

初級開発者のソフトウェアプロセス改善支援 ～ソフトウェア開発の状態遷移モデルに基づいたプロセス評価～

3P-01

若林 一博 福原 綾介 野中 誠 東 基衛

早稲田大学大学院 理工学研究科 経営システム工学専門分野

1. はじめに

ソフトウェア開発を効果的に行うための手段の一つとして、開発プロセスを改善することが重要であるという認識が高まりつつある。近年では、**PSP (Personal Software Process) [1]** などソフトウェアプロセスに関する研究が盛んに行われている。

本研究では、**PSP0** に従った初級開発者の開発プロセスに着目し、学習を考慮した開発プロセスの状態遷移モデルを提案する。さらに、そのモデルにおける状態遷移の様子を分析し、開発プロセスの評価を行う。

2. 開発プロセスの状態遷移モデル

福原らのペトリネットを用いた行動モデル [2] に改良を加えた状態遷移モデルを図 1 に示す。

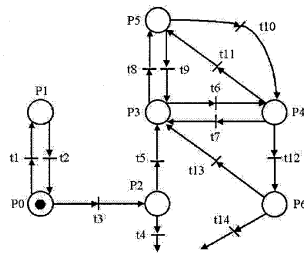


図 1: 開発プロセスの状態遷移モデル

P0:コードを書いている状態
P1:コーディング作業中に学習している状態
P2: 除去作業前のコンパイルまたはテストが終了した状態
P3:フォールトを推定している状態
P4:フォールトを除去している状態
P5:学習している状態
P6:除去作業後のコンパイルまたはテストが終了した状態
t1:コーディング作業中に疑問発生
t2:コーディング作業開始 (学習効果あり)
t3:コンパイルまたはテスト開始
t4:フォールトなし
t5:フォールト発生
t6:フォールト除去作業開始 (学習効果なし)
t7:フォールト推定し直し (除去の仕方不明)
t8:推定中に疑問発生
t9:推定し直し
t10:フォールト除去作業開始 (学習効果あり)
t11:除去中に疑問発生
t12:除去作業終了及びコンパイルまたはテスト中
t13:フォールト推定し直し (除去失敗)
t14:除去成功

Supporting Self-Improvement of Software Process for a Novice Developer - A Process Assessment based on the Software Development State-Transition Model - : Kazuhiro Wakabayashi, Ryosuke Fukuhara, Makoto Nonaka, Motoe Azuma, Graduate School of Sci. & Eng., Waseda Univ.

本研究では図 1 のモデルにおける **t13** の遷移を手戻り作業とみなす。すなわち、コンパイル工程における文法フォールトの除去失敗、テスト工程における論理フォールトの除去失敗を手戻り作業とする。

3. 初級開発者のプロセス測定実験

3.1 実験方法

我々は、初級開発者のプロセスにおけるデータを測定するためのプログラムエディタを開発した。本研究では、このプログラムエディタに状態遷移と滞留時間、欠陥の記録などの機能拡張を行ったものを用いて実験を行う。プログラミングエディタの概要を以下に示す。

- ・エディタ部 (コーディングの状態時に表示される。)
- ・コンパイル部 (コンパイルボタンを押すことでコンパイルが行われ、エラーメッセージが表示される。)
- ・API 参照部 (学習の状態時に画面に表示される。)
- ・特定欠陥記録部 (推定の状態時に、エラーメッセージとコードを見ることで、欠陥の特定・解決方法の検討を行う。)
- ・状態遷移記録部 (図 1 のモデルにおける状態遷移をボタンにより行う。)
- ・滞留時間記録部 (ボタンが押された時刻とその状態名を CSV 形式で保存する。)

3.2 実験手順

実験手順を以下に示す。

- ① 6 人の初級開発者 (A～F) に対し、**PSP** の課題 (Program1A～3A) のどれか 1 つを **PSP0** のプロセスに従い行わせる。
- ② 各状態の遷移は、プログラムエディタ上の画面の切り替えボタンを押すことにより行う。
- ③ コンパイルやテスト工程で欠陥が発生した場合、どの欠陥を除去の対象とするのか 1 つだけ特定してもらう。対象とした欠陥が除去されるまでの状態遷移の様子と滞留時間の計測を行う。

3.3 測定データ

実験において測定するデータを以下に示す。

- 1) 各状態における滞留時間
- 2) 各状態の遷移
- 3) 欠陥の型（文法もしくは論理フォールト）

これらのデータより、以下の尺度を算出する。

手戻り回数/欠陥数（＝手戻り率 **BR**）：欠陥（文法フォールトと論理フォールト）の除去時に発生した単位手戻り数。 **BR** は **CBR** と **TBR** の平均値である。

コンパイル工程における手戻り率（**CBR**）：文法フォールトの除去時に発生した単位手戻り数。

テスト工程における手戻り率（**TBR**）：論理フォールトの除去時に発生した単位手戻り数。

4. 実験結果と考察

実験結果を表 1 に示す。

表 1: 実験結果

	A	B	C	D	E	F	LOC/min との 相関係数 r
LOC/min	0.147	0.199	0.336	0.699	0.227	0.512	
BR	0.929	0.625	1.000	0.667	1.083	1.167	-0.112
CBR	0.714	0.500	0.333	0.176	0.889	0.333	-0.803
TBR	0.333	0.010	2.500	2.750	1.667	2.000	0.783

- 1) 手戻り率と生産性について

表 1 より、**CBR** が、生産性と強い負の相関を示している。初級開発者の場合コンパイル工程における手戻り率が高く、これが生産性を大きく低下させている要因であると考えられる。続いて、各手戻り率と状態の滞留時間について分析する。

- 2) 手戻り率と滞留時間について

各手戻り率と滞留時間の相関係数を表 2 に示す。

表 2: 手戻り率と滞留時間との相関係数

状態	BR	CBR	TBR
P0: コーディング	0.371	-0.297	0.196
P1: 学習 (コーディング)	0.579	0.913	-0.332
P2: コンパイル	-0.643	0.465	-0.661
P3: 推定 (コンパイル)	-0.873	-0.446	-0.022
P5: 学習 (コンパイル)	0.145	0.817	-0.645
P4: 除去 (コンパイル)	-0.265	0.731	-0.462
P2: テスト	-0.561	-0.645	0.332
P3: 推定 (テスト)	0.043	-0.772	0.806
P5: 学習 (テスト)	0.329	0.684	-0.349
P4: 除去 (テスト)	0.023	-0.360	0.547

表 2 より、コーディング工程の学習時間と **CBR** が $r=0.913$ と強い相関を示している。また、コーディング工程における学習状態の滞留時間と発生回数に関して相関係数を求めたところ、 $r=0.908$ の強い相関が見られ

た。このことから、初級開発者がコーディング工程において学習する場合、学習状態の滞留時間が多いほど、学習状態の発生回数も多いことがわかる。**CBR** と学習（コーディング）の散布図を図 2 に示す。

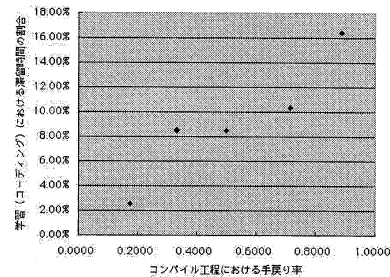


図 2: **CBR** と学習（コーディング）の散布図

以上より、コーディング工程において多く学習を行う初級開発者ほど、文法フォールトの除去における手戻り作業が発生しやすいという傾向が抽出できた。

5. まとめ

本研究における実験により、初級開発者の開発プロセスにおいては、コンパイルプロセスにおける手戻り作業が生産性を低下させる大きな要因であるという結論が得られた。また、その手戻り作業は、コーディング工程における学習に多く時間を費やしている被験者ほど高い確率で発生している。

また、コーディング工程における学習時間が長い被験者は、学習の発生回数も多いことがわかった。このことから、初級開発者はコーディング工程において非効率な学習を多く行っており、その結果コンパイル工程において手戻り作業を多く発生させている。これが生産性の低下につながっていると考えられる。

6. 今後の課題

初級開発者のプロセス改善支援の第一段階として、学習を考慮した開発プロセスにおける状態遷移モデルについての分析を行った。今後は、測定した尺度からどのようにプロセス改善を支援するのか、検討していく必要がある。

[参考文献]

- [1] Humphrey, W.S., "A Discipline for Software Engineering", Addison-Wesley, 1995
- [2] 福原綾介他, "ソフトウェア開発システムにおける人的資源の質の改善について—初級プログラマの作業中における学習と知識のモデル化—", 情報処理学会第 58 回全国大会, 1999