

# エンピリカルソフトウェア工学の現状と展望：SELが遺した13の教訓

Beyond the SEL: 13 Lessons Learned by SEL and EASE/SEC Approach in Empirical Software Engineering



奈良先端科学技術大学院大学教授  
工学博士

松本 健一

Nara Institute of Science and Technology, Professor, Dr. Eng.  
Kenichi Matsumoto

ソフトウェアやその開発過程から得られる定量的データに基づいてソフトウェアの生産性や品質の向上を目指す実証的アプローチ（エンピリカルアプローチ）が注目されている。本稿では、米国SEL（Software Engineering Laboratory）の研究者たちが遺した「プロセス改善におけるエンピリカルアプローチに関する13の教訓」を紹介すると共に、それら教訓と対比させる形でEASE（Empirical Approach to Software Engineering）プロジェクトの現状とSECへの期待について述べる。

This paper shows 13 lessons learned about the empirical approach to software process improvement by members of the Software Engineering Laboratory, in order to clarify the current status of the EASE project which aims to establish Empirical Software Engineering to improve software productivity and reliability in Japan. This paper also shows some expectations for SEC about sharing empirical data of software projects among software industry members.

Key Words & Phrases : エンピリカルソフトウェア工学, ソフトウェア測定, ソフトウェアメトリクス, ソフトウェアプロセス改善  
Empirical Software Engineering, Software measurement, Software metrics, Software Process Improvement

## 1 はじめに

近年、ソフトウェアやその開発過程から得られる定量的データに基づいてソフトウェアの生産性や品質の向上を目指す実証的アプローチ（エンピリカルアプローチ）が注目されている[1]。受注ソフトウェア開発を対象として社団法人情報サービス産業協会が行ったアンケート調査の結果[2]でも、ソースコード行数、ファンクションポイントなど規模に関して何らかの測定を行っている企業は85%，不具合数／密度，レビュー回数，テストケース数など品質に関して何らかの測定を行っている企業は93%にものぼる。

ただし、（苦労して）測定したデータの利用率は必ずしも高くない。同調査結果によれば、コスト見積りを測定データに基づいて行っている企業は51%に過ぎず、43%の企業では依然として、担当者の経験や類推に基づく見積りが行われている。エンピリカルアプローチをソフトウェア開発に定着させためにはまだまだ工夫と努力が

必要なようである。

本稿では、米国SEL（Software Engineering Laboratory）の研究者たちが遺した「プロセス改善におけるエンピリカルアプローチに関する13の教訓」を紹介する。次に、著者らが2003年度から推進しているEASE（Empirical Approach to Software Engineering）プロジェクトの現状を、それら13の教訓と対比させながら述べる。最後に、SECが取り得る、SELよりも高次なエンピリカルアプローチの可能性について指摘する。

## 2 SELが遺した教訓

SELは、ソフトウェアプロセス改善の研究を行うために、University of Maryland (UMD) のV. R. Basili教授によって1976年に設立された組織である。その具体的な研究目標は次のとおりである。

- ・ NASA/Goddard Space Flight Center (NASA/GSFC) におけるソフトウェア開発プロセスの理解
- ・ 様々な方法論、ツール、モデルが持つ影響力の測定

・ソフトウェア開発の成功例の収集と実際のソフトウェア開発への適用

2001年にその25年の歴史を閉じるまでの間、NASA/GSFC向けソフトウェアの開発において測定したデータに基づいて、QIP (Quality Improvement Paradigm), GQM (Goal/Question/Metric), EF (Experience Factory) といったフレームワークを提案するとともに、250編を超える学術論文を発表している。その功績が認められ、1994年には、米国IEEE (The Institute of Electrical and Electronics Engineers) とSEI (Software Engineering Institute) によるProcess Achievement Awardの最初の受賞者となっている。

ただし、研究の様々な局面で十分に取り組むことができなかつた課題があったことをSELのメンバーは認めている。彼らはこうした課題を「プロセス改善におけるエンピリカルアプローチに関する13の教訓」(図1)としてまとめている[3]。教訓は次のとおり4つのカテゴリに分類されている。なお各教訓につけられた番号はSELにおいて得られた順番を示す。

### カテゴリ1 データ収集に関する教訓

教訓1：データ収集は厳密なプロセスと専門のスタッフを必要とする。

教訓2：得られる情報の量には妥協も必要である。

教訓5：組織の成果物、プロセス、目標のベースラインを作ることは、全ての改善プログラムにおいて非常に重要である。

教訓6：データ測定の精度は常に疑わしい。その疑わしさにうまく対処し、また、その限界を理解しなければならない。

### カテゴリ2 経営陣の積極的参加に関する教訓

教訓8：研究と開発の双方に共通のコミットメントを持つことは極めて重要である。

教訓11：より上層の経営陣の支援を得ることは、プロセス改善を継続的に成功させるために重要である。

教訓12：改善のためのプロセスが組織の外にあるとプロセス改善は成功しない。

教訓13：技術中心の組織では、任務を遂行する上でソフトウェア工学がいかに重要であるのかを認識させることは難しい。

### SELの教訓

カテゴリ1 データ収集	教訓1:データ収集は厳密なプロセスと専門のスタッフを必要とする。 教訓2:得られる情報の量には妥協も必要である。 教訓5:組織の成果物、プロセス、目標のベースラインを作ることは、全ての改善プログラムにおいて非常に重要である。 教訓6:データ測定の精度は常に疑わしい。その疑わしさにうまく対処し、また、その限界を理解しなければならない。
----------------	--

カテゴリ2 経営陣の 積極的参加	教訓8:研究と開発の双方に共通のコミットメントを持つことは極めて重要である。 教訓11:より上層の経営陣の支援を得ることは、プロセス改善を継続的に成功させるために重要である。 教訓12:改善のためのプロセスが組織の外にあるとプロセス改善は成功しない。 教訓13:技術中心の組織では、任務を遂行する上でソフトウェア工学がいかに重要であるのかを認識させることは難しい。
------------------------	---

カテゴリ3 研究指針	教訓9:ソフトウェア工学の研究と実践は共生的関係にある。双方の活動は、相互作用を通じて利益を得る。 教訓10:研究者と開発者が(文字通り)近い距離にいることが双方にとって良い結果を生む。
---------------	--

カテゴリ4 継続的 スタッフ支援	教訓3:データ収集のためのスタッフトレーニングに終わりはない。 教訓4:データ収集は重要であるが、ソフトウェアの納期はさらに重要である。 教訓7:開発者に情報を迅速にフィードバックすることの必要性と、収集データを十分に分析する時間を確保することの必要性の間には、いつも緊張関係がある。
------------------------	--

### EASEプロジェクト

EPM (Empirical Project Monitor)	システム
EPDG (Electronic Process Data Guidebook)	
協調フィルタリングを用いたプロジェクトデータ予測	方法
潜在的意味解析法を用いたソフトウェア分類	
企業やSECへのボストン派遣	活動
ISERNへの参画	
エンピリカルソフトウェア工学ラボ	組織
EPM産業部会	

図1 SELの教訓とEASEプロジェクト

### カテゴリ3 研究指針に関する教訓

教訓9：ソフトウェア工学の研究と実践は共生的関係にある。双方の活動は、相互作用を通じて利益を得る。

教訓10：研究者と開発者が（文字通り）近い距離にいることが双方にとってよい結果を生む。

### カテゴリ4 繙続的スタッフ支援に関する教訓

教訓3：データ収集のためのスタッフトレーニングに終わりはない。

教訓4：データ収集は重要であるが、ソフトウェアの納期はさらに重要である。

教訓7：開発者に情報を迅速にフィードバックすることの必要性と、収集データを十分に分析する時間を確保することの必要性の間には、いつでも緊張関係がある。

## 3 教訓を踏まえたエンピリカルアプローチ

### 3.1 EASEプロジェクト

EASE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2003年から5年計画で実施されている[4]。その目標は、エンピリカルソフトウェア工学の成果をソフトウェアプロダクトとして具体化し、産学官連携のもとに、エンピリカルアプローチの普及と更なる発展を実現することにある。

プロジェクトの詳細については、プロジェクトホームページ[2]等をご参照いただくとして、ここでは、先に示した「SELが遺した13の教訓」との対応関係を示しながら、開発中の主なシステムやデータ分析方法、実施中の活動や組織について述べる（図1）。

### 3.2 システム

#### (1) EPM

EPM (Empirical Project Monitor) は、現在広く普及している開発支援フリーウェア (CVS, Mailman, GNATS 等) と連携することによって、ソフトウェア開発プロジェクトデータをリアルタイムに収集し、定量的データ分析を可能にするシステムである（図2）[6]。EPMを利用したソフトウェア開発では、プロジェクト管理者や開発者は、プロジェクトの進捗状況や作業状況を客観的に把握することが可能となる。プロジェクトの問題点を迅速に発見

することで効果的なプロセス改善も期待される。EPMの適用実験では、開発者に大きな作業負荷を与えることなく、対象プロジェクトの状況を分析可能であることが確認されている。

EPMは、構成管理、メーリングリスト管理、障害管理等、広く普及し利用されているソフトウェア開発支援システムからデータを収集する。プロジェクト管理者や開発者がデータ収集のために特別な作業を行う必要はない。データ収集が優先され納期が犠牲になるということではなく「教訓4」を踏まえたシステムとなっている。

また、EPMは導入コストが小さく、開発の環境やスタイルを大きく変更する必要もない。データ収集のためにフリーウェアを導入することに抵抗のある企業もあるであろうが、特定の企業のツールやシステムを導入することに比べれば、導入可能な企業やプロジェクトの範囲は広い。もし自社製開発管理システムなどが既に導入されているのであれば、無理にフリーウェアを導入せず、収集データの形式をEPMの標準データフォーマットに変換するなどで対応してもよい。いずれにしても、プロセスを組織内におくことを容易にする、「教訓12」を踏まえたシステムといえる。

さらに、EPMは、探索的で多くの時間を要するデータ分析作業を支援するため、システム上でユーザが実行したデータ分析の内容を保存、再実行する機能を有している。データのリアルタイム収集機能と合わせて、分析結果の迅速なフィードバックを目指すものであり、「教訓7」を踏まえたシステムといえる。

#### (2) EPDG

EPDG (Electronic Process Data Guidebook) は、定量的データに基づくプロセス管理の実施を支援するためのシ



図2 EPMにおける分析結果表示例

ステムである（図3）[7]。「測定情報モデル[8]」に基づいた諸測定量の定義や実測値を一元管理することができる。データ収集を行う者には、具体的な分析手順や測定情報モデルの構造、収集手順等についての正しい理解を与える。また、データ分析を行う者には、測定値の持つ意味と用途に対する正確な情報を提供する。

より具体的には、CMMI連続表現[9]における「プロジェクト管理のためのプロセス領域QPM（定量的プロジェクト管理）」と「QPMを支援するプロセス領域MA（測定と分析）」において要求される実践項目（プラクティス）の実現を想定している。現バージョンのEPDGを導入し積極的に活用すれば、MAにおいては「CMMIの連続表現における能力度レベル3」までの実践がすべて達成可能である。また、EPMなど、ソフトウェアプロセスの測定や分析結果のフィードバックが可能なシステムと組み合わせることで、QPMとMA双方の領域において、より高い能力レベルの達成を支援可能である。CMMIに沿った測定が行える点からも、EPDGは、「教訓1」を踏まえたシステムといえる。

EPDGは、特定の測定環境への依存を避けるため、自動的な測定や分析の機能は備えていないが、過去の測定データや具体的な分析事例を参照することができる。測

定データの精度を直接保証することは難しいものの、測定プロセスの検証や再利用を通じて、測定データの精度向上に間接的に貢献することは可能であり、「教訓6」を踏まえたシステムもある。

### 3.3 方 法

#### (1) 協調フィルタリングによるプロジェクトデータ予測

協調フィルタリングはamazon.comの書籍推薦システムなどで用いられている技術で、膨大な情報の中から有益と思われる情報を、システム利用者ごとに選び出すことができる。書籍推薦システムでは、各利用者の未購入書籍それぞれに対して、その利用者が購入した場合に付けるであろう評価点を予測し、評価点の高い書籍の購入を薦める。予測において、他の利用者が書籍に付けた評価点を利用するが、個々の利用者が評価点を付けている書籍は、システムに登録されている書籍のごく一部に過ぎない。いわば、利用できるデータ（評価点）の大部分が欠損した状況（未記入の状態）で予測を行っていることになる。

ソフトウェア開発におけるデータ収集では、データ欠損はある程度避けられない。完了した工程のデータを収集することはできないし、収集システムの一時的な不具

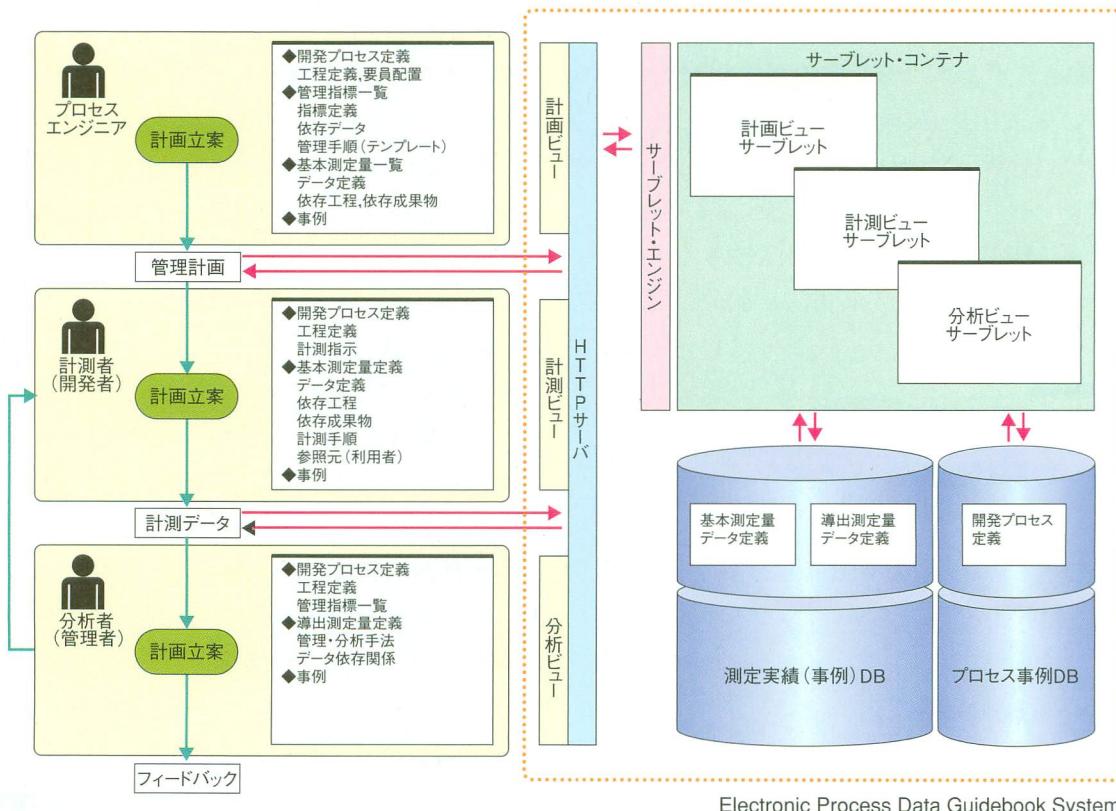


図3 EPDGシステムアーキテクチャ

合などで、ある期間のデータが抜け落ちることもある。また、組織横断的にデータを利用しようとしても、データ分析の目的や開発組織が異なれば、収集データも異なる。テスト工程を評価しようとしているプロジェクトにおいて（評価に直接必要のない）設計工程のデータまで収集する時間的、費用的余裕があるとは限らない。収集データを共通化するために、開発モデル、ツール、ドキュメントなどを全組織で共通とすることも現実的ではない。

重回帰分析やニューラルネット等の多変数モデルでは、ソフトウェアやその開発過程の特性値をできるだけ多く説明変数の候補とすることで、バグ数や開発工数といった目的変数の予測精度を高めようとする。しかし、候補とする特性値が増えれば増えるほど、それらの算出に必要なデータが全て収集されているソフトウェア開発プロジェクト数（データセット数）は少なくなる。データセット数が少なくなると、説明変数が多くてもモデルの予測精度は低くなり、適用範囲も限定される。データ欠損を補う方法も開発されているが、欠損率が30%を超えると予測精度は著しく低下する[10]。

協調フィルタリング技術を用いた予測法では、データ欠損率が70%程度までであれば、バグ数や開発工数などを比較的高い精度で予測することができる。ある企業の1,081件のソフトウェアプロジェクトで測定された14個の特性値（データ欠損率約60%）を用いて試験工数を予測したケーススタディでは、予測値の平均相対誤差やPred25の値が、従来方法（欠損値補完法を併用したステップワイズ重回帰分析）より大幅に改善されることが確認された（図4）[11]。データ欠損を許容する本手法は、得られる情報量の現実的な妥協点を見出しやすくするものであり、「教訓2」を踏まえた分析方法となっている。

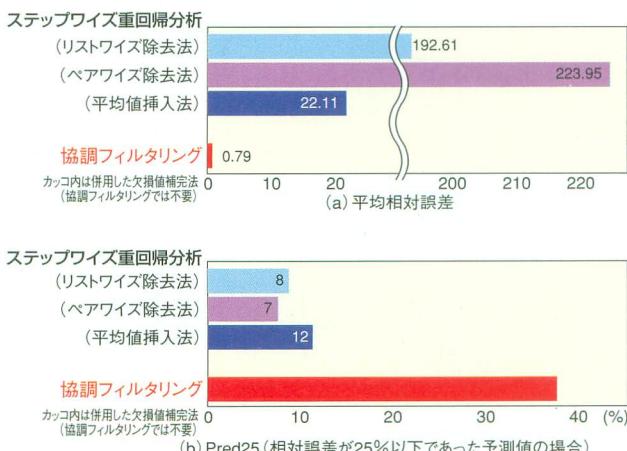


図4 欠損率60%のデータにおける予測誤差

## (2) 潜在的意味解析法を用いたソフトウェア分類

1つのモデルであらゆるソフトウェア開発プロジェクトのコスト見積りを行うことに無理があるのと同様に、ソフトウェアやその開発過程に対して唯一無二のベースラインを設定することは現実的ではない。PMBOKでも、ソフトウェア開発プロジェクトは、独自性（個別性）の高いものとされている[12]。ソフトウェアやその開発過程を特性によって分類し、カテゴリー毎にベースラインを設定し、利用する方がより現実的である。

潜在的意味解析法 LSA (Latent Semantic Analysis) とは、自然言語で書かれた文書、単語の類似度を測定する方法である。ベクトル空間モデルに従った手法の1つであるが、ベクトル空間モデルでは検出できない間接的な関連の抽出を可能にしている。ソースコードを文書に、ソースコード上の識別子（変数名、関数名、型名）を単語に、それぞれ対応させることで、ソースコード間の類似度評価、分類（クラスタリング）が可能になる[13]（図5）。

SourceForge.netで公開されている41個のソフトウェアを分類するケーススタディでは、開発環境やプログラミング言語に関する深い知識なしに、複数の観点からの分類が可能であることを確認した。ケーススタディでは、ソースコードを類似度測定の対象としているが、LSAはもともと自然言語で書かれた文書を対象としたものである。要求仕様書や設計ドキュメント、プロジェクト計画書やテスト実施計画書などへの適用も考えられる。ソフトウェア産業全体、組織単位、プロジェクト単位、プロダクト単位など、多様な粒度でのベースライン構築を可能とするものであり「教訓5」に沿った実用性の高い分析方法である。

## 3.4 活動

### (1) ポスドク派遣

EPM適用企業やエンピリカルアプローチを実践する企業などにEASEポスドクを派遣し、データ分析技術の評価を行うと共に、データ分析のニーズを把握し、新たな技術開発の目標設定などを行っている。

例えば、2004年3月から、EASEプロジェクトの中核企業の1つである日立公共システムエンジニアリング株式会社（日立GP）へのポスドク派遣を始めている。日立GPでは、2004年4月から9月まで、ある自社パッケージソフトウェアの開発プロジェクトにおいてEPMによるデータ収集を行った[5]。現在は、33項で紹介した「協調フ

イルタリング」や「潜在的意味解析法」をはじめとして、ポスドクが考案したデータ分析法を収集データに適用し、その結果の評価や解釈を、ポスドクと日立GPのプロジェクト管理者や開発者が共同して実施している。

データ収集はEPMを適用することなどにより、どの企業（組織）においてもある程度共通的に実施できる。しかし、収集データの利用目的やソフトウェア開発プロジェクトが抱える問題は、企業によって異なる。データ分析については、企業ごとに個別に検討し、時間をかけて問題解決や知識獲得を進めていく必要がある。緒に就いたばかりであるが、「教訓9」を踏まえ、ソフトウェア工学の研究と実践の共生を目指した活動を進めている。

## (2) ISERNへの参画

ISERN (International Software Engineering Research Network) は、1993年に創設されたエンピリカルソフトウェア工学の产学の研究者集団である[14]。ソフトウェアの開発／利用／管理を支援する技術について、理論面での議論だけでなく、技術の有用性を確かめる実証実験も行っている。現在は、米国、ドイツ、オーストラリア、日本など世界10カ国の37の研究機関が参加し、ソフトウェア工学分野で最も多くの学術論文を発表する組織となっている。2002年からは、著者らの提案により、产学から広く論文を募る国際会議ISESE (International Symposium on Empirical Software Engineering) を開催し、2003年からはInternational Advanced School of Empirical Software Engineeringと呼ばれる研修コースも開設されている。こうした活動は、データ収集技術の向上に資するものであり、教訓3を踏まえたものである。

## 3.5 組織

### (1) エンピリカルソフトウェア工学ラボ

大阪の千里中央に設置している「エンピリカルソフトウェア工学ラボ」は、中核企業4社のソフトウェア技術者とEASEポスドクが同居する場である。まさに教訓10を踏まえた組織である。そうした中からEPMが作り出され、ポスドクと技術者の共著による学術論文も既に発表されている[6]。

EASEプロジェクトでは、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サイクルを、できる限り科学的に実践しようとしている。これは、研究者だけではうまくいかないし、開発現場の日々の努力だけでも無理がある。従来の产学連携モデルでは、産と学の間に橋を架けることになるが、EASEでは産と学の自由な交流の場を設ける。中ノ島方式と呼ぶこのモデルで重要なことは、企業に対しては企業の言葉で対応し、学に対しては学の言葉で対応する人材の育成である。1人の人間でありながら2つの顔を持ち、必要に応じてそれぞれの言葉を使い分ける。些細なことだが、こうした人材を育てなければ、本当の产学連携などあり得ない。

### (2) EPM産業部会

EPM産業部会は、EPMのより広範な普及を促進、サポートを目的として、EASE中核企業や協力企業のソフトウェア技術者を主なメンバーとする組織である。2005年2月から本格的な活動を開始し、現在は9名のメンバーで、EASEプロジェクトおよびEPMへの産業界からの要望を集約し、EPM普及に向けた产学連携の具体的枠組みやEPMのデータ分析機能仕様などを検討する会合を毎月開

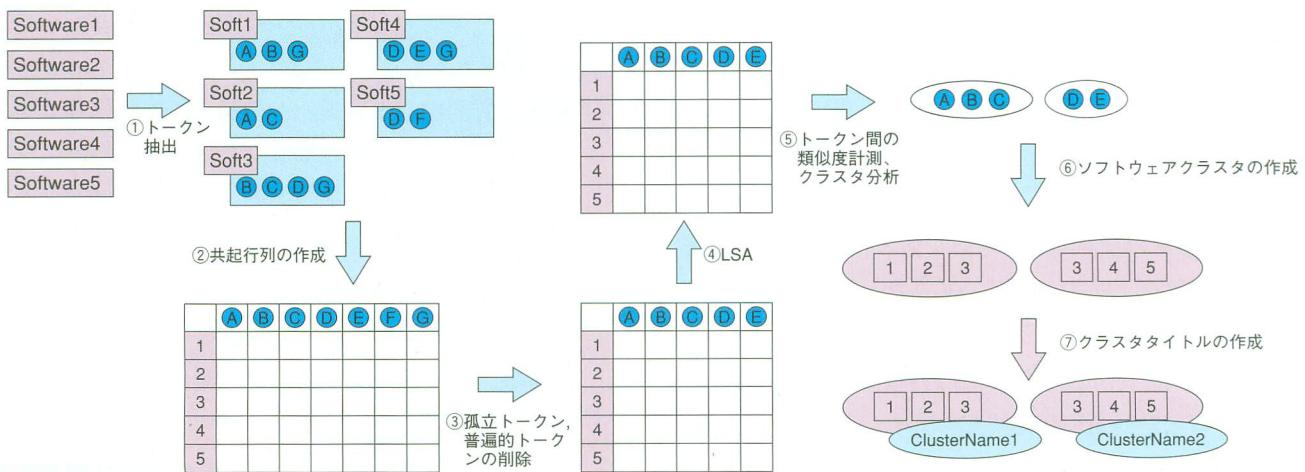


図5 潜在的意味解析法を用いたソフトウェア分類

催している。

EPMをはじめとして、EASEプロジェクトが提案するシステムや方法の多くは、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、その狙いや効果がプロジェクト管理者や開発者からはわかりづらいものとなる傾向にある。システムや方法の技術的な説明も重要であるが、教訓13を踏まえるならば、管理者や開発者から見て成功と思われる事例を具体的に示すことも必要となってくる。EPM産業部会では、成功事例の策定についても議論を進めていく予定である。

## 4 SELからSECへ

3節で示したとおり、EASEはSELの遺した教訓の多くを踏まえたプロジェクトと位置づけることができる。しかし、文部科学省のプロジェクトであることやその予算規模からすると、特に「経営陣の積極的参加に関する4つの教訓」については、十分に対応することが難しい。

これに対して、SECは、より多くの予算を有し、ソフトウェア開発会社が組織するいくつもの団体と太いパイプを持つ[15]。経営陣を巻き込み、「経営陣の積極的参加に関する4つの教訓」を踏まえた、より多様な活動也可能である。SECは、既に、多くの企業の経営陣の支持の下、1,000を超えるソフトウェア開発プロジェクトのデータを集積することに成功している。さらには、ソフトウェアエンジニアリング技術研究組合がSECの協力を得て進めている「先進ソフトウェア開発プロジェクト」における組織横断的なデータ収集にも着手している。ちなみに、SELは、25年間で130余りのプロジェクトのデータしか収集していない。

SELの教訓を単に踏まえるだけでなく、より高次なエンピリカルアプローチに挑戦することもSECには可能である。例えば、収集データや分析結果の公開や組織間での共有である。

SELはNASAの予算で運営されており、NASAにおけるプロセス改善を支援する組織であった。収集データは原則非公開で、ツール化など成果の普及策もあまり行われなかつた。実際、SELの遺した13の教訓には、収集データや分析結果の公開や組織間での共有に直接言及したものはない。これに対して、SECは、文字通り「センター」である。特定の企業のためにデータの収集、分析、蓄積

を行う組織ではない。収集データや分析結果を公開し、測定ツールの開発や提供など普及策も積極的に行い、ソフトウェア開発プロジェクトデータや知識の共有、さらには知識の再生産の中心的役割を担うことも可能である。

収集データや分析結果を公開することに抵抗のある企業は多い。しかし、オープンソースソフトウェアにおいても、ソースコードや開発過程を全てオープンにしたからといって開発に関わる重要な情報がすぐに誰かに筒抜けになるわけではない。情報を獲得し利用するためにはそれなりの知識や技術力が必要であり、明確な目的や強い動機付けがなければ参照すら行われない場合も多い。

それでも抵抗があるというのであれば、収集データや分析結果を（無理に）一箇所に集積するようなことは行わなければよい。コンピュータネットワークで結ばれた環境なら、自分たちが収集したデータや分析結果をローカルに保持することを各企業に許す一方、個人情報の保護や匿名性を確保しつつ、組織横断的にデータを集約し、分析する方式やシステムを実現することは可能である。

例えば、ソフトウェア開発プロジェクトデータの収集や分析に関する「質問回答サイト」を開設してはどうであろうか。質問回答サイトとは、サイトメンバーから発せられた質問に対して、答えを持つ別のメンバーが回答するコミュニティサイトの一種である[16][17]。インターネット上の膨大な情報の中から必要とする情報をみつけ出すには、検索エンジンとキーワード検索の組み合わせだけでは限界がある[2][18]として検索サイト自体が開設している場合も多い。

質問回答サイトを立ち上げること自体は技術的に難しくないかもしれない。ただし、質問が活発に行われ、多数の有益な回答が寄せられるためには、メンバー間のコミュニケーションを円滑にする次のようなしくみが必要である[18]。

①媒体：質問内容をメンバー全員に正しく伝達するためのメディア、データ収集用テンプレートやベースライン、さらには、収集データそのものが共通化されれば、それらが媒体となるかもしれない。

②回答コストの低減：回答コスト（回答に要する工数）が大きいと、回答したくても回答できないということになる。プロジェクト管理者や開発者が、回答のためにはわざわざ資料を作成したり整理したりするのではなく、通常の作業やそこで作り出されるプロダクトがそのまま回答や回答の資料になれば、回答コストは大幅

に抑えることができる。

③動機付け：有益な回答を確保するためには、ある程度以上の知識や経験を有するメンバーの存在が不可欠である。多くの質問回答サイトでは、一部のメンバーのみが頻繁に回答し疲弊してしまうという事態を避けるために、回答者に報酬を与える仕組みを提供している。

データの分析や知識化までも自動で行ってくれる方式やシステムが理想である。しかし、有能なプロジェクト管理者の知識の多くが暗黙知である現状からすると、様々なソフトウェアプロジェクトにおいて仮説やモデルに基づくデータ分析が可能となるまでにはもう少し時間が必要である。その実現を待って収集データを死蔵することになれば、SECも多くのソフトウェア企業の現状と同じになってしまう。産学官連携の下、様々な可能性への挑戦を期待したい。

## 5 おわりに

本稿では、米国SELの研究者たちが遺した「プロセス改善におけるエンピリカルアプローチに関する13の教訓」を紹介すると共に、それら教訓と対比させる形でEASEプロジェクトの現状とSECへの期待について述べた。

Basili教授らSELの中心メンバーは、その後、コスト見積りモデルCOCOMOの提案者として著名なB. Boehm教授らとCeBASE（Center for Empirically-Based Software Engineering）を設立し、研究を続けている[20]。自らが得た教訓を踏まえてか、CeBASEでは、SELで収集したデータの一部や新たに開発した測定ツールなどをホームページ上で公開している。名称も「センター」となり、その意味ではSECと同じであるが、NSF（National Science Foundation）がスポンサーとなっていることもあり、ホームページを見る限りでは、大学間連携が中心で、多数の企業が参加するプロジェクトとはなっていない。

エンピリカルソフトウェア工学において、多数の企業との連携が間違ったアプローチであるとは思えない。連携の具体的な形態や進め方についてはまだまだ議論が必要なのかもしれないが、今度ばかりは、CeBASEが教訓を遺してくれるのを待つ時間的余裕はないし、そもそも、多数の企業との連携についての教訓は遺してくれそうにない。ソフトウェアの生産性や品質の向上に実証データ

が必要であることは多くの人が認めるところであり、実際に多数の企業がデータ収集を行ってきており、産学官連携の下、どのような一歩を踏み出すことができるのか、その議論に本稿が役立てば幸いである。

謝辞 本稿の一部は、文部科学省「e-Society基盤ソフトウェアの総合開発」の委託に基づいて行われた研究成果に基づくものである。本稿の完成にあたり、SELの遺した教訓についてご意見をいただいたEASEプロジェクトメンバー、特に楠本真二先生、飯田元先生、門田暁人先生、大平雅雄先生、松村知子氏、森崎修司氏、大杉直樹氏に深く感謝する。また、掲載の機会をくださったSECの皆様に深くお礼を申し上げる。

### 参考文献

- [1] “特集：日本のソフト開発を取り戻せ”，日経コンピュータ，2004年12月13日号，pp.52-69, 2004
- [2] A. Ribak, M. Jacovi, V. Soroka : “Ask before you search: Peer support and community building with reach out,” Proc. of ACM conference on Computer supported cooperative work (CSCW'02), New Orleans, Louisiana, USA, pp.126-135, 2002
- [2] “情報サービス産業における受注ソフトウェア開発の技術的課題に関するアンケート調査”，社団法人情報サービス産業協会，2004
- [3] V. R. Basili, F. E. McGarry, R. Pajerski, M. V. Zelkowitz : “Lessons learned from 25 years of process improvement: The rise and fall of the NASA Software Engineering Laboratory,” Proc. of the 24th International Conference on Software Engineering (ICSE2002), Orlando, Florida, USA, May 19-25, 2002
- [4] 井上 克郎, 松本 健一, 鶴保 征城, 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, Vol.45, No.7, pp.722-728, 2004
- [5] EASEプロジェクトホームページ, <http://www.empirical.jp/>
- [6] 大平 雅雄, 横森 励士, 阪井 誠, 岩村 聰, 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, pp.228-239, 2005
- [7] 村上 弘, 飯田 元, 松本 健一：“ソフトウェア開発プロセス管理データの収集と活用の支援を目的とした電子ガイドの提案”，電子情報通信学会技術報告, ソフトウェアサイエンス研究会, No.SS2004-41, pp.43-48, 2004
- [8] JIS X0141:2004, ソフトウェア測定プロセス. (ISO/IEC 15939:2002, Software engineering \_Software measurement process.)
- [9] “CMMIモデル・公式日本語翻訳版”，Software Engineering Institute, <http://www.sei.cmu.edu/cmmi/translations/japanese/models/index.html>
- [10] J. Kromrey, C. Hines, “Non-randomly missing data in multiple regression: An empirical comparison of common missing-data treatments,” Educational and Psychological Measurement, Vol.54, No.3, pp.573-593, 1994
- [11] 角田 雅照, 大杉 直樹, 門田 暁人, 松本 健一, 佐藤 慎一：“協調フィルタリングを用いたソフトウェア開発工数予測方法”，情報処理学会論文誌（産学連携論文特集）, (2005年5月掲載予定)
- [12] “プロジェクトマネジメント知識体系ガイド (PMBOKガイド) 2000年版”, プロジェクトマネジメント協会, 2000
- [13] 川口 真司, 松下 誠, 井上 克郎：“潜在的意味解析法LSAを利用したソフトウェア分類システムの試作”, 情報処理学会研究報告, Vol.2003, No.22, pp.55-62, 2003
- [14] ISEN homepage, <http://www.iese.fhg.de/ISEN/>
- [15] SECホームページ, <http://www.ipa.go.jp/software/sec/index.php>
- [16] Experts Exchangeホームページ, <http://www.experts-exchange.com/>
- [17] 教えて！gooホームページ, <http://oshiete.goo.ne.jp/>
- [18] J. Teevan, C. Alvarado, M.S. Ackerman, D.R. Karger : “The perfect search engine is not enough: a study of orienteering behavior in directed search”, Proc. of 2004 Conference on Human Factors in Computing Systems (CHI'04), Vienna, Austria, pp.415-422, 2004
- [20] CeBaseホームページ, <http://www.cebase.org/>