

組み込み型マイクロプロセッサ用ソフトウェアにおける コストモデル

藤 村 直 美†

ソフトウェアの生産性を向上するためには、生産性に影響を与える要因が明確であること、それらを定量的に計測・評価できることが大前提である。これまで大規模なソフトウェア開発については生産性を評価するためにいくつかのコストモデルが提案されているが、組み込み型マイクロプロセッサ(MP と略す)用ソフトウェアのように、高機能、高品質で小規模なソフトウェアを小人数で短期間に開発するような開発形態に対応したコストモデルはほとんど検討されていない。本論文では数年間に渡って行ったアンケート調査に基づいて、組み込み型 MP 用ソフトウェアにおけるコストモデルを構築し、これが COCOMO の BASIC モデルと同様のコストモデルになることを示す。JEIDA モデルと名付けたこのコストモデルに基づいて、組み込み型 MP 用ソフトウェアの開発規模、実績工数、開発期間の関係を客観的に示す。すなわち、組み込み型 MP 用ソフトウェアの開発は、従来から検討されている大規模ソフトウェアの開発とは異なる特徴を有すること、また、CPU ビット、記述言語、製品用途によってソフトウェアの開発規模、実績工数、開発期間の関係に差異が存在することを示す。これらの特徴は従来直観的に理解されてはいたが、ここで示すように定量的に示されたことはない。また、このコストモデルの提案によって、組み込み型 MP 用ソフトウェアの生産性を定量的に議論する基盤を確立している。

A Cost Model for Software in Embedded Microprocessors

NAOMI FUJIMURA†

It is important to measure and evaluate software productivity quantitatively. Several cost models were proposed for large scale software projects. However, few cost models have been proposed yet for small scale software projects in which a few people develop software in a short period as in embedded microprocessors. We have been collecting data about software productivity in embedded microprocessors. We proposed a new cost model for small scale software projects and named JEIDA model. It is similar to the Basic COCOMO model. We also discuss the characteristics around the software size, actual man-months, and development period quantitatively. There is the significant difference between our JEIDA model and the traditional cost models such as COCOMO. There are the meaningful differences of the characteristics from the view point of CPU bits, description languages, and product purposes in our JEIDA model. We have also established the base to discuss the productivity of software in embedded microprocessors quantitatively.

1. はじめに

高機能、高品質のソフトウェアを効率良く生産できることは重要である¹⁾。一般的に工業製品の製造において生産性を向上するためには生産性に影響を与える要因が明確であること、それらを定量的に評価できることが大前提である。しかしながらソフトウェアは、従来の工業製品のように生産性に影響を与える要因が明確でなく、生産性の定量的な評価が困難である。そうした中で、ソフトウェアの生産性を定量的に評価するための試みとして、大規模ソフトウェアを対象として COCOMO モデル^{2),3)}や Putnam モデル⁴⁾のよう

なコストモデルが 1970 年代から提案されている⁵⁾。

ところで直接の関係者以外にはあまり注目されない組み込み型マイクロプロセッサ(MP と略す)が存在する。組み込み型 MP 用ソフトウェアは高機能、高品質という特徴に加えて、機器に組み込まれるために各種の制約を受け、規模は小さく、小人数で短期間に開発されるものが多い。組み込み型 MP は種類も数量も大変に多く、この分野におけるソフトウェアの生産性は社会的にも経済的にも大きな影響がある。しかしながら従来のコストモデルは大規模ソフトウェアを対象にしたものであること、コストモデルが提案されて以来、開発環境も大きく変化していることから、現在の組み込み型 MP 用ソフトウェアにおいてもこれらのモデルがそのまま適用可能であるかどうか不明で

† 九州芸術工科大学
Kyushu Institute of Design

ある。

本論文では、第2章において、組み込み型 MP 用ソフトウェアのコストモデルを構築するために行ったアンケート調査とその結果として得られたデータの検証について述べる。第3章では、このアンケート調査の結果に基づいて、組み込み型 MP 用ソフトウェアにおけるコストモデルを構築し^{6)~9)}、この新しいコストモデルが COCOMO の BASIC モデルと同様のコストモデルになることを示す。第4章では、このコストモデルに基づいて、組み込み型 MP 用ソフトウェアの開発規模、実績工数、開発期間が CPU ビット、記述言語、製品用途によってそれぞれ異なる特徴を有することを示す。最後に第5章では、アンケート調査と新しく提案したコストモデルから得られる全般的な知見や有用性について述べる。

2. 調査とデータ

2.1 アンケート調査

本研究で行ったアンケート調査の調査項目は表1の通りである。ソフトウェアの生産性を議論するためには生産期間と生産高を明確にする必要がある¹⁰⁾。まず生産期間について検討する。ソフトウェアのライフサイクルは JIS や ISO 等で定義されているばかりでなく¹¹⁾、ほかにも各種の定義が提案されている。しかしながらこれらの定義は類似してはいるが、統一されておらず、いくつかの工程が混然としている組み込み型 MP 用ソフトウェアの開発実態にはあっていない。ここでは表2のようなライフサイクルを考え、このうちのシステム設計から総合テストまでを生産期間とする。

次に生産高を検討する。ここでは注釈行を除いたソースプログラムの行数を生産高として採用する。これは直感的に分かりやすく、またアンケート調査において回答しやすいことを考慮したためである。実際にはすべてのプログラムを新しく開発する場合ばかりでなく、既存のプログラムを再利用する場合もある。そこで完成規模と開発規模は次のように考えるものとして、調査用紙に明確な説明を付けた。

完成規模 = 開発規模 + 流用行数

開発規模 = 新規行数 + 改造行数

このうち流用行数とは既存のプログラムを全く修正せずにそのまま再利用した行数、改造行数とは既存のプログラムを部分的に修正して再利用した行数とする。改造の場合はプログラムの中身を理解する必要がある

表 1 調査項目
Table 1 Questionnaire items.

CPU ビット	用 途	見積もり工数
実績工数	開発期間	ソフトウェアの規模
記述言語	再利用率	開発環境
プロジェクト特性	そ の 他	

表 2 ライフサイクルの定義
Table 2 Definition of life cycle.

工程	内 容
設計	要求分析 + システム設計
製作	プログラム作成
試験	単体テスト + 組み合わせテスト + 総合テスト
保守	保守 + 廃棄

表 3 回答件数
Table 3 The number of answers.

年 度	回答件数
1990	357
1991	490
1992	394
90-92	1241

ので、開発に要する工数は新規開発に準じて考える。

調査用紙は組み込み型 MP 用ソフトウェアを開発している各社（100～200 社、年度による）に発送した。参考として回答の回収件数を表3に示す。ここで、一社から複数のデータを回答してもらっているものがあるために発送数よりも回答件数が多くなっている。

2.2 データと検証

コストモデルを構築するためにはソフトウェアの規模、工数、開発期間のデータが必要である。開発期間は回答をそのまま利用して特に問題はないが、規模と工数は次に述べるように補正が必要である。

ソフトウェアの規模では再利用率が問題になる。図1に1992年度の再利用率（（完成規模－新規開発）/完成規模）の分布を示す。100%に近い方が再利用率が高いことを示している。0%未満のものは、開発はしたが製品に採用しなかった部分が多かったことを示している。既存のソフトウェアを再利用せずにすべて新規に開発したという回答（流用、改造ともに0行のもの）は134件（34%）であり、残りは程度はともかくとして何らかの再利用を行っている。全く新規開発を行わずにすべて既存ソフトウェアの改造と流用で済ませたという回答も16件（4%）ある。これらのことから組み込み型 MP 用ソフトウェアの開発においては

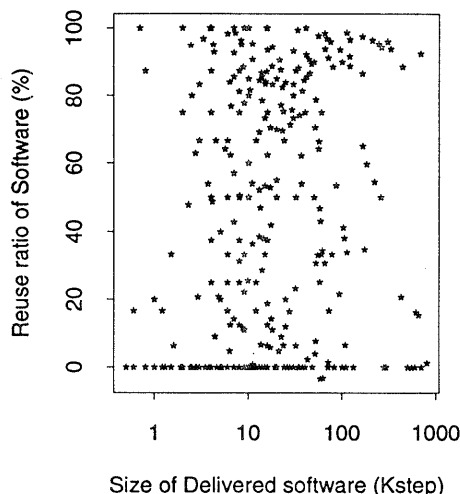


図 1 既存ソフトウェアの再利用率
Fig. 1 Reuse ratio of software.

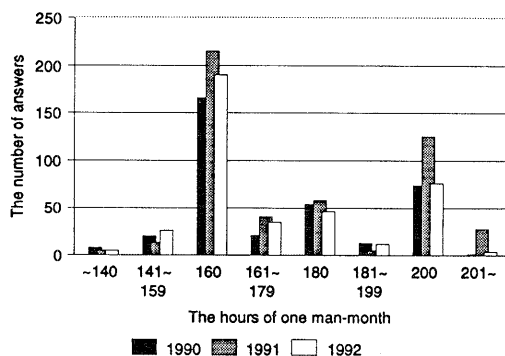


図 2 1人月の時間数の分布
Fig. 2 Distribution of the hours of one man-month.

再利用が活発に行われていることが分かる。したがってソフトウェアの規模として完成規模（出荷行数）を選択すると実績工数に比べて相対的に大規模なソフトウェアを開発していることになるデータが増加し、具合が悪い。ここではソフトウェアの規模は開発規模とする。

次に実績工数について検討する。実績工数は単位を『人月』で回答してもらっているが、この基準が各社ともまちまちである。各社における1人月の時間数の分布の様子を図2に示す。この図から一人当たり160時間の労働時間を1人月として計算しているところが多く、ついで、200時間、180時間が多いことが分かる。ここでは1人月は160時間とみなし、すべての実績工数を換算して使用する。

最後に上述の補正を行った上で、1人月当たりの開

表 4 開発効率の基本統計値

Table 4 Basic statistics of the development efficiency.

年度	平均	標準偏差	最小	最大
1990	1.45	2.95	0.073	40.7
1991	1.96	7.08	0.016	111.1
1992	1.64	4.66	0.008	61.6
90-92	1.71	5.41	0.008	111.1

単位：千行/人月

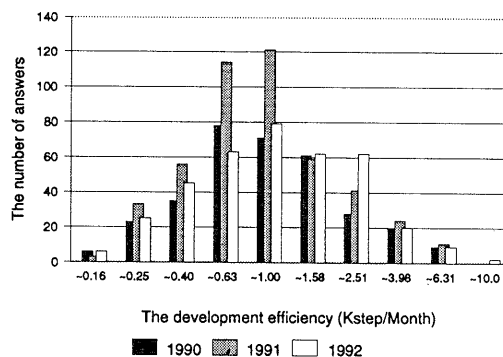


図 3 開発効率の分布
Fig. 3 Distribution of the development efficiency.

発規模（これを開発効率と呼ぶ）を検討する。表4に開発効率の基本統計値を示す。これによると1人月で20行も開発が進まないものや数十キロ行のソフトウェアを開発したことになるものなど、常識的には考えられないような回答がアンケートの回答には含まれていることを示している。そこで開発効率の対数値を適当な区間で棒グラフにして表示すると図3のようになる。なお図3の横軸で、例えば“ ~ 1.00 ”と表示されている区間は、 $0.63 < x \leq 1.00$ ($-0.20 < \log_{10}(x) \leq 0.0$)の区間であることを示している。図3から開発効率の対数値は直感的には正規分布をしているように見える。これがどの程度正規分布に近いかを検定するために、開発効率の対数値を正規 QQ プロット図として表示し、検討する。

ここでは例として1992年度の正規 QQ プロット図を図4に示す。図4から判断して、開発効率の対数値は平均を中心に標準偏差の2倍以内のデータがほぼ正規分布に仕上がっていると考えられる。1990年度と1991年度もほぼ同様である。したがって以後は開発効率の対数値が平均から標準偏差の2倍以内におさまるデータに限定して解析を行う。最終的に選択したデータの開発規模、実績工数、開発期間の基本統計値を表5に示す。表中のS, M, Tについては第3章に

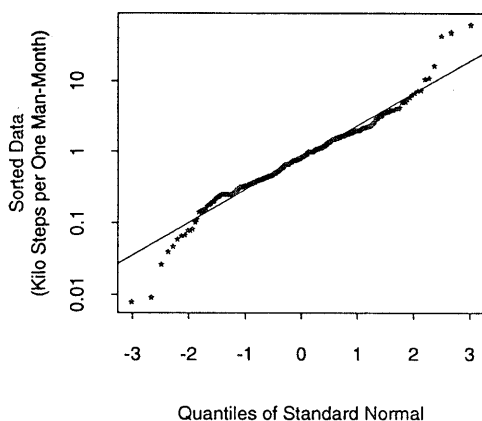


図 4 開発効率の正規 QQ プロット図

Fig. 4 Normal quantile-quantile plot of development efficiency.

表 5 開発規模, 実績工数, 開発期間の基本統計量
Table 5 Basic statistics on the size of software, the actual man-months, and the development period.

分 類	年 度	件 数	平 均	標 準 偏 差	最 小	最 大
開発規模 (S)	90	331	11.5	22.1	0.1	200
	91	462	18.5	41.0	0.2	390
	92	373	25.5	69.8	0.1	700
実績工数 (M)	90	331	15.0	33.9	0.1	467
	91	462	26.2	66.2	0.5	1013
	92	373	46.7	253.9	0.2	3096
開発期間 (T)	90	331	6.87	5.09	0.5	24
	91	462	7.39	6.12	0.5	36
	92	373	7.16	5.64	0.2	32

単位: 開発規模 (S)=Ksteps, 実績工数 (M)=人月, 開発期間 (T)=月

において改めて述べる。表 3 と表 5 の回答データ数を比較すると、開発効率の値に基づいて排除されたデータ数は 90 年度が 7%, 91 年度が 6%, 92 年度が 5% になっていることが分かる。

3. JEIDA モデルの構築

例として 1992 年度における開発規模と実績工数をそれぞれ対数軸で散布図として表示したものを図 5 に示す。同様に 1992 年度におけるソフトウェア開発の実績工数と開発期間をそれぞれ対数軸で散布図として表示したものを図 6 に示す。1990 年度と 1991 年度のデータにおいてもほぼ同様の図になるのでここでは省略する。これらの図において、“+”で示している点が開発効率の値に基づいて解析から除外したデータである。

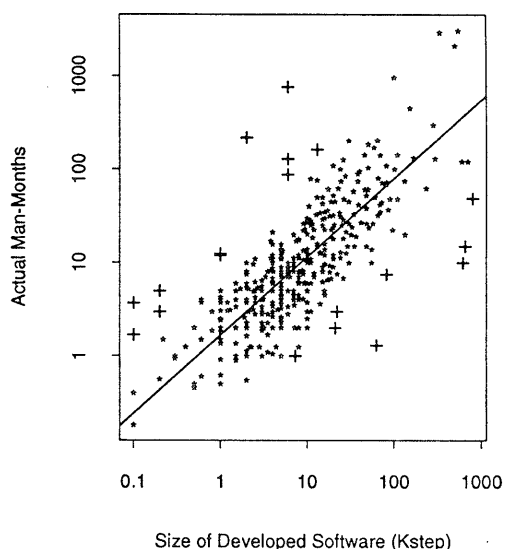


図 5 規模と実績工数の散布図 (1992)

Fig. 5 Scatter diagram between the size of software and the actual man-months.

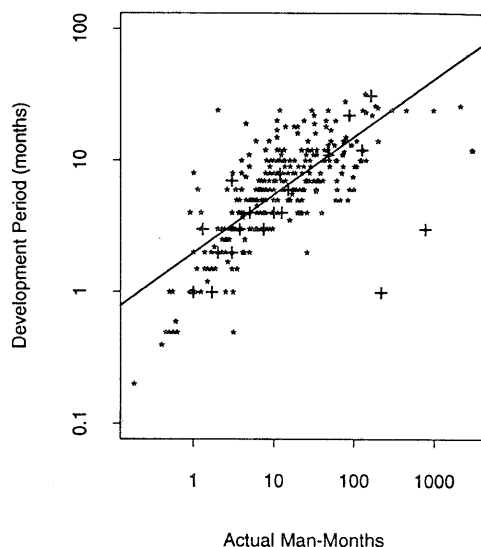


図 6 実績工数と開発期間の散布図 (1992)

Fig. 6 Scatter diagram between the actual man-months and the development period.

図 5 と図 6 を見ると、それぞれ二つの変数間には強い関連があることが分かる。したがってこれらの図からそれぞれ 1 次回帰直線の係数を求めることで、組み込み型 MP 用ソフトウェアのコストモデルを次のような式として表せることが分かる。

表 6 JEIDA モデルにおける係数
Table 6 The coefficients of JEIDA model.

年度	件数	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	DE	AM
90	331	1.95	0.76	2.44	0.41	1.05	1.85
91	462	1.92	0.82	1.71	0.51	1.00	2.62
92	373	1.66	0.84	1.98	0.44	1.18	4.26
90-92	1167	1.86	0.81	1.99	0.46	1.06	2.89

DE: 開発効率, AM: 平均開発人員

表 7 COCOMO の BASIC モデルにおける係数²⁾
Table 7 The coefficients of the Basic COCOMO model.

モード	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
ORGANIC	2.4	1.05	2.5	0.38
SEMIDETACHED	3.0	1.12	2.5	0.35
EMBEDDED	3.6	1.20	2.5	0.32

$$M = a \times S^b$$

$$T = c \times M^d$$

ここで, S は開発規模 (Kstep), M は開発工数 (人月), T は開発期間, a, b, c , および d はプロジェクトの困難度で決まる生産性に関連した係数である. このモデルは次に述べるように COCOMO の BASIC モデルと同様のモデル式になっていることが分かる. ここでは組み込み型マイクロプロセッサという特定の分野におけるこのコストモデルを JEIDA モデルと呼ぶことにする (JEIDA とは日本電子工業振興協会の英字名称 Japan Electronic Industry Development Association の略称). JEIDA モデルの係数 a, b, c, d , 開発効率 (DE), および平均開発人員 (AM) を表 6 に示す.

ここでは JEIDA モデルと COCOMO の BASIC モデル (表 7 参照) を比較して議論する. 表 6 の 1990 ~ 1992 年の各年度の係数と表 7 の ORGANIC モデル (係数の値が JEIDA モデルの対応する値に一番近い) の係数を元に, 図 7 と図 8 にそれぞれ開発規模と実績工数, 実績工数と開発期間の関係を示す. これらの値は CPU ビット, 記述言語などによって変動するが, 詳細は第 4 章で述べるので, ここでは詳しい議論をしない. なお表 5 の値から以後の図の横軸の右端を 300 Ksteps と 300 人月としている.

図 7 から, 数十キロ行以上のソフトウェア開発において, COCOMO モデルでは JEIDA モデルに比べて多大な実績工数を必要としていることが分かる. これは JEIDA モデルが比較的の小規模のソフトウェアを対象としており COCOMO モデルが大規模のソフト

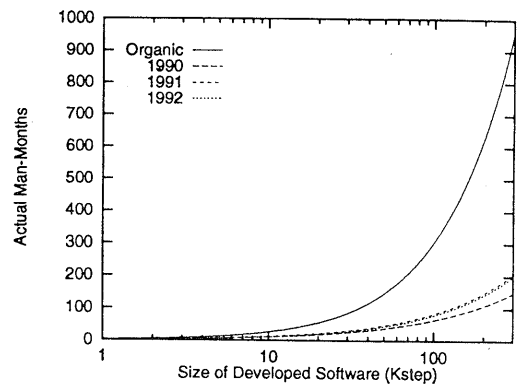


図 7 ソフトウェアの規模と実績工数の関係

Fig. 7 Relation between the size of software and the actual man-months.

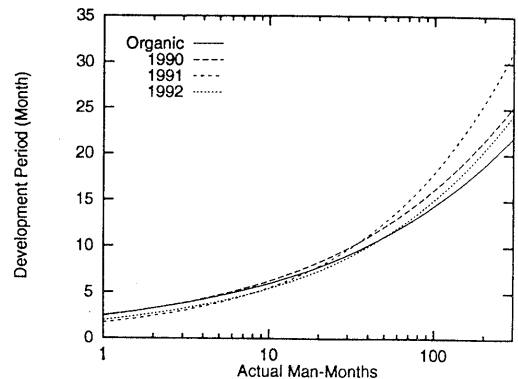


図 8 実績工数と開発期間の関係

Fig. 8 Relation between the actual man-months and the development period.

ウェアを対象としているというだけでなく, 二つのモデルがそれぞれ構築された時代におけるソフトウェアの開発技術や環境の違いも影響していると考えて良さそうである. 一方, 実績工数と開発期間の関係は図 8 から JEIDA モデルと COCOMO モデルにあまり大きな差がないことが分かる. これは製品の納期や投入可能な人員といった制約から, 必要な工数と開発期間との関係が両方のコストモデルにおいて類似の経営判断や工程管理の元で開発されていることを示唆している.

表 6 には開発効率 (DE) と平均開発人員 (AM) も示している. ここで開発効率の値は 1.0 を少し超える程度であることから 1 人月で開発できるソフトウェアの大きさは平均千行を少し越える程度であることが分かる. 一方, 平均開発人員はソフトウェアの開発期間を通じてプロジェクトにかかわった平均の要員数を表

しており、この値は90年度の1.85人から、91年度の2.62人、92年度の4.26人と毎年増加している。表5から開発規模や実績工数の平均値は平均開発人員と同様に毎年増加しているにもかかわらず、開発期間は毎年約7か月で変わっていないことから、規模や工数の増大に人員を投入して対応している様子がうかがえる。

4. JEIDA モデルの詳細

JEIDA モデルについて、CPU ビット別、記述言語別、製品の分野別に分類した場合の特徴について検討する。CPU ビット別のコストモデルの係数 $a \sim d$ 、開発効率 (DE)、平均開発人員 (AM) を表8に、係数 $a \sim d$ を元に開発規模と実績工数、実績工数と開発期間の関係を図9と図10に示す。記述言語別のコストモデルの係数 $a \sim d$ 、開発効率 (DE)、平均開発人員 (AM) を表9に、係数 $a \sim d$ を元に開発規模と実績工数、実績工数と開発期間の関係を図11と図12に示す。なお、表8と表9には90～92年度の値を示しているが、年度によっては係数の値が多少変動しているものもあるので、3年間全体のデータをまとめて解析した結果を併せて示している。この場合に開発効率によるデータ選択の基準が変動するので、データ数が各単年度の合計には必ずしもならない。製品用途別のコストモデルの係数 $a \sim d$ を元に開発規模と実績工

表8 CPU ビット別コストモデル
Table 8 Cost model in CPU bits.

分類	年度	件数	a	b	c	d	DE	AM
4 bit	90	77	2.16	0.60	1.55	0.61	0.89	1.25
	91	69	2.46	0.56	1.35	0.63	0.79	1.55
	92	69	1.79	0.58	1.22	0.77	1.05	1.14
	90-92	216	2.15	0.58	1.32	0.67	0.88	1.33
8 bit	90	130	2.46	0.61	1.77	0.56	1.07	1.60
	91	165	2.04	0.73	1.54	0.58	1.10	1.89
	92	124	1.64	0.81	1.63	0.56	1.16	1.95
	90-92	419	2.06	0.71	1.63	0.57	1.11	1.82
16 bit	90	101	1.67	0.85	3.98	0.24	1.12	2.42
	91	167	1.60	0.90	1.93	0.47	1.04	3.23
	92	125	2.14	0.78	2.92	0.31	1.12	3.87
	90-92	394	1.77	0.85	2.70	0.36	1.10	3.12
32 bit	90	23	1.52	0.89	5.53	0.22	1.17	2.79
	91	61	2.33	0.83	2.29	0.40	0.86	4.14
	92	55	1.37	0.91	3.69	0.25	1.49	14.28
	90-92	138	1.76	0.88	3.55	0.28	1.12	7.96

DE: 開発効率, AM: 平均開発人員

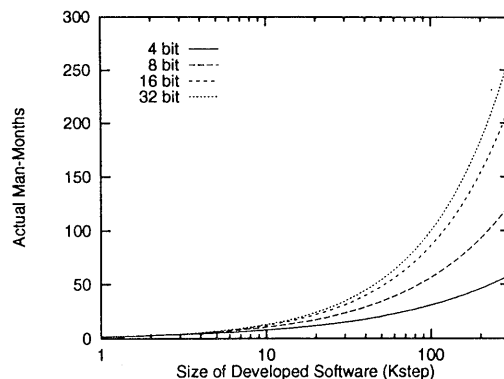


図9 CPU ビット別ソフトウェアの規模と実績工数の関係
Fig. 9 Relation between the size of software and the actual man-months in CPU bit.

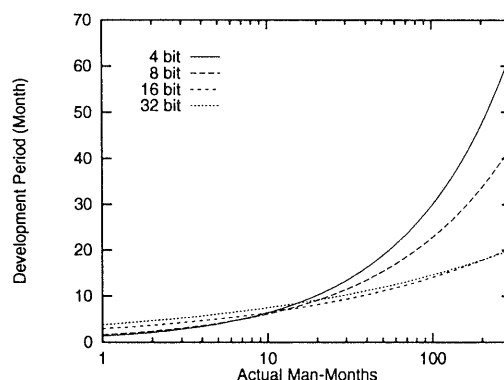


図10 CPU ビット別実績工数と開発期間の関係
Fig. 10 Relation between the actual man-months and the development period in CPU bit.

数、実績工数と開発期間の関係を図13と図14に示す。なお、図13と図14では線種を分かりやすくするために、図の右端 (300 Ksteps) における値の大きさの順に線種の説明を付けている。

図9、図10および表8からCPUビットによってそれぞれ開発規模と実績工数、実績工数と開発期間が異なる特徴を示していることが分かる。図9から同じ開発規模のソフトウェアならCPUのビット数が増えるに従って工数が多くなる事が分かる。一方、図10から実績工数と開発期間は4ビットと8ビットは16ビット、32ビットとは異なる関係を示しているが、16ビットと32ビットはほとんど同じ関係を有することが分かる。これは32ビットになっても16ビットの時と同様な開発を行っており、特に32ビットCPUの特徴を考慮した開発を行うには至っていないことを暗示している。また表8から開発効率

表 9 言語別のコストモデル
Table 9 Cost model in languages.

分類	年度	件数	a	b	c	d	DE	AM
ASM	90	255	1.96	0.72	2.35	0.44	1.06	1.47
	91	259	2.13	0.73	1.54	0.57	1.02	1.91
	92	218	1.73	0.81	1.73	0.52	1.07	3.01
	90-92	735	1.97	0.75	1.84	0.51	1.05	2.08
C	90	61	2.50	0.75	2.39	0.40	0.96	3.12
	91	109	1.79	0.89	1.79	0.47	0.91	3.32
	92	76	1.59	0.86	2.50	0.34	1.35	7.38
	90-92	244	1.91	0.84	2.16	0.41	1.04	4.38
C+ASM	90	—	—	—	—	—	—	—
	91	79	1.57	0.91	2.47	0.41	1.07	3.72
	92	60	1.54	0.84	1.82	0.47	1.35	3.51
	90-92	139	1.56	0.88	2.15	0.43	1.19	3.63
その他	90	15	1.69	0.83	3.92	0.27	1.21	3.26
	91	15	1.64	0.84	1.35	0.46	0.97	3.94
	92	19	1.48	0.95	3.70	0.24	1.14	8.59
	90-92	49	1.59	0.89	2.50	0.34	1.11	5.53

DE: 開発効率, AM: 平均開発人員,
ASM: アセンブリ言語, C: C言語,
C+ASM: C言語とアセンブリ言語の併用

(DE) は CPU によってはあまり変動しないこと, 平均開発人員 (AM) は CPU ビットが大きくなるにつれて大きくなることが分かる。

図 11, 図 12, および表 9 から記述言語による開発規模と実績工数, 実績工数と開発期間の関係はそれぞれ異なる特徴を示すことが分かる。図 11 からアセンブリ言語によるものと C 言語やその他の高水準言語を主体にして記述するものとで明確に異なる特徴を示しており, アセンブリ言語の方が同じ開発規模に対しては実績工数が少なく済んでいることが分かる。これは一見常識に反する結果のように見えるが, 機能的にはアセンブリ言語の数行が C 言語の一行に相当することから特に異常な結果ではないと考えられる。一方, 図 12 から実績工数と開発期間は記述言語が異なるとそれぞれ異なる関係を示すことが分かる。およそ 30 人月以上では同じ実績工数に対してはアセンブリ言語で記述するものが開発期間が長くなっていることが分かる。これは表 9 から平均開発人員 (AM) の割当が他の記述言語に比べて半分程度と少ないことから, 必要な工数が大きくなると必然的に開発期間が長くなると考えて良さそうである。

さらに製品分野によって開発規模と実績工数, 実績工数と開発期間の関係が異なるか否かについても調べた。結果は図 13 と図 14 に示し, 係数の詳細は

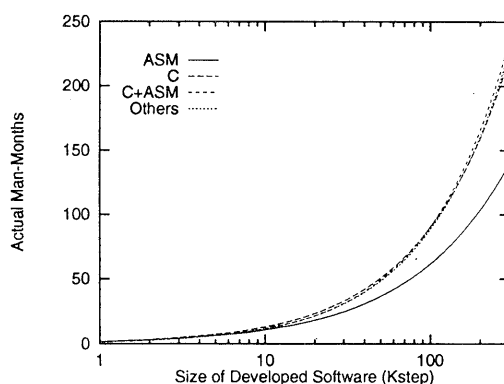


図 11 記述言語別ソフトウェアの規模と実績工数の関係
Fig. 11 Relation between the size of software and the actual man-months in languages.

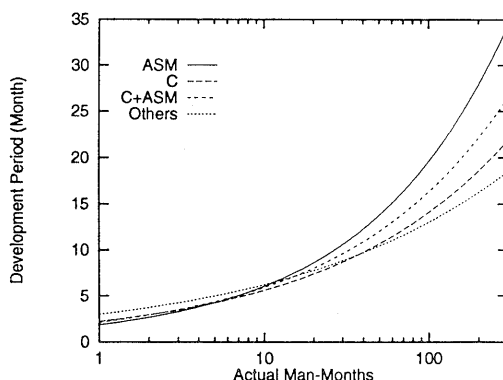


図 12 記述言語別実績工数と開発期間の関係
Fig. 12 Relation between the actual man-months and the development period in languages.

省略する。図 13 から OA 関連 (OA), 通信関連 (Comm), 周辺端末 (Peri), 家電機器 (Home) の四分野は規模と工数の関係がそれぞれ異なる特徴を示しているが, 制御関連 (Proc), 計測機器 (Meas), その他 (Others) の三分野はほぼ同じ特徴を示していることが分かる。一方, 図 14 から家電機器 (Home) と計測機器 (Meas) を除いた他の五分野はほぼ同じ特徴を示しており, 開発規模と実績工数の関係はほぼばらばらになっていない点が興味深い。また開発効率 (DE) に大きな変動はないが, 平均開発人員 (AM) は, 3 年間の平均で家電機器の 1.52 人から OA 機器の 6.14 人まで, 製品分野によって大きく変動していることが分かる。

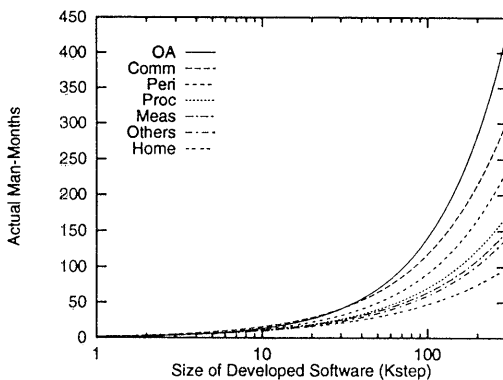


図 13 製品分野別ソフトウェアの規模と実績工数の関係
Fig. 13 Relation between the size of software and the actual man-months in product purposes.

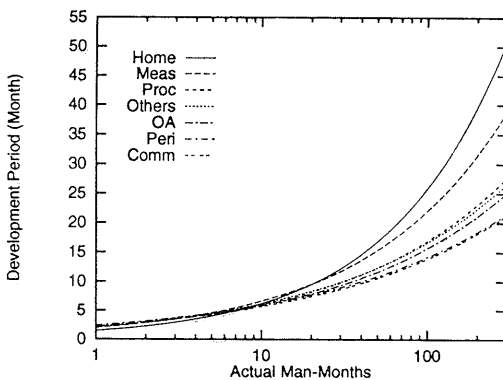


図 14 製品分野別実績工数と開発期間の関係
Fig. 14 Relation between the actual man-months and the development period in product purposes.

5. おわりに

これまで組み込み型 MP 用ソフトウェアにおけるコストモデルについては十分に検討されていなかった。本研究で JEIDA モデルと名付けた新しいコストモデルを構築し、提案することができた。このコストモデルはこれまで文献^{12)~16)}で年度ごとに報告してきたが、従来は解析時間や解析手段の都合でデータの検証が十分でないなど、信頼性に多少の不安があった。今回はそれぞれの調査年度のデータを十分に吟味・検討した上で統計解析パッケージ S を利用して解析を行ったこと、複数の年度を総合して考察していること、JEIDA モデルの係数が調査年度によらず比較的安定した値を示していることから、このモデルは十分に信頼できるものである。

今回、提案した JEIDA モデルによって、組み込み型 MP 用ソフトウェアの開発においてもコストモデルの概念を適用できること、JEIDA モデルは基本的には COCOMO の BASIC モデルと同様のモデルであるが、従来の大規模ソフトウェアの開発を対象として提案されたコストモデルとは異なる特徴を示すこと、CPU ビット、記述言語、製品用途によって規模、工数、開発期間はそれぞれ異なる特徴を示すことを明らかにすることができた。これらの特徴は従来は直観的に理解されてはいたが、今回初めて客観的かつ定量的に示すことができた。

現場のソフトウェア開発者は現在自分たちが関わっているソフトウェア開発の生産性を定量的に評価するための尺度を切実に必要としている。今回本稿で提供した JEIDA モデルと各種の係数値は、個々の組織で入手することが困難で、特定の開発現場に偏らない多数のソフトウェア開発のデータに基づいており、ソフトウェアの開発中や開発後にその開発規模から実績工数や開発期間が適切であったか否かを判断する評価基準として活用することができる。さらに例えば、ソフトウェアの開発前に開発規模を見積ることができれば実績工数や開発期間を推定することができるといった応用も可能である。

今回のコストモデルの提案によって、組み込み型 MP 用ソフトウェアの生産性を定量的に議論するための基盤を確立できた。この JEIDA モデルの信頼性を上げるには、今後も継続して調査を行うことが重要であろう。またモデル式の係数の経年変化を把握することも重要であると考えており、今後も継続して調査を行う予定である。一方、ここで報告した CPU タイプ、記述言語、製品分野の違いに加えて、開発環境やプロジェクトの特性についても調査を行っている。これらはまだパラメータとして組み込んでいないが、これらをパラメータとして組み込んだより精密なモデル式については改めて報告したい。

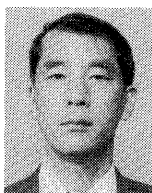
謝辞 本研究の一部は文部省科学研究費重点領域研究(1)(課題番号 02249110, 03235110, 04219110)によります。また日本電子工業振興協会(電子協と略す)のソフトウェア生産性尺度専門委員会(1990~1992)との共同研究でもあります^{12)~16)}。データの利用を快諾して頂いた電子協に感謝します。本論文をまとめるに当たって、九州大学工学部牛島和夫教授から有益な助言を頂きました。ここに記して関係者の方々に謝意を表します。

参 考 文 献

- 1) 花田收悦: ソフトウェア生産性の評価と管理, 情報処理, Vol. 21, No. 10, pp. 1057-1064 (1980).
- 2) Boehm, B. W.: Software Engineering Economics, *IEEE Trans. Softw. Eng.*, Vol. SE-10, No. 1, pp. 4-21 (1984).
- 3) Boehm, B. W.: *Software Engineering Economics*, Prentice-Hall (1981).
- 4) Putnam, L. H. and Fitzsimmons, A.: Estimating Software Cost, *Datamation*, pp. 189-198 (Sep.), pp. 171-178 (Oct.) and pp. 137-140 (Nov. 1979).
- 5) 大筆 豊: ソフトウェアのコストの見積り技術, 情報処理, Vol. 33, No. 8, pp. 906-911 (1992).
- 6) 藤村直美: 組み込み型マイクロプロセッサ用ソフトウェア生産の現状, 第 39 回情報処理学会全国大会論文集, pp. 1421-1422 (1989).
- 7) 藤村直美: 組み込み型マイクロプロセッサ用ソフトウェア開発の状況と生産性モデル式, 第 41 回情報処理学会全国大会論文集, No. 5, pp. 249-250 (1990).
- 8) Fujimura, N.: Software Productivity in Built-in Microprocessors, *Microprocessing and Microprogramming*, Vol. 28, Nos. 1-5, pp. 169-172 (Feb. 1989).
- 9) 藤村直美: 組み込み型マイクロプロセッサ用ソフトウェア開発状況と生産性, 第 46 回情報処理学会全国大会論文集, No. 5, pp. 255-256 (1993).
- 10) 上條史彦: ソフトウェアのコストについて, 情報処理, Vol. 21, No. 10, pp. 1050-1056 (1980).
- 11) 菅 忠義: ソフトウェア工学における標準化動向, 情報処理, Vol. 28, No. 9, pp. 1113-1126 (1987).
- 12) 電子協: マイクロコンピュータに関する調査報告書 (Ⅱ), pp. 1-106 (1989).
- 13) 電子協: マイコン技術フォーラム Proceeding, pp. 171-201 (1990).
- 14) 電子協: ME 知的ソフトウェア環境に関する調査報告書, pp. 167-240 (1991).
- 15) 電子協: ME 知的ソフトウェア環境に関する調査報告書, pp. 95-167 (1992).
- 16) 電子協: ME 知的ソフトウェア環境に関する調査報告書, pp. 273-355 (1993).

(平成 6 年 4 月 7 日受付)

(平成 6 年 9 月 6 日採録)



藤村 直美 (正会員)

1950 年生まれ。1973 年九州大学工学部電子工学科卒業, 1978 年九大大学院博士課程通信工学専攻単位取得退学, 同年九大工学部助手, 1981 年九大情報処理教育センター助教授, 1988 年九州芸術工科大学助教授。工学博士。計算機システムの性能の計測・評価, ソフトウェアの性能・品質・生産性の計測・評価, 広域計算機ネットワークに関心がある。電子情報通信学会, ソフトウェア科学会, ACM, IEEE, ISOC などの各会員。