

ユーザ視点に立ったソフトウェア開発データ活用技術の提案

松村知子[†] 森崎修司[†] 門田暁人[†]
楠本真二^{†††} 飯田元[†] 松本健一[†]
久保浩三^{††} 井上克郎^{†††}

ソフトウェアに大きく依存している社会において、安心・安全な社会維持の実現のためにはソフトウェアに対する高い信頼性が求められている。我々は、ソフトウェアユーザの視点に立って、安心して安全なソフトウェアを選択し利用するために、ソフトウェア開発情報共有の技術として、「ソフトウェアタグ」を提案する。ソフトウェアタグは、ソフトウェアのユーザが利用の可否を判断するために必要なソフトウェア開発プロセスや品質情報データを包含し、対象ソフトウェアのすべてのユーザに開示されるものである。本稿では、ソフトウェアタグの基本的な利用モデルや構造の構想について説明し、実運用に向けた具体化の進展状況について報告する。

Effective Use of Software Development Data from Software Users' View Points

TOMOKO MATSUMURA,[†] SHUJI MORISAKI,[†] AKITO MONDEN,[†]
SHINJI KUSUMOTO,^{†††} HAJIMU IIDA,[†] KEN-ICHI MATSUMOTO,[†]
KOUZOU KUBO^{††} and KATSURO INOUE^{†††}

Nowadays, high quality software system is required because our society increasingly depends on software systems. We propose "Software Tag", a technology to share the software development data, for software users to choose secure and reliable software. Software tags include data of software development process and quality of software artifacts, and should be shown to software users to judge if it would be usable or not. In this report, we explain the idea of software tags and usage models, and report the progress of detailed technical researches for practical application to the society.

1. はじめに

近年、ソフトウェアの品質が社会に与える影響は増大し、ソフトウェアの不具合による人命の危険の増大、莫大な経済的損失の発生が、一般の報道に取り上げられることも多い。それに伴って、ソフトウェア開発をめぐる訴訟が急増している。ソフトウェア開発企業では、ISOなどの品質規格やCMMIなどに従って、ソフトウェア開発プロセスの改善などによるソフトウェアの品質保証に努めている。

開発プロセスやプロダクトの計測・定量化と評価、フィードバックというエンピリカル(実証的)アプローチが、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。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¹⁾。また、エンピリカルアプローチによる研究の成果を業界や学会で共有するEMPEROR(Experience Management Portal using Empirical Results as Organizational Resources)アプローチも、既存の成果を広く社会に普及していくことを目指している²⁾¹⁰⁾。

STAGE (Software Traceability and Accountability of Global Software Engineering) プロジェクト⁴⁾⁸⁾では、文部科学省 次世代IT基盤構築のための研究開発「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」という課題に取り組んでいる。この研究は、これまでにはなかったソフトウェアユーザの視点に立ったエンピリカルアプローチの実現手段と

[†] 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute
of Science and Technology

^{††} 奈良先端科学技術大学院大学先端科学技術研究調査センター
Research Center for Advanced Science and Technology,
Nara Institute of Science and Technology

^{†††} 大阪大学大学院基礎工学研究科
Graduate School of Information Science and Technol-
ogy, Osaka University

して、「ソフトウェアタグ」を提案する。ソフトウェアタグは、ソフトウェアの開発プロセス中に収集された管理データや品質情報から構成され、ソフトウェアに付随して流通する。ソフトウェアユーザは、タグの内容を見ることによって、対象ソフトウェアの開発過程や品質保証のプロセス、成果物（プログラムなど）の品質情報を入手でき、安心して安全なソフトウェアを利用することが可能になる。ここでの「ソフトウェアユーザ」は、受注開発における発注者、パッケージソフトウェアの利用者、ソフトウェア部品を流用・改造するソフトウェア開発者などが考えられる。以降、特記しない限り、「ユーザ」はこれらの人々を包含した意味を持つ。

ソフトウェアタグの普及により、ユーザにとってこれまで不透明であったソフトウェアの開発過程や品質管理プロセス、品質情報が見えることになり、安心してソフトウェアを購入・利用することができるようになると共に、適正な商品の選択をすることが可能になる。また、ソフトウェアに不具合が発生した時や保守を行う場合に、問題の特定や修正などを迅速、正確に行えるようになり、ソフトウェアの問題による社会的な影響を軽減することができる。さらに、ソフトウェアの問題について発注者・ベンダ間で紛争に発展した場合に、その解決を助ける証拠の情報が保存され適正に開示されることにより、早期に公正な判定を行うことが可能になる。

一方、ソフトウェアタグの普及は、ソフトウェア開発業界側にも、以下のような点で利益がある、と考えられる。

- 定量的プロジェクト管理の促進・プロセス改善が普及する
- データ項目の標準化によって、ツール・計測技術の開発が促進・普及され、計測や分析のコストが低減する
- 統一基準による組織・プロジェクト・システム評価により、適正競争が促進される
- 国際競争力が強化される

STAGE プロジェクトでは、ソフトウェアタグの実現と普及のため、ソフトウェアタグ項目の規格化、タグ情報収集支援、タグの生成・流通方法の検討、タグ情報の可視化・評価、タグを用いた法的問題解決方法の検討を行っている。

以降、本稿では、2章でソフトウェアタグの利用モデルについて説明し、3章でソフトウェアタグの分類、4章でソフトウェアタグの具体的な構想について述べる。5章ではソフトウェアタグの実現・普及のための支援技術について述べ、最後に今後の研究方針について述べる。

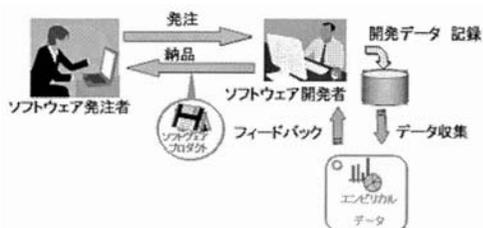


図1 従来のエンピリカルアプローチによるソフトウェア開発⁴⁾

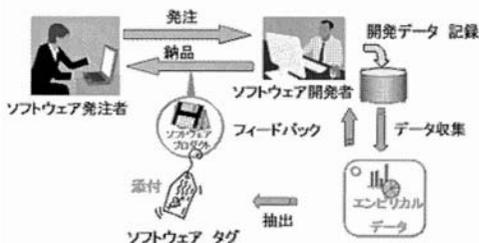


図2 発注者とベンダ間のプロジェクトデータ共有⁴⁾

2. ソフトウェアタグの利用モデル

本章では、ソフトウェア開発現場でのタグの活用モデルを挙げて、それぞれの具体的な活用方法と有効性について述べる。

2.1 発注者への開発透明性の確保

ソフトウェア開発の透明性を確保するための活動は、ベンダ内でこれまでも積極的に進められてきた。図1は、従来のソフトウェア開発組織に閉じたエンピリカルデータによるプロジェクト管理を示す。データは開発者や管理者によって収集され、組織内の品質保証チームやプロジェクト管理者などの分析担当者によって分析される。そこから得た知見やノウハウは対象プロジェクトの関係者や後続のプロジェクトにフィードバックされ、生産性や品質の向上に役立てられる。

図2に、発注者とベンダ間のソフトウェア開発プロセスやソフトウェアの品質の透明性を上げるためのソフトウェアタグの利用モデルを示す。図2では、収集されたエンピリカルデータから有用なデータを抽出し、ソフトウェアタグという形態で納品するソフトウェア製品に添付する。このモデルは、一次請負ベンダとそのベンダから作業を二次請負する組織間でも用いることができる。また、このソフトウェア製品が流用を前提とした部品である場合、この部品を利用するユーザ、すなわち開発者に開示されたり、家電などに組み込まれるソフトウェアの場合、その製品を購入する一般ユーザにも開示されることも考えられる。

ソフトウェアタグは、検収時の品質検証、製品選択

● 開発過程で付随的に発生するデータ。仕様変更管理票、障害票、議事録など。
シークレットタグをオープンタグと分離するのは、以下の理由による。

- シークレットタグは、一般的に開発側の組織内部の情報（コスト、要員、工数など）を含むため、開示に際して開発側組織の利益の確保や個人情報など機密データの保護に留意した取扱いが求められる。
- シークレットタグは、基本的に開発側組織の必要性に応じて作成されるデータで構成される。開発対象や開発体制などにより、データの量や質、種類などは、多様であるため、一般ユーザの理解や分析には適さない。

シークレットタグは、通常はソフトウェア開発側の組織に保管され、外部へ開示されることはないが、ソフトウェアの不具合発生時の検証や法的紛争発生時の解決のために、第三者機関に委託され、分析される場合がある。図3で調査されるエンピリカルデータが、シークレットタグに含まれるデータである。

4. ソフトウェアタグの規格化

ソフトウェアタグの普及のためには、異なる開発分野や組織間での流通を可能にする規格化が必要である。本章では、ソフトウェアタグの規格化の推進体制、及びオープンタグの具体的な規格案について説明する。

4.1 タグ規格化の体制

オープンタグ、およびそれに伴うシークレットタグの規格化は、奈良先端科学技術大学院大学、大阪大学主催で、企業のプロジェクト管理者や品質保証担当者、法的問題に関する有識者（弁護士、弁理士）が参加する「ソフトウェアタグ技術規格委員会」（以降、タグ規格委員会）、「ソフトウェア構築可視化に伴う法的諸問題に関する検討委員会」（以降、法的問題検討委員会）において、進められている。

タグ規格委員会は、オープンタグに含まれるデータ項目の抽出やタグの活用方法の検討、タグ普及のためのツールや支援技術に関する調査などを行っている。具体的には、企業参加者から各社内で用いているプロジェクト管理、品質管理のための定量的データの情報を持ち寄って、タグとしての有用性や入手方法、理解容易性などについて議論したり、タグの活用構想を検討したり、データ収集ツールについて調査したりしている。さらに、プロセスアセスメント規格⁷⁾やモデル契約書¹⁶⁾、ある企業でのCMMレベル取得時の計測データ事例などの情報を加味してタグ項目を洗い出し、SLCP(Software Life Cycle Process)¹¹⁾に対応したタグのライフサイクルやタグの構造、タグのフォーマットについての調査・議論を進めている。

法的問題検討委員会では、これまでにソフトウェア

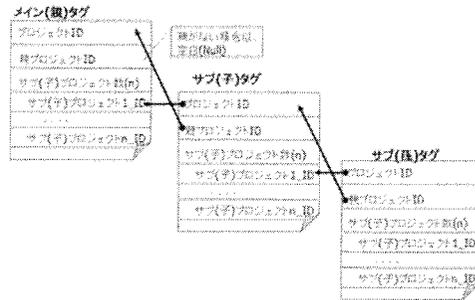


図5 タグの再編構造

の不具合や運用時の問題によって、発注者・ベンダ間で法廷で争われた事例を収集し、問題の原因やその判決例から、ソフトウェアに関する法的問題の解決に有用な情報を調査している。

以降、タグ規格委員会で進められているオープンタグの規格化の各観点と、その具体案について説明する。

4.2 オープンタグ項目

ユーザに対してソフトウェアの品質や開発プロセスを「見せる」ためのデータとして、オープンタグは以下の3分類のデータから構成される。

- Profile Info. : 開発プロジェクト及びシステムの基本情報。
- Process[/Progress/Trajectory] Info. : 開発プロセス（進捗、軌跡）情報。
- Product Info. : 成果物情報。

Profile Info. は、主にシステムの概要や開発組織の情報、その他のデータを解釈する際に参考にするコンテキストデータ、例えばシステムの種類、開発言語、品質要件、などを含む。これらは、SECの400項目データ⁵⁾などでも使われている。

Process Info. は、ソフトウェアの開発プロセス情報となる。例えば、成果物の規模推移、テスト項目の消化数（率）、不具合数（密度）などが考えられる。これらの情報は、通常のプロジェクト管理ですですに収集・活用されている企業が多いが、ユーザへの開示は一般的ではない。

Product Info. は、主に最終成果物（プログラムコードやシステム全体）に関する情報である。例えば、プログラムコードの品質として、複雑度やテストカバレッジ（網羅率）が挙げられる。また、システム全体の品質として、機能性、信頼性、保守性などが考えられる。

一方、タグは、流用や再利用、既成部品の組込みなどに対応するため、再帰的構造を取れるようにタグ管理情報を持つ。図5にタグの管理情報と再帰的構造を示す。タグは、ユニークなプロジェクトIDを持ち、再帰的な構造をとるために、親プロジェクトID、子プロジェクト（サブプロジェクト）の数とそのプロジェ

クト ID の項目を持つ。この具体的な利用方法については、4.4 節に述べる。

オープンタグ項目の例を付録に示す。

さらに、対象ソフトウェアの種類や開発形態に応じて適切なタグ項目選択を容易にするため、タグ項目には属性を付加する。例えば、使用目的別に、開発プロセスの妥当性を示すための項目、流用資産の品質を保証する項目、開発システムの進捗を裏付ける項目、といった属性を付加することが考えられる。他にも、タグのプライオリティ（すべてのプロジェクトに必須、推奨、任意）や、プロジェクトの規模、組込み系やエンタプライズ系など開発システム分野、開発・保守・運用などの工程によって有用・不用品目があり、その選択を支援するための属性を項目に付加する。

4.3 タグのライフサイクル

オープンタグのライフサイクルは、通常のソフトウェア開発プロセスだけではなく、保守・運用・再利用も対象とする。図 6 に、SLCP¹¹⁾ やモデル契約書¹⁶⁾ の開発工程に対応した、受注開発におけるタグのライフサイクルの一例を示す。

ソフトウェアタグに関する具体的な作業は、契約にソフトウェアタグに関する条項を盛り込むため、開発の企画段階から発注者・ベンダ間の調整が開始される。発注者・ベンダ間で取り決める必要がある情報としては、オープンタグの項目、システム構造や開発体制に応じたオープンタグの構造、データ入力やデータ開示タイミングなどオープンタグの運用ルール、シークレットタグの開示条件などが挙げられる。図 6 では、契約に従って、ソフトウェア開発開始時にソフトウェアタグを生成し、このときにプロジェクトやシステムの概要を示す情報 **Profile Info.** が入力される。タグの運用ルールに従って、進捗管理・品質管理に使う場合は定期的に開発プロセス情報 **Process Info.** を入力し、ユーザへ納入、検証をすることが考えられる。システムによっては、開発完了時に **Process/Product Info.** を入力するだけで、途中の段階で何もする必要がない場合も考えられる。

開発完了後、運用や保守についても、ソフトウェアタグによる管理が必要と考えている。図 6 では、運用や保守工程では、サブ（子）タグを生成して、データを入力する例を示している。これは、運用や保守工程では、体制や要員、期間などプロジェクトに関する情報が変わる可能性があるためである。タグへ入力されるデータも、開発工程とは異なり、運用工程では発見された不具合や修正情報など、保守工程では小規模な新規開発と同様のデータが考えられる。

4.4 タグ構造の変遷

4.2 節で述べたとおり、タグは再帰的な構造を可能とするが、その具体的な活用方法について説明する。新規開発時は、図 7 に示すように、プロダクトの構造にあわせてタグを構成することが望ましい。これは、

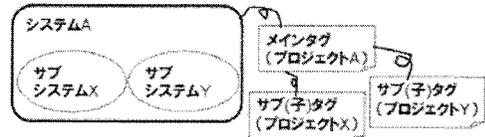


図 7 新規システム開発時のタグ構造

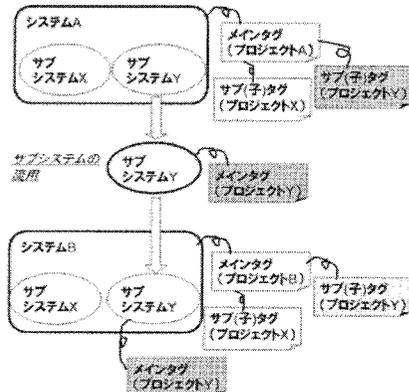


図 8 サブシステム流用時のタグ構造

将来的に再利用や流用を容易にするためだが、一方、進捗管理や品質管理の単位でもあり、収集・分析データの流用が可能であるため、コストが低減できる。

図 8 は、他のシステムの部品を流用する際のタグの構造を示している。濃い網掛けのタグが同一タグになる。薄い網掛けのタグは、流用するサブシステムのタグ（濃い網掛け）から作成されるが、システム B への組込み作業によってタグデータが上書きされる。

流用があらかじめ予期されていない場合、流用するサブシステムのタグを開発完了後に新たに生成しなければいけないことも考えられる。そのようなケースでは、シークレットタグから部品情報を取り出すことが望ましい。

図 9 は、開発されたシステムの運用・保守を継続する際のタグ構造を示す。タグは、メイン開発時のタグのサブ（子）タグとして生成されるが、4.3 節で述べたとおり、入力されるデータは異なると考えられる。

4.5 タグフォーマットの標準化

オープンタグは、流通を容易にするため、標準フォーマットを決める必要がある。なぜなら、タグの活用現場としては、システム開発だけではなく、法廷や一般家庭も考えられるからである。タグのフォーマットには、タグ項目の意味定義や 4.4 節で述べたようなタグ間の構造（親子関係、兄弟関係等）の定義も包含される。また、受け取ったデータの解釈、加工、表示に関する情報も組み込めるフォーマットが望ましい。

現在、電子的な財務情報の作成・流通・再利用を可能にする XBRL¹²⁾などを参考にしながら、XML に

プロジェクトライフサイクル

SLQP-2007	企画プロセス			開発プロセス			運用プロセス	保守プロセス
モデル契約書による工程	システム化の方向性	システム化計画	要件定義	システム設計	ソフトウェア開発 プログラミング ソフトウェアテスト	システムテスト	運用テスト	運用 保守

ソフトウェアタグライフサイクル

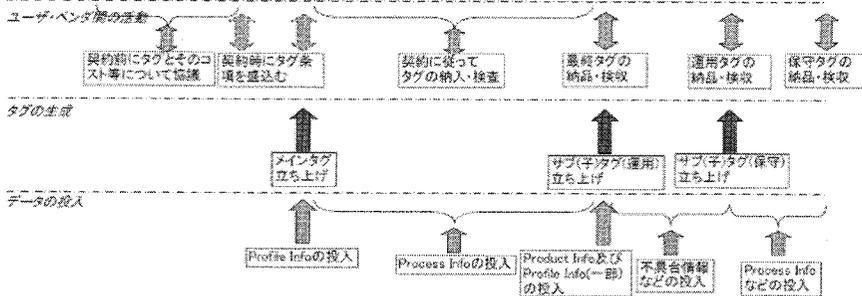


図 6 オープンタグのライフサイクル

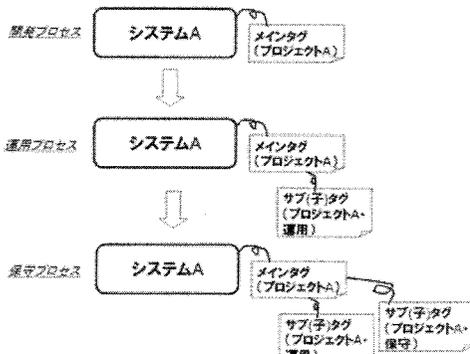


図 9 システム運用・保守のタグ構造

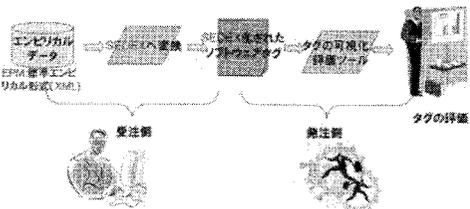


図 10 標準タグフォーマットによるデータの流通

よる標準化を検討中である。XBRLでは、財務情報の標準化、Report形式の標準化、外部報告形式の標準化などの活動が行われているが、これはソフトウェアタグにおけるタグ項目の規格化、タグデータの定型化、タグデータの可視化、などに類似した考え方といっている。

図 10 に、XML フォーマット化されたデータの活用

例を示す。SEDEXとは、Software Engineering Data EXchange languageの略で、ソフトウェアタグの標準XMLフォーマットを意味する。

5. ソフトウェアタグの支援環境

ソフトウェアタグの普及には、導入計画からデータの収集・分析・可視化・評価など、多面からの環境の整備を必要とする。本章では、従来のエンピリカルデータによる開発支援研究から、ソフトウェアタグの支援に有効な研究の一部を紹介する。これらの技術やツールを活用することによって、導入のオーバーヘッドを低減することが可能になると考えている。

伏田らによるEPDG²⁽¹⁴⁾は、ソフトウェアベンダ組織内でプロジェクトの定量的管理を支援するためのガイドブックとして提案されているシステムであるが、これを発注者・ベンダ間のソフトウェアタグ導入計画に適用する研究が行われている。

開発プロセスデータ収集ツールとしては、EASEプロジェクト*により開発されたEPM¹³⁾が、時系列の規模データや不具合発生データなどを自動的に収集する。一方、プログラムコードの分析・解析ツールとしては、CCFinder⁶⁾が、ソースコードのクローン(類似コード)を検出する機能を提供する。Gemini¹⁹⁾は、検出されたコードクローンの可視化とクローン率など定量的データを用いて、プログラムコードの保守性を計測することを可能にしている。

ProjectReplayer³⁽¹⁵⁾は、構成管理システムや障害管理システムから収集されたデータを用いて、プロジェクトの軌跡を視覚的に再現するツールである。EPM-

* Empirical Approach to Software Engineering : <http://www.empirical.jp/>

ProStar¹⁸⁾は、同じく構成管理システムや障害管理システムから定量的データを加工・抽出し、グラフ化することで、仕様の安定度、要員の適正配置など計測目的に応じたプロセスの可視化を支援する。これらのツールによって、ソフトウェア開発の知識や分析技術を持たないユーザがタグデータを容易に理解・評価することが可能になる。

マイクロプロセス分析⁹⁾¹⁷⁾は、構成管理システムや障害管理システムなどから取得できるログデータをプロセスモデルとつき合わせることで、定義済みプロセスの遵守度を計測することができる。開発完了後に決められたプロセスでの作業が正確に行われていたか、ログが収集されていたかを確認することができる。

さらに、より詳細なデータ分析やタグデータの信頼性や品質を保証するために、第三者によるタグの認証機関の設置を検討している。この機関では、専門的な知識や技術を持った調停者が、ソフトウェアの不具合が法的な問題に発展した場合、開発組織やユーザからシークレットタグの提供を受け、問題原因の検証や判定を行ったり、タグデータそのものの信憑性や信頼性を検証したりする。それによって、タグの適正な普及・活用を促進することが可能になる、と考えている。

6. おわりに

本稿では、ソフトウェアの安全性を確保し、安心してソフトウェアに依存する社会で暮らすことを可能にするソフトウェアタグの構想について述べた。ソフトウェアタグの普及により、ソフトウェアの発注者や利用者、既成ソフトウェアの再利用をする開発者は、これまで不透明であったソフトウェアの開発過程や品質管理プロセス、品質情報が見えることになり、安心してソフトウェアを購入・利用することができるようになる。また、ソフトウェアに不具合が発生した時や保守を行う場合に、問題の特定や修正などを迅速、正確に行えるようになり、ソフトウェアの問題による社会的な影響を軽減することができる。

STAGEプロジェクトでは、ソフトウェアタグの普及のため、ソフトウェアタグの規格化、データ収集支援、データ可視化・評価の支援、流通のためのソフトウェアタグフォーマットの標準化などの技術開発と導入支援に取り組んでいる。具体的には、現場でのソフトウェアタグの利用モデルを提案し、利用目的や範囲に応じてオープンタグとシークレットタグの2種類に分類した。オープンタグが保持するデータとして、**Profile Info.**、**Process Info.**、**Product Info.**の3種類のデータを抽出し、また、ソフトウェアの再利用や流用に対応するため、ソフトウェアタグの再帰構造を実現可能な項目を持たせた。一方、ソフトウェアタグの具体的な利用イメージを持つために、ソフトウェア・ライフサイクルプロセス(SLCP)に対応した

ソフトウェアタグのライフサイクルの例を示した。

また、STAGEプロジェクトでは、ソフトウェアタグは拡大するオフショア開発市場での品質やコミュニケーションの問題解決に有効な技術と考え、ATGSE⁴⁾の主催などを通じた国際連携も進めている。

今後は、ソフトウェアタグの実運用を支援するための技術や研究を推進していく予定である。

謝辞 本研究は文部科学省「次世代IT基盤のための研究開発」の委託に基づいて行われた。本研究にあたり多くのご協力を頂いたソフトウェアタグ規格技術委員会、ソフトウェア構築可視化に伴う法的諸問題に関する検討委員会の以下の関係諸氏に感謝します。

江口順一、小柴昌也(帝塚山大学)、青江秀史、茶園茂樹(大阪大学)、松田貴典(大阪成蹊大学)、大西幸男(西銀座法律事務所)、岩崎新一(日本電気)、木谷強、大杉直樹(NTTデータ)、清田辰巳(東京証券取引所)、高木富士雄(シャープ)、福地豊(日立製作所)、松尾昭彦(富士通研究所)、村山浩之(デンソー)、山田淳、森俊樹(東芝)、松村好高、阪井誠(SRA先端技術研究所)、鶴保証城、神谷芳樹(IPA/SEC)、片平真史、宮本祐子(JAXA)(敬称略、順不同)

参考文献

- 1) Endres, A. and Rombach, D.: *A Handbook of Software and Systems Engineering, Empirical Observations, Laws and Theories*, Pearson Education Limited, UK (2003).
- 2) Feldmann, R. L., Shull, F. and Shaw, M. A.: Decision support for best practices: Lessons learned on bridging the gap between research and applied practice, *Defense Acquisition Review Journal*, pp.234-247 (2007).
- 3) Goto, K., Hanakawa, N. and Iida, H.: Project Replayer - An Investigation Tool to Revisit Processes of Past Projects, *Proceedings of International Software Process Workshop and International Workshop on Software Process Simulation and Modeling (SPW/ProSim 2006)*, pp.72-79 (2006).
- 4) Inoue, K.: Software Tag: Empirical Software Engineering Data for Traceability and Transparency of Software Project, *Workshop on Accountability and Traceability in Global Software Engineering (ATGSE2007) Proceedings*, Nagoya, Japan, pp.35-36 (2007).
- 5) IPA/SEC 著作監修: ソフトウェア開発データ白書 2005~IT企業1000プロジェクトの定量データを徹底分析, 日経コンピュータ (2006).
- 6) Kamiya, T., Kusumoto, S. and Inoue, K.: CCFinder: A Multilinguistic Token-Based Code Clone Detection System for Large Scale Source Code, *IEEE Trans. on Software Engi-*

neering, Vol.28, No.7, pp.654-670 (2002).

- 7) Loon, H.V.: Process Assessment and ISO/IEC 15504: Process Assessment Standard, A Reference Book (2007).
- 8) Matsumoto, K.: A Purchaser-Centered Approach for Empirical Software Engineering : Accountability and Traceability in Global software Engineering, *Workshop on Accountability and Traceability in Global Software Engineering (ATGSE2007) Proceedings*, Nagoya, Japan, pp.33-34 (2007).
- 9) Morisaki, S., Matsumura, T., Ohkura, K., Fushida, K., Kawaguchi, S. and Iida, H.: Fine-grained Software Process Analysis to Ongoing Distributed Software Development, *In Proc. Workshop on Measurement-based Cockpits for Distributed Software and Systems Engineering Projects*, pp.26-30 (2007).
- 10) Shaw, M.A., Feldmann, R.L. and Shull, F.: Decision Support with EMPEROR, *Proc. in First International Symposium on Empirical Software Engineering and Measurement, 2007. ESEM 2007.*, pp.495-495 (2007).
- 11) 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：共通フレーム 2007～経営者、業務部門が参画するシステム開発および取引のために (2007).
- 12) XBRL Japan マーケット・アンド・コミュニケーション (マーコム) 委員会: XBRL FACT BOOK Ver.9.0 (2007).
- 13) 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, 電子情報通信学会論文誌 D-I, Vol.J88-D-I, No.2, pp.228-239 (2005).
- 14) 伏田享平: 企業間の請負開発プロセス検証を目的とした枠組みの提案, ウィンターワークショップ 2008・イン・道後 (WW2007) 論文集, pp.107-108 (2008).
- 15) 大蔵君治, 後藤慶多, 川口真司, 花川典子, 飯田元: ソフトウェア開発における知識還元のためのプロジェクト再現ツール, ソフトウェアエンジニアリングシンポジウム 2006 論文集, pp.75-78 (2006).
- 16) 経済産業省商務情報政策局情報処理振興課: 情報システムの信頼性向上のための取引慣行・契約に関する研究会～情報システム・モデル取引・契約書～ (受託開発 (一部企画を含む)、保守運用) <第一版> (2007).
- 17) 森崎修司, 松村知子, 大蔵君治, 伏田享平, 川口真司, 飯田元: エンピリカルデータを対象としたマイクロプロセス分析, 情報処理学会研究報告ソフトウェア工学研究会, Vol.2006, No.125, pp.9-15 (2006).

- 18) 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, 奈良先端科学技術大学院大学テクニカルレポート, No.NAIST-IS-TR2007002 (2007).
- 19) 肥後芳樹, 吉田則裕, 楠本真二, 井上克郎: 産学連携に基づいたコードクローン可視化手法の改良と実装, 情報処理学会論文誌, Vol.48, No.2, pp.811-822 (2007).

付 録

オープンタグ項目例

Profile Info.	
基本情報	開発組織名
	プロジェクト名
	プロジェクト期間
...	...
要員情報	延べ開発人員
	要因スキル
...	...
品質要件	効率性
	信頼性
	保守性
...	...
(プロジェクト)管理情報	開発工数
	発見バグ数
	機能数
...	...
タグ管理情報	親プロジェクト名
	サブ(子)プロジェクト数
...	...
再利用・第3者ソフトウェア	利用パッケージ
	再利用規模
...	...
Process Info.	
要件	規模(推移)
	変更(推移)
	...
設計成果物	ドキュメント変更(推移)
	変更影響度
...	...
プログラム	コード変更量(推移)
	...
テスト	テスト密度
	進化テスト項目推移
...	...
レビュー	レビュー対象規模
	レビュー指摘密度
...	...
バグ	バグ件数(推移)
	不具合消滅率(推移)
...	...
リスク	累積リスク項目数
	リスク項目の滞留時間
...	...
Product Info.	
プログラム品質	テストカバレッジ
	コードクローン比率
	...
システム品質	機能性
	信頼性
	効率性
	使用性
...	...