

# ソフトウェア開発プロジェクトの可視化(「見える化」)に関する 実証的な取り組みについて

## Empirical Study about the “MIERUKA” (Visualization) of Software Development Project

神谷 芳樹†  
Yoshiki Mitani

### 1. はじめに

IPA/SEC では開設以来産学連携の場を通してソフトウェア・エンジニアリングの様々な課題に関する調査・研究活動を進めてきた。その中で、本来そのままでは見えにくいソフトウェア開発プロジェクトの可視化というテーマに取り組んだ。5年以上の活動を通して、ソフトウェア開発プロジェクト可視化のための手法を書籍を中心にまとめ、「IT プロジェクトの「見える化」」手法として産業界へ推奨できるようになった[1][2]。これには産業界のノウハウを満載したチェックシートや大量の失敗事例集、網羅的な測定分析データ一覧表、そして産学連携プロジェクトの成果であるインプロセス計測のためのモニタリング・ツールが含まれている。ここでは、そのあらましを背景にある問題認識を含めて報告し、あわせて暗黙知の形式知化という課題に挑んだ産学連携の場の意義について述べたい。

### 2. 問題認識と活動の動機

ソフトウェア開発プロジェクトの「可視化」は研究課題としていつも理解が得られるとは限らない。たとえば大学キャンパスでの少人数の開発プロジェクトなどを念頭に、「見たければ見ればいいではないか」という反応を得ることもある。しかしながら一般に産業界でのソフトウェア開発は、「可視化」に課題として取り組まねばならないほど複雑な組織構成の中で進められることが多い。はじめにこの論点に理解を得るために 2 つの事例を紹介する。図 1、図 2 は、SEC がモニタする機会を得た 2 つの実開発プロジェクト (SEC 実証プロジェクト) の組織体制の事例を示している。

事例 1 は、数 10 億円の費用、発注側の視点で 7 社体制、工期 2 年半で 2 回程度の開発サイクル、100 人弱の開発関係者を要した典型的なマルチベンダのエンタープライズ型開発の体制例である[3][4]。大手システムインテグレータに全体のプロジェクトマネージャ (PM) を置き、このマネジメントのもとに大手ソフトウェア企業 5 社が、Linux サーバ上の情報システムを分担開発し、全体を組み上げて一つのシステムとして稼働させた。サブシステムの開発を分担する各社のもとに、いわゆる協会社体制が組まれている。協会社体制は各社各様で、そこには一般性は認められない。プログラミングを含むソフトウェア開発環境は、組織の最下端に近いところにある。ポイントは、開発に参加する各社は垂直方向にそれぞれ企業間の契約にもとづいて行動し、開発に参加する各要員はそれぞれの企業に属していて、最終的には所属する企業の服務規程に沿って業務を進めている。プロジェクト進行に関する情報には様々な秘匿条件があり、組織全体でフラットにオープンなわけではない。また参加している企業規模には大きな幅がある。従業員数万人の企業から、数百人あるいは百数十人の企業もある。そしてこの体制はこのプロジェクトだけの一過性のものである。このような体制は、ある種の領域では典型的、と表現できるが、決して広い一般性があるわけではない。

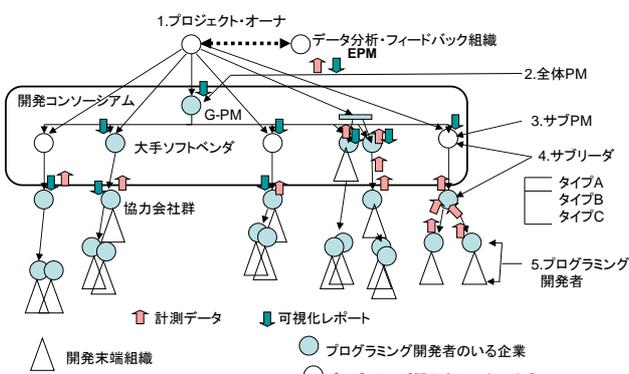


図1 ソフトウェア開発組織体制例 (事例1)

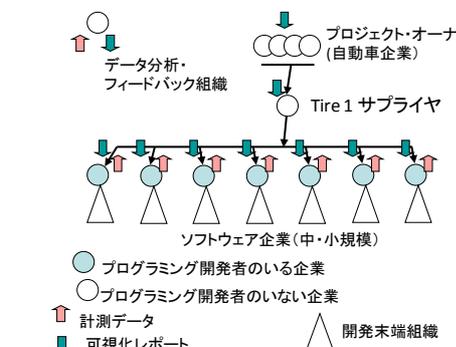


図2 ソフトウェア開発組織体制例 (事例2)

事例 2 は、自動車製造業の領域で、車載用のコンピュータのミドルソフトウェアを分担開発した体制の例で、事例 1 との共通性は少ない。大手自動車製造企業のもとに、tire1 サプライヤと呼ばれる大手部品供給企業が位置し、そのマネジメントのもとで 7 社の比較的小規模のソフトウェア企業が分担開発した。各社の開発したサブシステムは最終的に 1 つのミドルソフトウェアとして組み上げられ、車載コンピュータに搭載され、実働に供された。やはり数 10 億円の費用で工期は 3 年、この間に 2 回程度の開発サイクルがあり、約 60 人の要員が参加した。この場合も、各企業と要員はそれぞれの契約と服務規程に沿って行動し、またプロジェクト進行に関する情報はそれぞれ秘匿条件があり、組織全体でフラットではない。参加企業の規模の幅は

† (独) 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,  
(兼) 奈良先端科学技術大学院大学 NAIST

事例 1 よりもさらに大きい。そしてこの体制はこのプロジェクトだけの一過性のものである。

このように産業界のソフトウェア開発は、産業構造を反映した複雑な、一過性の、そして一般性の少ない組織体制の中で進められるので、その可視化には、単純に開発現場を観察しその情報を皆で共有するというだけでは済まない様々な工夫、道具立てが必要となる。これがこの活動の背景にある動機である。

### 3. 「見える化」手法の案出

IPA/SEC は開設時から、実証活動、定量データを重視するエンピリカルソフトウェア工学を注視してきた。そのなかで、まずベンチマーキングデータの収集と分析に取り組み、これを「ソフトウェア開発データ白書（データ白書）」の発行として結実させた。この活動のなかで、別の視点の要求が出てきた。「データ白書」では、一つのプロジェクトのデータは基本的にプロジェクト終了後に集められ、1,000 を超すプロジェクトの中の一つとして評価される。これでは、当該プロジェクトを進めている開発者には直接的な利便が薄い。そこで進行中のプロジェクトを計測・評価し、直接当該プロジェクト運用に役立てる方法の検討が求められた。「データ白書」がポストプロセス計測であるのに対し、インプロセス計測となる。

SEC の中に志を同じくする約 20 人の専門家による会合（部会）が組織され、以降、メンバーを少しずつ入れ替えながら 5 年余の検討と成果のまとめがすすめられた。部会を構成したのは、ソフトウェア企業から、いわゆる PM（プロジェクトマネジメント）系、あるいは PMO (Project Management Office) の専門家、そして品証系（品質保証系）の専門家、社会的に重要度の高いシステムを開発しているユーザ企業のシステム部門の専門家、学界からエンピリカルソフトウェア工学の専門家である。検討の方法は、これら経験豊富な専門家の知恵、そして手持ちの情報を集積し、シナジーを生んで、皆で幅広く使えるように整理して提示しようというものである。

こうしてまとめた手法の内容説明の前に、その活動の基盤となった考え方をいくつか述べる。

まず、全体を通して、「暗黙知」の「形式知」化ということが意識された。この領域で経験豊富なベテランたちは、ソフトウェアや情報システムを巡る激しい技術革新の波の中で、普遍性をもっていたソフトウェア開発管理技術の継承が不十分なことを実感していた。そこで、このプロジェクトを通してベテランの内に秘められた「暗黙知」の「形式知」化という努力が強く意識された。世代を超えて技術は継承されにくいという現実が認識されていたので、「世代間の遺言」という言葉も生まれた。具体的には、知恵を可能な限り文字にすること、それも読み下し文だけでなく、図表やデータシートなどで実用的なツールにしてゆくことが考えられた。またもちろん可能なものはソフトウェア・ツールに組み込んでゆくこととした。

次に、皆で持ち寄った知恵の整理方法として、3 つの視点を考えた。一つは工程分類に沿ったもので、全体を「上流工程」「中流工程」「下流工程」と区切って整理した。必ずしもウォーターフォール型開発を前提としたわけではないが、広範なソフトウェア開発工程の可視化を考える際、その立ち位置を明確にする必要があり、このような一般に考えやすい区分とした。概略、要件定義とシステム設計を

「上流」、ソフトウェア設計、プログラミング、ソフトウェア・テストを「中流」、システム・テストと運用テストを「下流」としている。

次いで、可視化にあたって、「定性的アプローチ」と「定量的アプローチ」の 2 つを考えた。いうまでもなくソフトウェア開発には人間的で定性的な要素と機械処理的で定量的な要素が含まれ、いずれかに偏ることは適切でないと考えた。

そしてもうひとつ、知恵の整理法として PMBOK (Project Management Body Of Knowledge : プロジェクトマネジメント知識体系) の知識エリアを用いた。プロジェクトマネジメントに関する知識を整理した PMBOK ではこの領域を整理するのに 9 つの知識エリアを設定している。そこで持ち寄ったチェック項目や失敗事例をこのエリアに沿って整理したところ、この 9 つではどうしてもしっくりせず、結論として新たに 6 つの知識エリアを設定し、合計 15 の知識エリアで考えることとした (図 3)。

No		知識エリア
1	PMBOK	統合
2		スコープ
3		タイム
4		コスト
5		品質
6		人的資源
7		コミュニケーション
8		リスク
9		調達
10	拡張知識 エリア	顧客
11		技術
12		組織
13		基本動作
14		モチベーション
15		課題管理

プロジェクトのステークホルダのうち、システムの仕様および予算について最終決定権を持っている人もしくは組織。  
ソフトウェア・エンジニアリングにおけるソフトウェア開発技術ないしはシステム構築技術のマネジメント。  
システム開発プロジェクトにかかわる組織。  
システム開発における常識および開発者が身につけておくべき当たり前の動作。  
システム開発マネジメントの内容も含む。  
システム開発に携わる人のやる気、動機づけ。  
システム開発技術における課題管理に関わるマネジメント項目。

図 3 15 の知識エリア

「見える化」手法ではこうした整理法と、これらを参照するいくつかの工夫によって、集められた貴重な知恵を定式化し、ひろく活用できるものとするを狙った。

### 4. 「見える化」手法の概要

SEC の部会でまとめた「見える化」手法は、大略図 4 に示すように、定性的アプローチ、定量的アプローチ、統合的アプローチから構成している。これを前述のように、上流、中流、下流の各工程ごとに用意している。

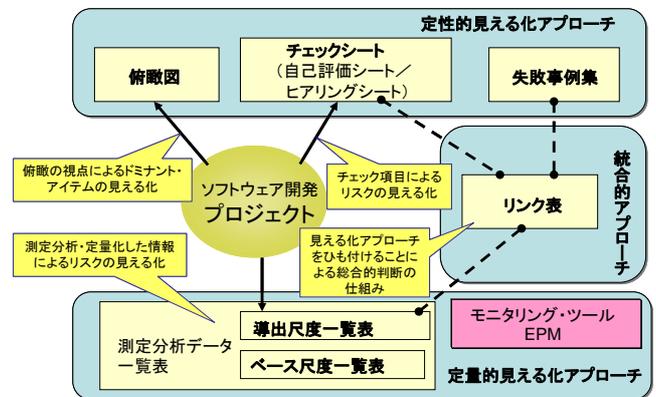


図 4 SEC の「見える化」手法の構成

定性的アプローチでは、まず、俯瞰図と呼ぶ全体を見通すチャートを描くことを推奨し、いくつかのサンプルを提

示している。次に、自己評価シート、ヒアリングシートと呼ぶ 2 種類のチェックシートを提供している。そして最後に豊富な失敗事例を提供している。

定量的アプローチでは、網羅的な測定分析データ一覧表を示している。これには、ベース尺度と導出尺度がある。また定量的アプローチでは、ソフトウェア開発環境のツールの中から開発管理に有用な情報を自動収集しグラフ等に可視化する Empirical Project Monitor (EPM) と名付けたモニタリング・ツールを提供している[5]。

統合的アプローチでは、チェックシートのチェック項目、失敗事例、測定分析データ間のリンク表を提供している。いずれかの項目から関連する他の 2 つの項目を検索する情報を提供している。

以下に主な機能の概要を述べる。

### (1) 俯瞰図

俯瞰図は、1 枚のチャートによってプロジェクトの全体像を把握し、「木を見て森を見ず」といった弊害を排除し、「ドミナント・アイテム」と呼ぶ、プロジェクトの成否を左右する少数の支配的要因を継続的にシステム横断的に把握することを狙いとしている。具体的にステークホルダー俯瞰図、プロジェクト推進体制俯瞰図、周辺システム構成俯瞰図、システム構成俯瞰図、スケジュール俯瞰図、要員遷移俯瞰図など各工程 4~7 種のチャートのサンプルを提示している。

### (2) チェックシート

プロジェクトマネジメントの要点を再確認し、客観的視点でのチェックによる見落としを排除し、リスクの明確化を図ることを目的としたチェックリストである。プロジェクトマネージャによる自己評価シートと、専門家チーム（たとえば PMO）によるヒアリングのためのシートからなる。自己評価は自己チェックによる「気付き」を狙い、ヒアリングは専門家からの客観的チェックを狙う。シートは各工程毎に用意され、自己評価用が約 40 問、各問いに 3 段階の回答を求め、回答に 40 分程度を要する。同じくヒアリング用が 70 問程度で、5 段階の回答を記入し、ヒアリングに 2 時間半程度を要する。集計用の簡易なツールが付属しており、回答投入後、自己評価とヒアリングの結果を前述の 15 の知識エリアを評価軸にして対比できるレーダーチャートとヒストグラムを出力する。

自己評価と専門家の診断の差が示されることになり、専門家からの対策案の提示やマネジメントの過不足の把握、対策の検討に役立てることができる。

### (3) 失敗事例集

全工程で 190 を超す失敗事例が集められている。それぞれ工程と知識エリアで分類され、事例のほかに、事例ごとに詳細な考え方を示している。上流工程では、「事例における見切り内容」「捉えるべき兆候」「本来の見切りの考え方」「対処例」、中流工程では、「事例における見切り内容」「本来の判断」「対処例」で、対処例はさらに「SI ベンダ」用と「顧客」用、それに「共通」用が示されている。下流工程では「原因」「対策」「対処療法」「再発防止策（教訓）」としている。

成功例ではなく失敗事例を収集した背景には、成功例にはどうしても美化要素が含まれていて真実を曇らせ、失敗にこそ学ぶべき対象があるという考えがある。この事例集

の中には、結果的に今日のソフトウェア開発現場の課題が余すところ無く描かれており、事例ごとに示された説明は、SEC の部会の議論の成果としての現実解と問題提起、あるいはその限界を示している。またマクロに見ると、先の俯瞰図の目指すところやチェックシートの項目と対になっていることがわかる。

### (4) 測定分析データ一覧表

ソフトウェア開発プロジェクトに関する、いわゆるベース尺度と導出尺度の網羅的な一覧を示している。ベース尺度としては、工程全体で 170 項目超、導出尺度としては 230 項目あまりを測定方法やデータの見方などとあわせてリストアップしている。総集編では導出尺度一覧として全工程分を整理し、「知識エリア」「測定の目的」「導出尺度」「工程分類」、そして優先づけのガイドを示している。

### (5) モニタリング・ツール：Empirical Project Monitor (EPM)

EPM は、ソフトウェア開発環境の中で、ソフトウェア開発管理に利用可能なデータを半自動的に収集し、その推移や分布などをグラフなど可視的な形で表示するモニタリング・ツールである。具体的にはプロジェクト進行中に構成管理（版管理）ツール、障害追跡ツール、メール管理ツールなどから自動的にデータ収集し、チェックイン契機、チェックアウトの頻度と契機、ソースコード規模推移、障害票発行状況とその分析、メール頻度などをグラフ化する（図 5）。近年その計測範囲を、要求定義や設計工程で描かれる各種のダイアグラムやレビュー記録に広げることを検討している。

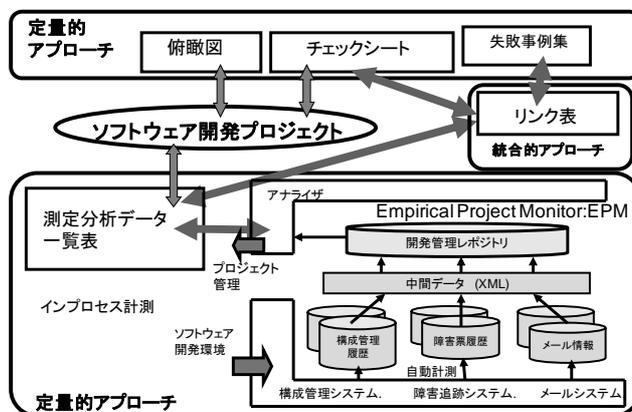


図 5 モニタリング・ツール EPM

### (6) リンク表

チェックシート、測定分析データ一覧表、失敗事例集を連携させ、統合的な視点でリスクを洗い出すことを目的としたリンク表である。このような各情報の関連付け動作は、本来ベテランのプロジェクトマネージャが頭の中で、経験と獲得した情報を組み合わせて行っていた行為のシミュレーションになっていると考えられる。SEC の部会活動のなかで意識された暗黙知の形式化化という考え方を推し進めた結果である。

ここで実際のチェックシートや失敗事例の一部、あるいはチェックシートや EPM によって可視化された情報の例を示してその内容を紹介したいところであるが、限られた

紙面ではほんの断片にしかならず、その全貌はもとより、その内容の精緻さを表現することが難しい。要約的な紹介が誤解のもとになることを恐れて、ここでは参考文献を示すにとどめる。

## 5. 可視化への多様なニーズ（実証プロジェクトから）

ここまで述べた手法・ツールを活用したソフトウェア開発プロジェクトの可視化とそのプロジェクト運営へのフィードバックの有用性はいくつかの場面で目撃してきたが、ここでそのうち2.で示した2つの事例で得た反応を紹介したい。

事例1では、チェックシートによる見える化、EPMや関連ツールによるインプロセスの計測、ベンチマークデータと協調フィルタリングの技術を活用した過去の類似プロジェクト検索とそこからの工数予測、コードクローン分析、プロジェクト参加者のスキル計測など、総合的なプロジェクト計測と可視化、フィードバックを試みた[3]。

事例2では、EPMによるインプロセスの計測と合わせて、WBS (Work Breakdown Structure)による進捗管理ツールからの進捗管理データ集計、SECの組込みシステム開発リファレンス ESxR (Embedded System development exemplar Reference)に沿った品質データの計測、それにスキル計測[6]が組み合わされた。事例1、事例2いずれも計測データは1つの特別な組織に集計され、分析・可視化されて、プロジェクトの各組織にそれぞれの秘匿条件に沿ってフィードバックされた。以下はそのフィードバックで得た反応をもとにしている。

### 5. 1 事例1

事例1では、可視化に対し全体で4階層、さらによく観察すると6つの必ずしも同じではないニーズがあることが明らかになった[4]。それらは大略次のようなものである。

1) プロジェクト・オーナー（発注者）：

プロジェクトのマクロな進捗状況、オーナーとしての判断事項の有無を把握したい。

2) プロジェクトマネージャ：

各サブシステム担当ベンダ毎の作業特性を把握したい。

3) サブリーダー A、プロジェクト管理能力に優れているが、多くのプロジェクトを抱え、プロジェクト把握のために十分な資源（時間）を避けないリーダー：

この場合、計測による自社プロジェクト進捗の可視化に大きな期待がある。比較的マクロな、プロジェクトの傾向、危機の予兆を示すデータに関心を持つ。

4) サブリーダー B、管理対象プロジェクトに密着し、当該プロジェクト管理に比較的十分な時間を避けるリーダー：

この場合、計測データの分析により、目視ではわからないより深い開発状況、少しでも速い危機の徴候の把握に期待している。比較的ミクロなデータ分析に関心を持つ。

5) サブリーダー C、様々な理由で、プロジェクト管理の経験が少なく、多くを下位の組織に任せているリーダー：

この場合、可視化情報を提示しても評価できない。情報はその下位の組織で活用する必要がある。

6) 組織末端の開発グループ：

可視化によって見られていることに必ずしも否定的ではない。上位の組織との一体感を増し、組織としての迅速な

判断などプロジェクト運営上の利点に期待している。またフィードバックデータを自らのプロセス改善へ活用・反映できることが期待される。

### 5. 2 事例2

事例2では、可視化に対して、開発を進める組織体の使命に沿って多くの異なったニーズが明らかになった。これらは対象領域特有のもので事例1とはかなり異なったものである。

1) 最終製品ベンダ（プロジェクト・オーナー）：

進捗よりも品質に責任を持っているため週次の進捗データよりも、進捗の中で品質について判断できる情報を切望している。そのためのデータ蓄積を強く要望された。

また今後説明責任が強くなることが想定され、国際的に認められたプロセスや計測にもとづいて製造されていることを示せる追跡性を求めている。

2) サプライヤのプロジェクトマネージャ：

可視化へのいくつかのニーズが認められた。従来の人間的な経験によるプロジェクト把握にエビデンスを与えるものとして歓迎する意見があった。さらに可視化されたデータにより、進捗状況はもとより、各開発グループのレビュー、試験、プロジェクト管理への姿勢を読み取ることができ、これらのベンダの相違を社間結合試験の管理などに反映できることを評価する意見が得られた。

一方、可視化情報の提示を歓迎しない見解もあった。事例のようにソフトウェア・ベンダが7社程度であれば可視化情報を活用できるが、一般にサプライヤのマネージャが対応する、たとえば100社というような規模では、マネジメントへの活用は困難になる。こうした環境では、可視化情報は各ソフトウェア・ベンダの中で自らのマネジメントに活用すべきで、サプライヤに提示しても活用できないという見解である。

3) ソフトウェア企業：

中規模以上の企業は、こうした可視化データをプロセス改善に反映してゆきたい希望をもつが、一方1チーム5人以下といった小規模な企業の中には、人間中心のマネジメントを考へており、計測オーバーヘッドが無視できない可視化を希望しない意見も見られた。

このように、ソフトウェア開発プロジェクトの可視化といっても、そこに期待されることは極めて多様である。いくつかの実証実験から得られたこととして、ソフトウェア開発の広い世界をきめ細かく観察し、ソフトウェアを開発する産業構造をきちんと反映した計測と可視化、フィードバックの機能を実現してゆくことの必要性が明らかになった。

## 6. 知恵の定式化に関する考察

最後に、ここで示したような手法やツールの案出・整備に関する活動の位置づけについて、若干の考察を述べたい。

1) 「暗黙知の形式知化」

この概念は3.でも述べたようにこの活動で当初から強く意識された。近年、アジャイル開発手法の一つ「スクラム」の源流となった「ラグビーメタファ」の原典である野中郁次郎の「知識創造企業」の中で連続的に知を産み出す企業の条件としてその重要性が強く訴えられている概念で

ある [7]。ここで述べてきた「見える化」手法は、チェックシートや失敗事例集、これらの組み合わせ手法など、まさにその具現化のひとつと考えられる。

## 2) 「オープンイノベーション」

チェスブロウの指摘した、各企業が自己完結的な R&D だけに頼っている継続的なイノベーションで競争に勝ち抜くことが難しく、企業間によるイノベーションの相互乗り入れが必要という考え方で、今日その支持が広がっている [8]。IPA/SEC の活動をその機能で考えてみると、産業界や学界から広く専門家の参集を求め、その知恵やデータを持ち寄って、官界の知恵も加えてシナジー効果による知的生産物として形にし、これをひろく公開・普及させる「場」を提供していることになる。まさに、チェスブロウの言う「オープンイノベーション」実行のための強力な環境（「場」）を実現していることに相当すると考えることができる。

## 3) 新しい産学連携：「テクノロジーアービトラージ」と「マーケットメイク」

IPA/SEC は公式のミッションにある通り産学連携活動を実践している。しかし、その活動は対象とするソフトウェア産業の特性を反映して、伝統的な、「学界にある技術の産業界への移転」、あるいは、「産業界の課題の学界への持ち込みと解決」という枠組みには当てはまらない。その底流には神谷の指摘した、仲介者による「テクノロジーアービトラージ」、と、「マーケットメイク」という高度な役割が含まれている [9][10]。

IPA/SEC は、一面で産業界と学界の間に存在する双方向のテクノロジーの価値の落差に着目し、その流通を図ることで自らの存在価値を創出している。たとえば、ソフトウェア開発プロジェクトに関する膨大なデータ、ケースが産業界に死蔵されているが、これらは学界にとっては高い価値がある。一方これらのデータ分析法や様々なソフトウェア・ツールがその考え方とともに学界に眠っていて、それらが R&D の不足しがちな産業界で高い有用性がある場合がある。仲介組織はこれらの価値の落差を利用してその情報・知的価値の流通を図り、広い意味でその「さや取り（アービトラージ）」によって自らの存在価値を産み出すことが出来る。

一方、産業界と学界の知的価値の流通は、双方をいわば売り手と買い手として単純に引き合わせるような、いわゆる「オークション」方式の取引形態では容易に実現できない。そこには、双方の知的価値について見通せる視点の高い仲介者が居て、仲介者のリスクに於いてその流通を図る、つまり金融界の「マーケットメイク」に相当する行為が必要である。

IPA/SEC はその産学連携活動に於いて、こうした仲介者の役割の一部を担っていて、例えば「見える化」施策の中の計測ツールの実用化と普及や、産学協同の実証プロジェクトなどで具体化して来たと考えられることができる。

## おわりに

IPA/SEC でのソフトウェア開発プロジェクト可視化への取り組みについて述べた。SEC の部会活動等でまとめた手法の紹介とあわせて、実証プロジェクトでの経験を背景に、ソフトウェア開発を行う産業構造を反映した可視化

データのフィードバックが重要なことを述べた。今春よりここで紹介した「見える化」手法の一部を英訳して、Web サイトから公開している。“MIERUKA”という表現も積極的にアピールしている。グローバルな環境を含めて手法の普及を図り、またその反応を得て、次の展開を図ってゆきたい。

## 謝辞

日頃、調査・研究活動を支援いただく、経済産業省、文部科学省、StagE プロジェクト、SEC-IT プロジェクトの見える化部会の各位に謝意を表します。

## 参考文献

- [1] IPA/SEC：IT プロジェクトの「見える化」；上流工程編、中流工程編、下流工程編、総集編、日経 BP 社、2006-2008
- [2] 神谷芳樹、樋口 登：（オピニオン）IT プロジェクトの「見える化」、電子情報通信学会誌、Vol.91, No.6, 2008, pp.510-514
- [3] IPA/SEC：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
- [4] Yoshiki Mitani, Tomoko Matsumura, Katsuro Inoue, Mike Barker, Akito Monden, Ken-ichi Matsumoto: An Empirical Study of the Feedback of the In-Process Measurement in a Japanese Consortium-type Software Project; 21th International Conference on Software Engineering & Knowledge Engineering (SEKE 2009), Boston, USA 2009-Jul. pp.631-636
- [5] 鳥居宏次（監修）、EASE プロジェクト：ソフトウェア開発におけるエンピリカルアプローチ、アスキー、2008
- [6] 国プロ推進 WG・プロセス構築チーム、北本桂造ほか：JASPAR・国プロ推進 WG における ETSS の導入、SEC journal Vol.5 No.2, 2009-4, pp.120-123
- [7] 野中郁次郎、竹内弘高：知識創造企業、東洋経済新報社、1996
- [8] ヘンリー・チェスブロウ：オープンビジネスモデル、翔泳社、2007
- [9] 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、2001
- [10] 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（産学連携学会）、Vol.2.No.2 2006-3, pp.26-37

文献 [1][3][6] は IPA/SEC の Web サイトから、[6][10] は電子ジャーナルの JSTAGE のそれぞれのサイトからダウンロード出来る。[4] は当該国際会議のサイトから Proceedings としてダウンロード出来る。

<http://www.ksi.edu/seke/seke09.html>

文献 [1] については、部分英訳版を、「ソフトウェア開発データ白書」の英訳版と合わせて IPA/SEC の Web サイトで公開している。

なおモニタリング・ツール EPM は、IPA/SEC のサイトにあるガイド（EPM 検証プロジェクト）に沿って申し込むことで、本論、事例 1 で述べた関連ツールとあわせて、媒体で提供されている。

以上