

ソフトウェア開発者の特性計測のための尺度作成に向けて

角 田 雅 照^{†1} 玉 田 春 昭^{†2} 畑 秀 明^{†3} 井 垣 宏^{†4}

ソフトウェア開発チームにおいて、メンバーのパーソナリティ(性格)の影響は無視できない。本稿ではソフトウェア開発者の特性計測のための新たな尺度を作成するために必要な分析について述べるとともに、予備調査を行った。

Preliminary Analysis for Measuring Personality of Software Developers

MASATERU TSUNODA^{†1} HARUAKI TAMADA^{†2} HIDEAKI HATA^{†3} HIROSHI IGAKI^{†3}

In a software development team, influences of personality of team members are not ignorable. In this paper, we explain how to make a new measurement to identify personality of software developers. Also, we conducted preliminary analysis.

1. はじめに

ソフトウェアは多くの場合、プロジェクトマネージャ、システムエンジニア、プログラマーなどのチームのメンバーがそれぞれ役割を分担しつつ、相互にコミュニケーションを取りながら開発が行われる。そのため、ソフトウェア開発において、人間的側面は無視できない影響を及ぼす。

ソフトウェア開発チームのメンバーのパーソナリティと、チームのパフォーマンスとの関係は、これまで複数の研究において関連があることが示されている。例えば Gorla ら^{†1}は、開発者のパーソナリティ、開発者の役割、及び開発者が所属していたチームのパフォーマンスについてアンケートを行い、役割ごとにパフォーマンスが高まるパーソナリティが存在することを示している。

2. ソフトウェア開発者の特性計測

これまでの研究結果より、ソフトウェア開発において、開発者のパーソナリティは無視できない要因であるといえる。よって、パーソナリティをソフトウェア開発の分析に活用することにより有用な知見が得られると考えられるが、従来法では、主に以下の 2 点が不足している。

- (1) 分類がソフトウェア工学に特化していない。
- (2) アンケートベースで計測する必要がある。

従来法は(ソフトウェア開発に限らない)一般に当てはまる分類であり、ソフトウェア開発に適用するにはやや粒度が大きすぎる可能性がある。より適切な分類にすることにより、さらに有用な知見が得られる可能性がある。また、従来法では、ソフトウェアリポジトリなどから自動的にパーソナリティを計測すること前提としていない。自動で計測するためには、分類および計測方法を再検討する必要がある。自動計測により、データの分析や予測モデルの構築の際に、パーソナリティの利用が容易になることが期待される。

本稿では、前者の問題を解決するために、ソフトウェア開発者の特性(パーソナリティ)を計測する尺度を作成することを目指す。以降では信頼性、妥当性の高い尺度を作成するために必要となる分析について述べる。新たな尺度を作成するためには、以下の手順に従って分析を行う必要がある。

1. 開発者の特性を構成する要因(仮説)を設定する。
2. 各要因の特徴を表す質問項目を考える。
3. 主に学生を対象として、質問紙による予備調査を行う。
4. 予備調査の結果に基づき、クラスタリングを行う。適切な結果が得られない場合、要因や質問項目について再検討する。
5. オープンソースソフトウェア開発者に対して質

†1 近畿大学

Kindai University

†2 京都産業大学

Kyoto Sangyo University

†3 奈良先端科学技術大学院大学

Nara Institute of Science and Technology

†4 大阪大学

Osaka University

表 1 各分類に対応する質問項目

質問項目
風のエンジニア プログラムはまず動くものを作る。 完璧なプログラムの作成を目指している。
林のエンジニア 技術の学習を面倒に思うことがある。 古い技術を使っているプログラムは修正をしたくなる。
火のエンジニア プログラムやレポートに間違いがないかを探すことを、面倒に思うことがある。 プログラムのテストをおろそかにしてしまうことがある。
山のエンジニア 原因不明のバグが発生した場合、問題の切り分けが得意である。 課題をこなす場合、少しのトラブルは、むしろあったほうがよい。

表 2 開発者のクラスタリングと各項目の平均値

クラスタ	件数	風	林	火	山
1	6	4.1	2.1	1.5	1.8
2	3	1.8	3.3	2.5	3.1
3	2	3.8	3.0	4.8	4.0
4	1	3.5	2.5	2.5	2.0

問紙調査を行う。

- 手順 5 の結果に基づき、手順 4 と同様の分析を行う。

手順 1 の要因設定に関しては、既存のソフトウェア開発者のパーソナリティ分類のうち、(統計的に検証されていないが)妥当性が比較的高いと思われるものに基づいて行うこととする。本稿では、小野²⁾が提案している 4 つの分類(風林火山)を採用する。小野はソフトウェア開発者を以下の 4 種類に分類している。

- 風のエンジニア:迅速な設計/実装によってチームを加速させる。
- 林のエンジニア:突発的なトラブルが発生しても冷静に対処し、チームに乱れぬベースを提供する。
- 火のエンジニア:新しい技術/方法/ツールの積極的な導入によってチームやその成果物の競争力を高める。
- 山のエンジニア:厳密なエラーチェックと堅牢なプログラミングによって成果物の安定性を高める。

手順 2 では、上記の風林火山に基づき、各分類に

複数の質問項目を設定することとする。各項目は 5 段階のリッカート尺度とする。表 1 に各分類と、新たに提案する質問項目を示す。ここでは学生に質問することを想定した項目している。

3. 予備調査

情報科学を専攻している大学 3, 4 年生 13 人に対し、表 1 の質問票を用いて予備調査を実施した。内的整合性を確認するためにクロンバックの α を求めると、風、林、火、山のエンジニアにおける値はそれぞれ 0.36, 0.73, 0.54, 0.90 となった。風のクロンバックの α が 0.5 を大きく下回っていることから、風のエンジニアに関する項目については再検討が必要であると考えられる。

調査結果に基づき、データを k-means 法で 4 つに分類した。表 2 はクラスタリングの結果と、クラスタごとの各アンケート項目の平均値を示している。各項目における最大値を太字で示している。表より、クラスタ 1 は風、クラスタ 2 は林、クラスタ 3 は火と山のエンジニアを示していると考えられる。火と山の属性については、比較的関連が強い可能性がある。

4. おわりに

本稿では、ソフトウェア開発者の特性(パーソナリティ)を計測する尺度を作成するために必要と考えられる分析とその際に用いる質問項目について述べた。さらに予備調査に基づく分析結果を示した。ワークショップでは、どのリポジトリに記録されるどのアクティビティに基づいてパーソナリティを計測するか、また、各パーソナリティの開発者に対し、どのようなタスクをアサインすべきかについて議論したい。

謝辞 本研究の一部は、文部科学省科学研究補助費(挑戦的萌芽:課題番号 26540029, 基盤 C:課題番号 25330090)による助成を受けた。

参考文献

- 1) Gorla, N., and Lam, Y.: Who should work with whom?: building effective software project teams, *Commun. ACM*, vol. 47, issue 6, pp. 79-82 (2004).
- 2) 小野和俊:プログラマー風林火山, <http://blog.livedoor.jp/lalha/archives/50065532.html>