

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 24 年 6 月 18 日現在

機関番号：14603

研究種目：研究活動スタート支援

研究期間：2010～2011

課題番号：22800043

研究課題名（和文）

過去の高品質な開発プロセスの再利用による開発プロセス記述支援環境の構築

研究課題名（英文）

Development process description support environment reusing high-quality software processes

研究代表者

伏田 享平 (KYOHEI FUSHIDA)

奈良先端科学技術大学院大学・情報科学研究科・特任助教

研究者番号：20585145

研究成果の概要（和文）：

過去に実施された開発プロセスの再利用による開発プロセス記述の支援を目指して研究を実施し、下記の成果を得た。(1)開発プロセスの品質を定量的に表す尺度として、作業の並列性、複雑性に着目したプロセスメトリクスを考案した。(2)(1)で考案したプロセスメトリクスを、実際の開発プロジェクトのデータを対象に抽出し、計測可能であることを確認した。(3)プロセスメトリクスとプログラムの複雑度メトリクスを利用して、実際のソフトウェア開発プロジェクトを対象に定量的なプロセスの品質評価を実施した。その結果、プロセスメトリクスを用いて作業が複雑となるプロセスを特定することが可能となった。

研究成果の概要（英文）：

In this research project, we aim to develop a process description support environment reusing high-quality software processes. We have proposed metrics derived from complexity and concurrency of software process. Using a case study of software projects, we were able to measure these metrics. Furthermore, using these metrics, we were able to measure the quality of software process.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2010 年度	1,260,000	378,000	1,638,000
2011 年度	1,160,000	348,000	1,508,000
年度			
年度			
年度			
総計	2,420,000	726,000	3,146,000

研究分野：工学

科研費の分科・細目：ソフトウェア

キーワード：ソフトウェアプロセス、プロセス評価、ソフトウェアメトリクス、品質評価

1. 研究開始当初の背景

近年、社会におけるソフトウェアの役割の重要性が増大している。一方で、欠陥発生時

の社会的損失も大きくなっている。例えば、米国では、ソフトウェアの欠陥による年間損失額は 600 億ドルにのぼる。このように、欠

陥発生時の社会的影響が大きくなっていることから、ソフトウェアを高品質に生産するための技術が必要となっている。

ソフトウェアを高品質に生産するためには、ソフトウェアの品質を評価する手法が必要である。そのためには、ソフトウェア作成後にソフトウェアそのものの品質を評価するプロダクト評価と、ソフトウェア開発中にソフトウェア開発プロセス（ソフトウェア開発に関わる作業の系列）を評価するプロセス評価を行うことが重要である。特に、プロセス評価はソフトウェアの作成中に随時実施でき、早期に問題を検出することが可能である。このため近年では特に注目されており、CMMI*2をはじめとして、開発組織においてプロセスを改善するための枠組みが多く提案されている。

開発プロセスを実施する際には、従来の実績や将来の予測をもとに計画を立てる。実施後は、計画に沿っていたかを確認し、計画に沿っていない部分を分析し原因を追及し、次のプロジェクトでの開発プロセスを改善する。このように継続的に改善を続けて行くことで、プロセスの品質を向上させて行く。

しかし実際には、開発プロセスの品質を向上させようとしても、その分析・改善に十分な時間を割くことができず、分析結果を次のプロジェクトに反映することが困難である。

2. 研究の目的

本研究では、開発プロセスの分析・評価を迅速に行い改善するため、ソフトウェア開発プロジェクトの計画時に高品質な開発プロセスを迅速に作成するための支援環境を構築することを研究目標とする。開発プロセスをプロジェクト計画時に自動的に評価することで、開発プロセス中の問題がある部分を迅速に検出することが可能となる。また、開発プロセスの作成を迅速に行うために、過去のプロジェクトでの成功例（ベストプラクティス）を、次のプロジェクトでも活かすことに着目する。これは、開発計画の立案時に、過去に実施された開発プロセスの中で品質の良かった箇所を再利用することで、高品質な開発プロセスを作成できると考えられるためである。

本研究では以下に示す4つのサブタスクを設定した。

(1) 開発プロセスの品質を評価するための尺度（プロセスメトリクス）の選定：

これまでどのような開発プロセスが品質の良いプロセスであるか、その具体的な評価基準が存在せず、評価することは難しい。開発プロセスの品質を自動的に評価するためには、開発プロセスに対して何らかの定量的な評価尺度を与える必要がある。本研究では、このような開発プロセスの品質を評価する

ための尺度（プロセスメトリクス）を選定する。

(2) 過去の開発プロセスからのベストプラクティス抽出手法：

開発組織では膨大な数のプロジェクトの結果が集約されている。このような大規模なデータの中から、次回以降のプロジェクトで再利用可能な、開発プロセス中の品質が良い箇所をベストプラクティスとし、ベストプラクティスを検出する手法を確立する。ベストプラクティスの検出にあたっては、開発プロセスにおけるパターン検出手法と、(1)で選定したプロセスメトリクスを用いる。

(3) プロジェクト実施前における開発プロセスの品質評価手法：

作成された開発プロセスの品質を定量的に評価する手法を確立する。開発プロセスの品質評価にあたっては、(1)で選定したプロセスメトリクスを利用する。この手法が確立されることで、プロジェクトを実施する前に作成した開発プロセスの品質を評価でき、その善し悪しの判断に役立つ。

(4) 開発プロセスの記述支援環境の構築：
(2)、(3)で開発した手法を統合して、高品質な開発プロセスを迅速に作成するための支援環境を構築する。これにより、開発プロセスの分析結果を十分に反映した上で、迅速に高品質な開発プロセスを作成できることができる。

3. 研究の方法

2で挙げた目的に対して、それぞれ以下のアプローチをもって研究に取り組むことを計画した。

(1) 開発プロセスの品質を評価するための尺度（プロセスメトリクス）の選定：
過去に実施された開発プロセスを分析し、開発プロセスの品質を定量的に示す尺度（プロセスメトリクス）を選定する。抽出すべきプロセスメトリクスは、個々の開発プロセスの実施状況を表現でき、自動的に抽出できる定量的な尺度である必要がある。

(2) 過去の開発プロセスからのベストプラクティス抽出手法：

過去に実施された開発プロセスの中から、ベストプラクティスを抽出する手法を確立する。これまでに、過去の開発手順のパターンを抽出する手法は数多く提案されている。本研究では、これら既存の手法を利用し、開発手順のパターン（開発プロセスの部分集合）を抽出する。そして、抽出したパターンに対して、(1)で選定したプロセスメトリクスを利用してプロセスの品質評価を行い、品質の良いパターンを絞り込む。

このような品質の良いパターンをベストプラクティクスと判断し、開発プロセスを新規に記述する際に利用できる形で蓄積してお

く。ベストプラクティクスは、開発プロセスの中でも品質の良い部分と言え、これらを用いて開発プロセスを記述することで、高品質なプロセスを迅速に作成することができる。

(3) プロジェクト実施前における開発プロセスの品質評価手法：

プロジェクト計画時に、プロジェクト管理者が作成する開発プロセスを自動的に品質評価する手法を確立する。プロジェクト計画立案作業を実施しているプロジェクト管理者が作成する開発プロセスを入力とし、その品質を立案作業中に評価する技術を開発する。

(1) で選定したメトリクスを用いることで、開発プロセスを実施する前に、その品質を定量的に評価することができる。また、(1) でプロセスメトリクスを選定する際に利用したプロダクトメトリクスから、仮に作成したプロセスを実施した際にどのような結果が生じるか(欠陥がどの程度含まれるか、など)を算出することも可能となる。

(4) 開発プロセスの記述支援環境の構築
過去に実施した開発プロセスの結果を利用して、新たな開発プロセスの記述を支援する技術を開発する。開発するシステムでは、(2) で提案した手法により、過去のプロジェクトから抽出された品質の良い開発手順のパターンをプロジェクト管理者に提示する。提示するにあたっては、開発手順のパターンを、プロセスメトリクスによる評価結果に基づき順位付けを行った上で提示する。また、(3) で提案した手法により、記述途中の開発プロセス記述をプロジェクト実施前に評価し、プロジェクト管理者にプロセスの改善点を提示する。

4. 研究成果

ソフトウェア開発の中でも、特にソフトウェアの保守プロセスに着目して研究を実施した。保守プロセスで実施される作業に関してはこれまでに多くの分析が行われており、分析対象のデータはオープンソースソフトウェア開発における開発データを中心に粒度が細かく分析に適したものが数多く存在しているためである。

一方で開発プロセスの品質を特徴付けるメトリクスはこれまで明らかにはなっていなかった。本研究では開発プロセスの並行性と複雑性に着目し、それらを定量的に評価するためのメトリクスを考案した。

開発プロセスの並行性は、1人の開発者が同時に行う作業の数や、1つの成果物を作成するのにかかる作業の数で表される。開発プロセスの複雑性は、1つの成果物を作成するのにかかる手順の多さを指す。

このように提案したメトリクスを開発プロジェクトのデータから抽出(測定)可能かを検証した。対象は中規模のオープンソース

ソフトウェア開発プロジェクトで利用されているバグ管理システムである。バグ管理システムには開発中に発見されたソフトウェアの欠陥(バグ)の対応状況が記録されている。本研究では、バグ1件への一連の対応作業(発見、修正、修正確認)を保守工程におけるプロセス(保守プロセス)と見なした。その上で、この保守プロセスの品質を評価するため、先に挙げたプロセスメトリクスを抽出した。併せて、プロダクト(ソースコード)のメトリクスも計測した。分析の結果から、提案するメトリクスが自動的に抽出できることを確認した。また、バグ管理システムに記録されていた履歴情報やソースコードの複雑度メトリクスを併用して分析することで、成果物が複雑であり、作業(バグ修正作業)も複雑となるプロセスを特定できることを明らかにした。

次に提案するプロセスメトリクスを用いて実際の開発プロセス実施中に異常(作業実施者の作業誤り)が検出できるかを、実際の開発データをもとに検証を行った。検証にあたっては、中規模のオープンソースソフトウェア開発プロジェクトのデータ、および大学の演習授業の小規模なデータセットを利用した。この分析では、プロセスメトリクスの変動を時系列に観測することで、プロセス品質が悪化する際の傾向を分析した。

今回の研究成果により、新規プロジェクトの実施前に、開発プロセスの定量的な品質評価を行うことが期待される。一方で、目標の一つとして掲げていた、プロセスのベストプラクティスの自動抽出に関しては、手動での定義・抽出にとどまっている。自動抽出に当たっては、開発プロセスのモデル化、及び形式的な記述方式についても検討を行う必要がある。また、最終的な目標として掲げていた、開発プロセスの記述支援環境に関しては、研究代表者のグループで開発を進めてきた計画立案支援システムをベースに、開発プロセス記述支援環境のプロトタイプ開発の検討を行った。

今後は今回の研究成果をもとに、過去の開発プロセスの中で高品質な部分を再利用することで、新たな開発プロセスの記述を支援する環境の構築を目指す。

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1件)

①伏田 享平, 飯田 元, ソフトウェアプロ

セスのモデリングとそれに基づく管理手法,
'' コンピュータソフトウェア, Vol. 29, No. 1,
pp. 61-74, February 2012.

〔学会発表〕(計 7 件)

①Raula Gaikovina Kula, Kyohei Fushida,
Norihiro Yoshida, and Hajimu Iida, ``Using
Program Slicing Metrics for the Analysis
of Bug Fixing Processes,`` The
International Workshop on Empirical
Software Engineering in Practice 2010
(IWSEEP 2010), 2010.12.7, Nara, Japan.

②伏田 享平, ``開発リポジトリデータを利用
した要件とコストの関連分析支援,`` ソ
フトウェアエンジニアリングシンポジウム
2011 ワークショップ「開発マネジメントを
取り巻く環境と課題」, 2011.1.20, 静岡.

③ Kenji Fujiwara, Kyohei Fushida,
Norihiro Yoshida, and Hajimu Iida, ``An
Approach to Investigating How a Lack of
Software Refactoring Effects Defect
Density,`` 電子情報通信学会ソフトウェア
サイエンス研究会, 2011.7.1, ソウル(韓国).

④伏田 享平, ``開発リポジトリデータを利用
した要件とコストの関連分析支援,`` ソ
フトウェアエンジニアリングシンポジウム
2011 ワークショップ「開発マネジメントを
取り巻く環境と課題」, 2011.9.12, 東京.

⑤藤原 賢二, 伏田 享平, 吉田 則裕, 飯田 元,
``オープンソースソフトウェアを対象とし
たリファクタリングが欠陥混入に与える影
響の調査,`` 日本ソフトウェア科学会 第28
回大会, 2011.9.29, 沖縄.

⑥ Shunsuke Yoshioka, Norihiro Yoshida,
Kyohei Fushida and Hajimu Iida, ``Scalable
Detection of Semantic Clones Based on
Two-Stage Clustering,`` The 22nd annual
International Symposium on Software
Reliability Engineering (ISSRE 2011)
(Fast Abstract Session), 2011.11.29,
Hiroshima, Japan.

⑦伏田 享平, 玉田 春昭, 井垣 宏, 藤原
賢二, 吉田 則裕, ``プログラミング演習にお
ける初学者を対象としたコーディング傾向
の分析,`` 電子情報通信学会ソフトウェアサ
イエンス研究会, 2012.3.13, 沖縄.

6. 研究組織

(1) 研究代表者

伏田 享平 (KYOHEI FUSHIDA)

奈良先端科学技術大学院大学・情報科学研
究科・特任助教

研究者番号: 20585145