

# 写像経路に着目したソフトウェア開発プロジェクトモデルを用いた工数—工期の理論モデル構築

榮谷 昭宏<sup>1,2</sup> 高田 眞吾<sup>3</sup> 前野 隆司<sup>1</sup>

受付日 2015年5月8日, 採録日 2015年11月5日

**概要:** 本論文では、工数と工期の経験モデルがなぜ累乗関数で表されるのか、そして、その乗数、指数がなぜ様々な値をとるのか、理論モデルを用いて明らかにし、理論的に工数と工期の関係式を導出することを目的とする。その理論モデルはプロジェクト内でやりとりされる設計情報の写像経路と、その写像の難しさを変数としたモデルを用いる。本論文の結果、乗数は写像経路数とその写像の難しさに依存した関数で規定され、指数は写像経路数だけに依存した関数で規定されることが分かった。そして、これらの関数をもとに理論的に算出した近似式と、実際のプロジェクトデータとを比較し、本理論モデルが実際のプロジェクトの工数と工期の関係を再現していることを実証検証した。以上により、理論モデルによって、工数と工期の経験式と理論式の整合性を確認し、そのメカニズムを明らかにした。

**キーワード:** 理論モデル, 工数, 工期, プロジェクトモデル, 写像経路

## Theoretical Formula for Man-hour and Development Time Based on Project Model Focused on Mapping Paths

AKIHIRO SAKAEDANI<sup>1,2</sup> SHINGO TAKADA<sup>3</sup> TAKASHI MAENO<sup>1</sup>

Received: May 8, 2015, Accepted: November 5, 2015

**Abstract:** This research develops a theoretical formula that describes the relationship between man-hour and development schedule. This study theoretically discusses why mathematical models in previous works have exponential relationships, and why their multiplier factors and exponents have various value. Our research theory is based on a project model determined by the number of mapping paths for design information, and difficulty of mapping. We show that the multiplier factor is defined by the number of mapping paths and the difficulty of mapping, and exponent is specified by the number of mapping paths. We applied our model to actual development data, and found that our model significantly correlates with actual data.

**Keywords:** theoretical formula, man-hour, development schedule, project model, mapping paths

### 1. はじめに

ソフトウェア開発プロジェクトの工数（プロジェクトの各作業に関わる要員数とその作業期間の積の総和（人

月）と工期（プロジェクトの開発開始から終了までの期間）の関係は、実証データをもとにした経験モデルとして、累乗関数（工期 =  $a \times$  工数 <sup>$b$</sup> ）で表せることが知られている。しかし、その累乗関数において、乗数  $a$ 、指数  $b$  は先行研究 [1], [2], [3], [4], [5], [6], [7], [8] によって様々である（表 1）。この様々となる理由は、環境差であるといわれているが [9], [10]、具体的にどのようなことが影響し、乗数・指数を決めるのか、その因果関係は明らかではない。COCOMO [1], [2] 等では、乗数、指数の決定要因をあげているが、諸説いろいろとあり [9]、明らかになったとは

<sup>1</sup> 慶應義塾大学大学院システムデザイン・マネジメント研究科  
Graduate School of System Design and Management, Keio University, Yokohama, Kanagawa 223-8526, Japan

<sup>2</sup> NTT コムウェア株式会社  
NTT COMWARE CORPORATION, Minato, Tokyo 108-8019, Japan

<sup>3</sup> 慶應義塾大学理工学部  
Faculty of Science and Technology, Keio University, Yokohama, Kanagawa 223-8522, Japan

表 1 各経験モデルの乗数・指数  
Table 1 Multiplier and exponent of models.

研究事例		乗数	指数
COCOMO I [1]	Organic	2.5	0.38
	Semi-detached	2.5	0.35
	Embedded	2.5	0.32
COCOMO II [2]		3.67-5.25	0.28-0.34
C. E. Walston, C. P. Felix [3]		2.47	0.35
D. Reifer[4]	Web-based electronic commerce	2.0	(0.32 for larger)
	Financial/trading application	2.2	
	Business-to-business application	1.5	
	Web-based portals	1.8	
	Web-based information utilities	2.0	
ソフトウェア開発データ白書	新規開発	—	0.32
2012-2013[5]	改良開発	—	0.31
ソフトウェアメトリクス報告 2011 年度版[6]		2.54	0.33
経済調査研究所研究レポート	新規開発	2.44	0.34
	改良開発	1.73	0.39
戸田, 松本,	成功プロジェクト事例	2.38	0.33
押野, 高橋[8]	失敗プロジェクト事例	2.94	0.27
	Java による成功事例	1.64	0.42

いい難い。一方、理論モデルとしては Putnam モデル [11] が有名であるが、プロジェクトの工数や工期に関連した研究は、ほかに Design Structure Matrix (DSM) [12] を応用した研究 [13], [14], [15], [16], [17], [18], [19], [20] や、System Dynamics [21] を応用した研究 [22], [23], [24], [25], [26], [27], [28], [29], [30], [31] 等がある。しかし、それらの理論モデルは経験則にマッチしない等の問題がある。

また、経験モデルを理論的に説明することができていない。つまり、理論モデルと経験モデルの間で、整合性がとれていない。この問題を解決するには、次の3つの課題を解決する必要がある。

課題1は、母集団データの分布を理論的に明らかにすることである。経験モデルは、いくつかのサンプルデータを用いて構築されたが、そもそも、各モデル構築に用いられた各サンプルデータのすべてを包含した母集団の分布状態・分布条件は分かっていない。どのような環境下のプロジェクトのデータが、母集団のどこに分布するのかが分からないのである。すなわち、各モデル構築に用いた異なるサンプルデータの条件の違いが明確ではない。したがって、そのサンプルデータで構築されたモデルの乗数・指数が異なる理由も、説明することは難しい。

次に課題2は、理論モデルからメタモデル ( $y = ax^b$ ) を導出することである。すなわち、なぜ、工数—工期の関係式が  $y = ax^b$  の形式で、多くのモデルが一致するのか、説明ができていないのである [1]。そのため、工数—工期の関係を規定する要因とそれらの間の関係を、理論的に解明す

る必要がある。

最後に課題3は、理論モデルから経験モデルを導出することである。Putnam モデル [11] では、1カ月かけて行う100人月の仕事を、2カ月かけて行った場合、理論的には、6.25人月で済んでしまう。他のモデルに比べて極端であるとの指摘がある [1]。また、Putnam モデルは経験論に対する説明もないとの指摘もある [32]。

以上をふまえて本論文では、課題1より、工数と工期の母集団データの分布を理論的に明らかにする。次に課題2より、その母集団データから、経験モデルがなぜ累乗関数 ( $y = ax^b$ ) で表されるのか、また、その乗数、指数がなぜ様々な値をとるのか、理論的に明らかにする。そして課題3として、導出した理論モデルと経験モデルを比較検証することで、その整合性を確認することを目的とする。

理論モデルはプロジェクト内でやりとりされる設計情報の写像経路と、その写像の難しさを変数としたモデルを用いる。以下に、なぜそのようなモデルを用いたか理由を述べる。この理論モデルは写像経路やその写像の難しさに着目したモデルであり、他の理論モデルのようにプロセスの組み立て方で工数と工期を試算する方法とは異なる。前述の先行研究としてあげた理論モデル (DSM [12] 等) では、特にプロセスに着目したモデルが多く、プロダクトの構造や要求仕様が工数や工期に及ぼす影響を定性的に議論していても数学的に整理するには至っていない。しかし、経験モデルでも COCOMO II [2] 等では、プロダクトや要求仕様も考慮した説明がされている。したがって、経験モデルと比較検証するうえで、その精度を向上させるためには、プロセスだけでなく、プロダクトや要求仕様を要素として考慮した理論モデルを用いることが必要と考えたことが理由である。

このような理論モデルを用いて、本論文では、経験モデルで示された累乗関数の乗数は写像経路とその写像の難しさに依存した関数で規定でき、またその指数は写像経路数のみに依存した関数で規定できることを示す。そして、これらの関数をもとに理論的に導出した近似式と、実際のプロジェクトデータとを比較し、本理論モデルが十分に実際のプロジェクトを説明していることを実証検証する。

本論文の構成は次のとおりである。まず2章で写像経路に着目したプロジェクトモデルから、工数—工期の決定要因とその関係を明らかにする。次に3章で理論モデルを構築する。3.1節では、2章で導かれた関係をもとに、プロジェクトとして生じうる工数—工期のデータの母集団の分布を示す。全データの母集団の分布をもとに、工数—工期のメタモデルとして、近似曲線  $y = ax^b$  を導出する。そして、近似曲線が  $y = ax^b$  の形式となる理由を考察する。さらに3.2節では、2章で示した工数—工期の決定要因が近似曲線の各係数 ( $a, b$ ) へどのような影響を及ぼすのか明らかにし、理論モデルを構築する。その後、4章において、

実際のプロジェクトデータと、3章までで導き出した理論モデルによる近似式（累乗関数）を比較し、本論文で示した理論モデルによって、経験モデルを説明できることを検証する。最後に、5章で、本論文をまとめる。

## 2. 工数—工期の基礎理論

本章ではまず、先行研究における工数—工期のとりえ方を説明し、次に本論文における工数—工期のとりえ方を説明する。

### 2.1 工数—工期の決定要因とその関係

#### 2.1.1 先行研究における工数と工期のとりえ方

先行研究 [9] において、多くのコスト要因が紹介されている。これらのコスト要因は、工数や工期に影響し、数学的にとらえるならば、工数や工期は多変数関数 ( $f_c, f_t$ ) と見なすことができる。ここで、 $x_i$  はコスト要因とする。

$$\text{工数} = f_c(x_1, x_3, \dots, x_i, \dots) \quad (1)$$

$$\text{工期} = f_t(x_1, x_2, \dots, x_j, \dots) \quad (2)$$

Boehm ら [33] がバリューチェーンで示したように、各要素が様々に影響して、“コスト”を構成している。そのため、工数や工期とその決定要因の関係を理解することは難しく、そのコントロールも難しいと考えられている。

#### 2.1.2 本論文における工数と工期のとりえ方・考え方

本論文では、工数—工期を以下のようにとらえ、その関係を検証する。まず、情報写像モデル [34] では、工数および、生産性（1人あたりの平均作業日数）が“複雑性”という変数と相関があることを示した（詳細は下記 2.1.3 項で述べる）。

$$\text{工数} \propto \text{複雑性} \times \text{要員数} \quad (3)$$

$$\text{生産性 (1人あたりの平均作業日数)} \propto \text{複雑性} \quad (4)$$

$$\text{複雑性} = \text{写像の難しさの平均} \times \text{全写像経路数} \quad (5)$$

なお、本論文では生産性を狭義にとらえ、式 (4) のとおり、1人あたりの平均作業日数としている [34]。

上記をもとに、本論文では生産性（1人あたりの平均作業日数）と工期の関係を明らかにし、工期が、工数と複雑性の関数  $F$  となっていることを示す。

$$\text{工期} = F(\text{工数}, \text{複雑性}) \quad (6)$$

#### 2.1.3 ソフトウェア開発プロジェクトモデル [34]

本論文で用いるプロジェクトモデル、写像、複雑性等は、我々の過去の研究 [34] に基づく。本項ではそれをまとめる。まず、プロジェクトモデルは情報写像モデルと呼ばれ、公理的設計論 [35] に基づく（図 1）。これによりソフトウェア開発プロジェクト内での設計情報の写像過程を表現する。本モデルでは、プロジェクトを構成する要素と

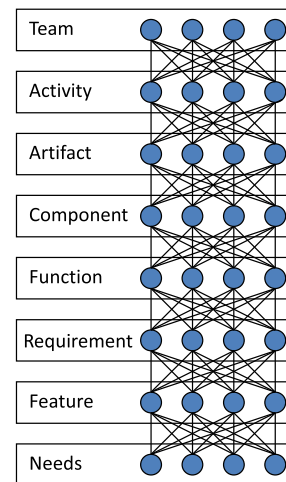


図 1 本論文で用いるプロジェクトモデル  
Fig. 1 Software development project model.

して、Team, Activity, Artifact, Component, Function, Requirement, Feature, Needs を抽出する（図 1 の丸印）。実際にモデリングするにあたっては、たとえば、Team とは個々の開発者とするケースもあれば、要求管理チーム、アーキテクチャ設計チーム等、プロジェクト内のチームとすることも可能である。Activity は開発で行われる作業である。Artifact とは、自然言語等で記述された設計書やソースコード等を指す。Component は個々のパッケージ、またはさらに詳細な粒度でモデリングするならば、Java クラス等を指す。その場合、Function は個々のメソッド等が該当する。Requirement は要求仕様、Feature は基本要件、Needs はニーズである。

また、これら要素間を結んだ線（図 1 の実線）は、設計情報の写像経路を示す。写像とは、たとえば自動車開発においては、自動車という 3次元で表される実体と、2次元で表現された設計書が存在する。どちらも同じ自動車を対象としているが、一方は 2次元で、また一方は 3次元で自動車という像を写し出している。このような関係を写像という。本モデルは、その各像を構成する要素をトレースし、それを線で結ぶことで、写像経路として示している。

#### 2.1.4 ソフトウェア開発における写像過程

次に、このような写像という考え方によって、開発作業がどのように解釈されるのか説明する。

たとえば、複数の設計書（Artifact）によって、1つの部品（Component）が設計されている場合、Artifact と Component の写像関係は  $n$  対 1 の関係になる。このとき、ある開発担当（Team）は、ある設計作業（Activity）を通して、他開発者（Team）の設計書（Artifact）を参照し、自担当の設計書（Artifact）との整合をとることで、1つの部品（Component）を正しく作ることができる。万一、ある設計書に 1つのミスが発見されれば、その設計書の作成作業の手戻りはもちろん、多くの関連作業の手戻りにもつな

がり、生産性低下の要因となる。さらに、写像関係が1対1であったとしても、たとえば設計書が独特な表記によって作成されていたのであれば、それを部品に置き換えることは難しい。独自表記を読み解き、正しく理解し、誤りなく部品を作成することが必要となる。そのため、ある既知のルールに基づいた設計書をもとにした場合に比べ、読み解き、正しく理解するために作業時間を要し、生産性は低下する。このようにプロジェクトを構成する要素の特性によっても、設計書から実体へ設計情報を写像することの難しさも変わる。すなわち、写像経路数と同様に、写像の難しさによっても生産性は変化する。

以上のような考え方から、設計情報の写像過程において、写像経路が少ないほど、作業の生産性は高く、また、プロジェクトを構成する各要素の写像が容易なほど、生産性が高い。すなわち、要素間の相互依存は冗長性が低く、かつ要素自体は写像が容易な特性であることが望ましい。一方、式(4)より、生産性が高まることは複雑性が低くなることと同意である。したがって、次項では、この複雑性の定義について説明する。

### 2.1.5 情報写像モデルの数理的表現

先行研究 [34] では、この情報写像モデルで示された要素間の関係を7つの行列の積で表現する数学的モデルを示した(式(7))。

$$\begin{array}{c}
 \text{プロジェクト} \\
 \text{行列}_A = \begin{array}{c} \text{Needs} \\ \text{Feature} \end{array} \times \begin{array}{c} \text{Requirement} \\ \text{Feature} \end{array} \times \begin{array}{c} \text{Requirement} \\ \text{Requirement} \end{array} \times \begin{array}{c} \text{Function} \\ \text{Function} \end{array} \\
 \times \begin{array}{c} \text{Component} \\ \text{Function} \end{array} \times \begin{array}{c} \text{Component} \\ \text{Component} \end{array} \times \begin{array}{c} \text{Artifact} \\ \text{Artifact} \end{array} \times \begin{array}{c} \text{Activity} \\ \text{Activity} \end{array} \times \begin{array}{c} \text{Team} \\ \text{Team} \end{array}
 \end{array}
 \tag{7}$$

このプロジェクト行列  $A$  により写像経路数、写像の難しさを算出する。各要素の持つ写像の難しさを 0.1 から 1.9 までで評価し (1.0 を基準とする)、その相乗平均値を成分値とした各行列の積を行列  $A$  としたとき、プロジェクトの写像の難しさを式(8)で定義した ( $a_{ij}$  は行列  $A$  の成分値)。

$$\text{写像の難しさ} = |A| = \Sigma a_{ij} \tag{8}$$

また、1つの写像経路は、図1に示したように、Team から Needs までの各要素を縦につなげた1列である。隣の列の要素と相互依存することなく、Team, Activity, Artifact というように、上下に1つずつ相互依存する。このような写像経路の数を並行経路数と呼び、

$$\text{並行経路数} = \text{行列 } A \text{ の次数} \tag{9}$$

と表した。それ以外の写像経路は、

$$\begin{array}{l}
 \text{冗長な経路数} = |A' - E| = \Sigma |a'_{ij} - e_{ij}| \\
 (e_{ij} \text{ は単位行列 } E \text{ の成分値})
 \end{array} \tag{10}$$

と定義した。ただし、行列  $A'$  は行列  $A$  の成分値を 0, または 1 (要素間の関連があれば 1, 関連がなければ 0) とした行列であり、 $E$  は単位行列である。 $a'_{ij}$  は行列  $A'$  の成分値を示す。

これらの写像経路数と写像の難しさの平均という2つの変数により、プロジェクトの複雑性を以下の方程式で定義した。ただし、式(11)の写像の難しさの平均とは、式(8)を式(9)、(10)の合計本数で割った値である。

$$\begin{array}{l}
 \text{複雑性} = \text{写像の難しさの平均} \times \text{並行経路数} \\
 \times \left( 1 + \frac{\text{冗長な経路数}}{\text{並行経路数}} \right)
 \end{array} \tag{11}$$

$$= \sum_i^{\text{全写像経路数}} \text{写像の難しさ}(i) \tag{12}$$

なお、式(5)は式(11)を変形した結果である。そして、先行研究 [34] では、この複雑性をもとに式(3)、(4)が成立することを実証的に示している。なお、この情報写像モデルは正方行列を前提としているが、必ずしも各要素がすべて同じ数となるとは限らない。したがって、要素数が不一致の場合は、成分値を 0 (要素間の関連なし) としたダミーの行または列を作成し、すべての要素数を合わせ、正方行列を作ることが必要である [34]。

## 2.2 本論文における工数と工期の関係式

工数と工期の関係を前項で説明した先行研究をふまえて整理する。まず一般的に工数と工期、要員数の関係は以下のように表される。なお、プロジェクトで行われる作業項目を Work Breakdown Structure (WBS) [36] として記述する。

$$\begin{array}{l}
 \text{工数} = \text{工期} \times \text{要員数} \\
 = \Sigma \{ WBS(i) \text{ の作業期間} \\
 \times WBS(i) \text{ の要員数} \}
 \end{array} \tag{13}$$

ここで、 $WBS(i)$  を写像経路  $i$  の写像作業ととらえると、式(13)は次のように変形できる。

$$\begin{array}{l}
 \text{工数} = \Sigma \{ \text{写像経路}(i) \text{ の写像期間} \\
 \times \text{写像経路}(i) \text{ に関わる要員数} \}
 \end{array} \tag{14}$$

すなわち、プロジェクトの工数は、ある写像経路1本の写像作業を担うサブプロジェクトの工数の総和となる。写像経路  $i$  を担うサブプロジェクトでは、式(12)より、その複雑性は、写像経路  $i$  の写像の難しさで決まる。さらに、式(4)より、その複雑性は、1人あたりの写像経路  $i$  の平均作業日数、すなわち、写像経路  $i$  の写像期間に比例する。したがって、式(14)は次のような比例関係で表せる。

$$\begin{array}{l}
 \text{工数} \propto \Sigma \{ \text{写像経路}(i) \text{ の複雑性} \\
 \times \text{写像経路}(i) \text{ に関わる要員数} \}
 \end{array} \tag{15}$$

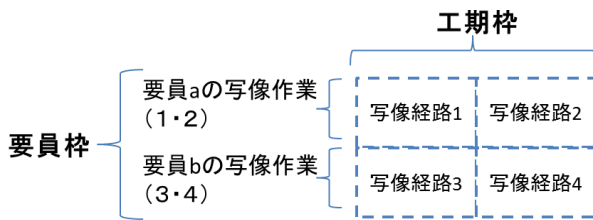


図2 要員数・工期・写像経路数の図形的解釈

Fig. 2 Diagram analysis for relation between mapping paths and personnel, development schedule.

前述のとおり、写像経路  $i$  の複雑性は、式 (12) より、写像経路 ( $i$ ) の写像の難しさに等しいので、式 (15) は次のように書き換えられる。

$$\text{工数} \propto \Sigma\{\text{写像経路 } (i) \text{ の写像の難しさ} \times \text{写像経路 } (i) \text{ に関わる要員数}\} \quad (16)$$

写像の難しさを 1 とすれば、ある写像経路に関わる要員数の合計が工数に比例することとなる。

### 2.3 写像経路数・要員数・工期の組合せの整理

前項をもとに、プロジェクトで存在しうる写像経路数、要員数および工期の組合せを洗い出す。

まず、式 (16) を図 2 のような長方形のモデルで考える。図 2 では、プロジェクトの各要員の割当て先となる要員枠を短辺にとり、各写像作業の工期枠 (写像の難しさの平均値  $k$  を 1 カ月/人とすれば、1 つの工期枠は 1 カ月。ただし、以降では、写像の難しさは、式 (4) における比例係数を含めたものとした) を長辺にとっている。図 2 では、写像経路 4 本を取り上げており、要員数 2 人、工期 2 カ月、すなわち工数は 4 人月プロジェクトが完遂できることを示している。写像経路 1 と 2 は、写像の難しさ 1 カ月/人で要員 a が 1 人で作業する。また、写像 3 と 4 は、写像の難しさ 1 カ月/人で要員 b が 1 人で作業する。式 (16) によれば、これらの合計がプロジェクトの全工数となる。

さらに写像経路 4 本の場合では、要員数・工期について、図 3 のようなバリエーションが考えられる (図 3 の (3) は、図 2 と同じ)。以下で、そのバリエーションを説明する。

図 3 の (1) と (5) は、それぞれ要員数 1 人で工期 4 カ月、要員数 4 人で工期 1 カ月、すなわち工数は 4 人月というケースを示している。図 3 の (2) は、要員数 2 名で工期 3 カ月というケースを示す。このケースでは、要員の 1 人が 3 本の写像作業を行ってしまえば、他の 1 人は 1 本の写像作業をするのみでプロジェクトは完遂する (工数は 4 人月)。また別に、1 本の写像作業しか受け持っていない 1 人の要員に、もう 1 人の写像作業に加わってもらい、協働で写像作業をすることも考えられる。したがって、写像経路 4 本を要員数 2 人で工期 3 カ月対処する場合のプロジェクトでは、工数が 4 人月、5 人月、6 人月となるケースが考えられる。上記は、要員 3 人で工数 2 カ月 (図 3 の (4)) で

No.	要員	工期	工数枠				写像経路の割当方と工数											
			要員①				要員②				要員③				要員④			
(1)	1人	4ヶ月	工期4ヶ月 1 2 3 4				4人月 要員① a b c d											
(2)	2人	3ヶ月	工期3ヶ月 1 2 3 4 5 6				4人月 要員① a b c 要員② d				5人月 a b c d b				6人月 a b c d b b			
(3)	2人	2ヶ月	工期2ヶ月 1 2 3 4				4人月 要員① a b 要員② c d											
(4)	3人	2ヶ月	工期2ヶ月 1 4 2 5 3 6				4人月 要員① a d 要員② b 要員③ c				5人月 a d b b c				6人月 a d b b c b			
(5)	4人	1ヶ月	工期1ヶ月 1 2 3 4				4人月 要員① a 要員② b 要員③ c 要員④ d											

図3 写像経路 4 本のとときの要員と工期の割当て方法

Fig. 3 In case of 4 mapping paths, variations of developers and development time.

も同様である。

以上のように、ある写像経路数に対して、想定されるすべての要員数・工期を一覧表に整理する。そして、その写像経路の枠数を数えることで、工数を求める。このような手順で、写像経路数、工期、工数のすべての組合せを洗い出し、プロジェクトの母集団データを整理する (その結果は、3.1 節でグラフ化して示す)。なお、上記では  $k = 1$  を前提とした。  $k \neq 1$  の場合は、式 (16) により工期が  $k$  倍 (図形の長辺が  $k$  倍) に伸びるが、長方形の枠数の考え方は変わらない。

## 3. 理論モデルの構築

前章の基礎理論をもとに、本章では工数工期の母集団データがどのような分布になるのかを明らかにする。それをもとに、近似式 (メタモデル) を導出する。そして、導出した近似式の累乗関数の乗数と指数の特性を説明する。

### 3.1 工数—工期の母集団データとその近似曲線

#### 3.1.1 母集団データとメタモデル

2.3 節で示した方法により、写像経路 1 本~10 本、15 本、20 本 (写像の難しさの平均は 1) でとりうる工数と工期を洗い出した。その結果、母集団データは図 4 のとおりとなった。

また、その近似曲線は累乗関数、 $\text{工期} = 2.0009 \times \text{工数}^{0.3695}$  となった。相関は弱いだが、次の 3.1.2 項で示すように母集団データの分布特性上、その近似曲線の関数の型 (累乗関数、対数関数等)、すなわちメタモデルは、累乗関数となる。

なお、本グラフは写像経路数が前述のとおり最大 20 本のケースである。さらに写像経路数が多いプロジェクトを母集団データに組み入れた場合は、半楕円形のような図形で示されたデータ分布は膨れていく (詳細は 3.2 節参照)。その結果、ある工数でとりうる工期の最大値も延びていく。たとえば、失敗プロジェクトの多くは、要求仕様の曖昧さ

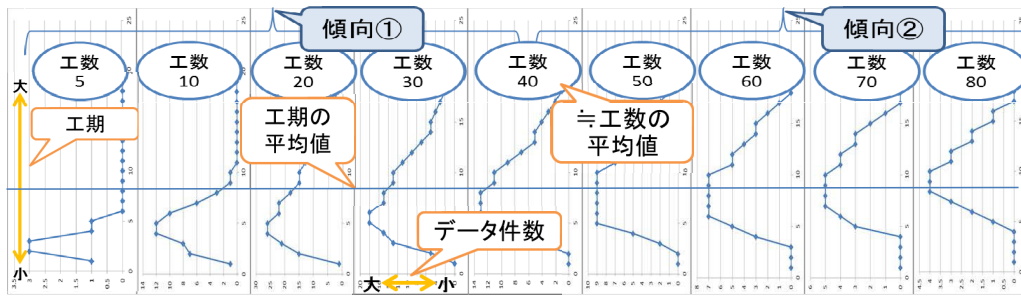


図 5 データの分布傾向

Fig. 5 Distribution state of data from Fig. 4.

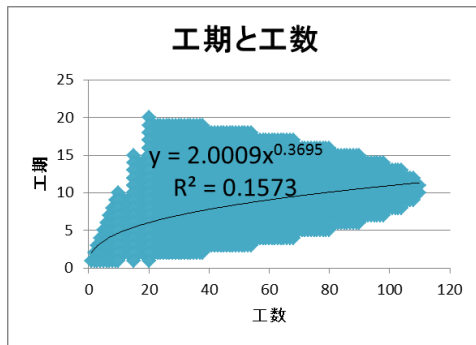


図 4 工数—工期の母集団データと近似曲線

Fig. 4 Population of man-month and development time data along with its fitted curve.

が指摘され、その整理や設計のために開発要員が増員される。これは図 1 に示したプロジェクトモデルで考えると、Requirement や Team の要素数が、プロジェクト開始当初に比較して、増加している状態である。すなわち、写像経路数は増大し、また曖昧な Requirements 等のため、写像の難しさの平均も増大している。このような状況下では、上記で述べたように、図 4 で示されたデータ分布も工数は増大する方向へ、そして工期も増大する方向へ膨らんでいくこととなる。その結果、近似曲線の形状も変化していく。その変化の特性については 3.2 節で説明する。

### 3.1.2 累乗関数となる要因

各工数でとりうる開発期間のデータ件数の分布傾向を示す。その傾向から、累乗関数となる理由を説明する。

図 5 は、図 4 を輪切りにした状態（各工数単位に、縦軸は開発期間、横軸はデータ数としたグラフ）を示した。各工数別グラフの横軸は、左側にいくほどデータ件数が多くなることを示している。図 3 で示したように、たとえば工期 2 カ月で工数 4 人月という組合せは 2 通りある。このように図 4 の各工数—工期のプロットは複数存在するものもある。図 5 のデータ件数はそれを示し、そのデータ分布の傾向は以下のようにまとめられる。

傾向 ① として、工数が 5 から平均値の 39.7 程度まででは、開発期間の平均値 (8.3) よりも小さい値をピークにデータが分布しており、平均値との乖離が大きい。傾向 ② として、工数の平均値 (39.7) を超えたあたりから、ほぼ

平均値程度の値をピークとしたデータ分布をしており、平均値との乖離が小さい。このため、各データと平均値との差分に着目する最小 2 乗法により、母集団の近似曲線を求めると、工数—工期の関係式は、指数が 1 以下の累乗関数となる。なお、工数 = 1 で工期 = 0 ではないため、対数関数は近似式とはなりえない。

### 3.2 メタモデルから理論モデルの導出

次に、メタモデルの性質について述べ、理論モデルを導出する。近似曲線は、プロジェクトの複雑性（写像経路数、写像の難しさの平均）を変化させると、近似曲線も変化する。この性質により、各経験モデルも説明できる。

本節では、工期が下記の式 (17) で表される理論モデルとなることを示す。

$$\begin{aligned} \text{工期} &= f(\text{写像経路数}, \text{写像の難しさの平均}) \\ &\quad \times \text{工数}^g(\text{写像経路数}) \\ &= f(\text{複雑性}) \times \text{工数}^g(\text{写像経路数}) \end{aligned} \quad (17)$$

なお、上記理論モデルの各係数の具体的な決め方については、4.2 節で述べる。

#### 3.2.1 写像経路数と乗数・指数の関係

本項では、写像経路数の変化が乗数・指数にどのように影響し、式 (17) のような関係式となるのか説明する。まず、写像経路数を変えた際に、3.1 節で示した近似曲線がどのように変化するか調べた。図 6 の上図のように、近似曲線の乗数と写像経路数の関係は、写像経路数が増えるに従い、乗数の値も増加する比例関係にある。その関係式は、

$$Y = 0.065X + 0.7121 \quad (18)$$

で表される。その  $R^2$  は 0.9994 であり、高い相関関係にあることが分かる。また、近似式の指数は、図 6 の下図のように写像経路数が増加するに従って減少する傾向があり、

$$Y = 0.8092X^{-0.26} \quad (19)$$

で表される。その  $R^2$  は 0.995 であり、高い相関関係にあることが分かる。

#### 3.2.2 写像経路数による乗数・指数の変化要因

本項では、写像経路数の増減により、なぜ、前項のよう

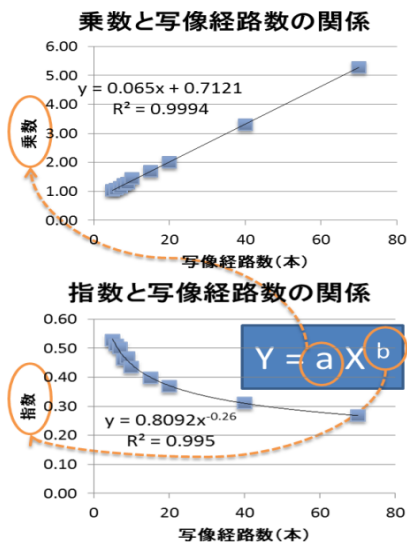


図 6 写像経路数と乗数・指数の関係

Fig. 6 Relations between number of mapping paths and multiplier, exponent.

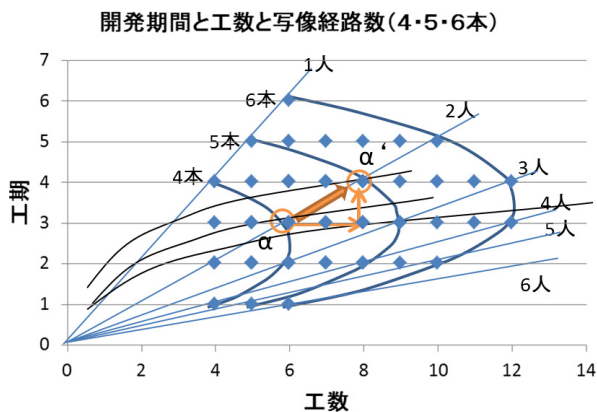


図 7 写像経路数と工数—工期の関係 (図 4 の拡大図)

Fig. 7 Relation between mapping paths and man-months, development schedule (magnified Fig. 4).

な変化をするのか説明する。図 7 は、前述の図 4 を拡大したものである。工数を横軸に、工期（開発期間）を縦軸にとっている。グラフ上の四角い点はプロジェクトでとらえる工数—工期のデータである。データを結んだ楕円の曲線はプロジェクトの写像経路数を示している。

すなわち、4本の写像経路を持ったプロジェクトでは、工数が4人月で工期が4カ月、または工数が6人月で工期が3カ月、工数が6人月で工期が2カ月、工数が4人月で工期が1カ月というプロジェクトが存在していることを示している。ただし、このとき、写像の難しさの平均は1カ月/人である。

また、もう1つデータを結んだ直線は、要員数を表している。すなわち工数4人月で工期4カ月ならば要員数は1名、また工数5人月で工期5カ月でも同様に要員数1名のプロジェクトである。

このようにグラフを見ると、たとえば、工期3カ月・工

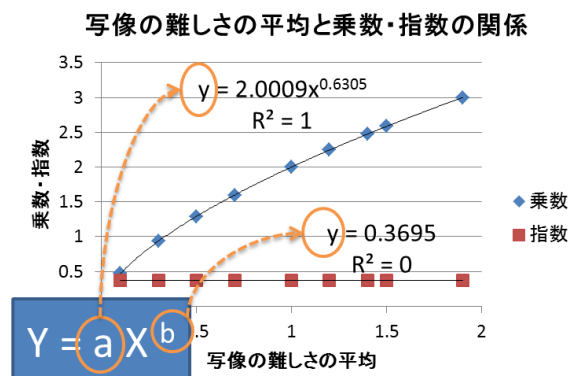


図 8 写像の難しさと乗数・指数の関係

Fig. 8 Relation between difficulty of mapping paths and multiplier, exponent.

数6人月のプロジェクト (α) に対し、経路を1本追加する。その際、追加した経路の写像作業分、工数が膨らむ。しかし、要員数は変わらない。そのため、α'にデータは移行する。その際、経路数が増加した場合のX軸に対する変化を説明する。経路数が増加することによって、処理すべき写像経路数が増えたために工数が追加が必要となる。そのため、近似曲線はX軸に対して伸びることとなる。すなわち、近似曲線の指数が、小さい値に変化する。次に、経路数が増加した場合のY軸に対する変化を説明する。経路数が増加することにより、写像処理を行うための工期が必要となる。したがって、元の経路数のときに比べて工期が延長する。すなわち、近似曲線の乗数が、大きい値に変化する。

### 3.2.3 写像の難しさと乗数・指数の関係

本項では、当初1に固定していた写像の難しさの平均を変化させた場合に、近似曲線の乗数・指数の変化を調べた。

図 8 より、写像の難しさの平均が増すに従って、乗数は増加するが、指数は一定値をとることが分かった。写像の難しさの平均に対する乗数の変化は

$$Y = 2.0009 \times X^{0.6305} \quad (20)$$

で表される。そのときの  $R^2$  は1であり、非常に高い相関関係にあることが分かる。

また、写像の難しさの平均に対する指数の変化は

$$Y = 0.3695 \quad (21)$$

で表される。 $R^2$  は0であり、写像の難しさの平均とはまったく相関しないことが分かった。ただし、本サンプルでは写像経路数が20本のプロジェクトをもとにしているため、この値となっている。経路数を変えれば、式(21)の定数値も変化する。詳細は次項で説明する。

### 3.2.4 写像の難しさによる乗数・指数の変化要因

本項では、写像の難しさの平均の変化が乗数・指数にどのように影響するのか説明する。

写像経路数が変わらないため、プロジェクトモデルの構

造も変わらない。しかし、写像の難しさの平均値が  $k$  倍になれば、工数と工期は  $k$  倍される。したがって、データ分布状態は変化しないため、指数は変わらず、乗数のみ変化する。

$x$  人月の工数で  $y$  カ月の写像作業（要員数  $n$  人）を行ったとき、乗数を  $a$ 、指数を  $b$  とすると

$$y = a \times x^b \quad (22)$$

$k$  倍された工数 ( $x \cdot k$  人月) の作業では、工期が  $y \cdot k$  カ月となる（プロジェクト構造は変化しないので、要員数は  $n$  人のまま）。

$$y \cdot k = a' \times (x \cdot k)^b \quad (23)$$

上記式 (23) を式 (22) で割り、式を整理することにより、

$$a' = a \times k^{1-b} \quad (24)$$

すなわち、20 本の経路数における乗数と指数を元の乗数、元の指数とすれば、

$$\text{乗数} = \text{元の乗数} \times \text{写像の難しさの平均}^{(1-\text{元の指数})} \quad (25)$$

として表される。

### 3.3 モデル粒度の規格化（補正方法）

前項までで説明した理論の正しさを検証するために、実際のプロジェクトデータと比較検証を行う。しかし、実際のプロジェクトデータを用いるためには、1 点留意すべき点がある。モデル化する際、それを構成する要素の粒度は規定されていない。したがって、同じプロジェクトを異なる粒度でモデル化することもある。その場合、同じプロジェクトでも写像経路数や写像の難しさが異なってしまうため、それらを補正し、規格化する必要がある。

たとえば、同じプロジェクトでも、以下の 2 つのようなモデル化が可能である。1 つは、Team, Needs をそれぞれ 1 つの要素で、写像経路を 1 本でモデル化する。また別に Team, Needs をそれぞれ 2 つの要素で、写像経路で各要素をすべてつないだ 4 本の写像経路でモデル化することもできる。もし、この 2 つのモデルが同一のプロジェクトであるならば、工期は等しくなければならない。そこで、本論文では経路数が 1 本、すなわち Team, Activity 等各カテゴリの要素を 1 つのみのモデルを基準とする。規格化前のモデルの経路数を、この基準に換算して規格化後の経路数を算出する。以下がその算出方法である。

上記のとおり、経路数 1 本の基準モデルは、その要素の関係を表すプロジェクト行列（式 (7)）の次数が 1 次の行列として表現できる。そのとき、規格化前のモデルの次数を  $n'$ 、また、その写像経路数を  $m'$  とすると、規格化前のモデルのプロジェクト行列は  $n'$  次の行列となり、Team と Needs の組合せは、基準モデルの  $n' \times n'$  倍となる。すな

わち、 $n' \times n'$  倍の細かな粒度でモデルが構成されている。したがって、以下のような比例関係が成り立つ

$$\begin{aligned} 1 \text{ 本} : m' \text{ 本} &= 1 \text{ 次行列の成分数} : n' \text{ 次行列の成分数} \\ &= 1 \times 1 : n' \times n' \end{aligned} \quad (26)$$

式 (26) を変形し、以下の換算式を導出した。

$$m' \div (n' \times n') = 1 \quad (27)$$

写像の難しさの平均についても同様に考え、以下の関係式を導出した。（ $k$  は規格化後、 $k'$  は規格化前の写像の難しさの平均）

$$k = k' \times n' \times n' \quad (28)$$

上記の 2 つの式で規格化された  $k, m$  に換算・補正する。

## 4. 理論検証

前章で理論的に導いた複雑性と乗数・指数の関係式を、実際のプロジェクトのデータにあてはめ、理論的に近似式を導出する。そして、実際のプロジェクトデータと比較検証を行い、本論文で説明した理論の整合性を確認する。

### 4.1 プロジェクトの概要

検証にあたっては、ゲームソフトウェア開発プロジェクトの開発データを用いた。対象は 21 プロジェクトであり、それらの工期・工数・写像経路数・写像の難しさの平均値は表 2 のとおりである。ただし、Boehm [1] と同様に、1 カ月は 19 日として工数—工期を算出している。また、写像の難しさの平均値は以下のように算出した。

式 (3) を変形すると、

$$\text{工期} = \frac{\text{工数}}{\text{要員数}} \propto \text{複雑性} \quad (29)$$

となる。さらに式 (5) より、式 (29) は式 (30) のように変形できる。

$$\text{工期} \propto \text{写像の難しさの平均} \times \text{全写像経路数} \quad (30)$$

したがって、写像の難しさの平均値は式 (31) によって求まる。

$$\text{写像の難しさの平均} \propto \frac{\text{工期}}{\text{全写像経路数}} \quad (31)$$

なお、表 2 は、写像の難しさの平均値を定数倍し、式 (31) を比例式ではなく等式とした場合の値である。

### 4.2 近似式の理論的導出

本節では近似式の理論モデルを策定する。はじめに、写像経路数の補正值を算出する。補正值は前述の式 (27) より、以下の算出式から求める。なお、補正前の写像経路数・写像の難しさは、表 2 の全 21 プロジェクトの平均値を用いた。



表 2 実プロジェクトの検証用データ

Table 2 Data for validation of actual software development projects.

プロジェクト	工数 (人月)	工期 (月)①	全写像 経路数②	写像の難しさの 平均値(①÷②)
1	3.187	0.83	1566	5.28.E-04
2	3.761	0.97	1566	6.19.E-04
3	2.850	0.74	1566	4.74.E-04
4	3.334	0.86	1566	5.51.E-04
5	3.008	0.78	1566	4.99.E-04
6	2.950	0.77	1566	4.90.E-04
7	3.342	0.70	1794	3.92.E-04
8	2.618	0.68	1566	4.33.E-04
9	2.861	0.74	1566	4.71.E-04
10	2.782	0.72	1566	4.59.E-04
11	2.834	0.73	1566	4.67.E-04
12	3.155	0.81	1566	5.18.E-04
13	3.303	0.85	1566	5.42.E-04
14	3.111	0.68	1794	3.77.E-04
15	2.442	0.63	1566	4.02.E-04
16	2.605	0.67	1566	4.28.E-04
17	2.337	0.60	1566	3.85.E-04
18	2.863	0.73	1566	4.69.E-04
19	2.495	0.64	1566	4.10.E-04
20	2.758	0.71	1566	4.52.E-04
21	2.621	0.57	1794	3.17.E-04

$$\begin{aligned}
 m &= \text{全 21 プロジェクトの写像経路数平均値} \\
 &\div (\text{行列の次数} \times \text{行列の次数}) \\
 &= 1599 \div (18 \times 18) \tag{32}
 \end{aligned}$$

この補正值  $m$  を用いると、乗数係数と写像経路数の関係式 (式 (18), (19)) により、乗数と指数は以下のとおりとなる。

$$a_0 = 0.065 \times m + 0.7121 = 1.0329 \tag{33}$$

$$b = 0.8092 \times m^{-0.26} = 0.5343 \tag{34}$$

次に、写像経路数と同様に、写像の難しさの補正を式 (28) により算出する。

$$\begin{aligned}
 k &= \text{写像の難しさの全 21 プロジェクトの平均値} \\
 &\times (\text{行列の次数} \times \text{行列の次数}) \\
 &= 0.000461 \times 18 \times 18 \tag{35}
 \end{aligned}$$

この補正值  $k$  を用いて、指数係数と写像の難しさの平均の関係式 (式 (24)) より、乗数を確定する。

$$a = a_0 \times k^{1-b} = 0.4261 \tag{36}$$

となる。

以上から、工期と工数の関係式は以下のように定義される。

$$\text{工期} = 0.4261 \times \text{工数}^{0.5343} \tag{37}$$

### 4.3 工数—工期の理論値と実測値の比較検証結果

前述のとおり、全プロジェクトの平均値から算出した工

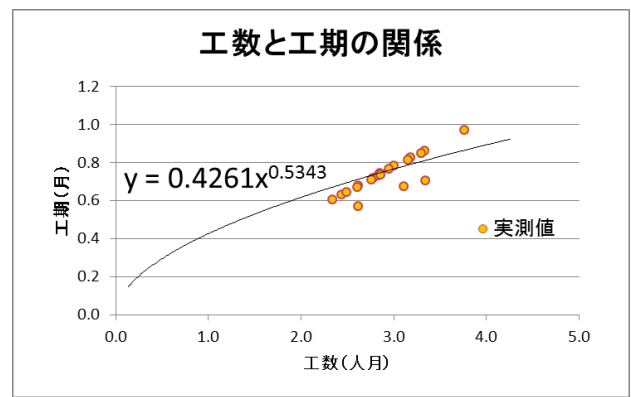


図 9 理論値と実測値の比較結果

Fig. 9 Comparison of theory and measured results.

期と工数の関係式の理論曲線 (式 (37)) と実測値を比較する。図 9 のとおり、実測値は、理論曲線上の近傍に現れている。以下で、この理論値と実測値の関係について考察する。

まず、経験モデルとして、実測値から線形近似 (式 (38)) を導出した。

$$\text{工期} = 0.2343 \times \text{工数} + 0.0508 \tag{38}$$

式 (38) と実測値の相関係数は 0.867 である。同様に、実測値から累乗近似 (式 (39)) を導出した。

$$\text{工期} = 0.274 \times \text{工数}^{0.9191} \tag{39}$$

式 (39) と実測値の相関係数は 0.858 である。線形近似より多少相関が弱い。一方、理論的に求めた累乗近似 (式 (37)) では、相関係数が 0.865 となり、線形近似の場合に近い値となった。なお、 $p$  値は、どのケースも 0.005 以下である。

以上のように、今回の実証検証では、線形近似の方が累乗近似よりも相関係数が高い結果となった。しかし、一般的に経験モデルが累乗関数となることは、表 1 でも示したとおり、多数の研究でいわれていることである。本論文で示した理論モデルは、経験モデルの累乗近似よりも高い相関を示し、かつ、本検証で用いた 21 プロジェクトの実測値とは高い相関を示した線形近似と比較しても遜色のない相関係数となっている。したがって、本論文で示した理論モデルは経験モデルのメカニズムを明らかにしたと考える。

### 4.4 工期の実測値と理論値の相関と誤差

前節では、21 プロジェクトの平均値より理論モデルを求めた。本節では、21 プロジェクト個々の実測値と理論値の比較結果を示す。

表 2 で示した各プロジェクトの写像経路数、写像の難しさの平均により理論式を算出し、実測値の工数を代入することで工期—理論値を求めた。また、理論値で用いた工数に対応する実測した工期を工期—実測値とした。この 2 つ

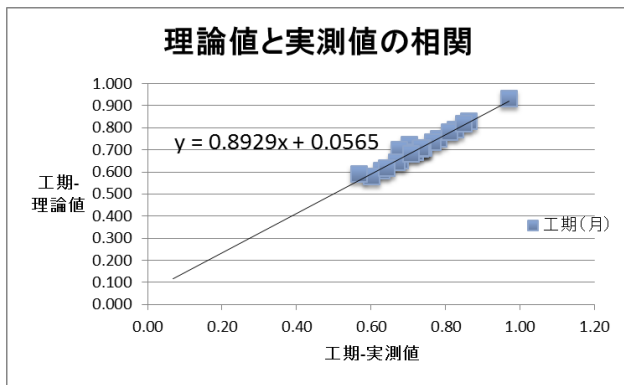


図 10 理論値と実測値の相関

Fig. 10 Correlations between theoretical value and observed value.

の工期を Y 軸, X 軸にとり, グラフ (図 10) を作成した. すなわち, 理論値が正確に実測値を再現しているほど, 近似式の傾きは 1 に近い値となり, また y 切片も 0 に近いこととなる. 図 10 のように本論文で構築したモデルは, 傾きも 1 に近く, また y 切片も 0 に近い. したがって, 理論値は実測値を正確に再現できていることが分かる. 実際, その相関係数は 0.982 である. また, 理論値と実測値の誤差は 1 日以下であり, 工期の実測に対し, 理論値との誤差は「-0.40 日~0.7 日」であった. 以上から, 本論文で構築した理論モデルは, 個々の実測値をほぼ再現できるモデルであると考えられる.

## 5. 結論

本論文では, 工数と工期の経験モデルがなぜ累乗関数 ( $y = ax^b$ ) で表されるのか, そして, その乗数, 指数がなぜ様々な値をとるのか, 理論モデルを用いて明らかにした. その理論モデルはプロジェクト内でやりとりされる設計情報の写像経路と, その写像の難しさの平均を変数としたモデルを用いて構築した. 本論文の結果, 乗数は写像経路とその写像の難しさの平均に依存した関数で規定され, 指数は写像経路数のみに依存した関数で規定されることを示した.

そして, これらの関数をもとに理論的に算出した近似式と, 実際のプロジェクトデータとを比較し, 本理論モデルが十分に実際のプロジェクトを再現していることを実証検証した. すなわち, 写像経路数, 写像の難しさの値が様々なプロジェクトをもとに, 工数—工期のモデルを構築したため, 先行研究のモデルは様々なものとなっていることが分かった. 同時に, 以下に示すとおり, 先行研究の課題を解決した. まず, 課題 1 として, 経験モデルのデータ分布状態を理論モデルから説明することができていなかった. 本論文により, 理論モデルから, 発生し得るデータ分布 (データの母集団) を明確化した. 次に, 課題 2 として, 理論モデルからメタモデルを導出することができていなかっ

た. 本論文により, データの母集団から近似曲線を求め, 関係式  $y = ax^b$  を導出した. 最後に, 課題 3 として, 理論モデルから経験モデルを導出することができていなかった. 本論文により,  $y = ax^b$  の各係数値の決定要因は, 写像経路数と写像の難しさであることを示した. 以上により, 工数—工期の関係における課題を解決した.

最後に, 本論文では, 汎用性を考慮したプロジェクトで検証を行っている. しかし, ゲーム開発プロジェクトに閉じた検証しかできておらず, 汎用性を十分に検証したとはいえない. 今後は, 様々なプロジェクトで検証を行い, 本論文成果の汎用性についても実証していきたい.

**謝辞** 本論文に対し, 多大なご協力をいただいたイレギュラーズアンドパートナーズ株式会社代表取締役山本一郎様, チーフディレクター渡邊謙太様に感謝いたします. また, 本研究の一部は「慶應義塾クラスター研究推進プロジェクトプログラム (安全)」の援助により行われた. 記して謝意を表する.

## 参考文献

- [1] Boehm, B.W.: *Software Engineering Economics*, Prentice-Hall, Engle-wood, Cliffs, N.J. (1981).
- [2] Boehm, B.W. et al.: *Software Cost Estimation with COCOMO II*, Prentice Hall (2000).
- [3] Walston, C.E. and Felix, C.P.: A method of programming measurement and estimation, *IBM Systems Journal*, Vol.16, No.1, pp.54-73 (Mar. 1977).
- [4] Reifer, D.: Estimating web development costs: there are differences, *Cross talk*, June (2002).
- [5] ソフトウェア開発 データ白書 2012-2013, 独立行政法人情報処理推進機構 (IPA) 技術本部 ソフトウェア・エンジニアリング・センター (SEC), 入手先 (<http://sec.ipa.go.jp/users/publish/SEC-TN12-002.pdf>) (参照 2014/08/11).
- [6] ユーザー企業 ソフトウェアメトリックス調査【調査報告書】2011 年度版, 2011 年 9 月, 社団法人日本情報システム・ユーザー協会, 入手先 (<http://www.juas.or.jp/servey/library/pdf/11swm.pdf>) (参照 2014/08/11).
- [7] 経済調査研究所 研究レポート 2010 ソフトウェア開発データリポジトリの分析, 2010 年 7 月, 財団法人 経済調査会 経済調査研究所, 入手先 ([http://www.zai-keicho.or.jp/pdf/software\\_metrics/pdf/ERA\\_Analysis\\_of\\_Software\\_Projects\\_Data\\_Repository\\_201007.pdf](http://www.zai-keicho.or.jp/pdf/software_metrics/pdf/ERA_Analysis_of_Software_Projects_Data_Repository_201007.pdf)) (参照 2014/08/11).
- [8] 戸田航史, 松本健一, 押野智樹, 高橋明彦: ソフトウェア開発における適正工期に関する分析, *経済調査研究レビュー*, Vol.7, pp.37-46 (2010).
- [9] Bailey, J.W. and Basili, V.R.: A meta-model for software development resource expenditures, *Proc. 5th International Conference on Software Engineering* (1981).
- [10] Kemerer, C.F.: An Empirical Validation of Software Cost Estimation Models, *Communications of the Association for Computing Machinery*, Vol.30, No.5, pp.416-429 (1987).
- [11] Putnam, L.H.: A general empirical solution to the macro software sizing and estimating problem, *IEEE Trans. Softw. Eng.*, Vol.4, No.4, pp.345-361 (1987).
- [12] Steward, D.V.: The design structure system: A method for managing the design of complex systems, *IEEE Trans. Engineering Management*, Vol.28, No.3,

pp.71-74 (1981).

[13] Sosa, M.E., Eppinger, S.D. and Rowles, C.M.: Are your engineers talking to one another when they should?, *Harvard Business Review*, Vol.85, No.11, pp.133-142 (2007).

[14] Eppinger, S.D.: Model-based approaches to managing concurrent engineering, *Journal of Engineering Design*, Vol.2, No.4, pp.283-290 (1991).

[15] Eppinger, S.D.: Using the design structure matrix to estimate product development time, *Proc. ASME Design Engineering Technical Conferences* (1998).

[16] Eppinger, S.D.: Innovation at the speed of information, *Harvard Business Review*, Vol.79, No.1, pp.149-158 (2001).

[17] Eppinger, S.D. and Salminen, V.: Patterns of product development interactions, *International Conference on Engineering Design, ICED '01*, August 21-23 (2001).

[18] Danilovic, M. and Sandkull, B.: The use of dependence structure matrix and domain mapping matrix in management uncertainty in multiple project situations, *International Journal of Project Management*, Vol.23, No.3, pp.193-203 (2005).

[19] Danilovic, M. and Browning, T.R.: Managing complex product development projects with design structure matrices and domain mapping matrices, *International Journal of Project Management*, Vol.25, pp.300-314 (2007).

[20] Lindeman, U., Maurer, M. and Braun, T.: *Structural Complexity Management - An Approach for the Field of Product Design*, Springer, Berlin (2009).

[21] Forrester, J.W.: *Industrial Dynamics*, MIT Press, MA. (1961).

[22] Lyneis, J.M. and Ford, D.N.: System Dynamics Applied to Project Management: A survey, assessment, and directions for future research, *System Dynamics Review*, Vol.23, No.4, pp.157-189 (2007).

[23] Abdel-Hamid, T.: The Dynamics of Software Development Project Management, Doctoral Thesis, MIT, Cambridge, MA. (1984).

[24] Cooper, K.: Naval Ship Production: A Claim Settled and a Framework Built, *Interfaces*, Vol.10, No.6, (1980).

[25] Ford, D. and Sterman, J.: Dynamic Modeling of Product Development Processes, Working Paper, 3943-97, MIT Sloan School of Management, Cambridge, MA. (1997).

[26] Homer, J., Sterman, J., Greenwood, B. and Perkola, M.: Delivery time reduction in pulp and paper mill construction projects, *Proc. 1993 International System Dynamics Conference*, The System Dynamics Society, Cancun, Mexico (1993).

[27] Rodrigues, A. and Bowers, J.: System dynamics in project management: a comparative analysis with traditional methods, *System Dynamics Review*, Vol.12, No.2, pp.121-139 (1996).

[28] Lyneis, J.M., Cooper, K.G. and Els, S.A., Strategic management of complex projects: a case study using system dynamics, *System Dynamics Review*, Vol.17, No.3, pp.237-260 (2001).

[29] Park, M. and Pena-Mora, F.: Dynamic change management for construction: introducing the change cycle into model-based project management, *System Dynamics Review*, Vol.19, No.3, pp.213-242 (2003).

[30] Taylor, T. and Ford, D.N.: Tipping point failure and robustness in single development projects, *System Dynamics Review*, Vol.22, No.1, pp.51-71 (2006).

[31] Lyneis, J.M. and Ford, D.N.: System dynamics applied

to project management: A survey, assessment, and directions for future research, *System Dynamics Review*, Vol.23, No.2/3, pp.157-189 (2007).

[32] Heemstra, F.J.: Software cost estimation, *Information & Software Technology*, Vol.34, No.10, pp.627-639 (1992).

[33] Boehm, B.W. and Papaccio, P.N.: Understanding and Controlling Software Costs, *IEEE Trans. Softw. Eng.*, Vol.14, No.10, pp.1462-1477 (1988).

[34] 榮谷昭宏, 牧野泰才, 前野隆司: 設計情報の写像経路に着目したソフトウェア開発プロジェクトの複雑性指標の提案, 情報処理学会論文誌, Vol.55, No.5, pp.1453-1470 (2014).

[35] Suh, N.P.: *Axiomatic Design: Advances and Applications*, Oxford University Press, New York (2001).

[36] Project Management Institute, Inc.: プロジェクトマネジメント知識体系ガイド (PMBOK ガイド) 第4版 (2008).



榮谷 昭宏 (学生会員)

1992年学習院大学理学部物理学科卒業, 1994年同大学大学院自然科学研究科物理学専攻修士課程修了, 同年NTT(日本電信電話株式会社)入社. 2010年慶應義塾大学大学院システムデザイン・マネジメント研究科修士課程修了. 現在, 同研究科後期博士課程在籍, NTTコムウェア株式会社勤務.



高田 眞吾 (正会員)

1990年慶應義塾大学理工学部卒業. 1992年同大学大学院理工学研究科修士課程修了. 1995年同博士課程修了. 博士(工学). 同年奈良先端科学技術大学院大学情報科学研究科助手. 1999年慶應義塾大学理工学部情報工学科専任講師. 助教授(准教授)を経て, 2015年より同大学教授. ソフトウェア工学の研究に従事. 電子情報通信学会, 日本ソフトウェア科学会, ACM, IEEE CS各会員.



前野 隆司

1984年東京工業大学工学部機械工学科卒業, 1986年同大学大学院理工学研究科機械工学専攻修士課程修了, 同年キヤノン株式会社入社. 1993年博士(工学)学位取得(東京工業大学). 1995年慶應義塾大学専任講師, 同大学助教授を経て, 現在, 慶應義塾大学大学院システムデザイン・マネジメント研究科委員長/教授.