



Title	学と産の連携による基盤ソフトウェアの先進的開発 : 5. 産官学連携によるエンピリカルソフトウェア工学の実践データに基づく実証的アプローチ
Author(s)	松本, 健一; 井上, 克郎; 鶴保, 征城 他
Citation	情報処理. 2008, 49(11), p. 1257-1264
Version Type	VoR
URL	https://hdl.handle.net/11094/50177
rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

産官学連携による エンピリカルソフトウェア工学の実践 データに基づく実証的アプローチ

松本 健一*¹
鶴保 征城*³

井上 克郎*²
鳥居 宏次*⁴

*1 奈良先端科学技術大学院大学情報科学研究科

*2 大阪大学大学院情報科学研究科

*3 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/
高知工科大学情報システム工学科

*4 奈良先端科学技術大学院大学

背景

ソフトウェアは、現代経済社会の基盤として、あらゆる産業を支え、かつ、製品やサービスの付加価値を実現している。また、ソフトウェアが社会の隅々にまで行きわたり、その社会的重要性が高まるにつれ、ソフトウェアの欠陥（バグ）が引き起こすシステム故障（障害）の影響は、多くの国民にとって非常に大きく、そして、身近なものとなってきている。

一方で、開発当初に計画されたコストと期間の範囲で、要求された機能と品質を持つソフトウェアを開発できる割合は低く、国内外共に30%前後にとどまっている。ソフトウェア開発の多くは、「プロジェクト」と称され、PMBOKで指摘されているように「何らかの点で他と異なる（個性性が高い）」ものである。開発中に起きることをすべて予見することは容易ではなく、綿密な計画とともに、計画からの逸脱や想定外の事態を検知し、それに対処する手段を持たなければ、プロジェクトの成功は望めない。

特に近年のソフトウェア開発では、ソフトウェア機能と開発プロセス双方の細分化と複雑化が進んでいる。既製ソフトウェアを組み合わせ、複数組織で分担開発されたシステムで障害が発生した場合、その原因を突き止めるためのコストはこれまで以上に膨大になる。オフショア開発の活発化が、こうした問題の拡大にさらに拍車をかけている。

ソフトウェア開発におけるこうした流れは、ソフトウェア発注者（ユーザ）にも変化をもたらしている。ベンダ選定の基準は「プロジェクト遂行能力」に移りつつあるとされている。開発体制が明確であるか、品質管理・生産管理ツールを使用しているか、過去にどのような生産性向上の施策を準備・実践してきたか、などが重視されるようになってきている。

学問分野においても、従来のソフトウェア工学と比較して、ソフトウェア開発に関する実証データや実績

データがより重視されるようになってきており、「エンピリカルソフトウェア工学（Empirical Software Engineering）」と呼ばれている。ソフトウェア工学分野で最も権威ある国際会議の1つ「ソフトウェア工学国際会議（International Conference on Software Engineering）」でも、エンピリカルソフトウェア工学は投稿論文テーマの第4位となっている（2008年度大会データ）。なお、1位はprogram testing and analysis、2位はtools、3位はsoftware architecture and designである。

このように、今日のソフトウェア開発では、ベンダにとってもユーザにとっても、そして、ソフトウェア工学という学問分野においても、開発プロジェクトを組織的かつ定量的に捉え、分析し、生産性や品質の向上に役立てる仕組みや技術の開発と実践がますます重要となってきている。本稿では、エンピリカルソフトウェア工学の基本的な枠組みを説明するとともに、その実践を目指した文部科学省EASEプロジェクトの活動と産官学連携事例を3件紹介する。

エンピリカルソフトウェア工学

ロングマン現代アメリカ英語辞典には“empirical : based on scientific testing or practical experience, not on ideas”とある。ここではソフトウェア開発に限定して、

Empirical（エンピリカル）＝

Experimental（大学などでのソフトウェア開発実験に基づく）

+ Experienced（ソフトウェア開発現場での経験や実績に基づく）

とする。日本語で表記するときには、便宜的に「実証的」や「データに基づく」とすることもある。意識するならば、「観察や実験を通じて技術を検証する」や「理論だけでなく実践経験に基づいて技術を評価する」といった

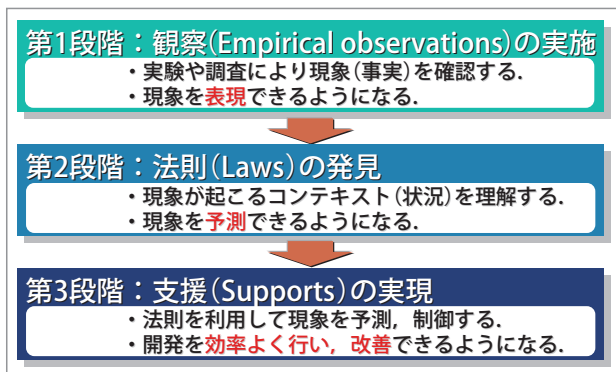


図-1 エンピリカルアプローチの3段階

ところである。

エンピリカルアプローチには3つの段階がある(図-1参照)。最初の段階は「観察の実施」である。これは文字通り、ソフトウェア開発がどのように行われているのか、硬い表現で言えば「現象」を実験や調査によって確認することである。次の段階が「法則の発見」である。観察を重ねることで、ある現象が発生するための条件は何か、その現象がどれぐらいの頻度で発生するのか、といった傾向なりパターンなりが分かってくる。すなわち、現象が起こる「コンテキスト」がはっきりしてくる。コンテキストが分かれば、その出現を捉えることで、現象の予測ができるようになる。産学連携事例として後述する「関連ルールの発見」などは文字通り法則の発見にあたるが、ソフトウェアの生産性や品質の標準値や分散といったリファレンスデータの構築、プロジェクト特性間の関係やそれら特性が生産性や品質へ与える影響の大きさ(寄与率)を明らかにする要因分析なども、ソフトウェア開発のコンテキストを知る手段となる。最後の段階が「支援の実現」である。第2段階において発見された法則を利用して現象を予測し、必要に応じて制御することで、開発作業の効率化や改善を実現する。具体的な支援内容としては次のようなものがある。

- ・プロジェクトの工数、日程、品質を高精度で見積もる。
- ・進行中のプロジェクトの品質、コスト、納期をコントロールする。
- ・進行中のトラブルプロジェクトを再計画する。
- ・組織内の全プロジェクトに対して、リソース割り当ての全体計画を作成する。
- ・開発プロセスがどの程度改善されたかをモニタする。

先述の通り、実証データや実績データをより重視するということから、エンピリカルソフトウェア工学研究において産官学連携は自然な流れである。図-2に産学連携の例を示す。たとえば、産において、対象ソフトウェアとその開発コンテキストを観測し、観測データを学において分析することで、対象ソフトウェアとその開発コ

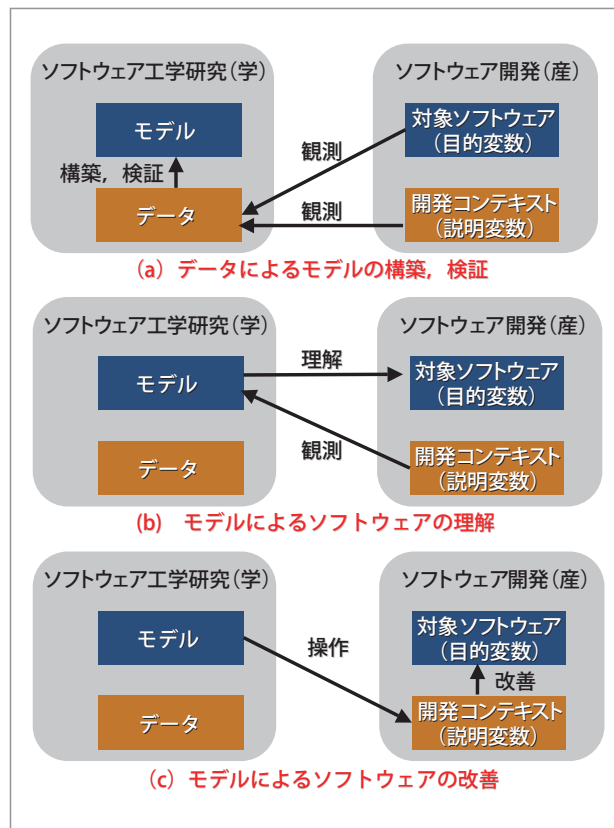


図-2 エンピリカルソフトウェア工学における産学連携

ンテキストの間の関係を表すモデルを構築、あるいは、検証することができる(図-2(a))。そして、学において検証されたモデルを利用すれば、開発コンテキストを知ること(入力パラメータとしてモデルに与えることで)、対象ソフトウェアをより詳細に理解することができる(図-2(b))。さらに、主要開発言語、アーキテクチャ、平均要員数といった開発コンテキストを操作することで、対象ソフトウェアの生産性や品質の改善を図ることができる(図-2(c))。図-2は、産と学が1対1で連携する場合を表しているが、官がリエゾン役として産と学の間に入ること、組織横断的な連携を実現することができる。

経済産業省・産業構造審議会・情報経済分科会・情報サービス・ソフトウェア小委員会が2006年9月に発表した報告書「情報サービス・ソフトウェア産業維新～魅力ある情報サービス・ソフトウェア産業の実現に向けて～」にも、

「ソフトウェアの信頼性・生産性の向上という観点からは、プロジェクトの計画、実装、試験、運用段階における定量的な運営手法などのソフトウェアエンジニアリングに関する実証的な研究を産学官が連携して進めるとともに、その普及に努めることが重要である」

「新しい手法の現場への導入を促進するためには、

5. 産官学連携によるエンピリカルソフトウェア工学の実践

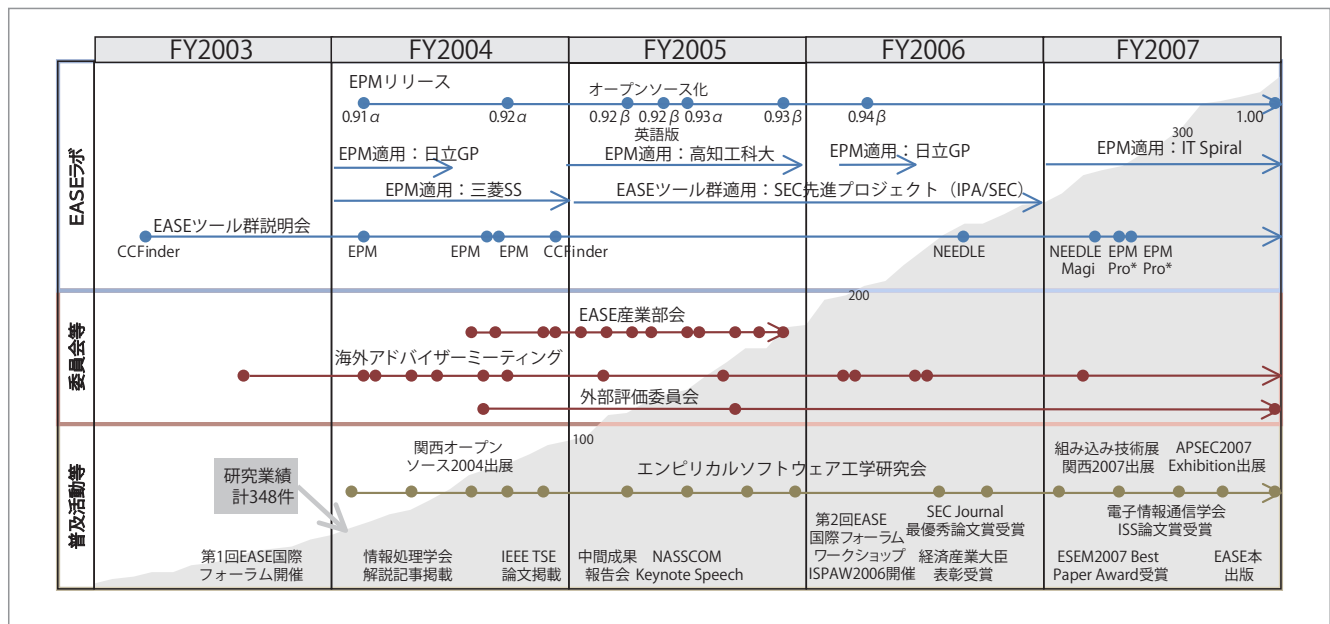


図-3 EASE プロジェクト年表

理論やガイドラインのみでは十分ではなく、実際の現場へ適用して成功例を示すことや、現場からのフィードバックを得て、さらに理論等を改善していくことが重要である」

とある。「定量的な運営手法」「実証的な研究」「成功例を示す」「現場からのフィードバック」といった表現は、まさにソフトウェア工学におけるエンピリカルなアプローチに属するものであり、産学連携の重要性を説くものでもある。

エンピリカルなアプローチは、ソフトウェア工学の従来の枠組みには収まらないものというよりは、産官学連携の下に従来の枠組みを補強し研究開発を促進するものと言える。たとえば、研究者によって新しい技術が提案されても、それがソフトウェアの生産性や品質の向上にどのようにつながるのか、客観的なデータに基づく科学的な評価結果が示されなければ、その技術を適用することに、ソフトウェア開発の現場は躊躇することになる。逆に、きちんとした科学的な評価がなされれば、その技術はソフトウェア開発の現場に積極的に導入されるようになる。同様に、ソフトウェア開発者の直感や開発現場に蓄積されているノウハウも、客観的なデータの裏付けがあれば、研究の対象となり、より一般的で適用範囲の広いモデルや手法の開発につながる。

EASE プロジェクト

EASE (Empirical Approach to Software Engineering) プロジェクトは、文部科学省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」の一環として、奈良先端科学技術大学院大学と大阪大学が2003年度から2007年度までの5年間実施した研究プロジェクトである。EASEプロジェクトの目標は、ソフトウェア開発の分野において、計測、定量化、評価、およびフィードバックによる改善という手法を実践し、エンピリカルソフトウェア工学の成果をソフトウェアツールとして具体化すると同時に、産業界と学界が連携してエンピリカルアプローチの普及とさらなる発展とを後押しすることにあった。

【エンピリカルソフトウェア工学ラボ】

EASEプロジェクトの中核は、大阪の千里中央に設置した「エンピリカルソフトウェア工学ラボ（以下、EASEラボと略す）」である。協力企業4社（日立公共システムエンジニアリング、日立製作所、NTTソフトウェア、SRA先端技術研究所）のソフトウェア技術者と大学のポストドク研究員が同居する場であり、工学の原点に立ち返り、地道にソフトウェア開発データを集め、その評価や分析を通じて改善案を提案し開発現場に適用するという、図-2で示したような産学連携サイクルにおける橋渡し役でもあった。EASEラボの活動を中心に、産官学連携や国際連携、研究業績など、プロジェクト5年間の活動概要を図-3に示す。

プロジェクト前半、EASEラボは、主に、ソフトウェア開発における自動的なデータ収集と分析のためのプラットフォームEPM (Empirical Project Monitor) を開発する場であった。一般のソフトウェア開発と異なりEPMの業務要件は明確ではなかったため、フィージビリティ検討やプロトタイピングを中心に進められた。そ

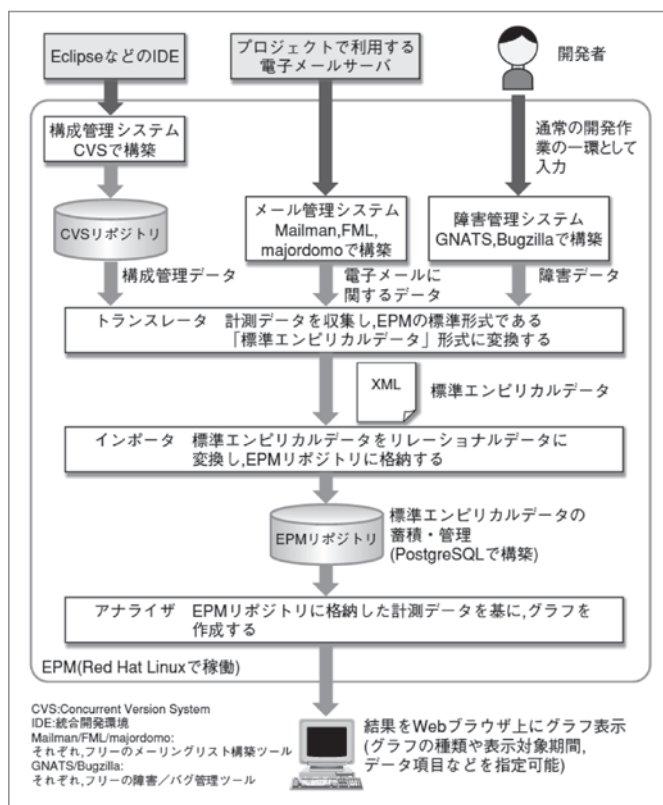


図-4 EPM (Empirical Project Monitor)

の結果、EPMでは、ソフトウェア開発者の負担を小さくするため、ソフトウェア開発プロジェクトで広く使用されている、構成管理システム、メール管理システム、障害管理システムからデータを収集する方式となった（図-4）。主な収集データは、構成管理システムからは、

ファイル参照・変更・削除・移動操作情報、対象ファイル／ディレクトリ名、メール管理システムからは、メールの送受信者名、送信時刻、件名、本文行数、障害管理システムからは、障害表の変更履歴、障害の概要・優先度・重要度、などである。

プロジェクト後半、EASE ラボは、分析手法の洗練、データ解釈、技術開発のための議論の場となった。その成果の1つが、ソフトウェア開発におけるエンピリカルアプローチを支援するための15個のツール「EASE ツール群」である。EASE ツール群の名称と機能の概要を表-1に示す。ツールそれぞれの具体的な機能や適用事例の紹介、また、いくつかについてはその実行形式が、EASE プロジェクト Web ページで公開されているので参照されたい⁶⁾。

EASE ツール群の1つ Project Replayer は、EPM 等で収集されたソフトウェア開発データを基に、ソフトウェア開発プロジェクトを再現し、ビデオプレイヤーのような感覚でプロジェクトの流れを分析できるツールである（図-5）。5つのビュー（イベントリストビュー、グラフビュー、メンバビュー、ファイルビュー、メールビュー）と、再生日時を示すタイムバーから構成され、開発者自身はもちろんのこと、研究者も分析対象のプロジェクトをより詳細に、素早く理解することができる。大学におけるケーススタディでは、ソフトウェア開発プロジェクトにまったくかかわっていない学生であっても、開発が滞っていた期間の特定やその理由の把握など、開発プロセスの問題点を容易に認識できることが確認さ

	名称	機能概要
プロセス計測・分析	EPM (Empirical Project Monitor)	ソフトウェア開発データの自動収集・分析環境で、開発支援ツールの履歴を収集、分析する。
	EPM Pro*	EPMのトランスレータによって作成されたXMLファイルを入力として、開発プロジェクトの分析・可視化を行う。
	EP-Cluster Suite	ソフトウェア開発データの定性的分析補助のために、自然語で記述された文書のクラスタリングを行う。
	MPA (Micro Process Analysis)-Plot/type 1	ソフトウェア開発データから抽出された細粒度レベルの作業系列を図示することで、マイクロプロセス分析を支援する。
	Project Replayer	EPM等で収集されたソフトウェア開発データを基に、ソフトウェア開発プロジェクトを再現する。
	EASE.CORE	ソフトウェア開発データを開発者個人レベルで収集、分析する。
プロダクト計測・分析	MUDABlue	潜在的意味解析手法を用いて、ソースコードのカテゴリ集合を作成し、ソフトウェアの分類を自動的に行う。
	Stigmata for Web	Javaクラスファイルからバースマークを抽出し、比較する。
	CHAN (Clone History ANalyzer)	版管理システムに登録されているソースコードの情報から、コードクローンの履歴分析を行う。
	D-CCFinder	80台のコンピュータを用いて分散処理型コードクローン検出を行う。
	SHINOBI	ソフトウェア保守時にコードクローンを自動的に検出する。
プロジェクト分析	Magi	EASE協調フィルタリング法を利用して、ソフトウェアの開発規模や工数の見積もりを行う。
	NEEDLE	ソフトウェア開発データに対して、相関ルールマイニングを行う。
開発管理支援	EPDG2 (Electric Process Data Guidebook 2)	定量的管理を取り入れたソフトウェア開発計画の立案を支援する。
	D-SNS (Dynamic Social Networking System)	EPM等で収集、蓄積されたソフトウェア開発データに基づいて、開発者が必要としている知識を有する人物を推薦し、知識協創を支援する。

表-1 EASE ツール群

5. 産官学連携によるエンピリカルソフトウェア工学の実践

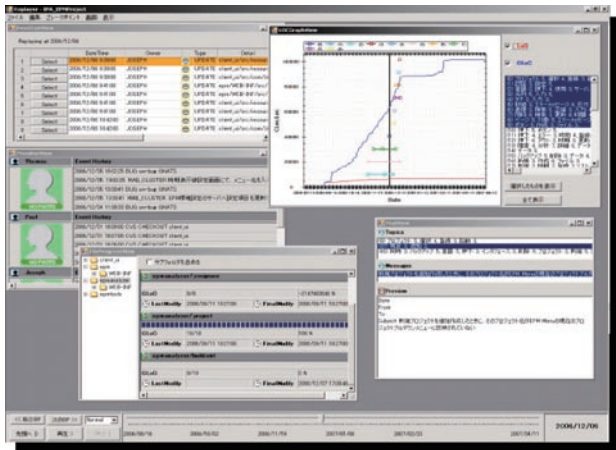


図-5 プロセス再現・分析ツール Project Replayer

れている。

Magi は、EASE 協調フィルタリング法を利用した、ワンクリック見積もり & データ診断ツールである(図-6)。情報検索技術の1つである協調フィルタリング法を応用することで、見積もりに利用するデータに大量の欠損値が含まれていても、高い精度での見積もりが可能となっている。見積もりに利用するデータがあらかじめ定められているわけではないが、代表的なものとしては、開発期間、使用言語、使用アーキテクチャ、ファンクションポイント、欠陥数などがある。レビューで発見された欠陥数などから試験工数を見積もるケーススタディでは、データ欠損率が約 60% という悪条件にもかかわらず、相対誤差平均が従来法の 2211% に対して 79% となった。Magi は、見積もりに利用するデータの量や質を診断する機能も備えており、定量データに基づく見積もりに不慣れなユーザでも、見積もり結果の解釈や利用が容易となっている。

【EASE における産官学連携活動】

● IPA/SEC との連携

EASE プロジェクトにおける産官学連携の代表例の1つが、2005 年度から実施した、独立行政法人 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 (IPA/SEC) との連携である。IPA/SEC が、経済産業省の指導の下に 2005～06 年度に実施した「SEC 先進プロジェクト」では、EASE プロジェクトが、データ収集・分析手法や EPM 等のツールを同プロジェクトに提供し、プロジェクトメンバによる人的支援も行うことで、エンピリカル環境の大規模な適用実験が実施された。

IPA/SEC との連携は、EASE プロジェクトで開発された手法やツールの有用性を示す場が得られたというだけでなく、産業界への普及という点からも、大きな成果を挙げたと言える。活動の一部は、連携事例として後述

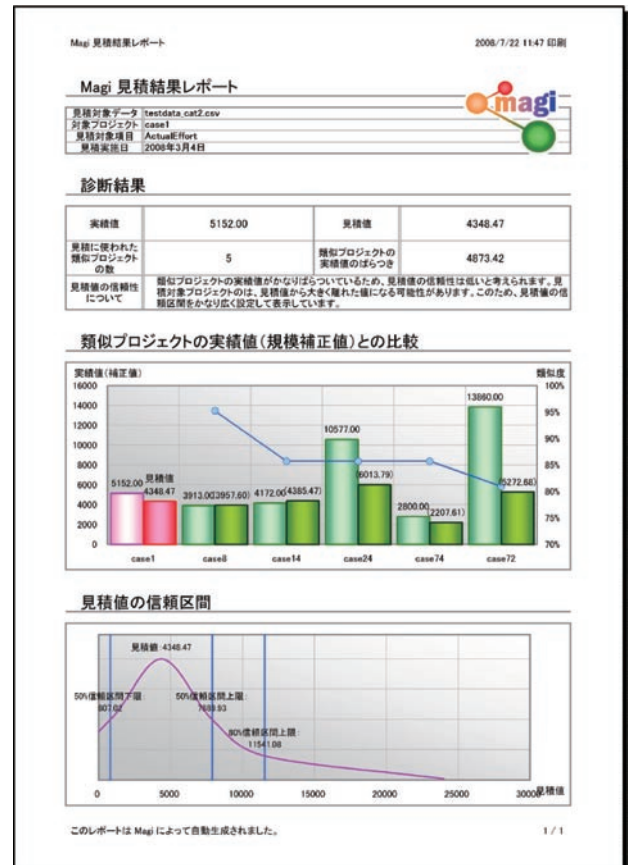


図-6 ワンクリック見積もり & データ診断ツール Magi

する。より詳しくは、文献 2) を参照されたい。

●エンピリカルソフトウェア工学研究会

エンピリカルソフトウェア工学研究会は、2004 年 4 月の第 1 回にはじまり 2008 年 3 月までに計 16 回開催され、毎回 20～50 社の企業の 50～100 名の参加者を集めた。当初は、EASE プロジェクトからの活動報告が中心であったが、IPA/SEC や企業との連携や共同研究がスタートすると、EASE ツール群の適用事例が、連携企業から多数報告されるようになった。適用事例に基づく具体的な議論は、ソフトウェア開発におけるエンピリカルアプローチの有効性を参加者に強く印象付けるものであった。

ほかにも、国際会議の参加報告など海外の研究動向を随時紹介し、エンピリカルソフトウェア工学に対する関心の高さやその重要性を産業界に訴えてきた。参加者へのアンケートや意見交換会を通じて、人的ネットワークの構築にも努めた。本研究会がトリガとなり共同研究開発に発展したケースも多数あり、産学の距離を縮めた先駆的な研究会として、産業界から高く評価されている。

●EASE 産業部会

EASE 産業部会は、EASE プロジェクトのより広範な

普及を目的として、SIベンダのソフトウェア技術者をメンバーとする組織である。参加企業は、NTT データ、NTT ソフトウェア、住商情報システム、オービス総研、日本電子計算、JFE システムズ、SRA 先端技術研究所、日立システムアンドサービスなどである。

EASE プロジェクトが提案する手法やツールの多くは、産業界のニーズに基づいて開発が始められたものである。しかし、学による検討や具体化の過程を経るにつれ、その狙いや効果がソフトウェア開発現場の技術者からは分かりづらいものとなる場合があった。EASE 産業部会は、先進的な企業の事例から成功要件を策定することなどにより、EASE プロジェクトにおける産の視点の維持に大きく寄与した。

連携事例

ここまで、エンピリカルソフトウェア工学の基本的な枠組みや産官学連携の現状を EASE プロジェクトの活動を中心に述べてきたが、最後に、具体的な産官学連携事例を 3 件紹介する。

【マルチベンダ開発における障害修正工数の要因分析】

最初の例は、先に紹介した SEC 先進プロジェクトにおける障害修正工数の要因分析である³⁾。開発プロジェクトは物理的に離れた計 7 拠点で実施され、プロジェクト発足当初から EASE プロジェクトと IPA/SEC それぞれの研究員がプロジェクト定例会議に参加し、データ収集体制を構築するとともにデータ分析を担当した。

データ収集は、設計工程から総合試験までを対象とし、開発で利用される構成管理ツールと障害管理ツールを介して行われ、分析には EASE プロジェクトで開発された EPM と EPM Pro* が用いられた (図-7)。観察された主な現象は次の通りである。

- 障害の発見が遅れると修正工数が極端に大きくなる。
総合試験での平均障害修正工数は、コーディング／単体試験でのその約 5 倍であった。特に、詳細設計工程で混入した障害はコーディング／単体試験までに、また、コーディング工程で混入した障害は結合試験工程までに、それぞれ発見されると、平均修正工数が小さく抑えられることが分かった。
- 障害がプログラム中に滞留する時間が長くなると修正工数は大きくなる。
障害の混入工程と発見工程の差が 2 工程となると、平均修正工数はそれ以下の場合の約 4 倍となった。

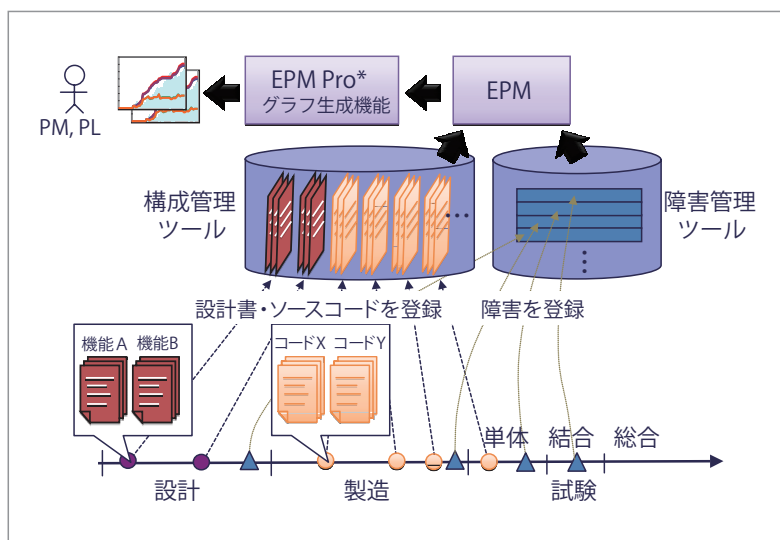


図-7 SEC 先進プロジェクトにおけるデータ収集・分析

- 障害の再現度や重要度も修正工数に影響する。

障害の再現度が低いと平均修正工数は約 2 倍となった。重要度が「軽微」な障害については、平均修正工数が半分以下となった。

観察された現象のいくつかは、Boehm らが 1970 年代に実施した実験の結果¹⁾と同様の傾向を示すものであるが、障害単位での修正工数に基づいた分析、障害の混入工程との関係や再現度、重要度、発見遅延の原因にまで言及した事例は少ない。開発作業や障害対応の遅延を検出し、進行中のプロジェクトをコントロールするといった応用が期待されている。

【システム障害対応データの傾向分析】

2 つ目の事例は、EASE ラボの協力企業の 1 つ日立製作所／日立システムアンドサービスとの連携によるものである⁵⁾。システム障害対応に見られる傾向を相関ルールとして抽出したものである。

障害対応データの分析は、平均対応日数や平均対応工数など、障害対応全体として見られる傾向に関するものと、特定の障害属性における傾向に関するものに大別できる。障害対応の改善を考える上では、後者がより有用であるが、単にデータを眺めているだけでは分析の手がかりが得られない。この事例では、相関ルール分析技術を用いることで、特定の障害属性における傾向分析を実現している。

分析対象は、日立システムアンドサービスで 1999 年から 2006 年までに蓄積された 601 件の障害対応データである。障害対応ごとに、発生日、対策日、対策所要日数、顧客名、システム、バージョン、現象コード、重要度、開発部署名、回答内訳コード、等が記録されている。相関ルールの抽出には、EASE プロジェクトで開発した

5. 産官学連携によるエンピリカルソフトウェア工学の実践

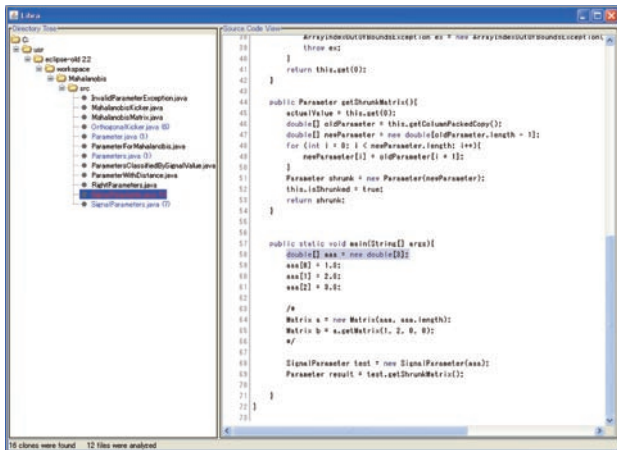


図-8 コードクローン検索ツールLibra

関連ルールマイニングツール NEEDLE が用いられた。

NEEDLE は「項目定義作成機能」「前処理機能」「ルール抽出機能」「例外ルール抽出機能」「ルールフィルタ機能」から構成される。項目定義作成機能は、対象データを量的（数値）、質的（カテゴリ）に自動的に判断するものである。前処理機能によって、量的データを離散化した上で、ルール抽出機能により、対象データに含まれる規則性を「 $A \rightarrow B$ （ A ならば B ）」の形式で列挙する。出現頻度の下限値を与えることで、列挙されるルールを絞り込むことも可能である。例外ルール抽出機能は、常識ルール（出現頻度の高いルール）を少し変形することで得られる減多に起こらないルール（例外ルール）を抽出する機能である。

最低支持度（対象データが関連ルールを満たす割合）を 3% として関連ルールを抽出し、ルール結論部に、「重要度」か「対策所要日数」を含むものについて、障害対応データに詳しい技術者と検討した結果、次のようなルールが、障害対応の改善に有用であるとされた（部署名や顧客名は、固有名詞に変わり A 、 X 等で表記する）。

- 部署 A で開発され、顧客 X に納入されたシステム P で重要度“中”の障害が多い。
- 部署 B で重要度の大きな障害が多い。
- 部署 C で設計されたシステムでの発生障害は 3 日以内に対策される傾向にあるが、部署 D で設計されたシステムでの発生障害には対策が長期化するものがある。

【コードクローン検索による類似不具合検出】

最後の事例は、奈良先端科学技術大学院大学と大阪大学、そして、パナソニック MSE の三者の連携によるものである⁴⁾。不具合（バグ）修正前のソースコード片を検索キーとしたコードクローン検索による類似不具合検出の有用性を実証的に評価した。最初の 2 つの事例が、

ソフトウェア開発の「支援の実現」に向けた「観察の実施」や「法則の発見」であるのに対し、最後のこの事例は、「支援の実現」に向けたものではあるが、コードクローンという大学で開発された技術が、類似不具合発見という支援に役立つかどうかを確かめようとしたものである。

コードクローン検索による類似不具合発見の有用性は、オープンソースソフトウェアを対象とした研究により確認されていた。この事例では、商用開発のソフトウェアを対象とし、評価における効率性（適合率）と網羅性（再現率）を示し、商用開発の現場への適用の指針となることを目指した。対象はパナソニック MSE において開発・保守されている商用システムのうち、OS とプログラミング言語の組合せが異なるソースコード 3 件で、試験工程での不具合修正履歴が記録されたりリリース済みのものである。修正履歴から不具合修正前のソースコード片を取り出し、取り出したソースコード片を検索キーとしてコードクローン検索を実施し、検索結果と修正履歴から類似の不具合がどの程度検出できているかを実証的に評価した。コードクローン検索には、CCFinder とその GUI ラッパーである Libra を用いた（図-8）。

評価の結果、コードクロンの検索結果数が 10 件未満となるようなソースコード片を対象とした 3 つの事例では、類似不具合検出の適合率は平均で 78.6%、同じく再現率は平均で 87.4% となった。また、検索結果数を限定しない別の事例では、適合率 37.2%、再現率 89.3% となった。特に、検索キーとなるソースコード片のトークン数が大きな場合に大きな適合率が得られることが分かった。

コードクローン検索にかかる人的、時間的コストは非常に小さく、網羅的な検索が可能である。システムテストや運用中に緊急で類似不具合を探さなければならない場合、派生ソフトウェア製品に対して横断的な類似不具合を発見しなければならない場合、などにおいて有用と考えられる。

活発化するエンピリカルアプローチ

EASE ラボに参加した協力企業のうち NTT ソフトウェアでは、EASE ラボでの活動を受けて、2006 年春から EPM の全社導入を開始している。また、EPM をはじめとして EASE プロジェクトが開発した手法やツールが適用された「SEC 先進プロジェクト」の参加企業は、ソフトウェア開発におけるエンピリカルアプローチの有用性を確信し、その後、自らの資源を投入し、新たなソフトウェアエンジニアリング施策へ歩を進めている。

IPA/SEC 自体でも、SEC 先進プロジェクトで用いたエンピリカル環境を幅広く普及しその成果を再現可能と

特集 学と産の連携による基盤ソフトウェアの先進的開発

するため、その機能・性能拡充と配布キット化を図る「EPM ツール検証プロジェクト」が推進されている。現在 40 社余りの企業が同検証プロジェクトに参加し、ツールの適用や評価が進行中である。また、組込みソフトウェア分野における新たな実証プロジェクトも進行中である。中間法人 JasPar による自動車制御用基盤ソフトウェア開発において、EPM 等によるプロジェクト計測が実施計画に組み込まれている。さらに、こうした活動や成果を受け、2つの業界団体、情報サービス産業協会 (JISA) と Information Technology Alliance (ITA) では、専門の部会が組織され、EPM の実適用作業や相互啓発活動が活発に進められている。

産における実証的アプローチの活発化に合わせるように、学においても実証性を重視した研究は盛んになりつつある。エンピリカルソフトウェア工学に関する国際会議 International Conference on Empirical Software Engineering and Measurement (ESEM) は、2002 年に奈良で第 1 回が開催された International Symposium on Empirical Software Engineering を前身とするものであるが、2007 年 9 月にスペインのマドリッドで開催された大会では、発表論文 44 編のうち 5 編は日本からのものであり、うち 1 編が最優秀論文賞を受賞するなど、日本勢も健闘している。

官においても、ソフトウェア開発における実証的アプローチの重要性や有用性は強く認識されており、2007 年 8 月には、文部科学省次世代 IT 基盤構築のための研究開発「ソフトウェア構築状況の可視化技術の普及：エンピリカルデータに基づくソフトウェアタグ技術の開発と普及」がスタートした⁷⁾。Stage プロジェクトと名付けられたこの研究開発では、ソフトウェア開発データを「ソフトウェアタグ」としてソフトウェア製品に添付することで、ソフトウェアの発注者やエンドユーザにもオープンにする。さらに、オフショア開発や法的問題への対応という新しい視点を加えることで、ソフトウェア

の生産性と品質に関する実証的アプローチの適用範囲を広げ、エンピリカルソフトウェア工学のより先進的な実践が進められている。

参考文献

- 1) Boehm, B. W. : Software Engineering, IEEE Trans. on Computers, Vol.25, No.12, pp.1226-1241 (1976).
- 2) 情報処理推進機構ソフトウェア・エンジニアリング・センター編著：ソフトウェアエンジニアリングの実践 先進ソフトウェア開発プロジェクトの記録、翔泳社(2007)。
- 3) 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 (May 2007).
- 4) 森崎修司, 吉田則裕, 肥後芳樹, 楠本真二, 井上克郎, 佐々木健介：コードクローン検索による類似不具合検出の実証的評価, 電子情報通信学会論文誌, Vol.J91-D, No.10, pp.2466-2477 (Oct. 2008).
- 5) 十九川博幸, 森崎修司, 松村知子, 門田暁人, 松本健一：相関ルールを用いたシステム障害対応データの傾向分析, 情報処理学会第 70 回全国大会 (Mar. 2008).
- 6) <http://www.empirical.jp>
- 7) <http://www.stage-project.jp>

(平成 20 年 7 月 31 日受付)

松本 健一(正会員) matumoto@is.naist.jp

平成元年大阪大学基礎工学部情報工学科博士課程中退。同年同大助手。平成 5 年奈良先端科学技術大学院大学助教授。平成 13 年同大教授。工学博士。文部科学省 Stage プロジェクト研究代表者。電子情報通信学会、日本ソフトウェア科学会、ACM 各会員。IEEE Senior Member。

井上 克郎(正会員) inoue@ist.osaka-u.ac.jp

昭和 59 年大阪大学基礎工学部情報工学科博士課程修了。同年同学科助手。昭和 59 ～ 61 年ハワイ大マノア校情報工学科助教授。平成元年大阪大学基礎工学部情報工学科講師。平成 3 年同学科助教授。平成 7 年同学科教授。平成 14 年同大情報工学科コンピュータサイエンス教授。工学博士。ソフトウェア工学の研究に従事。日本ソフトウェア科学会、IEEE、ACM 各会員。

鶴保 征城(名誉会員) tsuruho@ipa.go.jp

工学博士。昭和 41 年大阪大学修士課程修了。NTT (株)、(株) NTT データを経て、NTT ソフトウェア(株)代表取締役社長。平成 15 年より高知工科大学教授。平成 16 年より独立行政法人情報処理推進機構 (IPA) ソフトウェア・エンジニアリング・センター (SEC) 所長。

鳥居 宏次(正会員) torii@is.naist.jp

電気試験所、大阪大学、奈良先端科学技術大学院大学を歴任後、平成 13 ～ 17 年同学長。EASE プロジェクト代表。平成 20 年合同会社「EASE 創研」設立。ACM、IEEE、電子情報通信学会、本会各フェロー。