

# 組み込みソフトウェア開発プロジェクトに関する プロジェクト管理モデルの検討

岩田 員典<sup>†</sup> 阿南 佳之<sup>††</sup> 中島 豊四郎<sup>†††</sup>

近年の情報化の進展にともない、ソフトウェアを開発している企業や部門においては、それらのソフトウェアを定められた品質、コスト、期間を確保しながらいかに効率良く開発するかが非常に重要になってきている<sup>1),2)</sup>。そのため、これらのソフトウェア開発を行っている企業・部門はプロセス改善をはじめとしたさまざまな改善を行っている<sup>3)~5)</sup>。そこで、本論文では企業と連携することによって、通常は公開されないソフトウェア開発プロジェクトの実績データの分析を行い、その分析結果から見積り時よりも工数がかかってしまう失敗プロジェクトの判別手法について検討した。そして、この判別手法に基づいたプロジェクト管理を支援するツールを、すでに導入されている進捗状況管理ツールに統合する形で実装した。さらに、これまで利用されていた新規開発プロジェクトの工数を見積る手法の見直しについて検討し、統計的手法を利用することで新規モデルを作成した。このように本研究では産学連携することによって企業内のみで扱われている入手困難なプロジェクトの実績データを扱える貴重な機会を得ることができた。さらに、企業側としてはプロジェクトの分析・評価に関する研究を企業内の人的資源は確保したまま効率的にコストを抑えて行うことができた。

## Studies on Project Management Models for Embedded Software Development Projects

KAZUNORI IWATA,<sup>†</sup> YOSHIYUKI ANAN<sup>††</sup>  
and TOYOSHIRO NAKASHIMA<sup>†††</sup>

Software productivity and quality are very important for companies with the advance of information technologies. Many companies or departments improve the processes of embedded software developments. In this paper, we analyze the practical data of embedded software developments, which are normally veiled, in cooperation with a company. We discuss and develop a model to judge whether a software development project will be success or failure. We implement the model as the software tool integrated into the current project management system. In addition, we review the current measure to estimate the number of man-day of new software development project and discuss a new statistical measure.

### 1. はじめに

近年の情報化の進展にともない社会の広い範囲で情報機器が使用されることとなり、その機器も高性能・高機能化を極め、搭載されている組み込みソフトウェアも複雑化してきている<sup>1),2)</sup>。したがって、これらの組み込みソフトウェアを開発している企業や部門においては、それらのソフトウェアを定められた品質、コスト、期間を確保しながら、いかに効率良く開発するか

が非常に重要になってきている。そのため、これらのソフトウェア開発を行っている企業・部門はプロセス改善をはじめとしたさまざまな改善を行っている<sup>3)~5)</sup>。しかしながら、これらの改善は1つの企業や1つの部門での限定的な取り組みにとどまっている場合が多い。その理由としてそのような組織において、このような改善にスタッフを配置することは、人的資源の関係から効率的とはいえずコスト面から見ても望ましくないからである。さらにその組織固有のノウハウを含んでいるため、その詳細まで公にされることは非常に少ない。

一方で、大学においては研究のスタッフは確保でき、かつ発展的な研究ができるにもかかわらず、これらの改善に企業と大学が協力して進めた事例も非常に少ない。なぜなら、企業とのつながりが少ないため、そのよ

<sup>†</sup> 愛知大学経営学部

Faculty of Business Administration, Aichi University

<sup>††</sup> オムロンソフトウェア株式会社業務革新部 SPI 推進グループ  
Omron Software Co., Ltd.

<sup>†††</sup> 椋山文学園大学文化情報学部

Department of Culture-Information, Sugiyama Jogakuen University

うな改善計画に参加する機会が得られないからである。そこで、企業と大学が連携すればお互いの不足部分を補い合うだけでなく、双方の利益にもつながる。このような状況にあって、従来から企業の開発マネージャと面識があり、ソフトウェアのプロセス改善について不定期に意見の交換をしていた第3筆者の知人を通じて今回の産学連携研究を進めるところとなった。本論文ではこのような背景のもと、O社において金融機関向け端末機器のソフトウェアを開発している部門との共同研究によるソフトウェア開発プロジェクトの実績データの分析・評価とプロジェクト管理を支援するツールの開発について述べる。さらに、これまで利用されてきた新規開発プロジェクトの工数を見積る手法の見直しについて検討する。

2章ではソフトウェア開発プロジェクト管理について、現状での問題点と本研究で行うべき課題について説明する。3章ではデータベースを分析しプロジェクトの状況を判断するモデルの確立とプロジェクト管理を支援するツールの実装について述べる。4章では新規プロジェクトの工数を見積る手法の検討について述べる。最後に5章で本研究のまとめと今後の課題について述べる。

## 2. ソフトウェア開発プロジェクト管理と課題

O社の金融機関向け端末機器の組み込みソフトウェアの開発プロジェクトでは、基本開発されたソフトウェアをベースにカスタマイズし、より多くの顧客に納入するという手法を採用している。基本開発においては、カスタマイズ時に新たにプログラムコードを書くのができるだけ少なくするために、パラメータで制御する仕組みを組み込むなど、設計技術に関しても、生産性・品質の向上に積極的に取り組んできている。このような取り組みを続けている中、近年コストダウン要請が従来以上に強くなり、納期だけでなく、品質・コストの大幅な向上が必要となってきた。そこで、開発プロジェクトの管理のため、進捗状況や結果をデータベースに入力・管理するツールを運用し、このデータベースを利用することで組織全体として品質や生産性の向上を図ってきた。しかし、それでも失敗プロジェクトが皆無になることはなかった。そこで、ここでは計画から遅れてしまうプロジェクトにはどのような問題があるのかについてデータベースから分析し、このような失敗プロジェクトを減らすモデルの検討を進める。このソフトウェア開発部門の従来のプロジェクト管理は以下のように進められていた。

(a) プロジェクトリーダー、担当者とも、納期のみ

に着目して進捗状況を判断する。

- (b) プロジェクトリーダーや担当者が各自の経験と勘でプロジェクト遅延における問題の有無を判断する。
- (c) 問題が発生した時点でその場しのぎの対応をする消火型である。

このような属人的な方法が失敗してしまうプロジェクトを発生させていると思われた。そこで、従来の管理に対しては次のような方法が効果的と思われる。

- (i) 進捗状況は納期にかかわらず、工数や要員といった資源の投下状況、成果物の出来具合と出来高、マイルストーンの達成状況といった複数の情報に基づき判断する。
- (ii) 閾値や管理限界といった定量的な指標を設定し、それに基づいて問題の有無を判断する。
- (iii) 事前に検討していた対応策を真摯に実施する。

(i) については、各種のプロジェクト管理技法やCMM<sup>6)</sup>に代表されるプロセスモデルで取り上げられた分野であり、一般的な知識を獲得し実践するスキルを身につければ、実行できる。(iii)については、開発対象ソフトウェアの機能や使用領域、開発プロセスへの依存度が高く定性的である。これは組織としてノウハウを蓄積し展開する仕組みを確立することで実現できる。

一方、(ii)については定量的な取り組みが必要である。過去の実績データから成功、もしくは失敗したプロジェクトを分析し統計的にモデルを確立できれば、そのモデルと進行中プロジェクトを照らし合わせることで将来の予測が可能である。また、成功プロジェクトを基に新規プロジェクトを見積るモデルを確立すれば、そのプロジェクトが成功する割合が高くなる。しかしながら、このソフトウェア開発部門においては間接的なコスト削減が1つの重要な課題となっているため、時間をかけたデータ分析ができず、実績データを活用できていなかった。そこで、本研究では前述(ii)のような方式を確立し定着・推進するために既存データを分析しプロジェクト管理モデルの確立を行った。

## 3. プロジェクト管理モデル

### 3.1 データの選択

O社の金融関連機器のソフトウェア開発部門の開発工程は基本的にはウォーターフォールモデル<sup>7)</sup>を採用しており、その工程の概要を表1に示す。この機器ご

(R)CMM and Capability Maturity Model are registered in the U.S. Patent and Trademark Office.

表 2 蓄積データの分類と採否  
Table 2 Classifications and selections of data.

分類	データ	採否	理由
内部工数	役割別作業工数の予定と実績	採	工数管理システムなどで収集される，定量的で精度の高いデータである．
規模情報	新規・修正・修正元・流用元の各ステップ数	採	定量的で精度の高いデータである．
工数情報	工程別の実績工数	採	工数管理システムなどで収集される，定量的で精度の高いデータである．
商品情報	工程別の手戻り工数	否	手戻りの定義が一貫しておらず，データの精度が低い．
	商品区分と機種	採	商品の特性や開発プロセスの特性を，モデルに反映させる必要がある．
	顧客名とサブテーマ名	否	定性的要素が強く，精度の高いデータを収集できない．
外注関係	開発区分（新規 or 改造）	否	2 分類しかないのでモデルのパラメータとしては不十分である．
	出荷時期	否	時期による変動はほとんどないので採用しない．
	外注額の予定と実績	否	外注額は営業要素を含んでいるため，プロジェクトの進捗の実態には則していない．
品質情報	外注先作業工数の予定と実績	否	外注額から算出するため，営業要素が含まれるので採用しない．
	工程別の不具合数	否	品質程度の評価は納入先の品質基準に依存することから，モデルに必要な定量的な情報を得ることは難しい．

表 1 ソフトウェア開発の工程  
Table 1 Processes of software development.

	工程名	工程の作業内容
1	構想設計 (CD)	いわゆるシステムエンジニアリング業務である．顧客要件を分析し，ソフトウェアで実現する項目を開発要件として具体化する．
2	設計	CD 工程で作成された開発要件に従って，ソフトウェアの機能分割・ソフトウェアのモジュールの組合せ・ソースプログラムの作成などの設計業務を行う．
3	デバッグ (DB)	設計工程で作成された成果物が，設計どおりに作成されていることを実機を使って動作確認する．設計作業を行った当事者が実施する．
4	テスト (TG)	デバッグが完了したソフトウェアが，顧客要件を満足していることを確認する．設計・デバッグ作業とは異なる第三者が実施する．

とのソフトウェア開発をテーマと呼び次のように定義する．

### 定義 3.1 テーマ

システムを構成している機器ごとのソフトウェア開発をテーマと呼ぶ．

また，開発に際してプロジェクトの生産性や品質に関する各種データは，それぞれプロジェクトの開始前，実行中，プロジェクトの完了時に表 2 に示すように内部工数，規模情報などの各種データをとることとなっている．これらの蓄積データのうち工数の見積りに関するデータは，ベテランのプロジェクトリーダーが類似のテーマの実工数を参考に算出している．算出におい

ては COCOMO<sup>8)</sup> の見積り手法と同じように開発規模を主にソースコード行数 (LOC: Lines of Code) で表す方法を用いている．この見積りは仕様変更などが発生した場合に当初見積りを見直して修正される．また，COCOMO II<sup>9)</sup> のように LOC にオブジェクトポイント，またファンクションポイントを加味し，かつ開発サイクルを初期のアプリケーション組み立てモデル，初期設計モデル，ポストアーキテクチャモデルの 3 段階で見積る方式はとっていない．その理由は，この部門が開発しているソフトウェアは開発規模的には 0.5 ~ 10 人月の範囲とかなり小規模であることがあげられる．また，開発のタイプはプロダクトライン型開発<sup>10)</sup> が多いことから見積りが開発の初期段階で比較的容易に確度良くできるためと，COCOMO の調整要因である開発設備，スタッフ，手法，ツールなどが比較的均質であり，これらを調整要因として考慮する必要がないからである．

ファンクションポイント<sup>11)</sup> による見積りについても 7, 8 年前に検討されたが，組み込みソフトウェアは業務システムのソフトウェアに対してファンクションの定義やファンクションの重み付け係数を算出するのが非常に難しく，本部門において試行はされたが本格的に導入するには至っていない．

分析に先立ち，これらの蓄積された各種データについてモデル導出の検討対象とするかどうかの採否を判定した．データの分類，採否とその理由を表 2 に示す．

### 3.2 工数比率によるデータの分析とモデル化

3.1 節で採用することになったデータを対象に分析をした．まず，データの傾向は機種により異なるので

表 3 テーマの判定結果と工数比率  
Table 3 Results of project themes and man-day rate.

判定	テーマ数	CD 工数比率	設計工数比率(平均)	DB 工数比率(平均)	TG 工数比率(平均)
成功	8	6.57%	31.29%	26.93%	35.21%
失敗	10	10.51%	37.45%	25.41%	25.75%
例外	9	3.44%	34.25%	32.50%	28.61%
空白	59	6.95%	40.61%	26.38%	25.73%

「商品区分と機種」を基に分類し機種ごとに分析をした。機種は機種 X、機種 Y と機種 Z の 3 種類に分類可能であったが、機種 X を除いてはデータ数が少ないため、統計的分析をするのに有効なデータ数が得られた機種 X のプロジェクトを分析の対象とした。モデルの検討において、品質に関するデータは不具合の件数が基となっているが、この不具合がその製品の機能・性能にどの程度の影響を及ぼすかのレベル付けがなされていないため、成功・失敗の判断が困難である。また、開発期間については開発工数が計画時点の工数内であれば失敗とはいえないので、工数について検討する。ウォータフォールモデルでは、一般には上流工程で品質を作り込むことが重要であり、下流工程よりも上流工程に多くを配分すべきであるといわれている。そこで、工程別の工数比率に着目し、対象のテーマについて成功テーマ、失敗テーマ、例外テーマ、空白テーマに分類して以下のように定義する。

#### 定義 3.2 成功テーマ

テーマの計画工数に対する実績工数の比率が  $100 \pm 20\%$  以内のとき、そのテーマを成功テーマとする。

#### 定義 3.3 失敗テーマ

テーマの計画工数に対する実績工数の比率が  $120\%$  より大きいとき、失敗テーマとする。

#### 定義 3.4 例外テーマ

テーマの計画工数に対する実績工数の比率が  $80\%$  より小さいとき、例外テーマとする。

#### 定義 3.5 空白テーマ

データの一部が未登録なテーマは判定が不能なため、空白テーマとする。

ここで、成功テーマは当初の計画どおりに進んだテーマを意味している。また、例外テーマは計画工数よりも実績工数があまりにも少ないため、計画に問題があったと見なし分析から取り除かれるテーマを意味する。

上記定義に従い、ある一定期間の 96 テーマについて、テーマごとに成功か失敗かを判定し、成功・失敗の工数比率の平均を求めた。テーマごとの成否判定結果と工数比率の平均を表 3 に示す。表 3 において下流工程の工数比率が多いのは、組み込みソフトウェア

開発の特徴である実機を使ったデバッグ・テストでかなりの時間を割いていることを示している。

ウォータフォールモデルと表 3 から上流工程 (CD+設計) と下流工程 (DB+TG) の割合に着目しプロジェクトの状況を判断するモデルを以下のように定義する。本モデルで当該テーマの指標値を計算し遂行されているプロジェクトの進捗状況を判定する。

#### 定義 3.6 プロジェクト判断モデル

指標値：設計工数比率 = (CD 工数 + 設計工数) / 総工数

指標値	判定	説明
32%以上 ~ 42%未満	成功	リスクが小さく 問題はないプロジェクト
22%以上 ~ 32%未満	危険	リスク管理対象
42%以上 ~ 57%未満		
0%以上 ~ 22%未満	失敗	すでに問題が顕在化して おり、是正処置が必要
57%以上 ~ 100%以下		

### 3.3 モデルの進捗管理への応用

本モデルはプロジェクトの設計工程が完了した時点で利用し、当該プロジェクトの進捗管理を行うことができる。その手順は以下のとおりである。

- (i) 設計工程が完了した時点で、それまでの実績工数を集計する。
- (ii) 後続工程 (DB, TG) の工数は、計画策定時点から変化しないものと仮定したうえでモデルと照らし合わせる。
- (iii) モデルによる判定を得る。
- (iv) 判定結果から必要に応じて後続工程の工数の追加、もしくは再設計など完了済み工程工数の追加を実施する。

### 3.4 モデルの評価

本モデルを 3.3 節の手順に従い新規完了プロジェク

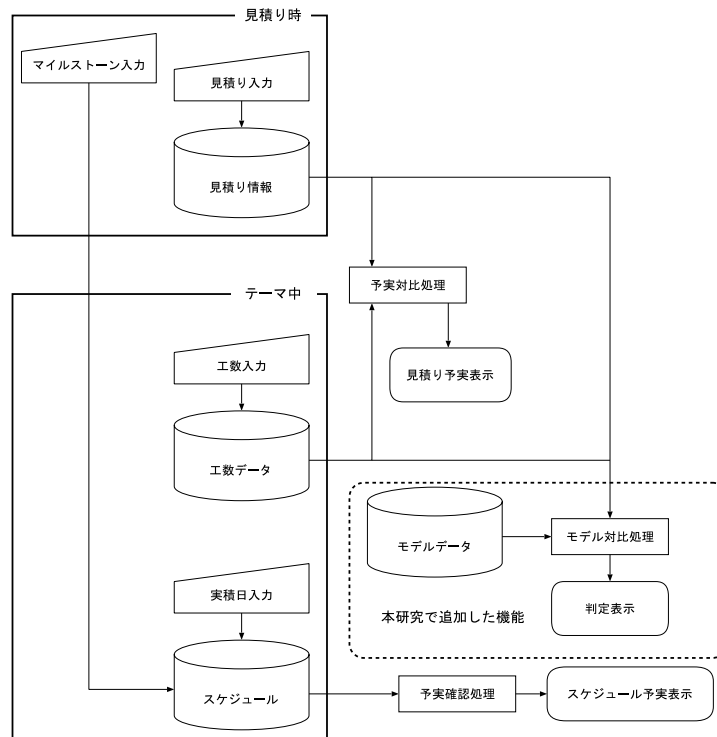


図1 プロジェクト管理モデル

Fig. 1 Our project management model.

ト 18 件（成功 8 件，失敗 10 件）に適用し進捗状況の判断を行った．結果は次のとおりである．

- 成功テーマを正しく判断した件数：3 件
- 成功テーマを誤って判断した件数：5 件
- 失敗テーマを正しく判断した件数：7 件
- 失敗テーマを誤って判断した件数：3 件

このようにこのモデルは，失敗テーマの 7 割を正しく判断できた．残りの 3 割の成功テーマと誤って判断したケースを分析したところ，設計量の割合に比べて総工数が極端に少ない分類上異なるテーマにすべきものであることが分かった．このような特殊なテーマに関しては本モデルの適用外とするか，新たなモデルを検討する必要がある．また，成功テーマを失敗テーマと誤って判断するケースがあるが，これについてはフェイルセーフ的に考えて運用でカバーすることとする．このように，本モデルは完全とはいえないが，実用上問題ないと思われる．

### 3.5 モデルの実装

定義 3.6 に示したモデルをユーザが使用する際の新規ツール利用の負担軽減もあり，すでに導入されている進捗状況管理ツールに統合する形で実装する（図 1）．

従来ツールは HTML と VBScript により作成され

ており，市販のデータベースソフトと連携している．そこで，本モデルで利用するデータをモデルデータとして追加し，指標値を計算する部分は VBScript で実装した．これによりモデルデータを変更するだけでプロジェクトの評価基準を容易に変更することが可能である．

さらに，本システムはプロジェクトを成功予測の場合は青色，危険な場合は黄色，失敗予測の場合は赤色で表示することで，プロジェクト管理者が瞬時に判断できるようにしている（図 2）．これにより，プロジェクト管理者はプロジェクトがどのような状態にあるのかを判断しプロジェクトを管理することができる．

## 4. プロジェクトの工数見積りモデル

### 4.1 従来モデル

従来，ソフトウェア開発プロジェクトの工数見積りには，十数年前に経験豊富なベテランのプロジェクトの経験者らによって制定された式 (1) がこれまで運用されてきた．右辺の各要素がプロジェクト発足時に見積られ，それによって計画工数が求められる．

$$S = X_{new} + X_{modify} + \alpha_1 \times X_{survey} + \alpha_2 \times X_{reuse} \quad (1)$$

すべてのプロジェクトをチェックし完了します。現在  
現在の色分けは以下のようになっています。 0.00 0.40 0.80 1.00

プロジェクト	開発内容	se	okaki	total	ct	hanna	ohno	Total	CO-2R PCT/total
0	5.75	43	0	7	9.25	305	40	408	68%
05	1.46	1	7	7	5.04	9	3	17	50%
0	5.00	0	0	2	0.00	2	0	2	50%
4	13.71	43	2	0.5	9.76	343	20	373	50%
05	5.33	2	1	0.5	3.67	8	0	12	50%
1	2.50	2	0	0.5	2.50	115	0	117	50%
3	2.50	2	0	0.5	2.50	115	0	117	50%
3	7.28	4	0	0	5.42	24	0	29	50%
7	20.08	22	10	2	22.40	84	0	106	50%
50	20.47	10	0	2	20.48	80	0	100	50%
50	27.68	30	0	2	31.60	1175	0	1206	50%
30	6.00	0	0	2	0.00	12	0	12	50%
2	1.00	0	0	1	0.00	10	0	10	50%
05	1.46	1	0	0	1.04	5	0	6	50%
3	11.67	7	2	0	8.20	30	0	38	50%
3	2.04	0.5	0	0	1.48	7	0	8	50%

図 2 ツールの実行例

Fig. 2 An example of tool.

ここで、 $S$  は開発対象容量を表し、 $X_{new}$ 、 $X_{modify}$ 、 $X_{survey}$ 、 $X_{reuse}$  は、それぞれ新規容量、修正容量、修正元容量、流用元容量を意味する。また、 $\alpha_1$ 、 $\alpha_2$  は、あらかじめ固定値（1 未満の係数）が定められている。この開発対象容量  $S$  に生産性指標を乗じて計画工数  $M$  の算出を行う。

各要素は以下の意味を持つ。

新規容量：対象範囲の中で新しく作成し追加する関数のステップ数

修正容量：対象範囲の中で元の関数を修正（追加および変更）するステップ数

修正元容量：修正した関数の元のステップ数、修正していないが流用するために関数の中身を検証した関数のステップ数と削除したステップ数

流用元容量：対象範囲の設計のために中身を見ずに外部仕様だけで流用した関数のステップ数

式 (1) は COCOMO と同様に開発規模を LOC により表し、プロダクトライン型開発の特徴である新規開発、修正、流用を全体の LOC に反映したものである。つまり、 $\alpha_1$  と  $\alpha_2$  に 1 よりも小さな数を持つてくることで、工数の見積りは新規容量と修正容量に重点をおく形で用いられている。しかし、今日では従来に比べて新規容量が微少でも流用元容量が莫大となっており、開発対象容量も大きくなるといった状況に式 (1) は対応しきれなくなっているのではないかと想定される。そこで、本研究では式 (1) よりもより精度の高い式が導出できるかについて検討する。

#### 4.2 重回帰分析によるモデルの算出

##### 4.2.1 モデルの算出

従来モデルは  $\alpha_1$  と  $\alpha_2$  の 2 つの係数しか使用していないため、新規容量と流用元容量の関係などを表現できていない。またどのプロジェクトでも開発量に関係のない作業も必要である。そこで式 (2)<sup>12)</sup> の係数を 3.2 節で使用した機種 X のデータを用いて重回帰分析によって求める。ここで、式 (1) とは異なり式 (2) で

表 4 工数比率による検証結果

Table 4 Verification of our estimation model I.

計画工数/実工数の平均 (%)	分散	標準偏差 (%)	平均 $\pm 1\sigma$ (%)
101.30	2.67	16.35	84.95 ~ 117.65

表 5 修正容量と他の独立変数の相関

Table 5 Correlation of modification steps with other independent variables.

	新規容量	修正元容量	流用元容量
修正容量	0.333	0.655	-0.205

は計画工数  $M$  を直接算出する。算出においては、プロジェクトは開発範囲・技術を限定したカスタマイズ開発のため開発対象容量  $S$  に乗ずる生産性指標に大きな変動はないと仮定し、式 (1) の左辺は工数に置き換えている。

$$M = \alpha_1 \times X_{new} + \alpha_2 \times X_{modify} + \alpha_3 \times X_{survey} + \alpha_4 \times X_{reuse} + \beta \quad (2)$$

分析はデータを標準化したものに対して行い以下の式 (3) を算出した。

$$M = 30.05 \times X_{new} + 1.65 \times X_{modify} + 23.84 \times X_{survey} + 17.42 \times X_{reuse} + 55.90 \quad (3)$$

##### 4.2.2 モデルの検証

式 (3) のモデルが正しく工数を見積ることができるかどうかを「計画工数/実工数」という工数の比率に着目し検証した。その結果を表 4 に示す。

この結果から、本モデルはプロジェクトの約 68% を工数比率 84.95% ~ 117.65% の範囲内に納めて予測できることが分かる。また、プロジェクトの約 95% は 68.60% ~ 134.00% の範囲内に納めて予測できるため、実用上利用可能な精度であるといえる。

##### 4.2.3 標準回帰係数の検証

式 (3) より、修正容量の標準回帰係数 1.65 は他の標準回帰係数よりもかなり小さいことが分かる。このことから修正容量は計画工数の算出に寄与していない可能性がある<sup>13),14)</sup>。そこで、修正容量と他の独立変数の相関を算出すると表 5 のようになった。

この結果から修正容量は修正元容量と正の相関があり、そのため計画工数の算出に寄与していないと考えられる。そこで、さらに t 値を利用して修正容量が計画工数の算出に必要なかどうか以下のように仮説を立て判断する。

- 「仮説：修正容量の真の標準回帰係数は 0 である」が優位水準 5% で棄却されるか否か。
- 仮説に対する t 値は 0.76 となる。
- 本研究で用いたデータの場合は t 値の絶対値が

表 6 工数比率による検証結果 2  
Table 6 Verification of our estimation model II.

計画工数/実工数の平均 (%)	分散	標準偏差 (%)	平均 $\pm 1\sigma$ (%)
101.40	2.55	15.96	85.44 ~ 117.36

2.09 以上のとき仮説は優位水準 5%で棄却される。このことから  $0.76 < 2.09$  となるので仮説は棄却されず「修正容量の真の標準回帰係数は 0 である」が成り立つ。よって、修正容量の標準回帰係数は 0 となり計画工数算出には必要ないことが分かる。また、同様の分析を他の係数に行ったところ、仮説はすべて棄却されたので、新規容量、修正元容量、流用元容量は計画工数算出には必要である。

以上から、本研究で分析したプロジェクトに関しては、修正容量は計画工数の算出には直接は寄与しておらず、修正元容量に内包される形で寄与していることが分かる。

#### 4.2.4 修正容量を使用しないモデル

4.2.3 項から、修正容量は計画工数に寄与しないので、あらためて、新規容量、修正元容量、流用元容量のみを使用し重回帰分析を行った。その結果を式 (4) に示す。

$$M = 30.03 \times X_{new} + 24.93 \times X_{survey} + 17.10 \times X_{reuse} + 55.90 \quad (4)$$

本モデルが正しく工数を見積ることができるかどうかを 4.2.2 項と同様に「計画工数/実工数」という工数の比率に着目し検証した。その結果を表 6 に示す。

表 6 から修正容量を用いない方が表 4 よりも分散が小さくなっており精度が向上したといえる。

このように従来式 (1) は現状と合っていないことが明らかになり、早急な対応が必要であることが明らかになった。

ここでは、従来の工数の見積りに用いている式を蓄積データを基に見直ししたが、見積りの手法には前述したファンクションポイント法や COCOMO II などの手法がある。これらの手法は開発が比較的大規模なビジネスアプリケーション分野を対象にしたもので、組み込みシステムにそのまま適用することは難しいが、それらがこの部門で適用できるか否かを検討していく必要がある。ただし、組み込みソフトウェア開発はプロダクトライン型開発であると同時に、開発されたシステムはハードウェアと協調してはじめて動作する。そのため応答時間、タイミング、CPU やメモリなどのリソース面での制約を検討に際して考慮しなければならない。また、フェールセーフやフェールソフトなどの対策も重要である。これらの非機能要求の実装の

困難性など<sup>10)</sup>をどのように取り扱うかなども考慮する必要がある。

## 5. おわりに

### 5.1 まとめ

本論文では企業との産学連携共同研究により行ったソフトウェア開発プロジェクトの実績データを分析・評価し、プロジェクト管理を支援するツールの開発について説明した。

本研究によってこれまで活用しきれていなかったソフトウェア開発プロジェクト管理のためのデータベースを分析し、進行中のプロジェクトの状況を判断するモデルを作成した。さらに、プロジェクト管理ツールと連携する形でモデルを実装することで、利用者の負担を軽減する形での新たなツール導入を行った。また、従来利用されてきた工数見積りの式を見直し、新規モデルを作成した。その結果から本研究で分析したプロジェクトに関しては、従来工数見積りに必要であると考えられていた修正容量は必要ないことが判明するとともに、新しいモデルへの変更が必要なことが明らかになった。本研究ではモデルの作成と検証に同じデータを用いたが、O 社の組み込みソフトウェア開発プロジェクトはかなり成熟しているため、分析したデータから大きく逸脱するプロジェクトはまれである。そのため、今後のプロジェクトから得られるデータも本モデルに適合するといえる。また、その他の開発プロジェクトや他社におけるプロジェクトなどに本モデルを適用する場合には、モデルの本質を変更することはなくデータの様式に従って変数の数などを変更することで対応できる。

産学連携の本研究では、プロジェクトの分析・評価に関する研究を企業内の人的資源は確保したまま効率的にコストを抑えて行うことができた。さらに、大学側としては企業内のみで扱われている入手困難なプロジェクトの実績データを扱える貴重な機会となった。また、ここで検討した組み込みソフトウェアは、これまでハードウェアの付帯物として開発された経緯があり、必ずしも工学的な手法を駆使した開発がなされていない。一方、ソフトウェア工学はビジネスアプリケーション分野を対象に発展した経緯から、組み込みソフトウェア開発に必ずしも十分にその力を発揮できていなかった<sup>1)</sup>。その意味で本研究は本部門におけるソフトウェア工学に基づいた取り組みのスタートになることが期待できる。

### 5.2 今後の課題

今後の課題には次のようなものがあげられる。

- さらなる完了プロジェクトのデータ蓄積  
現在蓄積されているデータは部分的に不完全なものが多くそれらは分析に利用できなかった。今後は利用可能なデータを蓄積していく必要がある。
- 生産性指標の変動の検証  
新規プロジェクトの工数見積りを行う際に生産性指標に大きな変動はないと仮定したが、新たなデータの蓄積が進むと生産性指標に大きな変動が現れる可能性がある。そのため、新規のデータが蓄積された時点で生産性指標の変動幅の検証を進めていく予定である。

謝辞 本研究の機会を与えていただいたオムロンソフトウェア株式会社の関係各位に深く感謝いたします。

### 参 考 文 献

- 1) 平山雅之: 組み込みソフトウェア開発の現状, 情報処理, Vol.45, No.7, pp.677-681 (2004).
- 2) 中本幸一, 高田弘幸, 田丸喜一郎: 組み込みシステム技術の現状と動向, 情報処理, Vol.38, No.10, pp.871-878 (1997).
- 3) 小笠原秀人, 小島昌一: 定着を重視プロセス改善活動, 情報処理, Vol.44, No.4, pp.334-340 (2003).
- 4) 込山俊博: ソフトウェアプロセス成熟度向上のための基盤技術の開発と展開, 情報処理, Vol.44, No.4, pp.341-347 (2003).
- 5) 高木徳生: ソフトウェア中心の大規模システム開発プロジェクト事例にみる成功要因と考察, 情報処理, Vol.44, No.4, pp.348-356 (2003).
- 6) Paulk, M.C., Curtis, B., Chrissis, M.B. and Weber, C.V.: Capability Maturity Model for Software, Version 1.1, CMU/SEI-93-TR-24 (1993).
- 7) Boehm, B.W.: Software engineering, *IEEE Trans. Softw. Eng.*, Vol.C-25, No.12, pp.1226-1241 (1976).
- 8) Boehm, B.W.: *Software Engineering Economics*, Englewood Cliffs, NJ: Prentice Hall (1981).
- 9) Boehm, B.W., et al.: *Software Cost Estimation with COCOMO II*, NJ: Prentice Hall (2000)
- 10) 鷓林尚靖: 組み込みソフトウェアのモデリング技術, 情報処理, Vol.45, No.7, pp.682-693 (2004).
- 11) Jones, C.: *Applied Software Measurement*, McGraw-Hill (1991).
- 12) Humphrey, S.W.: *A Discipline for Software*

*Engineering*, Addison-Wesley (1995).

- 13) Manly, B.F.J.: *Multivariate Statistical Methods A PRIMER*, Chapman and Hall Ltd, London (1986). 村上正康, 田栗正章 (訳): 多変量解析の基礎, 培風館 (1992).
- 14) 長谷川勝也: ホントにわかる多変量解析, 共立出版株式会社 (1998).

(平成 16 年 9 月 2 日受付)

(平成 17 年 2 月 1 日採録)



岩田 員典 (正会員)

1975 年生。1998 年名古屋工業大学知能情報システム学科卒業。2000 年同大学大学院電気情報工学専攻博士前期課程修了。2003 年同大学院電気情報工学専攻博士後期課程修了。博士 (工学)。同年 4 月愛知大学経営学部講師。現在に至る。エージェント指向技術, 計算論に興味を持つ。人工知能学会会員。



阿南 佳之

1966 年生。1988 年鳥取大学工学部電気工学科卒業。同年オムロンソフトウェア入社。金融機関向け端末機器のソフトウェア開発に従事。1999 年ソフトウェアプロセス改善活動に専任。2004 年業務革新部門にて, 社内のプロセス改善活動推進に従事。専門: ソフトウェア工学。



中島豊四郎 (正会員)

1946 年生。1970 年名古屋工業大学電気工学科卒業。1972 年同大学大学院修士課程修了。同年立石電気 (現オムロン) 入社。現在, 椋山女学園大学文化情報学部教授。この間, PLC (Programmable Logic Controller), 流通システム (基本・アプリケーションソフトウェアを含む) などの研究開発, ソフトウェア開発の生産性・品質向上のための標準化活動, ソフトウェア工学, 社会情報, 情報処理の教育・研究に従事。博士 (工学)。電気学会, 日本ソフトウェア科学会, 電子情報通信学会, 日本シミュレーション学会, IEEE 各会員。