

■ ファンクションポイント(Function Point)=FP

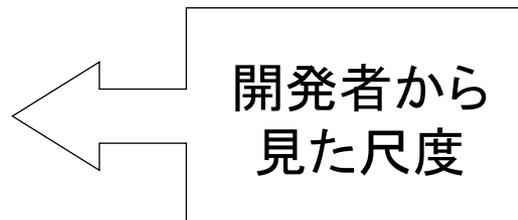
- ソフトウェアの「開発規模」を測る尺度のひとつ
=大きさ(サイズ)
- 単位:FP
 - 例. このソフトの規模は820FPだ。
- Allan J.Albrechtが、
1979年に考え方を発表し、
1984年に現在に近い形で提唱
- 1986年秋 ファンクションポイント法の推進団体IFPUGが正式発足

■ファンクションポイント法(Function Point Analysis)＝FP法

- FPを測るためのルール
- ソフトウェアの「機能」を点数化
 - データストア 例. DBテーブル
 - データ入出力処理 例. 画面での入出力処理
- ユーザが理解できるもの

■これまでの規模の尺度

- 開発費用
- 開発工数
- ステップ数 (Lines of Code)
-



1-3 FPと他の規模尺度の相違

No.	規模尺度	特徴
1	FP	<ul style="list-style-type: none">・Web, GUI, オブジェクト指向等にも適応。・ユーザにとって理解しやすい。・国際標準化が進行中。・算出作業は教育と手法への習熟が必要。
2	ステップ数	<ul style="list-style-type: none">・自動測定が可能。・プログラム言語が異なると比較できない。・メインフレーム&COBOLでは開発コストとの相関関係大。・同じ要件の実装でも技術者によって規模が大きく異なる。
3	画面帳票数	<ul style="list-style-type: none">・ユーザにとって理解しやすい。・画面の設計/実装の仕方によって、同じ要件でも個数が異なる。・処理量, アルゴリズム, データ量などの複雑さが反映されない。
4	ドキュメント ページ数	<ul style="list-style-type: none">・ベンダにとってイメージしやすい。・事前の完成量予測が難しい。・開発コストとの相関関係大。

(参考説明)

※ ステップ数: プログラムのソースコードの行数

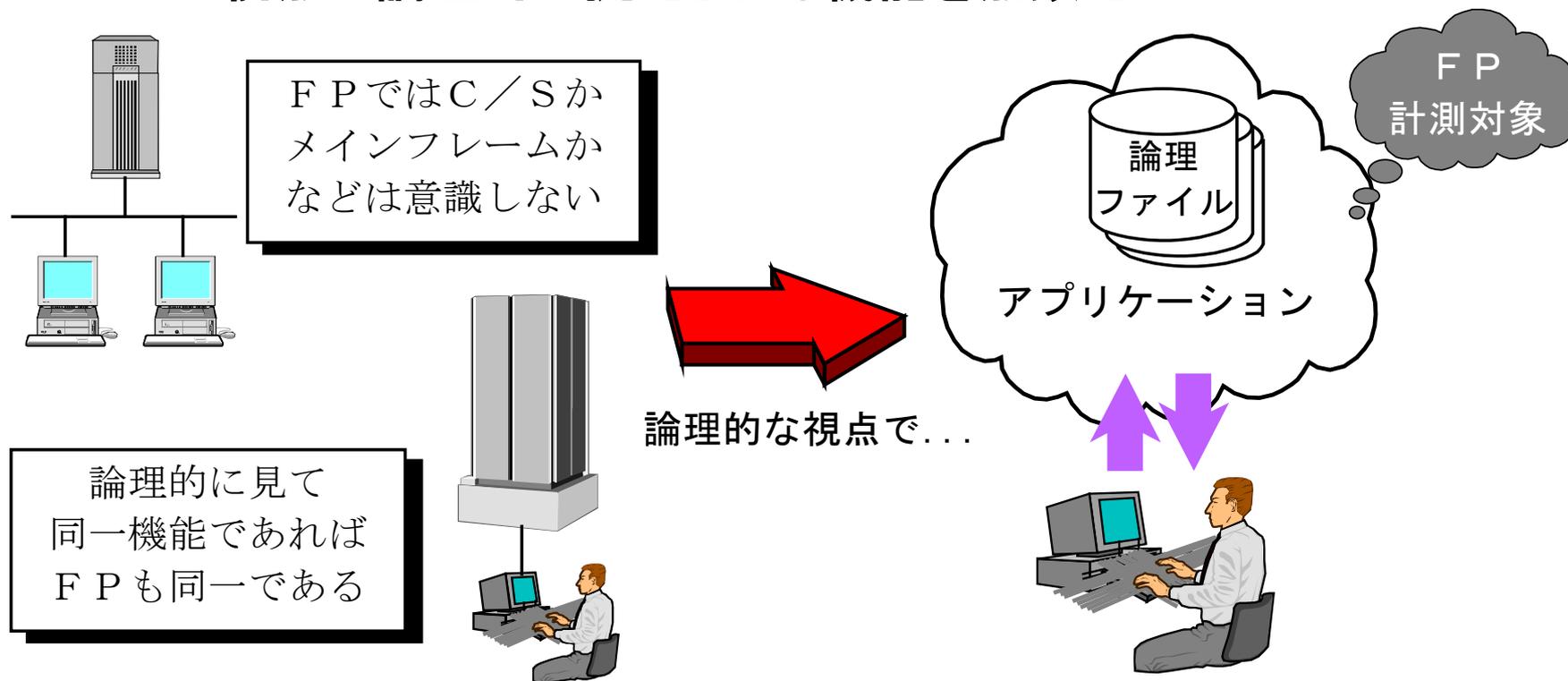
➢ 実は数え方の標準が存在しない。各社で数え方が異なる。

※ ドキュメントページ数: ここで言うドキュメントは、開発で作成する納入物, 成果物が対象。

➢ 設計メモ等非公式なドキュメントは含まない。

⇒ FPだけでは開発規模を的確に把握できない。さまざまな尺度の開発規模を常に把握し、状況に応じて使い分けることが大切。

■ユーザの視点で論理的に捉えられる機能を点数化

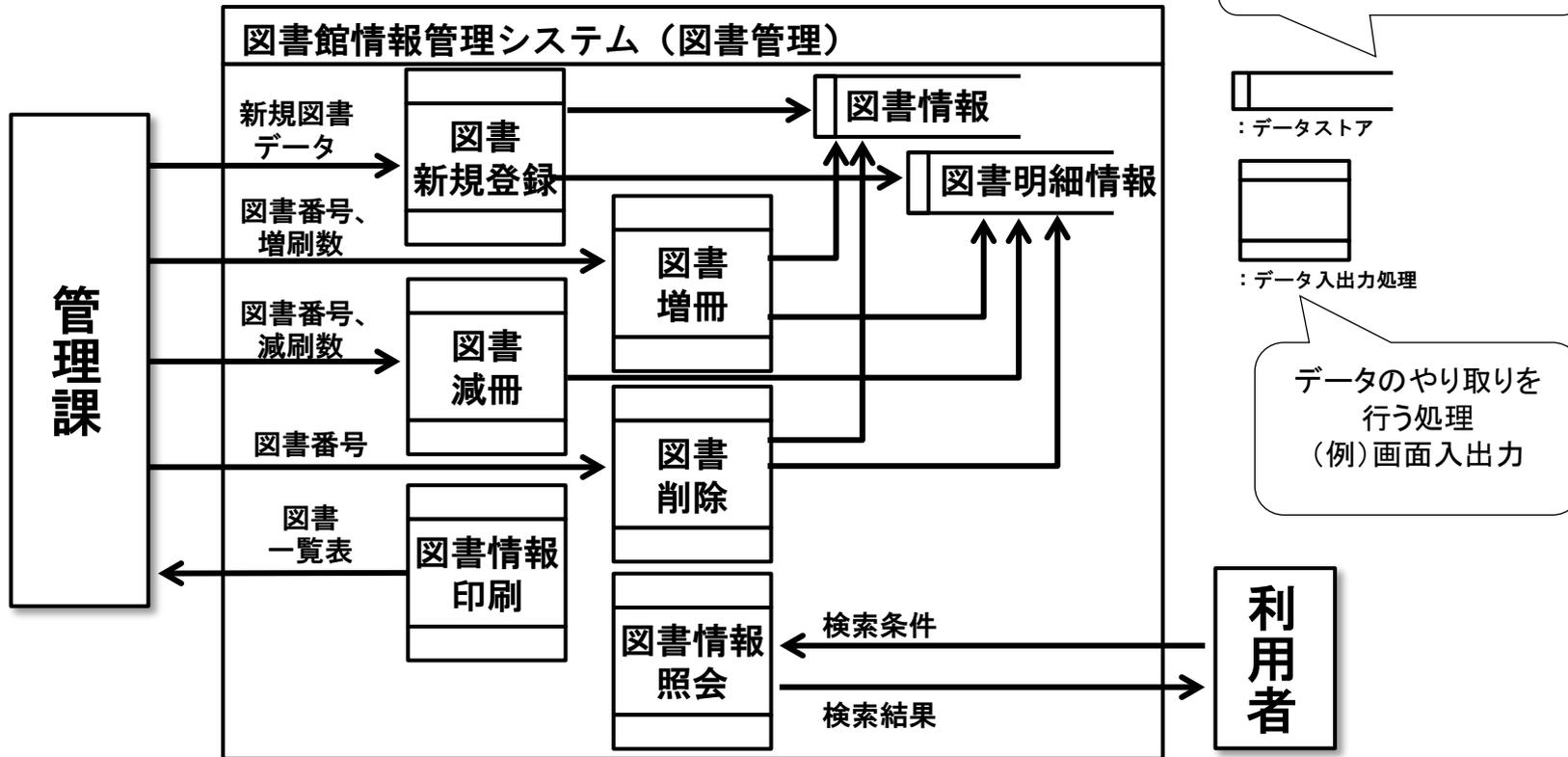


- ユーザから見るとソフトウェアの“対価”を判断する有効な尺度
...「作成される機能に対して対価を払っているのであって、ソースコード量(ステップ数)や作業量(人月)に払っているのではない」

1-5 FP法の基本(1)

■ 下記2種類の機能を点数化

- データストア 例. DBテーブル
- データ入出力処理 例. 画面での入出力処理



■ 2種類のデータストア

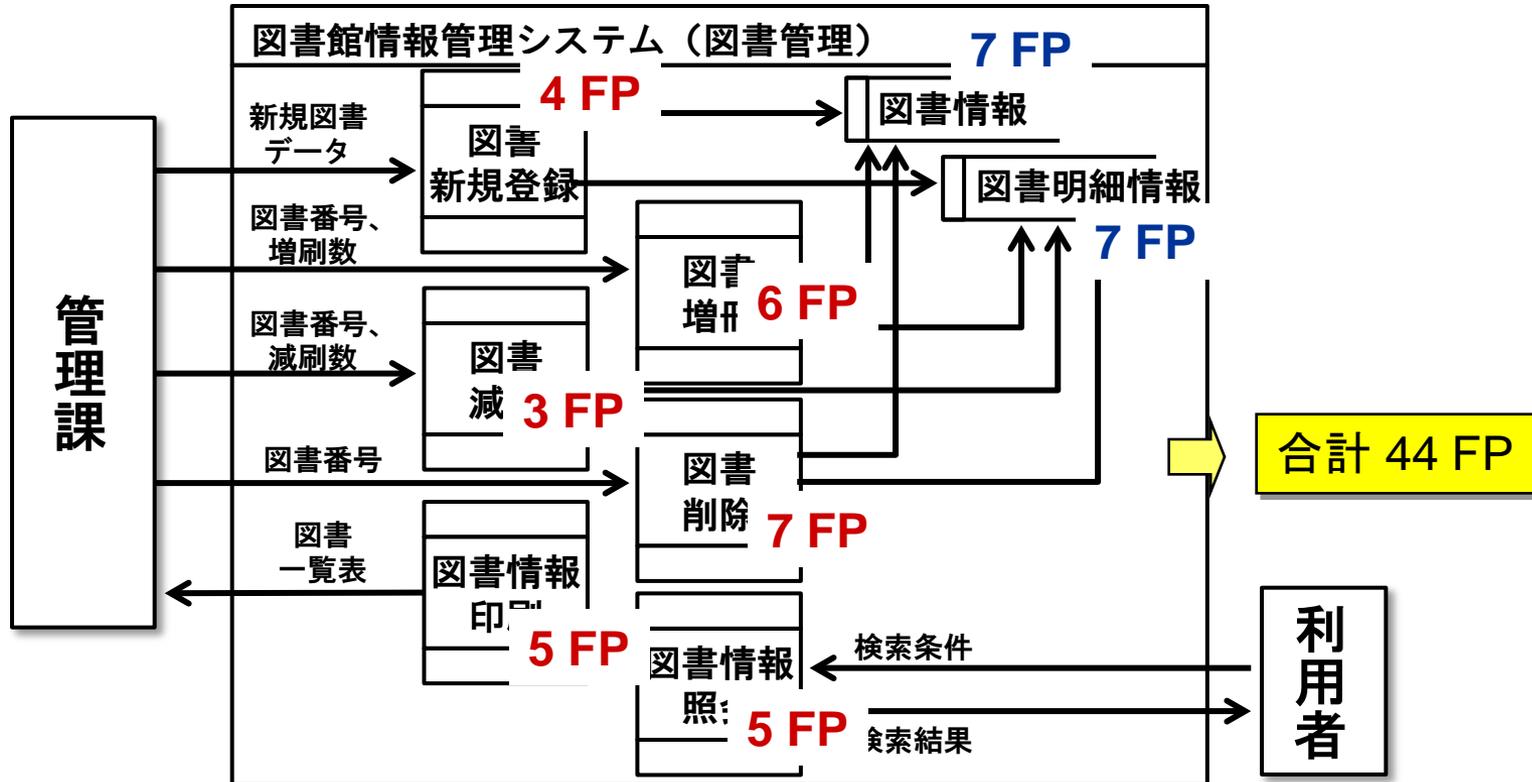
■ 6種類のデータ入出力処理

を持つアプリケーションがあった場合

1-6 FP法の基本(2)

■ 下記2種類の機能を点数化

- データストア 例. DBテーブル
- データ入出力処理 例. 画面での入出力処理



■ ユーザが理解できるものを点数化

■ 実装技術(プログラミング言語, 開発支援ツール等)に左右されない

- Webでもメインフレームでも同じ機能なら同じ点数