

Early Function Point Counting 開発初期段階でのFPカウント方法

NESMA (Netherlands Software Metrics Association) では、システム開発の初期段階で使用できる二つのFPカウント方法を開発した。:

- FP概算法 (estimated function point count)
- FP試算法 (indicative function point count)

(後者のFP試算法は、"オランダ方式 (the Dutch method)"としてFPAリストサーバ/カナダケベック大学運営に、載っている。)

ここでは、FPカウントの二つの異なる方法と、その適応性、検証の結果について説明する。

次の内容で説明を進める。

- [IFPUG法 : The \(detailed\) Function point count](#)
- [FP概算法 : The estimated function point count](#)
- [FP試算法 : The indicative function point count](#)
- [IFPUG法、FP概算法、FP試算法の比較例](#)
- [どのカウント方式をいつ使えば良いか](#)
- [100以上のプロジェクトのデータによる検証結果](#)

IFPUG法 : The (detailed) Function point count

IFPUG法は正式のカウント方法で、下記の手順で行う。

- 全てのファンクションについて、ファンクションタイプ (ILF, EIF, EI, EO, EQ) を決定する。
- 全てのファンクションの複雑度 (Low, Average, High) を評価する。
- 未調整ファンクションポイントを計算する。

[概略の説明へ戻る](#)

FP概算法 : The estimated function point count

FP概算法は、下記の手順で行う。

- 全てのファンクションについて、ファンクションタイプ (ILF, EIF, EI, EO, EQ) を決定する。
- 全てのデータファンクション (ILF, EIF) の複雑度をLowに、
全てのトランザクションファンクション (EI, EO, EQ) の複雑度をAverageとする。
- 未調整ファンクションポイントを計算する。

従って、正式のカウント方法との唯一の違いは、その複雑度の評価をファンクション毎に行うのではなく、デフォルト値で決める点である。

[概略の説明へ戻る](#)

FP試算法 : The indicative function point count

FP試算法は、下記の手順で行う。

- データファンクション (ILF, EIF) の数を決定する。
- アプリケーションの未調整FP値を下記の計算で求める。

$$\text{FP試算値 (FP)} = 35 \times \text{ILFの数} + 15 \times \text{EIFの数}$$

即ち、このFP試算値は、論理ファイル (ILF, EIF) のみに基づいて算出される。

FP試算法は、下記の仮定に基づいている。

- ILFには、平均して、3つのEI (ILFに対する追加、変更、削除) と、2つのEO、1つのEQを伴う。
- EIFには、平均して、1つのEO、1つのEQを伴う。

[概略の説明へ戻る](#)

IFPUG法、FP概算法、FP試算法の比較例

この節では、三つのFPカウント方法を簡単な事例で説明する。事例は、顧客データと製品データを維持管理し、供給者データを参照するアプリケーションである。

より正確なFP値を求めるには、より詳細なユーザ要求仕様が必要となる。そのため、この事例では三つの方法をFP試算法、FP概算法、IFPUG法の順番で説明する。

FP試算法

ユーザ要求 :

- ユーザは、顧客データと、製品データを維持管理し、供給者データを参照することを要求している。

この (ラフな) 仕様でも、FP試算法には十分である。

- ILF: 顧客データ、製品データ
- EIF: 供給者データ

データファンクション	ファンクションタイプ	FP値 (デフォルト)
顧客データ	ILF	35
製品データ	ILF	35
供給者データ	EIF	15
FP試算値		85 fp

FP概算法

FP概算法を行うには、トランザクション・ファンクションについての情報も必要なので、より詳細なユーザ要求が必要となる。

ユーザ要求：

- ユーザは、顧客データに対して、追加、更新、削除、及び照会、そして、計算データを含む4種類のレポートを要求している。
- ユーザは、製品データに対して、追加、更新、削除、及び照会、そして、計算データを含む1つのレポートを要求している。
- ユーザは、供給者データに対して、供給者番号での照会機能と集計結果の帳票出力を要求している。

これら、より詳細なユーザ要求仕様には、実際のトランザクション・ファンクションの総数が示されていて、FP概算法が可能になる。

データ/トランザクション ファンクション	ファンクション タイプ	複雑度 (デフォルト)	FP 値 (未調整 値)
顧客データ	ILF	Low	7
製品データ	ILF	Low	7
供給者データ	EIF	Low	5
顧客データの追加	EI	Average	4
顧客データの更新	EI	Average	4
顧客データの削除	EI	Average	4
顧客データの照会	EQ	Average	4
顧客レポート1	EO	Average	5
顧客レポート2	EO	Average	5
顧客レポート3	EO	Average	5
顧客レポート4	EO	Average	5
製品データの追加	EI	Average	4
製品データの更新	EI	Average	4
製品データの削除	EI	Average	4

製品データの照会	EQ	Average	4
製品関連のレポート	EO	Average	5
供給者番号照会	EQ	Average	4
供給者関連のレポート	EO	Average	5
FP概算値			85 fp

IFPUG法

IFPUG法を行うには、ファンクションタイプ (ILF, EIF, EI, EO, EQ) 毎の数だけでなく、個々の機能について複雑度 (Low,Average,High) も決める必要がある。FPA法では、データファンクションとトランザクションファンクションの複雑度は、その機能に関連するDET,RET,FTRに基づいて決まる。このため、ユーザ要求（この事例のFP概算法の説明で記述しているが）をより詳細に分析する必要がある。つまり、どのDETとFTRがトランザクションファンクション (EI,EO,EQ) で使われているか、どのようなRETとDETでデータファンクション (ILF,EIF) が構成されているか、である。このようにユーザ要求の詳細な分析をすると、その結果、例えば以下のようにFP値が求められる。

データ/トランザクション ファンクション	ファンクション タイプ	複雑度	FP値 (未調整値)
顧客データ	ILF	Average	10
製品データ	ILF	Low	7
供給者データ	EIF	Low	5
顧客データの追加	EI	High	6
顧客データの更新	EI	Average	4
顧客データの削除	EI	Low	3
顧客データの照会	EQ	Low	3
顧客レポート1	EO	Low	4
顧客レポート2	EO	Average	5
顧客レポート3	EO	Low	4
顧客レポート4	EO	High	7
製品データの追加	EI	Average	4
製品データの更新	EI	Low	3
製品データの削除	EI	Low	3

製品データの照会	EQ	Average	4
製品関連のレポート	EO	Average	5
供給者番号照会	EQ	Low	3
供給者関連のレポート	EO	Average	5
FP値			85 fp

結論

この特殊な事例では、三つのカウント方法が、85FPで同じ機能規模となった。計測結果は、通常同じとはならないが、かなり近い値となる。この報告の最後に、FP概算法とFP試算法の精緻な検証結果を載せる。

[概略の説明へ戻る](#)

どのカウント方式をいつ使えば良いか

正式のFPカウント方式は、もちろん概算法や試算法に比べ、より正確である。しかし、より多くの時間と、より詳細な仕様を必要とする。

どのカウント方式を使うかは、プロジェクトマネージャの判断と、システムのライフサイクルのフェーズに依存している。

多くのアプリケーションに適用した結果、FP試算法は、驚くほど良い見積もり結果を示している。データモデルは、ほとんど労力を掛けずに作成または入手できるため、多くの場合FP試算法で算出する事は比較的容易である。

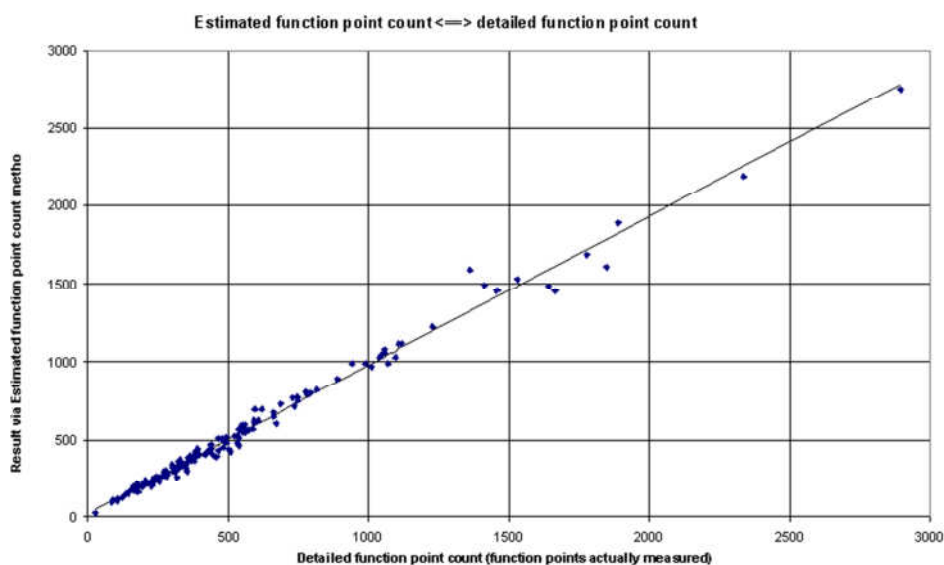
[概略の説明へ戻る](#)

100以上のプロジェクトのデータによる検証結果

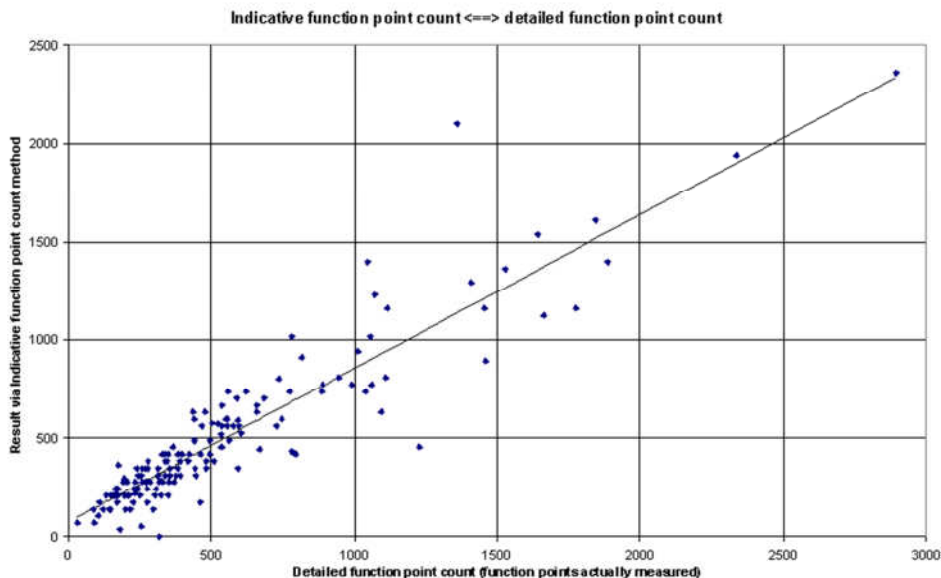
およそ100の開発済みのアプリケーションのデータを使用し、NESMAではFP概算法及びFP試算法の精度について調査を行った。

提供されたアプリケーションについて、3通りのカウント方法で同時に測定した。その結果を下記の2つのグラフに示す。

- FP概算値(縦軸)と正式なFP測定値(横軸)の相関



- FP試算値(縦軸)と正式なFP測定値(横軸)の相関



どちらのケースも、良い相関（直線的な）を示している。しかしながら、試算法のグラフにおいては、いくつかのケースで、かなりの差異（50%に到るずれ）が見られる。これが、FP試算法を使用する際の注意点である。このカウント方法の強みは、非常に短い時間で、アプリケーションのラフな規模見積りができることである。

出力の数が平均に比べ多い（または少ない）場合には、35や15という係数を変更することが必要になるだろう。しかし、この方式の考え方は、一般的に使用できるものである。

FP概算法での見積結果と、正式な計測結果とは非常に近い値となっている

[概略の説明へ戻る](#)

このドキュメントは、日本ファンクションポイントユーザー会 (JFPUG) が翻訳しました。

This document is translated by JFPUG (Japan Function Point User's Group).