

ソフトウェアタグ規格

第1.0版

2008年10月14日

StagEプロジェクト
奈良先端科学技術大学院大学
大阪大学

目次

前文	1
(1) 経緯.....	1
(2) 目的.....	1
(3) 用語定義.....	3
(4) 留意事項.....	5
(5) 本規格が支援するポイント	5
(6) 本規格による効果	5
(7) 信頼性ガイドライン等とソフトウェアタグの関連	6
(8) まとめ	7
1. 本規格の構成.....	9
2. 規格の使用法（運用ガイドライン）	9
3. データ共有の目的と関連タグ項目例.....	9
3.1 製品品質の把握	9
3.2 開発状況の把握	9
3.3 紛争処理	9
4. タグの構造	10
4.1 階層構造	10
4.2 項目の繰り返し	11
5. 標準タグ項目一覧の概要.....	11
5.1 一覧表の見方.....	11
5.2 標準タグ項目の位置づけ	11
6. 標準タグ項目一覧表.....	13
付録	16
体制	16
参考文献.....	17

前文

(1) 経緯

近年、ソフトウェア開発プロジェクトにおいて、ソフトウェア発注者（ユーザ¹）の参画が重要視されてきている。従来、ソフトウェア発注者（ユーザ）は要件定義など上流工程からソフトウェア受注者（ベンダ）まかせであるケースが多かった。しかし、ユーザ不在の開発では、要件のあいまいさから設計ミスや頻繁な仕様変更が発生し、品質低下、重大な不具合によるシステムダウン、コスト増大、納期遅れなどの原因となっている。また、障害発生時には、ユーザによる原因や責任所在の追及が困難で、障害対応が遅れ、甚大な損害が発生するケースも多い。さらに、法的な係争に発展すると、裁判が長期化し、解決までユーザ・ベンダ双方の損害が増大する。一方、多重請負・オフショア開発などの複雑な開発体制や多重の受発注関係では、潜在的発注者（ユーザ）が多く存在し、統制されたプロジェクトの管理を困難にし、管理コストを増大させている。

一方、携帯電話や家電製品への組込ソフトウェアの分野では、社内ソフトウェア部品を流用した派生開発が頻繁に行われている。また、ソフトウェア開発ベンダが、品質の確保や納期短縮のため外部からパッケージソフトウェアを購入し、システムに組込む事例も多い。このようなケースでは、長期にわたる流用・改造によって設計情報などが散逸したり、開発した組織が異なるため必要な情報を得られないなどの理由で、開発が困難になったり、予期しない品質問題によってダメージを受けることも多い。

これまで、ソフトウェア開発におけるデータの収集・分析は、当該組織でのプロセス改善やプロジェクト内部の進捗・品質管理を目的とし、組織間共有や一般ユーザの視点に立って行われてこなかった。しかし、ソフトウェアの問題の社会的な影響度の拡大や、マルチベンダ開発やオフショア開発など開発組織の複雑化に対して、このような開発プロジェクトデータのより幅広い活用が求められてきている。例えば、ソフトウェアの不具合による法的紛争では、様々な開発データに基づいて責任の所在が追及される。

(2) 目的

Stage (Software Traceability and Accountability of Global Software Engineering) プロジェクト²では、ソフトウェア開発データのユーザとの共有のための基盤研究開発を目的とし、ソフトウェアの品質や由来をユーザに示すための技術として「ソフトウェアタグ」の研究・開発を行っている。ソフトウェアタグは、ソフトウェアの開発中に収集された管理データや品質情報など複数のデータから構成され、各データ種類をタグ項目と呼ぶ。タグ項目に従って、開発成果物や副次的に発生するデータ（実証データ/エンピリカルデータ）からタグデータが開発中、もしくは開発完了時に作成され、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、設計書やプログラムなどの成果物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。ソフトウェアタグは、プロジェクト進行中にプ

¹ 本規格では、ソフトウェア開発関係者として、主に「ユーザ」「ベンダ」という2つの言葉を用いる。特記しない限り、「ユーザ」はソフトウェア発注者、ソフトウェア利用者（エンドユーザ）などを含み、「ベンダ」はソフトウェア受注者、ソフトウェア開発者などソフトウェアを供給する人・組織を示す。（詳細は、「用語定義」参照）

² 文部科学省のプロジェクト「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」の略称

プロジェクト管理のための定量的データを共有するために用いることも可能で、プロジェクト進行中に発生する問題の早期発見・解決が可能になる。また、ユーザ・ベンダ間の紛争処理では実際の開発データに基づいた詳細かつ正確な検証が可能になる。

「ソフトウェアタグ規格」は、ソフトウェアタグに含まれることが望ましい標準的なデータ項目とタグの運用ガイドラインを提供する。例えば、ソフトウェア発注者（ユーザ）と受注者（ベンダ）は、本規格に基づいて、両者で共有すべきデータを取り決め、どのように運用・管理するかといった具体的な活用方法を検討することができ、ユーザ・ベンダ両者が参画する定量データによるプロジェクト管理の導入が容易になる。また、収集データの標準化によりデータの組織間の共有が容易になり、統一基準による適正な評価も促進される。

以下に、本規格の構築に先立って検討されたソフトウェアタグの具体的な利用目的を挙げる。利用目的は、プロジェクトやシステムに応じて異なり、下記に限定されるものではないが、産業界において必要性の高い場面を想定している。

- 製品品質の把握
 - 流用資産の保証書／部品として使う製品と高信頼システム全体の信頼性認証：開発プロセスとプロダクトの動作に関する試験結果を監査し、製品及び高信頼システムを認証するために用いる。（図1参照）
 - 製品情報：工業製品や農業製品についているプロフィール情報（生産者、原産地、原材料、添加物など）のソフトウェア版。使っている基盤ソフト、パッケージソフト、開発言語、開発規模、開発者（企業名）、サービス提供者などのほかに、試験結果など品質情報もオプションとして加えても良い。
- 開発状況の把握
 - 要件定義の充実度：要件定義が曖昧であることがプロジェクトの成否の鍵を握るといわれているが、定量的な分析はあまりない。要件定義の充実度を図るメトリクス及びプロジェクトの成功/失敗についてのカテゴリを定義することで、どのような要件定義をすればよいか、明らかにする。仕様変更などもプロジェクトの成否にかかわる有益な情報である。
 - 開発中の情報共有：開発中や検収時に、ユーザ（顧客）に対して示すプロジェクトデータをタグとして共有する。納品時の事後分析だけでなく、開発中の進捗状況の把握などにも用いことができる。（図2参照）
- 紛争処理
 - 事故調査：ハードウェアの世界では一般的な事故調査委員会（飛行機/電車事故など）が検査するためのデータのソフトウェア版。原因を究明して、プロセス問題に帰着させ、プロセス改善に結びつける。
 - 法的係争時のプロセスの証拠：障害や事故の責任の所在を明らかにする。プロセス遵守度・プロセス規定度・プロセス管理度など。（図3参照）

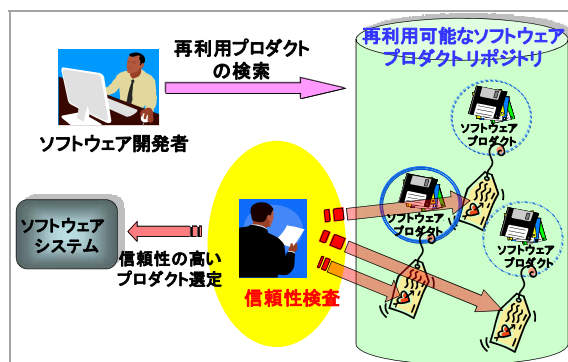


図 1 流用資産の保証書／部品として使う製品と高信頼システム全体の信頼性認証

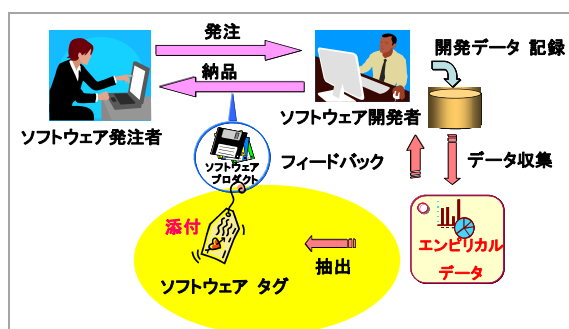


図 2 開発中の情報共有

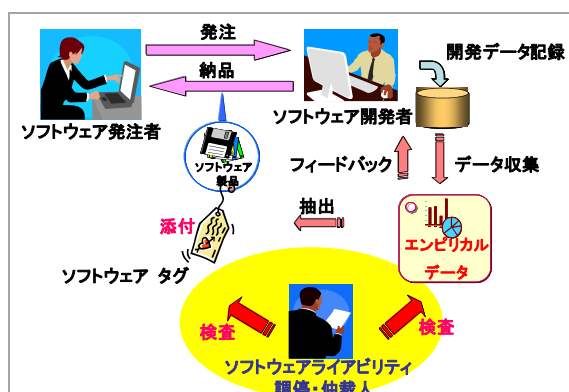


図 3 法的係争時のプロセスの証拠

(3) 用語定義

データ

記録の形態や方法にかかわらず、記録された情報、もしくはそれらから加工・抽出・算出された情報。

定量的データ

数値化されたデータ。規模や件数など対象物の計測で得られるデータのほか、欠陥密度など複数のデータから導出されるデータを含む。また、重要度、難易度などの定性的な属性も、集計などによって定量的に扱われる場合、定量的データに含まれる。

実証データ（エンピリカルデータ）

対象の実測により得られるデータ。ソフトウェア開発プロジェクトにおいては、最終成果物であるシステムから得られるデータだけでなく、開発過程で生じる副次的なデータ、例えば議事録、欠陥レポート、工程管理表なども含まれる。

メトリクス

定量的な評価尺度。

プロジェクト

一般的には、目標達成のための計画から実施を含む作業の単位。本文中では、ソフトウェア開発プロジェクトの意味で用いるが、必ずしもシステムの構築が目標ではなく、要件定義、設計など工程別に行われるプロジェクトも対象とする。標準タグ項目での「プロジェクト名」「サブプロジェクト」などは、タグの作成・流通単位としてのプロジェクトであり、ソフトウェア開発プロジェクト全体とは異なる狭義の意味を持つ。

プロダクト（成果物／製品）

ソフトウェア開発プロジェクトでは、顧客や最終利用者への納入のための作業成果物。プログラム、品質保証データ、設計書、マニュアルなど、プロジェクトによってプロダクトは異なる。

ベンダ

ソフトウェア開発における狭義のベンダを意味する。ソフトウェア開発における「ベンダ」は、ソフトウェア受注者、ソフトウェア開発者などソフトウェアを供給する人・組織を示す。本文中では、パッケージソフトウェアの販売代理店など、実際に開発過程にかかわらない組織は、対象としない。

ユーザ

委託開発における発注者、システムのエンドユーザ、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挙げられる。

発注者・受注者

2者間契約による委託開発における受発注関係にある組織を指す。これは、ユーザと1次請けベンダの関係だけではなく、ベンダからの外部委託などの関係も含む。従って、1組織が2つの立場（発注者であり、受注者）になる場合もある。

ソフトウェアタグ

ソフトウェアの開発中に収集された管理データや品質情報をパッケージ化したもの。ソフトウェア製品に付随して流通し、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、設計書やプログラムなどの成果物の品質情報を入手することができる。

タグ項目

ソフトウェアタグを構成する個々のデータ種類。

タグデータ

ソフトウェアタグに含まれる実際の値や情報。基本的に、実証データ（エンピリカルデータ）から抽出されるが、実証データそのものを用いることも可能である。タグデータは、実証データによって裏づけが可能なデータであることを前提とする。

(4) 留意事項

汎用性

本規格は、できるだけ汎用的な枠組みとするため、特定のプロジェクト・分野の個別性の高い内容は含めない。タグ項目などに具体性に欠ける場合、説明のための詳細情報として、具体例を示す。特定の目的・分野ごとに必要な具体化は、利用シナリオで行う。

強制力

本規格は、ユーザ・ベンダ間（受発注者間）での協議のベースなどに用いる枠組みを提供するものであり、そのまますべてのプロジェクトやソフトウェアに導入を強制するものではない。また、本規格の利用者による追加・変更・削除などを制限しない。

具体例の活用

本規格には、汎用性を保つため抽象的なレベルでとどめている記述がある。プロジェクトでの活用を容易にするため、データの具体的なデータ収集・加工の手法/手段として具体化例や実証データ例を示す。ただし、例示するにとどめ、すべての具体化方法を網羅するものではない。

(5) 本規格が支援するポイント

① データ収集

ソフトウェア開発プロジェクトにおいて、収集すべきデータを明確にする。このデータは、プロジェクトの進捗やコストをリアルタイムに管理し、納期の遅延やコスト超過を防止したり要求されている品質を保証したりするために、開発側（ベンダ）が必要とする情報をベースとするが、特にユーザと共有することが望まれるデータ（紛争時の解決に有効なデータなど）についても取り上げる。

② データ加工・可視化

本規格では、ユーザと効果的にデータを共有するために、収集されたデータを加工・可視化する方法や技術の例を示す。収集されるデータは膨大な量であり、そのまま開示しても分析や解釈のコストが増大し、効率的・効果的にデータが共有できない。そのため、ユーザと共有するデータは、双方の目的に応じて必要最小限の量であることが望ましい。本規格では、開発中の収集データからどのような定量的データを加工し、そのデータをどのようにユーザに見せるかといった点について、例を提供する。

③ ユーザ・ベンダ間のデータ運用

本規格では、ユーザ・ベンダの目的やプロジェクトの特性に合わせたデータ運用の枠組みを提供し、ユーザ・ベンダ間でどのようにデータを用いたプロジェクト・ソフトウェア管理を行うか、その手順を明確にする。具体的には、目的に応じた収集データの選択、契約への記載、データ共有タイミング、データ授受方法、それに伴うコストなどである。

(6) 本規格による効果

① データ収集の促進とデータ項目の標準化

一般的なデータ項目と収集・加工方法などの情報の利用により、データ収集の導入が容易になり、定量的プロジェクト管理・プロセス改善が促進される。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され、計測や分析のコストが低減する。

② ユーザ・ベンダ間のデータ共有の促進

組織間のデータ項目の統一により、データ共有・流通が容易になる。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、国際競争力が強化される。

- ③ ユーザのデータ収集・分析への理解・協力
データ共有の目的の明確化やプロジェクト管理への理解向上により、ユーザの協力（コスト負担など）が得やすくなる。
- ④ 障害対応や紛争処理
あらかじめ収集されているデータが明確になり、要約されたデータが共有されていることによって、障害発生時には、ユーザ・ベンダ間の協力による迅速な原因究明・問題解決が可能になる。また、ユーザ・ベンダ間の紛争に発展した場合にも、公正かつ迅速な解決が図れる。

(7) 信頼性ガイドライン等とソフトウェアタグの関連

経済産業省は、情報産業強化施策の一環として、「産業構造・市場取引の可視化」³を挙げ、ユーザ・ベンダ一体となった生産性向上の促進を目指している。「情報システムの信頼性向上に関するガイドライン」⁴（以降、信頼性ガイドライン）では、“情報システムの利用者及び供給者の双方が応分の責務を担う”とし、「企画・開発における、信頼性・安全性の水準検討」「障害発生時の手順の文書化・記録」「定量的な手法を取り入れたプロジェクトマネジメントの実行」などについて、受発注者双方の果たすべき役割について述べている。それを受けて、「情報システムの信頼性向上に関する評価指標（試行版）」⁵（以降、信頼性評価指標）では、ユーザ・ベンダ双方のガイドラインの遵守状況を評価のための評価指標を示し、具体的なチェックリストを公開している。一方、「情報システム信頼性向上のための取引慣行・契約に関する研究会」最終報告書⁶（以降、モデル契約書）では、ユーザの主体的なプロジェクトへのかかわりを促進することを目的として、契約書にユーザやベンダの役割・責任分担を明確に示すことが重要である、と述べている。契約に明記することによって、ユーザによるベンダへの“丸投げ”を防ぎ、仕様の固まりにくいプロジェクトの円滑な進行を促進するという効果を期待している。また、モデル契約書では、ユーザ・ベンダ間でのソフトウェア開発に関する紛争解決方法として、裁判外紛争処理（ADR）の活用を推奨している。専門的・技術的視点による詳細な事実認定や当事者間の話し合いによって、紛争処理を迅速かつ柔軟に行うことが可能である、と考えられている。

ソフトウェアタグは、これらのガイドライン等に対して、それぞれの目的に応じて具体的なデータに基づいて支援することが可能である（図 4 参照）。信頼性ガイドラインに挙げられている「定量データを活用した管理」は、本規格の標準タグ項目をベースにユーザ・ベンダ間で検討することによって、導入が容易になると考えられる。また、ソフトウェアタグとして収集されたデータを用いて、ガイドラインの遵守状況をより詳細に評価することが可能になる。一方、ソフトウェアタグに関する条項を契約書に盛り込むことによって、プロジェクト管理に関するユーザ・ベンダ双方の責任・役割を明確にすることができ、データ収集に伴うコストなども契約時に協議することが可能になる。一方、ADR による紛争処理がおこなわれる際には、実際の開発データに基づいた詳細かつ正確な検証が可能になる。

³ http://www.meti.go.jp/policy/it_policy/softseibi/index.html

⁴ 2006年6月15日公表，経済産業省

⁵ 2007年4月13日公表，経済産業省

⁶ 2007年4月13日公表，経済産業省

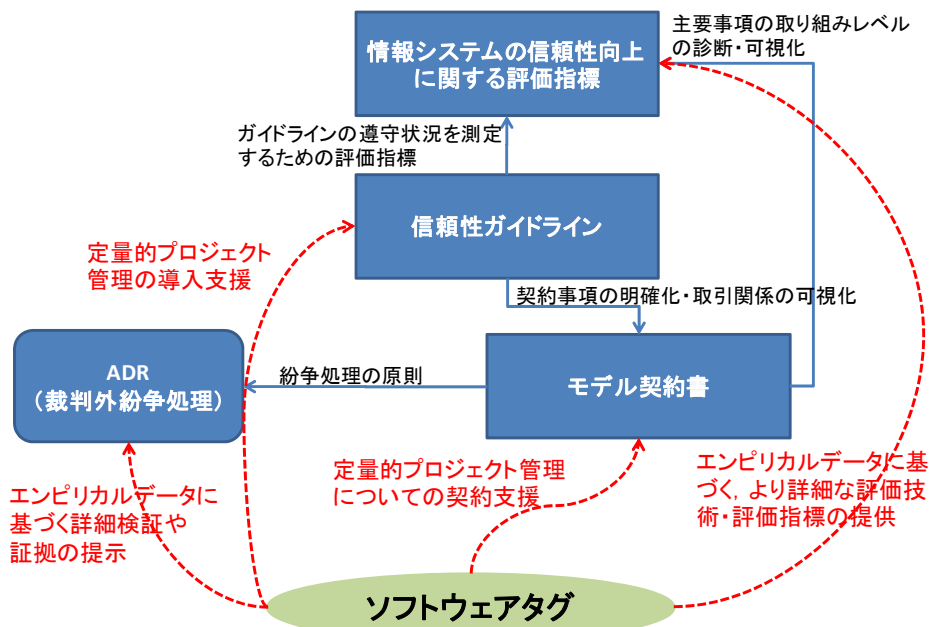


図 4 情報政策とソフトウェアタグの関連

一方、JISA(社団法人情報サービス産業協会)では、「信頼性向上ベストプラクティスを実現する管理指標調査報告書」⁷を刊行し、ベンダ視点から信頼性向上のための具体的な指標の効果や活用方法を示した。この中で、顧客や協力会社との対話における指標活用のポイントとして、「活用目的と意義の共有」「活用する指標の早期合意の重要性」「定例会議での指標の報告・確認と議論の効果」などが挙げられている。

ソフトウェアタグ規格は、ユーザに重点を置いた指標を選択することによって、組織間のデータ共有・活用を支援すると考えられる。

(8) まとめ

上記のとおり、ソフトウェアタグは、これまでのソフトウェア開発プロジェクト管理を大きく変更するものではなく、ユーザ視点によるデータ活用を促進するために、既存の技術や知識を再構成することによって実現される。本規格は、その具体的な方法、すなわちデータの種類や活用方法を検討するために、ユーザ・ベンダ間で用いられる 1 つの指針として作成される。

ソフトウェアタグに含まれるデータ項目やデータ構造などは、本規格で具体的に示されているが、今後の課題として以下のことが考えられる。

- 実プロジェクトにおいてどんなタイミングで情報を取るのかなど、本規格のユーザが運用の具体的なイメージを持つための、利用シナリオの充実
- 本規格に準じたソフトウェアタグの普及を促進するため、ユーザサイドへの啓発活動の促進（経済産業省、SEC などとの連携や、タグの認証や検証を行う第三者機関の設立など）
- 国際規格化
- データ収集・分析ツールの開発や既存ツールを活用した導入や普及の支援
- 本規格を保守・運用・管理する体制の整備

⁷ 2008年4月、社団法人情報サービス協会。概要は、以下 URL 参照。
<http://www.jisa.or.jp/report/2007/19-J009.pdf>

1. 本規格の構成

(1) 規格の使用方法（運用ガイドライン）

ソフトウェア開発プロジェクトの企画段階で、ユーザ・ベンダ間でのデータ共有に関する取り決めを行うにあたっての観点・手順を示す。

(2) データ共有の目的と関連タグ項目

ユーザ・ベンダ間で共有するデータの項目を決定するために、データ共有の主要な目的とそれに関連する標準的なタグ項目の例を示す。実際には、より詳細な目的を設定し、目的に応じた具体的なタグ項目を選択することが必要である。

(3) タグの構造

開発プロジェクトの管理や部品の流用・再利用を考慮した、推奨するソフトウェアタグ構造を示す。

(4) 標準タグ項目一覧の概要

(5)の標準タグ項目一覧表の見方と、運用上の位置づけを示す。

(5) 標準タグ項目一覧表

標準的にプロジェクト管理に用いられるタグ項目の一覧表を示す。

2. 規格の使用方法（運用ガイドライン）

ソフトウェア開発プロジェクトの企画段階で、ユーザ・ベンダ間でのデータ共有に関する取り決めを行うにあたっての観点・手順を示す。また、各ステップで参照すべき本規格の章を示す。

(1) データ収集・共有目的の設定

「3. データ共有の目的と関連タグ項目例」参照

(2) 目的に応じたデータ項目の選択

「3. データ共有の目的と関連タグ項目例」「5. 標準タグ項目一覧の概要」「6. 標準タグ項目一覧表」参照

(3) データ定義・収集方法・開示データレベルの決定

「5. 標準タグ項目一覧の概要」「6. 標準タグ項目一覧表」参照

(4) システム・プロジェクト構成に応じたタグ構造の決定

「4. タグの構造」参照

(5) データの収集・計測・開示タイミングの決定

(6) 開示データの活用方法（説明，議論，解釈/判定，フィードバック）の決定

(7) 開発完了後（保守，運用，再利用等）におけるタグ活用の検討

3. データ共有の目的と関連タグ項目例

ユーザ・ベンダ間で共有するデータの項目を決定するために、データ共有の主要な目的とそれに関連する標準的なデータ項目の例を示す。実際には、より詳細な目的を設定し、目的に応じた具体的なデータ項目を選択することが必要である。

3.1 製品品質の把握

- 品質の把握に関する項目：バグ数，レビュー状況，静的解析結果など
- ソフトウェアを部品としての再利用するために必要な情報に関する項目：システム構成，テスト項目数，リスク項目数など

3.2 開発状況の把握

- 開発環境（リソース等）に関する項目：開発体制，開発手法など
- プロセス情報に関する項目：要件/設計変更，テスト消化，プロセス管理情報など

3.3 紛争処理

- 試験に関する項目：バグ数，レビュー状況，テスト項目数など

- ユーザ・ベンダ間合意に関する項目：ユーザヒアリング情報、要件変更、会議実施状況など

4. タグの構造

開発プロジェクトの管理や部品の流用・再利用を考慮した、推奨するソフトウェアタグ構造を示す。

4.1 階層構造

ソフトウェアタグは、システムの構成やプロジェクトの契約単位、開発チーム構成などに応じて階層構造で持つ。

図 5(a)のシステム A のように 2 つのサブシステム X,Y から構成されるシステムの場合、ソフトウェアタグは 1 つのメインタグとそれにリンクする 2 つのサブ (子) タグから構成することができる。このように進捗管理や品質管理の単位で構成すると、収集・分析データの流用も容易であり、タグの管理コストが低減できる。

また、タグを階層構造にすると、サブシステムの再利用や流用も容易になる。図 5(b)は、サブシステム Y を他のシステム B の部品として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)から新規に作成され、システム B への組込み作業のデータを持つ。このような構造をとることによって、流用部品の過去のデータ管理が容易になり、ソフトウェア部品のトレーサビリティを確保できる。

開発されたシステムの運用・保守を別契約で行う際も、図 5(c)のようにタグを階層構造にすることが望ましい。この場合、メインタグとサブタグで重複する情報を省略し、各サブタグは、運用・保守プロセス作業で発生するデータのみを持つことになる。

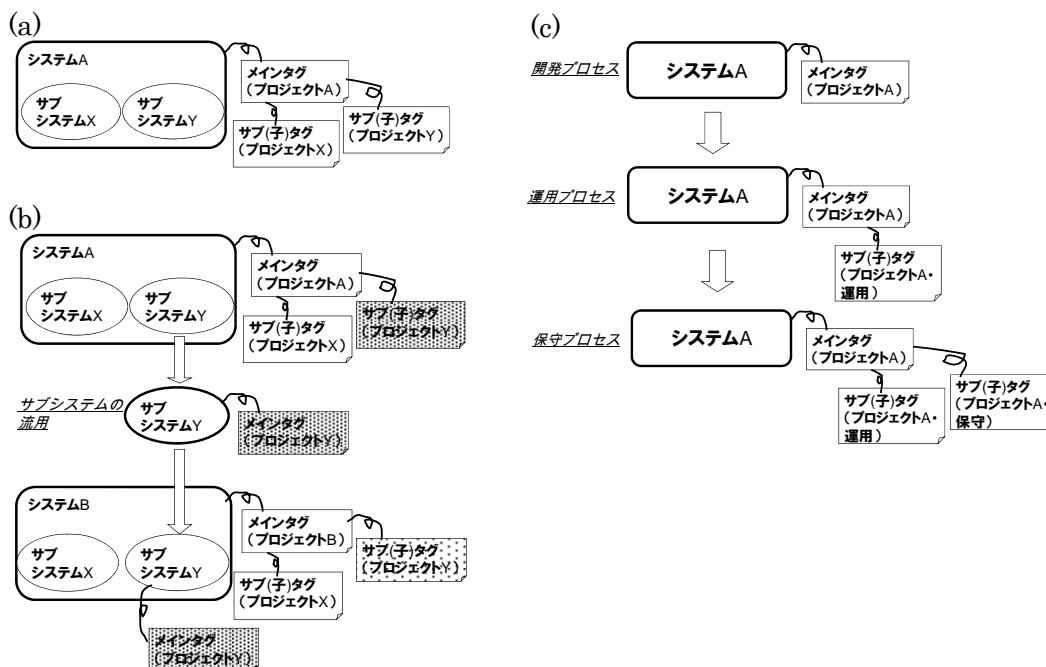


図 5 タグの構造

(a)システム構成時のタグ構造 (b)流用時のタグ構造 (c)保守・運用時のタグ構造

4.2 項目の繰り返し

タグ項目は、開発工程やプロダクトの性質に応じて、複数持つことが適切な場合、同一項目の繰り返しを許容する。例えば、進捗情報の「設計」分類に関して、基本設計、詳細設計などの工程が存在する場合、「設計」は同一タグ内に複数存在するのが適切と考えられる。また、進捗情報の「プログラミング->規模」についても、派生開発などで新規作成・改造・再利用（流用）など種別に計測することが必要な場合、「プログラミング->規模」項目が3つ存在することを許す。

5. 標準タグ項目一覧の概要

標準的にプロジェクト管理に用いられるタグ項目の一覧と、運用上の位置づけを示す。

5.1 一覧表の見方

プロジェクト情報

プロジェクトに関する情報・プロジェクト中に不変なデータ・プロジェクトを特徴付けるタグ

※下記一覧は、ユーザ・ベンダ間で共有するタグ項目の候補である。このなかから目的に応じてタグ項目を選択し、ソフトウェアタグを構成する。
 ※タグのデータを解釈・分析するために必要なデータや他のプロジェクトと比較するための情報を含む。
 ※必要に応じて、具体的なデータの取り方の情報を含む。

分類	項番	タグ項目	説明	具体化例	実証データ例	予定・実績の要否	工程別の要否	備考
基本情報	1	プロジェクト名	プロジェクトを一意に決定するための識別名		プロジェクト計画書 など			必須項目
	2	開発組織の情報	当該プロジェクトの開発を担当する組織の情報。一般には、受注者となる組織情報となる。	ISO/IEC 15504 Process Assessment, 米カーネギーメロン大CMMIなどを用いたアセスメント結果のプロセス達成度一覧または組織成熟度レベル 開発組織名, 対象業種の経験 など	発注仕様書 アセスメント結果に関する情報 体制表 キャリアシート など			類似対象業種のプロジェクトに関するプロセス領域ごとの能力レベルの達成度など、合意のもとに詳細を共有することも可能
	3	開発プロジェクト情報	開発プロジェクトの特徴や当該タグデータの対象とするプロジェクトの種類を示す情報。タグデータの解釈や分析時に必要なデータ。	開発プロジェクト種別: 新規・改造・保守・運用・流用など 設計書再利用率・ソースコード再利用率・コンポーネント再利用率など 開発プロジェクト形態 (商用パッケージ, 受託開発), 利用パッケージ など	要件定義書 プロジェクト計画書 品質計画書 設計ドキュメント ソースコード テスト計画書 など	○		開発作業を伴わない場合も入力 進捗情報の各成果物の規模から導出可能な場合もある

*注) 進捗情報タグ項目には、「工程別の要否」欄はない。進捗情報は、管理単位を自由に設定することが可能で、工程別・日次・週次などが考えられる。

5.2 標準タグ項目の位置づけ

- 利用目的やプロジェクトの特性に応じたカスタマイズを前提としている。従って、すべての項目がすべてのプロジェクトにおいて網羅的にタグ化されることを求めない。また、必要なデータの追加も可能とする。
- データ収集の対象物、対象範囲、粒度、収集期間（工程）は、別途ユーザ・ベンダ間で取り決めなどに準じ、本規格では特に規定しない。
- 収集データのタグ化のタイミングやユーザへの開示タイミングは、ユーザ・ベンダ間の協議などにより決定されるものとする。納品時、工程毎、一定時間ごと（週次、日次）などが考えられる。
- タグの階層や項目の繰り返しなどのタグの具体的な構造については、システムやプロジェクトに応じてカスタマイズをする。（「4. タグの構造」参照）

- ソフトウェアタグとするデータの具体化レベルは、選択可能とする。エンピリカルデータ・生データ、一次加工データ（数値化されたデータ）、二次加工データ（2つの測定値の比率など）が考えられる。たとえば、進捗情報の「プログラミング->規模」は、プログラムコードそのもの、時系列のステップ数、予定規模に対する完成割合などが考えられる。この点についても、ユーザ・ベンダ間で協議などにより決定されるものとする。
- 他の項目から算出可能な数値は、項目から除外する。ただし、欠陥密度など一般的なものは1項目として挙げている。
- 他の文書（契約書等）に記載され、ユーザ・ベンダ間ですでに共有されている内容（品質要件、受発注金額など）は標準タグ項目から除外しているが、変更管理などのため追加することも可能である。

6. 標準タグ項目一覧表

タグ項目一覧

プロジェクト情報

プロジェクトに関する情報・プロジェクト中に不変なデータ・プロジェクトを特徴付けるタグ

※下記一覧は、ユーザ・ベンダ間で共有するタグ項目の候補である。このなかから目的に応じてタグ項目を選択し、ソフトウェアタグを構成する。
 ※タグのデータを解釈・分析するために必要なデータや他のプロジェクトと比較するための情報を含む。
 ※必要に応じて、具体的なデータの取り方の情報を含む。

分類	項番	タグ項目	説明	具体化例	実証データ例	予定・実績の要否	工程別の要否	備考
基本情報	1	プロジェクト名	プロジェクトを一意に決定するための識別名		プロジェクト計画書 など			必須項目
	2	開発組織の情報	当該プロジェクトの開発を担当する組織の情報。一般には、受注者となる組織情報となる。	ISO/IEC 15504 Process Assessment, 米カーネギーメロン大CMMI®などを用いたアセスメント結果のプロセス達成度一覧または組織成熟度レベル 開発組織名, 対象業種の経験 など	発注仕様書 アセスメント結果に関する情報 体制表 キャリアシート など			類似対象業種のプロジェクトに関するプロセス領域ごとの能力レベルの達成度など、合意のもとに詳細を共有することも可能
	3	開発プロジェクト情報	開発プロジェクトの特徴や当該タグデータの対象とするプロジェクトの種類を示す情報。タグデータの解釈や分析時に必要なデータ。	開発プロジェクト種別: 新規・改造・保守・運用・流用など 設計書再利用率・ソースコード再利用率・コンポーネント再利用率など 開発プロジェクト形態 (商用パッケージ, 受託開発), 利用パッケージ など	要件定義書 プロジェクト計画書 品質計画書 設計ドキュメント ソースコード テスト計画書 など	○		開発作業を伴わない場合も入力 進捗情報の各成果物の規模から導出可能な場合もある
	4	顧客情報	当該システムのユーザ、もしくは第1発注者となる組織の情報。	顧客名, 新規顧客か否か など	顧客との議事録 プロジェクト計画書 など			通常は、発注者自身なので不要。ただし、受注者からの外部委託を行う場合、最終顧客(エンドユーザ)情報を記入
システム情報	5	システム構成	開発システム構成の特徴や当該タグデータの対象とするシステムの種類を示す情報。タグデータの解釈や分析時に必要なデータ。	利用したサブシステムの安定度: 利用ハードウェア、ライブラリ、OS等、調達したサブシステムの安定度合い(初回、不安定等) サブシステムの検証期間/工数: 利用を検討したサブシステムの検証期間と工数 利用したサブシステムに関する法的要件 など	基本設計書 プロジェクト計画書 工程管理表 勤務実績データ など			はじめて使うサブシステムには学習工数が必要のため 検証時に性能把握や設計情報の細部を詰めることができるため、検証期間/工数から設計、テスト品質を推測するため
	6	システム規模	開発システムの規模、計画値と最終実績値とする。進捗情報に同じ情報が含まれる場合は、省略可。	受注者側の想定する顧客要件の数, 実装した要件数など 工程終了時のソースコード行数, 機能, FP, 処理データ量, 処理データ数など	基本設計書 ソースコード 要件定義書 など	○	○	SLOCはプログラム単位, その他はプロジェクト単位
開発情報	7	開発手法	開発システム開発に用いたプロセスや手法についての情報。タグデータの解釈や分析時に必要なデータ。	適用プロセスタイプ (ウォーターフォール・アジャイル・プロトタイプ), 適用手法 (OOAD, SASD, DDE) など	プロジェクト計画書 プロジェクト生産管理計画書 など			
	8	開発体制	開発側の要員に関する情報。タグデータの解釈や分析時に必要なデータ。 ※開示対象範囲は、発注者・受注者側での協議により決定する	組織図・人員配置図 作業体制の複雑さ・親密さ(結合度・分散度): 組織図・人員配置図から計測 プロジェクト体制の階層数 など 外注率, ピーク時の開発人員, 延べ開発人員, チーム別の開発要員 など 経験年数, レベル別割合(上級・中級・初級), 新人比率, 類似プロジェクトの経験(開発, ユーザともに類似プロジェクト参加の経験があるか: 経験なし, 3回未満, 5回以上), 要員スキル (PM, 業務, 分析, 設計, 言語・ツール, 開発プラットフォーム: ITSSなどに準拠) など	体制表 発注書 キャリアシート 勤務実績データ 工程管理表 など		○	担当機能毎, 担当職務毎メンバーの学習工数を推測するため
	9	プロジェクト期間	当該プロジェクトの開発期間に関する情報	稼働時期, 開始, 終了を含む(時間, 日, 月) など 工期 など	勤務実績データ 工程管理表 など	○	○	発注から納品までのリードタイム(=全体工期)
	10	親プロジェクト情報	本プロジェクトが別のプロジェクトのサブ(子)プロジェクトである場合, 付加	親プロジェクトのプロジェクト名	プロジェクト計画書 機能設計書 構造設計書 など			「No.1 プロジェクト名」と一致するプロジェクトのタグが存在しなければならない
11	サブ(子)プロジェクト情報	本プロジェクトがサブ(子)プロジェクトを持つ場合, その数やサブ(子)プロジェクトに関する情報	サブ(子)プロジェクトの数, プロジェクト名など	プロジェクト計画書 機能設計書 構造設計書 など				
その他	12	特記事項	その他, タグデータの解釈や分析時に必要, もしくは有用なデータ。					タグデータ管理情報(適用管理指標, 収集データ群)など

進捗情報

成果物や作業の進捗状況、成果物やプロセスの品質を表すタグ

※下記一覧は、ユーザ・ベンダ間で共有するタグ項目の候補である。このなかから目的に応じてタグ項目を選択し、ソフトウェアタグを構成する。
 ※[推移]は、時系列もしくは一定間隔で計測された連続データも想定される。
 計測間隔は、工程毎・日次・週次・開発のm/n(n=0,1,2,...,n-1 / n=16 or 32 ...)などが考えられる
 ※進捗に関する各計測項目は、必要に応じて全体／工程毎／タスク毎に測定する。
 ※必要に応じて、具体的なデータの取り方の情報を含む。

分類	項番	タグ項目	説明	具体化例	実証データ例	予定・実績の異否	備考
要件定義	13	ユーザヒアリング情報	要件に関してユーザに行ったヒアリングに関する情報	ユーザヒアリング実施件数(回) ユーザヒアリング項目数(件)、 ユーザヒアリング回答率(ユーザヒアリング回答数÷ユーザヒアリング項目数) など	ユーザヒアリング議事録 ユーザヒアリング質問票 など	○	
	14	規模[推移]	開発側で作成した要件数	画面、機能項目、ユースケース、アクター、顧客要件、機能、FPなど	要件定義書 など	○	何を要件の基本単位とするかは、要合意事項
	15	変更[推移]	変更された要件数	規模の計測単位に依存	要件定義書 要件定義書の変更履歴 など		
設計	16	規模[推移]	設計成果物の規模 ※新規・改造・再利用(流用)毎に計測する	機能設計(ページ数・帳票数・画面数・ファイル数・項目数・UML図の数、クラス数、パッチプログラム数、重要な機能数など) 構造設計(データ項目数、DFDデータ数、DFDプロセス数、DBテーブル数など) など	基本設計書 機能設計書 構造設計書 詳細設計書 など	○	
	17	変更[推移]	変更された設計成果物の数、もしくは変更量	規模の計測単位に依存	基本設計書 機能設計書 構造設計書 詳細設計書 各設計書の変更履歴 など		
	18	要件の網羅率	要件定義で作成された要件の実装率	設計に取り入れられた要件数÷要件数	要件定義書 基本設計書 機能設計書 構造設計書 詳細設計書 など		
	19	規模[推移]	プログラミング成果物の規模 ※新規・改造・再利用(流用)毎に計測する	LOCなど	ソースコード など	○	ファイル別・モジュール別
プログラミング	20	変更[推移]	変更されたプログラムの数、もしくは変更量	変更回数、削除行数、追加行数など	ソースコード リポジトリ更新履歴 など		ファイル別・モジュール別
	21	複雑度	プログラムの品質(保守性)	Halstead, McCabe, CK, クロウン含有率, コメント平均利用率など	ソースコード など		ファイル別・モジュール別
	22	規模[推移]	テストの規模 ※新規・改造・再利用(流用)毎に計測する	テスト項目数など	テストケース テスト計画書 など	○	
テスト	23	変更[推移]	変更されたテスト項目数や変更量	追加テスト項目数、変更テスト項目数など	テストケース 変更履歴 など		
	24	密度	テストの品質	テスト項目数÷システム規模(ソースコード行数など) C0(命令網羅), C1(分岐網羅), 要件に対するカバレッジ率(結合/受け入れテスト)など	テストケース テスト結果 ソースコード 要件定義書 など	○	「No.6 システム規模」と「No.22 規模」から導出可
	25	消化	テストの進捗, プログラムの品質	テストケース進捗, 消化テスト項目推移, テストプログラム消化率, クリティカルパス進捗など	テストケース テスト計画書 テスト結果 工程管理表 など	○	
	26	レビュー状況	成果物(仕様書, 設計書, プログラムコード, テスト仕様書など)のレビューに関する情報	レビュー回数, 対象規模, 実績回数, ユーザ参加の有無, レビュー時間など	レビュー議事録 など	○	
品質	27	レビュー作業密度	レビュープロセスの品質, もしくはレビュー対象の品質	レビュー工数率, レビュー密度など	レビュー議事録 レビュー対象(ソースコード, 各設計書) など		
	28	レビュー指摘率[推移]	レビュープロセスの品質, もしくはレビュー対象の品質	レビュー指摘数, 指摘密度, 不具合指摘数推移など	レビュー議事録 レビュー対象(ソースコード, 各設計書) など		
	29	欠陥件数[推移]	テスト設計の品質とコード品質		テスト結果 障害管理票 品質管理表 など	○	欠陥が発生した工程別, 原因ごとにも集計
	30	欠陥対応件数	欠陥の対応進捗, 対応内容	不具合消化数, 障害滞留時間, 検出欠陥の分析実施回数, 類似欠陥調査実施回数など	テスト結果 障害管理票 品質管理表 など		
	31	欠陥密度	テスト設計の品質とコード品質	検出欠陥数をシステム規模で割る:不良検出件数÷ステップ数など	テスト結果 障害管理票 品質管理表 ソースコード など	○	「No.6 システム規模」と「No.29 欠陥件数」から導出可

	32	欠陥指摘率	テスト設計の品質	不良抽出件数÷テストケース消化数	テスト計画書 テストケース テスト結果 障害管理表 品質管理表 など	○	「No.25 消化」と「No.29 欠陥件数」から導出可
	33	静的チェックの結果	プログラムの品質(保守性)	チェックによる総エラー数、チェックによる総警告数など	ソースコード など		
工数	34	作業工数	作業に要する工数、仕様変更作業工数		工程管理表 勤務実績データ など	○	
	35	生産性	工数に対する成果物の比率	規模(画面数・帳票数)÷工数 頁÷工数(設計工程) ステップ数÷工数(製造工程) 不良抽出件数÷工数(テスト工程) など	構造設計書 勤務実績データ ソースコード テスト結果 など	○	「No.34 作業工数」と各成果物の規模などから導出可
計画・管理	36	プロセス管理情報	開発プロセスの管理に関する情報	プロセス規定度(提出されたプロセス記述の詳細さに基づく:WBSのタスク数など)、標準プロセスに対する網羅度、プロセス遵守度(プロセス記述と実際の作業間の適合の度合いに基づく)、プロセス管理度(各ステップごとに設定された会議体の数や提供された管理指標の数に基づく)など	プロジェクト計画書 プロジェクト生産管理計画書 勤務実績データ 工程管理表 議事録 など		
	37	会議実施状況	ユーザー・ベンダ間、ベンダ間での情報共有状況を把握	各会議の時間、参加人数、資料量、議事数、議事録量、情報共有者数など	議事録 など		会議:要件検討会、進捗会議、納入/検収など
	38	累積リスク項目数	リスク認識が十分であったかを把握	進捗会議で挙げられたリスク項目数の累積、軽微、重大、その他の項目でわかる	議事録 リスク管理表 など	○	
	39	リスク項目の滞留時間	リスク対策が適切になされていたかを把握	リスク管理項目の最長、平均滞留時間	議事録 リスク管理表 など		
その他成果物	40	規模[推移]	対象成果物の規模 ※新規・改定・再利用(流用)毎に計測する		ドキュメント 計画書 など	○	対象成果物:マニュアル、計画書、スケジュール表など
	41	変更[推移]	変更された対象成果物の数、もしくは変更量	変更回数、変更ページ数など	ドキュメント変更履歴 など		

付録

体制

- ソフトウェアタグ規格技術委員会
第1回（2007年7月9日） ～ 第13回（2008年10月14日）
構成員 14組織 31人

アドバイザー（50音順・敬称略）

大杉 直樹	株式会社NTT データ
木谷 強	株式会社NTT データ
鶴保 征城	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）
神谷 芳樹	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）
村山 浩之	株式会社デンソー

企業（50音順・敬称略）

岩崎 新一	日本電気株式会社
片平 真史	独立行政法人宇宙航空研究開発機構（JAXA）
清田 辰巳	株式会社東京証券取引所
阪井 誠	株式会社SRA 先端技術研究所
高木 富士雄	シャープ株式会社
福地 豊	株式会社日立製作所
松尾 昭彦	株式会社富士通研究所
松村 好高	株式会社SRA 先端技術研究所
宮本 祐子	独立行政法人宇宙航空研究開発機構（JAXA）
森 俊樹	株式会社東芝
山田 淳	株式会社東芝

大学（50音順・敬称略）

飯田 元	奈良先端科学技術大学院大学
井上 克郎	大阪大学
川口 真司	奈良先端科学技術大学院大学
楠本 真二	大阪大学
久保 浩三	奈良先端科学技術大学院大学
黒崎 章	元 奈良先端科学技術大学院大学
小柴 昌也	奈良先端科学技術大学院大学
角田 雅照	奈良先端科学技術大学院大学
鳥居 宏次	奈良先端科学技術大学院大学
名倉 正剛	奈良先端科学技術大学院大学
松村 知子	奈良先端科学技術大学院大学
松本 健一	奈良先端科学技術大学院大学
森崎 修司	奈良先端科学技術大学院大学
門田 暁人	奈良先端科学技術大学院大学
リビエリ・シモネ	大阪大学

- ソフトウェア構築可視化法の問題検討委員会
第1回（2007年11月16日）～ 第5回（2008年10月8日）
構成員 7組織 11名（50音順・敬称略）

青江 秀史	大阪大学
飯田 元	奈良先端科学技術大学院大学
井上 克郎	大阪大学
江口 順一	帝塚山大学
大西 幸男	西銀座法律事務所
楠本 真二	大阪大学
久保 浩三	奈良先端科学技術大学院大学
小柴 昌也	奈良先端科学技術大学院大学
茶園 成樹	大阪大学
松田 貴典	大阪成蹊大学
松本 健一	奈良先端科学技術大学院大学

参考文献

- [1] ISO9001 : 2000 (JIS Q 9001/2000) 品質マネジメントシステム
- [2] 開発のための CMMI[®](CMMI-DEV) 1.2 版 公式日本語翻訳版, CMMI 成果物チーム, 技術報告書 CMU/SEI-2006-TR-008.
<http://www.sei.cmu.edu/cmmi/translations/japanese/models/dev-v12-abstract-j.html>
- [3] 保田勝通, ソフトウェア品質保証の考え方と実際, 日科技連, 1995年11月
- [4] IPA/SEC, ITプロジェクトの見える化 下流工程編, 日経BP, 2007年6月
- [5] IPA/SEC, ITプロジェクトの見える化 上流工程編, 日経BP, 2007年5月
- [6] 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
- [7] IPA/SEC, ソフトウェア開発データ白書 2007, 日経BP, 2007年7月
- [8] XBRL FACTBOOK :
http://www.xbrl-jp.org/download/pdf/factbook/2007/XBRLFACTBOOK_ver.9.pdf
- [9] 飯塚悦功編: “ソフトウェアの品質保証-ISO9000-3対訳と解説-”, 日本規格協会(1992).
- [10] “品質を考える技術”, 組込みプレス, 5, pp.13-34(2007).
- [11] 「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年4月13日公表, 経済産業省)
- [12] 「情報システムの信頼性向上に関するガイドライン」(2006年6月15日公表, 経済産業省)
- [13] 「情報システムの信頼性向上に関する評価指標(試行版)」(2007年4月13日公表, 経済産業省)
- [14] 「信頼性向上ベストプラクティスを実現する管理指標調査報告書」(2008年4月社団法人情報サービス協会)

ソフトウェアタグ規格 第 1.0 版
2008 年 10 月 14 日
StagE プロジェクト

奈良先端科学技術大学院大学 情報科学研究科
StagE プロジェクト研究代表者
松本 健一

大阪大学 大学院情報科学研究科
ソフトウェアタグ規格技術委員会委員長
井上 克郎

E-mail:stage-contact@is.naist.jp
URL:www.stage-project.jp

本規格は、文部科学省の科学技術試験研究委託事業による委託業務として、奈良先端科学技術大学院大学と大阪大学が共同で実施した、平成20年度の「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」の成果を取りまとめたものです。従って、本規格の著作権は、文部科学省に帰属しており、本規格の全部または一部の無断複製等の行為は、法律で認められたときを除き、著作権の侵害にあたるので、これらの利用行為を行うときは、文部科学省の承認手続きが必要です。

注) StagE プロジェクトは、文部科学省の委託業務「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開発普及」として、奈良先端科学技術大学院大学と大阪大学が共同で実施する研究開発プロジェクト「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」の略称です。