



Title	ファンクションポイント法を応用した早期見積技法の提案とそのシステム化
Author(s)	原田, 晃; 幕田, 行雄; 石川, 貞裕 他
Citation	電子情報通信学会論文誌D. 2006, J89-D(4), p. 755-766
Version Type	VoR
URL	https://hdl.handle.net/11094/26484
rights	copyright©2006 IEICE
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

ファンクションポイント法を応用した早期見積技法の提案とそのシステム化

原田 晃[†] 幕田 行雄^{††} 石川 貞裕^{†††} 大野 治^{††††}
 楠本 真二^{†††††} 井上 克郎^{†††††}

Early Software Size Estimation Method Based on Function Point

Akira HARADA[†], Ikuo MAKUTA^{††}, Sadahiro ISHIKAWA^{†††}, Osamu OHNO^{††††},
 Shinji KUSUMOTO^{†††††}, and Katsuro INOUE^{†††††}

あらまし 業務ソフトウェアの開発では、機能仕様の明確になっていない開発の前段階で開発費、開発期間を見積ることが重要であると考えられている。開発費、開発期間は、開発対象の業務ソフトウェアの規模を見積り、それに信頼性等のシステム特性を加味して算出する。最近、業務ソフトウェアの規模を、ファンクションポイントという機能量で見積るファンクションポイント法が普及してきた。しかし、開発の前段階でファンクションポイント法により見積りを行うには、いくつかの困難を伴う。本論文では、開発の前段階でも、業務ソフトウェアの規模を見積ることができる、ファンクションポイント法を応用した早期見積技法の提案と、そのシステム化について報告する。多数のプロジェクトに適用し評価したところ、非常に高い精度で、しかも簡単に、業務ソフトウェアの規模を見積ることができるという結果を得た。

キーワード 見積り、ファンクションポイント、FP、プロジェクト管理、業務ソフトウェア

1. ま え が き

インターネットや Web の普及拡大に代表されるように情報化社会の進展は著しく、それに伴い、業務ソフトウェアの開発においても大規模化、高機能化、短納期化、低コスト化の要求が急速に高まってきている。

これを実現するためには、プロジェクトの開発計画を立て、開発計画どおりに開発が進むようにプロジェクト管理を行うことが必要である [1] ~ [3]。開発計画の中でも、開発費、開発工数、開発期間は、特に重要であると考えられており、これらを予測することを見積りという。通常、開発費、開発工数と開発期間は、まず対象となる業務ソフトウェアの開発規模を見積り、それに品質や性能等の要素を加味して予測されることが多い [4] ~ [6]。

従来、ソフトウェアの規模はプログラム行数 (SLOC: Software Lines Of Codes) で測定されてきた。しかし、プログラム行数を尺度とする場合、開発言語に依存、開発者の技術レベルに依存、再利用可能プログラムの行数の扱い方の不統一等の課題が多く、最近では、ソフトウェアの機能量をファンクションポイント (以後、FP という) という尺度によって定量的に計測する、ファンクションポイント法 (以後、FP 法という) が普及してきており、特に業務ソフトウェアでは FP 法が標準になりつつある [7]。

FP 法は 1979 年に Albrecht によって提案された [8]。これはユーザの視点から機能量を定量化するため、開

[†] 日本電子計算株式会社, 東京都
 IT Planning and Support Division, Japan Information Processing Service Co., Ltd., 2-4-24 Toyo, Koto-ku, Tokyo, 135-8388 Japan

^{††} 日立ソフトウェアエンジニアリング株式会社, 東京都
 Process Improvement Technology Center, Hitachi Software Engineering Co., Ltd., 4-12-7 Higashishinagawa, Shinagawa-ku, Tokyo, 140-0002 Japan

^{†††} (株) 日立製作所情報・通信グループ生産技術本部, 川崎市
 Information and Telecommunication Systems, Engineering Support Division, Hitachi, Ltd., Hitachi Systemplaza Shinkawasaki 890 Kashimada, Saiwai-ku, Kawasaki-shi, 212-8567 Japan

^{††††} (株) 日立製作所情報システム事業部, 東京都
 Information Technology Division, Hitachi, Ltd., 6 Kanda-Surugadai 4-Chome, Chiyoda-ku, Tokyo, 101-8010 Japan

^{†††††} 大阪大学大学院情報科学研究科, 豊中市
 Department of Computer Science, Graduate School of Information Science and Technology, Osaka University, 1-3 Machikaneyama-cho, Toyonaka-shi, 560-8531 Japan

発言語や実装方法に影響されない値を得ることができ、優れた見積技法であると考えられている。

FP 法による業務ソフトウェアの規模見積りの時点には、企画段階、開発の前段階、要求要件設計終了段階等があるが、プロジェクトの開発計画を立てるためには、開発の前段階での規模見積りが必要である。ところが、この段階では発注者から提案依頼書 (RFP : Request For Proposal) が提示されるのみで、実装すべき機能仕様が詳細には定まっておらず、FP 法による規模見積りを実施するには、いくつかの困難を伴う。この課題を改善したものととして NESMA 法 [11]、ユースケースポイント法 [12]、電中研法 [13] 等いくつか考察されている。

本論文では、開発の前段階でも FP を計測できる、要素見積法と呼ぶ見積技法を提案する。また、この見積技法を実装した見積支援システム (AP-Estimate) について説明する。

画面上の 1 個の GUI ボタンや 1 回のファンクションキー押下で実行される、ユーザ視点での入出力の最小単位を要素機能と呼ぶ。要素見積法では、16 種類の要素機能をあらかじめ定義しておき、対象となるソフトウェアに、これらの要素機能が、それぞれいくつ含まれているかを計測して FP を算出するものである。

これまで約 200 プロジェクトで、開発の前段階で AP-Estimate を用いた規模見積りを実施しており、精度の高い結果を得ている。

本論文では、2. で IFPUG 法の概要と課題を、3. で要素見積法を、4. で AP-Estimate の実現方式を、5. で適用事例を、6. で評価を行い、7. ではまとめを述べる。

2. 業務ソフトウェアの見積技法

2.1 業務ソフトウェアのライフサイクルと見積りの時点

(1) ソフトウェアライフサイクル

図 1 に示すように、業務ソフトウェアの開発では、開発を委託する側と、委託を受けて開発する側の二つの組織が考えられる [14] に倣って、開発を委託する側を購入者、開発を担う側を供給者と呼ぶことにする。購入者は新しく開発する業務ソフトウェアの企画を行った後、高品質のソフトウェアを低コストかつ短時間で開発できる供給者を選んで、開発を委託することが多い。そのために、購入者は開発するソフトウェアの要件を記載した RFP を作成し、複数の供給者に提案書の提出を依頼する。供給者は、その要件を満たすソフトウェアのアーキテクチャ、開発するためのコスト、期間等を記載する提案書を作成し、購入者に提出する。購入者は、複数の供給者から提出された提案書の内容、供給者の能力を評価して、供給者を決定し、開発を委託する。

委託を受けた供給者はプロジェクトを組織し、プロジェクト計画を作成して、開発を進めていくことになる。

開発は、要求分析、業務要件設計、ソフトウェア方式設計、ソフトウェア詳細設計、プログラミング&単体テスト、組合テスト、総合・システムテストというように複数の工程に分割され、この順序で行われていく。

このような開発プロセスは [14] で標準化されており、上記の七つの工程も標準化された開発プロセスと対応を付けることができる。

(2) 見積りの時点

ソフトウェアライフサイクルの様々な時点で見積りを行うことは、企画の作成や、開発を進めていく上で

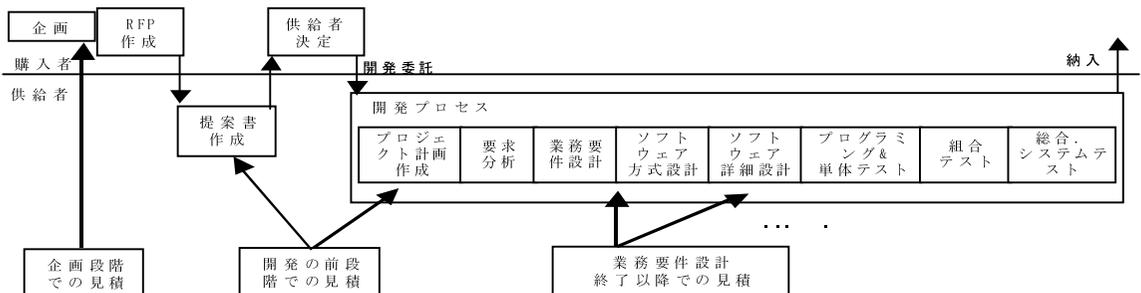


図 1 ソフトウェアライフサイクルと見積り時点の関係
Fig. 1 Relation of software life cycle and the timing for software sizing.

の改善に役立つが、特に、図 1 に示す次の三つの時点での見積りが重要であると考えられる。

(a) 企画段階での見積り

予算化等を目的に開発費、開発期間を算出するために行う見積りである。企画段階では、機能要件が明確になっていないことが多く、ここで見積る開発費は非常に粗い概算というべきものになる。

(b) 開発の前段階での見積り

図 1 に示すように、提案書の作成時点やプロジェクト計画作成時点等の開発の前段階で、開発費、開発工数、開発期間を算出するために行う見積りである。購入者が作成した RFP に記載されている要件をもとに見積りを行う。RFP には、業務要件、ユーザ要件、機能要件が記載されている。業務要件とは、購入者である組織が、ソフトウェアを購入する理由や達成したいと望む目的を表す。ユーザ要件とは、ユーザがソフトウェアを使って実行できなければならない目標や仕事を表す。機能要件は、ユーザが仕事を遂行して業務要件を満たすために、ソフトウェアが備えていなければならない機能を表す。すなわち、RFP には、実現してほしいことは記載されているが、それをどのように実現するかは記載されていない。どのように実現するかは、開発プロセスの中で明確にされていく。したがって、RFP には、ユーザの観点から直接、必要となる機能が記載されているが、その機能を実現するために必要となる二次的な機能について明確になっていないことが多く、精度の高い見積りを行うことは難しい場合が多い。ここで見積る開発費、開発工数、開発期間は企画段階のものよりも精度は高いものの、概算というべきものになる。

(c) 業務要件設計終了以降での見積り

購入者、供給者間の契約内容の見直しや、プロジェクトの開発計画をより詳細化すること等を目的に、図 1 に示すように、業務要件設計やソフトウェア方式設計完了時に、開発するソフトウェアで実現する機能、性能、信頼性等の機能仕様が明確になった時点で行う見積りである。高い精度での見積りが可能な場合が多い。なお、今後、本論文では、業務ソフトウェアを単にソフトウェアと呼ぶことにする。また、見積りとはソフトウェアの規模を見積ることを指すことにする。

2.2 FP 法の概要と課題

FP 法には、IBM 法 [9]、IFPUG 法 [10] 等、数十種類の計測方法があるが、現在では IFPUG 法が主流となっており、本論文では IFPUG 法を用いる。

(1) IFPUG 法の概要

IFPUG 法には、開発するソフトウェアの規模を表すアプリケーション FP、新規にソフトウェアを開発するプロジェクトの規模を知るために使用する新規開発プロジェクト FP、既存ソフトウェアを機能改良するプロジェクトの規模を知るために使用する機能改良プロジェクト FP の 3 種類がある。本論文は、ソフトウェアの規模を計測することを対象としており、アプリケーション FP について説明する。また、アプリケーション FP はシステム特性を反映したものであるが、本論文ではシステム特性を反映していない未調整 FP のみを扱う。IFPUG 法では、未調整 FP を次の手順で計測する [10]。なお、説明の中に出てくるアプリケーションとはソフトウェアと同じ意味である。

(Step 1) アプリケーション境界と計測範囲の設定

FP を計測する範囲を明確にするために、計測対象のアプリケーション境界 (ソフトウェアの境界) を設定する。アプリケーション境界を今後、計測境界と呼ぶ。

(Step 2) データファンクションの計測

データファンクションとは、計測対象となるソフトウェアから参照、若しくは更新を行う論理的な意味でのデータのまとまりのことである。データファンクションは、計測対象のソフトウェアから更新される内部論理ファイル (ILF : Internal Logical File) と、計測対象のソフトウェアから参照されるのみの外部インタフェースファイル (EIF : External Interface File) の 2 種類のファンクションタイプに分類できる。

最初に、計測対象のソフトウェアからデータファンクションを抽出し、ファンクションタイプを決定する。次に、それぞれのデータファンクションの複雑度、データ項目数、レコード種類数の二つのパラメータによって、低、中、高の 3 段階に分類する。データ項目とはデータを定義する最小の単位であり、データ項目数は一つのデータファンクションの中に存在するデータ項目の個数である。また、一つのデータファンクションの中に、異なる意味合いをもつデータのまとまりが混在している場合、その個数をレコード種類数という。混在していない場合は、レコード種類数は 1 となる。

(Step 3) トランザクションファンクションの計測

ソフトウェアに対するデータの入出力を伴う処理をトランザクションファンクションという。トランザクションファンクションは、計測境界外からのデータ入力によってデータファンクションの更新を行う外部入力 (EI : External Input) 、計測境界外へ派生デー

表 1 未調整 FP 算出表
Table 1 Calculation table of unadjustment FP.

タイプ	低	中	高	合計
ILF	a1×7	a2×10	a3×15	ILFの合計
EIF	b1×5	b2×7	b3×10	EIFの合計
EI	c1×3	c2×4	c3×6	EIの合計
EO	d1×4	d2×5	d3×7	EOの合計
EQ	e1×3	e2×4	e3×6	EQの合計
				総計(未調整FP)

タ(計算や条件判断など何らかの加工を必要とするデータ項目)の出力を行う外部出力(EO: External Output), 計測境界外へデータをそのまま出力する外部照会(EQ: External Inquiry)の3種類のファンクションタイプに分類できる。

最初に, 計測対象のソフトウェアからトランザクションファンクションを抽出し, ファンクションタイプを決定する。次に, それぞれのトランザクションファンクションの複雑度を, 関連ファイル数, データ項目数の二つのパラメータによって, 低, 中, 高の3段階に分類する。関連ファイル数とは, 対象となるトランザクションファンクションで参照, 若しくは更新するデータファンクションの個数である。データ項目数は, 実際に参照, 若しくは更新するデータ項目の個数である。

(Step 4) 未調整ファンクションポイントの算出

Step 2, Step 3の結果をもとに, 表1を用いて未調整ファンクションポイント(未調整FP)を計算する。ここで a1, a2, a3 は計測対象ソフトウェアに含まれる複雑度が低, 中, 高のILFの数, b1, b2, b3 は計測対象ソフトウェアに含まれる複雑度が低, 中, 高のEIFの数, c1, c2, c3 は計測対象ソフトウェアに含まれる複雑度が低, 中, 高のEIの数, d1, d2, d3 は計測対象ソフトウェアに含まれる複雑度が低, 中, 高のEOの数, e1, e2, e3 は計測対象ソフトウェアに含まれる複雑度が低, 中, 高のEQの数である。なお, 今後, 未調整FPを単にFPと呼ぶ。

(2) IFPUG法の課題

どの時点の見積りも重要であるが, 特に開発の前段階での見積りが重要であると考えられる。この時点での見積りが誤って過小になった場合, 実際にかかった開発費, 開発工数, 開発期間は計画したものよりも大きくなる。その結果, 例えば, 購入者は新しいソフトウェアの稼動開始時期が遅れて事業の好機を逃すことになり, 供給者は購入者から支払われる金額を超える

開発費がかかり, いわゆる, 赤字になる。すなわち, 見積りを誤った場合の影響が大きい。また, 筆者らが, 開発費が当初の計画よりも超過したプロジェクトの原因を分析したところ, 見積り不良によるものが非常に大きな割合を占めていることが判明した。したがって, 開発の前段階で精度の高い見積りを行うことが重要となる。この時点では, RFPに記載されている機能要件をもとに見積りを行う必要がある。しかし, 記載されている機能に漏れがある場合や, 業務要件設計やソフトウェア方式設計が完了していないために詳細部分が固まっていない場合が多い。したがって, IFPUG法に習熟していない人が計測しようとした場合, 次のような課題があると考えられる。

- Step 3でのデータファンクションの複雑度を分類するときのパラメータである, レコード種類数, データ項目数を正確に把握することが難しい。
- Step 4でのトランザクションファンクションの複雑度を分類するときのパラメータである, 関連ファイル数, データ項目数を正確に把握することが難しい。
- Step 4でのEI, EO, EQの3種類のファンクションタイプは, データの入出力を伴う処理を, 最も小さな単位に細分化したものであり, 抽象的で, 具体的な処理や操作との対応付けが難しい。その結果, トランザクションファンクションの抽出が難しい。

このようなIFPUG法の課題を改善するものとして, NESMA法[11], 電中研法[13]等いくつか考案されている。NESMA法には, FP概算法とFP試算法の2種類の見積方法がある。

FP概算法は, データファンクションやトランザクションファンクションの抽出及びファンクションタイプの決定の仕方は, IFPUG法と同じであるが, 複雑度をデータ項目数, レコード種類数, 関連ファイル数で分類するだけでなく, それぞれあらかじめ設定されているデフォルト値を当てはめる。したがって, IFPUG法での複雑度の分類が難しいという課題は解決されている。しかし, 実際の業務ソフトウェアでは, 3種類のトランザクションファンクションは, 低, 中, 高, いずれの複雑度もとり得るが, デフォルト値が一意に当てはめられるために, 精度が粗くなると考えられる。また, IFPUG法と同様に, トランザクションファンクションの抽出が難しいと考えられる。

FP試算法は, データファンクションの計測はFP概

表 2 要素機能一覧
Table 2 List of element functions.

	要素機能名	機能概要	FP単価
更新系	新規登録	登録データ入力後、マスタファイルへ新規データ登録。例えばSQLのinsert文。	5
	既存データ変更	変更データ入力後、マスタファイルのデータ変更。例えばSQLのupdate文。	5
	既存データ削除	削除データの指示により、マスタファイルのデータ削除。	4
	マスタメンテナンス	マスタファイルの内容をメンテナンスする機能。	12
	その他更新	更新データ入力後、追加、変更、削除が混在する更新。	6
画面出力系	問合せ応答	画面上からの条件入力により、データを表示。	5
	一覧照会	画面上からの条件入力により、該当データの一覧を表示。	5
	明細照会	画面上で一覧から選択により、該当データの詳細内容を表示。	5
	計算結果表示	画面上からデータ入力後、計算結果を表示。	6
	更新のための照会	変更や削除のために、前もって現在値を照会。	4
	選択肢一覧	ListBox や PopUp らで商品コード、商品名の選択肢を表示。	3
	その他照会	上記にあてはまらない照会機能。	5
	その他系	帳票出力	印刷帳票の出力、PDF ファイルの作成、EUR 帳票出力のためのデータの引渡し。
	CSV 出力	電子ファイルの出力機能。	5
	その他データ出力	帳票や電子ファイル以外のデータ出力。	5
	他システムへの出力	他システム送信用のファイル作成、他システム送信用の電文の作成、API 呼び出しによる他システムへのデータ引渡し。	5

算法と同じであるが、トランザクションファンクションの抽出を行わず、データファンクションの ILF, EIF の個数から、計算式により FP を算出する。したがって、FP 概算法よりも更に精度が粗いと考えられる。

電中研法は、機能の要素として画面、帳票、ファイル、電文など分かりやすいものを用いているため、早期の見積りに適しているが、IFPUG 法とは互換性がない。

本論文では、開発の前段階でも高い精度で見積りを行えるように IFPUG 法を改善した「要素見積法」を提案する。要素見積法は FP 概算法と類似の手法である。

3. 要素見積法

要素見積法では、2. で挙げた IFPUG 法の課題の改善のために、データファンクションの複雑度の分類の簡易化と、トランザクションファンクションの抽出及び複雑度の分類の簡易化を図っている。

(1) データファンクションの計測

IFPUG 法と同様に、データファンクションを抽出し、ILF, EIF のいずれかにファンクションタイプを決定する。ただし、複雑度をレコード種類数、データ項目数の二つのパラメータによって分類するのでなく、FP のデフォルト値 (ILF : 8, EIF : 5) を FP 単価として用いる。

(2) トランザクションファンクションの計測

要素見積法では、トランザクションファンクション

のファンクションタイプを EI, EO, EQ の 3 種類に分類するのでなく、表 2 に示す、要素機能と呼ぶ 16 種類のファンクションタイプに分類する。

要素機能は、IFPUG 法ではトランザクションファンクション EI, EO, EQ の抽出が難しいので、これらを想起しやすくするための処理を用意しようという着想から生まれた。例えば、EI として抽出される処理には、「新規登録」、「既存データ変更」、「既存データ削除」があり、これらを要素機能として設定した。IFPUG 法の EI, EO, EQ はデータの入出力を伴う処理を最小の単位に細分化したものであるが、要素見積法での 16 種類の要素機能は、画面上の 1 個の GUI ボタンや 1 回のファンクションキーを押すことで実行される具体的な処理であり、ユーザ視点での入出力の最小単位と考えることができる。

要素見積法でのトランザクションファンクションの計測では、最初に要素機能を抽出する。これは、IFPUG 法におけるトランザクションファンクションの抽出と、ファンクションタイプの決定に該当する。要素機能の複雑度は、IFPUG 法のように、関連ファイル数やデータ項目数で分類するのでなく、表 2 の「FP 単価」の列に示す、各要素機能の 1 個当りの FP のデフォルト値を当てはめる。したがって、IFPUG 法と同様に、FP を算出することができる。

(3) 要素見積法での FP 単価の設定と評価

データファンクション ILF, EIF の FP 単価、要素機能の FP 単価は、次のような考え方で設定した。

表 3 要素機能ごとの FP 単価表
Table 3 List of unit FP of element functions.

		新規登録	既存データ変更	既存データ削除	マスタメンテナンス	その他更新	問合せ応答	一覧照会	明細照会	計算結果表示	更新のための照会	選択肢一覧	その他照会	帳票出力	CSV出力	その他データ出力	他システムへの出力
EI	出現頻度	1.0	1.0	1.0	3.0	1.0											
	FP	5	5	4	3	6											
EO	出現頻度						0.5	0.5	0.5	1.0			0.5	1.0	0.5	0.5	0.5
	FP						6	6	6	6			6	6	6	6	6
EQ	出現頻度				1.0		0.5	0.5	0.5		1.0	1.0	0.5		0.5	0.5	0.5
	FP				3		4	4	4		4	3	4		4	4	4
要素機能の FP 単価		5	5	4	12	6	5	5	5	6	4	3	5	6	5	5	5

筆者らが所属する組織では、1996～1998年にかけて実施されたプロジェクトの IFPUG 法による実績から、データファンクション、トランザクションファンクションの FP を、複雑度を評価せずに、ILF:8, EIF:5, EI:5, EO:6, EQ:4 と FP 単価を設定すると、実用上、有用であることが経験的に得られていた。そこで、要素見積法における FP 単価は、これらの値をもとに設定した。

(a) データファンクションの FP 単価

ILF, EIF の FP 単価は上記の値をそのままを採用し、ILF:8, EIF:5 とした。また、これらの FP 単価が妥当であるかを、金融分野の業務ソフトウェアから一つ、製造分野の業務ソフトウェアから二つ、公共分野の業務ソフトウェアから三つの計 6 業務ソフトウェアの IFPUG 法と要素見積法による実測データを用いて評価した。IFPUG 法による実測結果では、ILF, EIF の FP の平均値は、それぞれ 7.3, 5.0 であり、要素見積法の FP 単価との誤差は 10%未満であり、実用的に問題がないため、要素見積法で設定したデータファンクションの FP 単価は妥当であると判断した。

(b) 要素機能の FP 単価

各要素機能の FP 単価は、次のような考え方で設定した。

要素機能はデータの入出力を伴う処理であるので、その中には IFPUG 法でのトランザクションファンクション EI, EO, EQ が含まれる。そこで、各要素機能でのトランザクションファンクションの出現頻度と FP を表 3 のように想定して、要素機能の FP 単価を設定した。また、ここでの EI, EO, EQ の FP は、先

に説明した経験的に得られた EI, EO, EQ の FP 単価、すなわち EI:5, EO:6, EQ:4 をもとに設定した。

例えば「新規登録」はデータファンクションの更新があり、EI のみが必ず 1 回出現するので、FP は EI の FP 単価の 5 を、そのまま採用した。同様に「既存データ削除」は、EI のみが必ず 1 回出現するが、削除処理ではデータ項目数が少なく複雑度が低くなる傾向にあるので FP には 4 を採用した。「問合せ応答」は EO または EQ の可能性があるが、それぞれの発生確率は、ほぼ同等であると考え、EO, EQ の出現頻度を 0.5 ずつに設定した。また、そのときの FP は FP 単価をそのまま採用した。

このように、IFPUG 法による実測の経験から表 3 を作成し要素機能の FP 単価を設定したが、これらの FP 単価の妥当性について、データファンクションの FP 単価と同様に六つの業務ソフトウェアについて、IFPUG 法と要素見積法による実測データを比較することで評価した。その結果を表 4 に示す。

FP 単価の評価では、表 4 における FP 単価—FP 平均値が FP 平均値の -10～+25%の範囲に収まっているかを妥当であるかの基準とした[4]。したがって、「その他更新」、「その他照会」、「CSV 出力」を除いて、FP 単価は妥当なものであると判断した。この範囲に収まっていない要素機能については、出現頻度を加味して評価した。例えば「その他更新」は +57.9%と大きく外れているが、一方、要素機能の出現頻度の平均値が 6.7%に対して「その他更新」の出現頻度は 1.8%と少なく、実質的な影響は少ない。同様に「その

他照会」「CSV 出力」も実質的な影響は少ないと評価できる。これらの結果より、我々が対象としているドメインにおいては、設定した FP 単価の値は妥当なものであると判断した。

なお、これまでの説明から、要素見積法で測定した FP と IFPUG 法で測定した FP は、精度を別にすれば、基本的には同じものとなる事が理解できる。

4. 要素見積法のシステム化 (AP-Estimate)

見積りの容易化、見積結果の蓄積等を目的に、見積支援システム AP-Estimate を開発した。本章では AP-Estimate の構成、機能について説明する。

4.1 AP-Estimate の構成

図 2 に AP-Estimate の構成を示す。

AP-Estimate は図 2 で示すように、1 台のサーバと

表 4 六つの業務ソフトウェアの実測結果
Table 4 The result of the actual estimation of 6 business application software.

要素機能	個数	出現頻度(%)	FP 合計	FP 比率(%)	FP 平均値	FP 単価	(FP 単価-FP 平均値) / FP 平均値 (%)
新規登録	580	11.8	2526	12.0	4.4	5	13.6
既存データ変更	712	14.5	2980	14.2	4.2	5	19.0
既存データ削除	439	9.0	1671	8.0	3.8	4	5.3
その他更新	87	1.8	334	1.6	3.8	6	57.9
問合せ応答	537	11.0	2362	11.2	4.4	5	13.6
一覧照会	565	11.5	2564	12.4	4.6	5	8.7
明細照会	131	2.7	692	3.3	5.3	5	-5.7
計算結果表示	12	0.2	61	0.3	5.1	6	17.6
更新のための照会	94	1.9	455	2.2	4.8	4	-16.7
選択肢一覧	650	13.3	2045	9.7	3.1	3	-3.2
その他照会	17	0.3	59	0.3	3.5	5	42.9
帳票出力	586	12.0	3170	15.1	5.4	6	11.1
CSV 出力	133	2.7	433	2.1	3.3	5	51.5
その他データ出力	294	6.0	1285	6.1	4.4	5	13.6
他システムへの出力	61	1.2	329	1.6	5.4	5	-7.4
合計	4898	100.0	20996	100.0	65.5	74	221.8

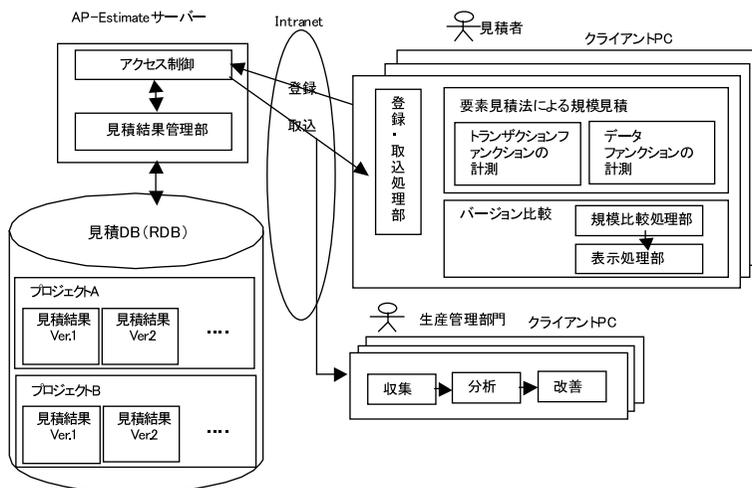


図 2 AP-Estimate の構成
Fig. 2 The structure of AP-Estimate.

サブプロジェクト名	FP	マスタ分FP	機能分FP	画面数	帳票数	見積ks
共通	317	90	227	31	2	75.98
プロジェクト全体	317	90	227	31	2	75.98

項目	値	項目	値
オン/パッチ区分	オンライン処理	個数合計	2
仕様の明確度合い	×不明点多い	新規登録	2
FP計	10	既存データ変更	0
STEP数	10	既存データ削除	0
層	1.35	マスタメンテナンス	0
P層画面	1.35	その他更新	0
P層帳票	0.00	問い合わせ応答	0

図 3 トランザクション関数の FP 算出のための入力例

Fig. 3 An example of input operation for calculation FP of transaction functions.

個数	低	中	高	不定	FP	個数
更新あり	0	0	0	10	90	10
(FP/個)	7	10	15	8		
参照のみ	0	0	0	2	10	2
(FP/個)	5	7	10	5		
合計					90	12

図 4 データ関数の FP 算出のための入力例

Fig. 4 An example of input operation for calculation FP of data functions.

複数のクライアント PC から構成され、イントラネットを介して通信を行う。サーバには見積 DB が接続されている。見積りを行うプロジェクト管理者や開発者は、クライアント PC から必要な情報を入力して見積結果 (FP) を得る。また、この結果は見積 DB に格納される。見積技法の開発を担当している生産管理部門は、見積 DB に格納されている多くの見積結果を分析し、例えば要素機能の FP 単価へのフィードバックからの見積精度の改善を実施する。

4.2 AP-Estimate の機能

(1) 要素見積法のサポート

利用者は、図 3 で示すように、クライアント PC から各要素機能の個数を入力することによりトランザクション関数の FP を算出できる。同様に図 4 で示すように、ILF, EIF の個数を入力することによりデータ関数の FP を算出できる。

(2) 見積結果の蓄積

見積結果を見積 DB に格納する。見積りはプロジェ

クトごとに実施するが、企画段階、開発の前段階、業務要件設計終了以降等、複数回、実施することが多いと考えられるので、プロジェクト名、格納した年月日及び時刻を見積結果に付加し、区別できるようにした。

見積り実施時期で区別された各見積結果をバージョンと呼ぶ。なお、業務要件設計終了以降は IFPUG 法によって見積りを行うことが多いと考え、AP-Estimate では IFPUG 法による見積結果も蓄積できるようにしている。

(3) バージョン比較

バージョン間を比較し、差異を分析することにより、より精度の高い見積りを実施できる。

(4) アクセス制御

見積結果にパスワードを付加して登録することにより、特定の人以上の参照を禁止することができる。

5. 適用事例

一般に RFP に記載されている機能要件ではソフト

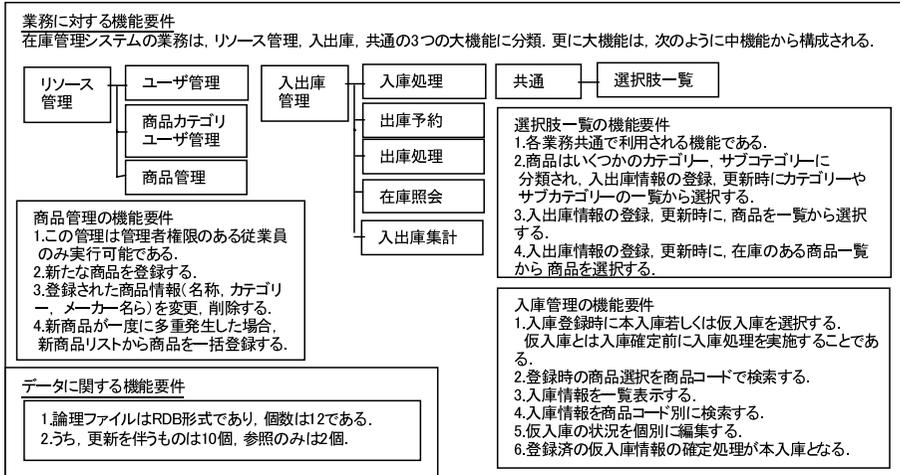


図 5 在庫管理ソフトウェアの機能要件
Fig. 5 Function requirements of stock control software.

ウェアの機能仕様は明確でなく、IFPUG 法による見積りは難しいと考えられている。5. では、RFP に記載されている機能要件から要素見積法で規模の見積りを行えることを、在庫管理ソフトウェアでの実例を用いて説明する。

5.1 在庫管理ソフトウェアの機能要件

データに関する機能要件、業務に関する機能要件を図 5 に示す。ただし、図 5 は、業務仕様に関しては、商品管理、入庫管理、選択肢一覧についてのみ、機能要件を示しており、他は省略してある。

5.2 FP の計測

(1) データファンクションの計測

図 5 に記載されているデータに関する機能要件から、ILF、EIF の個数は、それぞれ 10、2 であることが分かるので、データファンクションの FP は、 $10 \times 8 + 2 \times 5 = 90$ と算出できる。ここで 8、5 は、それぞれ ILF、EIF の FP 単価である。

(2) トランザクションファンクションの計測

共通機能である選択肢一覧と商品管理の FP を算出する。算出のやり方は、図 5 に記載されている機能要件に対して、見積り者が更に想定する仕様を追加し、要素機能を抽出して、個数を数えるものである。

(a) 選択肢一覧

- 機能要件「2. 商品はいくつかのカテゴリー、サブカテゴリーに分類され、入出庫情報の登録、更新時にカテゴリーやサブカテゴリーの一覧から選択する」には、カテゴリーから選択、サブカテ

グリーから選択の 2 種類の処理があり、それぞれが要素機能「選択肢一覧」に該当するので、「選択肢一覧」が 2 個存在する。

- 機能要件「3. 入出庫情報の登録、更新時に、商品を一覧から選択する」には、要素機能「選択肢一覧」が 1 個存在する。
 - 機能要件「4. 入出庫情報の登録、更新時に、在庫のある商品一覧から商品を選択する」には、要素機能「選択肢一覧」が 1 個存在する。
 - したがって、選択肢一覧には、要素機能「選択肢一覧」が 4 個存在する。表 3 より、選択肢一覧の FP 単価は 3 ゆえ、中機能選択肢一覧の FP は、 $4 \times 3 = 12$ となる。
- (b) 商品管理
- 機能要件の「2. 新たな商品を登録する」には、要素機能「新規登録」が 1 個存在する。
 - 機能要件の「3. 登録された商品情報(名称, カテゴリー, メーカー名)を変更, 削除する」には、商品を選択し、その商品の登録情報を照会し、変更若しくは削除を行う処理が想定される。商品の選択のうち、カテゴリー、サブカテゴリーからの選択は共通機能の選択肢一覧に該当する。したがって、要素機能「一覧照会」、「明細照会」、「既存データ変更」、「既存データ削除」が各 1 個ずつ存在する。
 - 機能要件「4. 新商品が一度に多重発生した場合、新商品リストから商品を一括登録する」は、要素

機能「新規登録」が1個存在する。

- したがって、要素機能「新規登録」が2個、「既存データ変更」が1個、「既存データ削除」が1個、「一覧照会」が1個、「明細照会」が1個であり、FPは $2 \times 5 + 1 \times 5 + 1 \times 4 + 1 \times 5 + 1 \times 5 = 29$ となる。

他の中機能についても、同様の方法でFPを算出できる。

6. 評価

(1) 要素見積法の精度

要素見積法の精度の評価では、要素機能のFP単価の精度評価と、要素機能の抽出精度の二つの面から行う必要がある。

(a) FP単価の精度

要素見積法では、各要素機能のFP単価を表2のFP単価の列に示すように設定している。したがって、このFP単価の設定値が要素見積法の精度に影響を与えたと考えられる。この精度分析も、3.で要素機能のFP単価の評価と同時に行った。図6は、六つの業務ソフトウェアについて、IFPUG法で計測したFPを100%として、要素見積法で算出したFPを評価したものである。図6に示すように、要素見積法は、IFPUG法に対し、-4~+18%の範囲に収まっていることを確認できた。

(b) 要素機能の抽出精度

要素見積法では、開発の前段階において、RFPに記載されている機能要件から要素機能を精度高く抽出できることを、出張旅費精算を行うソフトウェアを例にとり評価する。RFPでは、出張旅費精算ソフトウェアの機能要件について、表5の程度しか示されていないことは、よくあることである。IFPUG法やNESMA FP概算法では、この情報からトランザクションファンクションを抽出することになる。手法に習熟していない人は、表6(1)のように見積りの可能性が高く、これでは過小見積りとなってしまふ。一方、見積手法に習熟している人は、機能の説明とトランザクションファンクションの関係を把握して、表6(2)のような適切な見積りを行うが、ここまで習熟するには一定の経験を必要とする。IFPUG法やNESMA法で、このような習熟した考え方にたどりつけない要因として以下の2点があると考えられる。

- 「2段階の照会」、「修正の陰に削除あり」等、いわゆる機能説明の行間にあたるものを、EI, EO,

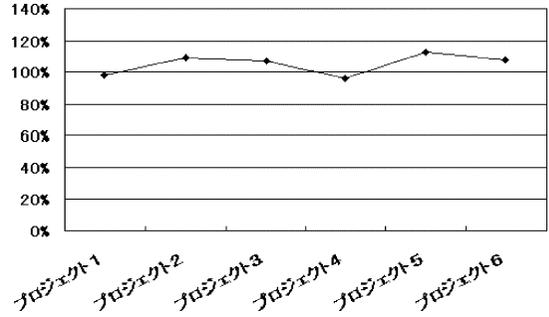


図6 要素見積法の精度分析

Fig. 6 Accuracy analysis of element sizing method.

表5 出張旅費精算ソフトウェアのRFP

Table 5 RFP of the travel expense reimbursement software.

機 能
1) 事前に登録しておいた出張案件から、精算したい案件を選択する。
2) 旅費に関する事項を入力システムに登録する。
3) 登録した旅費精算の内容を後で修正できる。
4) 領収書の提出などのために、旅費精算書という帳票を出力する。

EQのファンクションタイプの定義だけから想起することは容易でない。

- 開発の前段階での見積りでは、EOかEQかの判別は難しい。なぜならEQではなくEOとなる要件は「処理中に計算が含まれる」か「データファンクション更新を伴う照会である」のいずれかであるが、いずれも、その時点では明確でないことが多い。

要素見積法での見積り結果は表7のようになる。表7で示すように、要素機能は具体的な機能を表すので、その機能に関連する要素機能が想起しやすく、要素機能の抽出漏れを防止するナビゲーションとしての役割を果たしていることが分かる。このように、IFPUG法やNESMA法等に比べて、要素見積法は見積りの初心者であってもトランザクションファンクションの抽出を容易に行うことができる。

(2) その他の評価

AP-Estimateにより規模の見積りを実施した開発者に、AP-Estimateの効果についてアンケートをとり、次のような評価を得た。

- (a) 各要素機能の個数の分布を評価することで、特定の要素機能の個数が多すぎる、少なすぎるということが判断でき、見積り誤りを防止できる。

- (b) IFPUG法では、トランザクションファンクション、データファンクションの複雑度の評価に時間

表 6 IFPUG 法または NESMA 法による見積結果
Table 6 Estimation result by IFPUG method or NESMA method.

(1)未習熟者の見積		
機能		見積結果
1) 事前に登録しておいた出張案件から、精算したい案件を選択する。		EQ×1
2) 旅費に関する事項を入力しシステムに登録する。		EI×1
3) 登録した旅費精算の内容を後で修正できる。		EI×1
4) 領収書の提出などのために、旅費精算書という帳票を出力する。		E0×1
		FP 合計:17

(2)習熟者の見積		
習熟者の考え方		見積結果
1) 精算したい案件を選択するには、まず、条件入力候補一覧というステップがあり、続いて 1 件選択→精算案件内容表示となる。即ちここには 2 つの照会が存在し、さらに前者は計算を伴う可能性が高いので E0 と判断できる。		E0×1
2) 文面どおり、登録のトランザクションファンクションがあると判断する。		EI×1
3) 修正するには登録した内容を表示する必要がある。その表示は 1) のケースと同様に 2 段階の照会となる。また、修正には削除機能も含まれると解釈できる。		E0×1 E0×1 EI×2
4) 帳票の出力は通常計算を伴うので E0 と判断できる。		E0×1
		FP 合計:32

表 6(1), (2) の FP 合計の算出では EI, E0, EQ の複雑度を”中”と仮定している。

表 7 要素見積法による見積結果
Table 7 Estimation result by element function estimation method.

要素見積法での考え方	見積結果
1) 「案件の選択」とあるので案件を表示する“明細照会”が洗い出せるが“明細照会”があるなら“一覧照会”も存在するのではないかと、ということが想起される。	一覧照会 明細照会
2) 旅費精算の登録は“新規登録”であるが“新規登録”があると“既存データ変更”や“既存データ削除”を伴うことが多い。変更や削除は 3) で示されているのでそちらで計上する。	新規登録
3) 修正が“既存データ変更”で、変更があれば“既存データ削除”も存在すると想起される。また“更新のための照会”という要素機能種別により、変更・削除には登録内容の照会を伴うことが想起され、それが 1) のケースと同様に 2 段階の照会となるかどうかという連鎖思考に結びつく。	既存データ変更 既存データ削除 一覧照会 明細照会
4) 帳票の出力は EQ である可能性もある。早期見積時点では E0 か EQ かは判断がつかないことが多いが、要素見積法では E0 か EQ かは判断せず“帳票出力”として見積る。	帳票出力
	FP 合計:40

を費やすが、要素見積法では複雑度の評価が不要である。実際にソフトウェアの FP を、IFPUG 法と要素見積法の二つの技法で計測したところ、要素見積法の計測速度は IFPUG 法に対して 3~5 倍速かった。

7. むすび

本論文では、IFPUG 法による見積りが難しいと考えられている開発の前段階でも見積りが可能な要素見積法の提案と、それをシステム化した AP-Estimate の開発を行った。6. で示すように、要素見積法は (1) 精度が IFPUG 法の -4~+18% の範囲に収まっており (2) 仕様の明確でない開発の前段階でも要素機能

の抽出が容易であり (3) 計測速度が IFPUG 法の 3~5 倍速く、開発の前段階での見積技法として、十分なものであることを確認できた。また、見積支援システム AP-Estimate は、膨大な量の見積結果の蓄積ができる。しかも、多くのソフトウェアの見積結果、同一のソフトウェアのライフサイクルの各時点での見積結果、要素見積法や IFPUG 法等異なる見積技法による見積結果が蓄積される。したがって、これらの膨大なデータを分析することで、より精度が高く、適用の容易な見積技法への改善が期待できる。今後は、見積結果及び実績に関するデータを十分に蓄積し、分析を行い、より精度の高い見積技法に改善していきたい。

謝辞 本研究において、多大な協力を賜りました(株)日立システムアンドサービスの加藤允基氏に、心より御礼申し上げます。

文 献

- [1] H. Kerzner, Project Management, John Wiley&Sons, New York, 2001.
 - [2] R. Rada and J. Craparo, "Standardizing software projects," Commun. ACM, vol.43, no.12, pp.21-25, Dec. 2000.
 - [3] W.S. Humphrey, Managing the Software Process, Addison-Wesley, Winthrop, 1989.
 - [4] B.W. Boehm, C. Abts, A.W. Brown, S. Chulani, B.K. Clark, E. Horowitz, R. Madachy, D. Reifer, and B. Steece, Software Cost Estimation With COCOMO II, Prentice Hall, New Jersey, 2000.
 - [5] 高橋宗雄, クライアント/サーバシステム開発の工数見積り技法, ソフト・リサーチ・センター, 東京, 1998.
 - [6] 野中 誠, 角瀬章広, ブカーリイサム, 東 基衛, "画面仕様書に基づく対話型ソフトウェアの複雑度重みつき機能規模の測定技法," 情処学論, vol.43, no.12, pp.3993-4004, Dec. 2002.
 - [7] C. Jones, Applied Software Measurement, McGraw-Hill, New York, 1996.
 - [8] A.J. Albrecht, "Measuring application development productivity," Proc. Joint Share, Guide, and IBM Application Development Symposium, pp.83-92, 1979.
 - [9] 西山 茂, "ソフトウェア規模の見積り技術の最近の流れ—行数による評価から機能量による評価へ," 情処学論, vol.35, no.4, pp.289-298, April 1994.
 - [10] Function Point Counting Practices Manual, Release 4.0, International Function Point Users Group, 1994.
 - [11] <http://www.nesma.nl/>
 - [12] C.E. Walston and J.P. Winters, Applying Use Cases, 2nd ed., Addison Wesley, Winthrop, 2001.
 - [13] 高橋光裕, 菱谷 淳, 情報システムの規模見積り手法, 電力中央研究所研究報告 (R93016), 1994.
 - [14] 共通フレーム 98 -SLCP-JCF98- (1998 年版), SLCP-JCF98 委員会 (株)通産資料調査会, 東京, 1998.
- (平成 17 年 4 月 11 日受付, 8 月 9 日再受付)



原田 晃 (学生員)

1970 埼玉大・理工・数学卒。1975 名古屋大学大学院博士課程了(数学専攻)。同年(株)日立製作所入社。基本ソフトウェアの開発に従事した後、情報システム構築の生産技術に従事。2005 年 6 月に日本電子計算(株)に転属し、現在に至る。情報処理学会, プロジェクトマネジメント学会各会員。



幕田 行雄

1986 千葉大・工・機械卒。1988 同大大学院修士課程了。同年(株)日立製作所入社習志野工場に勤務。情報・通信グループで情報システム構築の生産技術に従事した後、2005 年 4 月に日立ソフトウェアエンジニアリング(株)に Outreach, 現在に至る。

PMI 会員。



石川 貞裕

1984 青山学院大・理工・経営工学卒。同年(株)日立製作所入社。現在、情報・通信グループ生産技術本部に所属。エンタープライズ系システムのアプリケーションアーキテクト, 開発技術の整備に従事。



大野 治 (正員)

1969 宇都工業高等専門学校電器工学卒, 同年より(株)日立製作所コンピューター事業部に勤務。現在、情報システム事業部事業部長。工博。プロジェクトマネジメント学会, 情報処理学会各会員。



楠本 真二 (正員)

昭 63 阪大・基礎工・情報卒。平 3 同大大学院博士課程中退。同年同大・基礎工・情報・助手。平 8 同学科講師。平 11 同学科助教授。平 14 阪大・情報・コンピュータサイエンス・助教授。博士(工学)。ソフトウェアの生産性や品質の定量的評価, プロジェクト管理に関する研究に従事。情報処理学会, IEEE, IFPUG, PM 各会員。



井上 克郎 (正員)

昭 54 阪大・基礎工・情報卒。昭 59 同大大学院博士課程了。同年同大・基礎工・情報・助手。昭 59~61 ハワイ大マノア校・情報工学科・助教授。平元阪大・基礎工・情報・講師。平 3 同学科・助教授。平 7 同学科・教授。平 14 阪大・情報・コンピュータサイエンス・教授。工博。ソフトウェア工学の研究に従事。情報処理学会, 日本ソフトウェア科学会, IEEE, ACM 各会員。