

# 学生によるシステム開発プロジェクトにおける ユースケースポイント法を用いた見積もりと実績の評価

内藤 正樹<sup>†</sup> 菊池 純男<sup>†</sup> 駒谷 昇一<sup>†</sup> 田中 二郎<sup>†</sup>

<sup>†</sup>筑波大学大学院 コンピュータサイエンス専攻

## 1. はじめに

情報システムの開発プロジェクトにおいて、システムの規模や工数を見積もることが、プロジェクトのコストの予測やスケジュールの策定のために重要である。

近年、大学教育で PBL 手法による教育が注目されている。授業の一環として行われるプロジェクトについても計画的で効果的な学習を目指す上でシステムの開発規模の見積もりは重要である。

一般に実施されるプロジェクトに関しては、見積もりの手法は重要なノウハウであるとして具体的な内容が外部に公開されない。しかしながら、前述のように授業の一環として行われるプロジェクトについても、設計や開発に関する教育効果を計画するために、ステップ数等で表されるシステムの開発規模を見積もる需要が多いのではないかと考えた。

本稿では、学生によるシステム開発プロジェクトの実績データをもとに、新たなプロジェクトにおける開発規模の予測方法とその実証結果を述べる。

## 2. ユースケースポイント法による見積もり

システムの見積もりを行う手法としては、ファンクションポイント法 (FP 法) [1]が有名である。FP 法で見積もりを行うためには、ある程度の設計が進みシステムへの入出力などが明らかになっている必要がある。

これに対して、システムの要件が明らかになった段階で見積もりを行うことができる、ユースケースポイント法 (UCP 法) が提案されている [2]。UCP 法はシステムの利用ケースを表したユースケースを基に見積もりを行う。

UCP 法は FP 法に比べて早期に見積もることができ、教育現場で適用すれば早期に授業計画を見通すことが期待できる。

## 3. プロジェクトの概要とステップ数

対象としたプロジェクトについて表 1 に示す。これらは 2007 年度～2008 年度に実施された。ステップ数はシステムを構成するソースコードのコメントと空行を除いた有効行数である。ステップ数の計数には Project Amateras の StepCounter<sup>1</sup>を使用した。

表 1: プロジェクトの概要とステップ数

名称	開発対象	言語	人数	ステップ数
A	Web システム	PHP	5 名	2711
B	同上	同上	4 名	6232
C	同上	同上	3 名	7588
D	同上	同上	4 名	4183

## 4. UCP の算出

表 1 のそれぞれのプロジェクトについて、UCP を算出した結果を表 2 に示す。なお、UUCP は未調整 UCP、TFactor は Technical Factor (技術要因)、EFactor は Environmental Factor (環境要因)、UCP がユースケースポイントである。

UUCP は個々のアクタの複雑度と個々のユースケースの複雑度の和で表される。また、TFactor は内部処理の複雑度など 13 項目について 5 段階の評価と項目に対応する重みの積和をとったものである。EFactor はオブジェクト指向技術の経験など 8 項目について 5 段階の評価と項目に対応する重みの積和をとったものである。

TFactor および EFactor の各項目の評価はプロジェクトの成果物やインタビューを参考に決定した。

表 2: UCP の算出

名称	UUCP	TFactor	EFactor	UCP
A	163	28.5	-1	206.3
B	348	28.5	-1	440.4
C	608	26.5	-1	752.1
D	250	23.5	-1	298.5

The estimation of students' projects and evaluation with use case point method

Masaki Naito<sup>†</sup>, Sumio Kikuchi<sup>†</sup>, Shoichi Komaya<sup>†</sup> and Jiro Tanaka<sup>†</sup>

<sup>†</sup>Department of Computer Science, University of Tsukuba

<sup>1</sup> StepCounter – Project Amateras  
<<http://amateras.sourceforge.jp/cgi-bin/fswiki/wiki.cgi?page=StepCounter>>

なお、UCPは(1)式であらわされる。

$$UCP = UUCP \times TCF \times EF \quad (1)$$

ここで、TCFとEFは(2)式、(3)式で定義される。

$$TCF = 0.6 + (0.01 \times TFactor) \quad (2)$$

$$EF = 1.4 + (-0.03 \times EFactor) \quad (3)$$

## 5. UCP を基にしたステップ数の予測

表 1 と表 2 をもとにステップ数を予測する式を求めるところを試みた。

UCP とステップ数の相関を調べた結果、UCP とステップ数には相関がみられた (図 1)。

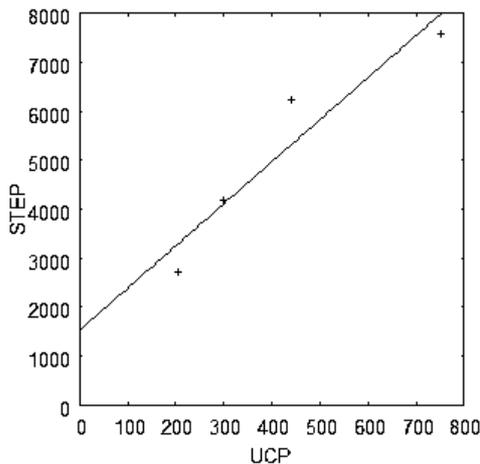


図 1 : UCP とステップ数の関係

そこで、独立変数を UCP、従属変数をステップ数として単回帰分析を行った。独立変数としている UCP は、それぞれ別プロジェクトのものであり無相関であると考えられる。

回帰分析の結果、(4)式であらわされる回帰直線を得た (図 1)。

$$Step = 8.59 \times UCP + 1532 \quad (4)$$

ここで、Stepはステップ数である。

次に回帰分析の分散分析を行った。この結果、有意水準 5%で有意であるといえた。また、従属変数の係数についても有意水準 5%で有意であるといえた。

## 6. 回帰式による予測の評価

プロジェクト A~D とは別のプロジェクト E および F で(4)式を評価した。両プロジェクトの概要を表 3 に示す。2009 年度に実施された PHP による Web システム開発プロジェクトである。

表 3 : 評価プロジェクト

名称	開発対象	言語	UCP	ステップ数
E	Web システム	PHP	314.1	4463
F	同上	同上	428.1	6060

プロジェクト E について、(5)式および(6)式で UCP から予測されるステップ数  $Step_E$  と、予測されたステップ数と実際のステップ数との誤差  $Err_E$  を求めた。

$$Step_E = 8.59 \times 314.1 + 1532 = 4230step \quad (5)$$

$$Err_E = \frac{4230 - 4463}{4463} \times 100 = -5.22\% \quad (6)$$

プロジェクト F についても同様に、

$$Step_F = 8.59 \times 428.1 + 1532 = 5209step \quad (7)$$

$$Err_F = \frac{5209 - 6060}{6060} \times 100 = -14.04\% \quad (8)$$

(7)式でステップ数を予測し、(8)式で予測との誤差を求めた。

この結果、プロジェクト E については約-5%の誤差、プロジェクト F については約-14%の誤差で UCP からステップ数が予測できた。これはプロジェクト初期の見積もりとしては十分な精度であると考えられる。

## 7. おわりに

学生によるプロジェクトで開発されるシステムのステップ数を UCP から予測する方法を示した。

いくつかのプロジェクトの実績から得られた予測式を他のプロジェクトに適用した結果、プロジェクト初期の見積もりとしては十分と考えられる精度でステップ数を予測できた。

本稿で述べた方法は、学生によるプロジェクトにおけるシステムの開発規模の早期予測に役立つと考えられる。

## 謝辞

本稿で対象としたプロジェクトは、筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻・高度 IT 人材育成のための実践的ソフトウェア開発専修プログラムで実施された PBL 型のシステム開発によるものである。本稿の執筆に当たりご協力をいただいたプロジェクト関係者の皆様に感謝申し上げます。

## 参考文献

- [1] A.J.Albrecht, "Measuring application development productivity", Proc. of IBM Applic. Dev. Joint SHARE/GUIDE Symposium, Monterey, CA, pp.83-92, 1979.
- [2] G. Karner, "Resource Estimation for Objectory Projects", "Objective Systems SF AB (Rational Software), 1993.
- [3] Toshihiro Kamiya, Shinji Kusumoto, and Katsuro Inoue, "CCFinder: AMulti-Linguistic Token-based Code Clone Detection System for Large Scale Source Code", IEEE Trans. Software Engineering, Vol.28, No.7, pp. 654-670, 2002.