

B-039

アジャイルソフトウェア開発におけるプロジェクト能力計測ツールの開発
 A management tool of measuring project capability for agile software development

花川 典子†
 Noriko Hanakawa

大蔵 君治†
 Kimiharu Ohkura

1. はじめに

近年、新しく提案されたソフトウェア開発モデルの1つに、アジャイルソフトウェア開発がある。アジャイルソフトウェア開発は、包括的なドキュメントより動作するソフトウェアを重視するという特色を持つ[2]。したがって、管理に必要なプロジェクト能力の分析に、従来のようなドキュメントベース開発で用いられていた分析手法を適応させることは難しい。この問題に対応するには、従来とは違ったアプローチでプロジェクト能力の計測を行う必要がある。本研究は、開発時のコミュニケーションに視点を置くことによって、ドキュメントに依存せずプロジェクト能力の計測を行うことを目的としている。

2. 基本的な考え方

本研究は、開発者間のコミュニケーション回数及びコミュニケーション発生時間のみ着目する。全てのコミュニケーションをいくつかの話題に分け、各話題でコミュニケーションの集中度を調べる。コミュニケーションの分散が小さいほどそのプロジェクトの能力は高いと言える(図1参照)。何故なら1つの話題に対して集中的に議論がなされ、問題を解決していると考えられるからである。一方、コミュニケーションの分散が大きい状態というのは、議論が散発的で問題解決までに時間が掛かっていると考えられ、能力は低いと言える。具体的な能力値を得るための方法として、本研究では2種類のコミュニケーション分析法を提案する。

2.1. 標準偏差と尖度を用いた分析法

1つめの分析法は、コミュニケーションの集中度を計測するために、各話題におけるコミュニケーション発生時間の標準偏差と尖度を用いる。標準偏差が低いほどコミュニケーションが集中しており、尖度が高いほど問題の解決が早いと考えられる。よって、標準偏差が低く、尖度が高いほどグループとしての能力は高いと言える。能力値は、これら二つの値から簡潔に導かれる。計算に使用するモデル式は次の通りである。

$$Cap1 = \frac{1}{\frac{1}{N} \sum_{i=1}^N SD_i} \times \frac{1}{N} \sum_{i=1}^N Kurt_i \times 100$$

$$SD_i = \sqrt{\frac{1}{m} \sum_{ii=1}^m (x_{i,ii} - \bar{x}_i)^2}, \quad Kurt_i = \frac{1}{m} \sum_{ii=1}^m \frac{(x_{i,ii} - \bar{x}_i)^4}{SD_i^4}$$

Cap1 : プロジェクト能力値
 SD_i : i 番目の話題の標準偏差。m の値が2以下の時、SD は計算できない
 Kurt_i : i 番目の話題の尖度。m の値が3以下の時、Kurt は計算できない

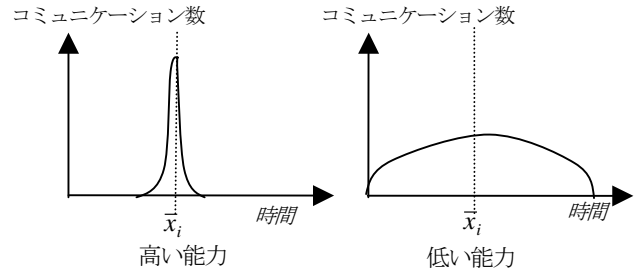


図1. プロジェクト能力のイメージ

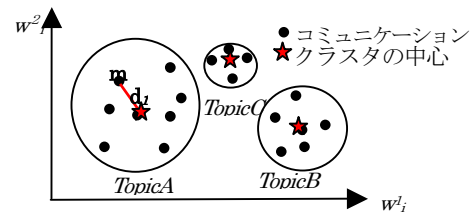


図2. 話題の分類

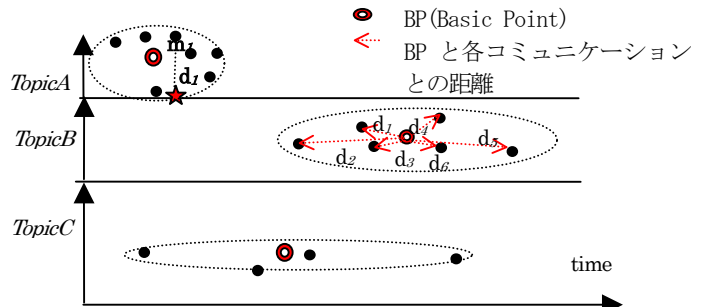


図3. 時間軸上へプロット

N : 話題の総数
 m : i 番目の話題のコミュニケーション総数
 $x_{i,ii}$: i 番目の話題の ii 番目コミュニケーション発生時間。開発時間を100として正規化した値(発生時間/実際の開発時間×100)
 \bar{x}_i : $x_{i,ii}$ の平均

2.2. ベクトル空間法を用いた分析法

2つめの分析法は、ベクトル空間法を用いてコミュニケーションの集中度を計測する。標準偏差と尖度を用いる1つめの分析法では、コミュニケーションログを予め話題別に分類しておく必要があった。ベクトル空間法を用いれば、類似した内容の会話を自動で選出し、クラスタリングすることが可能である。図2はベクトル空間法によって形成された話題クラスタの例である。各クラスタの中心から各コミュニケーションへの距離をY軸として、時間軸上にプロットしたものが図3である。時間とは、各コミュニケーションの発生時間のことである。BPとは、各コミュニケーション座標の平均を取った値である。BPから各コミュニケーションへの平均距離が、その話題におけるコミュニケーションの集

† 阪南大学 経営情報学部

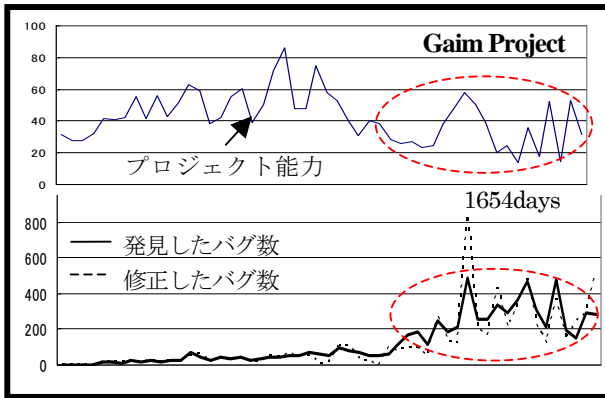


図5. 月毎の能力値と修正バグ数の推移比較

中度であると考えられる。能力値の計算式は次の通りである。

$$Cap2 = \frac{\sum_{i=1}^T D_i}{T}, \quad D_i = \frac{\sum_{j=1}^{M_i} (distance(m_j^i, BP_i))}{M_i}$$

- Cap2 : プロジェクト能力値
- T : 話題クラスタの総数
- m_j^i : i 番目話題の j 番目コミュニケーション
- M_i : i 番目話題のコミュニケーション総数
- BP_i : i 番目話題の重心
- $BP_i(x) = \text{average}(mi(x)), BP_i(y) = \text{average}(mi(y))$
- Distance(a,b) : a, b 間のユークリッド距離

3. 分析実験

本実験は、モデル式の有効性と適応性を検証する。分析の対象となるデータは、SourceForge.net[3]で公開されているプロジェクトの中から7プロジェクトを利用した。SourceForge.net は世界最大のオープンソースソフトウェア開発サイトである。分析は、各プロジェクトのバグトラッキングを対象に行った。使用した分析法は、標準偏差と尖度を用いた1つめの分析法である。

4. 実験結果

計測結果と各プロジェクトのデータを図4の表に示す。開発期間が最も長い“Gaim”プロジェクトについて、能力値の推移と修正したバグ数を比較したものが図5である。修正したバグ数と計測した能力値に相関が見られる。特にプロジェクト終盤では、発見したバグ数や修正したバグ数が増加している。それに対応するように、能力値が乱高下しながら徐々に減少している(図5の赤破線丸参照)。もし、バグレポートやバグ集計結果が存在しないとしても、計算された能力値で最近のプロジェクトの混乱状態を予測することができる。“ScummVM”プロジェクト、

プロジェクト名	経過日数	開発者数	発見バグ数	残存バグ数	コミュニケーション総数	能力値
Gaim	1654	12	6818	303	30744	44.4
Azureus	339	12	663	5	2776	50.1
ScummVM	963	32	1538	10	3358	51.4
phpMyAdmin	1274	7	1323	90	7772	48.6
eGroupWare	408	31	807	44	3481	45.1
Visual boy advance	604	4	109	9	496	48
CDexOS	1646	10	309	54	1070	16.3

図4. プロジェクトのデータと計測結果

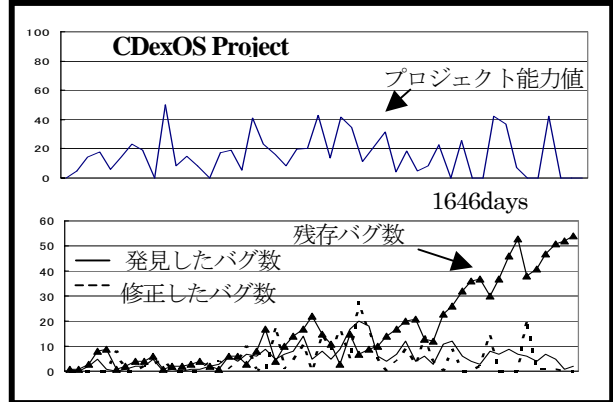


図6. 月毎の能力値と残存バグ数の推移比較

“phpMyAdmin”プロジェクト、“eGroupWare”プロジェクトにも同様の相関が見られた。“VisualboyAdvance”プロジェクトにおいても幾つか同様の相関は見られたものの、コミュニケーション数の少なさから能力値計算のできない月がしばしば存在した。

図6は、最も能力値の低かった“CDexOS”プロジェクトについて、能力値の推移と残存バグ数を比較したものである。全体として開発ペースが遅く、バグの発見に修正が追いついていないことがわかる。したがって能力値は終始低迷し、更に終盤では頻繁に0になっている。これは残存バグ数が急激に増加するプロジェクトの不活発な状態を良くあらわしている。

5. 今後の発展

今回の研究にて、提案する能力値が、バグレポートから推測できるプロジェクト状態をある程度表現できることが確認できた。今後は、企業にて本方法を採用してもらうために、コミュニケーションログの自動収集や自動分析機能を持ち合わせるツールの作成を行う。具体的には、現在は独自のスクリプト (Ruby[4])、及びプログラム (Java2 SDK 1.4[5]) のみによって実験を行っているが、Empirical Project Monitor[6]と連動させる。自動収集された標準エンピリカル SE (ソフトウェア工学) データを分析対象とすれば、CVS や Mailman、GNATS 等の開発支援ソフトのデータを透過的に分析することができる。さらに一連の分析処理をプラグイン化し、Analyzer に組み込む予定である

このようにツールを装備した本方法にて企業のプロジェクトのデータを自動収集・分析することにより、計算された能力値の精度を向上させることができるだろう。また、プロジェクトにおけるコミュニケーションとプロジェクトの様々な要素間の新しい関係や傾向の発見を期待できる。

参考文献

- [1] Noriko Hanakawa, Kimiharu Okura, “A project management support tool using communication for agile software development”, 11th Asia-Pacific Software Engineering Conference (Apec2004) (in submitting)
- [2] Manifesto for Agile Software Development <http://agilemanifesto.org>
- [3] <http://sourceforge.net/>
- [4] <http://www.ruby-lang.org/ja/>
- [5] <http://java.sun.com/>
- [6] <http://www.empirical.jp/>