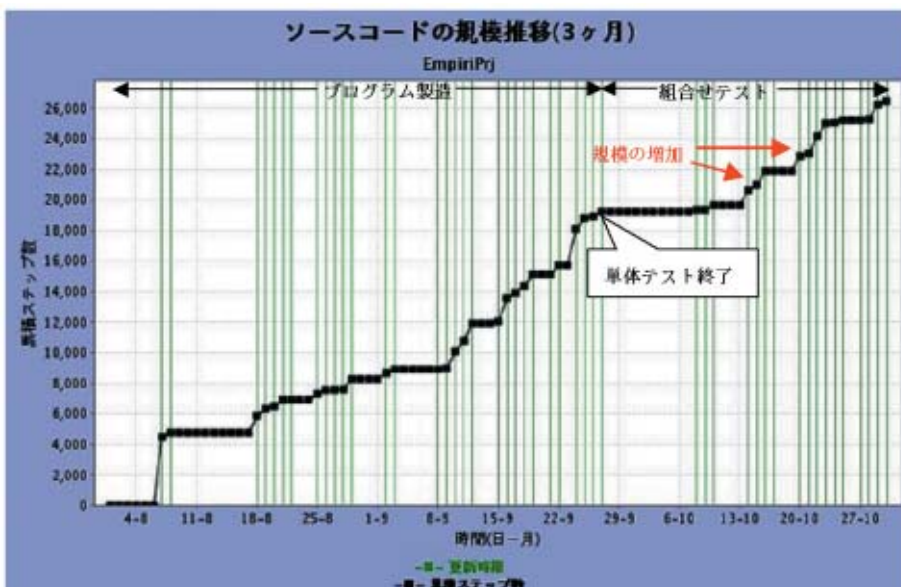


図3 ソースコードの規模推移
組合せテスト工程でソースコードの規模増加が基準を超えたらプロジェクト管理者は原因を分析して対策をたてる。



7 今後の課題

EASEは継続的なプロセス改善を実現する有効な支援システムである。しかしソフトウェア産業界への普及には課題も多い。EASEは、オープンソースの開発環境をベースにしているが、多くの情報サービス企業は独自の開発環境を構築している。各企業の開発環境とEASEを結合する標準エンピリカルデータを中心にした連携技術の開発が必要になる。またプロジェクトデータの計測がプログラム製造工程中心になっている。プロジェクトトラブルの大きな原因は見積や要求仕様の設計不良などであり、EASEの上流工程サポートが必要である。

これらの課題には、EASEラボを中心に産学連携を強めて解決していきたい。

8 おわりに

EASEは、ソフトウェア産業の体質強化を目的とした産学官プロジェクトである。大学の研究成果を産業界に展開して、得られたデータから大学が新たな知見を発見してフィードバックするエンピリカルサイクルを確立していきたい。

2004年10月、独立行政法人情報処理推進機構 (IPA) にソフトウェアエンジニアリングセンター (SEC) が設置された。主な活動は、定量データ分析による統計、見積手法の標準化、開発プロセスの共有化である。本プロジェクトの成果をSECで活用できるように連携していきたい。

最後に、本稿執筆にあたり協力いただいたEASEラボの神谷氏、阪井氏、小野氏、岩村氏に深く感謝いたします。

参考文献

- 1) (社)情報サービス産業協会 (JISA) 「2003年受注ソフトウェアの技術課題にかかわるアンケート調査」
- 2) (社)情報サービス産業協会 (JISA) 「2004年版情報サービス産業基本統計調査」
- 3) 松原友夫、ソフトウェア業界にもデフレがやってくる、情報処理、Vol.44, No4 (2003-4)

- 4) V. Basili, The Experimental Software Engineering Groupe: A Perspective, ICSE '00 award presentation, Limerick, Ireland, June 5-10, 2000
- 5) V. Zelkowitz, D. R. Wallace, Experimental Models for Validating Technology, IEEE Computer, pp.23-31, May 1998.6) 柿元健、角田雅照、大杉直樹、門田暁人、松本健一、協調フィルタリングに基づく工数見積りのロバスト性評価、ソフトウェア工学の基礎XI、日本ソフトウェア科学会 FOSE2004, pp.73-84, Nov. 2004.
- 7) 横森 励士、梅森 文彰、西 秀雄、山本 哲男、松下 誠、楠本 真二、井上 克郎、Javaソフトウェア部品検索システム SPARS-J電子情報通信学会論文誌、Vol. J87-D-I, No.12, pp1060-1068 (2004)
- 8) 神谷 年洋、コードクローンとは、コードクローンが引き起こす問題、その対策の現状、電子情報通信学会誌、Vol.87,

執筆者紹介



井上 克郎
大阪大学
大学院情報科学研究科
コンピュータサイエンス専攻 教授



松本 健一
奈良先端科学技術大学院大学
情報科学研究科 教授



津田 道夫
1970年入社
生産技術部 主管技師長
現在、ソフトウェア生産技術に従事
奈良先端科学技術大学院大学 非常勤講師
E-mail: m-tsuda@hitachi-system.co.jp



新海 平
2001年入社:
研究開発センタ
現在、EASEの開発に従事
E-mail: tshinkai@empirical.jp