

## 品質メトリクスの適用と基準値設定に関する一考察

4L-4

増沢 香、山田 淳、岡野 浩  
株式会社 東芝

## 1 はじめに

ソフトウェアの開発では、開発の初期段階から品質に注意し、品質を作り込んでいくことが重要であり、さまざまなメトリクスが利用されている（[1] 参照）。今回、実システム開発のプログラム作成工程において品質メトリクスを適用した。一般に、プログラムの規模を元にして、検出する不具合数の推測をする場合が多いが、本論文ではプログラムの複雑度を考慮に入れた品質メトリクスを使用する。また、品質メトリクスの利用では基準値の設定が難しいが、その設定方法について考察し、設定指針を与える。

## 2 重点的探査プロセス

我々は、レビュー、テストコスト（工数）の制限内で最適な不具合検出を行なうために、重点的探査プロセスモデルを設定している。これは、全体を一様に探査するのではなく、不具合を多く含む兆候を示した部分に重みをおいて探査することにより、効率的に不具合を検出しようというプロセスである。ここでは、プログラムのレベルで、こうした部分を要注意プログラムとして抽出することを行なう。

## 3 品質メトリクスの適用

今回、試行に用いたシステムは、

- 言語：COBOL
- 種類：事務処理アプリケーションソフト
- 規模：プログラム数 63  
最小 29(LOC)～最大 1952(LOC),  
平均 505(LOC), 合計約 34(KLOC)

であり、プログラム作成工程において、3つの観点から以下の7つのメトリクスを使用した（[2] 参照）。

## &lt; COBOL に対する品質メトリクス &gt;

- 全ステップ数
- 1段落の平均ステップ数
- 手続き部の最大ピリオド間ステップ数
- 手続き部 1ステップ当たりの条件判定数
- 条件文の最大ネスト数
- 手続き部 1ステップ当たりのプログラムコール数
- 代入かつ参照に使われているデータの比率

## 4 基準値の設定に関する考察

基準値の設定方法として、大きく分けて組織値とプロジェクト値の2種類を考えた。組織値とは、複数のプロジェクトを横並びに見るために部門などで共通の基準値として使うものであり、絶対的な値（数値）で与える。一方プロジェクト値は、システムを構成する複数のプログラムの分布を見るために、各プロジェクト毎（プロジェクトとは1（サブ）システムの開発を意味する）にそのプロジェクト内での相対的な値（平均値からの偏差）で与える。（図1参照）。

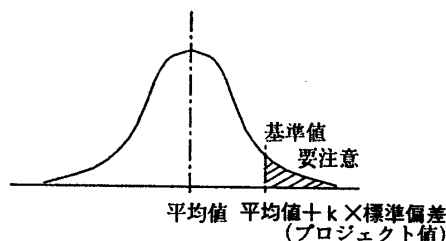
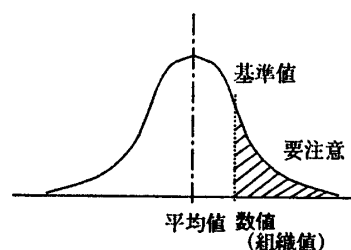


図1: 基準値の設定方法

A method to set criteria of program complexity for organization criteria.

Kaori Masuzawa, Atsushi Yamada  
Systems & Software engineering laboratory, Toshiba Corporation,  
70 Saiwai-ku Yanagicho, Kawasaki-shi, Kanagawa 210, Japan

Hiroshi Okano  
Tokyo system center, Toshiba Corporation,  
3-22, Katamachi, Fuchu-shi, Tokyo 183, Japan

1) 組織値の設定方法

これは過去のデータが揃わなくても次のやり方で初期設定する。

- a) 技術者の経験から設定した値
- b) 技術者が判断した過去の「良いシステム」の測定値 (平均値 + k × 標準偏差, k = 1.5)

2) プロジェクト値の設定方法

対象システムの測定値をそのまま用いれば良い (平均値 + k × 標準偏差, k = 1.5)。

組織値は組織内でどのプロジェクトが要注意であるかを見るのに適しているが、各プロジェクト内での要注意部分を調べる必要もある。そこでこれらを連動させることを考え、組織値とプロジェクト値を組合せて以下の5種類の基準値を設定し、総合試験と現地調整試験時の不具合との関係を調べた。

- A : 組織値 (技術者の経験から設定した値) を使用
- AC : 組織値 (技術者の経験から設定した値) と、プロジェクト値を併用
- B : 組織値 (技術者が判断した「良いシステム」から設定した値) を使用
- BC : 組織値 (技術者が判断した典型的に良いシステムから設定した値) と、プロジェクト値を併用
- C : プロジェクト値を使用

各基準値の設定方法で、要注意プログラムを抽出した時、要注意として抽出されたプログラムと抽出されなかったプログラムで、不具合の有無、不具合件数について、その平均値に差があるかどうかを検定した。使用したデータは3節で述べたシステムである。表1に、その比較結果を示す。

	基準値					全体
	A	AC	B	BC	C	
プログラム数						
抽出数	56	20	15	11	25	63
有不具合数	21	10	7	6	10	22
信頼水準 (%)						
不具合有無	91	86	94	90	74	—
不具合件数	99	87	91	88	76	—

表 1: 基準値オーバーと不具合

信頼水準  $\alpha$  % とは、要注意として抽出されたプログラム (どれかのメトリクスが基準値オーバーのプログラム) の集合と、抽出されなかったプログラムの集合で、各項目 (不具合有無、件数) の平均値に  $\alpha$  % の信頼度で違いがあることを意味する (信頼水準の値が大きいほど、抽出されたプログラムが有効であることを意味する)。従って、次の指針に従って基準値を決めることが望ましいといえる (図2参照)。

<基準値設定方法の指針>

- 1) 技術者が判断した過去の「良いシステム」の測定値を組織値とし、システムのプロジェクト値と併用する (BC)。
- 2) 過去の良いシステムがない場合は、技術者の経験から定めた組織値と、システムのプロジェクト値とを併用する (AC)。
- 3) 過去の良いシステム、技術者の経験のいずれもない場合は、計測しようとしているシステムのプロジェクト値のみを使用する (C)。

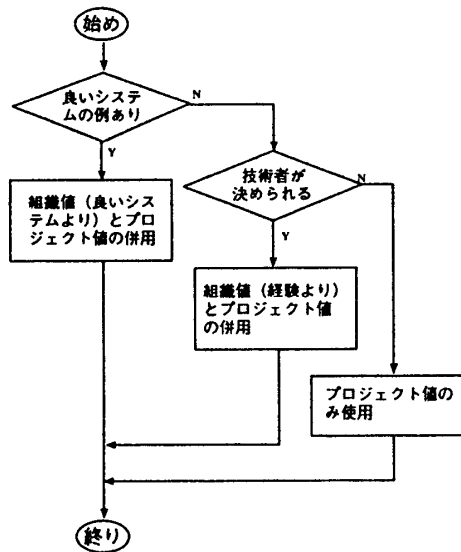


図 2: 基準値の設定指針

5 おわりに

メトリクスの基準値は、要注意プログラムを抽出する基準となる値であり、重点的探索プロセスにおいて重要な役割を果たしている。今回の分析により、基準値設定方法の指針を得られた。今後はこの結果を元に品質メトリクスを用いたソフトウェアの品質向上手法を複数のシステムに適用していき、抽出されたプログラムの中から、さらに効率良く早期に不具合を発見する方法について検討を加えていく予定である。

参考文献

- [1] 菅野文友 監修:「ソフトウェア品質管理事例集」, 日科技連出版社, 1990.
- [2] 増沢 香, 山田 淳:「事務処理プログラムにおけるデータに着目したメトリクスの提案」, 情報処理学会 第43回全国大会 講演論文集 P.5-311