

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成25年 5月 25日現在

機関番号：34425

研究種目：基盤研究（C）

研究期間：2009～2012

課題番号：21500045

研究課題名（和文）マイクロプロセス検出によるシステムの
複雑さの増加を防止するソフトウェア開発環境研究課題名（英文）An software development environment
for low complicate system with micro-process detection

研究代表者

花川 典子（HANAKAWA NORIKO）

阪南大学・経営情報学部・教授

研究者番号：60351673

研究成果の概要（和文）：本研究はシステムの複雑さに影響を与えるマイクロプロセスの自動検出とそれを提示する開発環境の構築をおこなった。システム開発を上流工程と下流工程に分け、上流工程では要件定義などのミーティングにおいてシステムの品質（複雑さ）に影響を与えるマイクロプロセスを定義した。実プロジェクトで検証した結果、リリース後の重大な障害発生を適合率100%、再現率66.7%で予測することができた。下流工程では、システムの複雑さを誘引するプロセスの複雑さの尺度を提案し、工程管理表の更新履歴からプロセスの複雑さを計測した。結果、リリース後の致命的な障害件数とプロセスの複雑さの関係を示すPCPQモデルを構築した。上流と下流工程でシステムの複雑さに影響を与える可能性のあるマイクロプロセスを特定し自動検知をするツール群を開発した。

研究成果の概要（英文）：We have proposed an environment for software development including automatic detection of micro-processes that cause system complexity. In upper development phases, we detected the micro-process in meetings for system requirement acquisition. The detected micro-process influenced system quality (complexity) in 100% precision and 66.7% recall. In lower development phases, we detected the micro-process in workflow management tables using new measurement "Process complexity". The micro-process influence system complexity, as a result, we found that the micro-process leads significant system faults after release. In addition, we have developed a system with automatic detecting these processes.

交付決定額

（金額単位：円）

	直接経費	間接経費	合計
2009年度	900,000	270,000	1,170,000
2010年度	800,000	240,000	1,040,000
2011年度	900,000	270,000	1,170,000
2012年度	700,000	210,000	910,000
総計	3,300,000	990,000	4,290,000

研究分野：ソフトウェア工学

科研費の分科・細目：情報学・ソフトウェア

キーワード：ソフトウェアプロセスの複雑さ、発話ログ、メトリクス

1. 研究開始当初の背景

近年ではコンピュータシステムの社会的

な重要性が高まっているにも関わらず、レガシーシステムに代表されるように、ソースコ



図1 発言分析用ツール

ードを始めとしてシステム構成や運用形態の複雑さが増し、安定的なシステム稼働を阻害する一因となっている。従来研究のシステムの複雑さの計測の研究は、すでに複雑になったシステムやソースコードの計測である。すでに複雑となってしまったシステムは容易に再構築できず、そのままリリースされ、その後、重大な障害を引き起こして社会に多大な影響を及ぼすケースもある。

つまり、システムが複雑になってしまう前に、システムが複雑になる可能性をあらかじめ察知する必要があった。ソースコード等の完成時期に近いプロダクトで複雑さを計測するだけでなく、ソースコードがまだ存在しない時期に、システムが複雑になる要因をあらかじめ排除する必要性が高まった。

2. 研究の目的

本研究の目的は、システムの複雑さに影響を与えるマイクロプロセスを自動検出し、システムの複雑さの増加を防止し、最終的に高品質なシステムを社会へ提供することである。ソースコードやシステム構成、運用形態などシステムの複雑さが増加すると、開発や保守が困難となり、結果としてシステムの品質が低下して重大な障害発生につながる。従来の研究では、システム開発工程の終盤に完成に近づいたシステム（ソースコード等）を計測し、システムの複雑さを明示した。しかし、プロジェクトは終盤となっており、リファクタリング等の大幅な工数確保が難しく、複雑なシステムのままリリースされる。保守や運用が困難になると同時に重大な障害の発生にもつながった。本研究はシステムが複雑になる前に、あらかじめ複雑になる可能性を察知し、システムの複雑さが増加することを防止し、高品質なシステムを社会へ提供することが研究の目的である。

3. 研究の方法

本研究では、システム開発を上流と下流に分ける。上流では要件定義や基本設計等のミーティングに着目し、ミーティングの中で発生するシステムの複雑さに影響を与えるプロセスを検出する。下流工程では、工程管理表の更新履歴からシステムの複雑さに影響

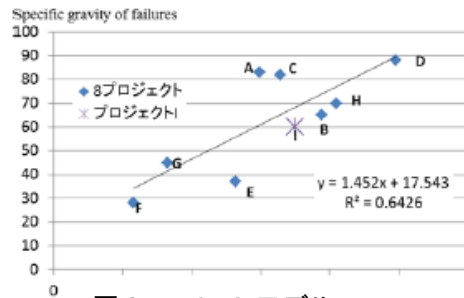


図2 PCPQ モデル

を与えるプロセスを検出する。

まず、上流工程では、要件定義や設計に関する会議の映像を蓄積する。会議では要件定義や基本設計について顧客とSEの間で合意を形成するとともに、懸案事項などを話し合う。この映像記録に関して、SEの発言と顧客の発言の時間的推移をツール（図1参照）で分析する。システム品質に影響を与える発言のマイクロプロセスを特定する。特定方法は、リリース後の重大な障害の原因や仕組みを話し合った会議の発言プロセスを抽出する。分析の結果、重大な障害の原因となる機能の会議での発話は、(1) SEの発言時間が少ない、(2) SEと顧客の発言が1対1でない（SEのひとつの問いかけに複数の顧客が何度も発言する）の特徴があることが分かった。この特徴をシステムの複雑さに影響を与えるプロセスとした。

次に、下流工程では工程管理表を利用する。あらかじめ計画された工程管理に、プロジェクト途中で追加された作業（フラグメントプロセスと呼ぶ）が、開発プロセスを複雑にして、最終的にリリース後に重大な障害を誘引すると考えた。その検証のために、8つの実プロジェクトの工程管理表と障害管理表を入手し、フラグメントプロセスを識別してプロセスの複雑さとリリース後の障害の関係を分析し、PCPQモデルを構築した（図2参照）。PCPQモデルはプロセスの複雑さとリリース後の重大な障害の関係を示すモデルである。プロセスの複雑さが大きくなるプロジェクトはリリース後の重大な障害の発生が多くなることが分かった。

上流工程、下流工程ともにプロジェクトの重要な資料や作業にかかわるので、研究代表者はそれぞれのプロジェクトに実際に参加した。これによって、上流工程のすべての要件定義会議、基本設計会議等に参加し、さらに工程管理表上に記録していない追加作業の情報を得ることができた。

4. 研究成果

研究成果として、(1) 上流工程のミーティングのマイクロプロセスの検出、(2) 下流工程の工程管理表するマイクロプロセス、(3) システムに影響を与えるマイクロプロセス

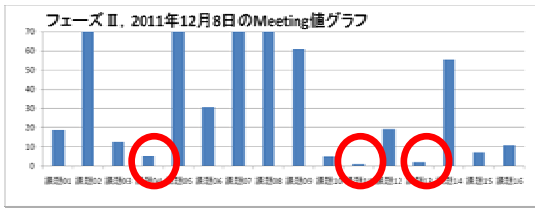


図3 1会議の各話題の Meeting 値

を自動検出する開発環境に分かれる。

(1) 上流工程のマイクロプロセス

前述したように、上流工程では要件定義や基本設計工程の顧客と SE が議論と合意をとる会議のマイクロプロセスに着目する。リリース後に重大な障害の原因となる機能や仕組みの議論は、SE の発言時間が短い、SE と顧客の質疑応答が 1 : 1 でない (SE の一つの問いかけに複数の顧客が連続して発言する) という発話プロセスがあることがわかった。この 2 つの特徴をつかって Meeting 値という定量的メトリクスを以下のように定義した。

$$\text{Meeting} = \text{SE}_{\text{talk}} / \text{CUS}_{\text{repeat}} \dots (1)$$

SE_{talk} : SE の 1 回の平均発言時間

$\text{CUS}_{\text{repeat}}$: SE のひとつの質問に対する顧客の連続発言回数 (最大値)

14 回の会議 (総時間 30 時間) の各話題の Meeting 値を計測した。図 3 に 1 回分の会議 (約 2 時間) の 16 話題の Meeting 値を示す。Meeting 値が極端に低い話題が議題 4、議題 11、議題 13 であった。Meeting 値の低い 3 つの議題とリリース後に発生した障害の原因の関係を調査すると、3 つの議題で議論された機能が原因で 2 つの障害が発生したことがわかった。

14 回すべての会議とリリース後の 46 件の障害で検証した結果、Meeting 値は適合率 100%、再現率 66.7% でリリース後の重大な障害の発生を予測することができた。したがって、要件定義や基本設計工程で Meeting 値を計測することによって、将来に重大な障害の原因となる箇所を予測することが可能となる。

(2) 下流工程のマイクロプロセス

下流工程では、工程管理表からフラグメントプロセスを抽出してプロセスの複雑さの尺度を計測した。フラグメントプロセスとは当初のプロジェクト計画にない途中で追加された一連の作業群を意味する。途中で急に作業が追加されることによって、プロジェクトのスムーズな進行が阻害され、結果としてプロダクトの品質が悪くなるという発想である。

8 プロジェクトの工程管理表を入手して、フラグメントプロセスを検出した。抽出方法は工程管理表の更新履歴を利用して、後から追加された作業をフラグメントプロセスとした (図 4 参照)。すべてのプロジェクトの

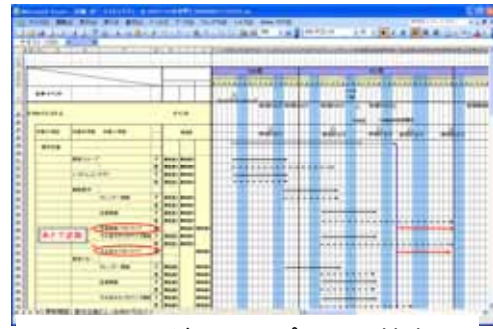


図4 フラグメントプロセス抽出

フラグメントプロセスを検出した後、提案したプロセスの複雑さの時系列変化を計測した。時刻 t のプロセスの複雑さの定義を以下に示す。

$$\text{PC}_{(t)} = (\text{Dev}_{(t)} * \text{term}_{(t)}) * \text{Concurrent}_{(t)}$$

$$\text{Concurrent}_{(t)} = 1 + L_{(t)} * m_{(t)} \dots (2)$$

$\text{Dev}_{(t)}$ はひとつのフラグメントプロセスに携わる開発者数、 $\text{term}_{(t)}$ はフラグメントプロセスの期間、 $L_{(t)}$ は当該フラグメントプロセスと同時実行されるプロセス数を示す。ひとつのプロジェクトのプロセスの複雑さの PC 値の時系列変化を図 5 に示す。

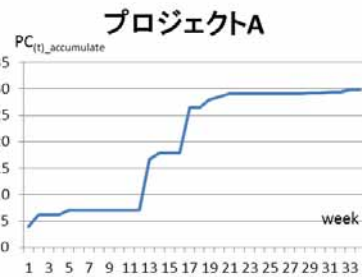


図5 プロセスの複雑さの時系列変化

PC 値の時系列変化を 8 つのプロジェクトで計算した。同時に障害管理表からリリース後の障害を分析し、障害を埋め込んだ可能性のある作業を工程管理表より特定した。その結果の一部を図 6 に示す。



図6 プロセスの複雑さと障害の原因の埋め込み作業の関係

図 6 の S とは重大な障害、SS はシステムダウンなどの致命的な障害を意味し、数値はその件数を示す。プロセスの複雑さの PC 値が急激に増加した前後の作業にて、重大な障害

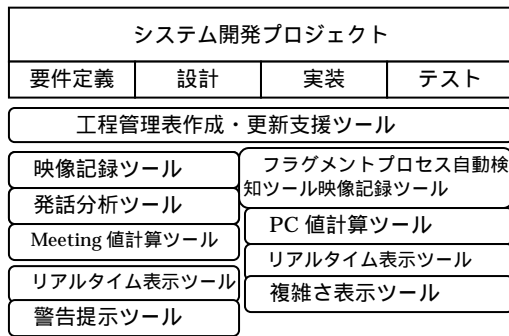


図7 開発環境のツール群

や致命的な障害の原因を埋め込んだ作業を実施していることがわかる。したがって、プロジェクト実施中の PC 値の変化をリアルタイムで監視することで、重大な障害を埋め込みやすい作業を予測することが可能となる。さらに、プロジェクト終了時の PC 値とリリース後の重大な障害の関係を示す PCPQ モデルを構築した(図2参照)。本モデルでは PC 値を計測することによって、他のプロジェクトのリリース後の品質の相対的な関係を知ることができる。つまり、過去に実施したプロジェクトと同等程度の品質である等ということがリリース時(まだ、運用される前)に予測することができる。

(3) システムに影響を与えるマイクロプロセスを自動検出する開発環境

上流工程、下流工程ともにシステムに影響するマイクロプロセスの検出方法を提案した。それぞれ、ミーティング映像記録、工程管理表等を用いて、システムの複雑さに影響を与えるマイクロプロセスを検出することが可能である。

しかし、それぞれのデータを手動で分析するとリアルタイム性に欠ける。つまり、蓄積されたデータを時間をかけて分析し、マイクロプロセスを検出できたとしても、そのプロジェクトはすでに先に進んでいて、マイクロプロセス検出の有効性が低くなる。

そこで、マイクロプロセスを自動で検出する開発環境を提供する。本開発環境は 10 個のツール群で構成されている(図7参照)。それぞれのツールについて紹介する。

まず、図7の から までは上流工程のミ

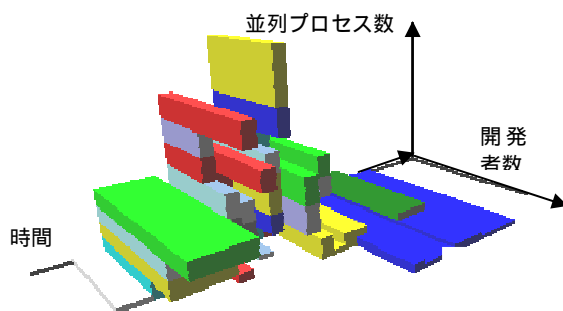


図9 複雑さ表示ツール

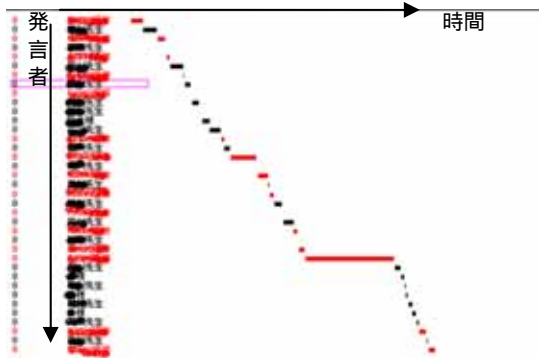


図8 発話リアルタイム表示ツール

ーティングにおいてシステムに影響を与えるマイクロプロセスを自動検出するためのツールである。まず、の映像記録ツールでは会議の映像に会議情報をタグとして付加し、データベースに蓄積する。は図1に示すツールであり、発話を分析する。では式(1)に従い、リアルタイムに記録され分析されたデータを利用して、Meeting 値を計算する。さらにその結果をグラフにてリアルタイム表示するのが、ツール(図8参照)であり、閾値以上になった時に警告を各ミーティング参加メンバーへメール配信するのがツールである。

下流工程では図の から のツール群で自動的にマイクロプロセスを検出する。まず、工程管理表からフラグメントプロセスを検出するので、工程管理表作成・更新をおこなうツールを市販の MS-Project をベースに開発した。さらに MS-Project のデータからフラグメントプロセスを自動で抽出するツールを作成し、ツールでフラグメントプロセスのデータをもとに PC 値を計算した。さらに PC 値を CSV ファイルに出力して、MS-Excel グラフ表示させるツールがある(出力例は図5)。さらに、プロセスが複雑に進化した様子を可視化して、管理者に把握しやすい環境をツールで提供した(図9参照)。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

〔雑誌論文〕(計 15 件)

1. 尾花将輝, 花川典子, 飯田元, ソフトウェア開発プロセスの並列作業に基づくプロセスの複雑さの提案, コンピュータソフトウェア論文誌, 査読有, ソフトウェア科学会, VOL.29 NO.4, pp.278-292, Nov. 2012.
2. Noriko Hanakawa, Masaki Obana, "A PLAGIARISM DETECTION SYSTEM FOR REPORTS BASED ON A LARGE-SCALE DISTRIBUTION ENVIRONMENT USING IDLE

- COMPUTERS ” , The 15th IASTED International Conference on Computers and Advanced Technology in Education CATE 2012 ,査読有 ,pp.12-19, Jun. 2012 .
3. Noriko Hanakawa, Masaki Obana, “ A Case Study: a metrics for meeting quality on a software requirement acquisition phase ”, International Conference on Product Software Development and Process Improvement Profes2012 , 査読有 , pp.260 274, Jun. 2012.
 4. Noriko Hanakawa, “ A process refactoring for software development with process complexity and activity priority lists ”, Proceedings of the Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement (IWSM/ Mensura2011), 査読有 , pp.209 214, Nov.2011.
 5. Masaki Obana, Noriko Hanakawa, Hajimu Iida, ” A process complexity- product quality (PCPQ) model based on process fragment with workflow management tables ”, International Conference on Product Software Development and Process Improvement Profes2011 , 査読有 , pp.171 pp.185, Jun. 2011.
 6. 尾花将輝, 花川典子, 飯田元 : ソフトウェア開発プロセスの複雑さに着目した PCPQ モデルの構築, ソフトウェアエンジニアリングシンポジウム 2011 (SES2011) , 査読有 , 学生奨励賞受賞 . 2011 年 9 月 .
 7. 尾花将輝, 花川典子, 携帯情報端末を用いた大人数授業改善のための p-HInT システムの構築, コンピュータソフトウェア論文誌 , 査読有 , ソフトウェア科学会 , Vol.27, No.4 , pp.114-132 , Nov. 2010.
 8. Noriko Hanakawa, “ A project reliability growth model based on communication for software development ”, International Journal of Knowledge Engineering and Software Engineering, 査読有 , Volume20, Issue5, pp.665-677, August 2010.
 9. Masaki Obana, Noriko Hanakawa, Norihiro Yoshida and Hajimu Iida, “ Process Fragment Based Process Complexity with Workflow Management Tables ”, International Workshop on Empirical Software Engineering in Practice IWESep2010, 査読有 , pp.7-12, Dec. 2010.
 10. Noriko Hanakawa, Masaki Obana, “ Lecture Improvement based on Twitter Logs and Lecture Video using p-HInT ”, the 18th International Conference on Computers in Education (ICCE2010), 査読有 , pp.328-335, Dec. 2010.
 11. Noriko Hanakawa, Masaki Obana, “ MOBILE GAME TERMINAL BASED INTERACTIVE EDUCATION ENVIRONMENT FOR LARGE-SCALE LECTURES ”, The Proceeding of the Eighth IASTED International Conference on Web-based Education (WBE2010), 査読有 , Mar. 2010.
 12. 尾花将輝, 花川典子, 飯田元 : 工程管理表のフラグメントプロセスに基づくソフトウェア開発プロセスの複雑さのメトリクス, ソフトウェアエンジニアリングシンポジウム 2010 (SES2010), 査読有 , pp.89-96, Sep.2010. 学生奨励賞受賞
 13. 尾花将輝, 花川典子 : プレンディッド開発プロセスにおける複雑さのメトリクスの提案 , 第 16 回ソフトウェア工学の基礎ワークショップ (FOSE 09), 査読有 , pp.221-228 , Nov. 2009.
 14. 花川典子, 尾花将輝 : システム仕様定義工程におけるミーティングの質を計測するメトリクスの提案, ソフトウェアエンジニアリングシンポジウム 2009 (SES2009) , 査読有 , pp.157-164 , Sep. 2009.
 15. 尾花将輝, 花川典子 : 使い捨てプログラムを組み込んだインクリメンタル開発プロセスの提案と実践, ソフトウェアエンジニアリングシンポジウム 2009 (SES2009) , 査読有 , pp.168 , Sep. 2009.
- [学会発表](計 0 件)
- [図書](計 4 件)
1. ソフトウェアエンジニアリング最前線 (社)情報処理学会ソフトウェア工学研究会 松下誠・紫合治編(執筆担当 工程管理表のフラグメントプロセスに基づくソフトウェア開発プロセスの複雑さのメトリクス ,pp.89-96)近代科学社 , Sep. 2010.
 2. ソフトウェア工学の基礎 XVI ソフトウェア科学会 FOSE 2009 , 中島震・鷺崎弘宜編(執筆担当 プレンディッド開発プロセスにおける複雑さのメトリクスの提案, pp.221-228) 近代科学社 , Nov. 2009 .
 3. ソフトウェアエンジニアリング最前線 (社)情報処理学会ソフトウェア工学研究会 鷺林尚靖・岸知二編(執筆担当 システム仕様定義工程におけるミーティングの質を計測するメトリクスの提

案, pp. 157-164) 近代科学社, Sep. 2009.

4. ソフトウェアエンジニアリング最前線
(社)情報処理学会ソフトウェア工学研究会 鷓林尚靖・岸知二編(執筆担当
使い捨てプログラムを組み込んだイン
クリメンタル開発プロセスの提案ト実
践 pp.186-186)近代科学社, Sep. 2009.

〔産業財産権〕

出願状況(計0件)

名称:
発明者:
権利者:
種類:
番号:
出願年月日:
国内外の別:

取得状況(計0件)

名称:
発明者:
権利者:
種類:
番号:
取得年月日:
国内外の別:

〔その他〕

ホームページ等

6. 研究組織

(1)研究代表者

花川 典子 (HANAKAWA NORIKO)
阪南大学・経営情報学部・教授
研究者番号: 60351673