修士学位論文

題目

ソフトウェアタグデータ収集システムの設計とその試作

指導教員
楠本 真二 教授

報告者
西川 倫道

平成 21 年 2 月 9 日

大阪大学 大学院情報科学研究科
コンピュータサイエンス専攻

平成 20 年度 修士学位論文

ソフトウェアタグデータ収集システムの設計とその試作

西川 倫道

内容梗概

ソフトウェアの開発における定量的データの収集・分析は，ソフトウェア受注者側のプロセス改善やプロジェクトの進捗・品質管理を目的としている．しかし近年，ソフトウェア発注者側の視点に立った開発プロジェクトデータのより幅広い活用が求められ，ソフトウェアへのトレーサビリティの適用が提案されている．ソフトウェアに対するトレーサビリティとは製品品質や開発状況の把握が発注者にも可能なことを意味する．このソフトウェアトレーサビリティを実現する技術としてソフトウェアタグに関する研究・開発が行われている．ソフトウェアタグとはソフトウェア開発中に収集された種々のメトリクスをパッケージ化したものであり，実現するにはソフトウェアタグの規格化，ソフトウェアタグの収集・可視化・評価手法の開発，実証実験の実施，法的諸問題の検討といった様々な課題がある．本研究ではソフトウェアタグ実現の一環として，ソフトウェア開発時に作成される実証データからソフトウェアタグデータを収集するシステムを提案する．提案するシステムでは，どのような実証データからどのようなソフトウェアタグを収集したいかを指定することで収集を行う．また，ソフトウェアタグデータ収集システムの設計を基にプロトタイプを実装した．この際，ソフトウェア開発時に利用される開発支援ツールの機能や特徴を調査し，開発支援ツールから作成される実証データから汎用的にタグデータを収集する方法を検討した．そして，実際の開発プロジェクトに適用してソフトウェアタグを作成し，ソフトウェアタグを用いた開発プロジェクトの分析を行った．また，プロトタイプの実装から適用を通じて得られた知見から，ソフトウェアタグ収集システムの実用上の課題を考察した．

主な用語

ソフトウェアメトリクス
データ収集
ソフトウェアトレーサビリティ
ソフトウェアタグ

目 次

1 まえがき **1**

2 ソフトウェアトレーサビリティの実現 **2**

2.1 ソフトウェア受発注における問題 2

2.2 ソフトウェアトレーサビリティ 2

2.3 StagE プロジェクト 3

2.4 ソフトウェアトレーサビリティ実現上の課題 3

3 ソフトウェアタグ **4**

3.1 概要 4

3.2 プロジェクト情報 4

3.3 進捗情報 6

3.4 タグ利用のシナリオ 6

4 ソフトウェアタグデータ収集システム **8**

4.1 設計方針 8

4.2 システム構成 8

4.2.1 収集方法特定サブシステム 9

4.2.2 対象データ特定サブシステム 10

4.2.3 タグ項目算出サブシステム 10

4.2.4 タグ集計サブシステム 10

4.3 プロトタイプシステムの開発 10

4.3.1 対象とするタグ項目 10

4.3.2 タグデータ収集の実現方法 11

4.3.3 プロトタイプにおける収集方法特定 12

4.3.4 プロトタイプにおける対象データ特定 12

4.3.5 プロトタイプにおけるタグ項目算出 13

4.3.6 プロトタイプにおけるタグ集計 14

4.4 プロトタイプシステム利用例 14

5 適用事例 **16**

5.1 適用の目的 16

5.2 対象プロジェクト 16

5.3 実プロジェクトデータのタグ 17

5.4 タグデータによる開発状況の把握 22

6　考察 **23**

7　あとがき **24**

謝辞 **25**

参考文献 **26**

付録 **28**

1　まえがき

近年，ソフトウェアの大規模化と社会的影響の拡大によりソフトウェアの重要性が高まると共に，様々な問題が発生している [2]．例えば，ソフトウェアに障害が発生することで，金融システムや交通システムなどの重大な社会インフラが停止したり，航空管制システムや自動車安全システムなどが人命に関わる危険を引き起こす．また，それに伴いユーザ・ベンダに莫大な経済的損失を与える．他にも，開発期間の短縮のためにコストの低減や生産性の向上が要求されたり，情報システム開発に関わる訴訟が頻繁に発生している [22]．

これらの問題を解決するアプローチの一つとして，ソフトウェアトレーサビリティの概念が提案されている．トレーサビリティとは本来，野菜や食肉等の物品の生産・流通履歴を確認可能であることを意味するが，これをソフトウェアに対し適用することでソフトウェアユーザによるソフトウェアの製品品質・開発状況の把握や，ユーザ・ベンダ間の法的係争発生時の処理の短期化を可能とすることが期待されている．

ソフトウェアトレーサビリティを実現するためにソフトウェアタグの研究開発が行われている [20]．ソフトウェアタグとはソフトウェア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であり，ユーザはソフトウェアタグを見ることでソフトウェアがいつ，どこで，誰に，どのように開発されたか把握可能になる．

ソフトウェアタグによるソフトウェアトレーサビリティの実現には様々な課題があり，その一つにソフトウェアタグとなるデータを収集するツールの開発がある．そこで本研究ではソフトウェアトレーサビリティ実現の一環として，タグデータ収集システムの提案を行う．具体的には，システムへの要求に応える設計方針を検討し，システム構成を設計した．また，プロトタイプを開発し実際のソフトウェア開発プロジェクトに適用してソフトウェアタグを作成した．そして作成したソフトウェアタグからユーザがどのようにソフトウェアの品質や開発状況を把握するのか検討した．

以降，2 章ではソフトウェアトレーサビリティ実現にむけた概況を述べ，3 章ではソフトウェアタグの定義について説明する．4 章ではソフトウェアタグ収集システムの設計とプロトタイプの実装について述べ，5 章では，プロトタイプの適用事例について，適用の結果得られたソフトウェアタグデータの分析を含めて述べる．6 章では，プロトタイプの実装から適用事例を通じて得られた知見から，ソフトウェアタグ収集システムの課題を考察する．最後に 7 節でまとめと今後の課題を述べる．

2　ソフトウェアトレーサビリティの実現

2.1　ソフトウェア受発注における問題

ソフトウェア開発においてユーザは要件定義などの上流工程からベンダ任せにするケースが従来多かった [8]．しかし，そういった開発では，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，納期遅れなどをもたらす．また，障害発生時には，ユーザによる原因や責任所在の追究が困難で，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，解決まで発注者・開発者双方の被害が増大する．

以上から近年，ソフトウェア開発プロジェクトにおいてユーザの参画が重要視されてきている．しかし，これまでソフトウェア開発におけるデータ収集・分析はベンダ側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，ユーザの視点に立って行われてこなかった．そこで，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いる．

2.2　ソフトウェアトレーサビリティ

2.1 節の問題の解決策としてソフトウェアトレーサビリティの概念が提案されている．トレーサビリティとは，物品の生産・流通履歴を確認可能であることを意味し,「追跡可能性」とも言われている．近年，食の安全上の問題から食品に対するトレーサビリティへの取り組みが活発に行われており，生産・処理・加工・流通・販売等の各段階で，情報の追跡・遡及が可能になるよう推し進められている (図 1)．

トレーサビリティの概念をソフトウェア開発に適用したソフトウェアトレーサビリティでは，ソフトウェア開発過程で作成される実証データからソフトウェアタグを作成し，ソフトウェア製品に添付して流通させる (図 2)．製品に添付されたソフトウェアタグからユーザはソフトウェアの製品品質や開発状況を把握する．

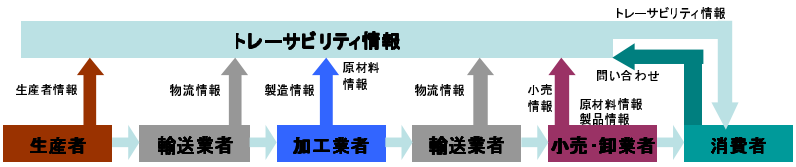

図 1: 食品のトレーサビリティ [20]

なお，過去にソフトウェア開発分野においてトレーサビリティという言葉は，能力成熟度モデル [10] で使用されてきた．能力成熟度モデル（CMMI）とは，ソフトウェア開発組織のソフトウェア開発能力向上のために利用されるプロセス改善モデルである．CMMI ではプロセス改善方法の一つとして要求仕様と計画・開発・成果物間のトレーサビリティの必要性を定義している．しかし，ここで

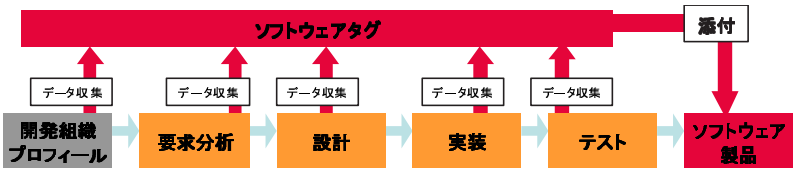

図 2: ソフトウェアトレーサビリティ [20]

述べられているトレーサビリティは開発組織のプロセス改善のためであり，ユーザ側にトレーサビリティを与えるものではない．

2.3 StagE プロジェクト

StagE(Software Traceability and Accountability of Global Software Engineering) プロジェクト [20] とは，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：ソフトウェア構築状況の可視化技術の開発と普及」の略称として，平成 19 年度から奈良先端科学技術大学院大学と大阪大学が産学官連携で進めているプロジェクトである．ソフトウェアトレーサビリティの実現を目的とし，ソフトウェアタグの研究開発を行っている．

2.4 ソフトウェアトレーサビリティ実現上の課題

ソフトウェアタグによるソフトウェアトレーサビリティの実現には以下の課題がある．

- ソフトウェアタグの規格化
- タグデータ収集ツールの開発
- タグデータの可視化・評価手法の開発
- 実証実験の実施
- 法的諸問題の検討

このうち，本研究ではタグデータ収集ツールの開発を目標としている．

3 ソフトウェアタグ

3.1 概要

現在，StagE プロジェクトではソフトウェアタグ規格 第 1.0 版が策定されている [19]．この規格においてソフトウェアタグは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．タグ項目はユーザ・ベンダ間で共有する候補として定義してあり，ユーザ・ベンダは用途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．

タグ項目は全 41 項目あり，プロジェクト情報と進捗情報に大別される．また，各タグ項目は分類，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，備考を要素としてもつ．各要素の説明を表 1 に示す．なお，予定・実績の要否とあるが，成果物やプロセス，目標についてベースラインを作成することはソフトウェア開発の改善において重要であるため [3]，いくつかのタグ項目に対し予定と実績による管理を要求している．

表 1: タグ項目構成要素

要素名	説明
項番	タグ項目に割り当てられる通し番号
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・タグデータのインデックス
説明	各タグ項目の意味や活用方法など
具体化例	タグデータの例
実証データ例	タグデータを抽出する元となるプロジェクト情報やドキュメント・生データ
予定・実績の要否	予定・実績値の差分による管理を行うデータ
工程別の要否	工程別に管理を行うデータ
備考	備考欄

3.2 プロジェクト情報

ソフトウェアタグの内，プロジェクト情報とは開発プロジェクトおよびシステムの基本的な情報を指す．プロジェクト情報はプロジェクトの途中で変更がなくプロジェクト全体を通して不変である．

プロジェクト情報内でタグ項目は 12 個あり，情報の種類ごとに以下の 5 つに分類される．

- 基本情報
- システム情報
- 開発情報
- プロジェクトの階層構造情報
- その他

表 2 においてプロジェクト情報の一覧を示す．

表 2: ソフトウェアタグ:プロジェクト情報

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
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		
	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	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		
				開発組織名，対象業種の経験 など			
	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	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		
				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			
	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		
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）	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		
				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			
	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	基本設計書 ソースコード 要件定義書 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		○	
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	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		○
				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			○
	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	勤務実績データ 工程管理表 など	○	○
				工期 など		○	○
プロジェクトの階層構造情報	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		
	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．				

3.3 進捗情報

ソフトウェアタグの内，進捗情報とはソフトウェア開発によって得られた作業の進捗状況や，成果物やプロセスの品質を表す情報を指す．進捗情報はプロジェクト情報とは異なりプロジェクト途中で変化する．

進捗情報内でタグ項目は 29 項目あり，そのタグ項目がどの開発プロセスに関するかでさらに以下の 8 つに分類される．

- 要件定義
- 設計
- プログラミング
- テスト
- 品質
- 工数
- 計画・管理
- その他成果物

表 3 において進捗情報の一覧を示す．なお，タグ項目名の後ろに [推移] という語句が付けられているものは，時系列で連続するデータが想定されるタグ項目である．

3.4 タグ利用のシナリオ

ソフトウェアタグ定義をユーザ・ベンダがどのように利用するのか一例を述べる．ソフトウェアタグの内，19 番プログラミングの規模としてソースコードからコード行数を，20 番プログラミングの変更としてソースコード行数の差分量を選択したとする．これにより，ユーザは実装時にコード行数がどのように推移して作成されたか，また，日々のコード行数の推移がどれくらいの変更・削除・追加によって行われたかを把握可能になる．他に，30 番の欠陥対応件数として不具合消化数を選択すれば，不具合を修正するためにどの程度ソースコードの修正を行ったかも分かる．

表 3: ソフトウェアタグ:進捗情報

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
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	○
	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、バッチプログラム数，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	
プログラミング	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	○
	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	
	21	複雑度	プログラムの品質（保守性）	Halstead，McCabe，CK，クローン含有率，コメント平均利用率など	ソースコード など	
テスト	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	○
	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）	テストケース テスト結果 ソースコード	○
				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	要件定義書 など	○
	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	○
品質	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，ユーザ参画の有無，レビュー時間など	レビュー議事録 など	○
	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	○
	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：不良摘出件数÷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	○

	32	**欠陥指摘率**	**テスト設計の品質**	不良摘出件数÷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	○
	33	**静的チェックの結果**	**プログラムの品質(保守性)**	チェッカによる総エラー数，チェッカ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) 不良摘出件数÷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	○
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・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	
	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、その他の項目でわける	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・改造・再利用(流用)毎に計測する**		ドキュメント 計画書 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	

4 ソフトウェアタグデータ収集システム

4.1 設計方針

ソフトウェアタグ収集システムに対する要求をどのように実現するか述べる．

まず，多くのソフトウェアタグデータに対応し収集可能であることが要求される．そこで，基本的なメトリクスの収集機能は収集システムが提供するとし，もし提供するメトリクスで対応出来ない場合は，システム利用者側でメトリクス収集機能を用意すれば，ツールに組み込み可能にする．

また，低コストでタグデータが収集可能であることも要求される．これには，極力タグデータはシステムが自動収集することで対応する．収集システムを用いるだけでソフトウェアタグが作成でき，余計な作業を必要とさせない．

タグデータ収集後に得られたソフトウェアタグが分析しやすいものであることも要求される．これは，タグデータ収集後に，ユーザが収集結果を理解しやすいよう視覚化等の分析が行われるためである．そこで，タグデータ分析ツールが利用しやすい形式でタグデータを出力することで対応する．

4.2 システム構成

4.1 節の方針に基づき設計したソフトウェアタグデータ収集システムの構成について述べる．

収集システムへの入力は，ソフトウェア開発時に発生する日々の開発履歴データと収集設定とする．日々の開発履歴データとは例えば，日々一定間隔でバックアップをとったプロジェクト開発フォルダや，構成管理ツールで管理されたデータなどである．また，入力として開発履歴データを得ただけでは，どのタグ項目を何から収集するのか判別できないため，収集対象と収集方法の特定を行う必要がある．そこで，これらの情報を特定するためにシステム利用者に収集設定の入力を求める．具体的には以下の項目を選択してもらう．

- 収集したいタグ項目は何か
- どの実証データから収集を行うか
- 収集対象はどのような開発支援ツールを介して得られたデータか
- 収集対象のデータ形式は何か

システムの出力はソフトウェアタグデータである．出力形式はタグデータ分析ツールの利用に適したものとする．

収集ツールではシステム利用者が使用したいメトリクスを提供していない場合が 2 通り考えられる．一つは，プログラムを作成すればメトリクス計測可能だが，収集システムでは提供していない場合である．もう一つは，メトリクスの計測が手作業でしか出来ない場合である．例えば，定型フォーマットの存在しない文章ファイルからメトリクスを計測する場合が挙げられる．前者に対してはメトリクス計測プログラムをシステム利用者側で作成してもらうことで対応する．プログラム作成には収集システムが API を提供し，収集方法の設定時に作成したメトリクス計測プログラムを指定することでプラグインとして導入可能にする．後者に対しては，利用者側でメトリクス計測後に特定のファイル形式で計測データを作成してもらう．そして，収集方法の設定時に入力データとして指定することで対応する．例えば，csv 形式や xml 形式で作成された計測データを指定し，収集したいタグデータとどのように対応をとるか設定する．

図 3 に設計したソフトウェアタグデータ収集システムの概要図を示す．開発データ履歴を入力とし，また，システム利用者から収集方法を設定してもらうことでソフトウェアタグデータを出力する．データ収集システム内部は収集方法特定，対象データ特定，タグ項目算出，タグ集計の 4 つのサブシステムで構成する．各サブシステムについて説明する．

4.2.1 収集方法特定サブシステム

収集設定をシステムユーザに入力してもらい収集方法を特定する．特定した情報のうち，タグ項目に関する情報はタグ項目算出サブシステムへ，収集対象となる実証データの情報は対象データ特定サブシステムへと送る．

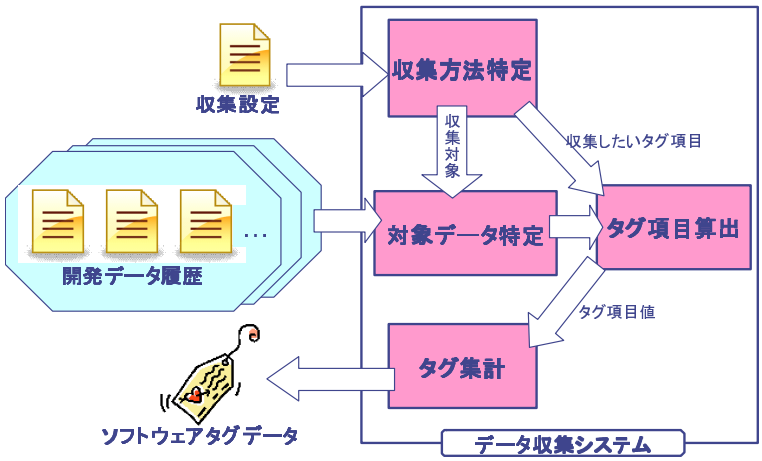

図 3: データ収集システム概要

4.2.2 対象データ特定サブシステム

収集方法特定サブシステムから送られた収集対象の情報を用いて，開発データ履歴から収集対象を特定する．例えば，収集対象となる実証データがどのような構成で履歴内に存在するのか，構成管理ツールで管理されているのかなどを判別し，各状況に即した特定を行う．特定した実証データはタグ項目算出サブシステムへと送る．

4.2.3 タグ項目算出サブシステム

タグ項目ごとに実証データからメトリクスを計測する．収集方法特定サブシステムから送られたタグ項目情報をもとに使用するメトリクスを決定し，対象データ特定サブシステムで得られた実証データから計測を行う．計測結果はタグ項目ごとにタグ集計サブシステムへと送る．

4.2.4 タグ集計サブシステム

タグ項目算出サブシステムから得られた各メトリクス値を集計し，ソフトウェアタグデータとして出力する．この際，タグデータ分析ツールが利用しやすい形式にデータ変換を行う．

4.3 プロトタイプシステムの開発

4.3.1 対象とするタグ項目

ソフトウェアタグ定義ではタグ項目の各メトリクス値として実際に何をとるかはユーザ・ベンダ間の協議で決めるものとし，具体化例を挙げるに留まっている．そこで，まずどのタグ項目に対し何をタグデータとして収集するのかを決定し，そのタグデータをどのような実証データから収集するか

表 4: プロトタイプが収集するタグデータ

分類	項番	タグ項目	タグデータ
要件定義	14	規模 [推移]	ユースケース・アクターの数
	15	変更 [推移]	追加，変更されたユースケース・アクターの数
設計	16	規模 [推移]	UML 図の数
	17	変更 [推移]	追加，変更された UML 図の数
プログラミング	19	規模 [推移]	コード行数
	20	変更 [推移]	追加，変更されたコード行数
	21	複雑度	CK メトリクス
品質	29	欠陥件数 [推移]	不具合数
	30	欠陥対応件数	不具合消化数
	31	欠陥対応密度	不具合数 ÷ コード行数

を決定する．今回開発したプロトタイプでは進捗情報の 10 項目を対象とし，各項目で何をタグデータとするかは表 4 のように選択した．

4.3.2 タグデータ収集の実現方法

特定のプロジェクトに特化せず，多くのプロジェクトに対応可能なデータ収集を行うことを目指した．そこで，まず開発現場でよく使用される開発支援ツールを調査し [7]，どうすればそれらツールから包括的にタグデータを収集できるか検討した．ソフトウェアタグの分類ごとにタグデータ収集方法を述べる．

要件定義ではユースケース図の構成要素であるユースケース・アクターの数を収集する．一般的にユースケース図の作成時には UML モデリングツールが使用される．実際の開発現場では UML モデリングツールとして，JUDE[14]，Rational Rose[12]，Microsoft office Visio[16] などが主に利用されている．これら UML モデリングツールは作成したモデルを XMI 形式 [23] で出力するという共通した機能をもつ．XMI(XML Metadata Interchange) は，抽象的なモデルをツール間で交換可能にするために OMG[17] が策定した標準規格である．そこで，この機能に着目し，XMI 出力されたファイルからユースケース図の情報を抽出する．

設計では，要件定義同様に UML 図作成のために UML モデリングツールが使用されるので，UML モデリングツールから XMI 出力されたファイルから UML 図の数を抽出する．

プログラミングでは，一般的に構成管理ツールが使用される．構成管理ツールとはソースコードやドキュメントの日々の進捗を管理するツールであり，Subversion[21] や CVS[6],[15] が知られている．これら構成管理ツールはプロジェクトデータを作業リポジトリに登録して用いられる．そこで，この作業リポジトリから日々登録されたソースコードを取得し，解析を行う．

品質では，バグ管理ツールが使用される．EPM[5] はソフトウェア開発で発生するデータからプロジェクトの情報可視化を行うツールであるが，データ収集対象となるバグ管理ツールとして bugzilla[11] や影舞 [9] を挙げている．これらバグ管理ツールは登録されたバグを様々な条件で検索し，結果を CSV 形式で出力する機能がある．そこで，この機能に着目し csv 出力されたバグ検索結果から不具合数，不具合消化数といった品質情報を求める．

4.3.3　プロトタイプにおける収集方法特定

収集方法の特定には収集したいタグ項目，収集対象の情報をシステムユーザが指定する必要がある．そこでユーザインターフェースを提供し，各情報を入力してもらう．例えばタグ項目としてプログラミングの規模を収集する場合に，開発支援ツールとして構成管理ツール subversion を，タグデータとしてコード行数をシステムユーザが指定する．

4.3.4　プロトタイプにおける対象データ特定

収集対象となる実証データについて使用した開発支援ツールやファイル形式が分かっていても，開発履歴フォルダの指定のみでは対象データの特定が正確に行えない．そこでソフトウェアタグ構造にならった図 4 のフォルダを用意し，各タグ項目に該当するフォルダに収集対象となる実証データを格納するという入力制限を加える．そしてこのフォルダ構造に基づいて実証データの特定を行う．

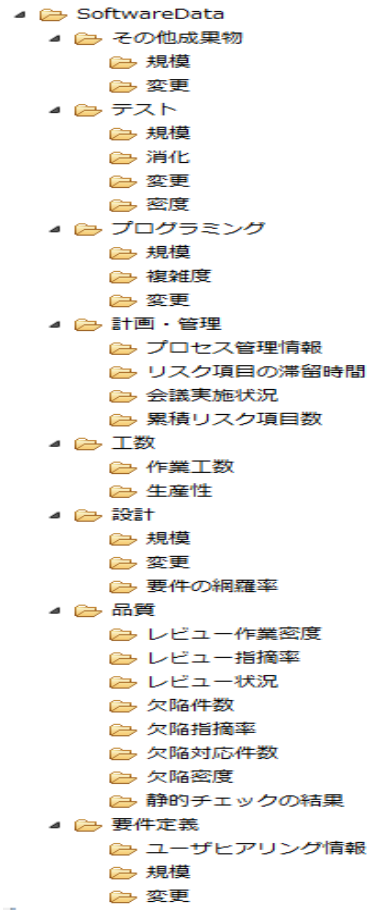

図 4: ソフトウェアタグ定義にならったフォルダ構造

4.3.5 プロトタイプにおけるタグ項目算出

各タグ項目でタグデータの導出が異なるため，分類ごとに分けて説明する

要件定義と設計ではユースケース図・UML 図に関する情報を UML モデリングツールの XMI 出力から収集する．そこで XMI ファイルの解析に SDMetrics[18] を利用した．SDMetrics は XMI ファイルから UML 図のデザイン計測を行うツールである．UML 図に含まれる種々の要素数を計測することが可能であり，また，2 つの XMI ファイル間で要素数に基づいた差分も計測できる．要素数の計測を用いてユースケースや UML 図の数を算出し、差分を用いて変更を算出する。

プログラミングではソースコードを構成管理ツールのリポジトリから得る．ソースコードからコード行数や CK メトリクスといったソフトウェアメトリクスを計測するために MASU[4] を使用する．MASU はソフトウェアメトリクスをプラグインとして自由に追加可能なメトリクス計測プラットフォームである．MASU の計測結果からソースコード行数と CK メトリクスを得る．なお変更については連続するリビジョン間の差分を求める．使用したソフトウェアメトリクスについて詳細をタグ項目ごとに示す.

- 項番 19，規模
  - LOC(Line of Code)
    ソースコードの行数．空行やコメント行は除く.
- 項番 20，変更
  - ADD
    追加されたソースコードの行数.
  - DELETE
    削除されたソースコードの行数.
- 項番 21，複雑度
  CK メトリクス [1] を使用.
  - WMC (Weighted Methods per Class)
    クラスのメソッドの重さ (複雑さの総和) を 計測する．WMC が高いほど複雑であり，メンテナンスのコストがかかることを意味する.
  - DIT (Depth of Inheritance Tree)
    クラスのスーパクラスの数を計測する．DIT が高いほど，継承されている変数やメソッドが多いことを表す.
  - NOC (Number Of Children)
    クラスのサブクラスの数を計測する．NOC が高いほどサブクラスへの影響力が強い.

– CBO (Coupling Between Objects)
クラスに関係しているクラスの数を計測する．CBO が高いほど他のクラスに依存していることを示す.

– RFC (Response For a Class)
クラスに関係しているメッセージの数を計測する．RFC が大きいほど受信したメッセージを遂行するために，発信しなければならないメッセージの数が多いことを示す.

– LCOM (Lack of Cohesion Of Methods)
クラスの凝集性の欠如を計測する．LCOM が小さいほど凝集性が高く，メソッドの強度が高いことを表す.

品質ではバグ管理ツールのバグ検索結果からバグに関する情報を収集する．バグ検索結果として表示される csv 出力の内容は自由に設定可能であり定型がない．そこで，タグデータとして何が欲しいのかによってシステムユーザが検索結果の出力項目を決める必要がある．プロトタイプではバグ検索結果として「バグが報告された日時」,「最終更新日時」,「バグが修正されたか」の情報をもつと仮定し，不具合数・不具合消化数を求めた.

4.3.6 プロトタイプにおけるタグ集計

対象データ特定において図 4 で示されるフォルダ構造を用いて開発履歴データの入力を行ったが，タグデータ収集結果も同様のフォルダ構造で出力する．例えばディレクトリ”C:\SoftWareTag”を出力先フォルダとしたとき，要件定義における規模の計測結果はディレクトリ”C:\SoftWareTag\要件定義\規模”下に格納する.

4.4 プロトタイプシステム利用例

開発したプロトタイプの利用例を示す．まず，起動すると図 5 の画面が表示される．＊1 で収集し

図 5: プロトタイプ:開始時画面

たいタグ項目の分類を選択し，＊2 でタグ項目を選択する．＊3 では収集対象となる実証データがど

のような開発支援ツールから得られたものかを選択する．そしてタグデータとして何をとるかを＊4で選択する．選択後に＊5の決定ボタンを押すと”！”マークが収集方法を選択した項目の前につく．要件定義の規模に当たるものとしてUMLモデリングツールJUDE[14]のXMI出力からアクターの数を計測するとしたときの収集設定画面を図6に示す．すべてのタグ分類，タグ項目に対して選択が

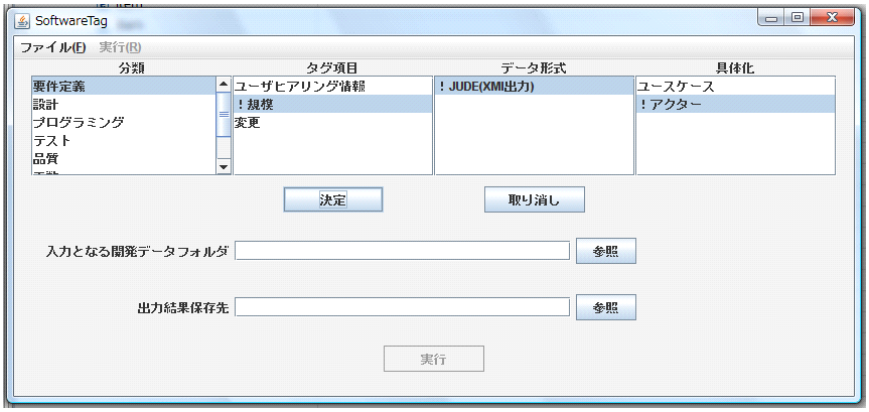

図 6: プロトタイプ:収集設定段階

終われば後は入力となるディレクトリ＊6と収集結果出力先＊7を指定し，実行ボタン＊8を押せばタグデータを収集する．

5 適用事例

5.1 適用の目的

ユーザがソフトウェアタグからソフトウェアの品質やプロジェクトの開発状況が把握可能であるかを調べることが適用の目的である．そこで，プロトタイプでは収集しない他のタグ項目についても手作業でメトリクスを収集し，ソフトウェアタグの効用をはかる．

5.2 対象プロジェクト

ITSpiral[13] で作成された実プロジェクト教材に対してプロトタイプを適用する．これは大学の履修登録プログラムを開発した際に得られた開発データであり，要件定義，設計，実装，テスト等の開発プロセスでさまざまな実証データが存在する．開発期間は5ヶ月で，総データ量は 450MB になる．実プロジェクト教材で収集されているデータを表 5 に示す．

表 5: 適用対象プロジェクトで収集されているデータ

プロセス名	収集データ
要件定義	要求仕様書，業務フロー図
要求	ユースケース図，アクター定義書，品質要求書
分析・設計	クラス図，コラボレーション図，シーケンス図
	アーキテクチャ仕様書，エンティティクラス設計書
	コントロールクラス設計書，バウンダリクラス設計書
実装	ソースコード，API ドキュメント，
	静的チェック記録（checkstyle,FindBugs,PMD,CPD,Jdepend）
テスト	テスト計画書・仕様書
プロジェクト管理	内部・外部レビュー記録，プロジェクト管理記録
	変更経緯説明書

なお，表 6 が示すように開発支援ツールを各開発プロセスで使用しており，収集データの一部としてツールの実行結果やプロジェクトデータが存在する．

表 6: 実プロジェクト教材で使用された開発支援ツール

ツール名	説明	使用開発プロセス
JUDE	UML モデリングツール	要求・分析・設計
Eclipse	統合開発環境	実装
影舞	バグ管理ツール	テスト
Subversion	構成管理ツール	実装
Jmeter	負荷テスト計測ツール	テスト

5.3 実プロジェクトデータのタグ

プロトタイプでは進捗情報 29 項目の内，10 項目に対応している．実プロジェクトデータから手作業で他のタグ項目についてメトリクスを収集した結果，進捗情報のうち表 7 の 12 項目のタグデータを新たに得た．

表 7: 手作業により得られたソフトウェアタグデータ

分類	項番	タグ項目	タグデータ
要件定義	13	ユーザヒアリング情報	ユーザヒアリング実施件数
テスト	22	規模 [推移]	テストケース数
	23	変更 [推移]	変更されたテストケース数
	24	密度	テストケース数/全体行数
	25	消化	消化テスト数
品質	26	レビュー状況	レビュー回数
	27	レビュー作業密度	レビュー時間/全体時間
	28	レビュー指摘率 [推移]	レビュー指摘数
	32	欠陥指摘率	欠陥件数/消化テスト数
	33	静的チェックの結果	FindBugs 指摘件数
工数	34	作業工数	作業時間
	35	生産性	行数/作業時間

計測したソフトウェアタグを示すがその際，どの程度細分化して情報を示すか決定する必要がある．タグ項目 19 番プログラミングの規模を例に挙げると，ソースコード行数の総和を示すのか，それともサブシステム毎に分けるのか，またはファイル毎やクラス毎に分けるのかを決定しないといけない．そこで，タグデータの粒度を表 8 のように設定する．レベル 1 は粒度をもっとも荒くした場合のタグデータであり，レベル 2 は一段階粒度を細かくしたタグデータである．

なお，タグ項番 13,29〜32 においてレベル 2 を設定していないが，これは今回の適用対象から得られた欠陥件数情報がサブシステムやクラス単位，またはバグの種類等の詳細データを持たず細かく分類できなかったためである．

表 8: タグデータの粒度

分類	項番	タグ項目	レベル 1	レベル 2
要件定義	13	ユーザヒアリング情報	ユーザヒアリング実施件数	
	14	規模 [推移]	ユースケース図数	ユースケース・アクター数
	15	変更 [推移]	変更されたユースケース図数	変更されたユースケース・アクター数
設計	16	規模 [推移]	UML 図数	クラス図数，シーケンス図数
				コラボレーション図数
				ステートチャート図数
				アクティビティ図数，配置図数
	17	変更 [推移]	変更された UML 図数	変更されたクラス図数
				シーケンス図数
				コラボレーション図数
				ステートチャート図数
				アクティビティ図数，配置図数
プログラミング	19	規模 [推移]	全体のソースコード行数	サブシステムのソースコード行数
	20	変更 [推移]	全体の変更されたソースコード行数	変更されたサブシステムのソースコード行数
	21	複雑度	全体の CK メトリクス	サブシステムの CK メトリクス
テスト	22	規模 [推移]	テストケース数	システムテスト・総合テスト数
	23	変更 [推移]	変更されたテストケース数	変更されたシステムテスト・総合テスト数
	24	密度	テストケース数/全体行数	システムテスト数/全体行数
				統合テスト数/全体行数
	25	消化	消化テスト数	システムテスト消化数
				総合テスト消化数
品質	26	レビュー状況	レビュー回数	内部レビュー回数
				外部レビュー回数
	27	レビュー作業密度	レビュー時間/全体時間	内部レビュー時間/全体時間
				外部レビュー時間/全体時間
	28	レビュー指摘率 [推移]	レビュー指摘数	内部レビュー指摘数
				外部レビュー指摘数
	29	欠陥件数 [推移]	全体欠陥件数	
	30	欠陥対応件数	全体欠陥対応件数	
品質	31	欠陥対応密度	全体欠陥件数/ソースコード行数	
	32	欠陥指摘率	全体欠陥件数/消化テスト数	
	33	静的チェックの結果	全体 FindBugs 指摘件数	サブシステムの FindBugs 指摘件数

プロジェクト情報の適用結果を表 9 に示す。次に進捗情報を粒度別に示す．表 10 と表 11 は，最終成果物についてそれぞれレベル 1 とレベル 2 で表したものである．表 7 はレベル 1 の日々の履歴を省略して表したものである。省略した部分は付録として添付する。

表 9: プロジェクト情報

分類	項番	タグ項目	計測値
基本情報	1	プロジェクト名	和歌山大学教務システム
	2	開発組織の情報	要求仕様作成チーム 要求仕様作成チーム:△△株式会社 責任者 ○○　主担当　××…
	3	開発プロジェクト情報	開発プロジェクト種別: 新規
	4	顧客情報	開発プロジェクト形態: 受託開発
システム情報	5	システム構成	顧客:和歌山大学 窓口:□□
			OS:Windows ブラウザ: Internet Explorer その他:　Adobe Reader
	6	システム規模	
開発情報	7	開発手法	
	8	開発体制	
	9	プロジェクト期間	実装期間2007年12月3日から2008年2月28日まで
プロジェクトの階層構造情報	10	親プロジェクト情報	
	11	サブ(子)プロジェクト情報	
その他	12	特記事項	

表 10: 進捗情報レベル 1

分類	項番	タグ項目	メトリクス	実測値
要件定義	13	ユーザヒアリング情報	ユーザヒアリング実施件数	3
	14	規模[推移]	ユースケース図数	12
	15	変更[推移]	ユースケース図数	48
設計	16	規模[推移]	UML図数	128
	17	変更[推移]	UML図数	434
	18	要件の網羅率		
プログラミング	19	規模[推移]	行数(全体)	26033
	20	変更[推移]	変更量(追加＋削除行数)	88841
	21	複雑度	WMC	6.28
			LCOM	10.96
			NOC	0.74
			DIT	2.81
			CBO	10.43
			RFC	13
テスト	22	規模[推移]	テストケース数	617
	23	変更[推移]	テストケース数	617
	24	密度	テストケース数/全体行数	0.02
	25	消化	消化テスト数	584
品質	26	レビュー状況	レビュー回数	21
	27	レビュー作業密度	レビュー時間/全体時間	1
	28	レビュー指摘率[推移]	レビュー指摘数	649
	29	欠陥件数[推移]	全体欠陥件数	19
	30	欠陥対応件数	全体欠陥対応件数	19
	31	欠陥密度	全体欠陥件数/行数	0.000730
	32	欠陥指摘率	全体欠陥件数/消化テスト数	0.0325
	33	静的チェックの結果	FIndBugs指摘件数	52
工数	34	作業工数	作業時間	86日
	35	生産性	行数/作業時間	302.7(行/日)
計画・管理	36	プロセス管理情報		
	37	会議実施状況		
	38	累積リスク項目数		
	39	リスク項目の滞留時間		
その他成果物	40	規模[推移]		
	41	変更[推移]		

表 11: 進捗情報レベル 2

分類	項番	タグ項目	メトリクスレベル1	メトリクスレベル2		実測値
要件定義	13	ユーザヒアリング情報				
	14	規模[推移]	ユースケース図数	ユースケース数		25
				アクタ数		7
	15	変更[推移]	ユースケース図数	ユースケース数		82
				アクタ数		22
設計	16	規模[推移]	UML図数	クラス図数		72
				シーケンス図数		20
				コラボレーション図数		21
				ステートチャート図数		0
				アクティビティ図数		2
				配置図数		1
	17	変更[推移]	UML図数	クラス図数		266
				シーケンス図数		69
				コラボレーション図数		97
				ステートチャート図数		0
				アクティビティ図数		2
				配置図数		2
	18	要件の網羅率				
プログラミング	19	規模[推移]	行数（全体）	行数（サブシステム）		
	20	変更[推移]	変更量（追加＋削除行数）	dbunit		253
				jp.ac.wakayama_u.kyoumu		672
				jp.ac.wakayama_u.kyoumu.boundary.act		24958
				jp.ac.wakayama_u.kyoumu.boundary.con		3564
				jp.ac.wakayama_u.kyoumu.boundary.forn		11606
				jp.ac.wakayama_u.kyoumu.boundary.tagl		46
				jp.ac.wakayama_u.kyoumu.db		712
				jp.ac.wakayama_u.kyoumu.db.dao		11766
				jp.ac.wakayama_u.kyoumu.db.hibernate		4260
				jp.ac.wakayama_u.kyoumu.dto		3267
				jp.ac.wakayama_u.kyoumu.entity		13750
				jp.ac.wakayama_u.kyoumu.filter		479
				jp.ac.wakayama_u.kyoumu.service		11731
				jp.ac.wakayama_u.kyoumu.util		980
				jp.ac.wakayama_u.kyoumu.util.converte		664
				jp.ac.wakayama_u.kyoumu.util.csv		73
				strutstest		60
	21	複雑度	WMC	WMC（サブシステム）		
			LCOM	LCOM(サブシステム)		
			NOC	NOC(サブシステム)		
			DIT	DIT(サブシステム)		
			CBO	CBO(サブシステム)		
			RFC	RFC(サブシステム)		
テスト	22	規模[推移]	テストケース数	システムテスト数	予定	76
				統合テスト数	予定	541
	23	変更[推移]	テストケース数	システムテスト数	予定	76
					実績	76
				統合テスト数	予定	541
					実績	508
	24	密度	テストケース数/全体行数	システムテストケース数/全体行数		0.00292
				統合テスト数/全体行数		0.0208
	25	消化	消化テスト数	システムテスト消化数	実績	76
				統合テスト消化数	実績	508
品質	26	レビュー状況	レビュー回数	内部レビュー		14
				外部レビュー		7
	27	レビュー作業密度	レビュー時間/全体時間	内部レビュー時間/全体時間		1
				外部レビュー時間/全体時間		1
	28	レビュー指摘率[推移]	レビュー指摘数	内部レビュー指摘数		369
				外部レビュー指摘数		280
	29	欠陥件数[推移]	全体欠陥件数			19
	30	欠陥対応件数	全体欠陥対応件数			19
	31	欠陥密度	全体欠陥件数/行数			0.000730
	32	欠陥指摘率	全体欠陥件数/消化テスト数			0.0325
	33	静的チェックの結果	FIndBugs指摘件数	jp/ac/wakayama_u/kyoumu/db/hibernate		6
				jp/ac/wakayama_u/kyoumu/entity/		6
				jp/ac/wakayama_u/kyoumu/service/		1
				jp/ac/wakayama_u/kyoumu/		14
				jp/ac/wakayama_u/kyoumu/boundary/		24
				jp/ac/wakayama_u/kyoumu/boundary/		1
工数	34	作業工数	作業時間			86日
	35	生産性	行数/作業時間			302.7(行/日
計画・管理	36	プロセス管理情報				
	37	会議実施状況		cBO(サブシステム1)		
	38	累積リスク項目数				
	39	リスク項目の滞留時間				
その他成果物	40	規模[推移]				
	41	変更[推移]				

表 12: 進捗情報レベル 1 履歴 (一部省略)

分類	項番	タグ/項目	メトリクス		開始日 3/27	12/9	12/10	12/11	12/12	12/13	12/14	2/25	2/26	終了日 2/27
要件定義	13	ユーザヒアリング情報	ユーザヒアリング回数											3
	14	規模[推移]	ユースケース図数		11	12	12	12	12	12	12	12	12	12
	15	変更[推移]	ユースケース図数	追加	11	0	0	0	0	0	0	0	0	0
				変更	0	0	0	0	0	0	0	0	0	0
				削除	0	0	0	0	0	0	0	0	0	0
設計	16	規模[推移]	UML図数		2	114	114	114	114	114	114	116	116	116
	17	変更[推移]	UML図数	追加	2	0	0	0	0	0	0	0	0	0
				変更	0	0	0	0	0	0	0	0	0	0
				削除	0	0	0	0	0	0	0	0	0	0
	18	要件の網羅率												
プログラミング	19	規模[推移]	行数(全体)		0	0	354	994	1207	2093	2996	26068	25988	26033
	20	変更[推移]	変更量(追加+削除行数)		0	0	866	946	664	1471	1400	21	974	58
	21	複雑度	WMC		0	0	2.14	2.4	2.86	4.35	4.45	6.27	6.27	6.28
			LCOM		0	0	2.5	5.12	4.86	6.74	5.1	10.96	10.96	10.96
			NOC		0	0	0.29	0.56	0.61	0.5	0.57	0.74	0.74	0.74
			DIT		0	0	2.21	2.8	2.79	2.65	2.67	2.81	2.81	2.81
			CBO		0	0	2.5	3.16	3.57	6.68	8.05	10.41	10.41	10.43
			RFC		0	0	2.93	2.84	4.14	7.56	8.17	13.01	13.02	13
テスト	22	規模[推移]	テストケース数	予定	0	76	76	76	76	187	187	587	613	597
	23	変更[推移]	テストケース数	予定	0	0	0	0	0	111	0	33	26	−16
				実績	0	0	0	0	0	0	0	174	140	68
	24	密度	テストケース数/全体行数		0.0000	0.0000	0.2147	0.0765	0.0630	0.0893	0.0624	0.0225	0.0236	0.0229
	25	消化	消化テスト数	実績	0	0	0	0	0	0	0	377	517	585
品質	26	レビュー状況	レビュー回数		0	17	17	17	17	17	17	21	21	21
	27	レビュー作業密度	レビュー時間/全体時間		0	0.92	0.92	0.92	0.92	0.92	0.92	1	1	1
	28	レビュー指摘率[推移]	レビュー指摘数		0	599	599	599	599	599	599	649	649	649
	29	欠陥件数[推移]	全体欠陥件数		0	0	0	0	0	0	0	19	19	19
	30	欠陥対応件数	全体欠陥対応件数		0	0	0	0	0	0	0	19	19	19
	31	欠陥密度	全体欠陥件数/行数		0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000729	0.000731	0.000730
	32	欠陥指摘率	全体欠陥件数/消化テスト数		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0504	0.0368	0.0325
	33	静的チェックの結果	FindBugs指摘件数		0	0	0	0	0	0	6	52	52	52
工数	34	作業工数	作業時間(日)											152
	35	生産性	行数/作業時間											171.3
計画・管理	36	プロセス管理情報												
	37	会議実施状況												
	38	累積リスク項目数												
	39	リスク項目の滞留時間												
その他成果物	40	規模[推移]												
	41	変更[推移]												

5.4 タグデータによる開発状況の把握

計測結果から，いくつかのタグデータをグラフ化したものを示す．図 7 はソースコード全体のコード行数と変更量の推移を表し，図 8 は CK メトリクス [1] の推移を表したものである．

*A の期間において，コード行数はほとんど変化していないが，変更量が大きい．そこで，同期間において CK メトリクスの推移を見ると，減少していることが分かる．CK メトリクスはプログラムの複雑度を表すメトリクスである．そこで，この期間では新たに追加や削除が行われたのではなく，リファクタリングが行われたのではないかと推測できる．実際に構成管理ツールからソースコードの変更内容を調べたところ，リファクタリングの実施を確認した．このように，ソフトウェアタグを用いることでユーザは適用対象プロジェクトの特徴を把握，分析可能である．

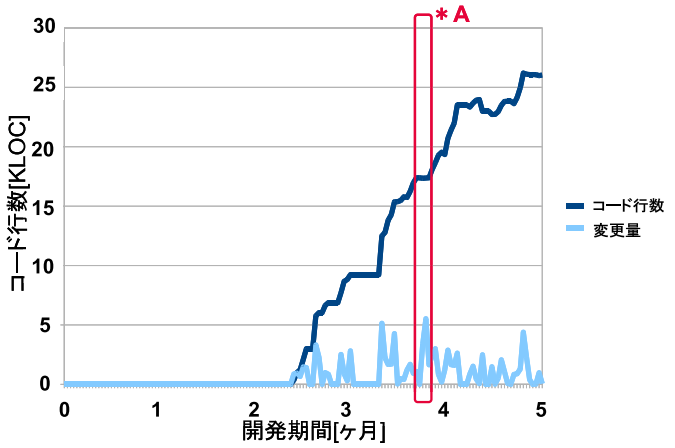

図 7: ソースコード行数と変更量の推移

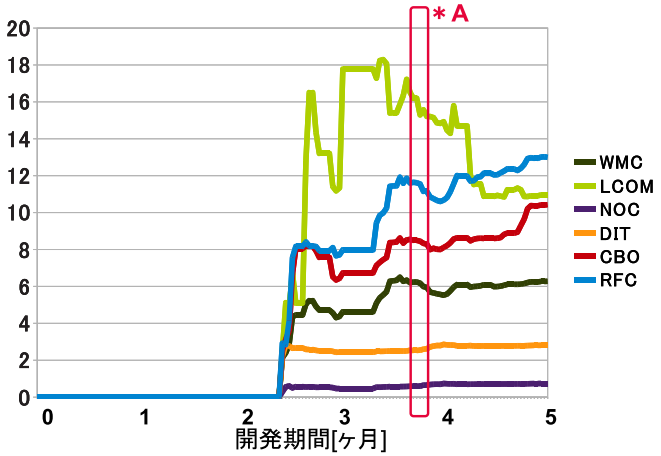

図 8: CK メトリクスの推移

6　考察

プロトタイプの適用結果を示す際に各タグデータを粒度でレベル分けして表示した．これは適用結果を分かりやすく示すために行ったが，本来はタグデータの粒度はユーザ・ベンダ間の協議で設定するものとソフトウェアタグ規格で設定している．しかし，ユーザ・ベンダがタグデータの粒度を自由に決定するとなると，ソフトウェアタグデータの収集やソフトウェアタグの視覚化といったツール開発の際に一概に対象データを決定することが出来ない．そこで，タグデータの粒度定義をソフトウェアタグ定義において行う必要があるのではないかと考える．ソフトウェアタグ定義において各具体化項目の粒度としてある程度の定義をおけば，その定義に基づいて各ツールの開発が容易になる．

以上から，粒度についてソフトウェアタグ定義において記述し，その定義をもとにユーザ・ベンダ間でタグデータと粒度の選択を行ったほうがよいと考える．

7　あとがき

本論文では，ソフトウェアトレーサビリティの実現を目的としてソフトウェアタグ収集システムを設計した．ソフトウェアタグ収集システムへの要求について実現方針を検討し，ソフトウェア開発履歴データとタグデータ収集設定情報からソフトウェアタグデータを得るシステムを提案した．さらに，設計したシステムに基づいて開発したプロトタイプを実際の開発プロジェクトデータへ適用し，ソフトウェアタグの有用性を確認した．

今後の課題としては，以下があげられる．

- 対応外メトリクスへの対応
  ソフトウェアタグ収集システムでは対応しないメトリクスを，ユーザが組み込める機能を実装する．
- 他開発プロジェクトへの適用
  他のプロジェクトデータに適用実験を行い，新たな知見や問題点を得ることで，ソフトウェアタグ収集システムを改良する．

謝辞

本研究の全過程を通して，常に適切な御指導および丁寧な御助言を賜りました大阪大学大学院情報科学研究科コンピュータサイエンス専攻 楠本真二教授に心から深く感謝申し上げます.

本研究に適切な御指導および御助言を頂きました大阪大学大学院情報科学研究科コンピュータサイエンス専攻 岡野浩三准教授に心から感謝いたします.

本研究に対して，適切な御指導および御助言を頂きました大阪大学大学院情報科学研究科コンピュータサイエンス専攻 肥後芳樹助教に深く感謝いたします.

本研究に対して，適切な御指導および御助言を頂きました大阪大学大学院情報科学研究科コンピュータサイエンス専攻 柿元健特任助教に感謝いたします.

プロトタイプの実装を御手伝い頂いた大阪大学大学院情報科学研究科博士前期課程 1 年 鵜久森将隆氏に感謝いたします.

ソフトウェアタグデータの分析を御手伝い頂いた大阪大学基礎工学部情報科学科 4 年 山田慎也氏に感謝いたします.

その他の大阪大学 大学院情報科学研究科コンピュータサイエンス専攻 楠本研究室の皆様のご協力に深く感謝致します.

参考文献

[1] S. R. Chidamber, C. F. Kemerer, “A Metrics Suite for Object-Oriented Design”, IEEE Trans. Software Eng., vol.20, pp.476-493, 1994.

[2] 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研究への取り組み”, 情報処理, Vol.45, No.7, pp.722-728, 2004

[3] 松本 健一, “エンピリカルソフトウェア工学の現状と展望：SEL が残した 13 の教訓 ”, SEC journal, No2, pp.6-13, 2005

[4] 三宅 達也, 肥後 芳樹, 井上 克郎: “メトリクス計測プラグインプラットフォーム MASU の開発 ”, ソフトウェアエンジニアリング最前線 2008, pp63-70, Sep 2008

[5] 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, Vol.J88-D-I, No.2, pp.228-239, 2005

[6] Thomas Zimmermann, Peter Weisgerber, “Preprocessiong CVS Data for Fine-grained analysis ”, 1st International Workshop on Mining Software Repositories, pp.2-6, 2004

[7] “特集 1：開発支援ツールの利用実態”, 日経 SYSTEMS, 2008 年 6 月号, pp.11-39, 2008

[8] “特集 1：ユーザーの言い分ベンダーの言い分”, 日経コンピュータ, 2008 年 10 月 1 日号, pp.40-55, 2008

[9] バグトラッキングシステム影舞

http://www.daifukuya.com/kagemai/

[10] CMMI V1.2 モデル-公式日本語翻訳版

http://www.sei.cmu.edu/cmmi/translations/japanese/models/index.html

[11] Home::Bugzilla::bugzilla.org ホームページ

http://www.bugzilla.org/

[12] IBM Rational ソフトウェア

http://www-06.ibm.com/jp/software/rational/

[13] IT スパイラル

http://it-spiral.ist.osaka-u.ac.jp/index.html

[14] JUDE:UML, ED, CRUC, DFD, Flowchart and Mind Map:Design & Modeling Tool
http://jude.change-vision.com/jude-web/index
.html

[15] Main Page - Ximbiot - CVS Wiki
http://ximbiot.com/cvs/wiki/

[16] Microsoft Office Visio ホームページ
http://office.microsoft.com/ja-jp/visio/default.aspx

[17] Object Managemant Group
http://www.omg.org/

[18] SDMetrics - the design quality metrics tool for UML models
http://www.sdmetrics.com/

[19] ソフトウェアタグ規格 第 1.0 版
http://www.stage-project.jp/kanri/data/dl/20081029151602.pdf

[20] StagE プロジェクト
http://www.stage-project.jp/

[21] subversion.tigris.org
http://subversion.tigris.org/

[22] 「動かないコンピュータ」の解決に裁判は役立つか
http://itpro.nikkeibp.co.jp/free/ITPro/OPINION/20040609/145573/

[23] XML Metadeta Interchange
http://www.omg.org/technology/documents/formal/xmi.htm

付録

本文 5.3 節において述べた実プロジェクトデータのソフトウェアタグ収集結果のうち、レベル１の日々の履歴について示す。

	15			変更	0	2	5	0	0	0	4	5	1	1	0	0	0	0	11	0	0	0	0	0	0	1	0		
				削除	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0		
設計	16	規模[推移]	UML図数		2	2	2	2	2	2	2	2	2	2	6	6	6	7	7	7	8	10	10	10	23	40	39	4	
	17	変更[推移]	UML図数	追加	2	0	0	0	0	0	0	0	0	0	4	0	0	1	0	0	2	2	0	0	13	17	1	1	
				変更	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	2	11	14	2	
				削除	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2		
	18	要件の網羅率																											
プログラミング	19	規模[推移]	行数(全体)		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	20	変更[推移]	変更量(追加+削除行数)		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	21	複雑度	WMC		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
			LCOM		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
			NOC		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
			DIT		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
			CBO		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
			RFC		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
テスト	22	規模[推移]	テストケース数	予定	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	23	変更[推移]	テストケース数	予定	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
				実績	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	24	密度	テストケース数/全体行数		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.000	
	25	消化	消化テスト数	実績	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
品質	26	レビュー状況	レビュー回数		0	0	0	0	0	0	1	2	3	3	3	3	3	4	4	4	4	5	5	5	5	5	6		
	27	レビュー作業密度	レビュー時間/全体時間		0	0	0	0	0	0	0.04	0.15	0.18	0.18	0.18	0.18	0.18	0.25	0.25	0.25	0.25	0.27	0.27	0.27	0.27	0.27	0.39	0.3	
	28	レビュー指摘率[推移]	レビュー指摘数		0	0	0	0	0	0	26	83	99	99	99	99	99	157	157	157	157	161	161	161	161	161	219	22	
	29	欠陥件数[推移]	全体欠陥件数		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	30	欠陥対応件数	全体欠陥対応件数		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	31	欠陥密度	全体欠陥件数/行数		0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	
	32	欠陥指摘率	全体欠陥件数/消化テスト数		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.000	
	33	静的チェックの結果	FIndBugs指摘件数		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
工数	34	作業工数	作業時間																										
	35	生産性	行数/作業時間																										
計画・管理	36	プロセス管理情報																											
	37	会議実施状況																											
	38	累積リスク項目数																											
	39	リスク項目の滞留時間																											
その他成果物	40	規模[推移]																											
	41	変更[推移]																											

	15			変更	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
				削除	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
設計	16	規模[推移]	UML図数		46	59	59	59	59	59	67	67	67	67	67	75	75	75	77	77	77	77	77	82	92	92	
	17	変更[推移]	UML図数	追加	4	18	0	0	0	0	10	0	0	0	0	9	0	4	2	4	0	0	2	3	7	12	0
				変更	0	1	0	0	0	0	0	0	0	0	0	5	1	2	4	9	0	0	1	2	1	1	0
				削除	0	5	0	0	0	0	2	0	0	0	0	1	0	4	0	4	0	0	2	3	2	2	0
	18	要件の網羅率																									
プログラミング	19	規模[推移]	行数(全体)		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	20	変更[推移]	変更量(追加＋削除行数)		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	21	複雑度	WMC		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
			LCOM		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
			NOC		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
			DIT		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
			CBO		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
			RFC		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
テスト	22	規模[推移]	テストケース数	予定	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	23	変更[推移]	テストケース数	予定	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
				実績	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	24	密度	テストケース数/全体行数		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
	25	消化	消化テスト数	実績	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
品質	26	レビュー状況	レビュー回数		9	9	9	9	9	9	9	9	9	9	9	9	9	9	10	11	11	11	12	30 12	12	12	13
	27	レビュー作業密度	レビュー時間/全体時間		0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.6	0.62	0.62	0.62	0.68	0.68	0.68	0.68	0.72
	28	レビュー指摘率[推移]	レビュー指摘数		309	309	309	309	309	309	309	309	309	309	309	309	309	309	375	389	389	389	423	423	423	423	514
	29	欠陥件数[推移]	全体欠陥件数		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	30	欠陥対応件数	全体欠陥対応件数		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	31	欠陥密度	全体欠陥件数/行数		0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
	32	欠陥指摘率	全体欠陥件数/消化テスト数		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
	33	静的チェックの結果	FIndBugs指摘件数		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
工数	34	作業工数	作業時間																								
	35	生産性	行数/作業時間																								
計画・管理	36	プロセス管理情報																									
	37	会議実施状況																									
	38	累積リスク項目数																									
	39	リスク項目の滞留時間																									
その他成果物	40	規模[推移]																									
	41	変更[推移]																									

	15			変更	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				削除	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
設計	16	規模[推移]	UML図数		117	117	117	112	112	114	114	114	114	114	114	114	114	114	114	114	114	114	114	114	114	114	114
	17	変更[推移]	UML図数	追加	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				変更	0	0	0	16	1	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				削除	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	18	要件の網羅率																									
プログラミング	19	規模[推移]	行数(全体)		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	354	994	1207	2093	2996	2996	2996	5764	
	20	変更[推移]	変更量(追加＋削除行数)		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	866	946	664	1471	1400	0	0	3317	
	21	複雑度	WMC		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.14	2.4	2.86	4.35	4.45	4.45	4.45	5.04	
			LCOM		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.5	5.12	4.86	6.74	5.1	5.1	5.1	12.8	
			NOC		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.29	0.56	0.61	0.5	0.57	0.57	0.57	0.54	
			DIT		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.21	2.8	2.79	2.65	2.67	2.67	2.67	2.59	
			CBO		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.5	3.16	3.57	6.68	8.05	8.05	8.05	8.19	
			RFC		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.93	2.84	4.14	7.56	8.17	8.17	8.17	8.41	
テスト	22	規模[推移]	テストケース数	予定	0	0	0	0	0	0	0	0	0	0	75	75	76	76	76	76	76	76	187	187	187	187	187	
	23	変更[推移]	テストケース数	予定	0	0	0	0	0	0	0	0	0	0	75	0	1	0	0	0	0	0	111	0	0	0	0	
				実績	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	24	密度	テストケース数/全体行数		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2147	0.0765	0.0630	0.0893	0.0624	0.0624	0.0624	0.0324	
	25	消化	消化テスト数	実績	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
品質	26	レビュー状況	レビュー回数		14	14	16	16	16	16	16	16	16	16	16	17	17	17	17	17	17	17	17	17	17	17	17	
	27	レビュー作業密度	レビュー時間/全体時間		0.83	0.83	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	
	28	レビュー指摘率[推移]	レビュー指摘数		559	559	584	584	584	584	584	584	584	584	584	599	599	599	599	599	599	599	599	599	599	599	599	
	29	欠陥件数[推移]	全体欠陥件数		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	30	欠陥対応件数	全体欠陥対応件数		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	31	欠陥密度	全体欠陥件数/行数		0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
	32	欠陥指摘率	全体欠陥件数/消化テスト数		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
	33	静的チェックの結果	FIndBugs指摘件数		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	6	6	12	
工数	34	作業工数	作業時間																									
	35	生産性	行数/作業時間																									
計画・管理	36	プロセス管理情報																										
	37	会議実施状況																										
	38	累積リスク項目数																										
	39	リスク項目の滞留時間																										
その他成果物	40	規模[推移]																										
	41	変更[推移]																										

	15			変更	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				削除	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
設計	16	規模[推移]	UML図数		114	114	114	114	114	114	114	114	114	114	114	114	114	114	114	114	114	114	114	114	114	114	114	114
	17	変更[推移]	UML図数	追加	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				変更	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				削除	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	18	要件の網羅率																										
プログラミング	19	規模[推移]	行数(全体)		7716	8667	8822	9228	9228	9228	9228	9228	9228	9228	9228	9228	9228	12492	12784	13813	14279	15357	15375	15475	15783	15731	16237	
	20	変更[推移]	変更量(追加+削除行数)		2513	901	300	2812	0	0	0	0	0	0	0	0	0	5136	2242	1673	1700	4266	33	475	410	1203	1679	
	21	複雑度	WMC		4.56	4.31	4.37	4.63	4.63	4.63	4.63	4.63	4.63	4.63	4.63	4.63	4.63	5.1	5.38	5.54	5.71	6.28	6.31	6.33	6.5	6.28	6.34	
			LCOM		11.42	11.18	11.34	17.77	17.77	17.77	17.77	17.77	17.77	17.77	17.77	17.77	17.77	17.29	18.24	18.29	18.07	15.39	15.41	15.41	15.86	16.35	17.22	
			NOC		0.52	0.46	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.54	0.54	0.55	0.56	0.57	0.57	0.57	0.57	0.57	0.58	
			DIT		2.48	2.43	2.44	2.44	2.44	2.44	2.44	2.44	2.44	2.44	2.44	2.44	2.44	2.48	2.48	2.48	2.49	2.5	2.5	2.5	2.5	2.5	2.5	
			CBO		6.52	6.35	6.44	6.73	6.73	6.73	6.73	6.73	6.73	6.73	6.73	6.73	6.73	7.13	7.3	7.45	7.56	8.38	8.38	8.41	8.62	8.35	8.52	
			RFC		8.1	7.66	7.71	7.98	7.98	7.98	7.98	7.98	7.98	7.98	7.98	7.98	7.98	9.44	9.86	9.98	10.24	11.42	11.45	11.45	11.92	11.59	11.86	
テスト	22	規模[推移]	テストケース数	予定	314	314	314	314	314	314	314	314	314	314	314	314	314	314	314	314	314	314	314	314	314	314	314	
	23	変更[推移]	テストケース数	予定	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
				実績	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	24	密度	テストケース数/全体行数		0.0407	0.0362	0.0356	0.0340	0.0340	0.0340	0.0340	0.0340	0.0340	0.0340	0.0340	0.0340	0.0340	0.0251	0.0246	0.0227	0.0220	0.0204	0.0204	0.0203	0.0199	0.0200	0.0193	
	25	消化	消化テスト数	実績	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
品質	26	レビュー状況	レビュー回数		18	18	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	
	27	レビュー作業密度	レビュー時間/全体時間		0.94	0.94	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	
	28	レビュー指摘率[推移]	レビュー指摘数		615	615	625	625	625	625	625	625	625	625	625	625	625	625	625	625	625	625	625	625	625	625	625	
	29	欠陥件数[推移]	全体欠陥件数		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	30	欠陥対応件数	全体欠陥対応件数		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	31	欠陥密度	全体欠陥件数/行数		0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
	32	欠陥指摘率	全体欠陥件数/消化テスト数		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
	33	静的チェックの結果	FIndBugs指摘件数		12	19	19	25	25	25	25	25	25	25	25	25	25	38	40	40	42	42	42	42	42	42	42	
工数	34	作業工数	作業時間																									
	35	生産性	行数/作業時間																									
計画・管理	36	プロセス管理情報																										
	37	会議実施状況																										
	38	累積リスク項目数																										
	39	リスク項目の滞留時間																										
その他成果物	40	規模[推移]																										
	41	変更[推移]																										

	15			変更	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				削除	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
設計	16	規模[推移]	UML図数		116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116
	17	変更[推移]	UML図数	追加	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				変更	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				削除	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	18	要件の網羅率																									
プログラミング	19	規模[推移]	行数(全体)		18657	19292	19536	19346	20712	21399	21978	23514	23514	23514	23514	23347	23673	23917	23974	22989	22989	22989	22715	22723	22955	23489	23810
	20	変更[推移]	変更量(追加+削除行数)		2994	829	193	1214	2875	1659	1594	2623	1	0	0	878	1521	456	57	2481	0	0	1475	17	525	2058	1070
	21	複雑度	WMC		5.65	5.6	5.57	5.52	5.58	5.76	5.96	6.1	6.1	6.1	6.1	5.96	6	6	6.03	6.09	6.09	6.09	5.99	5.99	6.01	6.06	6.11
			LCOM		15.15	14.86	14.83	14.93	14.47	14.3	15.79	14.7	14.7	14.7	14.7	12	11.52	11.56	11.56	10.88	10.88	10.88	10.88	10.94	10.89	10.85	11.24
			NOC		0.7	0.7	0.69	0.74	0.73	0.72	0.73	0.69	0.69	0.69	0.69	0.7	0.7	0.7	0.7	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
			DIT		2.77	2.8	2.79	2.86	2.83	2.82	2.82	2.78	2.78	2.78	2.78	2.8	2.77	2.77	2.77	2.78	2.78	2.78	2.78	2.78	2.78	2.78	2.78
			CBO		8.07	8.01	8	8.11	8.21	8.36	8.57	8.63	8.63	8.63	8.63	8.47	8.56	8.6	8.61	8.6	8.6	8.6	8.6	8.61	8.65	8.82	8.89
			RFC		10.76	10.66	10.6	10.67	10.81	11.13	11.56	12.01	11.98	11.98	11.98	11.71	11.72	11.92	11.95	12.13	12.13	12.13	12.04	12.05	12.15	12.25	12.37
テスト	22	規模[推移]	テストケース数	予定	366	366	366	366	471	471	471	529	529	529	529	529	532	542	542	558	558	558	558	558	558	558	554
	23	変更[推移]	テストケース数	予定	3	0	0	0	105	0	0	58	0	0	0	6	-3	10	0	16	0	0	0	0	0	0	-4
				実績	3	0	0	0	32	0	0	153	0	0	0	0	-1	1	0	16	0	0	0	0	0	0	-1
	24	密度	テストケース数/全体行数		0.0196	0.0190	0.0187	0.0189	0.0227	0.0220	0.0214	0.0225	0.0225	0.0225	0.0225	0.0227	0.0225	0.0227	0.0226	0.0243	0.0243	0.0243	0.0246	0.0246	0.0243	0.0238	0.0233
	25	消化	消化テスト数	実績	3	3	3	3	35	35	35	188	188	188	188	188	187	187	187	204	204	204	204	204	204	204	203
品質	26	レビュー状況	レビュー回数		20	20	20	20	20	20	20	20	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21
	27	レビュー作業密度	レビュー時間/全体時間		0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	28	レビュー指摘率[推移]	レビュー指摘数		636	636	636	636	636	636	636	636	649	649	649	649	649	649	649	649	649	649	649	649	649	649	649
	29	欠陥件数[推移]	全体欠陥件数		0	0	0	0	0	4	4	5	5	5	5	5	5	7	9	10	10	10	10	16	16	16	16
	30	欠陥対応件数	全体欠陥対応件数		0	0	0	0	0	0	3	3	3	3	3	3	3	3	4	10	10	10	10	10	15	15	15
	31	欠陥密度	全体欠陥件数/行数		0.000000	0.000000	0.000000	0.000000	0.000000	0.000187	0.000182	0.000213	0.000213	0.000213	0.000213	0.000214	0.000211	0.000293	0.000375	0.000435	0.000435	0.000435	0.000440	0.000704	0.000697	0.000681	0.000672	0.0
	32	欠陥指摘率	全体欠陥件数/消化テスト数		0.0000	0.0000	0.0000	0.0000	0.0000	0.1143	0.1143	0.0266	0.0266	0.0266	0.0266	0.0266	0.0267	0.0374	0.0481	0.0490	0.0490	0.0490	0.0490	0.0784	0.0784	0.0784	0.0788
	33	静的チェックの結果	FIndBugs指摘件数		51	53	53	53	55	56	72	72	72	72	72	72	71	71	71	52	52	52	52	52	52	52	52
工数	34	作業工数	作業時間																								
	35	生産性	行数/作業時間																								
計画・管理	36	プロセス管理情報																									
	37	会議実施状況																									
	38	累積リスク項目数																									
	39	リスク項目の滞留時間																									
その他成果物	40	規模[推移]																									
	41	変更[推移]																									

	15			変更	0	0	0	0	0
				削除	0	0	0	0	0
設計	16	規模[推移]	UML図数		116	116	116	116	116
	17	変更[推移]	UML図数	追加	0	0	0	0	0
				変更	0	0	0	0	0
				削除	0	0	0	0	0
	18	要件の網羅率							
プログラミング	19	規模[推移]	行数(全体)		26076	26076	26068	25988	26033
	20	変更[推移]	変更量(追加＋削除行数)		380	0	21	974	58
	21	複雑度	WMC		6.24	6.24	6.27	6.27	6.28
			LCOM		10.91	10.91	10.96	10.96	10.96
			NOC		0.74	0.74	0.74	0.74	0.74
			DIT		2.8	2.8	2.81	2.81	2.81
			CBO		10.37	10.37	10.41	10.41	10.43
			RFC		12.94	12.94	13.01	13.02	13
テスト	22	規模[推移]	テストケース数	予定	554	554	587	613	597
	23	変更[推移]	テストケース数	予定	0	0	33	26	-16
				実績	0	0	174	140	68
	24	密度	テストケース数/全体行数		0.0212	0.0212	0.0225	0.0236	0.0229
	25	消化	消化テスト数	実績	203	203	377	517	585
品質	26	レビュー状況	レビュー回数		21	21	21	21	21
	27	レビュー作業密度	レビュー時間/全体時間		1	1	1	1	1
	28	レビュー指摘率[推移]	レビュー指摘数		649	649	649	649	649
	29	欠陥件数[推移]	全体欠陥件数		19	19	19	19	19
	30	欠陥対応件数	全体欠陥対応件数		19	19	19	19	19
	31	欠陥密度	全体欠陥件数/行数		0.000729	0.000729	0.000729	0.000731	0.000730
	32	欠陥指摘率	全体欠陥件数/消化テスト数		0.0936	0.0936	0.0504	0.0368	0.0325
	33	静的チェックの結果	FIndBugs指摘件数		52	52	52	52	52
工数	34	作業工数	作業時間						
	35	生産性	行数/作業時間						
計画・管理	36	プロセス管理情報							
	37	会議実施状況							
	38	累積リスク項目数							
	39	リスク項目の滞留時間							
その他成果物	40	規模[推移]							
	41	変更[推移]							