

# ソフトウェア開発中に発生する調査時間の見積り手法

3Z-09

松下壮一 菅野亨太 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

開発者の能力といった人的要因は、ソフトウェア開発工数大きな影響を与える。COCOMO II [1]では、プログラマの能力が開発工数に与える影響を最大で1.85倍の差があるとしている。開発工数を高い精度で見積もるためには、このような影響度の大きい要因を定量的かつ客観的に評価することが必要である。

プログラマの能力評価は主観的に行われることが多いが、開発者個人の開発中の調査・学習時間を分析することにより[2]、客観的な評価が期待される。また、その評価に用いるデータは、容易に計測可能なものであることが望ましい。

本発表では、以上を考慮したソフトウェア開発中に発生する調査時間の見積り手法を提案する。

## 2 研究のアプローチ

本研究では開発に必要な知識量と開発者の知識量との差異が調査時間を発生させると考え、プロダクトメトリクスを用いて調査時間を客観的に推定する手法を提案する。開発者の知識量、および開発対象の持つ真の知識量の代用としてプロダクトに含まれるAPI (Application Programming Interface) メソッドの数を用いる。APIはソフトウェアを開発する際に使用できる命令や関数の集合であり、機能ごとに分類されている。またプログラムの種類によっても含まれるメソッドは異なる[3]。

本研究ではAPIメソッドの使用経験が、その使用法ばかりでなくアルゴリズムや関連事項の知識も同時に獲得できると考え、知識量の代用として用いる。また、APIメソッドはプロダクトから容易に計測することが可能であり、計測の簡易化につながると考えられる。

## 3 提案する手法

### 3.1 提案手法の概要

提案する手法の概要を図1に示す。開発対象の見積り規模と機能による分類から、開発対象に含まれる知識量を推定する。一方、開発者の過去の作業履歴から開発者の知識量を推定する。二つの知識量から不足知識量を求め、回帰分析により調査時間を見積もる。各部について以下に詳しく説明する。

積もる。各部について以下に詳しく説明する。

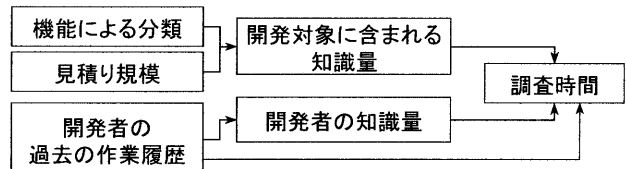


図1:提案手法の概要

機能による分類とは、開発対象プログラムの有する機能による分類であり、仕様より決定する。見積り規模はFunction Point法などにより測定した値をLOC (Lines Of Code) に変換したものとする。

開発者の作業履歴は、過去に開発したプログラムに使用されたAPIメソッドの種類と、過去の開発に要した調査時間である。なお調査時間はコーディング以降を計測対象とする。

### 3.2 開発対象の知識量の推定

開発対象に含まれる知識量は、発生する頻度の高いAPIメソッドの種類で表す。まず既存のプログラムを機能により分類し、各分類でのプログラム中のAPIメソッドの出現回数を事前に求めておく。これをプログラムの規模LOCで割った値の平均をAPIメソッド固有の出現スコアとする。

推定では、開発対象の仕様書からカテゴリを決定し、当該カテゴリでのAPIメソッドの出現スコアに見積りLOCを掛け、1回以上出現したAPIメソッドを発生する頻度の高いAPIメソッドとして開発対象に含まれるAPIメソッドとする。

### 3.3 開発者の知識量の推定

開発者の知識量は、過去の開発で使用したAPIメソッドの種類である。なお、過去の開発経験、もしくはその記録がない場合は、事前に既知のAPIメソッドを詳しくインタビューする必要がある。

### 3.4 調査時間の見積もり

以上により得られた開発対象に含まれる知識量と開発者の知識量の差分が不足APIメソッドとなる。過去の開発の調査時間と不足APIメソッド数とを回帰分析することにより、次の開発の調査時間を見積もる。回帰直線は開発が行われるたびに再計算され補正されるものとする。

## 4 調査時間の見積り実験

### 4.1 実験の準備と内容

今回は、開発に必要な知識としての API メソッドを推定せずに、毎回の開発で不足 API メソッドを計測し、不足知識と調査時間との関係を調べた。

被験者 2 名 (A, B) が作成した 5 個のプログラムセット (計 10 個) を用いて、開発中の調査時間と不足 API メソッドの関係を調べた。

実験に用いた課題は Personal Software Process[4]により提供される課題セット 1A~5A である。使用した言語は Java 言語である。また、被験者はプログラム経験が 1 年以上あり、基本的なプログラミング能力は備わっている。

### 4.2 実験の結果

被験者 A が各開発で使用した API メソッドとその使用回数を表 1 にまとめた。この中で、あるプロダクトにおいて、その開発以前に使用されていない API メソッドを不足メソッドとして網掛けしてある。不足 API メソッド数と調査時間をまとめたのが表 2 (被験者 A のみ) である。図 2 は両者の相関関係を示した散布図であり、回帰直線を引いた。

なお表 1 において、1A 開発時には不足 API メソッドは、使用回数が一回以上あるメソッド 13 個であるはずだが、事前のアンケートにより使用経験のある API メソッドを引くと 5 個になる。

表 1: 使用された API メソッド (被験者 A)

メソッド	1A	2A	3A	4A	5A
String	6	4	16	9	2
Double	3	0	0	5	1
FileReader	1	1	1	1	0
BufferedReader	1	1	1	1	0
Vector	0	0	9	0	0
Enumeration	0	0	3	0	0
System.out.println	6	5	10	17	6
System.exit()	1	1	1	1	1
String.length()	1	2	2	1	1
String.trim()	0	1	1	1	0
String.indexOf()	0	0	16	1	0
String.startsWith	0	1	1	0	0
String.endsWith()	0	2	2	0	0
String.substring()	0	0	1	2	0
Double.doubleValue	5	0	0	7	1
BufferedReader.read	1	3	2	1	0
BufferedReader.close	1	1	1	1	0
Vector.addElement	0	0	3	0	0
Vector.elements()	0	0	3	0	0
Vector.size()	0	0	1	0	0
Enumeration.hasMore	0	0	4	0	0
Enumeration.nextElement	0	0	4	0	0
Math.sqrt()	1	0	0	1	0
Math.exp()	0	0	0	0	1
Math.abs()	0	0	0	0	1
Object.equals()	3	0	0	4	4
Object.toString()	3	0	0	3	0

表 2: 調査時間と不足メソッド (被験者 A)

	1A	2A	3A	4A	5A
調査時間 (min)	44	10	38	5	14
不足 API メソッド	5	3	9	0	2

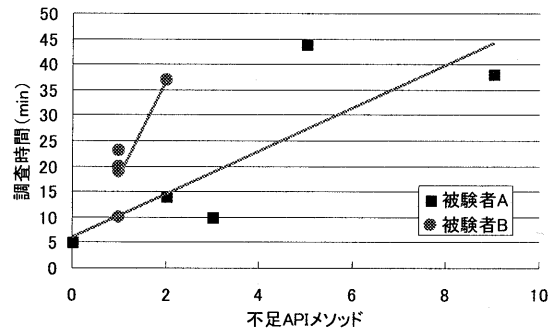


図 2: 不足メソッドと調査時間の散布図

### 4.3 考察

図 2 で得られた回帰直線の寄与率は被験者 A, B でそれぞれ 0.68, 0.75 と高い。図を見ると被験者 B は不足 API がどの開発でも少ないが、被験者 A と比べると、不足 API 数が多くなるほど調査時間が増大する傾向がある。

被験者が 2 名であり、しかも開発プロダクト数が 5 つと少ないが、本実験により不足 API メソッド数から調査時間を見積もることができることがわかった。

## 5 おわりに

本研究では、ソフトウェア開発に必要な知識量と開発者の知識量を API メソッドで表現し、両知識量の差から開発中に発生する調査時間を見積もる手法を提案した。また開発実験を行い提案した手法を検証した。その結果、寄与率の高い回帰直線が得られた。しかし、実験が小規模であるために個人の特性などを詳しく調べることができなかった。

また、手法全体を通しての見積り実験を行っていないため、実際に手法を適用した場合にどれほどの誤差が生まれるのかわからない。プロダクト分類の実験とともに、今後更に実験と検証を行う必要がある。

### 参考文献

- [1]COCOMO II Model Definition Manual Version 1.4, University of Southern California (1997).
- [2] 福原綾介他：初級開発者のソフトウェアプロセス改善支援－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－, 情報処理学会第 60 回全国大会, 2000
- [3]松下壮一他：ソフトウェア開発プロセス設計・適応支援システム－開発に必要な知識項目のモデル化－, 情報処理学会第 61 回全国大会, 2000
- [4] W.S.Humphrey: "A Discipline for Software Engineering", Addison Wesley, 1995