

# 原価率とプロジェクトメトリクスに着目した ソフトウェア開発プロジェクトの特徴分析

上野 秀剛\* 亀井 靖高\*\*† 門田 暁人\*\* 松本 健一\*\*

An Analysis of Characteristics of Software Projects Using Sales Cost Ratio and Project Metrics

Hidetake Uwano\* Yasutaka Kamei\*\*† Akito Monden\*\* Ken-ichi Matsumoto\*\*

本論文では予算超過プロジェクトの特徴を明らかにすることを目的として、ある組み込み系ソフトウェア開発企業の経理データに基づいて各プロジェクトの原価率を算出し、各開発工程の工数比率、外部委託率、コード再利用率との関係を統計的に分析する。分析の結果、原価率の低いプロジェクトは、外部設計に費やす工数の割合が高い傾向が見られた。また、外部委託率が高いプロジェクトは、低いプロジェクトに比べ原価率が高い傾向が見られた。さらに、コード再利用率が0%を超え99%未満であるプロジェクトは、原価率が高く、分散も大きい傾向が見られ、再利用に伴うトラブルがプロジェクトの収益性を低下させることが示唆された。

To clarify the characteristics of cost-overflow software projects, this paper calculated Sales Cost Ratio from accounting information of a company in the embedded systems domain, and analyzed the correlation with outsourcing ratio as well as reuse ratio and effort ratio of each development phase. As a result, we found that the lower Sales Cost Ratio projects had the higher effort ratio of architecture design phase. We also found that high outsourcing ratio projects had higher Sales Cost Ratio, and that projects having moderate reuse ratio (greater than 0% and below 99%) had lower and disperse Sales Cost Ratio, which suggests troubles in code reuse will degrade the project profit.

**Key Words & Phrases:** プロジェクト特性, 原価率, 開発工程, 外部委託, 再利用

Project Characteristics, Sales Cost Ratio, Development Phase, Outsource, Reuse

## 1. はじめに

ソフトウェア開発プロジェクトにおいては、開発費用がプロジェクト開始時の見積もりを上回ること（予算超過）による利益の減少や損失がしばしば発生する[1]。予算超過の原因の例としては、不完全な要求獲得、プロジェクトのマネジメント不足、工数見積もりの失敗、仕様変更の多発などによる工数超過があげられる。

このような失敗プロジェクトの発生を未然に防ぐために、過去の失敗プロジェクトの事例分析[2]

や、開発工程におけるリスク評価[3][4]などが行われている。また、統計的アプローチとして、失敗プロジェクトの特徴分析[1][5][6]や失敗プロジェクトを予測する手法の開発[7]が盛んに行われている。このように失敗プロジェクトの経験を前向きに活用することで、将来のプロジェクトにおいてリスクが表面化する前に検出し、失敗の低減に役立てることができる。

本稿では、企業の収益性に直結する指標である「原価率」に着目し、予算を超過した失敗プロジェクトと、予算以内に完了した成功プロジェクトの差異について統計的な分析を行う。どのような特徴を持ったプロジェクトが予算超過を起こしやすいかを把握できれば、特定の種類のプロジェクトについては、工数を多めに確保するといった経営判断や、リスク管理を重点的に行うなどのプロセス改善に役立つと期待される。このような原価率を用いたソフトウェア開発プロジェクトの分析事例は、従来、ほとんど公表されていない。

分析では、ある組み込み系ソフトウェア開発企

\*奈良工業高等専門学校 (Nara National College of Technology)

\*\*奈良先端科学技術大学院大学 (Nara Institute of Science and Technology)

†現在、カナダ クイーンズ大学 (Queen's University)

表1 分析に用いたデータ項目一覧

大分類	小分類	欠損率(%)	中央値	平均値	標準偏差
売上高 (千円)		0	15,574	34,398	46,042
製造原価 (千円)		0	13,620	30,298	39,986
工数 (人時)	要求分析	0	0	199	390
	外部設計	0	845	1,828	2,650
	内部設計	0	359	1,177	2,023
	実装	0	530	832	1,158
	単体試験	0	252	567	817
	SW 組合せ試験	0	366	823	1,421
	その他*	0	283	864	1,332
ソースコード 行数 (行)	新規ライン	21.1	14,354	61,110	181,828
	流用ライン	21.1	88,400	278,153	520,187
	変更ライン	21.1	0	4,096	8,883

\*運用, 教育, 保守など

業における経理データに基づいて、各プロジェクトで費やした費用の売上に対する割合である原価率を算出した。原価率は一般管理費を考慮せず、プロジェクト自体がどの程度の利益を得たかを示す値であり、開発組織ごとに定められた値を下回っていれば利益を出したプロジェクト（成功プロジェクト）、上回っていれば損失を出したプロジェクト（失敗プロジェクト）とみなすことができる。

プロジェクトの特徴を表す指標としては、各開発工程の工数比率（以下、工程比率）、外部委託率、ソースコードの再利用率に着目した。これらの指標はいずれもプロジェクトの開発形態や開発方針と密接に関係していることから、失敗しやすいプロジェクトの特徴を分析するのに適していると考えた。分析では、プロジェクトごとに各指標と原価率を算出し、成功プロジェクトと失敗プロジェクトの特徴がどのように異なっているか調査した。

以降、2章で分析対象のプロジェクトデータについて説明し、3章で分析に用いるメトリクスについて説明する。4章で分析の結果について考察し、5章でまとめと今後の課題について述べる。

## 2. 分析対象プロジェクト

本稿で分析対象とするデータは、あるソフトウェア開発企業で収集された95件のプロジェクトデータである。この企業では主に組み込みソフトウェアの開発を行っており、有線・無線通信や、画像処理などに関するソフトウェアの開発を行っている。

対象企業では請負開発が主であり、一つの契約単位を1プロジェクトとみなすことにする。各プロジェクトでは、外部設計以降の工程（外部設計、

内部設計、実装、単体試験、SW 組み合わせ試験）を中心に開発を行っている。本稿では対象企業の主要業務とは異なるプロジェクトの影響を排除するために、外部設計からSW 組み合わせ試験までの工程に50%以上の工数を費やしたプロジェクトのみを分析対象とする。

表1に収集したデータのうち、本稿で使用する項目の一覧および、欠損率、中央値、平均値、標準偏差の値を示す。製造原価は、人材費や材料費、外部委託費など、対象プロジェクトで直接的に消費した費用の合計とし、一般管理費は含まれない。また、ソースコード行数のうち、「新規ライン」は対象プロジェクトのために新規に作成されたコードの行数、「流用ライン」は既存のプロジェクトから変更することなく再利用されたコードの行数、「変更ライン」は既存のプロジェクトから変更を加えて再利用されたコードの行数をそれぞれ表している。なお、外部設計以降の工程を中心に行ったプロジェクトを対象としたため、要求分析に対する工数において、中央値が0となっている。また、規模は新規ラインや流用ラインとして計上されることが多く、変更ラインは大半のプロジェクトで0であったため、中央値が0となっている。

## 3. 分析に用いるメトリクス

本章では、分析対象とするプロジェクトの成否を判断するための指標である原価率と、プロジェクトの特徴を表す3種類の指標について説明する。表2に算出した指標の基本統計量を示す。

### 3.1 原価率

原価率は、製造原価の売上に対する割合（パー

表2 算出したメトリクス一覧

メトリクス		データ数	中央値	平均値	標準偏差
工程比率	要求分析	68	0.00	5.10	7.86
	外部設計		32.81	32.45	10.80
	内部設計		17.70	17.49	9.08
	実装		15.64	17.40	8.59
	単体試験		10.68	11.16	4.76
	SW 組み合わせ試験		14.46	16.40	7.71
外部委託率		95	52.54	43.33	27.11
コード再利用率		75	75.29	65.48	33.28

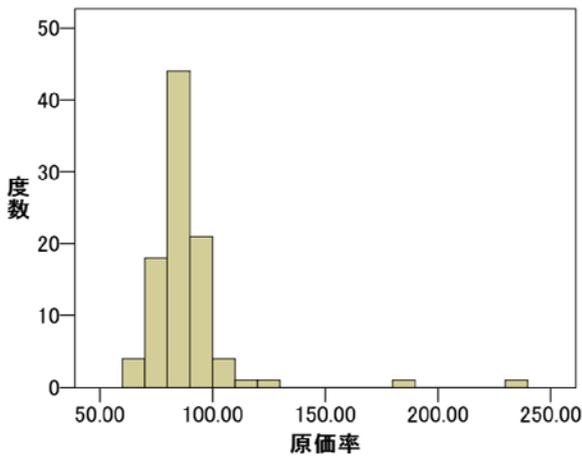


図1 原価率ごとの外部委託率

セント表記とする) であり、この値が 100 未満の場合、プロジェクトが単体で利益を得たといえる。しかし、最終的な利益へ寄与しているかを判断するためには、各プロジェクトが直接消費していない一般管理費（設備の維持費や賃料など）についても考慮する必要がある。

対象企業の管理者 2 名にインタビューを行った結果、同社の一般管理費が平均して売上高の 10% 程度であり、各プロジェクトの損益分岐点が 90% 前後であることが確認できた。そこで、本稿では、原価率が 90 以上のプロジェクトを失敗プロジェクト、90 未満のプロジェクトを成功プロジェクトとし、それぞれのプロジェクトがどのような特徴を持っているか分析する。なお、本研究で使用するプロジェクトデータについても管理者 2 名による確認を行い、原価率 90 でプロジェクトの成否を適切に分類できていることを確認した。図 1 に対象データにおける原価率の分布を示す。原価率 90 未満のプロジェクトの割合が 70% であり、全プロジェクトのうち、87% が原価率 70 以上 100 未満の範囲に集中している。

### 3.2 工程比率

工程比率はプロジェクトの各工程に対してど

のぐらいの工数を費やしたのかを示す値である。各工程に対する工程比率は対象プロジェクトの業種や開発形態（新規開発、派生開発など）ごとに異なる特徴を持っている [8][9]。他のプロジェクトと大きく異なった工程比率を持ったプロジェクトは、失敗するリスクがあると考えられる。例えば、要求分析や外部設計のような上位工程に費やす工数が通常より小さいプロジェクトでは、実装や試験のような下流工程での手戻りや不具合の発生が多くなり、予算を超過する危険性がある。

本稿では、分析対象のプロジェクトについて、各工程に費やした工数の合計を求め、外部設計から SW 組み合わせ試験の工程比率を算出する。95 件のプロジェクトデータのうち、外部設計から SW 組み合わせ試験までの全工程を行っている 68 件を分析対象とした。

### 3.3 外部委託率

開発中のシステムの一部機能や、一部の工程（例えば、実装工程や試験工程）を外部開発組織に委託することは、ソフトウェア開発の現場において広く行われている。開発に必要な人員の確保は企業経営上の重要な課題であり、外部委託を行うことで、開発人員を柔軟に調整できる。一方で、外部委託によってマネジメント上のリスクも高まる。

本稿では、製造原価のうち、外注費が占める割合を外部委託率とし、プロジェクトの原価率との関係を分析する。95 件すべてのプロジェクトデータを分析対象とした。

### 3.4 コード再利用率

あるソフトウェアを開発する際に、過去に開発した類似のソフトウェアからソースコードや設計を再利用することが一般に行われている。再利用率の低いプロジェクトでは、再利用率の高いプロジェクトに比べ、新規開発に伴う不具合発生の危険性が高い。一方で、再利用率の高いプロジェクトでは、既存システムの理解不足や変更作業の誤

表 3 各工程に対する平均工数比率と原価率

	原価率	中央値	p 値
要求分析 (%)	90 以上	3.37	0.103
	90 未満	0.00	
外部設計 (%)	90 以上	28.18	0.015
	90 未満	34.31	
内部設計 (%)	90 以上	19.86	0.396
	90 未満	17.22	
実装 (%)	90 以上	16.66	0.264
	90 未満	15.41	
単体試験 (%)	90 以上	10.29	0.545
	90 未満	10.68	
SW 組合 試験 (%)	90 以上	13.67	0.501
	90 未満	15.21	

りなどから想定外の工数が必要になる危険性がある。

近年のソフトウェア開発においては、既存システムの拡張開発や派生開発の割合が高く、再利用率の高さがプロジェクトの成否に与える影響を理解しておくことが重要となる。

本稿では、ソースコードの全行数(新規ライン、流用ライン、変更ラインの合計)に対する流用ライン数の割合からコード再利用率を求め、原価率との関係を分析する。分析では 95 件のプロジェクトのうち、データに欠損のない 75 件を分析対象とした。

#### 4. 結果と考察

##### 4.1 工程比率

表 3 に原価率が 90 未満の(予算を超過していない)プロジェクトと 90 以上の(予算を超過した)プロジェクトにおける工程比率を示す。原価率が 90 未満のプロジェクトでは、外部設計に対する工程割合が高く、要求分析に対する工程割合が低い傾向がみられた。内部設計、実装、単体試験、SW 組み合わせ試験については 2 つのグループでほとんど差が見られなかった。

外部設計工程の工程比率の箱ひげ図を図 2 に示す。図では、横軸に原価率が 90 未満のプロジェクトと 90 以上のプロジェクトを示し、縦軸は外部設計の工程比率を示す。原価率が 90 以上のプロジェクトは 90 未満のプロジェクトと比べ箱が大きく、分布がばらついていることを示している。

原価率が 90 以上のプロジェクトと、90 未満のプロジェクトで各工程について中央値の差の検定(Mann-Whitney の U 検定)を行ったところ、外部設計で有意差が見られた(p 値=0.015)。この結

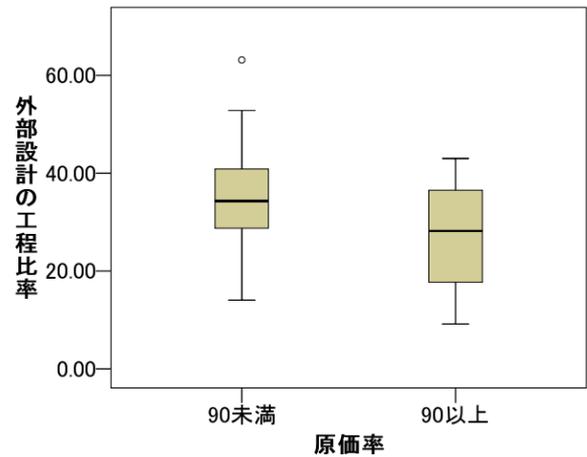


図 2 原価率ごとの外部設計の工程比率

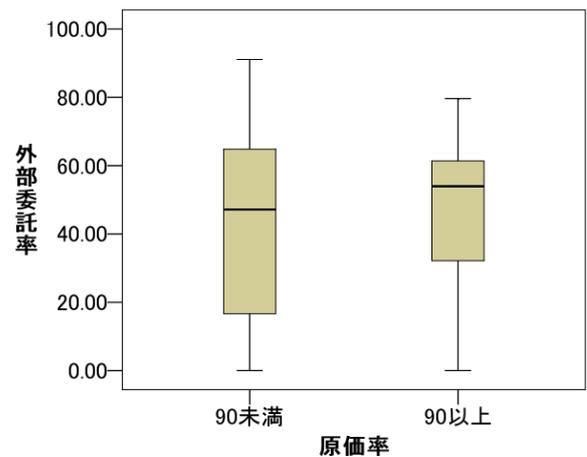


図 3 原価率ごとの外部委託率

果は、失敗プロジェクトでは上流工程である外部設計に費やした工数が小さく、下流での不具合や手戻り等が発生したことで、予算を超過した可能性を示している。一方で、成功プロジェクトでは上流工程を適切に行ったことで、下流工程での手戻りや不具合の発生を抑えることができた可能性がある。

##### 4.2 外部委託率

原価率が 90 未満のプロジェクトにおける外部委託率の中央値は 47.2%、90 以上のプロジェクトは 54.3%であり、予算を超過したプロジェクトは、外部委託率が高い傾向が見られた。ただし、図 3 に示すように、原価率が 90 未満、90 以上のいずれのグループも外部委託率の分散が大きく、2 つのグループ間に有意差は見られなかった(p 値=0.501)。

より詳細な分析として、プロジェクトを(1)外部委託率が特に高いプロジェクト、(2)特に低いプロジェクト、(3)その他のプロジェクトの 3 つに分類し、原価率の分布を調べた。図 4 に外部委託率ご

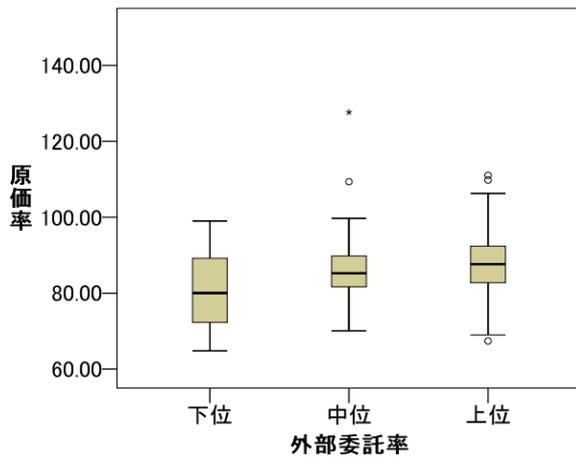


図4 外部委託率ごとの原価率

表4 外部委託率と原価率の中央値

分類	件数	外部委託率	原価率
上位	49	64.3%	87.6%
中位	29	31.2%	85.2%
下位	17	0.0%	80.1%

とのプロジェクトの原価率を示す。ここでは、外部委託率が 0%の（外部委託を行っていない）プロジェクトを「下位」、外部委託率が 50%以上のプロジェクトを「上位」、それ以外の平均的なプロジェクトを「中位」としている。図は外部委託率が高いプロジェクトほど、原価率が高くなる傾向があることを示している。

各グループの外部委託率と原価率の中央値を表4に示す。各グループについて中央値の差の検定を行ったところ、上位と下位の間には有意な差が見られた( $p=0.034$ )。外部委託率の高いプロジェクトでは、委託先とのミーティングや、委託先が作成した成果物の受け入れ検査のような、追加の作業が発生する。また、外部委託を行わなかった場合と比べ、発生した不具合に対する対応に時間がかかってしまうため、プロジェクト全体が遅延する。そのため、外部委託率が高いプロジェクトほど原価率が高い傾向が見られたと考えられる。

#### 4.3 コード再利用率

原価率が 90 未満のプロジェクトにおけるコード再利用率の中央値は 66.7%、90 以上のプロジェクトでは 87.0%であり、予算を超過したプロジェクトでは再利用率が高い傾向が見られた。ただし、図5に示すように、原価率が 90 未満、90 以上のいずれのグループにおいてもコード再利用率の高いプロジェクトが存在し、2つのグループ間に有意差は見られなかった ( $p$  値=0.139)。

より詳細な分析として、図6にコード再利用率

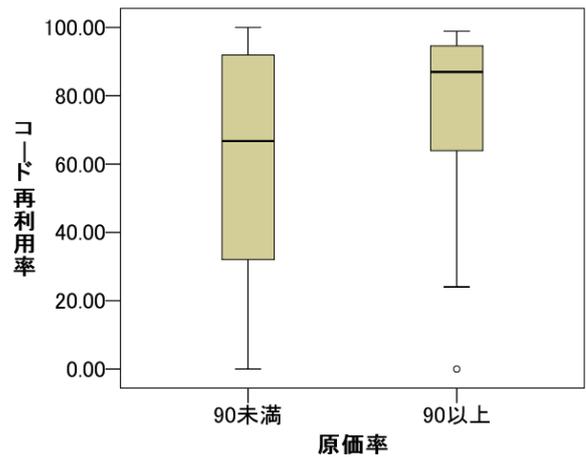


図5 原価率ごとのコード再利用率

を3つのグループに分類した時の原価率を示す。ここでは、再利用の割合に応じてプロジェクトの性質が異なるを考え、コード再利用率が 0%のプロジェクトを「新規開発」、99%以上のプロジェクトを「保守開発」、それ以外を「拡張開発」としている。この図は保守開発と新規開発で原価率が低く、拡張開発の原価率が高いことを示している。また、拡張開発は他と比べ分散が非常に大きく、見積もりから外れやすい傾向があることを示している。各グループのコード再利用率と原価率の中央値を表5に示す。それぞれのグループについて検定を行ったところ、「拡張開発」と「保守開発」のグループに有意な差が見られた ( $p$  値=0.033)。

流用率が極端に高いプロジェクトや低いプロジェクトにおいては、新規に作成されるソースコードと流用されるコードを結合する作業がほとんど発生しない。そのため、再利用に伴う想定外の工数増加が起こらず、予算以内に開発を終えることができ、見積もりとのずれが小さいと考えられる。

一方で、コードの流用率が中程度のプロジェクトでは、再利用に伴う不具合やトラブルが起きやすいため予算超過しやすく、見積もりから外れやすと考えられる。

「拡張開発」に含まれるプロジェクトについてさらに詳細に分析するために0%を超え80%未満、80%以上90%未満、90%以上99%未満の3つに分割した場合のコード再利用率と原価率の中央値を表6に示す。「拡張開発」に含まれるプロジェクトの中でも、コード再利用率が90%以上99%未満のプロジェクトで原価率が最も高い傾向が見られた。

検定の結果、90%以上99%未満の群と「新規開発」、「保守開発」の間に有意差が見られた。分析対象とした企業では、この範囲に含まれるプロジェクトにおいて特に見積もりのずれや再利用に伴うトラブルが発生している可能性があり、流用率

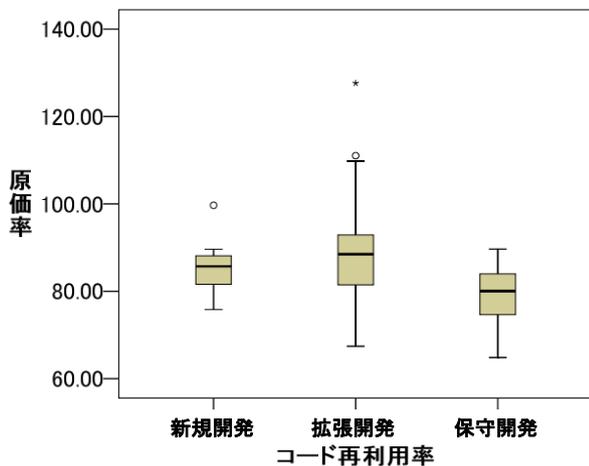


図6 コード再利用率ごとの原価率

表5 コード再利用率と原価率の中央値

分類	件数	コード再利用率	原価率
保守開発	7	99.8%	80.1%
拡張開発	58	77.2%	88.5%
新規開発	10	0.0%	85.7%

表6 拡張開発におけるコード再利用率

分類	件数	コード再利用率	原価率
90%以上 99%未満	17	97.0%	91.0%
80%以上 90%未満	11	87.0%	89.3%
0%超 80%未満	30	57.3%	84.9%

の低い「拡張開発」や、「保守開発」とは異なる傾向があることが分かった。

## 5. おわりに

本稿では、ソフトウェア開発プロジェクトの特徴と予算超過の関係について実際の売上データを用いた統計的な分析を行った。分析の結果、予算を超過したプロジェクトでは、予算内に開発を完了したプロジェクトと比べ外部設計工程に費やす工数の割合が低いことが分かった。

また、詳細な分析の結果、外部委託率が高いプロジェクトは外部委託を行っていないプロジェクトに比べ原価率が高い傾向が見られた。また、コード再利用率が中程度のプロジェクトは、原価率が高く、分散も大きい傾向が見られた。

本稿で行った分析の結果は、特定の企業を対象としたものであり、結果をすぐに一般化するのには困難と考えられる。しかし、対象とした企業と同様に外部設計以降の工程を中心に請負開発を行っている組織の場合、外部に委託する割合の高いプ

ロジェクトや、流用コードが少なからず存在するプロジェクトにおいては、事前にリスク評価を重点的に行うことが必要だと考えられる。

本稿の分析では、プロジェクトの特徴を区別するために、各プロジェクトの完了時に計測された指標の実測値を用いた。今後の課題として、プロジェクト開始時の見積値と完了時の実測値を比較し、原価率の悪化がプロジェクト計画時の特徴によるものか、計画との差異によるものかを明らかにすることがあげられる。

## 謝辞

本研究の一部は、文部科学省「次世代IT基盤構築のための研究開発」の委託に基づいて行われた。

## 参考文献

- [1] 古山恒夫, 菊地奈穂美, 安田守, 鶴保証城, “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,” 情報処理学会論文誌, Vol.48, No.8, pp.2608-2619, 2007.
- [2] Boehm, B.W., “Industrial Software Metrics Top 10 List,” IEEE Software, Vol.4, No.5, pp.84-85, 1987.
- [3] 福島利彦, 山田茂, “プロジェクトリスクを軽減するリスク・マネジメントと定量的分析,” プロジェクトマネジメント学会誌, Vol.8, No.4, pp.31-36, 2006.
- [4] 澤田美樹子, “プロジェクトマネジメントにおける定量的リスク評価,” プロジェクトマネジメント学会誌, Vol.7, No.3, pp.14-19, 2005.
- [5] Wohlin, C. and Andrews, A. A., “Prioritizing and Assessing Software Project Success Factors and Project Characteristics using Subjective Data,” Empirical Software Engineering, Vol.8, pp.285-303, 2003.
- [6] Avritzer, A. and Weyuker, E.J., “Metrics to Assess the Likelihood of Project Success Based on Architecture Reviews,” Empirical Software Engineering, Vol.4, pp.199-215, 1999.
- [7] 水野修, 安部誠也, 菊野亨, “プロジェクト混乱予測システムのベイズ識別器を利用した開発,” SEC journal, Vol.1, No.4, pp.24-35, 2005.
- [8] ISBSG, “The Software Metrics Compendium,” International Software Benchmarking Standards Group, 2002.
- [9] 情報処理推進機構ソフトウェア・エンジニア, “ソフトウェア開発データ白書 2008, SEC データ白書 2008,” 日経 BP 社, 2008