

実プログラミングとプロセス観察による短期技術者診断プログラムの試案

松澤 芳昭[†] 大岩 元[‡] 酒井 三四郎[†]静岡大学 情報学部[†] 慶應義塾大学 環境情報学部[‡]

概要

IT 業界でソフトウェア技術者の質量不足が指摘されるようになって久しいが、このうち質の問題に関しては、現在でも技術者の能力差がどの程度であるかさえも把握されていない。そこで、我々は、短時間で小さな仕事を行って技術者の基本的な技術力（問題の定義、見積もり、解法の設計と実装、および品質評価）を診断するためのプログラム試案を行った。このプログラムでは、PSP(Personal Software Process)とペアプログラミングによって成果とプロセスを診断する。実際の技術者（経験 1 年～15 年）に対して予備実験を行った結果、4～8 時間程度で技術者の能力診断が実現できるという感触が得られた。このプログラムによって技術者の生産性等が把握できれば、効果的な技術者教育への誘導を行うことができ、ソフトウェア開発の精緻な見積もりも可能になる。

1. はじめに

プログラムの能力差については経験的によく知られている。あるプログラムが一定時間で終わられる作業について、多大な時間を割いても、動作しない、または品質を保証できないプログラムを作成するプログラムもいる。

この生産性の差は、主としてプロセスの違いに起因する。プログラムの“条件分岐”さえ使いこなすことができれば、余程複雑で高精度が要求される問題でない限り、暗中模索的な試行錯誤のプロセスによっても、時間の制約さえなければプログラムを完成できる可能性がある。

プログラムはこれまで職人気質の職場であったため、そのプロセスが覗かれることは拒まれてきた。生産性が少々低くても、結果として正常に動作するプログラムが作成されるのであれば、かつ残業することによってカバーできる範囲であれば、問題が表出されないようにされてきた。結果として、プログラムの能力差は長年問題として認識はされてきたものの、現在でも実際にそれがどの程度であり、何に起因するものなのかが把握されていない。

近年では、ソフトウェア工学の研究成果に基づく開発ツールやフレームワークの利用によって、ソフトウェア開発における、いわゆる“コーディング”のプロセスの負担は軽減されている。しかしながら、問題を情報システムの仕様として落としこみ、実装したプログラムの結果を評価する過程は依然技術者の仕事であり、その工程全体を“プログラミング”と考えるのであれば、

そのプロセスと生産性が計測できることによって、効果的な技術者教育への誘導を行うことができ、ソフトウェア開発の精緻な見積もりも可能になるはずである。

こうした背景から、本研究では、コーディングだけでなく、ソフトウェア開発の基本的な技術力（問題の定義、見積もり、解法の設計と実装、および品質評価）を実際の作業とプロセス観察から診断するプログラムの開発を行う。

2. 診断プログラムの設計方針

2.1 問題の難易度

診断プログラムは実際の企業で実用可能なように設計する。従って、診断に使える時間は多くて 1 日（8 時間）程度である。この制約の中で診断できるよう、問題を 2 つ用意した。

[問題 A]は吉田[1]で紹介されている、三角形の種類識別問題（原典はマイヤーの著名なテスト技法の一節）である。これを仕様策定、アルゴリズム、テストを含む最小の問題と考えた。

[問題 B]は中級者用の問題として設計し、改行区切りのデータファイルを読み込み、その移動平均を求めてファイルに書き出すプログラムで、配列、ファイル操作の要素が含まれている。

2.2 プロセスの追跡

プロセスの記録については、ハンフリーによる PSP(Personal Software Process)[2]を利用する。しかしながら、PSP におけるデータ収集は作業へのオーバーヘッドがかかる。プログラミングによる問題解決は高度な知的作業であり、診断時の作業は、問題解決以外の思考要素は排除すべきである。

この解決案として、自動記録と自己回顧による分析が望ましいと考えているが、その装置がない当面、ペアプログラミングと組み合わせ、被験者と観察者のペアで作業の記録を行う。

A Proposal of Short-Term Skill Check Program for IT Engineers by Pair-Programming and Peer-Observation of Their Engineering Process

[†]Yoshiaki Matsuzawa, Shizuoka University

[‡]Hajime Ohiwa, Keio University

[†]Sanshiro Sakai, Shizuoka University

3. 予備実験(1)の結果

問題の難易度に関する適切性を調査するために、企業の技術者（異なる企業所属の経験は1年～15年）を被験者として、C言語を使った問題回答の時間を計測した（なお、この実験ではペアプログラミング、PSPを行う方式では実施しておらず、作業時間は自己申告による）。問題は被験者自らに選択させ、13名中課題Aを選択した被験者が4名、課題Bが9名であった。得られた結果を表1にまとめる。（各問題ごとに工数順に整列しており、LOCはコメント・空行を除いたプログラム行数である）

表1. 予備実験(1)の結果

被験者	問題	工数(h)	規模(LOC)
A	A	4	175
B	A	6	144
C	A	8	134
D	A	50	164
E	B	2	188
F	B	3	135
G	B	4	101
H	B	5	246
I	B	7	149
J	B	9	342
K	B	9.5	450
L	B	10	216
M	B	28	613

工数は、診断プログラムの目標である4時間～8時間程度に概ね収まっている。特徴的なのは課題A、Bともに大幅に時間がかかっている被験者が1名ずついる点である（被験者D、M）。規模の差はそれほどでもないことから、これらの被験者は何らかの大きなプロセス問題があることを示している。用意した問題によって、こうした大幅なプロセス改善が必要な被験者を割り出すことが出来ると考えられる。

4. 予備実験(2)の結果

PSPとペアプログラミングによるプロセス観察の効果を調査するために、企業の1年目（被験者N）と7年目（被験者0）の技術者に対してそれぞれ問題A、問題BをJava言語による実装で実施した。

実験はこの2人一組によって、被験者と観察を交代して行った。なお、この実験においては、実験時間の制約があったため、問題を解き終わらなくても時間で区切って途中終了としている。

被験者Nの結果を表2に示す。規模は途中終了時点で60LOCであった。プロセスは、設計が

ほとんど行われずに実装に入り、問題の解法を逐一Webで調べて、実装というサイクルを繰り返している（プログラミングの文法については調べていない）。テスト段階に入り仕様の間違いに気づき、仕様策定まで戻る等、各工程が入り乱れたプロセスになっている。

表2. 被験者Nの結果

工程	計画値(分)	実績値(分)
計画立案	-	6
設計	20	9
コーディング	50	60
テスト	20	30(途中終了)
事後分析	30	60

次に、被験者0の結果を表3に示す。規模は途中終了時点で66LOCであった。プロセスは、設計に工数を割り問題の定義を行っている。しかし、見積もり時間を大幅に超過している。コーディングでは、対象言語(Java)がしばらく業務で利用していないことを理由に、ファイル入出力、配列の言語仕様などをWebで調べながら実装しており、途中終了となっている。

表3. 被験者0の結果

工程	計画値(分)	実績値(分)
計画立案	-	5
設計	15	44
コーディング	60	56(途中終了)
テスト	40	-
事後分析	-	60

5. 総評・今後の課題

予備実験の結果、初級～中級者に対しては目標とした時間内に技術者の能力診断ができる感触が得られた。予備実験(2)の被験者が作業に集中できたことから、PSPとペアプログラミングにより、記録のオーバーヘッドがなく診断プログラムが実現できることも分かった。

本稿の実験は被験者も少なく、成果物の質を考慮した分析も行っていない。記録・分析の方法も文献[3]等、自動記録やプロトコル分析などの研究成果を参考に洗練させていく必要もある。

参考文献

- [1]吉田征, 技術の伝承と移転—芸術と技術と人間との接点からの発想, 日科技連出版社(1994).
- [2]Watts S. Humphrey: 「パーソナルソフトウェアプロセス入門」, 共立出版(2001).
- [3]前田恵三, 中野靖夫: 「プログラム作成過程の分析」, 日本教育工学雑誌, Vol. 19, No. 3, (1995).