

ソフトウェア開発プロジェクトの遂行に影響を与える要因の分析

古山 恒夫[†] 菊地 奈穂美^{††}
安田 守^{†††} 鶴保 征城^{††}

独立行政法人情報処理推進機構 (IPA) ソフトウェア・エンジニアリング・センター (SEC) では、国内のソフトウェアベンダのエンタプライズ系ソフトウェア開発に関するプロジェクトデータを収集している。このデータに含まれる、コスト・納期・品質それぞれに関するプロジェクト遂行の成否を示すデータに着目し、それらと規模や工数などの量的データやさまざまなプロジェクト特性を示す質的データとの関係を分析することにより、次のようなプロジェクトを計画どおりに遂行できなかった要因、すなわちコスト超過・納期遅延・品質低下を起こす要因を明らかにした。(1) プロジェクト規模そのものが大きく、特に工期あたりの規模が大きいプロジェクトはコスト超過を起こす割合が高く、(2) 要求仕様があいまいなプロジェクトは納期遅延や品質低下を起こす割合が高い。一方、(3) 事前に工期の妥当性を評価したプロジェクトでは納期遅延を起こす割合が低く、(4) 業務分野の経験者を揃えたプロジェクトでは納期遅延や品質低下を起こす割合が低く、(5) テスト体制を整備したプロジェクトでは、コスト超過・納期遅延・品質低下を起こす割合が低い。

Analysis of the Factors that Affect the Performance of Software Projects

TSUNEO FURUYAMA,[†] NAHOMI KIKUCHI,^{††} MAMORU YASUDA^{†††}
and SEISHIRO TSURUHO^{††}

The Software Engineering Center of the Information-technology Promotion Agency, Japan, has been collecting project data measured at organizations during developing enterprise software across Japan. The project data has quantitative data items such as software size and qualitative data items such as the skill of the project manager. It also includes performance results of software projects from the viewpoint of cost, delivery, and quality compared to their expected levels in the initial plan. We report the analysis results of clarifying factors that cause cost overruns, excessive time for delivery, and lower quality than planned in enterprise software projects. The analysis shows the following results. (1) The possibility of cost overrun depends significantly on the project size, in particular, the size of the software to be developed per period of development. (2) Ambiguous requirement specifications exacerbate problems with excessive time for delivery and lower quality than planned. (3) An evaluation of the project plans increases the likelihood of making the delivery time. (4) Projects that use developers who have experience with the target business areas make it less likely that excessive time to delivery or lower quality than planned will occur. (5) Projects that have test teams of high ability can improve the performance on their projects.

1. はじめに

ソフトウェア開発における顧客・経営者・プロジェクト管理者 (以下 PM) 共通の目標は、開発した品質

の高いソフトウェアを安いコストで予定の納期までに開発することである。それを達成するために PM は、プロジェクトの開始に先立って開発計画を作成し、その作業の中で品質、コスト、工期の見積りを行うのが普通である。

見積もり方法としては、規模から工数や工期を見積もる COCOMO¹⁾ や COCOMO II²⁾ などがよく知られている。それ以外にもニューラルネットワークを利用した見積もり方法³⁾、サイズの理論に基づいたモデル^{4),5)}、プロジェクト間の類似度を定義して類似なプロジェクトから工数を見積もる方法⁶⁾⁻⁸⁾、ソフト

[†] 東海大学理学部

School of Science, Tokai University

^{††} 独立行政法人情報処理推進機構ソフトウェアエンジニアリングセンター

Software Engineering Center, Information-technology Promotion Agency

^{†††} 株式会社野村総合研究所

Nomura Research Institute, Ltd.

ウェアプロジェクトデータのように欠損値が多い場合にも適用できる協調フィルタリングの手法を用いた見積もり方法⁸⁾などが提案されている。

工数の簡便な見積もり方法として、規模に生産性を掛けるものもある。生産性はプロジェクトの能力を表す代表的な指標の1つであり、顧客・経営者・プロジェクト管理者の大きな関心事である。生産性に影響を与える要因についてもこれまでに多くの分析結果が報告されている^{9)~12)}。

このように多くの見積もり方法が提案されているにもかかわらず、ソフトウェアプロジェクトでは、当初計画の目標どおり遂行できないことが少なくない。その主な理由の1つとして、ソフトウェア開発プロセスが、物理的現象を利用してものを製造する化学プラントなどと異なり、人間の知識・能力・組織力に依存しているため、ソフトウェア開発プロジェクトを取り巻く数多くの不確定要因による影響を受けやすいことが考えられる。

それではどのようなプロジェクトが成功し、どのようなプロジェクトは失敗するのであろうか。これまでもさまざまな要因が経験則として語られてきている¹³⁾。それらをデータによって科学的に裏付けることができれば、またデータを分析して新たな知見を示すことができれば、ソフトウェアプロジェクトの開発計画作成あるいはプロジェクト運営上の大きな助けとなるであろう。

ソフトウェア開発プロジェクトの収集データとして有名なものは、オーストラリアが中心となって収集している ISBSG (International Software Benchmarking Standards Group) のデータ¹⁴⁾がある。このデータには、米国、日本、フィンランド、カナダ、オランダなどからの 150 項目 4,000 プロジェクトのデータが含まれている。また、米国のコンサルタント会社 SPR 社は 8,000 以上のプロジェクトデータを収集し、それをさまざまな角度から分析し、書籍として出版している¹⁵⁾。

一方、わが国では独立行政法人情報処理推進機構 (IPA) に 2004 年度秋に設立されたソフトウェア・エンジニアリング・センター (SEC) が国内のソフトウェア開発企業の協力のもとにエンタプライズ系ソフトウェア開発のプロジェクトデータを収集し始めている。2005 年度に収集したデータ数は 1,419 件に達している¹⁶⁾。

SEC データには、規模・工数・工期・バグ数などの最も基本的なプロジェクトデータ項目 (量的データ) に加えて、プロジェクト特性を示すさまざまなデータ項

目 (質的データ) が含まれている。2005 年度に収集した SEC のデータ定義 Ver.2.0¹⁶⁾ には、プロジェクト特性を示すデータ項目として、プロジェクト遂行の成否を示すもの、すなわち品質・コスト・納期 (QCD: Quality, Cost, Delivery の頭文字) に関して各プロジェクトが計画どおりにプロジェクトを遂行することができたかどうかをプロジェクト自身が判断した結果を示す「実績の評価」というデータ項目も含まれている。このようなデータ項目は ISBSG のデータにも見られず、筆者らの調査の限りでは新しいと考える。

プロジェクト遂行の成否については、量的データや質的データの数値から類推することもできるが、各プロジェクトの判断結果は有力な情報となる。そこで、筆者らは「実績の評価」を目的変数とし、規模、工数、工期などの量的データ、およびさまざまなプロジェクト特性を表す質的データを説明変数とする統計分析を行うことにより、プロジェクトの遂行に及ぼす影響要因を明らかにすることを試みた。

分析方法は量的データを説明変数とする場合は、母集団の平均値の差の有意性を検定する t 検定を用いた。質的変数を説明変数とする場合は、 2×2 のクロス集計表を作成して、目的変数と説明変数が独立であるか否かを χ^2 検定あるいは Fisher の直接計算法を用いて検定した。その結果、コスト超過、納期遅延、品質低下などのプロジェクトの失敗に関係するさまざまな要因を明らかにすることができた。ここで得られた結果は、これまでにいわれてきた経験則と矛盾するものではなく、いわば経験則を信頼性の高いデータとそれに対する統計的分析で実証したものであることができる。

2 章で分析対象データの概要を、3 章で分析方法を紹介する。4 章で分析結果すなわちプロジェクト遂行の成否と関連の見られる要因の候補を示す。5 章で要因候補とプロジェクト遂行の成否について考察を加える。

2. 分析対象データの概要

分析対象データは、2005 年度に SEC で収集したエンタプライズ系ソフトウェアプロジェクトのデータ 1,419 件である¹⁶⁾。新規開発プロジェクトのデータが改修・保守プロジェクトなどのデータに比べてばらつきが少なく分析結果も安定していることから、1,419 件のうちの新規開発プロジェクトのデータ 846 件のみを分析対象とした。データ定義は参考文献 16) のものを使用している。

本論文における「品質」は JIS X0133 (ISO/IEC 9126) で定義されている品質特性の「信頼性」に相当する。

2.1 目的変数

目的変数は、プロジェクトの遂行実績に関する、① 実績の評価（コスト）、② 実績の評価（工期）、③ 実績の評価（品質）の3つである。これらの項目に回答があったプロジェクト数は、① が217件、② が215件、③ が98件である。① と② の共通集合のデータは214件でほとんど重なっている。③ のデータはすべてその共通集合に含まれている。各目的変数に対する回答内容とプロジェクト数を表1に示す。ただし、各目的変数とも5つの選択肢を2つに集約している。

目的変数間の関係の有無についてのP値（2つの母集団の平均が等しいなどのいわゆる帰無仮説が成立する可能性を表す確率で、小さいほど帰無仮説が否定される可能性が高い）と量的データにおける相関係数に相当するクラメールのV値を表2に示す。実績の評価における工期と品質のV値は0.41であり多少の関連が見られるが、コストと工期およびコストと品質の間のV値はいずれも0.3よりも小さく、全体としてお互いの関連は必ずしも高くない。したがって、それぞれを別々の目的変数として分析することにする。

2.2 説明変数

2.2.1 量的データ

最初に、ソフトウェアプロジェクトを評価する際に最も重要な要素である① 規模を選ぶ。規模はFPベースのものとする。次に、実績の評価（コスト）に対応

表1 目的変数と該当プロジェクト数

Table 1 Target variables and number of projects for each variable.

目的変数	内容	プロジェクト数
実績の評価（コスト）	コスト超過なし	180 (83%)
	コスト超過あり	37 (17%)
	合計	217
実績の評価（工期）	納期遅延なし	171 (80%)
	納期遅延あり	44 (20%)
	合計	215
実績の評価（品質）	品質低下なし	69 (70%)
	品質低下あり	29 (30%)
	合計	98

表2 実績の評価の相関

Table 2 Correlations of performance evaluation.

	コスト	工期	品質
コスト	—	1.7E-4	0.036
工期	0.257	—	4.8E-5
品質	0.212	0.411	—

(注) 右上はP値、左下はクラメールのV値

2つの質的変数の関連を表す指標で0から1の値をとる。値が0に近づくほど質的変数間の関連が弱く、1に近づくほど関連が強いことを表す。

する② 工数、実績の評価（工期）に対応する③ 工期、実績の評価（品質）と関連が深いと考えられる④ 開発時のバグ数を選ぶ。開発時のバグ数（以下では「バグ数」と省略）としては、回答数の多い「検出バグ現象数結合テスト」と「検出バグ現象数総合テスト（ベンダ確認）」の和を用いる。最後に工数、工期、バグ数とも規模と高い相関を持つことから、⑤ 規模あたりの工数（生産性）、⑥ 工期あたりの規模、⑦ 規模あたりのバグ数をとる。

2.2.2 質的データ

説明変数として取りあげた質的データを表3に示す。表3は、SECの収集データのうちから回答数の少ないデータ項目を除いたものであり、次のような項目が含まれている。分類は参考文献16)に基づいている。

- 開発プロジェクト全般：新技術の利用や作業スペースなど5項目
- 計画の評価：QCDそれぞれに関して計画の妥当性を事前に評価したか否かという項目（3項目）
- 利用局面：情報通信業や金融・保険業などの業種（大分類）（1項目）

表3 説明変数（質的データ）

Table 3 Predictor variables (categorical data).

分類	データ項目（*1）
開発プロジェクト全般	新技術利用/役割分担_責任所在/達成目標_優先度_明確度合い
	作業スペース/プロジェクト環境_騒音
	計画の評価
計画の評価	計画の評価（コスト/工期/品質）
利用局面	業種（大分類）
システム特性	アーキテクチャ
	主開発言語
開発の進め方	類似プロジェクト_有無
	ツール利用（プロジェクト管理ツール/構成管理ツール/設計支援ツール/ドキュメント作成ツール/デバッグ_テストツール/CASEツール/コードジェネレータ）
	開発方法論利用
ユーザ要求管理	要求仕様_明確度合い
	ユーザ担当者（要求仕様関与/受け入れ試験関与/システム経験/業務経験/設計内容理解度）
	ユーザとの役割分担_責任所在_明確度合い
	ユーザ承認有無（要求仕様/設計）
	要求レベル（信頼性/使用性/性能_効率性/保守性/移植性/ランニングコスト要求/セキュリティ）
	法的規制有無
	要員等スキル
要員スキル（業務分野経験/分析_設計経験/言語_ツール利用経験/開発プラットフォーム使用経験）	
テスト体制	定量的出荷品質基準_有無

(*1)「/」が項目の区切りを表す。

- システム特性：アーキテクチャ（2 階層クライアントサーバ/3 階層クライアントサーバなど）と主開発言語の 2 項目
- 開発の進め方：ツールの利用の有無など 9 項目
- ユーザ要求管理：ユーザの関与度合いやユーザの要求条件に関わる 17 項目
- 要員等スキル：PM や要員のさまざまなスキルレベルおよびテスト体制など（7 項目）

3. データ分析方法と有意性判定基準

3.1 説明変数が量的データの場合の分析方法

2.2.1 項で述べた 7 つの説明変数それぞれに対して、得られたデータのヒストグラムを描くといずれも左右対称にはならず、ヒストグラムの右側すなわち横軸の値の大きな側が裾を伸ばしたように長くなる。一方、それを対数変換したものは一般に正規分布の仮説が否定されない。このことから、7 つの説明変数に対してはすべて対数変換してから分析を進める。次に説明変数ごとに等分散の検定を行い、等分散の仮説が否定されない場合には等分散を仮定した t 検定を、等分散の仮説が否定される場合には等分散を仮定しない t 検定により有意性を判断するための P 値を求める。なお、欠損データは補正せず、該当するデータ項目の分析に対してのみ除外する。

分析結果は、P 値をもとに、① 危険率 1%で有意、② 危険率 5%で有意のほかに、③ 危険率 10%で有意の場合も考慮する。これはソフトウェアプロジェクトデータのように、多くの要因の影響を受けるデータでは、危険率 1%や 5%で有意な結果だけを採用するとほとんどの説明変数が「影響を与えていない」という結果になることが多いためである。P 値が 10%以下のものについては、その説明変数が目的変数に与える影響度合いを示す寄与率を計算する。

3.2 説明変数が質的データの場合の分析方法

各説明変数に対する回答数は必ずしも多くはないので、目的変数だけでなく説明変数も 2 つのレベルに集約して 2×2 のクロス集計表を作成して分析する。クロス集計表は χ^2 分布を仮定して χ^2 検定を行う。しかし、欄の中に極端に少ない数が含まれている場合は、 χ^2 分布による近似のための条件が満たされていないため、Fisher の直接計算法を用いる。ここでは統計学的な経験則に基づいて 4 つのすべての欄が 5 以上であるときは χ^2 検定を行い、少なくとも 1 つの欄が 4 以下のときは Fisher の直接計算法を用いる。質的データの場合も量的データと同様に危険率 1%、5%、10%を有意性の判定基準とする。

4. 分析結果

分析の結果得られた、QCD いずれかの目的変数に対する P 値が 10%以下の説明変数、いい換えるとプロジェクトのコスト超過・納期遅延・品質低下という遂行結果に影響を与えたと考えられる要因の候補、それぞれの P 値、および寄与率（量的データの場合）または V 値（質的データの場合）を表 4 に示す。ただし、V 値は P 値が 5%以下のものに対してだけ示している。なお、表 4 では、影響要因の候補をプロジェクト運営の視点から、開発プロジェクトではコントロールできない「外部要件」、開発プロジェクトでコントロールできる「開発側で管理可能な項目」、プロジェクト遂行後に結果の分かる「遂行結果」に分類している。特に質的説明変数に対しては、表 3 での分類体系を少し変形して「要求レベル」、「計画の評価」、「プロジェクト体制」、「ツールの利用」の 4 つに分類している。

P 値が 5%以下のものおよび寄与率が 10%以上のものに対しては、計算の根拠となった統計量を表 5、表 6、表 7、表 8 に示す。表 5 は品質低下に影響を及ぼす要因のクロス集計表である。表 6 は納期遅延の有無と要因の候補のクロス集計表である。表 5 と表 6 から、各要因の候補の項目のレベルが低い場合、それぞれ品質低下あるいは納期遅延が発生する割合が高いことが読み取れる。表 7 では実績の評価（コスト・工期・品質）に対する量的データの平均と分散を示す。表 8 は説明変数がテスト体制のクロス集計表である。表 8 でも、各項目のレベルが低い場合 QCD それぞれに問題が発生しやすいことが読み取れる。

4.1 量的データの影響要因の候補

t 検定の結果、危険率 5%以下（5%および 1%）で有意な結果が得られたのは、目的変数が実績の評価（コスト）で説明変数が ① 規模、② 工数、③ 工期、④ バグ数、⑤ 工期あたりの規模の場合と、目的変数が実績の評価（工期）で説明変数が ⑥ 規模当りのバグ数の場合だけであった（表 4）。危険率を 10%にまで上げた場合、実績の評価（品質）と ⑦ 規模あたりのバグ数、および実績の評価（工期）と ⑧ バグ数に関連があるという結果が得られた。

4.2 質的データの影響要因の候補

表 3 に示した 44 個の説明変数のうち、開発方法、主開発言語を除いて 15 項目（表 4 の「要求レベル」、「計画の評価」、「プロジェクト体制」、「ツールの利用」に属する項目、ただしテスト体制に関しては表 3 の分類に従って 1 項目とカウント）が、少なくともいず

表 4 プロジェクトの遂行結果と関連のある説明変数の寄与率 (または V 値) と P 値

Table 4 Contribution rates or Cramer's V values and P values of predictor variables correlated with variables of project performance.

分類	説明変数 (要因の候補)	プロジェクトの遂行結果			
		コスト 超過	納期 遅延	品質 低下	
外部要件	規模	8.6 (2.3)	—	—	
	工期	5.4 (1.3)	—	—	
	工期あたりの規模	11.1 (1.7)	—	—	
	要求レベル	要求仕様の明確度合い	—	0.27 (2.7)	— (9.4)
		要求レベル (移植性)	—	—	0.67 (0.6)
		要求レベル (性能・効率性)	—	—	0.53 (2.5)
	計画の評価	計画の評価 (コスト)	—	0.15 (4.9)	—
		計画の評価 (工期)	—	0.34 (0.001)	—
	開発側で管理可能な項目	ユーザとの役割分担・責任所在 _明確度合い	—	—	—
			達成目標_優先度 _明確度合い	—	—
PM スキル			—	—	— (9.3)
要員スキル (業務分野経験)		—	0.43 (0.008)	0.40 (4.2)	
		要員スキル (開発プラットフォーム使用経験)	—	—	—
テスト体制 (スキ&員数)			0.45 (3.8)	0.46 (2.6)	0.58 (1.2)
テスト体制 (スキルの有無)			—	—	0.58 (1.2)
テスト体制 (員数の充足性)			—	0.56 (1.2)	— (7.0)
ツールの利用		類似プロジェクト有無	—	—	— (6.1)
		プロジェクト管理ツール	—	—	— (9.8)
	構成管理ツール	—	—	— (7.8)	
	デバッグ_テストツール	—	—	— (8.9) (8.9)	
遂行結果	工数	5.8 (0.04)	—	—	
	バグ数	8.2 (0.3)	2.9 (8.0)	—	
	規模あたりのバグ数	—	14.4 (3.9)	27.3 (6.3)	

(注) 量的データの数値は、寄与率 [単位: %] (P 値 [単位: %]), 質的データの数値は、クラメールの V 値 (P 値 [単位: %]) である。

表 5 品質低下の有無と要因の候補のクロス集計表

Table 5 Candidate items that lower quality and quality performance.

要因の候補	項目	品質低下		合計
		なし	あり	
要求レベル _性能・効率性	きわめて高い+高い	8	1 (11%)	9
	中位 (+低い) (*1)	4	7 (64%)	11
	合計	12	8 (40%)	20
要求レベル _移植性	きわめて高い+高い+中位	10	2 (17%)	12
	低い (*2)	1	6 (86%)	7
	合計	11	8 (42%)	19
要員スキル _業務分野経験	全員が十分な経験+半数が十分な経験, 残り半数はいくらかの経験	22	4 (15%)	26
	半数がいくつかの経験, 残り半数は経験なし+全員が経験なし	3	4 (57%)	7
	合計	25	8 (24%)	33

(*1) 「低い」に属するプロジェクトはなかった。

(*2) 「中位」+「低い」では有意にならない。

表 6 納期遅延の有無と要因の候補のクロス集計表

Table 6 Candidate items that cause excessive time for delivery and delivery performance.

要因の候補	項目	納期遅延		合計
		なし	あり	
要求仕様_明確度合い	非常に明確+かなり明確	33	6 (15%)	39
	ややあいまい+非常にあいまい	17	11 (39%)	28
	合計	50	17 (25%)	67
計画の評価 (コスト)	コスト算定の根拠が明確で実行可能性を検討済み (*1)	114	27 (19%)	141
	コスト算定の根拠が不明確, または実行可能性を未検討 (*2)	13	8 (38%)	21
	合計	127	35 (22%)	162
計画の評価 (工期)	工期計画の根拠が明確で実行可能性を検討済み (*1)	119	24 (17%)	143
	工期計画の根拠が不明確, または実行可能性を未検討 (*2)	8	12 (60%)	20
	合計	127	36 (22%)	163
要員スキル (業務分野経験)	全員が十分な経験+半数が十分な経験, 残り半数はいくらかの経験	60	11 (15%)	71
	半数がいくつかの経験, 残り半数は経験なし+全員が経験なし	5	9 (64%)	14
	合計	65	20 (24%)	85

(*1) 「計画あり」と見なす。(*2) 「計画なし」と見なす。

表 7 実績の評価に対する量的データの平均と分散
Table 7 Comparison of means and variances between successful and failed projects.

		コスト超過	
		なし	あり
規模	平均	3	3.43
	分散	0.289	0.251
	標本数	50	10
工数	平均	4.09	4.48
	分散	0.373	0.253
	標本数	172	37
工期	平均	0.84	1
	分散	0.055	0.09
	標本数	97	17
バグ数	平均	2.31	2.74
	分散	0.351	0.288
	標本数	84	21
工期あたりの規模	平均	2.11	2.51
	分散	0.205	0.045
	標本数	43	8
		納期遅延	
		なし	あり
規模あたりのバグ数	平均	-0.81	-0.37
	分散	0.137	0.141
	標本数	26	4
		品質低下	
		なし	あり
規模あたりのバグ数	平均	-0.87	-0.36
	分散	0.119	0.008
	標本数	11	2

表 8 テスト体制とプロジェクトの遂行結果のクロス集計表
Table 8 Test team ability and QCD performances.

要因	項目	コスト超過		納期遅延		品質低下	
		なし	あり	なし	あり	なし	あり
スキルと員数	ともに十分	23	2	21	4	17	3
	上記以外	3	3	2	4	1	4
	合計	26	5	23	8	18	7
スキル	十分	23	3	21	5	17	3
	不十分	3	2	2	3	1	4
	合計	26	5	23	8	18	7
員数	十分	25	3	23	5	18	5
	不十分	1	2	0	3	0	2
	合計	26	5	23	8	18	7

(注)「テスト体制(スキル&員数)品質低下」と「テスト体制(スキル)品質低下」は同一データである。

れかの目的変数と10%以下(危険率1%, 5%, 10%)のP値を持つことが分かった(表4)。ユーザ担当者の関与の程度や経験, およびアーキテクチャの違いからは有意な結果は得られなかった。

コスト超過, 納期遅延, 品質低下と関連のある説明変数は, 次に示すように目的変数ごとに異なる。

(1) コスト超過と関連のある説明変数のうち危険率5%で有意となるものはテスト体制だけである。危

険率10%で有意となるものとして, ユーザとの役割分担・責任所在・明確度合い, 要員スキル(開発プラットフォーム使用経験), 類似プロジェクトの有無がある。

- (2) 納期遅延と関連のある説明変数のうち危険率1%で有意となるものとして計画の評価(工期)(の有無), 要員スキル(業務分野経験)があり, 危険率5%で有意となるものとして要求仕様・明確度合い, 計画の評価(コスト), テスト体制がある。危険率10%で有意となるものとして, 要求レベル(移植性), ツール利用(テスト・デバッグツール)がある。
- (3) 品質低下と関連のある説明変数のうち危険率1%で有意となるものとして要求レベル(移植性)があり, 危険率5%で有意となるものとして要求レベル(性能・効率性), 要員スキル(業務分野経験), テスト体制がある。危険率10%で有意となるものとして要求仕様・明確度合い, 達成目標・優先度・明確度合い, PMスキル, ツール利用(プロジェクト管理ツール, 構成管理ツール, テスト・デバッグツール)がある。

5. 考察

ここでは4章における結果に基づいて, プロジェクトの遂行に影響を与えそうな要因を最も有力な候補から順に検討を加えることにより, 真の影響要因を推定する。特に筆者らの経験に照らして疑わしい候補については, 分析対象となったプロジェクトデータを調査し, 他の要因の影響がないが再検討する。

5.1 外部要件の影響

5.1.1 コスト超過の要因

表7により, コスト超過と5%以下の危険率で有意な関連の見られる量的データを調べてみると, コスト超過を起こしたプロジェクトはそうでないプロジェクトよりも, 明らかに開発規模と工数が大きく工期が長い(表7の平均は対数変換後の値であるので, この平均に0.3の差があると対数変換を元に戻したときに2倍の差があることになる)。いわゆる「大規模プロジェクト」に相当するものである。これらのプロジェクトはバグ数も多い。

しかし, 開発規模で正規化した量と比較すると, 規模あたりの工数および規模あたりのバグ数では差が見られず, 工期あたりの規模にのみ有意な差が見られた(表4)。工期あたり規模の寄与率も11.1%と量的要因の中で最も高い。このことから, コスト超過は大規模プロジェクト, 特に工期あたりの規模の大きいプロジェクトで起きやすいといえる。この理由としては, 大規

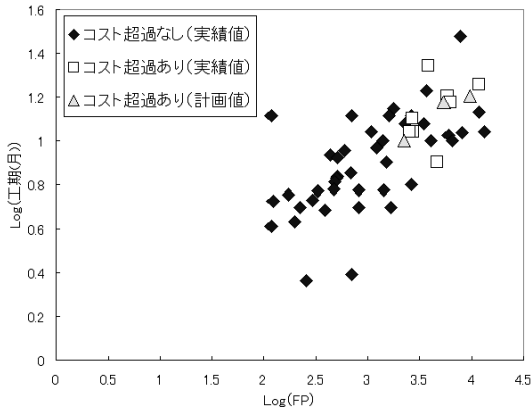


図1 コスト超過の有無に対する規模と工期の散布図

Fig.1 Scatter diagram of software size and development duration focusing on cost performance.

模プロジェクトでは一般に工期が長くなり、それに伴って仕様の追加・変更が増え、結果として当初予定していたコストを上回るということが考えられる¹⁵⁾。ただし、この結果は実績値に基づいたものであり、計画段階でリスク要因としてあげることができるかどうかは、計画値で分析する必要がある。

コスト超過を起こしたプロジェクトで規模と工期の両方の計画値が揃っているものは3件しかなかったが、規模と工期の散布図を描いてみると(図1)、上記の結論を揺るがすものではないことが分かる。なお、1カ月あたり200FP以下のプロジェクトではコスト超過は起きていなかった。また、コスト超過を起こしたプロジェクト群で業種・アーキテクチャの偏りは見られなかった。

5.1.2 納期遅延および品質低下の要因

量的データでは、規模あたりのバグ数が納期遅延と品質低下のいずれとも統計的に有意な相関を持ち、寄与率もそれぞれ14.4%、27.3%と高い(表4)。規模あたりのバグ数の増加は品質低下そのものであるが、それにより納期遅延も起きやすくなることが実証できた。

質的データに関しては、要求仕様が明確なプロジェクトに比較して要求仕様があいまいなプロジェクトは、納期遅延を起こす可能性が高く(5%有意)、品質低下を起こす傾向(10%有意)も見られる(表4)。この結果はGlassの法則としてよく知られている「要件の不備はプロジェクト失敗の根本的な原因である」¹³⁾ということを実証したものと見える。

一方、移植性に対する要求レベルが高いと品質低下を起こす可能性が低く(1%有意)、納期遅延を「起こさない」傾向が見られる(10%有意)という結果(表4および表5)に対しては、移植性の要求が高いために

表9 コストと工期に関する計画の評価の有無と納期遅延のクロス集計表

Table 9 Evaluation for both cost and delivery plans and delivery performance.

計画の評価		納期遅延	
コスト	工期	なし	あり
あり	あり	113	24
	なし	1	3
なし	あり	6	0
	なし	7	8
未回答	なし	0	1

可読性の高いコードを書くなどの工夫によって結果的に品質が高まるということによるものと思われる。

5.2 プロジェクトの努力と遂行能力の向上

表4における「開発側で管理可能な項目」によって、プロジェクトの遂行に影響を与える外部要件を緩和できると考えられる。

5.2.1 計画時のQCDの評価と納期遅延防止

計画の評価には、コスト・工期・品質に関するものがあるが、これらは「納期遅延」に対して最も顕著な効果を発揮している(表4)。

- (1) 工期の評価をきちんと行ったプロジェクトが納期遅延を起こす割合は17%、そうでないプロジェクトは60%と顕著な違いが見られる(表6)。
- (2) コストの評価をきちんと行ったプロジェクトは、コスト超過に対するよりも納期遅延に対して効果を発揮しているように見える(表4)。しかし、表6から分かるように計画の評価(コスト)と計画の評価(工期)のクロス集計表が似ているので、計画の評価(工期)の有無だけによるものか、計画の評価(コスト)の有無にも依存しているかを調べる必要がある。

コストと工期に対する計画の評価の有無と納期遅延の関係を表9に示す。コストと工期に対する計画の評価の有無に違いがあるのは10件(6%)だけであった(表9の太字部分)。表9をもとに、納期遅延が計画の評価(工期)の有無にだけによるものか計画の評価(コスト)の有無にも依存しているかを調べるために、改めて次の4つの場合についてクロス集計表を作成して納期遅延との関係を調べた。

- 計画の評価(コスト)が行われた場合と行われなかった場合「それぞれ」に対して、計画の評価(工期)が行われていたか否かによる納期遅延への影響
- 計画の評価(工期)が行われた場合と行われなかった場合「それぞれ」に対して、計画の評価(コスト)が行われていたか否かによる納期遅延への

表 10 納期遅延に及ぼす「計画の評価」の効果の詳細分析
Table 10 Effect of planning to avoid excessive time for delivery.

計画の評価		他の「計画の評価」の影響	P 値 (%)
コスト	あり	計画の評価 (工期) の有無	0.9
	なし		3.2
工期	あり	計画の評価 (コスト) の有無	32.5
	なし		42.6

表 11 計画の評価 (コスト) とコスト超過のクロス集計表
Table 11 Evaluation of cost and cost overruns.

	内容	コスト超過		合計	P 値
		なし	あり		
計画の評価 (コスト)	コスト算定の根拠が明確で実行可能性を検討済み	116	26 (18%)	142	11.0%
	コスト算定の根拠が不明確、または実行可能性を未検討	14	7 (33%)	21	
	合計	130	33	163	

影響

表 10 に分析結果を示す。表 10 から分かるように、計画の評価 (コスト) を行った場合も行わなかった場合も、計画の評価 (工期) の有無の差は小さな P 値を与える。すなわち計画の評価 (工期) の有無と納期遅延の有無の関連 (この場合は逆相関) が強いことが分かる。一方、計画の評価 (工期) を行った場合も行わなかった場合も、計画の評価 (コスト) の有無の差の P 値は大きく、計画の評価 (コスト) の有無が納期遅延に影響を与えない。

計画の評価 (コスト) の効果については、正確なコスト予測が出荷後品質および生産性と有意な相関を持っているという報告がある¹⁷⁾。この結果をふまえて作成した計画の評価 (コスト) と実績の評価 (コスト) の結果すなわちコスト超過の有無のクロス集計表を表 11 に示す。計画の評価 (コスト) がコスト超過に対して効果があるかどうかを示す P 値は 11.0% であって、今回設定した最も緩い有意水準 (P 値が 10%) に達していない (したがって表 4 には記載していない)。しかし、142 件中 26 件 (18%) のコスト超過なしと 21 件中 7 件 (33%) のコスト超過ありという結果の差は小さくない。つまり、この結果は「計画の評価 (コスト) はコスト超過に効果がない」というべきではなく、有意水準にはわずかに足りないが、計画の評価 (コスト) もコスト超過の防止に効果をもたらしている可能性があると考えた方がよい。

以上のことから計画の評価はプロジェクトの順調な遂行に重要な役割を果たしていると考えられる。

5.2.2 プロジェクト体制の影響

プロジェクトを組む際に優秀な PM・経験豊富な要員を揃えるのは理想であるが、現実にそのようなプロジェクトをつねに組織できるとは限らない。ここでは、プロジェクト体制とプロジェクト遂行能力の関係について検討する。

(1) 要員の業務分野経験が豊富なプロジェクトは、納期遅延と品質低下をともし起こしにくい (それぞれ 1% 有意と 5% 有意) (表 4)。

この結果は「良い設計にはアプリケーション領域の深い知識が必要である」という Curtis の法則¹³⁾と深く関連する。良い設計が納期遅延や品質低下を防止することを実証したものと見える。さらに、分析設計経験、言語・ツール利用経験・開発プラットフォーム利用経験は開発ソフトウェアの業務分野によらないソフトウェア開発共通のスキルのため PM も把握しやすく、計画段階で考慮に入れることが比較的容易なのに比べ、業務経験の有無は正確には把握しにくく計画段階で考慮しにくいことも考えられる。

(2) ユーザとの役割分担・責任所在・明確度合いが高いプロジェクトおよび要員スキル (開発プラットフォーム使用経験) が高いプロジェクトはいずれもコスト超過を起こしにくい傾向が見られる (いずれも 10% 有意) (表 4)。

この結果から、ユーザとの役割分担・責任所在・明確度合いが高いプロジェクトは開発コストの見積りが比較的行きやすいことによるのではないかと推察される。

(3) 達成目標・優先度・明確度合いの高いプロジェクトおよび PM スキルの高いプロジェクトは、いずれも品質低下を起こしにくい傾向が見られる (いずれも 10% 有意) (表 4)。

この結果によると、達成目標・優先度・明確度合いは作業の質を高める効果があるのかもしれない。PM スキルがコスト超過の有無と関係があるとはいえないのに対して品質低下の有無と有意な関係があるのは、PM スキルの高さがコスト見積りの中に織り込み済みであるのに対し、PM スキルの品質低下への影響の大きさがいまだ十分に認識されていないためではないかと推測される。

5.2.3 テスト体制の効果

優れたテスト体制はコスト・納期・品質いずれの面からプロジェクトの遂行により結果をもたらす (表 4)。

(1) スキル・員数とも十分なプロジェクトはそうでないプロジェクトに比較してコスト超過・納期遅延・

品質低下のいずれをも起こしにくい(5%有意).これはテスト作業の重要性を実証したものである.

- (2) テスト要員にスキルがあると納期遅延より品質低下の防止に効果が見られる(5%有意).このことは,品質低下が作業の量の問題ではなく,質の問題であることを示唆している.
- (3) テスト要員数が充足していると,品質低下より納期遅延の防止に効果が見られる(5%有意).このことは,納期遅延が主に作業の量の問題であることを示唆している.

5.2.4 ツールの利用

ツールの利用はプロジェクト遂行に良い影響を与える傾向が見られる.ただし,利用するツールによって効果の見られる QCD が異なる(表 4).

- (1) コスト超過の防止に有効と考えられるものは,「類似プロジェクトの存在」である(10%有意).
この結果は,コストが規模にほぼ比例することから,類似プロジェクトがあれば規模が異なってもコスト見積りの精度を向上させることができるからではないかと推察される.
- (2) 納期遅延の防止に有効と考えられるものは,「ツール利用(デバッグ・テストツール)」である(10%有意).
この結果は,デバッグ・テストツールの利用がデバッグ・テスト作業の効率化に役立ち,その結果納期遅延の防止に役立っているということの実証である.
- (3) 品質低下の防止に有効と考えられるものは,「ツール利用(プロジェクト管理ツール,構成管理ツール,デバッグ・テストツール)」である(いずれも 10%有意).

この結果は,構成管理ツールを利用することによりバグをきちんと管理でき,デバッグ・テストツールを利用することによりデバッグ・テストの効率を向上させることができ,これらが品質低下を避けることができるという経験則を実証したものと見える.

5.3 業種・主開発言語の違いによる影響

業種(大分類)と主開発言語に関しては,分類尺度の

選択肢が多く,かつ件数の少ないものが多い.そこで件数の多いもののみを選択して有意性の検証を行った.

5.3.1 業種(大分類)別

コスト超過を起こすプロジェクト数とその割合を業種(大分類)ごとに比較した結果を表 12 に示す. Fisher の直接計算法により,金融・保険業と情報通信業とでコスト超過を起こす割合に違いがあるかどうかを調べたところ,5%有意で金融・保険業の方がコスト超過を起こす割合が高いことが分かった.

コスト超過を起こしやすい理由が業種によるものか, 5.1.1 項で述べたように規模によるものかを調べるために,金融・保険業のうち規模データの揃っているプロジェクトに絞ってコスト超過の有無により規模に差があるかどうかを調べた(表 13).金融・保険業で規模データの揃っているものは 15 件しかなかったが,金融・保険業のコスト超過プロジェクトの比率は,表 12 では $15/57 = 26\%$,表 13 では $4/15 = 27\%$ と差がなく,規模データ有という条件を加えた絞り込みによる分布の変化は起きていないと見なせる.

t 検定の結果,コスト超過がある場合もない場合もそれぞれ金融・保険業と他の業種との間に規模の差はなく,金融・保険業のプロジェクトではコスト超過を起こしたプロジェクトは起こさなかったプロジェクトより規模が大きいことが分かった.このことから,コスト超過の原因は金融・保険業という業種によるものではなく,すでに 5.1.1 項で明らかにした事実,規模が大きいことによるものと考えられる.

表 12 業種(大分類)ごとのコスト超過を起こすプロジェクト数
Table 12 Number of cost overrun projects in each industry.

業種 (大分類)	コスト超過		合計
	なし	あり	
製造業	31	7 (18%)	38
情報通信業	20	1 (5%)	21
卸売・小売業	25	5 (17%)	30
金融・保険業	42	15 (26%)	57
公務	23	1 (4%)	24
合計	141	29	170

表 13 金融・保険業のプロジェクトにおける規模(FP)とコスト超過の有無による規模の差の検定結果

Table 13 Software size of the projects of finance/insurance industry and P values of t-test for sizes from the viewpoint of cost overruns.

	金融・保険業			左記以外の業種			全体			t 検定の P 値 (%)	
	標本数	平均	分散	標本数	平均	分散	標本数	平均	分散		
コスト超過	なし	11	3.02	0.308	39	2.99	0.291	50	3.00	0.289	90.0
	あり	4	3.55	0.031	6	3.35	0.413	10	3.43	0.251	48.7
	全体	15	3.16	0.287	45	3.04	0.313	60	3.07	0.304	47.4
t 検定の P 値 (%)			1.4			15.2			2.3		

(注) 平均と分散は $\log(\text{FP})$ に対して計算した値

5.3.2 開発言語別

主開発言語として使用しているプロジェクト数の多い Java, COBOL, C 言語, VB を対象にコスト超過と品質低下を起こすプロジェクト数を調べた結果を表 14 に示す。表 14 より, VB > C 言語 ≫ Java > COBOL の順にコスト超過を起こしにくく, COBOL > C 言語 > Java ≫ VB の順に品質低下を起こしにくいということが分かる。

表 14 の結果をもとに, 各言語の対ごとに 2 × 2 のクロス集計表を作成して χ^2 検定または Fisher の直接計算法を用いて, 開発言語によってコスト超過あるいは品質低下を起こす割合にそれぞれ差があるかどうかを調べた (表 15)。表 15 から分かるように VB だけが他の言語とコスト超過および品質低下の割合が異なっている。

コスト超過を起こした比率が最も高い COBOL と最も低い VB について規模との関係を調べた。すなわち, COBOL 使用でコスト超過を起こさなかったプロジェクトのうち規模データのあるもの 3 件, VB 使用でコスト超過を起こさなかったプロジェクトのうち規模データのあるもの 3 件, COBOL 使用でコスト超過を起こしたプロジェクトのうち規模データのあるもの 3 件のそれぞれの対で規模の差に関する t 検定を行った (表 16)。表 16 から COBOL 使用でコスト超過を起こしたプロジェクトが, COBOL または VB 使用で

コスト超過を起こさなかったプロジェクトよりも規模が大きい傾向があることが分かる。なお, COBOL と VB それぞれでコスト超過を起こさなかったプロジェクト群どうしでは規模の差は見られなかった。このことから, データ数が極端に少ないため断定はできないものの, コスト超過の原因は言語自体の性質というよりもその言語を使用したプロジェクトの規模の大きさによるものと考えた方がよいと思われる。

5.4 プロジェクト成功への指針

分析結果に基づいて, コスト・工期・品質の面で計画どおりにプロジェクトを遂行するために重要と思われることを次に示す。

- (1) コスト超過を避けるには, 工期あたりの規模を適正にし (全体規模をコントロールすることは困難と思われる), 類似プロジェクトの存在しない場合はできればあらかじめ高めにコストを設定しておく。ユーザとの役割分担・責任所在を明確にし, 要員には開発プラットフォーム利用経験者を揃えると効果があると思われる。また, 要員数・スキルともに十分なテスト体制を整えておくべきである。
- (2) 納期遅延を避けるためには, できる限り要求仕様を明確にし, 計画段階で工期の妥当性を評価しておくべきである。また, 業務経験の豊富な要員や少なくとも十分な数の要員を揃えたテスト体制を整えておくべきである。テスト・デバッグツールの利用も効果があると思われる。移植性の要求レベルが高い場合は, 納期遅延の可能性を十分に考慮してプロジェクト運営を進めるのがよい。
- (3) 品質低下を避けるためには, できる限り要求仕様を明確にすること, 業務経験の豊富な要員を揃えること, スキルのあるテスト要員を含む十分なテスト体制を整えるべきである。レベルの高い PM を配置すること, 達成目標と優先度を明確にすること, プロジェクト管理ツール, 構成管理ツール, デバッグ・テストツールを利用することも効果があると思

表 14 主開発言語ごとのコスト超過と品質低下を起こすプロジェクト数

Table 14 Number of projects with cost overruns and lower quality for each development language.

主開発言語	コスト超過		合計	品質低下		合計
	なし	あり		なし	あり	
COBOL	26	8	34	10	2	12
C 言語	28	2	30	14	3	17
VB	26	1	27	4	6	10
Java	40	10	50	14	4	18
合計	120	21	141	42	15	57

表 15 コスト超過の有無の主開発言語間での差に対する有意性検定の P 値

Table 15 P values of significance test for cost overruns between programming languages.

	COBOL	Java	C 言語	VB
COBOL	—	—	(6.3%)	3.1%
Java	—	—	(9.5%)	4.7%
C 言語	—	—	—	—
VB	4.8%	(5.7%)	3.4%	—

(注 1) 右上段: コスト超過に関するもの, 左下段: 品質低下に関するもの

(注 2) 欄内の「—」は有意でないことを示す。

(注 3) 括弧内は P 値が 5 ~ 10% の範囲のものである。

表 16 コスト超過の有無と主開発言語の観点から分類したプロジェクトの規模と規模の差に関する検定結果

Table 16 Software size of the projects categorized by cost performance and programming languages and P values of t-test for sizes between each category.

項番	コスト超過の有無	主開発言語	標本数	規模 (log FP)		項番 3 との t 検定の P 値
				平均	分散	
1	なし	COBOL	3	3.31	0.022	8.7%
2	なし	VB	3	3.25	0.033	7.2%
3	あり	COBOL	3	3.76	0.098	—

(注) 項番 1 と 2 の t 検定の結果は有意でない。

われる。特に、移植性や性能・効率性の要求レベルが高い場合は、品質低下を引き起こす可能性が高くなるので上記の施策を徹底すべきである。

6. おわりに

エンタプライズ系ソフトウェア開発におけるプロジェクトの遂行でコスト・納期・品質目標が当初計画どおりに達成できなかった要因を分析し、次のことを明らかにした。

- (1) プロジェクト規模そのものが大きく、特に工期あたりの規模が大きいプロジェクトはコスト超過を起こす割合が高い。プロジェクト規模そのものが大きいプロジェクトは工期も長くなるため仕様の追加・変更も多くなると考えられ、これがコスト超過に至る大きな要因となっていると考えられる。
- (2) 要求仕様があいまいなプロジェクトは納期遅延や品質低下を起こす割合が高い。これは Glass の法則をデータで実証したものと見える。
- (3) 事前に工期の妥当性を評価したプロジェクトでは納期遅延を起こす割合が低い。これは参考文献 17) で示されていることの拡張とも見える。
- (4) 業務分野の経験者を揃えたプロジェクトでは納期遅延や品質低下を起こす割合が低い。これは Curtis の法則を実証したものと見える。
- (5) テスト体制を整備したプロジェクトでは、コスト超過・納期遅延・品質低下を起こす割合が低い。これは経験的によく知られていることを実証したものと見える。

ここで示した知見は、すべて経験則あるいは類似の研究結果で示されているものである。本論文ではそれらをデータから実証的に示し、あるいは拡張したことに価値があると考えている。これらの知見はソフトウェア開発に普遍的なものであり、日本のエンタプライズ系ソフトウェア開発に限ったものではないと考える。

謝辞 本研究は、経済産業省の支援により、東海大学と IPA SEC が共同で実施したものである。定量データ分析部会委員各位および SEC 研究員である横山健次、本間周二両氏のご協力に感謝いたします。

参考文献

- 1) Boehm, B.W.: *Software Engineering Economics*, Prentice-Hall, Inc. (1981).
- 2) Boehm, B.W., et al.: *Software Cost Estimation with COCOMO II*, Prentice Hall PTR (2000).
- 3) Srinivasan, K. and Fisher, D.: *Machine Learning Approaches to Estimating Software De-*

velopment Effort, *IEEE Trans. Softw. Eng.*, Vol.21, No.2, pp.126–137 (1995).

- 4) Chulani, S., Boehm, B.W. and Steece, B.: Bayesian Analysis of Empirical Software Engineering Cost Models, *IEEE Trans. Softw. Eng.*, Vol.25, No.4, pp.573–583 (1999).
- 5) Pendharkar, P.C., Subramanian, G.H. and Rodger, J.A.: A Probabilistic Model for Predicting Software Development Effort, *IEEE Trans. Softw. Eng.*, Vol.31, No.7, pp.615–624 (2005).
- 6) Shepperd, M. and Schofield, C.: Estimating Software Project Effort Using Analogies, *IEEE Trans. Softw. Eng.*, Vol.23, No.12, pp.736–743 (1997).
- 7) Auer, M., Trendowicz, A., Graser, B., Haunschmid, E. and Biffl, S.: Optimal Project Feature Weights in Analogy-Based Cost Estimation: Improvement and Limitations, *IEEE Trans. Softw. Eng.*, Vol.32, No.2, pp.83–92 (2006).
- 8) 角田雅照, 大杉直樹, 門田暁人, 松本健一, 佐藤慎一: 協調フィルタリングを用いたソフトウェア開発工数予測方法, 情報処理学会論文誌, Vol.46, No.5, pp.1155–1164 (2005).
- 9) Maxwell, K.D., Van Wassenhove, L. and Dutta, S.: Software Development Productivity of European Space, Military and Industrial Applications, *IEEE Trans. Softw. Eng.*, Vol.22, No.10, pp.706–718 (1996).
- 10) Blackburn, J.D., Scudder, G.D. and Van Wassenhove, L.N.: Improving Speed and Productivity of Software Development: A Global Survey of Software Developers, *IEEE Trans. Softw. Eng.*, Vol.22, No.12, pp.875–885 (1996).
- 11) Maxwell, K.D. and Forselius, P.: Benchmarking Software Development Productivity, *IEEE Software*, Vol.17, No.1, pp.80–88 (2000).
- 12) Tsunoda, M., Monden, A., Yadohisa, H., Kikuchi, N. and Matsumoto, K.: Productivity Analysis of Japanese Enterprise Software Development Projects, *Proc. MSR'06 Workshop*, pp.14–17 (May 2006).
- 13) Rombach, E. (著), 吉舖紀子 (訳): ソフトウェア工学ハンドブック, コンピュータエージ社 (2005).
- 14) ISBSG: *The Benchmark Release 8*, ISBSG (2004).
- 15) Jones, C. (著), 富野 壽, 鶴保証城 (監修): ソフトウェア開発の定量化手法 (第 2 版), 共立出版 (1998).
- 16) 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター (著作監修): ソフトウェア開発データ白書 2006, 日経 BP 社 (2006).

- 17) Mizuno, O., et al.: Analyzing Effect of Cost Estimation Accuracy on Quality and Productivity, *Proc. 20th Int. Conf. on Software Engineering (ICSE'98)*, pp.410-419 (Apr. 1998).

(平成 18 年 11 月 30 日受付)
(平成 19 年 5 月 9 日採録)



古山 恒夫 (正会員)

1945 年生。1968 年東京大学工学部計数工学科卒業。1973 年同大学院博士課程修了。同年日本電信電話公社入社。横須賀電気通信研究所で、拡張型言語, Ada, Common LISP などの言語処理プログラムの研究開発に従事。日本電信電話(株)ソフトウェア研究所でソフトウェアプロジェクト管理法, ソフトウェア品質保証法, ソフトウェア見積り法などの研究実用化に従事。1996 年東海大学開発工学部情報通信工学科教授。2003 年より同大学理学部情報数理工学科教授。1994 年度情報処理学会山下記念研究賞受賞, 1997 年より日本ファンクションポイントユーザ会名誉会長, 1999~2002 年情報処理学会規格調査会 ISO/IEC JTC1 SC7/WG13 (ソフトウェア測定プロセス) 小委員会主査, 主な著書『実践的ソフトウェア測定』(監訳, 共立出版), 『ソフトウェア品質工学の尺度とモデル』(監訳, 共立出版)。工学博士。IEEE, 電子情報通信学会各会員。



菊地奈穂美 (正会員)

1985 年新潟大学理学部数学科卒業。1993 年 Stanford 大学コンピュータ・サイエンス修士 (MSCS)。2006 年大阪大学博士 (工学)。沖電気工業(株)にて、通信ソフトウェア仕様記述言語 SDL, MSC およびベトリネット等による設計&検証システムの研究と、支援ソフトウェアの設計・プロジェクト管理等を経験。近年は、ソフトウェアのプロセス評価, メトリクス, 設計&テスト技法/ツール, 見積技法等の研究・技術移転と改善推進, ソフトウェア信頼性技術等に従事。2007 年現在, 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター研究員を兼務。PM 学会, IEEE の会員。



安田 守

1961 年生。1986 年鳥取大学大学院土木工学修士課程修了。同年野村総合研究所 (NRI) 入社, 2003 年より生産性向上推進部部長。2004 年 10 月より 2006 年 9 月まで情報処理推進機構ソフトウェアエンジニアリングセンター (SEC) エンタプライズ系プロジェクトリーダーを兼務。野村総合研究所でプロジェクトリーダーを歴任したのち, 生産性向上推進の組織を立ち上げ, エンジニアリングを活用した生産性および品質向上の施策を NRI 社内の展開している。



鶴保 征城 (正会員)

1942 年生。工学博士。1966 年大阪大学大学院工学研究科修士課程修了。同年 4 月日本電信電話公社入社。1989 年 11 月 NTT ソフトウェア研究所長, 1993 年 6 月 NTT データ通信株式会社取締役開発本部長, 常務取締役技術開発本部長を歴任, 1997 年 6 月 NTT ソフトウェア株式会社代表取締役社長に就任。2003 年 6 月より高知工科大学工学部情報システム工学科教授, 2004 年 10 月よりソフトウェア・エンジニアリング・センター所長。2001~2003 年情報処理学会会長。日本学術会議連携会員, XML コンソーシアム会長, 日本 BPM 協会副会長, 実践的ソフトウェア教育コンソーシアム会長。主な著書『ソフトウェア開発の定量化手法』(監訳, 共立出版), 『ずっと受けたかったソフトウェアエンジニアリングの授業①, ②』全 2 巻 (翔泳社)。