

# ファンクションポイント法実践 — 見積の可視化を目指して —

生産品質強化本部  
統括PMOセンター  
品質・プロセス改善グループ

小川 明子



信託・証券システム  
ビジネスユニット  
年金業務ソリューション  
第一センター  
センター長

渡辺 浩史



## 1. はじめに

ソフトウェアの見積の場面で最も使われている手法としてLOC (Lines Of Code=プログラムの行数) があるが、近年プラットフォームや開発言語の多様化により、LOCでは対応しきれないケースも増加の一途を辿っている。未経験の業務や技術、言語による新規システム開発の際、プロジェクトマネージャの「KKD (勘・経験・度胸)」のみに頼った見積を行った結果、開発規模が増大し赤字が発生する事例も散見される。

また、見積時などに、発注者から客観的な尺度による説明責任を求められるケースが増えている。JUASの動向調査(参考文献8)でも、「開発ベンダーへの不満」という項目に対し、ユーザー企業802社中161社が「不満」と回答した中で、不満事項の2位が「見積金額の妥当性が不明」ということであった(1位は企画提案力不足)。

このような状況の中、見積に根拠を持たせ、実現可能な計画を立てることがプロジェクト成功のために不可欠となっており、そのために有用な指標の1つとして、ファンクションポイント法(FP法)に取り組む企業が増えてきている。FP法での見積を契約の条件とする発注者も徐々に増加しており、また、近年脚光を浴びているオフショア開発でも、発注先企業がFP法での見積を提出してくるケースが見られる。

本稿では、当社におけるFP法への取り組みについて報告する。

## 2. FP法とは

まずは、FP法の概要を解説する。

FP法は、その名のとおり機能(ファンクション)を点数(ポイント)化することによりソフトウェアの規模を計測する手法であり、下記の点に特徴がある。

- ユーザー視点で見た機能の量を計測
- 開発環境・言語などに左右されない
- 発注者や他社と共通の尺度である

元々は1979年にIBMのAllan J. Albrechtが考案し、その後改訂を加えて1984年に現在のIFPUG法に近い形で提唱されたものである。その後の改良により、フィーチャーポイント法、Mark II、COSMIC-FFP、3DFP、フルファンクションポイント法などさまざまな手法が提案されている。本稿ではFP法と言えはIFPUG法を指す。

IFPUGとは、1986年に米国で発足した非営利のユーザー団体であるInternational Function Point Users Groupのことであり、ソフトウェアの計測技術の開発と普及に努めている。日本でも1994年にJFPUG (Japan Function Point Users Group=日本ファンクションポイントユーザ会)が設立され、1996年にはIFPUGの支部となった。JFPUG会員企業は、2004年11月現在で約300社、当社も法人会員となっている。開発ベンダーだけでなく、ユーザー企業の参加も多い。

1998年にはISO/IECの機能規模測定(Function Size Measurement=FSM)手法として標準化され(ISO14143-1)、1999年にはJIS化された(JIS X 0135-1)。

FP法での計測手順は以下のとおりである(図1参照)。

- (1)計測種別(新規開発FP、機能拡張FP、アプリケーションFP)の決定

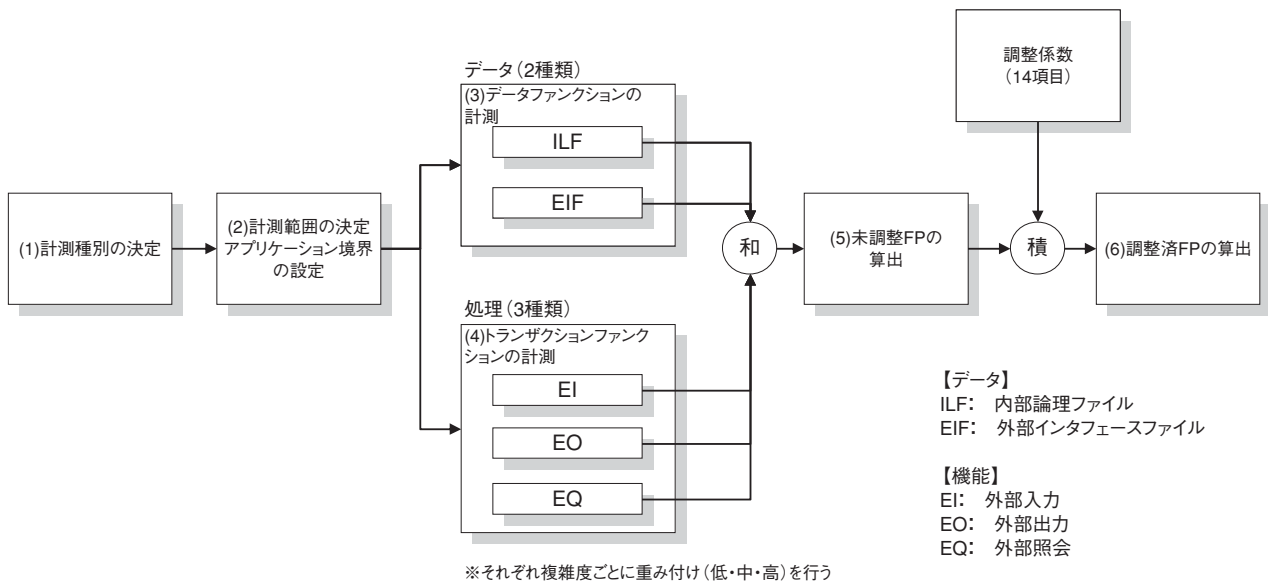


図1 FP計測の概略図

- (2)計測範囲とアプリケーション境界の決定
  - (3)データファンクション(ファイルとして捉えられる機能)の計測と複雑度の決定
  - (4)トランザクション・ファンクション(データ処理によって実行される機能)の計測と複雑度の決定
  - (5)未調整FPの算出
  - (6)調整係数の決定と調整済FPの算出
- ※上記項目番号は図1に記載の番号と連動

また、FP法の導入によって得られる効果は以下のとおりである。

1) 生産性の可視化

FP法を用いれば、インフラ、開発言語、開発ツールなどによらない、純粋なアプリケーション規模の測定が可能となる。これは、発注者からすれば、開発ベンダーからの提案時の開発工数(もしくは受注金額)や開発期間から、アプリケーション規模を基準に開発ベンダーの生産性を評価できることになる。逆に、開発ベンダーから見れば、生産性向上の工夫を可視化できることになり、他の開発ベンダーとの差別化を図ることが可能となる。さらに、開発ベンダーにとっては、開発における生産性リスクを判断する指標ともなりうる。

2) 品質の可視化

開発中であれば、アプリケーション規模を基準として、仕様変更発生数・障害発生数などを評価することにより、開発プロセスごとの品質を定量的に把握することができ、問題の早期発見・対処が可能となる。

本番稼働後には、同様の評価をすることによりアプリケーションの品質の把握が可能となり、保守体制構築、改修・再構築要否の判断材料として利用できると思われる。さらに、品質とは関係ないが、アプリケーション規模は適

正な最低限度の保守体制を判断する際の一要素となりうるのではないかと考える。

### 3. CACにおけるFP法の実践

従来、発注者の要請などによりプロジェクトで個別に計測を行ったり、一部の部署で自主的にFP法による見積りに取組んだりという動きはあったものの、当社が組織としてFP法への取組みを開始したのは2003年である。当初は、KM推進室(当時。現品質・プロセス改善グループ、以下品質改G)にて、動向調査や計測ガイドの作成、社内教育の準備等を行っていた。

そして2004年秋より、金融システムビジネスユニット(当時。後に金融システムビジネスユニットと信託・証券システムビジネスユニットに分割)にて、FP法実践委員会を立ち上げて計測を試みることになり、品質改Gが支援を行った。

FP法実践委員会は、ユニット内の各センターより1~2名ずつ、および営業1名を選出して構成し、開発済みの4システムについてアプリケーションFPを、新規開発中の1システムについて新規開発FP(システム基本設計終了時点)を計測した。品質改Gでは2004年夏より社内研修(これはJFPUG主催の研修テキストをベースとしたもので、テキストの使用はJFPUG会員企業が社内研修をする場合に限り認められている)をトライアル的に実施しており、2005年6月より正式開講の予定であるが、委員もまずこの研修を受講し、基礎知識を得た上で実際の計測作業に入った。

計測の前提は下記のとおりとした。

- 1) IFPUG法での計測を基本とし、可能であればFP試算法

およびFP概算法（詳細は5.5にて後述）でも計測する。

2) 他社と共同開発のシステムについては、当社の担当範囲について計測する。

3) 可能な限り、実績工数を使用して生産性を算出する。実績工数は、社員が日常的に記録している工程ごとの実績工数データを利用する。

計測作業の過程においては、アプリケーション境界確定、データファンクション計測、トランザクション計測といったタイミングでレビューを行った。とはいえ、正しいか否かを見極め判断するのは困難な面がある。JFPUGからも詳細なガイドラインが出版されていることからわかるように、FP法自体が機能をどのように解釈するかによって計測結果にぶれが生じるため、必ずしも正解が明確でない場合がある、という弱点を持っているからだ。実際には、一緒になってFP法のマニュアルであるCPM（参考文献1）を調べながら1つひとつ解釈していったと言うほうが正しいかもしれない。ただし、基本的な考え方である「ユーザー視点」から乖離することのないように、という点に可能な限りの注意を払い、指摘を行った。レビューでの確認事項はTIPS集としてまとめ、JFPUGの計測法検討委員会(CPC)に検証を依頼中である。

## 4. 考察

計測により明らかになった課題について考察する。

### 4.1 FP法に対する誤解

FP法で計測することにより、すぐに開発工数を正しく求めることができると現場の開発者は思いがちであるが、見積のためには実績の蓄積が必要であると知ると落胆するようだ。

そもそもFP法はアプリケーション全体の規模を計測するマクロな指標であり、個々の機能を正確に見積もるための手法ではない。機能の各要素を「低・中・高」の3段階の複雑度で括って評価することからも明らかである。FP法の問題として指摘される所以でもある。

開発工数はFP値×生産性\*1で求められるが、標準生産性こそが開発ベンダーのスキルであり、ノウハウである。よって、FP法を導入したからといって、即座に適正な開発工数が見積もれるわけではない。開発ベンダーの過去の開発実績を統計的に評価した結果得られるものである。しかし、そのためには、まずある程度の実績値を蓄積する必要があり、「卵が先か、鶏が先か」というジレンマが生じる。FP法に限らずLOCなどでもこの過程を経ているはずであるが、現場の開発者はすぐに使える指標を求めており、

ここにFP法の展開を阻む大きな壁があるように思われる。FP法を導入することの効果、現場の開発者が納得し、受入れることが重要である。

また、FP法では、システムの要素を最終的に調整係数で調整する。技術的な要素は日進月歩であり、経験値的な要素が非常に高くなる。また、成果物、顧客体制といった要素の影響も加味する必要があると思われる。調整係数こそが開発ベンダーのノウハウであり、過去実績の類似性と実績工数の差異分析から導き出されるものだろう。ISO/IECのFSMでは調整係数を必須としていないことと、また今回の活動においては、サンプリング対象としたシステムも少なく、類似性が低いため、調整係数の使用については今後の検討課題としている。

### 4.2 計測負荷の高さ

FP法は項目数を数える作業が多いため、負荷が大きく感じられるようだ。しかし、よく考えてみると、FP法の計測手法は、今までの工数見積の延長であることに気づく。つまり、作業負荷は今までとさほど変わらないのである。この点を強調し、理解を得る必要があると考える。

一般的には、画面、帳票、機能の一覧から、要素ごとに経験と勘で開発工数見積をしていると想像する。FP法もこの流れは変わらない。画面、帳票、機能の一覧がベースとなり、追加として下記情報が必要とされる。

- ① データの一覧と参照・更新の分類
- ② データが保有する項目数
- ③ 各要素が更新・参照するデータ数
- ④ 各要素の入出力項目数

ここで「項目数」「データ数」とあるが、FP法では、3段階の複雑度で評価される。つまり、あまり厳密に計測する意味がないとも取れる。項目数、データ数を数える作業は負荷が大きく感じられるため、作業負荷と効果とのバランスで、項目数、データ数の計測精度を検討する必要があるだろう。

FP値への影響が大きい要素は以下のとおりである。これらについては詳細に検討する必要がある。

- 1) 画面、帳票、機能、データの数
- 2) 論理的なアプリケーションの境界

項目数、データ数は、各要素の粒度が適切か否かの判断基準と考えられる。数が大きい場合は、要素の単位が粗い可能性があり、各要素の単位を論理的に見直す必要があると考える。この意味では、上記②④は概算値、①③はある程度の精度をもった計測が、負荷と効果のバランスからみて適切ではないだろうか。

また、アプリケーション境界については、アプリケー

\*1) 従来の生産性はLOC/人月で求めるが、FP法の実績値は人月/FPで求める。

ションの粒度という観点から、検証する必要がある。このためには、要素とデータのマトリクスを作成するのが効果的だ。これは③を整理していれば、おのずと可能である。計測プロセスの妥当性を評価するための補助資料ともなるう。

## 5. 今後の展開および考慮点

FP法を展開するために今後進めていく施策、および考慮点は次のとおりである。

### 5.1 計測の自動化

組織の標準係数を確立するためには、計測者の負担軽減とデータ蓄積の仕組みを作ることが必要となる。

計測者の負荷軽減には、計測の自動化を図ることになる。市販のツールもあるようだが、当面は表計算ソフトでの自作ツールでも十分だろう。現在は、今回の委員会活動を通して計測シートの改良ポイントを把握し、改良にとりかかっている。図2はその一部である。

※数値は例です。

1.FP値	
1 FP試算値	500 FP
2 FP概算値	600 FP
3 IFPLUG法(未調整)	620 FP
4 IFPLUG法(調整済)	640 FP

2.実績工数				
白い欄に、フェーズ別の実績時間を入力してください。フェーズ毎の工数、および配分率が表示されます。				
フェーズ	時間数	時間数	人員	工数実績配分 の比率(%)
1 システム化計画	A	1356.0	9.0	8.17%
2 システム基本設計	B	3250.0	21.7	19.59%
3 システム詳細設計	C	2705.0	18.0	16.31%
4 プログラム構造設計	D	-	-	-
5 モジュール設計	E	-	-	-
6 プログラミング	F	4623.0	30.8	27.87%
7 プログラム結合テスト	H	985.0	6.6	5.94%
8 システム統合テスト	I	1224.0	8.2	7.38%
9 システム総合テスト	J	1983.0	13.2	11.95%
10 運用テスト	K	-	-	-
11 導入	L	-	-	-
12 移行	M	463.0	3.1	2.79%
13 その他	X	-	-	-
合計		16589.0	110.6	100.00%

3.生産性(人時単位)				
1.および2で入力した結果より、FP計測法ごとの生産性(時間あたり)が自動計算されます。				
フェーズ	時間数	実績生産性(時間/FP)	IFPLUG法実績調整FP	IFPLUG法調整済FP
1 システム化計画	A	2.71	2.26	2.19
2 システム基本設計	B	6.50	5.42	5.24
3 システム詳細設計	C	5.41	4.51	4.36
4 プログラム構造設計	D	-	-	-
5 モジュール設計	E	-	-	-
6 プログラミング	F	9.25	7.71	7.46
7 プログラム結合テスト	H	1.97	1.84	1.59
8 システム統合テスト	I	2.45	2.04	1.97
9 システム総合テスト	J	3.97	3.31	3.20
10 運用テスト	K	-	-	-
11 導入	L	-	-	-
12 移行	M	0.93	0.77	0.75
13 その他	X	-	-	-
合計		33.18	27.65	26.76

4.生産性(人月単位)					
1.および2で入力した結果より、FP計測法ごとの生産性(月あたり)が自動計算されます。なお、1人月=150時間です。					
フェーズ	時間数	試算法FP	概算法FP	IFPLUG法未調整FP	IFPLUG法調整済FP
1 システム化計画	A	0.02	0.02	0.01	0.01
2 システム基本設計	B	0.04	0.04	0.03	0.03
3 システム詳細設計	C	0.04	0.03	0.03	0.03
4 プログラム構造設計	D	-	-	-	-
5 モジュール設計	E	-	-	-	-
6 プログラミング	F	0.06	0.05	0.05	0.05
7 プログラム結合テスト	H	0.01	0.01	0.01	0.01
8 システム統合テスト	I	0.02	0.01	0.01	0.01
9 システム総合テスト	J	0.03	0.02	0.02	0.02
10 運用テスト	K	-	-	-	-
11 導入	L	-	-	-	-
12 移行	M	0.01	0.01	0.00	0.00
13 その他	X	-	-	-	-
合計		0.22	0.18	0.18	0.17

図2 計測シート改良版イメージ

また、計測において大事なものは、「見積の過程」を記録しておくことである。「可視化」ということを考えれば、結果としてのFP値だけでなく、なぜそう見積もったのかという根拠を説明できるに足るだけの資料を提示できなく

ては意味がない。前述のマトリクスはまさにその過程を示すことになるため、計測シートの一部として組み込む予定である。

### 5.2 データ蓄積の仕組み作り

データ蓄積の仕組みについては、まだまだこれからという段階である。ただし、当社ではCMMレベル3に準拠したCSSP(CACソフトウェアプロセス標準、P.55からの記事参照)があり、そのプロセスの一環として昨年より新規開発プロジェクト終了時に「プロジェクト評価報告書」の提出を義務付けている。その中にFP実績値を記入する欄があり、CSSPの展開とともにFP法を展開していく、というのが現在の方針である。いずれは当たり前のようにFPの計測が行われ、マトリクスの一環として収集できるようになることを期待している。全社の品質目標として、今後はFPを利用していかうとする動きもある。

また、標準係数の算出のためには、日々の工程別実績工数の記録が正しく行われなければならない。今回残念だったのは、工程ごとの開発工数の記録があまり正確でなかったため、工程ごとの生産性算出を諦めたケースがあったことである。組織としての工数記録のルールは定められているのだが、面倒なのか、なかなか正確に記録されていないようだ。組織の生産性を求めるためには、まずここからは正していかななくてはならない。

### 5.3 研修時の考慮点

研修時の演習と実際のシステム計測との間にはどうしても乖離がある。これは研修というものの限界かもしれないが、研修における演習は「勉強のための演習」になりがちで現実感に欠ける点があるのは、時間の制約なども考えるとある程度は致し方ない。ただし、TIPS集を網羅したような例題を増やしたりすることは可能と考える。また、研修の受講者が実際のシステム計測に取りかかる際にスムーズに入っていけるよう、測定ガイドなどをさらに充実させる必要がある。とはいえ、計測を終えた委員によれば、FP法は「少し勉強すれば計測はできる」と感じられ、かつ「大きくは外れない」という感触をも持ったようだ。研修から実測へうまくつなげていければ、効果のある計測が可能となるだろう。

研修では、FP値への影響が大きい重要な概念を理解させる必要がある。以下の概念、考え方が重要と考える。

#### 1) アプリケーション境界の論理的な設定

導入当初は見積担当者感覚によるところが大きいと思われるが、アプリケーション境界の設定のしかたによっては、FP値への影響が大きく、論理的な設定方法が必要である。要素データ、要素処理を確定した段階でマトリクス表を作成し、境界を見せることが重要と考える。

## 2) ユーザー視点での要素データの定義

システム開発が後工程になるにつれ、データの単位が処理の都合や効率性から細分化され、ユーザー視点とはかけ離れたものとなる。今回、計測の対象としたシステムは、すでに開発が完了したものであったため、内部仕様が見えていだけに計測者にとっては判断に悩む場面が多かったようだ。これは、つい開発者の視点で解釈してしまうが故だろう。

たとえば、FP法では、必ずしも論理的なファイル=物理的なファイルと解釈するとは限らないが、RDBでは全テーブル、階層型DBでは全セグメントを数えるなど、物理的な実装にこだわってしまいがちだった。レビューにおいても、「迷った時はユーザー視点」という言葉を何度も繰り返し、原点に立ち戻る場面があった。

データ設計においては、システム的に考慮されるパターンを計測時の留意点として整理し、演習の中で体験させることが重要と考える。

## 3) ユーザー視点での要素処理の定義

「要素処理」は一般には以下のように定義されているが、非常に曖昧な定義と言わざるをえない。

- ユーザー目的にそった業務活動の最小単位である
- 自己完結し、計測アプリケーションを業務に矛盾のない状態に保つ

業務をどの単位とするかは非常に悩むところである。これらの概念を理解させるには、要素定義における要素処理の考え方の重要ポイントを整理し、演習に取り込むことが必要だろう。

## 5.4 計測者によるばらつき排除の工夫

FP法の弱点である「計測者によって結果にばらつきがある」という点をどのように解消していくかは、当社にとっても課題である。FP法の先進企業である株式会社CSKのCSFPA法のように、測定者の解釈による誤差を減少させる仕組みを作ること、解釈を統一化するためにカスタマイズすることは、信頼性の向上のためにも不可欠である。

JUASの調査(参考文献9)でも、FP計測を行っている企業のうち、自社用にカスタマイズして活用しているケースが半数を占めた。ただし、本来は世界共通のルールを適用することで世界共通の尺度となるため、「自社標準」が出来てしまうのは好ましいことではない。「自社標準」は、ばらつきを排除するためのルール、負荷軽減のための最低限度の遵守ルールとしての各社の工夫と思われるが、FP法の基本から乖離してしまうことは避けなければならない。

## 5.5 上流工程におけるFP計測の工夫

実際に見積を行う際には、まだ仕様の確定していない上流工程で計測することになるため、ある程度の仮定を置いた上で計測する「FP試算法」「FP概算法(推定法とも言う)」という方法がある。これはオランダのソフトウェアメトリクス研究団体NESMA(Netherlands Software Metrics Users Association)が提唱する方法であり、下記のルールで計測できる。

- FP試算法： $ILF \times 35 + EIF \times 15$
- FP概算法：「データファンクション=低、トランザクションファンクション=中」と仮置きして算出

ただし、FP試算法は誤差 $\pm 50\%$ と、見積としてはあまり有効ではない。FP概算法は誤差 $\pm 20\%$ と、それなりに有用な結果を出すことが可能となる。

今回の計測はFP試算法、FP概算法、IFPUG法の3種類で行ったが、その結果が表1である。FP概算法とIFPUG法での計測結果は概ね $\pm 20\%$ に収まっており、それなりに良い数字が出ていると言えるのではないかと(システムDについてはFP概算法よりもむしろFP試算法の結果がIFPUG法の結果に近いが、これは使用するテーブルが多くシステムが複雑であることにより、データファンクションが多くなったためと推測される)。

表1 計測結果一覧

(単位:FP 'ー'の部分は未計測)

	計測種別	システム形態	FP試算法	FP概算法	IFPUG法(未調整)
システムA	新規開発FP	C/S	2345	806	819
システムB	アプリケーションFP	ホスト	—	—	218
システムC	アプリケーションFP	Web	1025	757	733
システムD	アプリケーションFP	ホスト	360	321	407
システムE	アプリケーションFP	C/S	—	—	1468

上流工程では項目数など判断できないものであるし、精密さにこだわって作業負荷をかけるよりも、ある程度の割り切りを持ってFP概算法を使用するというのも1つの方法だろう。実績データ収集という意味でも、低負荷で多数のデータを集められるのは、収集初期には有効かもしれない(JFPUGは、FP概算法の使用については推奨も否定もしないという立場を取っている)。FP法はマクロな指標であるため、多少の誤差はあっても母数が多いほど一定の生産性の傾向が現れてくるものと期待できるためである。ただし、負荷の軽減にこだわってIFPUG法の基本がおろそかになることのないよう考慮が必要である。

## 6. 終わりに

初田(参考文献7)によれば、「プロジェクトの成否は、見積/契約で5~8割、システム仕様確定までに8~9割、開発で残り1~2割が決まる」とのことである。それほど見積は重要だと指摘しているわけだが、この点については

筆者もまったく同感である。

FP法が万能というわけではない。しかし、そもそも見積に「これなら必ず当たる」という魔法の方法はない。だからこそ精度を高める必要があり、これまでもそのためのさまざまな取り組みがされているのである。

いずれの方法であれ、一種類の方法で見積もっただけでは、精度が落ちる危険がある。本来は、可能な限り複数人で複数の方法を使って見積を行い、比較検討することが望ましい。その過程において、少なくともFP法は可視的かつ客観的な指標になりうるという点で、十分に利用価値があるといえる。

品質改Gには、当社の標準生産性についての問い合わせが現場から頻繁に寄せられている。そのことから、標準生産性の確立は急務であると考え。FP値による当社の標準生産性を確立するためにも、FP法での計測と、実績工数の正しい記録を社内に働きかけていきたいと考えている。

また、FP法に興味を持たれた方はぜひご一報頂きたい。FP法は社内に閉じこもってしまうのではなく、広く共通の尺度として使用されることで初めて意味のあるものとなる。共通の尺度を目指し、多くの方々と共に取り組んでいきたいと考えている。

## 〈参考文献〉

1. International Function Point Users Group (IFPUG) 著、日本ファンクションポイントユーザ会 (JFPUG) 訳：『ファンクションポイント計測マニュアル (CPM : Counting Practice Manual) リリース4.1.1』(2001年)
2. デービッド・ガーマス、デービッド・ヘロン著、小泉浩、中村永、向井清訳、児玉公信監訳：『ファンクションポイントの計測と分析』、株式会社ピアソン・エデュケーション (2002年)
3. 児玉公信：『実践 ファンクションポイント法』日本能率協会マネジメントセンター (1999年)
4. 株式会社アレア：『失敗のないファンクションポイント法』日経BP社 (2002年)
5. 本間周二：特集2 CSKに見るFP法の実践 プロジェクト定量化で品質と見積り精度向上、『日経ITプロフェッショナル』日経BP社 (2004年7月号)
6. 特集1 プロジェクト成否のカギを握る 本当に使える見積り技術、『日経ITプロフェッショナル』日経BP社 (2004年9月号)
7. 初田賢司：「ITプロジェクトにおける見積りプロセスのマネジメント」、ENAA/JPMF PMシンポジウム2004 F-1、財団法人エンジニアリング振興協会 (2004年)
8. (社) 日本情報システム・ユーザー協会 (JUAS)：「企業IT動向調査2005」、平成16年度第6回JUASアカデミー JUAS2005年2大調査報告会 (2005年)
9. (社) 日本情報システム・ユーザー協会 (JUAS)：「ユーザー企業向け2005年ソフトウェアメトリックス調査分析結果報告」、平成16年度第6回JUASアカデミー JUAS2005年2大調査報告会 (2005年)