

開発者の生産性を考慮した 開発プロジェクト間の人員最適配置に関する基礎的研究

吉田博哉[†] 田中成典[‡] 中村健二[†] 北野光一[‡]
 関西大学大学院[†] 関西大学総合情報学部[‡]

1. はじめに

近年，顧客ニーズの多様化やアジャイル開発手法の普及による開発プロジェクトの短期化が進み，同時に進行する開発プロジェクトの数が増加している．そのため，複数のプロジェクト間で，参加する開発者の能力や割り当てるタスクの複雑さを考慮し，各々のプロジェクトに適切な開発者を割り当て，開発のスケジューリングを行うことが困難である．この問題を解決するために，既研究では，タスクの遅延時のリスクに注目する手法[1]や，ファジィ認知マップを利用した手法[2]を用いて，効率的にタスクを割り当てるシステムを開発している．しかし，既存の手法は，開発者の生産性をプロジェクトマネージャの判断で設定する必要があり，開発者の生産性を適切に判断することが困難である．そこで，本研究では，システム開発の工数見積の指標である FunctionPoint とソースコード品質測定を行う C&K メトリクス[3]を使用して開発者が過去に作成したソースコードから生産性を分野別に算出[4][5]し，算出した生産性を利用して他のプロジェクトの開発スケジュールを考慮した開発スケジュールを立案する手法を提案し，本研究の有用性を検証する．

2. 研究の概要

本研究は，ソースコードを解析して開発者の生産性を測定し，適切なタスクの割り当てを行うことを目的とする．本システムは，図1に示すように，ソースコード解析機能とタスク割り当て機能により構成される．

2.1 ソースコード解析機能

ソースコード解析機能では，まず，MUDABlue アルゴリズム[4]を使用して，開発者が CVS に登録したソースコードをカテゴリ別に分類する．次に，分類したソースコードの品質

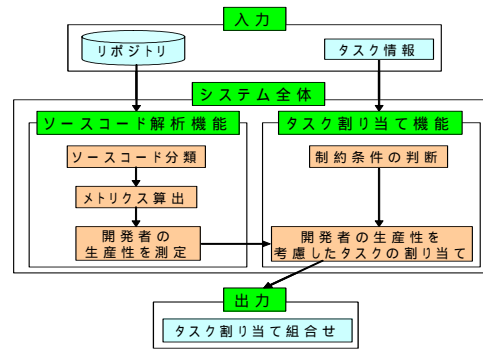


図1 本システムの概要

と工数の測定を行う．品質の測定では，秦野らの手法[3]を用いて開発者のソースコードの品質の偏差値を算出する．工数の測定では，Sneedの手法[5]を用いて FunctionPoint を算出する．次に，ソースコードが CVS に記録された時間情報と，ソースコードの評価の変化量から，開発時間を算出する．最後に，品質偏差値，FunctionPoint と開発時間から開発者の生産性を算出する．

2.2 タスク割り当て機能

タスク割り当て機能では，システム開発に参加する開発者の生産性情報と，開発するシステムのタスク情報の2つを解析し，最適なタスクの割り当てを行う．プロジェクトに参加する開発者の数や開発プロジェクトのタスクの数の変化により計算時間が大幅に増加することを防ぐため，遺伝的アルゴリズムを使用してタスクの割り当てを行う．まず，タスク開発者情報には，開発者の生産性と，開発者が所属する他のプロジェクトのスケジュール情報を使用する．次に，タスク情報には，タスクが開発される順序の制約条件と，タスクのカテゴリ情報が含まれるものとする．カテゴリ情報は，ソースコードの分類時に作成したカテゴリを使用するものとする．本提案で使用する遺伝的アルゴリズムは，個体をタスク割り当ての組合せとし，遺伝子を個々の開発者情報とし，開発者情報とタスク情報を制約条件として，実現可能なタスク割り当ての中で総開発時間が短くなる個体を算出する．

Fundamental Research on Best Arrangement of Developers between Development Projects with Developer's Productivity
[†]Hiroya Yoshida, Kenji Nakamura
 Graduate School of Informatics, Kansai University, 2-1-1 Ryouzenji-cho, Takatsuki-shi, Osaka, 569-1095, Japan
[‡]Shigenori Tanaka, Koichi Kitano
 Faculty of Informatics, Kansai University, 2-1-1 Ryouzenji-cho, Takatsuki-shi, Osaka, 569-1095, Japan

3. システムの実証実験と考察

実証実験では、本研究で考案した手法の有効性を確認するため、既存研究で提案された手法を用いたタスク割り当てとの比較し、実際に必要となった開発時間を比較した。本実験では、顧客管理システム、Web 予約システムと掲示板システムの開発プロジェクトを対象とした。

3.1 実証実験

実証実験として、まず、開発プロジェクトに参加する4人の開発者の生産性を算出した。なお、ソースコードの解析に必要となる品質の標準偏差の値は、多数のオープンソースプロジェクトのソースコードを保管しているSourceForge.jp[6]から20個のプロジェクトのソースコードをC&Kメトリクスによって品質測定を行った結果から算出した。次に、開発システムに必要な機能を洗い出し、機能ごとにタスク分割を行った。次に、本システムを用いてタスク割り当ての組合せを算出する作業と、既存の手法を用いてタスク割り当ての組合せを算出する作業を行った。最後に、本システムの評価として、開発に必要な日数を見積もり、その日数でスケジューリングを行い、実際に開発した時間を比較した。プロジェクトの開発に必要な日数を表1に示す。

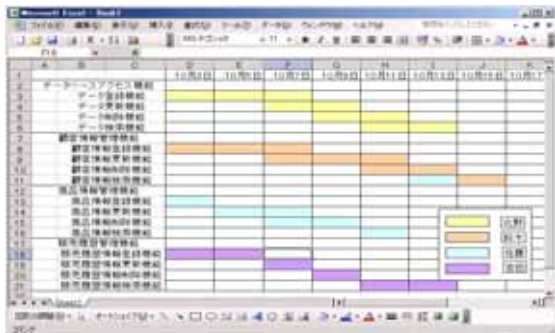


図2 実行結果

3.2 結果と考察

実験結果を元に各手法で算出された実際に必要となった開発時間を求めた結果を表2に示す。手法1は、八重樫らの手法[1]を用い、手法2は、Wojcieck らの手法[2]を用いてスケジュールの立案を行った。Web 予約システムと掲示板システムの開発時間が見積もりより短くなったのは、類似した機能の開発経験がある開発者が、類似した機能の開発に割り当てられたため、短期間で開発を行うことができたと考えられる。

4. おわりに

本研究では、ソースコードから開発者の生産性を測定し、ソフトウェア開発の準最適なスケジュールを作成する手法を考案した。実証実験

表1 各プロジェクトの開発必要日数見積

システム	日数
顧客管理システム	12.5
Web 予約システム	10.5
掲示板システム	10.0

表2 各手法で実際に開発した日数

評価項目	本手法	手法1	手法2
顧客管理システム	13.0	12.5	13.0
Web 予約システム	10.0	10.5	10.5
掲示板システム	9.0	10.5	9.5

では、既存の手法と比較して、システムをより短期間に開発できるタスク割り当ての組合せを算出することができた。この実験結果から、本研究で提案する手法の有効性を実証した。しかし、本研究で作成したシステムでは、タスクのカテゴリ情報を手動で設定しており、タスクをカテゴリ分けが不適切な場合には、正確にスケジューリングできない問題がある。そのため、入力するタスク情報の特性を自動解析し、最適なスケジューリングを行う手法の確立を目指す。

参考文献

- [1]八重樫理人, 木下大輔, 橋浦弘明, 上之園和宏, 林雄一郎, 古宮誠一: 工程遅延発生時におけるファーストラッキングによる対策案の自動立案, 電子情報通信学会論文誌, 電子情報通信学会, Vol.88, No.2, pp.215-227, 2005.2.
- [2]Wojcieck. Stach, Lukasz. Kurgan, Witold. Pedrycz, Marek. Reformat : Parallel Fuzzy Cognitive Maps as a Tool for Modeling Software Development Projects , Lecture Notes in Computer Science , Springer , Vol.1487, pp.28-33, 2004.6.
- [3]秦野克彦, 乃村能成, 谷口秀夫, 牛島和夫: ソフトウェアメトリクスを利用したリファクタリングの自動化支援機構, 情報処理学会論文誌, 情報処理学会, Vol.6, No.44, pp.1548-1557, 2003.6.
- [4]川口真司, Pankaj GARG, 松下誠, 井上克郎: MUDABlue: ソフトウェアリポジトリ自動分類システム, 電子情報通信学会論文誌, 電子情報通信学会, Vol.88, No.8, pp.1217-1225, 2005.8.
- [5]Harry Sneed : Extraction of Function-Points from Source-Code , Lecture Notes in Computer Science , Springer-Verlag , Vol.2006, pp.135-146, 2001.10.
- [6]SourceForge.jp : <http://sourceforge.jp>