

COSEプロジェクトにおけるEASE研究 適用事例～上流工程分析～

奈良先端科学技術大学院大学
STAGEプロジェクト研究員 松村知子

出典： 松村知子, 森崎修司, 玉田春昭, 大杉直樹, 門田暁人, 松本健一 : GQMモデルに基づく設計工程完成度計測手法の提案, NAIST-IS-TR2007005, March 2007

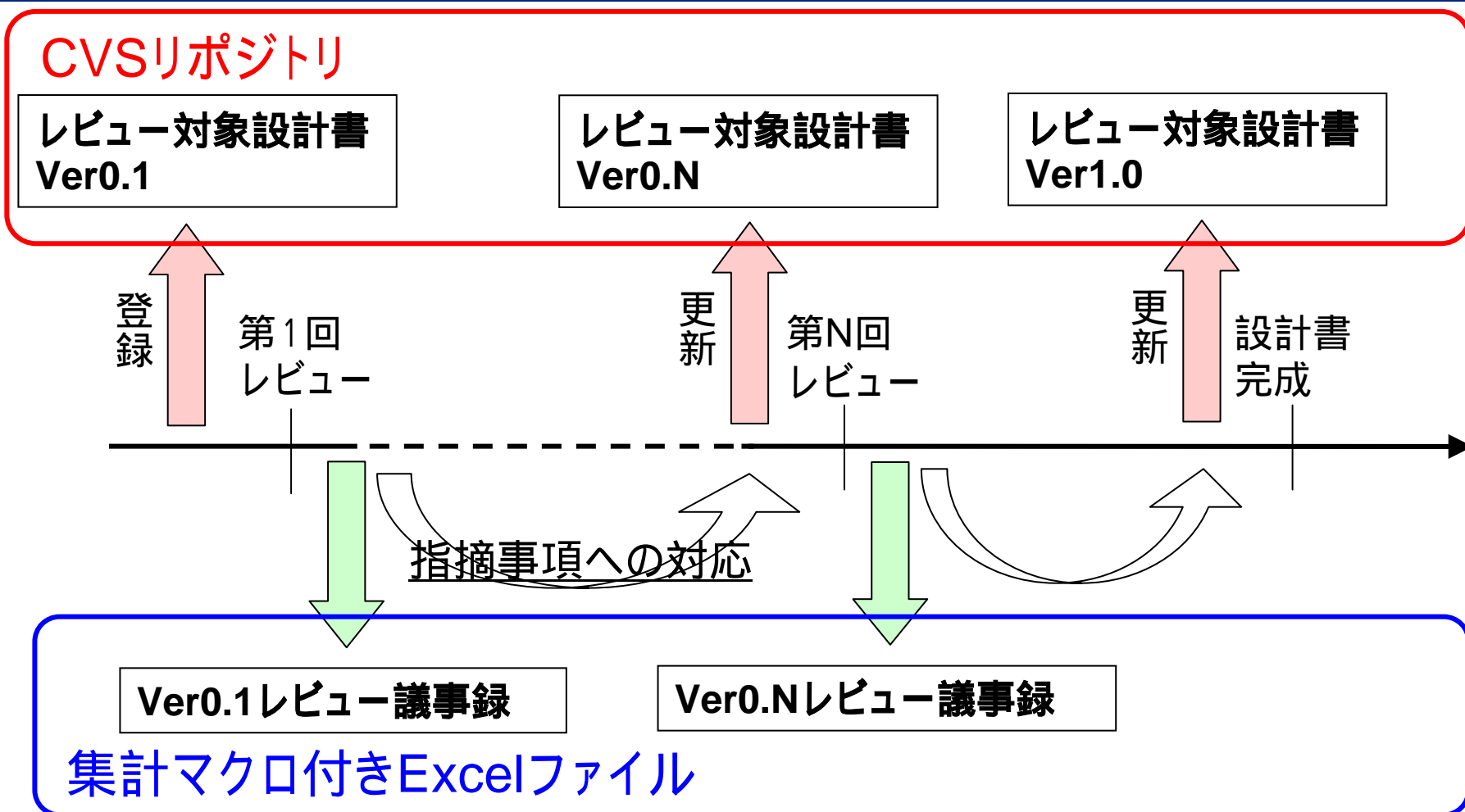
本研究の目的

- 設計成果物の変更量やレビュー結果データから抽出できる定量データを時系列でモニタし, 設計成果物の完成度を推定する
- 期待される活用方法
 - 次工程への移行可否, もしくは前工程の見直し要否を判定する
 - レビューや品質管理, バージョン管理が適切に行われているか判定する
 - 要員配置や作業分割が適切かどうか判断する

収集データ

- 基本設計書・詳細設計書・プログラム設計書の各バージョン(構成管理リポジトリ)
- レビュー議事録
 - 対象ドキュメントページ数, レビュー時間, 参加人数, レビュー日時 etc.
 - 項番, エラー現象, エラー原因, 重要度, 解決日時, (修正工数) etc.
- 仕様書・設計書の変更情報(履歴)
 - 変更日時, 変更事由(レビュー議事録ファイル名, 項番, 修正者 etc.

設計データ収集プロセス



変更量計測による設計完成度の計測

- 計測データ
 - バージョン毎の変更量
 - 変更箇所数 (DiffDoc使用)
 - 変更箇所中非重要項目 (変更履歴・表紙・目次等の変更) 数
 - 変更箇所の文字数
 - 追加 = 追加箇所の文字数 + 変更箇所の変更後文字数
 - 削除 = 追加箇所の文字数 + 変更箇所の変更前文字数
 - 変更 = 変更箇所の変更後文字数 + 変更箇所の変更前文字数
 - バージョン毎の対応欠陥数
 - レビュー議事録及び変更履歴
- 計測単位
 - 仕様書・設計書別

設計書の差分調査(DiffDoc出力例)

Diff Doc

Date and Time: Fri, 02/09/2007 9:35:20 AM

Original: C:\Documents and Settings\松村知子\My Documents\NAIST\ISESE2002\paper1.doc
Modified: C:\Documents and Settings\松村知子\My Documents\NAIST\ISESE2002\paper9.doc

Compare File Results:

7 Changes Found
2 Additions Found
3 Deletions Found

変更・追加・削除
箇所数

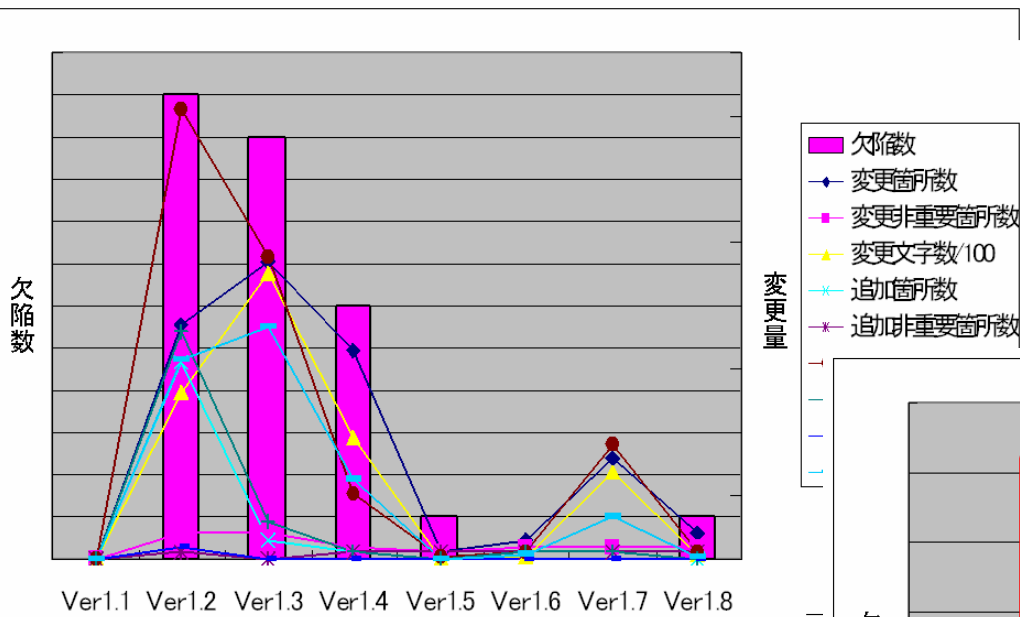
12	Graduate School of Information Science, Nara Institute of Science and Technology	
13	8916-5 Takayama-Cho, Ikoma,	
14	Nara 630-0101	
15	+81-743-72-5312	
16	akito-m@is.aist-nara.ac.jp	
17		8
18	Ken-ichi MATSUMOTO	9
19	Graduate School of Information Science, Nara Institute of Science and Technology	10
20	E-mail: tomoko-m, akito-m, matumoto @is.aist-nara.ac.jp 8916-5 Takayama-Cho, Ikoma,	11
21	Nara 630-0101	12
22	+81-743-72-5312	
23	matumoto@is.aist-nara.ac.jp	
24		13
25		14
26		15
27	Abstract ABSTRACT	16
28	In the field of legacy software maintenance, there unexpectedly arises a large number of implicit coding rules, which we regard as a cancer in software evolution. Since such rules are usually undocumented and each of them is recognized only by a few members in a maintenance team, a person who is not aware of a rule often violates it while doing various maintenance activities such as adding a new functionality or repairing faults. The problem here is not only such a violation introduces a new fault but also the same kind of fault will be generated again and again in the future by different maintainers.	17
	This paper proposes a method for detecting code fragments that violate implicit coding rules. In the method, an expert maintainer, firstly, investigates the cause field of legacy software maintenance, there unexpectedly arise a large number of each failure, described implicit coding rules — which are seldom written down in the past failure reports, and identifies specification documents or design documents — so software becomes more complicated than it used be.	

削除文章(青字)

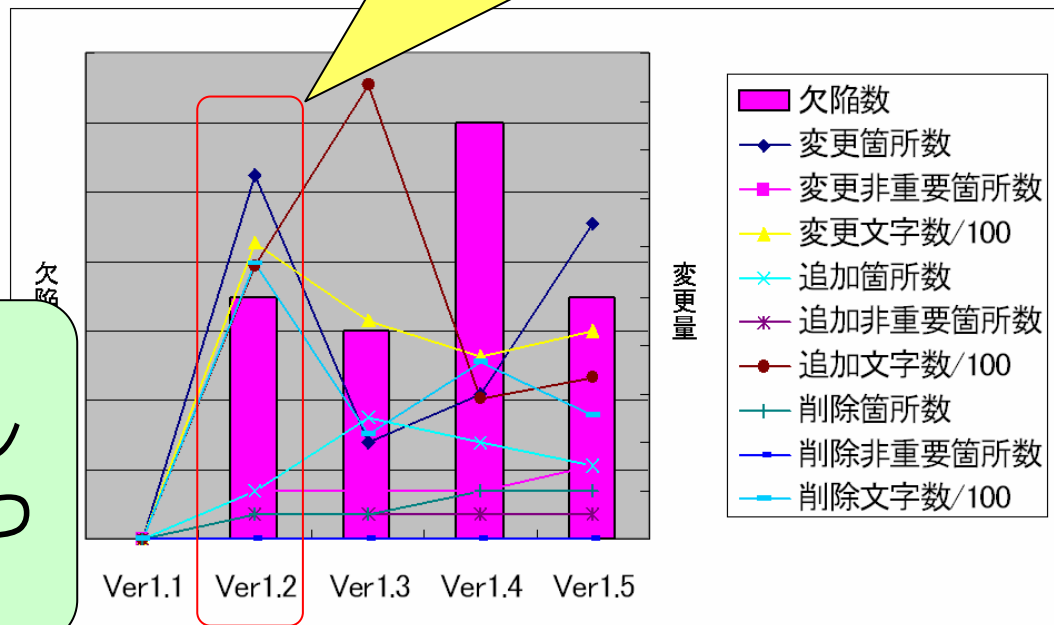
変更文章(赤字)

追加文章(緑字)

変更推移の計測・可視化



◆ Ver1,1で見つかった欠陥で Ver1.2で修正された欠陥数
 ◆ Ver1.1とVer1.2の文書の差分



設計工程品質評価会議で使用
 設計工程の状況についてインタビューし、設計書の完成度について確認

変更内容・変更量と品質の関係分析

- 上流工程での変更量・レビューデータと、下流工程発見された設計工程混入欠陥との関係を統計的に分析
 - 重回帰分析, 判別分析
- 説明変数(上流工程収集データ)
 - レビューデータ
 - 欠陥数, 重要欠陥数, 欠陥密度, 重大原因欠陥数 etc.
 - 変更量
 - 重要変更箇所数, 変更文字数, 重要追加箇所数, 追加文字数, 重要削除箇所数, 削除文字数 etc.
- 従属変数(下流工程収集データ)
 - 基本設計・詳細設計工程で混入し, テスト工程で発見された障害数

要求の確認不足
設計条件の確認不足
実現方式の検討不足

分析結果(一部)

	選択された変数		標準化係数	有意確率
	(定数)			0.148
基本設計 欠陥数	重大原因欠陥数	0.302	1.136	0.000
	重要欠陥密度(/全ページ数)	1.780	-0.696	0.004
	(定数)			0.171
	重要削除箇所数		1.850	0.000
	重要変更箇所数	0.058	-3.806	0.000
詳細設計 欠陥数	削除文字数/100	0.031	2.663	0.000
	欠陥密度(/全ページ)	2.45	0.921	0.000
	欠陥密度(/新規ページ)		-0.687	0.000
	レビュー時間(/新規ページ)		0.295	0.003
	新規ページ数		-0.292	0.006

重大原因欠陥数が多い場合、基本設計で見逃された(混入した)障害が多い

重要変更箇所数が多い場合、詳細設計で見逃された(混入した)障害は少ない

削除文字数が多い場合、詳細設計で見逃された(混入した)障害は多い

平成18年度の分析考察

- 可視化することにより，従来の欠陥密度よりも詳細な情報の入手が可能になった
- 欠陥の内容や変更の内容がわかると，さらに有益な情報となる
 - “欠陥”には誤字・脱字が含まれている場合がある
 - 文言の修正など，修正量が多いが重要度が低い変更がある
 - “欠陥”にカウントされていない変更も存在する
- レビュー形態を考慮する必要がある
 - 基本設計は社間で行い，詳細設計は社内で行った

END

今後の分析方針

- 変更内容の分類方法を検討し, 各分類の変更数(量)を計測することで, より明確な完成パターンを抽出する
- 分類例
 - 修正機能: 機能要件, 非機能要件, エラー処理
 - 記述形態: 図・表・説明文・見出し
 - 修正内容種類: 機能, 項目, 分類, 数値・パラメータ, 表現方法
 - 修正範囲: 単一セクション, 複数セクション, 複数文書
 - 修正種類: 追加, 変更, 削除, TBDへの追記

設計工程混入障害調査(参考)

- テスト工程で発見された障害の詳細調査
 - 混入工程, 問題現象, 問題原因等から, 実際の設計書上の記述を特定し, 変更履歴を精査
- 障害の混入状況
 - 直接設計書に問題はない(性能問題など): 2件
 - 社間の調整ミス(コミュニケーションミス): 2件(原因は1つ)
 - 前年度の潜在バグ: 1件
 - 問題箇所の特定不可: 2件
 - 改造時の修正ミス(Ver0.1): 3件
 - 修正中に混入した(Ver0.2以降の変更): 5件

EASEインプロセス分析手法の目的と適用

- プロジェクトをリアルタイムにモニタし, 遅延のリスクとなる問題状況の早期検出・対策を支援する
 - モニタ対象データ:
 - プログラムコード(構成管理リポジトリ)
 - テスト工程での障害(障害管理リポジトリ)
 - 利用ツール
 - EPM(Empirical Project Monitor): データ収集ツール
 - EPM Pro*: データ分析・グラフ化ツール
- COSEで2005～2006年度にわたって適用・検証*

*参考文献: 松村知子, 勝又敏次, 森崎修司, 玉田春昭, 大杉直樹, 門田暁人, 楠本真二, 松本健一 : 自動データ収集・可視化ツールを用いたリアルタイムフィードバックシステムの構築と試行, NAIST-IS-TR2007001, February 2007

GQM(Goal/Question/Metric)パラダイムとは？

- V. R. Basiliらによって提案された総合的なソフトウェア計測の枠組み*
- 計測はトップダウンで行われるべきである
 - まず計測の目標があって、その目標を遂行するために尺度が定義され、計測が行われなければならない
- データ分析は何らかの目的や仮説に基づいて行われるべきである
 - 例えば、コスト予測の改善に役立てるのか、それとも、コストを評価するのか、その目的を明確にした上で分析を行わなければならない

*V. R. Basili and D. M. Weiss: A methodology for collecting valid software engineering data, *IEEE Transactions on Software Engineering*, Vol.SE-10, No.6, pp.728-738, 1984.

EASEインプロセス分析手法で用いたモデル例

Goal

構成管理データ(CVS)と障害管理データ(GNATS)からプログラム変更,仕様変更パターンを解析し,特定のプロジェクトにおいて,プロジェクトマネージャ視点から,要求の不安定さについて評価する

その他のGoal(目的)

- ・設計の不完全さ
- ・リソース配置の適切さ
- ・コーディング品質

Question

プログラム変更頻度は?

プログラム変更規模は?

プログラム変更範囲は?

仕様変更数の増加傾向は?

Metric

A: ファイルの更新回数

B: (一定規模以上の)行削除回数

C: 追加行数

D: 削除行数

E: ファイル数

F: (一定規模以上)行削除のあったファイル数

G: プログラム行数(KSLOC)

H: 仕様変更数

上流工程への同モデルの適用

- “要求の不安定さ”, “設計の不完全さ”など, 上流工程で検出するべき
 - 下流工程で発見されても, 対応することができない
 - 対応できても, プロジェクトの遅延やコスト超過を招くことは避けられない
- 上流工程で同様のモニタリングはできないか？
 - プログラムコード 仕様書, 設計書など
 - 障害 レビュー議事録の指摘(欠陥)など

仕様書(設計書)完成度計測GQMモデル

Goal

仕様書の変更履歴とレビュー欠陥データから仕様書変更パターンを解析し、企業における特定のプロジェクトにおいて、プロジェクトマネージャの視点から、仕様書の完成度について評価する

Question

仕様書の変更頻度は？

仕様書の変更規模は？

仕様書の変更範囲は？

レビューコストは？

レビューによる欠陥量は？

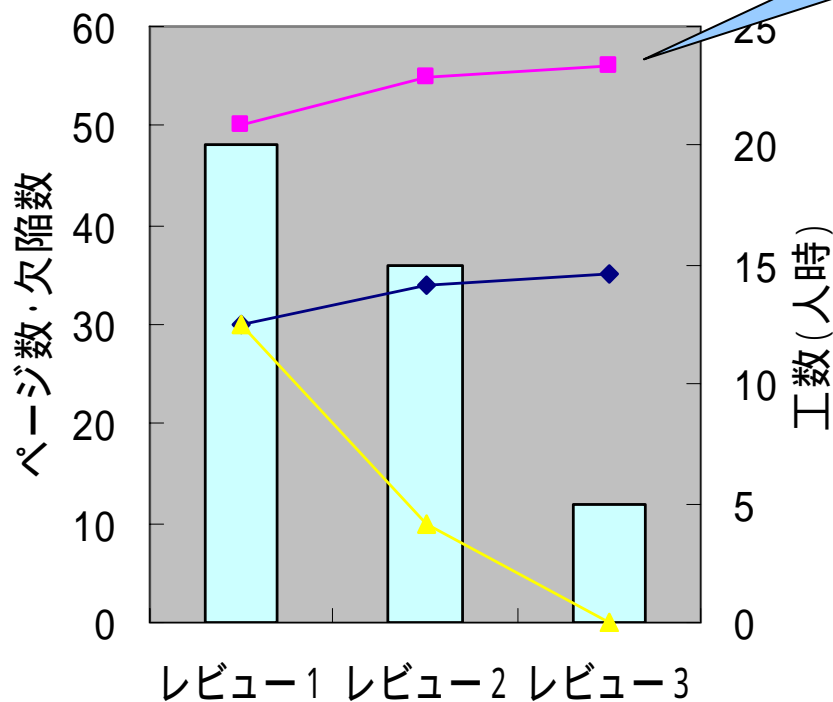
Metric

仕様書の変更管理情報(構成管理情報)から取得

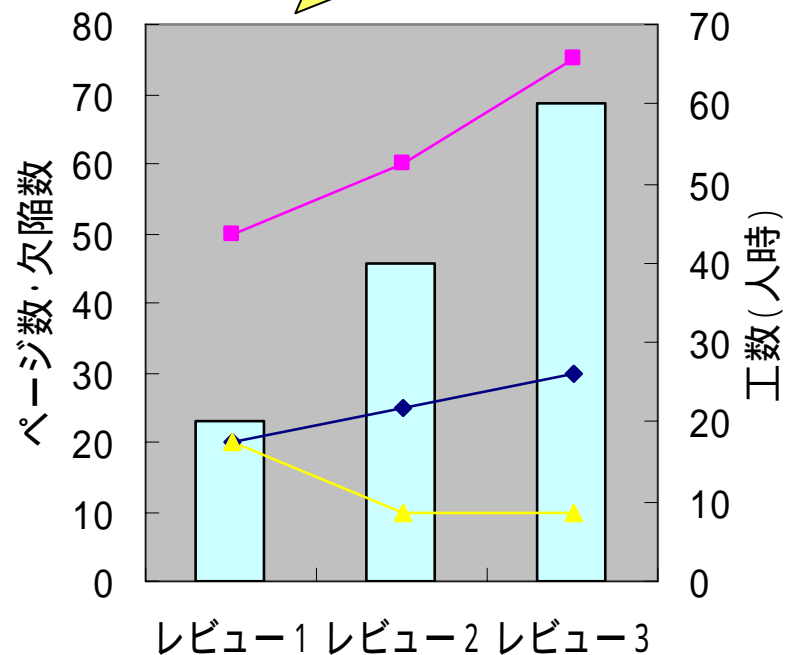
レビュー議事録(問題記述票)から取得

レビュー実績可視化例

順調に完成度が
上がっている



発見欠陥数が収束しない
解決しない欠陥がある
レビュー時間が増加し続ける
設計書が増加し続ける



変更量可視化例

順調に完成度が
上がっている

設計書が増加し続ける
変更範囲が拡大している
変更量・変更工数が収束しない

