

アプリケーション・マネジメント・サービス(AMS)における 開發生産性評価プロセスの確立

金子 英一 梶山 昌之

Establishment of Development Productivity Evaluation Process for Application Management Service (“AMS”)

Eiichi Kaneko Masayuki Kajiyama

筆者の担当しているアプリケーション・マネジメント・サービス(以下「AMS」と言う)アウトソーシングプロジェクトにおいては、開發生産性の向上率によるお客様の効果目標をコミットしており、開發生産性測定的手段としてファンクション・ポイント(以下「FP」と言う)法を導入している。本論文では、FPによる開發生産性の測定開始から、利用段階であるお客様との評価プロセスの確立までを含む、AMSアウトソーシングビジネスの価値の可視化手順および、経済効果の検証と評価手法について提案する。また、筆者の属するプロジェクトチームが、どのようにFPの定着とAMSアウトソーシングビジネスへの適用を図ったか実施例と評価を報告する。

The Application Management Services (“AMS”) outsourcing project in which the author is involved commits to clients the achievement of their individual goals measured by the rate of improvement in development productivity. The method introduced to measure such development productivity is the function point (“FP”) analysis. This paper proposes: (a) The procedures of visualizing the values of the AMS outsourcing business that starts from evaluation of development productivity through FP analysis, to the establishment of evaluation process with clients, and (b) The verification and evaluation techniques of the economic influence. Furthermore, the paper reports on the actual cases and the evaluation regarding the methods used by the project team, in which the author belongs, to establish the FP measurement and its application to AMS outsourcing business.

Key Words & Phrases : ファンクション・ポイント, ソフトウェア見積もり, AMS, アウトソーシング, 開發生産性, ソフトウェア定量化
function point, software estimation, AMS, outsourcing, development productivity, software quantification

1. はじめに

2004年のITサービス市場において、ユーザー企業によるコスト削減の強い要望がアウトソーシング案件にも波及し、案件の期間短縮化と規模縮小化の傾向が強まっている[1]。ユーザー企業がアウトソーシングを実施する場合、コスト削減を目的とする場合が多い[2]が、ユーザー企業のビジネスを取り巻く環境の変化や新しい戦略により、アウトソーシングへの期待効果や目的も変化し、同時に当初の目的が確実に達成されていることを証明するよう、サービス提供側は

求められている。また、ベンチマークなど外部のサービスを利用して、自社のアウトソーシングが他社と比較して十分な効果が上がっているか確認しようとする流れも始まっている。

このような場合、サービスの提供価値をユーザー企業とサービス提供側が共有することが必要であり、そのツールとしてサービスレベルアグリーメント(以下「SLA」と言う)が有効である。情報システムに係る政府調達へのSLA導入ガイドライン[3]においてSLAは、「ITサービスの提供者と委託者との間で、ITサービスの契約を締結する際に、提供するサービスの範囲・内容および前提となる諸事項を踏まえた上で、サービスの品質に対する要求水準を規定するとともに、規

提出日：2005年08月31日 再提出日：2005年12月9日

定した内容が適正に実現されるための運営ルールを両者の合意として明文化したものと定義している。

IBMのアプリケーション・マネジメント・サービス(以下「AMS」と言う)アウトソーシングプロジェクト(アプリケーション開発から保守まで,お客様のすべてのアプリケーション資産を総合的に運用し,お客様のビジネスプロセス効率化を支援するサービス)においては,契約金額が大きく期間が5年以上に渡る大規模プロジェクトは,ほとんどがSLAを契約している。しかし,その項目は,品質,納期もしくは見積もり差異に関するものが多く,生産性に関して契約しているプロジェクトは少ないのが現状である。

筆者の担当するAMSアウトソーシングプロジェクトは,契約期間中に開発生産性を向上しコストを削減することを提案している。具体的にはサービス開始初年度を基点として,サービス契約期間10年間の生産性向上率を効果目標としてコミットしている。効果目標は一律ではなく,お客様のビジネスの成長に最も結びつくe-Business開発領域は,その他のレガシー開発領域(既存の汎用OS系または分散OS系アプリケーションの開発)より高い効果目標を設定している。

ソフトウェア開発の生産性を考える場合,測定可能な量はソフトウェア規模と工数である。かつてはソフトウェア規模としてプログラムの行数が使用されていた。しかし,もし生産性が単位時間当たりにかかれたプログラムの行数で計量されるならば,高級言語を用いるとプログラムの行数で計量される生産性は高くなるどころか,かえって低くなる,というパラドックスに遭遇する[4]。

当プロジェクトでは以下の3つの目的を実現する手段として,ファンクション・ポイント(以下「FP」と言う)法を導入している。

- (1) 開発生産性を評価可能とする。
- (2) 一定の経済効果を保証する。
- (3) 開発生産性向上のメリットをお客様とIBM双方にもたらす。

FPは1979年にIBMのAlbrechtが発表したソフトウェアの規模を表す尺度であり,基盤やプログラミング言語などの違いを超えて,入出力とデータの種類の数,および複雑さといった機能の量でソフトウェアの規模を表す手法である。現在は,ソフトウェア規模定量化手法としてISO規格(ISO/IEC 14143)にもなっている。

FPを測定するためには,測定ルールを理解している測定者が,開発案件毎に機能を識別しFPを測定する。一方で案件毎の工数や案件の特性を示す情

報を収集し,開発生産性の算出と分析を行う。その上で全体の開発生産性の分析を行い,必要な対策を検討・実施していく必要がある。

当プロジェクトの経験からFPによる開発生産性評価プロセスは,以下の3段階を踏んで進めることが重要であると考える。

- (1) FPおよび開発生産性の利用目的を明確にし,測定を行う体制を構築し測定を開始する
 - (2) FP測定開始後に顕在化した課題の改善と,開発生産性の分析結果に基づき利用形態の検討を行う
 - (3) 開発生産性を測定した結果を利用する方法として,年度毎の開発生産性の目標設定と,実績の評価による経済効果の検証を行う
- (3)の段階で,AMSアウトソーシングビジネスにおける価値の可視化ならびにユーザー企業との共有が実現される。本論文では,2章でファンクション・ポイント法と開発生産性への適用状況を報告し,3章にてAMSの開発生産性評価へのFP法適用方法を提案する。さらに4章にて実施例と評価を報告する。

2. ファンクション・ポイント(FP)法とその開発生産性評価への適用状況

2.1 ファンクション・ポイント法

FPの測定法には代表的なIFPUG(International Function Point Users Group)法⁷⁾の他,開発初期段階でのFP測定に適したNESMA(Netherlands Software Metrics Association)法⁵⁾,リアルタイムシステムやOSなど基盤ソフトウェアに適したCOSMIC-FFP(The Common Software Metrics International Consortium-Full Function Point)法⁶⁾など幾つかの方法がある。

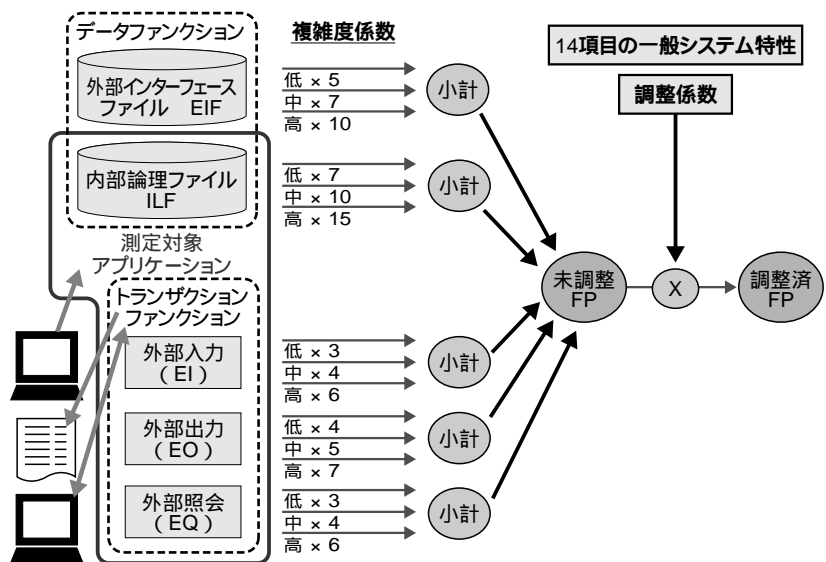


図1. IFPUG法測定概要 [7]

IFPUG法では図1のようにFPを測定する。

測定の流れは、1)測定種別の決定、2)測定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(内部論理ファイル、外部インターフェースファイル)の測定、4)トランザクションファンクション(外部入力、外部出力、外部照会)の測定、5)末調整ファンクション・ポイントの計算、6)調整計数の計算、7)調整済みファンクション・ポイントの計算である[8]。

2.2 FPの開発生産性への適用

FPは当初、規模算定、見積もり、ソフトウェアプロジェクトの測定のツールとして始まったが、利用者数の増大と共に利用の幅も拡大し、現在ではソフトウェア品質の分析、プログラミング言語の評価、本論文で提案するアウトソーシングの評価などに広がっている[9]。またガートナー社を始めとするベンチマーキングサービスにおいても、FPを利用した開発生産性の評価を行っている。このように、FPによる開発生産性の評価が一般化しつつある。

本論文では、開発生産性は「1人月当たりのFP」と定義し、案件毎FPの総和を案件毎工数の総和で除した値(加重平均)としている。目標と実績の関係は次のように説明できる。すなわち、基点となる年度の開発生産性の実績が10FP/人月であった場合、翌年度の生産性向上目標が10%であれば、 $10 \times (1 + 0.1) = 11\text{FP/人月}$ が開発生産性の目標となる。また翌年度の開発生産性の実績が12FP/人月であれば、 $(12 \div 10) - 1.0 = 20\%$ の生産性向上が達成されたことになる。

また、本論文ではFP法として、IBM社内システム開発やAMSアウトソーシングプロジェクトでの実績があり、ISO規格にも取り入れられているIFPUG法を基本にし、その簡便法を採用している。

3. AMSの開発生産性評価へのFP法の適用

ソフトウェアの測定をどのように導入し活用していくかについては、国際規格ISO/IEC 15939 "Software measurement process"のベースとなった著書 "Practical Software Measurement"[10]で体系的に論述されている。この著書では、測定を実装するためのフレームワークとして測定プロセスモデル(図2)を定義している。

このモデルではPDCAサイクルの「計画 - 実行 - 評価 - 改善」の手順に基づき、次の4つを主要なアクティビティとしている。

- ①測定に対するコミットメントの確立と維持
- ②測定計画の作成
- ③測定の実施

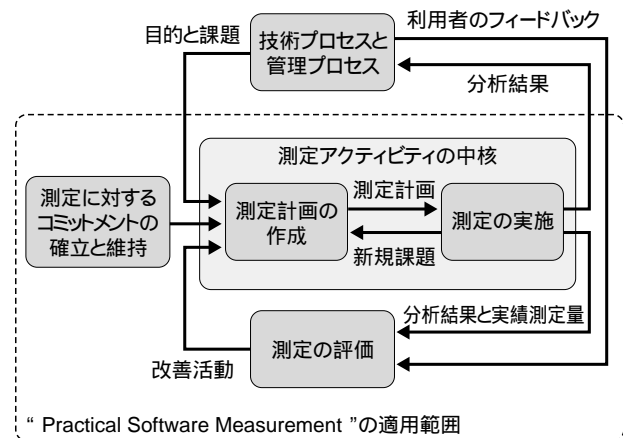


図2. 測定プロセスモデル[10]

④測定の評価

本論文の提案プロセスは、図2の測定プロセスモデルを基本に、測定尺度としてFPを導入している。以下の節で、上記4つのアクティビティを実施する上で必要な考慮点を列挙する。

3.1 測定に対するコミットメントの確立と維持

- (1) FPならびに開発生産性評価導入の目的を確認する。具体的には、開発生産性評価による経済効果の確認までとするのか、一定の経済効果の保証といった更なる目的をもつのかを確認する。
- (2) ユーザー企業とサービス提供側双方のメリットが何かを確認する。
- (3) 提案などで予め開発生産性向上目標があればサービス契約の全期間について設定、そうでない場合は、年度毎に実績を評価し翌年度の目標を設定する方法を検討する。
- (4) 開発領域毎に目標を設定する。全体平均の目標にした場合は、開発領域毎サービス量が年度毎に変化する影響を受けることになり、それを回避するためである。
- (5) 設定した目標は、SLA評価項目への追加を行う。インセンティブ・ペナルティの取り扱いは、SLA評価項目として一年以上の実績から検討する。
- (6) 測定を円滑に実施するために、マネジメントやキーパーソンの同意を得る。目的の提示と理解を促進するための計画を検討する。

3.2 測定計画の作成

- (1) 要件定義の進め方に改善の余地がある場合、ユーザー企業とサービス提供側の共同目標の設定を検討する。改善の施策はユーザー企業とサービス提供側の双方が実施することが望ましい。
- (2) FP測定方法として簡便法が利用可能かどうかを検討する。採用基準は、最低30件のIFPUG法

と簡便法のFP測定データの相関を確認し0.9以上であれば採用とする。統計的には、標本数は最低30以上であることが望ましい。

- (3) 測定は案件担当者からFP測定担当を兼任する要員を選出する方法で開始し、他の案件担当者へ対象を広げる方法を検討する。兼任のFP測定担当者が、開発グループの中でFP測定のエキスパートとなることで、FP測定のプレを抑える役割を担う。
- (4) 測定データが十分得られるよう計画する。対象開発領域の案件、工数の80%以上を目安とする。

3.3 測定の実施

- (1) 対象となる工数は外部設計から統合テストとする¹。ただし、AMS開始時に、工数は局面別に登録するプロセスにする。これで対象となる工数の局面が変更される場合でもデータの遡^{さかのぼ}りが可能となる。
- (2) アプリケーションや基盤の特性の違いを把握し、個別の開発領域を見極める。
- (3) 生産性の基準年度を定める。2年間の経年変化を比較し、変化の要因を把握し測定データの信頼性を確認する。
- (4) 標準偏差による生産性評価基準を定め、基準外案件の分析を進める。生産性向上阻害要因を見出し、施策へとつなげる。
- (5) 計画した十分な量の測定データが得られていることを確認する。

3.4 測定の評価

- (1) 測定負荷、測定時期など測定にかかわるデータを収集分析し、顕在化した課題に対して対策を検討する。月次に測定時期遵守状況の集計と各開発グループへの徹底を行い、確実にデータ量を確保する。
- (2) 月次に報告を行う。未達成目標に対する改善策の検討と進捗報告を行う。
- (3) 年度末に報告を行う。達成・未達成いずれの開発領域についても分析と評価を行い、翌年度の開発生産性向上施策につなげる。
- (4) 経済効果の評価を行う。効果目標による効果額と生産性実績による効果額を算出して、比較・評価する。

¹ サブシステム化がなされるような大規模開発において、下位レベル(コンポーネント間)のインターフェース検証を統合テストaとして開発実施局面で実施し、サブシステム間のインターフェース検証は、統合テストbとして開発実施後の独立した局面で実施する。

4. 提案プロセスの実施例と評価

4.1 測定体制の構築

当プロジェクトでは、開発生産性評価導入の目的は1章で述べた通りである。尺度としてのFPの導入に当たり、次の手順で測定体制の構築を行った。

- (1) FP測定者の選定。
- (2) IFPUG法をベースにプロジェクト固有の特性を加味した測定ガイドと事例集の作成。
- (3) 測定ガイドと事例集を使用し、測定者およびライン管理者向けの測定法講習会を実施。
- (4) 測定対象となるアプリケーションの棚卸し

FP測定者の選定に当たっては、開発グループに、開発とFP測定を兼任する担当者を若干名設置する方法をとった。この方法は、測定の負荷が集中するリスクがある反面、業務知識やアプリケーションについてすでに一定の理解があり、また測定者の違いによる測定のプレが少ないという2点でメリットがある。

また、FP測定を円滑に進めるためには、マネジメントやキーパーソンの協力が必要であり、それにはFPと開発生産性の利用目的の提示と理解が重要である。開発依頼者向けおよび開発者向けの説明会などでFP測定の意義や目的などを繰り返し説明し理解を求め、測定開始の了解を得た。

4.2 測定の開始

FP測定の時期は各案件の完了後一ヶ月以内と定め、測定を開始した。測定開始後は、以下の手順を繰り返し、FP測定精度の向上に努めた。

- (1) FP推進チームによる測定結果の検証と測定者へのフィードバック。
- (2) 測定ガイドの改訂、事例集の充実。
- (3) 測定者の異動などを機とした測定法講習会実施。測定結果の検証で判明した測定の間違いや新たな測定事例は、測定ガイドの改訂や事例集の充実により横展開を図っている。

4.3 顕在化した課題とその改善

測定開始から一年程度経過し、測定した案件数が100件を越えた辺りから、測定を継続する上で解決すべき次の3つの課題が明らかになった。

- (1) FP測定の負荷(工数)軽減が必要。
- (2) 精度を安定させるため、より簡素化した測定ルールが必要。
- (3) 案件の早い局面で測定可能な方法が必要。

これらの課題を解決するために、NESMA法も含めて、IFPUG法のルールを簡素化した幾つかの測定方法を検討した。検討の結果、IFPUG法のFP測定結果と最も近似した値が得られることから、IFPUG法の複

雑度係数をすべて中に固定する方法(以下「簡便法」と言う)を採用することにした。この方法の利点は、データ項目数やアクセスするファイル数などによる機能の複雑度係数の決定が不要となることである。適用の結果、開発工数に占めるFP測定工数は40%削減され、ほとんどの案件のFP測定が内部設計中に実施されるという、2点の効果が確認された。

4.4 利用形態の検討

4.4.1 開発生産性の定義の検討

効果目標の設定の考え方として、エンドユーザーの参画度合いや要求のあいまいさの影響が大きい要件定義とシステムテスト・移行の局面は、支援活動として向上率を0%とみなし、外部設計から統合テストbの局面は、開発者の努力によって開発生産性の向上が見込めるものとして向上率を設定している。その考え方にに基づき測定開始時は、外部設計から統合テストbまでの工数により生産性を算出した。ただし、外部設計から統合テストbの生産性が向上しても、逆に要件定義やシステムテストの生産性が低ければ全体では効果がないことになる。当プロジェクトでは、全体での評価を行うために、要件定義から移行まで案件すべての工数により開発生産性の算出と評価を行う形に変更した。また、評価範囲が要件定義から移行にまで広がったことから、効果目標を再設定した。

4.4.2 開発生産性の評価領域の検討

効果目標がe-Business開発領域とレガシー開発領域に分かれて設定されているため、開発生産性は、2つの領域に分けて算出と分析を行った。レガシー開発領域の分析の過程で、基盤の特性により明らかに生産性の違う領域が識別されたため、これを分離した。その結果領域は、e-Business開発領域、レガシー1開発領域(汎用OS系と分散OS系)、レガシー2開発領域(業界特有のOS)と3領域に分かれることとなり、効果目標も3領域別に設定し直した。

4.4.3 基準年度の検討

測定開始の初年度は、当初予定より3ヶ月遅れで当プロジェクトが開始されたこと、また予期せぬビジネス環境の激変が発生したことにより、対象案件数でみて全体の数%しか実績収集出来なかった。2年目では、年度始めから実績収集を行ったが、対象の案件数で70%、工数で62%の収集に留まった。一部の開発者の測定に対する理解が不十分であったことから、次の開発よりFP測定の優先順位が低く、FP測定の遅れが発生していたこと、そのフォローが後手にまわってしまったことが原因である。当プロジェクトでは、収集した割合がやや低いことと、単年度のデータでは、

それが通常の実績能力を表すのか判断が出来ないとの認識となり、さらに3年目の開発生産性実績収集を継続した。3年目では、簡便法の導入ならびに、月次に測定遅れのフォローを行ったことが効果を発揮し、対象の案件数でも工数でも85%以上が収集された。

2年目と3年目の開発生産性データを領域毎に比較検討すると、e-Business開発領域は、CMMI(Capability Maturity Model Integration)によるプロセス改善[11][12]や早期開発手法であるRAC(Rapid Application Center)といった各種のAMSソリューションを先行して適用した結果、開発生産性は大きく向上したが、レガシー開発領域は、ソリューションの適用が始まったばかりのためほとんど変化がなかった。データの収集量と生産性の変化状況を考慮して、3年目の開発生産性実績が生産能力を表すことを確認し、開発生産性の基点とした。

4.4.4 開発生産性分析プロセス

開発生産性は指数的な変化に従うことが知られている[4]が、当プロジェクトの2~3年目の開発生産性実績を統計手法により分析したところ、対数正規分布に従うことが確認された。この事実から、正規分布の標準偏差を用いて、案件データを以下の基準で分類した(図3参照)。

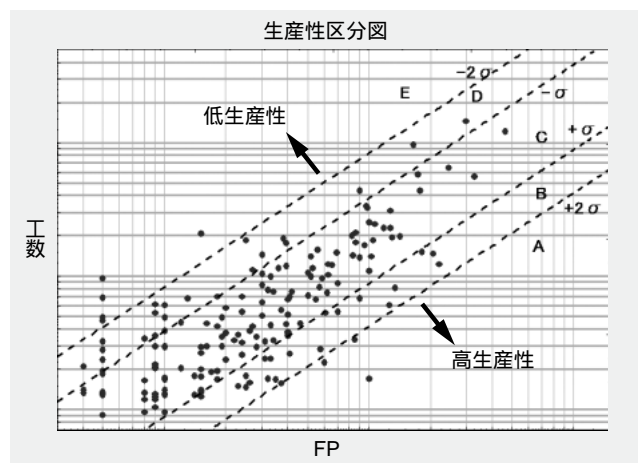


図3. 生産性区分図(両対数軸)

- ・ Aランク: +2 点生産性超の超高生産性
- ・ Bランク: +1 点生産性超の高生産性
- ・ Cランク: +1 点生産性以下-1 点生産性超の通常生産性
- ・ Dランク: -1 点生産性以下の低下生産性
- ・ Eランク: -2 点生産性以下の超低下生産性

また、開発生産性の高い案件や低い案件を調査していくと、幾つかの特性パターンが明らかになった。例えば、機能は少ないが非常に高度複雑な要件をも

つ処理の開発の場合は概して低い開發生産性となることや、すでに開発済みの処理プログラムやロジックを流用することで要求機能を提供できた場合は概して高い開發生産性となるといったもので、これらのパターンを整理し、報告を簡素化するためにチェックリスト(図4)を作成した。

案件特性: 当案件の案件特性にチェック(✓)してください	
1() 簡易改修	: 特定ロジックを複数機能へ反映した。多くの機能の一部を同様に変更した
2() 機能流用	: 既存機能を流用することにより、新規機能開発がされた
3() 定例作業	: 定期的に発生する作業で手順化されている
4() 基盤作業	: N/W系、汎用共通機能、H/W、S/W導入などの基盤系部分が工数に含まれる
5() ロジック偏重	: ロジック修正が多い割に機能が少ない案件である
6() 維持管理作業	: 不具合の修正、データ変更などの維持管理作業が工数に含まれる
7() 新規ツール(言語)	: 新規ツール・新規言語を適用した (ツール・言語名 <input type="text"/>)
8() 新規手法・技法	: 新規の手法や技法を適用した (手法・技法の名称 <input type="text"/>)
9() フォローアップ工数	: サービスイン後、納品前に一定期間のフォローアップ工数が当開発案件に含まれている

図4. 案件特性チェックリスト

4.4.5 FP課金方式の検討

1章で述べた3つの目的のうち、次の2つを実現する手段として、工数による課金方式から、機能すなわちFPによる課金方式に切り替えることを検討した。

- (1) 一定の経済効果を保証する
- (2) 開發生産性向上のメリットをお客様とIBMの双方にもたらす

この方式は、提供する機能であるFPの量に1FP当たりの単価をかけて請求金額(以下「FP金額」と言う)を設定する方式である。この方式を導入するため検討を続けた結果、工数課金方式とFP課金方式の違いを当プロジェクトの関係者が十分に理解する期間が必要との認識に至った。そこで工数課金方式と金額が大きく変わらず、しかもFPを身近に感じることが出来る折衷方式(以下「FP精算プロセス」と言う)を考案した。これは工数と工数単価による金額(以下「工数金額」と言う)で契約後、外部設計完了時にFPを測定し、FP金額により契約変更を行う方法である。ただし工数金額とFP金額の間で大きな差異が発生しないように、工数金額の95%から105%までの間で調整を行う仕掛けを組み込んでいる。

4.5 FP精算プロセスの導入

プロセスの導入に当たっては、当プロジェクトの関係者がFP精算プロセスを十分に理解することが必要である。そのため、次の準備を行った。

- (1) FP測定ガイド刷新と事例集の大幅拡充。
- (2) 開発担当者(PM, PLクラス)に対する測定法講習会の実施と開発担当者自身が測定する体制

への移行。

- (3) 開発作業の依頼者全員へのFP導入の目的、メリットと、FP測定法(依頼者に提出されるFP測定結果を確認できるようになるため)の説明会実施。導入から一年以上を経て、当プロジェクト関係者のFPと開發生産性への理解は深まっている。また、外部設計完了以降、納品前までにFP測定を行うことから、数ヶ月先の開發生産性が見通しがたつようになった。その状況を踏まえ、折衷方式から調整を行わない完全なFP課金方式への切り替えを、継続検討中である。

4.6 評価基準による分析プロセスの導入

開発領域毎の開發生産性向上を目指す上で、開發生産性の高い案件、開發生産性の低い案件を精査し、その要因を当プロジェクトの関係者へ報告できるようにすると同時に、事例を横展開するために、開發生産性の評価基準を設定することにした。評価基準は前年度実績を元に計算した標準偏差を用いて、案件に5段階のABCDEランク付けを行い、Cランクを除くすべての案件については、開発担当者がその要因となる案件の特性を考察し、FP測定結果と合わせてFP推進チームへの報告を義務付けることとした。

4.7 目標設定と実績評価プロセスの導入

4.7.1 目標設定

3年目の開發生産性実績を基点としているため、3年目以降10年目までの開發生産性向上率を使って4年目以降の開發生産性目標を開発領域毎に設定した(表1)。

表1. 効果目標と開發生産性目標例

効果目標				
開発領域	基準年度	2年目	3年目	4年目以降
e-Business	0	6.0%	18.0%	~
レガシー1	0	4.0%	13.0%	~
レガシー2	0	3.0%	10.0%	~

開發生産性目標				
開発領域	基準年度	2年目	3年目	4年目以降
e-Business	14	14.8	16.5	~
レガシー1	7	7.3	7.9	~
レガシー2	19	19.6	20.9	~

要件定義やシステムテストの工数を含めた開發生産性にしたことから、開発者側のソリューションや施策のみならず、依頼者側のプロセス標準化といった施策も前提とし、当プロジェクトの共同目標であることを確認した。これは、ベンダーであるIBMだけの目標より、結果に対して双方で責任をつという一歩踏み込んだ形となっている。

また、当プロジェクトのSLAは、当初は納期遵守率と品質として初期期間故障度および潜在障害故障度を契約していたが、目標設定に合わせて4年目からは開発生産性を契約に加えた。

4.7.2 実績評価プロセス

(1) 月次報告

当プロジェクトで実施しているプロジェクト関係者への月次の業務報告会にて、前月および累計の開発生産性実績と、目標より開発生産性が低い場合の分析結果とアクションプランを合わせて報告している。開発者側、依頼者側どちらに原因がある場合でも、確認や改善などを行い、次回報告会で結果の報告を行っている。

(2) 年次報告

開発領域毎の開発生産性目標と実績の間で、達成・未達成の評価を行うようにした。達成・未達成いずれの場合も図5のように統計手法による分析と達成・未達成要因の分析結果を報告し、翌年度の向上につなげるようにしている。

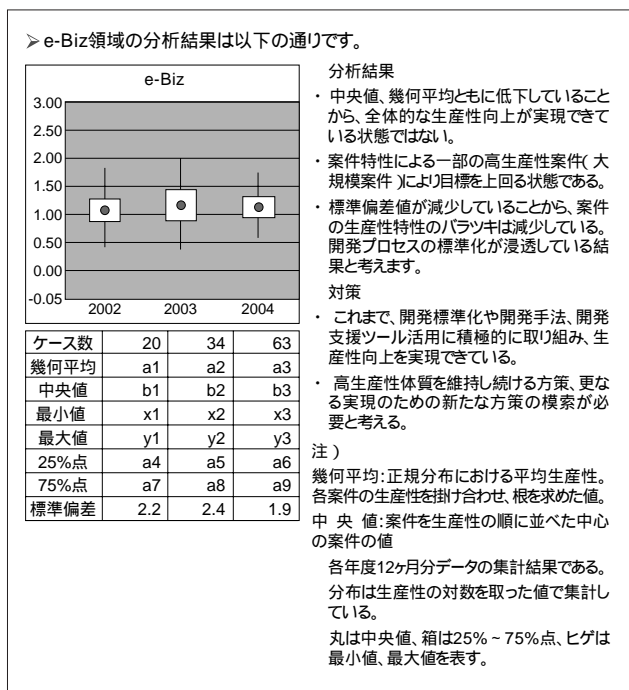


図5. 年度分析結果の事例

(3) 経済効果の評価

経済効果は次のA,Bのような形で評価することにした。まず開発領域毎に以下の計算を行う。

① 効果目標による想定自営予算 = サービス料金実績 × (1 + 目標開発生産性向上率)

② 開発生産性実績による想定自営予算 = サービス料金 × (1 + 実績開発生産性向上率)

A : 効果目標による効果額 = 領域毎の①の集計 -

サービス料金実績

B : 生産性実績による効果額 = 領域毎の②の集計 - サービス料金実績

最終的に、 $A \div B$ 100%で経済効果達成となる。

表2の例では、以下ようになる。

A : 効果目標による効果額 : 35.4-31.5 3.9億円

B : 生産性実績による効果額 : 35.7-31.5 4.2億円

・ 効果目標達成率 : $4.2 \div 3.9$ 107.7%

表2. 効果目標と生産性実績の例

開発領域	サービス料金実績	効果目標 (対基準年度)	効果目標の 想定自営予算	生産性実績 (対基準年度)	生産性実績の 想定自営予算
e-Business	5	18.0%	5.6	45.0%	7.3
レガシー1	10.5	13.0%	11.9	10.0%	11.6
レガシー2	16	10.0%	17.6	5.0%	16.8
合計	31.5		35.4		35.7

当手法により、当プロジェクト関係者の間で、各年度の経済効果を具体的な数値により評価することが可能となっている。

5. おわりに

開発生産性という指標は、単年度だけでとらえてもその数値の意味するところの判断が困難である。また、対数正規分布に従いばらつきがでることから、個々の案件レベルでは、平均より高い生産性や低い生産性の案件が発生する。そこで、開発領域毎に、アプリケーション毎の特性や、開発プロセス標準化の浸透に伴うプロセスの成熟度や、開発手法と技術の変化が、個々の案件レベルに対してどのような影響を与えるのかを把握し、全体的な開発生産性の経年変化を説明していくことが必要となる。

品質、納期に加えて、開発生産性による経済効果を明示すればAMSアウトソーシングの価値を可視化できる。日ごろから価値の可視化に取り組み、ユーザー企業とサービス提供側の間で価値の共有化を行えば、コスト削減の要望への答えとなるのみならず、サービス提供側の新たな提案への信頼にも寄与し、ユーザー企業が真の満足を得ることに繋がる。

謝辞

AMSの手法・メソドロジーや、他のAMSアウトソーシングプロジェクトの動向と事例情報をご提供頂いたAMSコンピテンシーの皆さん、開発生産性評価手法の進め方について助言を下されたPMの柘植弘夫さんとサービス営業部の池田恵美子さん、開発生産性データの整理や分析、さらに論文内容のチェックまで手伝って頂いたアルス社および日本システムランド社の担当者に深謝いたします。

参考文献

- [1] ガートナー・ジャパン , 日本のITサービス市場規模 予測(2005年上期)(Executive Summary) , 2005年8月
- [2] ガートナー・ジャパン , 2004年ITサービス・ユーザー 動向調査 : アウトソーシング(Executive Summary) , 2005年3月
- [3] 経済産業省 , 報道発表 , IT政府調達 ITサービス 評価政府ガイドラインの策定について , <http://www.meti.go.jp/kohosys/press/0005140/index.html> , 2004年4月14日
- [4] Capers Jones著 , 井上義祐 荒川淳三 監訳 , システム開発の生産性 , マグロウヒルブック , ISBN4-89501-097-X , pp.8-13 , 1986
- [5] NESMA , <http://www.nesma.nl/japanese/index.htm>
- [6] COSMIC , <http://www.cosmicon.com/>
- [7] 日本ファンクションポイントユーザー会(JFPUG) 著 , 教育コース 紹介コース教材 , <http://www.jfpug.gr.jp/>
- [8] (株)アレア著 , 失敗のないファンクションポイント法 , 日経BP社 , ISBN4-8222-8144-2 , 2002
- [9] Capers Jones著 , 鶴保征城 富野 壽 監訳 , ソフトウェア開発の定量化手法第2版 , 共立出版 , ISBN4-320-09722-X , pp.206-207 , 1998
- [10] John McGarry, David Card, Cheryl Jones, Beth Layman, Elizabeth Clark, Joseph Dean, Fred Hall著 , 古川恒夫 富野 壽 監訳 , 実践的ソフトウェア測定 , 共立出版 , ISBN4-320-09741-6 , pp.7-12 , 2004
- [11] Dennis M. Ahern, Richard Turner, Aaron Clouse著 , 前田卓雄訳 , CMMIモデルガイド , 日刊工業新聞社 , ISBN4-5260-4979-4 , 2002
- [12] Carnegie Mellon University/Software Engineering Institute著 , アクセンチュア監訳 , 成功するソフトウェア開発 - CMMによるガイドライン , オーム社 , ISBN4-274-07867-1 , 1998



株式会社JALインフォテック
アプリケーション管理グループ 次長
ICPアドバイザーPM

金子 英一 Eiichi Kaneko

【プロフィール】

1985年 , 日本IBMの社内システム開発 / 保守組織にSEとして入社 . プログラマー , SE , PMおよび管理職として開発 / 保守作業を担当すると同時に , AMSにおける組織運営やCMMIによる開発プロセスの改善などを経験 . 2001年よりJALインフォテック 出向 . 社内システム経験を基に , AMSのQC向上と価値の可視化に一貫して取り組んでいる . PMP .

eiichi.kaneko@jalinfotec.co.jp



株式会社JALインフォテック
アプリケーション管理グループ 担当次長
ITスペシャリスト

梶山 昌之 Masayuki Kajiyama

【プロフィール】

1981年 , 日本IBM入社 . 製品保証部門でソフトウェア信頼性の研究を行い習熟S字モデルを発表 . その後 , 光磁気ディスクドライブの開発 , 生産管理業務に従事 . 翻訳部門の経験を経て2003年よりAMSにてFPの分析を担当 . 現在 , JALインフォテックに出向し生産性の分析に取り組む .

masayuki.kajiyama@jalinfotec.co.jp