

複数組織データに見るシステムリスクと開発コストの関係

野中 誠¹

概要：IPA/SEC 高信頼性定量化部会では、IT プロジェクトの価格適正化に資する情報提供を目的として、「高度な IT システムの開発には相応のコストが必要である」という仮説を立てて調査を行った。調査では、部会委員の所属組織を対象に、開発対象ソフトウェアのシステムリスクが高い場合と低い場合のそれぞれについて、開発規模当たりの総開発工数、総テスト工数、および総テスト項目数の分布を比較した。その結果、開発規模の影響を吟味する必要があるものの、開発規模当たりの総開発工数と総テスト工数についてはシステムリスクが高い場合の方が工数の平均値が大きくなる傾向が一部確認できた。一方、総テスト項目数については、システムリスクが高い場合と低い場合で平均値に差があるとはいえないことが確認できた。

1. はじめに

ソフトウェア開発コストに最も影響する要因は開発規模である。これは、多くの調査研究が指摘していることであり [1][2][3]、ソフトウェア工学研究において共通認識が得られた記述といえる。

一方で、ソフトウェア開発コストに影響する開発規模以外の要因については共通認識が得られているとはいいがたく、対象の層によって傾向が異なる。例えば、開発コストに影響する要因として、システムの複雑さや開発者のスキル [1]、組織の成熟度レベル [4] などが指摘されている。その一方で、CMM (Capability Maturity Model) レベル 5 を達成した成熟度の高いプロジェクトという層に限定すると、開発コストに影響する要因は開発規模のみであり、複雑さや開発者のスキルなどは有意な要因ではないとした報告もある [5]。

産業界におけるソフトウェア事業という文脈において、ソフトウェア開発コストに影響する要因を実証的に示すには、ソフトウェア事業を営む複数の組織を対象とした調査が求められる。しかし、それには次に述べるいくつかの課題がある。

第一に、プロセスとメトリクスの組織間差異に起因する難しさである。プロセスやメトリクスの標準化とその研究は着実に進展している (例えば [6][7][8])。しかし、単体テスト一つを取り上げても、そのプロセスが意味する内容や含まれる活動は組織によって異なる。開発規模も、改良開発の場合に既存ソフトウェアの規模を含める組織、含めな

い組織、ある係数を乗じた上で含める組織など、その扱いはさまざまである。また、プロセスメトリクスや品質メトリクスは組織に根付いたものであることが多く、ひとたび定着すると定義の変更は容易でない。複数組織のソフトウェア開発データを横断的に分析した調査研究の中には、そのような組織間差異を埋める努力をしたものもあるが [3]、メトリクスの組織依存性に起因する分析結果の揺らぎを十分に解消できているとはいいがたい。

第二に、分析対象の層を均質化することの難しさである。プロジェクトは、元来、唯一無二の独自性という性質をもつ。また、組織によって業務ドメインやビジネスモデルなどが異なるため、分析対象の層を複数組織間で均質化することはほぼ不可能である。同じ組織内でさえ、均質なプロジェクトを集めて分析することは容易なことではない。

第三に、複数組織のデータを収集することの難しさである。生産性やコストなどを含むソフトウェア開発データは、ソフトウェア事業を営む組織の開発能力や品質管理能力を色濃く反映している。また、特に SI プロジェクトの開発データは顧客案件に強く結びついているため、生の開発データのまま組織の外に出すことはほぼ不可能である。

このような難しさがある中、情報処理推進機構ソフトウェア高信頼化センター (IPA/SEC) の高信頼性定量化部会 (筆者が主査) では、『ソフトウェア開発データ白書』 [3] とは別の取り組みとして、複数組織の開発データを横断的に分析し、ソフトウェア開発コストやソフトウェア品質に関わる要因を調査するという取り組みを試みている。その直接の目的の一つは、「高度な IT システムの開発には相応のコストが必要である」という一見すると当たり前と思え

¹ 東洋大学
Toyo University

る記述を、産業界におけるソフトウェア事業という文脈で実証的に示すことである。その背景には、IT プロジェクトに対する低価格化要求が強まっている現状において、この記述の正しさを示すことが IT プロジェクトの価格適正化へと促す力となり、ひいては IT 業界の健全な発展に結びつけていきたいという部会としての思いがある。

以降では、本取り組みで行った複数組織を対象としたデータ収集方法と、分析の結果として得られた知見を述べる。また、類似の分析を『ソフトウェア開発データ白書』のデータに適用した結果もあわせて報告する。

2. 仮説

本取り組みでは、先に述べた記述の「高度な IT システム」の部分で、「システムリスクの高いソフトウェア」に限定して扱っている。ソフトウェアの信頼性や品質を議論するとき、稼働後のシステム停止件数や看過できない故障などに着目し、それらを開発規模で割った不具合密度により評価することが多い [3][9]。しかし、品質要求や信頼性要求の違いに着目して調査した事例は多くないようである。本取り組みでは、信頼性要求の表れとしてシステムリスクの高低に着目している。そして、システムリスクという要因が、ソフトウェア開発コストの影響要因であるか否かを示すことを目的としている。

ソフトウェア開発コストを表すのに、本取り組みでは、規模当たりの総開発工数を用いる。そして、その一部である規模当たりの総テスト工数と、さらにその主たる要因と思われる規模当たりの総テスト項目に着目する。

システムリスクの高いソフトウェアを新規開発または改良開発する場合に、それぞれ、システムリスクの低いときに比べて、次の三つの仮説が当てはまるのではないかと筆者および部会委員は考えた。

- H1: 規模当たりの総開発工数 (総開発工数 / 開発規模) が多くなる。
- H2: 規模当たりの総テスト工数 (総テスト工数 / 開発規模) が多くなる。
- H3: 規模当たりの総テスト項目数 (総テスト項目数 / 開発規模) が多くなる。

これらの仮説は、いずれも開発規模を分母としている。これは、1章の冒頭で述べた通り、開発規模がコストに大きく影響することが明らかであり、その影響を除外するためである。そして、分子にはそれぞれ総開発工数 (H1)、総テスト工数 (H2)、総テスト項目数 (H3) を選んだ。H1 は、規模の影響を除いた上での、本取り組みの目的として示した通りの仮説である。H2 と H3 でテストに関わるメトリクスを選んだ理由は、システムリスクの高いプロジェクトは、テスト対象を絞り込んだリスクベーステスト [10] よりも、広範なテスト対象について品質保証のための網羅的なテストを重視すると考えたためである。そのようなテ

表 1 収集したメトリクス

開発種別 (新規または改良)
開発規模 (新規・変更コード行数)
開発規模当たりの総開発工数
開発規模当たりの総テスト工数
開発規模当たりの総テスト項目数

スト戦略を採用することで、規模当たりのテスト項目数が多くなり、結果的に規模当たりの総テスト工数が増加すると考えた。

3. データ収集のアプローチ

部会委員が所属する 5 社における、SI 事業を営む 6 組織のプロジェクトを対象とした。いずれの組織もソフトウェア開発における定量的管理の経験と実績を有している。

以下に、1章で述べた難しさを乗り越えながら複数組織からデータを収集するために工夫した方法を述べる。なお、本取り組みで収集したメトリクスの一覧は表 1 の通りである。

3.1 メトリクスの組織間差異の扱い

メトリクスの組織間差異を完全に埋めることは非現実的である。今回の取り組みでは、メトリクスの組織間での定義の違いを統一するのではなく、各社で運用されているメトリクスと測定済みデータをほぼそのまま用いた。これは、分析対象の組織がソフトウェアの定量的管理の経験と実績のある組織が多いことから、組織内の範囲であればメトリクスの標準化がなされていることが期待されたためである。そして、それぞれの組織内でデータ分析を行い、その結果を横断的に調査することで知見を得るというアプローチを採用した。

システムリスクについては、文献 [11] を参考に、開発対象システムの信頼性、公共性および社会性の三つの観点から、それぞれ高と低の 2 段階で評定し、「高」の評定が二つ以上ついたシステムを「システムリスクが高い」と判定した。これにより、「システムリスクが高い」または「低い」という 2 値データを作成した。評定は各組織の中で行ってもらい、できるだけ同一の評価者による一貫性のある評定となるようにした。

3.2 分析対象の層の均質化

分析対象の層の均質化を図るために、各組織において、次の条件に合致したプロジェクトを分析対象とするようにした。

- 品質保証部門などがプロセス監査を行う対象とするような、一定の受注金額以上の SI プロジェクト。
- 組織内におけるプロジェクトマネジメント上の問題で、当初計画よりも工数が大幅に膨らんだプロジェクトは対象から除外する。

- 顧客都合により当初計画よりも工数が大幅に膨らんだプロジェクトは対象とする。
- できるだけ同一の業種ドメインに属するプロジェクト、または、システムリスクの高低で分けたときにそれぞれにおいて業種ドメインの偏りが生じないよう標準化作業を工夫する。
- システムリスクの高低で分けたときに、開発規模の偏りが生じないよう標準化作業を工夫する。
- 新規開発と改良開発のそれぞれについて、システムリスクの高低で分けた各セグメントに 10 件以上の標本が得られるようにする。

実際にはすべての条件を満たせなかった組織もあった。ただし、満たしていない条件が一部分のみであれば、そのようなデータも分析対象に含めた。

3.3 複数組織からの情報収集

各組織が保有する生の開発データを、筆者または IPA/SEC が集めて分析することは容易ではない。そのため、次の手順を経てデータを加工し、加工後の情報のみを IPA/SEC が集めた。このように、各組織においてデータ分析作業の一部を担ってもらうことで、情報を収集しやすい状況を作った。

- (1) 新規開発と改良開発のそれぞれについて、システムリスクの高低で分けたデータセットを各組織で用意する。
- (2) 各組織においてデータを吟味し、外れ値を除外する。
- (3) システムリスクの高低のそれぞれについて、開発規模、H1、H2、および H3 に示した値の件数、平均値、五数要約、標準偏差をそれぞれ求める。また、システムリスクの高低を合わせた全体の平均値と標準偏差を求める。
- (4) システムリスクの高低のそれぞれについて求めたこれらの値を、高低を含めた全体の平均値と標準偏差を用いて計算した z 得点に変換する。変換後は、全体の平均値を 0、全体の標準偏差を 1 として標準化した値が得られる。
- (5) システムリスクの高低について、2 群の平均値の差の検定の手法である Welch の検定を行い、両側検定による p 値を求める。
- (6) その際、分布が右に歪んでいることが想定されるので、対数化したデータを用いて分析する。
- (7) 件数、標準化した平均値および五数要約、検定結果の p 値という情報のみを IPA/SEC に提出する。

このようなデータ収集の工夫によって複数組織から情報を集め、標準化された値を共有しながら部会内で議論を進めることができた。そして、データの見直しや再調査が必要な場合には各組織に依頼し、同様の分析手順により得られた情報を提出し直してもらうという作業を複数回繰り返した。また、例えば開発規模の測定方法など、議論にあ

表 2 各組織のデータ概要

種別	組織	標本サイズ		平均値の大小 (高:低)	対数の差の 検定 (p 値)
		高	低		
新規	A	4	15		0.9280
	B	8	13		0.9125
	C	10	10		
	D	15	50	<	0.2587
改良	A	8	15	<	0.1903.
	B	15	8		0.5930
	D	100	100		0.5108
	E	77	13	<	0.1293.
	F	26	64	>	0.0005***

***p < 0.01 **p < 0.05 *p < 0.10 .p < 0.20

たって共有しておくことが望ましい情報は特記事項として提出してもらった。

4. 分析結果

4.1 各組織のデータの概要

表 2 に、5 社 6 組織から得られたデータの概要として、新規開発と改良開発のそれぞれについて、各組織から得られたシステムリスクの高低それぞれの標本サイズと、システムリスクの高低における平均値の大小関係、および対数化した開発規模の差について Welch の両側検定を行ったときの p 値を示す。p 値はその大小に応じて記号を付してあり、0.20 未満の場合は太字で示している。一般に p 値の基準として 0.05 が用いられ、本稿における 0.20 というのはそれに比べて大幅に緩い基準である。しかし、両側検定は片側検定に比べて有意差を示しにくくなることと、層の均質化を図ったとはいえ顧客の関与度合いなど案件ごとの個別の影響が大きいため、0.20 という緩い基準を用いた。また、平均値の大小関係は p 値が 0.50 未満の場合のみ不等号を示した。p 値の空欄は、検定結果の情報が得られなかったことを示す。

表 2 より、新規開発では、システムリスクの高低の違いによって開発規模の平均値に差があるとはいえないという結果が得られた（組織 C を除く）。また、改良開発の組織 A と E は、システムリスクの高いプロジェクトの方が開発規模の平均値が小さく、F は逆にリスクが高い方が開発規模の平均値が大きいう結果が得られた。改良開発の組織 B と D は、開発規模の平均値に差があるとはいえないという結果が得られた。

4.2 H1：開発規模当たりの総開発工数

表 3 に、開発規模当たりの総開発工数 (H1) について、平均値の大小関係と、対数化した上での平均値の差の検定結果を示す。組織 C からは対数化していない値が提出されたため、他と区別するために括弧で示している。

表 3 より、新規開発については組織 B を除く 3 組織で、システムリスクの高いプロジェクトの方が総開発工数の平

表 3 開発規模当たりの総開発工数

種別	組織	平均値の大小 (高:低)	対数の差の検定 (p 値)
新規	A	>	0.0387**
	B	>	0.4504
	C	>	(0.0000***)
	D	>	0.0388**
改良	A	>	0.1153*
	B	>	0.2462
	D	>	0.2802
	E	>	0.1043.
	F	<	0.2774

***p < 0.01 **p < 0.05 *p < 0.10 .p < 0.20

均値が大きくなるという結果が得られた。これらの p 値は十分に小さな値であり、平均値に差があるという傾向が強く表れている。表 2 において、新規開発の組織 A および D では開発規模の平均値に差があるとはいえないことから、組織 A および D については仮説 H1 を支持する結果が得られたといえる。組織 C は開発規模の情報が見えないため断定はできないが、p 値の小ささから仮説 H1 を支持する結果が得られた可能性が高いと思われる。

一方、改良開発については、組織 A および E の 2 組織で、システムリスクの高いプロジェクトの方が総開発工数の平均値が大きくなるという結果が得られた。ただし、組織 F では、総開発工数の平均値に有意差があるとはいえないものの、平均値の大小関係が逆転している。表 2 を見ると、組織 F はシステムリスクの高いプロジェクトの方が開発規模の平均値が大きい。逆に、組織 A と E ではシステムリスクの高いプロジェクトの方が開発規模の平均値が小さい。これらの二つの表を見比べると、開発規模の平均値が大きいプロジェクトほど、総開発工数の平均値が小さいという結果が得られたように読み取ることもできる。

伝統的なソフトウェア工学の議論では、開発規模が増えたと開発者のコミュニケーションパスが増大したり [12]、開発プロセスにおけるアクティビティが増加したりすることにより開発の生産性が低下する [13] と言われている。しかし、改良開発の開発規模 (表 2) と規模当たりの総開発工数 (表 3) において、組織 A, E, F から得られた結果はその逆を示しているように読み取れる。これについては、規模当たりの総テスト工数の結果と合わせて 5 章で考察する。

このように、特に改良開発においては開発規模による影響を排除しきれておらず、さらに詳しく評価する必要がある。しかし、全体としては、システムリスクの高いプロジェクトは、低いものに比べて開発規模当たりの総開発工数の平均値が大きくなるという傾向を一部読み取ることができた。このことから、仮説 H1 を部分的に支持する結果が得られたと判断する。

表 4 開発規模当たりの総テスト工数

種別	組織	平均値の大小 (高:低)	対数の差の検定 (p 値)
新規	B	>	0.1431.
	C	>	(0.0665*)
	D	>	0.1108.
	改良	B	>
D			0.6382
E		>	0.0217**
F			0.7756

***p < 0.01 **p < 0.05 *p < 0.10 .p < 0.20

表 5 開発規模当たりの総テスト項目数

種別	組織	平均値の大小 (高:低)	対数の差の検定 (p 値)
新規	B		0.8240
	C		(0.9283)
改良	B	>	0.4980
	D		0.8838
	E	<	0.2513
	F	>	0.4584

***p < 0.01 **p < 0.05 *p < 0.10 .p < 0.20

4.3 H2: 開発規模当たりの総テスト工数

表 4 に、開発規模当たりの総テスト工数について、同様の分析を行った結果を示す (組織 C の括弧は非対数化を表す)。新規開発については、組織 B, C, D の 3 組織すべてにおいて、システムリスクの高いプロジェクトの方が低い方に比べて総テスト工数の平均値が大きくなるという結果が得られた。一方、改良開発については、組織 B および E の 2 組織において、同様の結果が得られた。H1 の議論と同様、これらも開発規模による影響を排除しきれていないものの、仮説 H2 を支持する結果が得られたと判断する。

4.4 H3: 開発規模当たりの総テスト項目数

表 5 に、開発規模当たりの総テスト項目数について、同様の分析を行った結果を示す (組織 C の括弧は非対数化を表す)。新規開発については組織 B と C の 2 組織しかデータが得られなかったが、いずれの組織においても、システムリスクの高いプロジェクトの方が低い方に比べて総テスト項目数の平均値が大きくなるとはいえないという結果が得られた。また、改良開発についても、統計的に有意な差があるとはいえないという結果が得られた。H1 および H2 の議論と同様、こちらも開発規模による影響を排除しきれていないものの、仮説 H3 を支持する結果は得られなかったと判断する。

4.5 IPA/SEC データ白書での業種間比較

2 章の仮説とは直接に結びつかない項目だが、『ソフトウェア開発データ白書 2014-2015』 [3] に記載の開発データを業種ごとに分けて、類似の分析を行った。ここでは、『データ白書 2014-2015』に含まれる合計 3541 プロジェクト

表 6 『IPA/SEC データ白書 2014-2015』での業種間比較

種別	メトリクス	業種	平均値	p 値
新規	開発規模	情報通信業	小	0.0000***
		金融・保険業	大	0.0013***
	総開発工数	情報通信業	大	0.0595*
		金融・保険業	大	0.0000***
	総テスト工数	情報通信業	大	0.0000***
		金融・保険業	大	0.0009***
総テスト項目数	情報通信業	大	0.0068***	
	金融・保険業	大	0.21	
改良	規模	情報通信業	小	0.0000***
		金融・保険業	小	0.115.
	総開発工数	情報通信業	大	0.0003***
		金融・保険業	大	0.0003***
	総テスト工数	情報通信業	大	0.0000***
		金融・保険業	大	0.456
	総テスト項目数	情報通信業	大	0.0078***
		金融・保険業	大	0.403

***p < 0.01 **p < 0.05 *p < 0.10 .p < 0.20

トのうち、条件を満たした新規開発 632 プロジェクト、改良開発 659 プロジェクトを対象にして、これまでの分析における「組織」の代わりに「業種」を指定して分析した。表 6 にその結果を示す。表 6 の平均値の欄は、それぞれの全体平均に対して各セグメントの平均値の大小を示している。例えば、表 6 の新規・規模・情報通信業は、このセグメントの平均値が全体平均に比べて小さかったことを示している。また、これまでと同様に、対数化した各メトリクスの差について両側検定を行ったときの p 値を示している。

表 6 より、情報通信業では、規模当たりの総開発工数、規模当たりの総テスト工数、および規模当たりの総テスト項目数において、新規開発と改良開発のいずれも、他の業種の全体平均に比べて平均値が大きいという結果が得られた。また、金融・保険業では、これらのうちのいくつかのメトリクスにおいて同様の結果が得られた。このことから、情報通信業と金融・保険業のプロジェクトは、他の業種に比べてソフトウェア開発コストがかかることを示唆している。ただし、『データ白書』はデータ提供企業によるプロジェクト選択の恣意性を排除できず、また、表 6 にあるとおり開発規模の平均値にも有意差が見られる。この点に留意した上で、この結果を捉える必要がある。

5. 考察

4 章で見てきたように、規模当たりの総開発工数 (H1) については仮説を部分的に支持するという結果が得られた。規模当たりの総テスト工数 (H2) については仮説を概ね支持するという結果が得られた。しかし、規模当たりの総テスト項目数 (H3) については仮説を支持するという結果は得られなかった。以下、この結果について考察する。

総テスト工数は総開発工数の一部なので、総開発工数に

は総テスト工数に比べてより多くの変動要因が影響する。そのため、総テスト工数に関する仮説 H2 が支持されたとはいえず、総開発工数に関する仮説 H1 が同様に支持されるとはいえない。したがって、これらの結果は妥当と考えることができる。

一方で、2 章で述べた通り、システムリスクの高いプロジェクトの方が低い方に比べて規模当たりの総テスト工数が増える理由として当初は次を考えていた。「システムリスクの高いプロジェクトの方がリスクベードテストよりも網羅的なテストが行われる傾向があるため、テスト項目数が増える。その結果として、総テスト工数も増加する。」しかし、仮説 H3 が支持されなかったことにより、この論理を積極的に支持する結果が得られたとはいえない。

総テスト工数に関する仮説 H2 が支持され、総テスト項目数に関する仮説 H3 が支持されなかった理由の一つとして、開発規模の影響が挙げられる。開発規模の大きいプロジェクトでは、その機能要求やそれらの組合せが膨大になるため、限られた納期と予算の範囲では網羅的なテストをすべてやり切ることができず、結果として、リスクベースドテストで対応せざるを得ない状況に追い込まれることがあり得る。本取り組みでは、開発コストに影響する要因としてシステムリスクに着目してきたが、納期と予算の圧力の前には、開発規模の大きさがテストの行動に強く影響しているのかもしれない。この仮説の妥当性を検証するデータを現時点では持ち合わせていないが、そうだとすれば、IT システムの価格と納期の適正化に向けてより強く働きかける必要がある。同時に、品質を追求することでコストを低く抑えるという品質管理の基本的な考え方を実践する努力が求められる。

6. おわりに

本稿では、IPA/SEC 高信頼性定量化部会の活動として行っている調査の一部を紹介した。「高度な IT システムの開発には相応のコストが必要である」という IT 業界の健全な発展のために必要な仮説について、調査を進める上での難しさを克服しながら、複数組織のデータを通じた客観的な分析を試みた。調査では、開発対象ソフトウェアのシステムリスクが高い場合と低い場合のそれぞれについて、開発規模当たりの総開発工数、総テスト工数、および総テスト項目数の分布を比較した。その結果、開発規模の影響を吟味する必要性があるものの、開発規模当たりの総開発工数と総テスト工数についてはシステムリスクが高い場合の方が工数の平均値が大きくなる傾向が一部確認できた。一方、総テスト項目数については、システムリスクが高い場合と低い場合で平均値に差があるとはいえないことが確認できた。

ソフトウェア開発コストの影響要因を分析するにあたって、手元のデータに対して多変量解析や機械学習などの手

法を適用し、試行錯誤を繰り返しながら要因を分析するのが一般的である。しかし、産業界におけるソフトウェア事業という文脈において、複数組織のデータを手元に集めて横断的に分析することは難しい。本取り組みにおけるデータ収集のアプローチは、高度な統計解析を行うのは難しいものの、データ提供組織の協力が得られれば、横断的な情報を比較的容易に集めることが期待できる。

今後の課題として、このような調査に協力してもらえる組織を増やし、より広い範囲で情報を集めることが挙げられる。本稿では5社6組織のデータしか得られなかったが、より多くのデータを集めることで、客観性の高い分析結果を示すことができる。そのためには、データ保有企業が容易に参加できるような仕組み作りが必要であり、本部会で継続検討したいと考えている。

謝辞 本取り組みを進めるにあたり、IPA/SECの山下博之氏、三縄俊信氏、佐伯正夫氏、塚元郁児氏には検討の段階からデータの収集と整理に至る過程で多大なご尽力をいただいた。また、IPA/SEC高信頼性定量化部会の委員の方々には、データの提供や得られた結果に対する議論などで多大なご貢献をいただいた。関係各位に対してここに謝意を表す。

参考文献

- [1] Boehm, B., Abts, C., Brown, A., Sunita, C., Clark, B., Horowitz, E., R., M., Reifer, D. and Steece, B.: *Software Cost Estimation with COCOMO II*, Prentice-Hall (2000).
- [2] Jorgensen, M. and Shepperd, M.: A Systematic Review of Software Development Cost Estimation Studies, *IEEE Transactions on Software Engineering*, Vol. 33, No. 1, pp. 33-53 (2007).
- [3] ソフトウェア高信頼化センター：ソフトウェア開発データ白書 2014-2015, 情報処理推進機構 (2014).
- [4] Clark, B.: Quantifying the Effects of Process Improvement on Effort, *IEEE Software*, Vol. 17, pp. 65-70 (2000).
- [5] Agrawal, M. and Chari, K.: Software Effort, Quality, and Cycle Time: A Study of CMM Level 5 Projects, *Software Engineering, IEEE Transactions on*, Vol. 33, No. 3, pp. 145-156 (2007).
- [6] ソフトウェア・エンジニアリング・センター：共通フレーム 2013～経営者、業務部門とともに取組む「使える」システムの実現～, 情報処理推進機構 (2013).
- [7] 野中誠：ソフトウェア品質の定量的管理における曖昧さ - ソフトウェア欠陥測定の実践 - , 経営論集, Vol. 76, pp. 99-109 (2010).
- [8] ソフトウェア・エンジニアリング・センター：組込みソフトウェア開発における品質向上の勧め [バグ管理手法編], 情報処理推進機構 (2013).
- [9] Cusumano, M., MacCormack, A., Kemerer, C. F. and Crandall, W.: Software Development Worldwide: The State of the Practice, *IEEE Software*, Vol. 20, pp. 28-34 (2003).
- [10] SQuBOK 策定部会編：ソフトウェア品質知識体系ガイド - SQuBOK ガイド - , オーム社 (2007).
- [11] ソフトウェア・エンジニアリング・センター：続定量的

品質予測のススメ～ITシステム開発における定量的品質管理の導入ノウハウと上流工程へのアプローチ～, 情報処理推進機構 (2011).

- [12] Pressman, R. and Maxim, B.: *Software Engineering: A Practitioner's Approach*, McGraw-Hill (2014).
- [13] Jones, C.: *Estimating Software Costs: Bringing Realism to Estimating*, 2nd ed., McGraw-Hill (2007).