

初級開発者のソフトウェアプロセス改善支援

3P-02

— 開発中の学習を考慮した開発作業時間見積もり手法の提案 —

福原綾介 若林一博 松野恭士 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

ツールや方法論の目覚ましい発達にも関わらず、ソフトウェア開発は依然として個人の能力に大きく依存している。ソフトウェア開発者が、自己のソフトウェア開発プロセスを改善する教則として PSP (Personal Software Process) があるが、開発中の学習については細かく触れていない[1]。しかし、常に新しい技術を学び続けることが開発者に要求される今日、開発中の学習を考慮したプロセス改善が重要である。

本研究では、初級開発者の自己のプロセス改善活動において利用可能な、開発中の学習を考慮に入れた開発作業時間の見積もり手法を提案する。

2. プロセスモデル

2.1. 学習プロセスの定義

本研究では、従来の開発プロセス (設計・コーディング・コンパイル・テストのような工程を含む。以下、**作業プロセス**) に加えて、**学習プロセス**を定義する。学習プロセスとは、ソフトウェア開発中に、開発者が知識を獲得するために行う活動である。例えば、開発中にプログラミング言語の仕様について調べたり、他のコードを参照したりすることである。ソフトウェアの総開発作業時間を TD、作業プロセスの時間 (以下、**作業時間**) の合計を TW、学習プロセスの時間 (以下、**学習時間**) の合計を TL とすると、TD は、

$$TD = TW + TL$$

と表すことができる。ソフトウェア開発作業は、要素作業 (例えば欠陥を1つ除去するなど) の集合である。各要素作業の所要時間は、作業時間と学習時間から成る。要素作業数を A とすると、TD は、

$$TD = \sum_{i=1}^A d_i, \quad d_i = w_i + l_i$$

d_i : i 番目の要素作業の所要時間

w_i : i 番目の要素作業における作業時間

l_i : i 番目の要素作業における学習時間

と表すことができる。Ds:設計、Cd:コーディング、Cm:コンパイル、Ts:テストとし、 W_p, PL_p をそれぞれ工程 p に含まれる要素作業の w_i, l_i の合計とすると、TW、TL はそれぞれ、

$$TW = W_{Ds} + W_{Cd} + W_{Cm} + W_{Ts}$$

$$TL = PL_{Ds} + PL_{Cd} + PL_{Cm} + PL_{Ts}$$

と表すことができる。

2.2. 学習プロセスの分類

知識とは、作業者が記憶すべきソフトウェア開発に必要な情報であり、ソフトウェア工学の基礎知識、技法 (方法論) の知識、システム開発ツール及び環境の知識、アプリケーション対象分野の知識、システム資源の知識の5つに分類される[2]。学習プロセスについても、そこで獲得する知識によって、これらの分類が可能である。東らの分類順に B:基礎知識、M:方法論、T:ツール、D:対象領域、R:資源とし、分類 c に属する学習プロセスの時間 l_i の合計値を CL_c とすると、TL は、

$$TL = CL_B + CL_M + CL_T + CL_D + CL_R$$

と表すことができる。

学習プロセスは発生した工程および獲得知識の分類の2つの視点から分けることが可能であり、工程 p で発生し、かつ分類 c に属する学習プロセスの l_i の合計を L_{pc} とすると、 PL_p, CL_c はそれぞれ、

$$PL_p = \sum_c L_{pc}, \quad CL_c = \sum_p L_{pc}$$

と表すことができる。

2.3. 学習プロセスの発生と学習時間

学習プロセスは、属する工程・分類 (2.2) によって、発生する数や時間が異なる。ここで、

$$L_{pc} = A \times LA_{pc} \quad (A \text{ は要素作業数})$$

の関係を作り立たせる LA_{pc} を定義する。学習プロセスが要素作業毎にある確率に従って発生し時間が決定されると考えると、 LA_{pc} は1要素作業あたりに発生する工程 p・分類 c の学習時間の期待値である。 LA_{pc} は開発者の、工程 p で適用する分類 c の知識が不足しているほど、大きくなると考えられる。

3. 提案する手法

3.1. 学習プロセスの計測

以下の手順で自己の学習プロセスの計測を行なう。

- ① 各プロジェクト毎に発生した学習プロセスについて、発生した工程 (p)、分類 (c)、学習時間 (li)、を計測し TL、PL_p、CL_c、L_{pc} をそれぞれ算出する。
- ② 各プロジェクト毎に LA_{pc} の代理値として、LK_{pc} を算出する。LK_{pc} は、A の代わりにプログラム規模 S を用いて算出した値である。
- ③ 全プロジェクトの LK_{pc} の累積値を算出し、分類別・工程毎の割合 LKR_{pc} を算出する。
- ④ 各プロジェクト毎に LK_{pc} の分類毎の合計値 JLK_c を算出する。

3.2. 学習時間の見積もり

計測した過去の学習プロセスデータから、以下の手順で新規プロジェクトにおける L_{pc} を見積もる。

- ① JLK_c を見積もる。分類毎の不足知識量の大きさ (2.2) の予想と、過去のプロジェクトの JLK_c を参考に見積もり値を定める。
- ② JLK_c の見積もり値に、前節で求めた LKR_{pc} を掛け LK_{pc} を見積もる。
- ③ 規模の見積もり値 S を、LK_{pc} の見積もり値に掛け、L_{pc} を見積もる。

4. 提案手法の評価

4.1. 適用実験の概要

手法を評価するため、初級開発者を対象にソフトウェア開発実験を行なった。プログラミング言語は JAVA を用い、学習プロセスを計測するプロジェクトとして、PSP のプログラミング課題 1A~4A、学習時間を見積もるプロジェクトとして同 5A を用いた。

4.2. 実験結果

L_{pc} の見積もり値と実測値を表 1、表 2 に示す。

表 1: L_{pc} 見積もり値(min.)

L _{pc}	c	B	M	T	D	R	合計 (PL _p)
		p	Ds	Cd	Cm	Ts	
L _{pc}	p	0.0	10.7	0.5	1.9	0.0	13.0
	Ds	0.0	29.1	6.8	1.8	0.0	37.7
	Cd	0.0	6.0	2.6	0.0	0.0	8.6
	Cm	0.0	7.9	0.4	1.7	0.0	10.0
合計 (CL _c)	計	0.0	53.7	10.2	5.3	0.0	69.3

表 2: L_{pc} 実測値(min.)

L _{pc}	c	B	M	T	D	R	合計 (PL _p)
		p	Ds	Cd	Cm	Ts	
L _{pc}	p	0.0	0.0	6.0	30.4	0.0	36.4
	Ds	0.0	22.0	3.0	55.1	0.0	80.1
	Cd	0.0	0.0	0.0	0.5	0.0	0.5
	Cm	0.0	0.0	0.0	12.0	0.0	12.0
合計 (CL _c)	計	0.0	22.0	9.0	98.0	0.0	129.0

4.3. 考察

実験では、規模の見積もり値 (0.147KLOC) が実測値 (0.297KLOC) より小さかった。規模を正しく見積もることができていれば、L_{pc} の見積もり値は実測値により近づく。

LK_{pc} の見積もり値と実測値をそれぞれ比較したところ、工程毎の合計値には大きな差がみられなかった (図 1) が、分類毎の合計値 (JLK) には、大きな差がみられた (図 2)。

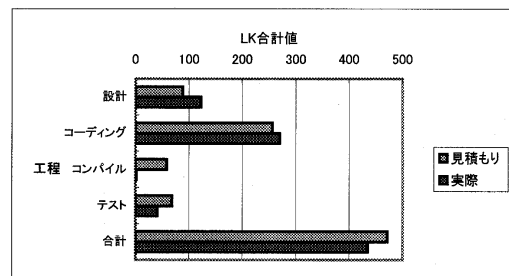


図 1: 工程毎の LK 合計値

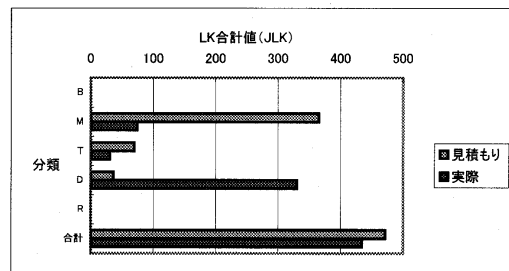


図 2: 分類毎の LK 合計値 (JLK)

このことから、LKR を用いて L を工程毎に見積もることは有効であると言える。JLK については、開発者が正しく見積もるスキルを身につけることができれば、正しい学習時間を見積もることが可能であると期待できる。

5. 今後の課題

手法の有効性を統計学的に示すために、今後更なるデータの収集を行い、検証するべきである。また、今回対象外とした要素作業数や不足知識量の計測方法、および学習プロセス発生を決定する確率分布 (2.3) について、研究を進める必要がある。

参考文献

- [1] W. S. Humphrey: "A Discipline for Software Engineering", Addison Wesley, 1995
- [2] 東基衛ほか: "情報技術者自己診断システム—能力モデルと診断の枠組み", 情報処理学会第 56 回全国大会, 1998