

序 文

本書の初版から第2版にいたる間に、コンピュータ産業とソフトウェアコスト見積りの技術に多くの変化が起きた。

10種類もの「アジャイル」開発アプローチを含め、20以上もの新しい開発手法が現れた。オブジェクト指向（OO）手法はさらに広がりを見せ、ユースケースやUMLがソフトウェア設計手法の主流に加わった。Software Engineering Institute（SEI）は新しい能力成熟度—統合化モデル（CMMI）^{（訳注1）}を公表した。これらの新しいアプローチのすべては、正確なコストとスケジュールの見積りが生命線となるソフトウェアプロジェクトで用いられている。

それらの新しい手法の中には見積りと測定の新しい尺度を作り出したものもある。したがって2007年時点では、ソフトウェアプロジェクト管理者はソースコード行数やファンクションポイント尺度だけでなく、COSMIC ファンクションポイント、エンジニアリングファンクションポイント、オブジェクトポイント、ストーリーポイント、ウェブオブジェクトポイント^{（訳注2）}、ユースケースポイントおよびその他の数多くの尺度について理解しなければならない。

しかしながら、これら新しい尺度はまだ実験段階であり、大規模な経験データは存在していない。標準ファンクションポイントは25,000以上ものプロジェクトの測定に使われているのに対し、他の尺度は合計しても300以下のプロジェクトでしか用いられていない。

見積手法については活発な研究が行われており、不正確な見積りの原因についても分析が進んでいる。見積りの誤りはまだ一般的に見られるが、いまや我々は

（訳注1） 今までCapability Maturity Model Integration：CMMIの日本語訳としてIPAなどで用いている「能力成熟度モデル統合」が用いられてきているが、本書ではあえて「能力成熟度—統合化モデル」を用いる。

（訳注2） これらの尺度については本文参照。

vi 序 文

以下のようにその原因を理解している。

- ソフトウェアプロジェクトは開発期間を通して約2%／月の割合で規模が増大する。
- バグまたは欠陥の多発は、テストスケジュールの遅延を引き起こす。
- 文書の作成コストはソースコードの作成コストをしばしば上まわる。
- 楽観的見積りが、恣意的に正確な見積りに取って代わる事態が頻繁に見られる。

いまや我々はソフトウェアコストの主要な誤りの源を知っており、それらを制御できるところにまで到っている。上記のソフトウェア見積りの3つの主要な誤りの技術的発生源は現実に除去することができる。しかし、第4の楽観的見積りの横行は、いまだに厄介である。

ソフトウェア見積りは作成後、開発ソフトウェアの依頼者および上級マネジメントによって承認されなければならない。依頼者あるいは上級マネジメントが急なビジネスニーズに迫られている場合には、正確な見積りを拒絶する強い傾向がある。そのことゆえに、大規模あるいは複雑なソフトウェアアプリケーションの構築は時間がかかり、高価なものになってしまっている。

したがって、チームと技術能力に基づく正確なコスト見積りに基づいてソフトウェアプロジェクトを開始する代わりに、多くのプロジェクトは外部からのビジネス的側面から決められる期限を守ることを強いられる。それらのプロジェクトはまた、すべての機能を含めるのに必要な予算を獲得できない。これらはひとえに社会学的な課題であって解決が難しい。

正確な見積りが別のものと置き換えられるのを防ぐためには、類似のプロジェクトの経験データの利用が必要である。もし、特定種類のプロジェクトが過去に正規の見積りより短期あるいは安価に構築されたことがないことが経験データによって裏付けられるなら、見積りは置き換えを免れることができる。しかし、確かな経験データがない限り、見積りはおそらく拒絶に遭い、別のものと置き換えられるだろう。したがって、ベンチマーキングや正確な経験データは正確な見積りのための重要な前提である。

筆者は、1971年以來、過去のデータを集めソフトウェアコスト見積ツールを設計し開発することを主要な仕事としてきた。実は、筆者の見積りに関する仕事は、IBMにおいてソフトウェアコストの影響要因の分析を指示されたときに始まった。IBMでコストデータを1年ほど集め、ソフトウェアコストに関する社

外の文献を調査するうちに、IBMのシステムソフトウェア開発プロジェクトについて、工数、スケジュール、コストの自動見積ツールを構築することが可能であるように思えてきた。それは、要求定義、設計、設計レビュー、コーディング、コードインスペクション、テスト、ユーザ用文書作成、プロジェクト管理などの見積りを含むものであった。

ソフトウェアコスト見積ツール開発の提案はIBMの上級管理者に受け入れられ、1972年に開発が始まった。それ以来、私は10本以上のコスト見積ツールを設計・開発し、個別の企業や見積ツールの市場に提供してきた。本書の目的は、ソフトウェアコストモデルを作成するために必要な情報の種類を明らかにし、コスト見積ビジネスに携わる者から見た視点を示そうとするものである。

本書は、ソフトウェアコスト見積りの基本的な課題を論じようとするものである。今まで見積りおよび見積尺度に関する他の多くの専門家がアイデアや情報を提供してきており、本書ではそれらを繰り返し引用している。これらの専門家の中にはAllan Albrecht, Dr. Barry Boehm, Frank Freiman, Dr. Randall Jensen, Steve McConnell, Larry Putnam, Dr. Howard Rubin, Charles Symonsなどが含まれるが、我々は競合関係にあるので、占有する情報をそれぞれに共有することは少ない。

筆者も、またこれらの競合する専門家も、正確なコスト見積りは世界経済にとっても非常に重要であると考えている。あらゆるソフトウェアプロジェクト管理者、ソフトウェア品質保証専門家はもとより、多くのソフトウェア技術者、システムアナリスト、プログラマは、ソフトウェアコスト見積りの基本的な概念を理解すべきである。このことは市販のソフトウェアコスト見積ツールのベンダに共通した見方である。

ソフトウェアコスト見積ビジネスにかかわりを持つ我々のような専門家は、多くの手作業による見積アルゴリズムを熟知しているし、小規模プロジェクトにおいては必要に応じて手作業による見積手法を用いてもいる。しかし、手作業による見積りが大規模ソフトウェアプロジェクトのライフサイクルに対する見積りとして十分だとはだれも考えていない。もし手作業による手法で十分ならば、市販のソフトウェアコスト見積ツールなどは存在しないはずである。

開発遅延やコスト超過など、ソフトウェアに関連する悲劇のほとんどは、不注意で楽観的な手作業のコスト見積りに起因していることは、はなはだ興味深い。COCOMO II, GECOMO, SLIM, PRICE-S, ProQMS, SEER, SoftCostや当社のツール(SPQR/20, CHECKPOINT, KnowledgePlan)などの正規の見積ツールを用いるプロジェクトは通常、予算内に納まり、深刻な問題なしに完了することが格

段に多いことが立証されている。

大規模システムで手作業による見積りが失敗に帰す理由は、「複雑さ」の一言で片付けることができる。ソフトウェアプロジェクトの成果を決定する要因は何百もあり、単純なアルゴリズムや手作業によっては、これらの要因の組合せを考えるのは無理である。

本書では、複雑で多様な見積りの問題をコスト見積ツールとソフトウェアプロジェクト管理ツールの側面から眺めている。小規模システムでの手法についても論じているが、大規模システムの見積りの問題が主な対象テーマである。

現在、プロジェクト管理機能の中で自動化のサポートがあるのは見積りだけではない。本書では、他の多くのプロジェクト管理ツールとの関係で見積ツールを眺め、さらに追加のツールの必要性について論じている。

ソフトウェアコスト見積ツールは、コスト見積り、品質見積り、スケジュール計画、手法またはプロセス管理、リスク分析、マイルストーン追跡等を含むソフトウェアプロジェクト管理ツールセットの重要な一部となっている。

これらのツールは、ワードプロセッサの世界でスプレッドシート、データベース、グラフィックツールなどと組み合わせて既の実現しているような洗練されたシームレスな統合化レベルにはまだ至っていないが、先は見えてきている。

ソフトウェアコスト見積りは非常に多くの因子の値と何百もの調整要因を含む複雑な課題であるから本書の構造も複雑であり、6部で構成している。

第1部は導入部で、ソフトウェアコスト見積ツールの機能を調べる。一部ではツールにどのようなコストがかかり、どのような価値を作り出すかについて、ソフトウェアプロジェクト管理のビジネス的視点から論じる。ここでは読者の事前知識の有無は問わない。

第2部では、プロジェクトの要求が完全に理解される以前のごく早期の見積りを行う手法について取り扱う。部分的な知識による早期の見積りは、見積りの形式の中で最も難しいものの1つであるが、最も重要なものでもある。早期の見積りが変更できない目標とされたり、あるいはプロジェクトの公式の見積りになってしまうことがあまりにもしばしば起こる。

ここでは経験則を用いた手作業による見積手法と併せて、市販のソフトウェア見積ツールが提供するより洗練された予備的な見積手法について論じる。

第3部では、種々のソフトウェア成果物の規模予測を取り扱う。すべての市販のソフトウェア見積ツールは何らかの形の規模情報を必要とし、規模の取扱いには驚くほど多くのやり方が存在する。

本書を執筆している時点では、ソフトウェア見積りの世界ではファンクション

ポイント尺度による規模見積りが支配的となってきたが、ソースコード行数による規模予測やさらに抽象的な作業についての規模予測も行われている。

「ユースケースポイント」や「オブジェクトポイント」などの新しい実験的尺度もある。しかし、それらにはまだ十分な経験データがない。オブジェクト指向(OO)の分野では数多くの特化した尺度もある。ただし、それらはOOプロジェクトでは有効であっても他の尺度を用いるプロジェクトとの比較はまだできない。

第4部では、ソフトウェアコスト見積ツールの調整要因の取扱いについて示す。ソフトウェアプロジェクトの成果に影響を与える要因には、チーム能力、超過勤務、手法、ツール、およびオフィススペースや環境などを含む100以上の項目がある。

市販のソフトウェアコスト見積ツールは多くの重要項目に関してデフォルト値を提供しているが、不確実性の範囲は非常に広いので、ユーザは最終結果に大きな影響を与える報酬レベルや、間接経費、要員の経験レベルなどの重要なパラメータについては、一般的な「業界平均」を自らの企業の経験に基づく特定の値に置き換える必要がある。

第5部では、多くのソフトウェアプロジェクトで発生する以下の10のアクティビティについてのコスト見積りの原理を説明する。

- 要求の収集および分析
- プロトタイピング
- 設計
- 正規の設計インスペクション
- コーディング
- 正規のコードインスペクション
- 変更管理または構成管理
- ユーザ用文書作成
- テスト
- プロジェクト管理

無論、もっと多くのアクティビティが存在するが、これらの10のアクティビティは非常に多くのプロジェクトで見られるので、これらに対する見積りが正確でないかぎり全体のプロジェクトレベルでの見積精度は保てない。

第6部では、21種類の保守および機能拡張のアクティビティベースのコスト見積りの原理を取り扱う。保守の見積りは開発の見積りよりもはるかに複雑である。なぜなら、対象とするアプリケーションの年代と構造の影響力が大きいから

x 序 文

である。また、欠陥多発モジュールの除去、フィールドサービス、再現不能欠陥などが発生した場合には、コストに非常に大きな影響を及ぼすので、多くの特殊な種類の保守見積りがある。

第6部では何百ものアプリケーションに影響する著しくコストのかかる特定日付の変更や夏時間の変更などの新しい形式の保守についても論ずる。

- Capers Jones