

Title	要求仕様書に対するファンクションポイント計測ツールの試作
Author(s)	柏本, 隆志; 石田, 直也; 神谷, 年洋 他
Citation	電子情報通信学会技術研究報告. SS, ソフトウェアサイエンス. 1998, 97(630), p. 87-93
Version Type	VoR
URL	https://hdl.handle.net/11094/26679
rights	Copyright © 1998 IEICE
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

要求仕様書に対するファンクションポイント計測ツールの試作

柏本隆志[†] 石田直也[†] 神谷年洋[†] 楠本真二[†] 井上克郎[†]

鈴木文音[‡] 勝瑞雅也[‡] 湯浦克彦[‡] 津田道夫[‡]

[†]大阪大学大学院 基礎工学研究科 情報数理系専攻

〒560-8531 大阪府豊中市待兼山町1-3 Tel 06-850-6571, Fax 06-850-6574

E-mail: {kasimoto, ishida, kamiya, kusumoto, inoue}@ics.es.osaka-u.ac.jp

[‡]株式会社日立製作所

あらまし ソフトウェアを測定する上で規模は基本となる尺度である。ソフトウェアの規模を測定する手法の一つにファンクションポイント法がある。ファンクションポイント法は、開発するシステムに存在するデータを5種類のタイプに分類し、それぞれ点数をつけて数値化したものである。しかし、ファンクションポイント法は計測者によっては30%以上の誤差が出ることや開発プロセスの最上流過程で作成される要求仕様書に対する計測方法は確立されていないといったことがある。本研究では、要求仕様定義支援ツールREQUARIOで作成された要求仕様書に対してファンクションポイントを計測するための詳細な手順を提案し、その手順に基づくファンクションポイント計測ツールの試作を行なった。開発したツールは、VC++5.0で作成しWindows95上で動作する。サイズは約7000行である。また、実際の現場で作成された仕様書に対するツールの適用も行なった。

キーワード ファンクションポイント法, 要求仕様, 開発計画, 開発管理, 見積り

Function Point Measurement Tool for Requirements Specification

Takashi KASIMOTO[†] Naoya ISHIDA[†] Toshihiro KAMIYA[†] Shinji KUSUMOTO[†]

Katsurou INOUE[†] Ayane SUZUKI[‡] Masaya SHOZUI[‡] Katsuhiko YUURA[‡] Michio TSUDA[†]

[†] Graduate School of Engineering Science, Osaka University, 1-3, Machikaneyama, Toyonaka,

Osaka 560-8531, Japan

Tel: +81-6-850-6571, Fax +81-6-850-6574

E-mail: {kasimoto, ishida, kamiya, kusumoto, inoue}@ics.es.osaka-u.ac.jp

[‡]Hitachi,Ltd.

Abstract The method of Function Point Analysis(FPA) was proposed by Albrecht to help measure the size of a computerized business information system. It is widely used in the actual software development. However, it has been pointed out that FPA leads to 30 percent difference for same product by people. Also, the application of FPA to requirements specification has not been established. In this paper, we propose detailed FPA measurement method for the requirements specification analyzed using the requirements analysis system REQUARIO, and develop the function point measurement tool based on the method. The tool is implemented by Visual C++5.0 on Windows95. The size of the tool is approximately 7000 lines. Moreover, we have applied the tool to requirements specification developed in a computer company.

Key word Function Point Analysis, requirements specification, development plan, software development management, estimation

1 まえがき

現代のような高度情報化社会において、コンピュータシステムへの依存度が高まるにつれて、その主要な構成要素であるソフトウェアはますます大規模化・複雑化・多様化してきている。さらに、ソフトウェアの需要量も増大し、開発期間の短縮化が求められるようになってきた。このような背景から、高い品質を持ったソフトウェアを効率良く開発するために、明確な開発計画の下で開発プロセスの全工程を系統づけて管理する必要性が高まってきている。

明確な開発計画を立てる上で、開発中に起こるさまざまな事象を予測し、あらかじめ必要な手段を講じておくことは重要である。ソフトウェア開発に関して予測の対象となるものに規模、投入する工数、開発期間、開発に使用される技術、品質などがある。中でも重要なものは開発工数と開発期間である。

通常、開発工数や開発規模を予測するのに、まずソフトウェアの規模を見積もり、これに基づいて開発工数と開発期間を予測する手法がとられている[5]。ソフトウェアの規模を見積もる手法として、ファンクションポイント法(FP法)が提案されている。ファンクションポイント法は、Albrechtによって1979年に提案された[5]。その後、これをベースに種々のファンクションポイント法が提案されている。現在、IFPUG(International Function Point Users Group)法[2]とMarkII法[7]の2つの主流技法がある。また、ISO規格の制定作業も始まっており、参考規格として制定される見込みである。

しかし、ファンクションポイント法には、次のような問題が指摘されている。まず、同じ組織内の人間が同じプロダクトに対して測定した場合は12%、違う組織の人間が測定した場合は30%以上の誤差が出る[1]。また、ファンクションポイント法は開発されたプログラムや設計書に対する計測方法は提案されているが、開発プロセスの最上流過程で作成される要求仕様書に対しての計測方法は確立されていない。

本研究では次の(1)、(2)を目的とする。(1)開発の最上流工程で作成される要求仕様書に対してファンクションポイントを計測すること。(2)計測者によって誤差が出ないようにファンクションポイント計測ツールを試作すること。具体的には、要求仕様定義支援ツールREQUARIOで作成された要求仕様書に対してファンクションポイントを計測するための詳細な手順を開発し、その手順に基づくファンクションポイント計測ツールの試作を行なった。

以降、2.ではソフトウェアの規模を測定するための手法であるファンクションポイント法について述べる。3.では要求仕様定義支援ツールREQUARIOについて述べる。4.ではファンクションポイント計測手法につ

いて述べる。5.ではファンクションポイント計測ツールについて述べる。最後に、6.でまとめと今後の課題について述べる。

2 ファンクションポイント法

2.1 概要

ファンクションポイント法とは、ソフトウェア内部で使用されたり生成されたりするデータからソフトウェアの機能を測定し定量化したものである[10]。

ファンクションポイント法の長所は

- ユーザの要求する機能を測定できる。
- 開発で用いられる技術に関係なく、ソフトウェア開発と機能改良について測定できる。

という二点である[2]。ファンクションポイント法には種々のバリエーションがある。例えば、IBM法[5]、IFPUG法[2]、3D Function Points法[4]、Feature Point法[4]、MarkII法[7]などがある。

本研究では、数多くのファンクションポイント法の中から、計測するためのルールが詳細に設定されているIFPUG法を使用してファンクションポイントを計測する。

2.2 IFPUG法

IFPUG法は、Albrechtのファンクションポイント法に対して複雑さの評価の客観化やルールの精密化・適正化などの変更を行なったバージョンである。

IFPUG法は次のStep1～Step7でファンクションポイントを計測する[2]。

Step1 (算出種類の選択): 算出種類を、

- アプリケーションFP
アプリケーションソフトウェアの大きさを表すファンクションポイント。
- 新規開発プロジェクトFP
新規にアプリケーションを開発するプロジェクトの規模を知るために使用するファンクションポイント。
- 機能改良プロジェクトFP
すでに存在するアプリケーションをエンハンスするプロジェクトの規模を知るために使用するファンクションポイント。

の3種類の中から選択する。

Step2 (計測境界の設定): ファンクションポイントを測定する範囲(計測境界)を設定する。

Step3 (データファンクションの計測): まず, アプリケーション中にあり, ユーザが認識できる論理的な意味でのデータのまとまりをデータファンクションとして抽出し, ファンクションタイプを決定する. ファンクションタイプには以下の2種類がある.

- 内部論理ファイル
アプリケーション内でデータが更新される論理的な関連を持ったデータの集合.
- 外部インタフェースファイル
アプリケーションによってデータが参照されるデータの集合.

次に, タイプ毎に用意されている複雑さ決定表に従って複雑さを計測する(表1参照).

表1: データファンクションの複雑さ決定表

RET \ DET	1-19	20-50	51-
1	低	低	中
2-5	低	中	高
6-	中	高	高

RET:レコード種類数 DET:データ項目

Step4 (トランザクションファンクションの計測): まず, アプリケーションに対するデータの出入りを伴う処理をトランザクションファンクションとして抽出し, ファンクションタイプを決定する. ファンクションタイプには以下の3種類がある.

- 外部入力
アプリケーション境界を越えてシステムにデータを入力する機能.
- 外部出力
アプリケーション境界を越えてシステムにデータを出力する機能.
- 外部照会
システム内のデータを参照する機能.

次に, タイプ毎に用意されている複雑さ決定表に従って複雑さを計測する.

Step5 (未調整ファンクションポイント算出): Step3, Step4の結果をもとに, 種類別重み別に個数を数え, それぞれに対して重みづけをし, 加え合わせる. 加え合わせたものが未調整ファンクションポイントとなる.

Step6 (調整係数の算出): 表2に示すシステム特性の14項目を6段階で評価し, 14項目の全ての評価値

の合計を求め, システム特性の評価値とする. そして次の式で調整係数を求める.

$$\text{調整係数} = 0.65 + \frac{\text{評価値の合計}}{100}$$

Step7 (最終ファンクションポイント算出): 未調整ファンクションポイントと調整係数を用いて最終ファンクションポイントを算出する.

表2: システム特性

項番	システム特性
1	処理の複雑さ
2	データ通信機能
3	データのオンライン
4	オンライン更新
5	分散データ処理
6	複数組織での使用
7	トランザクション量
8	性能条件
9	高負荷環境での使用
10	エンドユーザの使い易さ
11	ユーザに提供される変更機能
12	システム運用の容易さ
13	再利用の考慮
14	インストールの容易さ

3 オブジェクト指向要求仕様書

3.1 対象とする要求仕様書

要求仕様書とは, ソフトウェアの使用目的や外部から与えられる条件を明確にし, 課題や仕事を理解し, それを検討したものを厳密かつ完全に記述したものである.

本研究で使用する要求仕様書は, REQUARIO[3]という要求仕様定義支援ツールで作成されたものとする. REQUARIOは開発するシステムへの要求仕様をアニメーションで表示するツールである. 従来の要求仕様書では理解しにくかったシステムの動作順序やデータの流れが短期間に, 正確に理解できるという特長を持っている.

3.2 REQUARIOの概要

REQUARIOでは, ストーリー, シナリオ, シーン, キャラクターという概念で要求仕様を定義する. これらの概念構成を図1に示す[3].

3.2.1 ストーリー

ストーリーは, 要求仕様の断片的な具体例の集合である. REQUARIOの成果物をストーリーという単位で管理している.

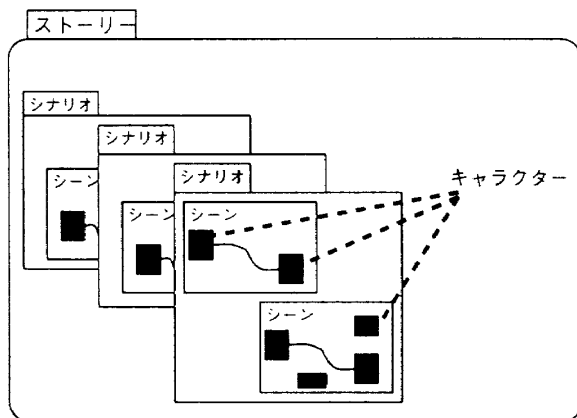


図 1: REQUARIO の構成

3.2.2 シナリオ

ストーリー中の「要求仕様の断片的な具体例」をシナリオと言う。シナリオは、ストーリーに登場する「ひと」あるいは「もの」と、その「ふるまい」(動作・処理)の流れから構成されている。つまり、「誰が(どこが)」、「誰に(どこに)」、「何を(どのようなイベントを)」発行するかを表したものである。

3.2.3 シーン

ストーリーに登場する「ひと」あるいは「もの」をキャラクターと言い、その「ふるまい」(動作・処理)の流れをシーンと言う。REQUARIOは、最小単位であるシーンを定義していくことでシナリオを作っていく。

シーンには、次のような表現を付加することができる。

- キャラクターが持っているデータが自分自身または他のキャラクターのデータ代入によって変化する様子。
- シーンに関係するキャラクターの関与する様子。

図 2 にシーンの一例を示す。このシーンでは、顧客が営業に見積依頼書を渡し、見積を依頼するというふるまいを表現している。

3.2.4 キャラクター

ストーリーに登場する「ひと」あるいは「もの」をキャラクターと言う。キャラクター同士がシーンを構成していくことでシナリオとなり、ストーリーとなる。

キャラクターは2つの属性を持っている。ひとつは、キャラクターの持つデータを表現した“属性”と、もうひとつは、キャラクターの持つふるまいを表現した“メッセージ”である。

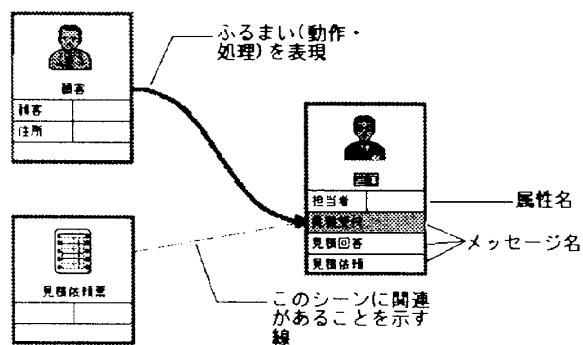


図 2: シーンの表示例

4 ファンクションポイント計測手法

4.1 基本方針

REQUARIOで作成された要求仕様書に対してIF-PUG法を適用するために、次のStep1～Step7より成る手法を用いる。

Step1 (算出種類の選択):

算出種類は新規開発プロジェクトファンクションポイントのみにする。

Step2 (計測境界の設定):

計測境界はREQUARIOのストーリー単位とし、計測するストーリーファイルはユーザが指定する。

Step3 (データファンクションの計測):

指定したストーリーファイルのデータファンクションを、4.2で説明するルールに従って抽出する。

Step4 (トランザクションファンクションの計測):

指定したストーリーファイルのトランザクションファンクションを、4.3で説明するルールに従って抽出する。

Step5 (未調整ファンクションポイント算出):

Step3, Step4の結果をもとにファンクションタイプ別複雑さ別に個数をカウントし、それぞれに対して重み付けをし、加え合わせて未調整ファンクションポイントを算出する。

Step6 (調整係数の算出):

システム特性の影響度をユーザに入力してもらい、調整係数を算出する。

Step7 (最終ファンクションポイント算出):

未調整ファンクションポイントと調整係数から最終ファンクションポイントを算出する。

4.2 データファンクション抽出ルール

IFPUG法でのデータファンクションの抽出ルールとファンクションタイプの決定ルールは次のように決められている [2].

- (1) ユーザが認識できるものである.
- (2) マスタファイルまたはコード表に相当するデータのまとまりであり, それらはアプリケーションによって永久に保持される対象である.
- (3) 内部論理ファイル
計測対象のアプリケーションによってデータが更新(生成・追加・変更・削除)されるデータファンクションは内部論理ファイルである.
- (4) 外部インターフェースファイル
計測対象のアプリケーションによってデータが参照されるデータファンクションは外部インターフェースファイルである.

(1)については, REQUARIOの記述でユーザが認識できるのは, キャラクター単位であるのでキャラクターごとに判別するようにした.

(2)については, データのまとまりはDBで表現されることが多いので, DBで表現されるキャラクターをデータファンクションにすることにした.

(3), (4)については, REQUARIOの記述からはデータが参照されているかどうかを判別することができない. そこで, データが更新されていれば内部論理ファイル, データが更新されていれば外部インターフェースファイルと判別するようにした.

複雑さの判定は, データ項目数とレコード種類数によって決定される. データ項目数はデータファンクションを構成しているデータ項目の数なので, キャラクターの属性の数を数えることにした. レコード種類数は, REQUARIOの記述では判別できないが, REQUARIOを使用して作られた要求仕様書ではほとんどが1種類なので, レコード種類数は1種類とすることにした.

以上より, データファンクションの抽出ルールを以下のようなものとした.

- (1) ユーザは, データファンクションの指定方法を以下の3つから複数選択することができる.
 - DBのビットマップで表示されているキャラクターをデータファンクションにする.
 - キャラクター名の中の識別子(複数指定可)で判断する.
 - キャラクターのビットマップの種類(複数指定可)で判断する.

(2) (1)の選択をもとにして抽出されたデータファンクションの中で, データを更新されるものを内部論理ファイル, 更新されないものを外部インターフェースファイルとする.

(3) レコード種類数は1種類, データ項目はキャラクターの属性の数とし, 複雑さ決定表を用いて各データファンクションの複雑さを判定する.

4.3 トランザクションファンクション抽出ルール

IFPUG法でのトランザクションファンクションの抽出ルールとファンクションタイプの決定ルールは次のように決められている [2].

- (1) ユーザにとって意味があり最小作業単位となる処理である.
- (2) 計測境界外と可変なデータの入出力を伴う処理である.
- (3) 外部入力
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が更新されるトランザクションファンクションは外部入力である.
- (4) 外部出力
計測境界外へのデータ出力を含む処理のうち, 出力データに派生データを含むトランザクションファンクションは外部出力である.
- (5) 外部照会
計測境界外へのデータ出力を含む処理のうち, 出力データに派生データを含まず, データファンクションを更新しないトランザクションファンクションは外部照会である.

(1)については, REQUARIOの記述で最小作業単位はシーンなので, 1シーンごとに判別するようにした.

(3), (4), (5)については, シーンにデータファンクションが関係しているものをトランザクションファンクションとした. またファンクションタイプの判別は, データファンクションの更新の有無, 出力されたデータ項目の比較で判別できるので, それに合わせてルールを決定した.

複雑さの判定は, データ項目数と関連ファイル数によって決定される. まず, データ項目数は入出力されたデータファンクションの属性の数とした. また関連ファイル数はトランザクションファンクションとして抽出されたシーンに登場するデータファンクションの数とした.

トランザクションファンクションの抽出ルールを以下のようなものにした。

- (1) ルール(図3参照)に従い、トランザクションファンクションを抽出し、その種類(外部入力, 外部出力, 外部照会)を決定する。
- (2) 関連ファイル数はシーンに参与しているデータファンクションの数, データ項目は入出力されたデータファンクションの属性の数とし, 複雑さ決定表を用いて, 各トランザクションファンクションの複雑さを判定する。

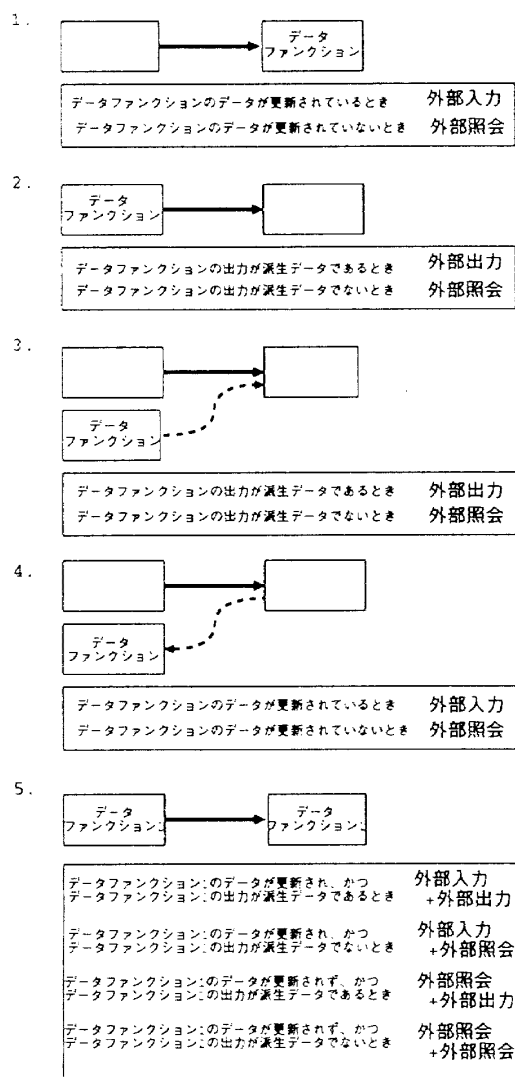


図3: トランザクションファンクション判別ルール

5 ファンクションポイント計測ツール

5.1 動作環境

Windows95上で動作するツールをMFC(クラスライブラリ)を用いて, C++で実装した。プログラムサイズは約7000行である。

5.2 システム構成

システム構成を図4に示す。

- 解析部: REQUARIOの出力ファイルを構文解析し, ファンクションポイント計測のために必要であるデータを抽出する。
- 解析DB: 解析部で抽出したデータを格納しておく。
- 計測部: 実際にファンクションポイントを計算する。
- 計測DB: 計測結果(ファンクションポイント, 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)を格納しておく。
- インターフェース部: 計測結果(ファンクションポイント, データファンクションの候補, トランザクションファンクションの候補)を表示する。

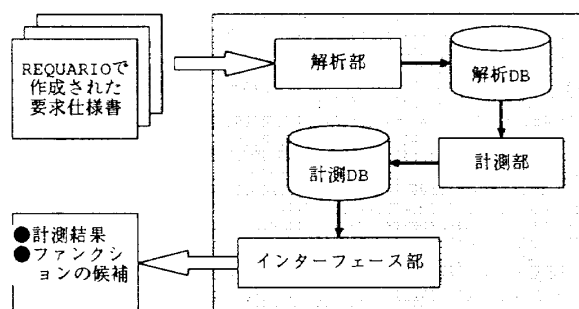


図4: システム構成

5.3 適用例

REQUARIOで作成した在庫管理システム[11]の要求仕様書を入力として, ファンクションポイント計測ツールを使用した。要求仕様書はREQUARIOの出力ファイルで22KBで, シナリオ数10, シーン数23, キャラクター数10であった。

ツールの画面例を図5に示す。

6 まとめと今後の課題

REQUARIOで作成された要求仕様書に対して, ファンクションポイントを求める手法を提案し, その手順に基づくファンクションポイント計測ツールを試作した。

今後の課題としては, 開発したツールを現場で適用し, ツールが計算した値とファンクションポイント計

機能度	低	中	高	合計
内部処理ファイル	1 × 7 = 7	0 × 10 = 0	0 × 15 = 0	7
外部インターフェースファイル	1 × 5 = 5	0 × 7 = 0	0 × 10 = 0	5
外部入力	2 × 3 = 6	0 × 4 = 0	0 × 6 = 0	6
外部出力	2 × 4 = 8	0 × 5 = 0	0 × 7 = 0	8
外部呼出	5 × 3 = 15	0 × 4 = 0	0 × 6 = 0	15
未認識ファンクションポイント				44

(a) メインスクリーン

機能名	種別	数
任意不定長格納	その他	1
任意不定長格納	ファイル	1
任意不定長格納	外部インターフェースファイル	1

(b) データファンクションの一覧表示

機能名	種別	数
任意不定長格納	その他	1
任意不定長格納	ファイル	1
任意不定長格納	外部インターフェースファイル	1
任意不定長格納	外部入力	1
任意不定長格納	外部出力	1
任意不定長格納	外部呼出	1

(c) トランザクションファンクションの一覧表示

図 5: ファンクションポイント計測ツールの画面例

測の熟練者が測定した値を比較し、評価する。また、今回提案したファンクションポイント計測手法やツールのインターフェースの改良と拡張、及び一般のオブジェクト指向要求仕様書に対してもファンクションポイントが算出できるようにルールを拡張するといったことがある。

謝辞 本研究に関して御協力頂いた株式会社日立製作所、初田賢司氏、森田真理氏、倉重誠氏に深謝致します。

参考文献

[1] B.A.Kitchenham, "The Problem with Function Points", IEEE Software, Vol.14, No.2, March/April, 1997, pp.29-31.

[2] IFPUG: "Function Point Counting Practices Manual, Release 4.0", International Function Point Users Group (1994).

[3] 日立: "要求仕様書定義支援 REQUARIO"(1997).

[4] 中村 永: "科学技術計算とリアルタイム制御に向く

ソフト計測手法", 日経エレクトロニクス, no.658, pp.175-185(1996-03).

[5] 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", 情報処理学会誌, 35, 4, pp.289-298(1994-04).

[6] 西山 茂: "機能的規模見積技法の標準化の動向", 研究報告 vol.96, no.71 (96-SE-110), pp.31-38, 情報処理学会, 1996年7月.

[7] C.Symons: "Software Sizing and Estimating", John Wiley & Sons(1991).

[8] 高橋 光裕: "機能量尺度に基づく情報システムのコスト評価", 技術研究報告, vol.96, no.283, pp.25-32(SS96-21), 電子情報通信学会, (1996-09).

[9] 高橋 光裕, 菱谷 淳: "ソフトウェア見積もり手法を日本の事情に合わせて改良", 日経エレクトロニクス, no.625, pp.91-101(1995-01).

[10] 山田, 高橋: "ソフトウェアマネジメントモデル入門", 共立出版(1993).

[11] 山崎利治: "共通問題によるプログラム設計技法解説", 情報処理学会誌, 25, 9, p.934(1984).