

論文

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、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],[2],[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],[5],[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]等いくつか考案されている。

[†] 日本電子計算株式会社, 東京都

[‡] 日立ソフトウェアエンジニアリング(株), 東京都

[§] (株)日立製作所, 東京都

^{**} 大阪大学大学院情報科学研究科, 大阪府

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種類の要素機能を予め定義しておき、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を用いた規模見積を実施しており、精度の高い結果を得ている。

本論文では、第2章でIFPUG法の概要と課題を、第3章で要素見積法を、第4章でAP-Estimateの実現方式を、第5章で適用事例を、第6章で評価を行い、第7章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2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期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7つの工程も標準化された開発プロセスと対応をつけることができる。

(2) 見積の時点

ソフトウェアライフサイクルの様々な時点で見積を行うことは、企画の作成や、開発を進めていくうえでの改善に役立つが、特に、図1に示す次の3つの時点での見積が重要であると考えられる。

(a) 企画段階での見積

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

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である。購入者が作成したRFPに記載されている要件を基に見積を行う。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。即ち、RFPには、実現して欲しいことは記載さ

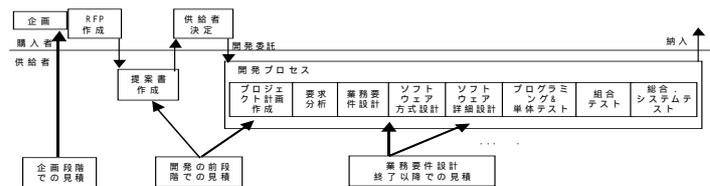


図1 ソフトウェアライフサイクルと見積の時点の関係
Fig.1 Relation of software life cycle and the timing for software sizing

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2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

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である。高い精度での見積が可能なが多い。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、説明の中でで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。

(Step1) 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。

(Step2) 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を、データ項目数、レコード種類数の2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1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、1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纏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。

(Step3) 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2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。

(Step4)未調整ファンクションポイントの算出

Step2, Step3の結果を基に、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 E0 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。

表1 未調整FP算出表

Calculation table of unadjustment FP

タイプ	低	中	高	合計
		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
E0	d1 × 4	d2 × 5	d3 × 7	E0の合計
EQ	e1 × 3	e2 × 4	e3 × 6	EQの合計
				総計(未調整FP)

(2) IFPUG法の課題

どの時点の見積も重要であるが、特に開発の前段階での見積が重要であると考えられる。この時点での見積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働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、所謂、赤字になる。即ち、見積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基に見積を行う必要がある。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。従って、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。

- ・ Step3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。
- ・ Step4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。
- ・ Step4でのEI, E0, 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づけが難しい。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のではなく、それぞれ予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えるが、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。

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, E0:6, 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，これらの値を基に設定した．

(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 1 つ、製造分野の業務ソフトウェアから 2 つ、公共分野の業務ソフトウェアから 3 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, E0, 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, E0, EQ の FP は，先に説明した経験的に得られた EI, E0, EQ の FP 単価，即ち EI:5, E0:6, EQ:4 を基に設定した．

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 E0 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，E0, 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．

		新	既	既	マ	そ	問	一	明	計	更	選	そ	帳	C	そ	他
					ス	の	合	覧	細	算	新	択	他	票	S	他	シ
									照	結		肢		出	V		ス
EI	出現頻度	1.0				1.0											
	FP	5	5	4		6											
E0	出現頻度						0.5	0.5	0.5	1.0			0.5	1.0	0.5	0.5	
EQ					3												

要素機能の

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 6 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%の範囲に納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．この範囲に納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．

表4 6つの業務ソフトウェアの実測結果

Table4 The result of the actual measurement 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

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ことが理解できる。

4. 要素見積法のシステム化(AP-Estimate)

見積の容易化、見積結果の蓄積を目的に、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台のサーバーと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。サーバーには見積DBが接続されている。見積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

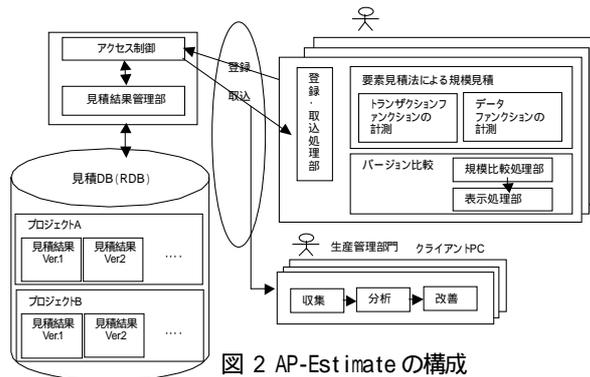


図2 AP-Estimateの構成
Fig.2 The structure of AP-Estimate

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ファンクションの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ファンクションの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。見積はプロジェクト毎に実施するが、企画段階、開発の



図3 トランザクションファンクションのFP算出のための入力例
Fig.3 An example of input operation for calculation FP of transaction functions

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、特定の人以外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ウェアの機能仕様は明確でなく、IFPUG 法による見積は難しいと考えられている。第 5 章では、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 であることがわかるので、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、サブカテゴリーから選択の 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。



図4 データファンクションのFP算出のための入力例
Fig.4 An example of input operation for calculation FP of data functions

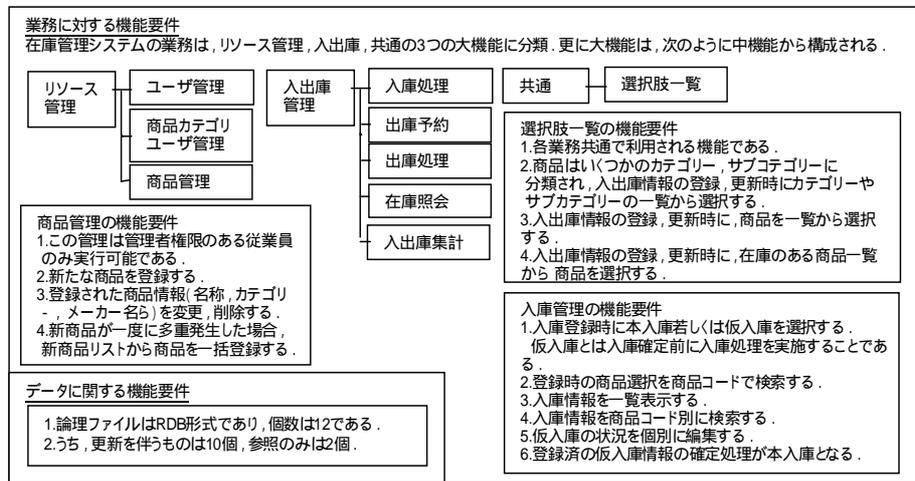


図5 在庫管理ソフトウェアの機能要件
Fig.5 Function requirements of stock control software

- ・機能要件「3.入在庫情報の登録,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.入在庫情報の登録,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,4×3=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,メーカー名ら)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2×5+1×5+1×4+1×5+1×5=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2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られ考えられる。この精度分析も,第3章で要素機能のFP単価の評価と同時に行った。図6は,6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%の範囲に納まっていることを確認できた。

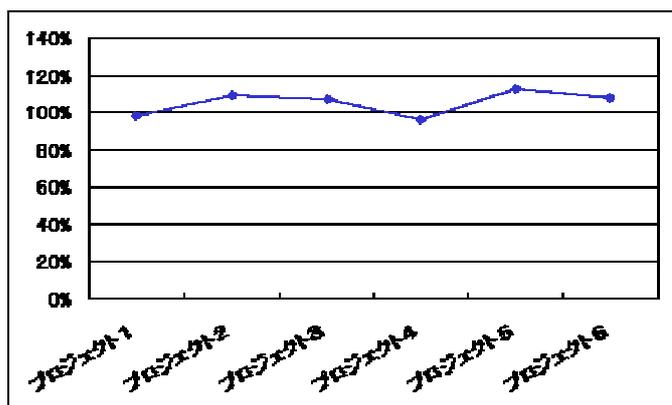


図6 要素見積法の精度分析
Fig.6 Accuracy analysis of element sizing method

(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。

表5 旅費精算ソフトウェアのRFP
Table5 RFP of 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し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。

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)のように見積る可能性が高く,これでは過小見積となってしまふ。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、E1、E0、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では、E0かEQかの判別は難しい。何故ならEQではなくE0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わかる。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2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倍～5倍速かった。

7. むすび

本論文では、IFPUG法による見積が難しいと考えられている開発の前段階でも見積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。第6章で示すように、要素見積法は、1)精度がIFPUG法の-4%～+18%の範囲に納まっており、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倍～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

表6 IFPUG法またはNESMA法による見積結果

Table6 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。	E1×1
3)登録した旅費精算の内容を後で修正できる。	E1×1
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。		E1×1
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 E1×2
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E1、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

Table7 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

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 法ら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。

参考文献

- [1] H. Kerzner, Project Management, John Wiley&Sons, Inc. New York, 2001.
- [2] R. Rada, J. Craparo, " Standardizing software projects," Communications of the ACM, vol.43, no.12, pp.21-25, Dec. 2000.
- [3] Watts 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 , 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, プカーレイサム, 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," 情報処理学会論文誌, vol.43, no12, pp3993-4004, Dec. 2002.
- [7] C. Jones, Applied Software Measurement, The McGraw-Hill Companies, New York, 1996.
- [8] A. J. Albrecht, "Measuring Application Development Productivity, " Proc. of the 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- ," 情報処理学会論文誌, 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 Edition, Addison Wesley, Winthrop, 2001.
- [13] 高橋光裕他, 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.

著者紹介

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．2005.6 に日本電子計算株式会社に転属し，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・工学・機械卒．1988 同大大学院修士過程了．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，2005.4に日立ソフトウェアエンジニアリング（株）に出向し，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チャ、開発技術の整備に従事．

大野 治（正会員）

1969 宇部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・通信グループ事業主管兼CIO．プロジェクトマネジメント学会，情報処理学会各会員．

井上 克郎（正会員）

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～昭61八王子大マノア校・情報工学科・助教授．平1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．工学博士．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．

楠本 真二（正会員）

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．

【英文アブストラクト】

ABSTRACT. In the development of the business application software, it is considered to be very important to estimate the development cost and schedule at the early stage, which the function specification is yet unconfirmed. To figure out the development cost and schedule, we firstly estimate the size of business application software, and then make an additional consideration of the system characteristics such as reliability.

Recently, the function point method, which estimates the size of business application software by quantifying its functionality, has been in widespread use. However, there are some difficulties in estimating the size at the early stage of software development by using the function point method. In this thesis, we report on the proposal of the early estimation technique based on the function point method and its systematization. After applying it to a lot of projects, we gained a satisfactory result in terms of accuracy and simplicity.

【英文キーワード】

[KEYWORD]

software size estimation, function point, FP, project management, business application software

【図及び表ネーム】

図1 ソフトウェアライフサイクルと見積時点の関係

Fig.1 Relation of software life cycle and the timing for software sizing

図2 AP-Estimateの構成

Fig.2 The structure of AP-Estimate

図3 トランザクションファンクションのFP算出のための入力例

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

図4 データファンクションのFP算出のための入力例

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

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

図6 要素見積法の精度分析

Fig.6 Accuracy analysis of element sizing method

表1 未調整FP算出表

Table1 Calculation table of unadjustment FP

表2 要素機能一覧

Table2 List of element functions

表3 要素機能毎のFP単価表

Table3 List of unit FP of element functions

表4 6つの業務ソフトウェアの実測結果

Table4 The result of the actual estimation of 6 business application software

表5 出張旅費精算ソフトウェアのRFP

Table5 RFP of the travel expense reimbursement software

表6 IFPUG法またはNESMA法による見積結果

Table6 Estimation result by IFPUG method or NESMA method

表7 要素見積法による見積結果

Table7 Estimation result by element function estimation method

英文著者紹介

Akira Harada

Japan Information Processing Service Co., Ltd., IT Planning and Support Division
2-4-24 Toyo, Koto-ku, Tokyo, 135-8388, Japan

Ikuo Makuta

Hitachi Software Engineering Co., Ltd., Process Improvement Technology Center
4-12-7, Higashishinagawa, Shinagawa-ku, Tokyo, 140-0002, Japan

Sadahiro Ishikawa

Hitachi, Ltd., Information and Telecommunication Systems, Engineering Support Division
Hitachi Systemplaza Shinkawasaki 890 Kashimada, Saiwai-ku,
Kawasaki, Kanagawa, 212-8567, Japan

Osamu Ohno

Hitachi, Ltd., Information Technology Division.
6, Kanda-Surugadai 4-Chome, Chiyoda-ku, Tokyo, 101-8010, Japan

Shinji Kusumoto

Department of Computer Science Graduate School of Information Science and Technology
Osaka University
1-3 Machikaneyama-cho, Toyonaka-City, Osaka 560-8531, Japan

Katsuro Inoue

Department of Computer Science Graduate School of Information Science and Technology
Osaka University
1-3 Machikaneyama-cho, Toyonaka-City, Osaka 560-8531, Japan