

プログラミングスキルレベル評価手法の研究

朽木 拓† 山田 敬三† 佐々木 淳†

†岩手県立大学ソフトウェア情報学部

1 はじめに

ソフトウェア開発プロジェクトのプログラミング生産性向上を図るためには、プログラムのスキルレベルを考慮した人材の最適配置が重要である。また、プログラムのスキルレベルが把握できていないと実装工程での作業見積もりが立てにくいといった問題がある。

本研究では、プログラムのスキルレベルの要素として、ロジック構成能力、プログラム生産能力、デバッグ能力について着目し、その定量的評価方法を明らかにすることを目的としている。本稿では、キー入力情報を活用したプログラム生産能力についての評価手法の提案を行い、簡単なツールを用いた試行実験の結果について述べる。

2 既存研究

プログラミングスキルの評価に関する既存研究としては、プログラムのデバッグ能力に着目したものはいくつかある。

例えば、松本ら¹⁾は、エラー寿命^{*1}に着目している。この研究では、能力の高いプログラムほどエラーを作り込まず、エラーが含まれていても短い期間でそれを取り除くことができるという考えに基づき、スキルレベルを定式化している。また、高田ら²⁾は、プログラムのデバッグ能力^{*2}をキーストロークから測定する方法を提案している。

既存研究ではいずれの方法もデバッグ作業からのみプログラミングスキルを測定しており、コーディングには着目していない。このため、プログラムの初期生成の段階からの評価とはいえない。

また、既存研究ではキーストロークを測定しているが、キー入力情報・キー入力数等が考慮されていないため、プログラムの能力を客観的に評価して利用できる情報量が不足している。さらに、無操作時間(思考時間)、調査時間等が考慮されておらず、ロジック構成能

力も考慮されていないため、プログラミング能力の総合的評価につながらない。

3 プログラミングスキルの評価方法

3.1 考え方

前記問題の解決を図るために我々は、デバッグ能力以外の観点からも能力を測定することで、より正確なプログラミングスキルを測定する方法について検討した。

プログラミングスキルの評価は、バグ数(B)とソースコードと作業ログの二つを評価することで行う。ソースコードから、サイクロマチック数($V(g')$)³⁾*3, Halstead 複雑度(V)⁴⁾*4, 最終ソースコード行数(LOC)を求める。作業ログからは、一次プロダクト作成時間:Pt(秒)、デバッグ時間:Dt(秒)を抽出する。BとDtからデバッグスキル(Ds)を求める。また、P, V, $V(g')$ からプログラム初期作成スキル(Cs)を求める。ただし、LOCが大きいかほどスキルが高くなるわけではないため、単位時間当りのキー入力数(K)が多く、コピー&ペースト回数(Cp)が少ないほどスキルが高いと考えた。そこで、LOCの代わりに $K - Cp - B$ を用いることを提案する。以上の要素から、プログラミングスキル(Ps)を求める。以下に、プログラミングスキル導出式を示す。

$$Ds = \frac{B}{Dt} \quad (1)$$

$$Cs = \frac{K - Cp - B}{Pt} f(V, V(g')) \quad (2)$$

$$Ps = Cs + Ds \quad (3)$$

3.2 提案手法

我々が提案するプログラミングスキル評価モデルを図1に示す。

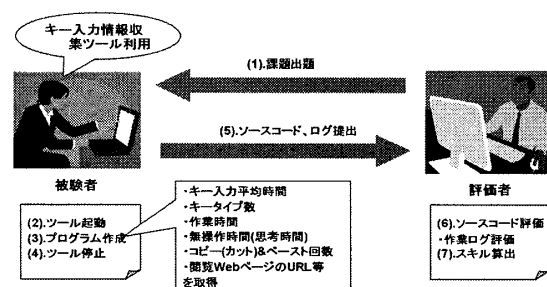


図1. プログラミングスキル評価手法モデル

A study on evaluation methods of programming skill level

†Taku KUCHIKI †Keizou YAMADA †Jun SASAKI

†Faculty of Software and Information Science, Iwate Prefectural University

*1 各エラーが作成されてから除去されるまでの時間

*2 1個のバグを発見してから修正するまでの時間的効率

*3 ソースコード中の分岐数+1

*4 オペランドとオペレータの種類数と出現数から求める

図 1 のモデルについて説明する。(1) 評価者が被験者にプログラミングの課題を提示する。被験者は、(2) キー入力情報収集ツールを起動し、(3) プログラムを作成する。このとき、全機能を実装するまでデバッグも行うこととする。完成するまでデバッグを行い完成した時点で、(5) ソースコードと作業ログを提出する。(6) 評価者は、ソースコードと作業ログを評価し、(7) プログラミングスキルレベルを算出する。

今回、作業時間、キータイプ数、コピー (カット) & ペースト回数、無操作時間、閲覧した Web ページの URL とタイトルを取得する。作業時間はプログラム初期作成時間とデバッグ時間を取得する。キータイプ数、コピー (カット) & ペースト回数は、スキルレベルとの相関関係を調査するために取得する。無操作時間は、思考時間と仮定し、短いほど思考力が高いまたは思考しながら作業を行うことができ、能力が高いと仮定した。そこで、スキルレベルとの相関関係を調査するために取得する。閲覧した Web ページの URL とタイトルも取得し、プログラム作成の際に、どのような調査を行っているかを調べる。

4 評価ツールの開発

上記提案に基づき効率的にプログラミングスキルを評価するために、以下の機能を有するツールを開発した。以下にツールの機能を記す。

- (1) キー入力情報記録機能
キーボードから入力したキーをすべて記録する。
- (2) キー入力数取得機能
記録したキー入力数を取得する。入力数を記録するキーは、全キー入力数、バックスペースキー入力数、エンターキー入力数、アルファベット+数字キー入力数、方向キー入力数である。
- (3) コピー (カット) & ペースト検出機能
コピー(カット)&ペーストを行った際の文字列と、コピー&カット回数、ペースト回数を記録する。
- (4) 作業時間計測機能
プログラミング作業とデバッグ作業の時間を取得する。
- (5) 無操作時間 (思考時間) 計測機能
キーボード入力及びマウス操作を行っていない時間を計測する。
- (6) 閲覧 Web ページ URL 取得機能
閲覧した Web ページの URL とページタイトル、閲覧 Web ページ数を取得する。対応 Web ブラ

ウザは Internet Explorer 8 及び Google Chrome とした。

5 評価実験

5.1 実験方法

提案手法に基づき、手法の有効性を確認するための評価実験を行った。実験は、被験者 (学生 8 名) に簡単な要求仕様提案による問題を出題し、プログラムを作成させる (プログラミング言語は PHP とする)。なお被験者のスキルレベルは、PHP のコードを読むことができ、小規模 Web アプリケーションを構築できる程度である。

5.2 実験計画

本研究の提案手法を評価するために 2010 年 1 月以降、岩手県立大学ソフトウェア乗法学部の学生の協力を得て実験を行う予定である。

6 まとめ

本研究では、既存研究で考察されていない、初期段階からの評価、ロジック構成能力などを考察した新しいプログラミングスキル評価手法を提案し、キー入力情報収集ツールの開発を行った。また、本ツールを用いた実験評価を行い、提案手法の有効性を確認した。今後は、プログラミングスキル評価手法及び評価ツールの改良を行い、評価の自動化に向けた研究を行う予定である。

参考文献

- 1) 松本 健一, 井上 克郎, 菊野 亨, 鳥居 宏次: エラー寿命に基づくプログラマ性能の実験的評価 - 大学環境におけるプログラム開発-, 電子情報通信学会論文誌 D-I Vol. J71-D No.10 pp.1959-1965 (1988).
- 2) 高田 義広, 鳥居 宏次: プログラマのデバッグ能力をキーストロークから測定する方法, 電子情報通信学会論文誌 D-I Vol. J77-D-I No.9 pp.646-655 (1994).
- 3) T. McCabe. A software complexity measure. IEEE Transactions on Software Engineering, SE-2:308-320 (1976).
- 4) Halstead, M. H. : Elements of Software Science. Elsevier North-Holland, New York (1977).