

ソフトウェア品質メトリクスの適用

1 S-4

三宅 武司、 小田 英雄、 竹中 市郎
NTT ソフトウェア研究所

1. はじめに

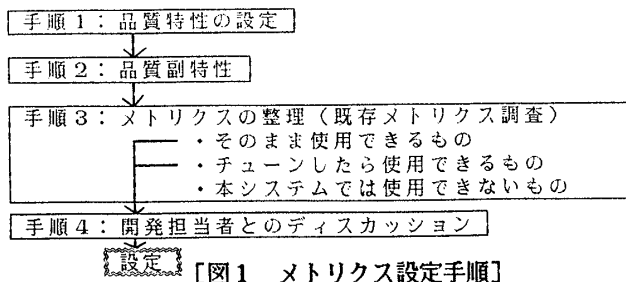
高品質ソフトウェアを開発するためには、設計の早期段階からの品質保証が重要であることはよく知られている。設計段階でソフトウェアの品質を保証するためには、まず、その時点での品質の把握、すなわち計測が不可欠であり、そのための尺度すなわちメトリクスと呼ばれる定量的な尺度を用いて品質の計測を行うことが必要である。本発表では、ある開発プロジェクトを対象に、新規開発フェーズおよび同ソフトウェアの機能拡張フェーズの2回に渡りメトリクスを用いて設計段階から品質保証を行った結果について報告する。

2. メトリクス適用の概要

2.1 メトリクスの設定

メトリクスを用いた品質保証の対象となったプロジェクトは、「OS検定システム」である。まず、メトリクスを設定するために、ユーザに対してOS検定システムに要求される品質特性に対するアンケートおよび、要求仕様書中の品質関連単語出現頻度分析を実施した。この結果、「機能性」「信頼性」「使用性」の各品質特性を重点特性として、品質保証を行うこととした。

次に、前記の3特性について具体的にメトリクスを設定する。設定手順を図1に示す。この表に記述されており、メトリクスを対象ソフトウェアにチューンさせるために、実際に開発に携わっている開発者とディスカッションを行った。この作業は、メトリクスの客観性と計測精度を高めるために必要であると考えられる。



2.2 メトリクスに基づいた品質の計測

品質の計測作業は本来ならば設計/開発担当とは独立した第三者が行うのが望ましい。今回の計測作業は客観的に計測が可能なように、開発担当2グループ間のクロスチェックを行った。つまり、設計書を対象とした計測の終了時に両者間で計測結果のレビューを行った。これは、メトリクスの定義に曖昧なものがあり、計測に計測者の主観が入ってしまい、計測結果にばらつきが出てしまう可能性があると考えたからである。

2.3 メトリクスによる品質保証の結果

メトリクスによる品質保証の結果を表1に示す。

(1) 品質向上効果: サンプル数が少なく確定的なことは言えないが、本報告では以下に示すような知見が得られた。

【直接的効果】

図2に示すとおり、メトリクスの適用により早い段階での品質のチェックポイントを設けたことにより、品質の向上を可能とした。なお、図中の品質1とは新規開発時の値、品質2は機能拡張時の値、標準値とはソフトウェア研究所の過去の類似ソフトウェアの平均値である。設計時の品質とは、設計書レビュー時に指摘された指摘項目である。また、製造時の品質には、コードインスペクションおよび単体試験で指摘された項目である。

新規開発時では機能設計時および、詳細設計時の品質は標準値に近いが、製造工程以降で上流工程におけるメトリクスによる品質保証の効果から、抽出バグ率の大幅な削減を可能としている。一方、機能拡張時では機能設計工程で標準値に比べて約5.6倍のバグを抽出している。これは、メトリクスによる計測を経験した開発担当者が、前回の経験によりメトリクスを基にしたレビューを実施したため、より詳細なレビューが可能となったためと考えられる。その結果、詳細設計工程ではバグの抽出率が標準値の約40%となっている。

【間接的効果】

品質の計測はプロジェクト内の一部を計測の対象としたが、他の計測対象以外の部分についても、過去のデータと比較して品質が向上している。これは、メトリクスによる計測に際し、事前にメトリクスを設計者に与えていたために、早い段階から品質に対する関心を高めたことと。特別な品質管理を行ったことによるホーソン効果が考えられる。

なお、本報告の品質計測でのバグ抽出時期は、設計工程（レビュー時）、製造工程（含む単体試験）および試験工程である。本来の品質とはユーザの手にソフトウェアが渡ってから評価されるべきものであるが、これについての評価は今後の課題となっている。

(2) 工数削減効果

図3に示すとおり、メトリクスを用いた品質保証により設計工程（機能/詳細設計）ではそれぞれ、新規開発時が8.7%、10.4%、改造時が2.0%、6.7%のそれぞれオーバーヘッドとなっているが、試験工程では工数の大幅な削減が見られる。さらに、機能拡張時ではメトリクスの設定等の準備期間を大幅に短縮できるので、オーバーヘッドが減少している。なお、図中の工数1とは新規開発時の工数、工数2とは機能拡張時の工数、平均値とはソフトウェア研究所の過去の類似ソフトウェアの平均値である。

[表1 品質保証実施結果]

対象工程	メトリクス数		工数比		計測工数		バグ発生比	
	新規	拡張	新規	拡張	新規	拡張	新規	拡張
機能設計	19	13	118%	123%	8.7%	2.0%	96%	560%
詳細設計	15	16	58%	40%	10.4%	6.7%	90%	40%
製造	-	-	49%	32%	-	-	76%	35%
試験	-	-	47%	50%	-	-	16%	65%

*メトリクスによる品質計測は、機能設計工程・詳細設計工程のみを実施した。

*工数比とは、実績工数と過去の類似ソフトのソフト研標準値の比である。

*バグ発生比とは、実績摘出個数と過去の類似ソフトのソフト研標準値との比である。

メトリクス・スコア

対象工程	新規開発時	改造時
機能設計	0.836	0.873
詳細設計	0.862	0.937

(3) その他

従来からのレビューが完全に行われていると仮定すれば、メトリクスのスコアは高得点を得るはずであるが、実際にはメトリクスを用いた定量的な品質計測によりいくつかの問題点を発見することができた。したがって、有効なメトリクスを設定することができれば、従来からのレビューと異なった観点からの品質の評価が可能であることを示唆しているものと考えられる。

3. メトリクスを導入する上での問題点

(1) 専門家グループの育成

メトリクスを用いた品質保証を実践するためには、メトリクスの設定と、これを用いて精度良く計測しなければならない。しかしながら、客観性の高いメトリクスはソフトウェアに依存しており、ソフトウェア個々に設定しなければならない。そして、計測作業も特定の基準を持って計測に臨まなくては客観的な測定は不可能となる。これらの作業は特化した作業であり、ノウハウの蓄積が重要な鍵を握っているため、メトリクスを用いた品質保証活動を専門とするグループの設置が望ましい。

(2) プロジェクト・リーダーの意識

メトリクスによる測定は一見オーバーヘッドに思える。しかしながら、正しく品質保証活動が実行されるならば品質と開発工数の面で明らかな効果が期待できる。したがって、プロジェクト・リーダーの説得が鍵となる。

(3) 開発者の意識

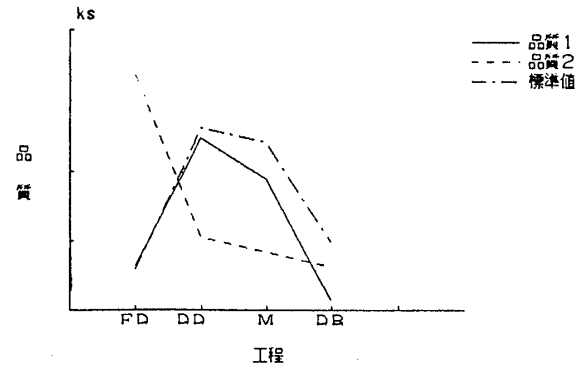
開発担当者の中にはえてして、ソフトウェアの品質を軽視する傾向がある。本事例のように設計工程から設計/開発者にメトリクスを提示することにより、開発の早い段階から品質への関心を持たせるなどの働きかけが必要となる。

4. 今後に向けて

以下に示す問題点に対して、今後検討を進める。

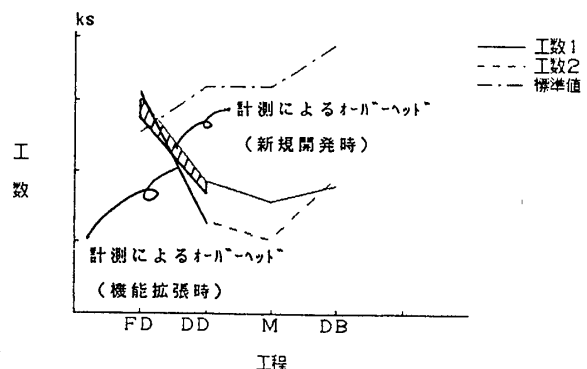
- ・本報告の対象となったソフトウェアのユーザ視点での品質評価を行い、本試行の最終整理を行う。
- ・メトリクスの値と最終製品の品質との関係の解明。
- ・より容易に計測可能でかつ、効果的なメトリクスへ向けてのメトリクスのブラッシュアップ。
- ・レビュー・チェックポイントとしてのメトリクスの使用。と共に、レビュー網羅率としての活用。
- ・メトリクス・データベースの構築および運用。
- ・メトリクスから設計ガイドラインの導出手順化。

品質/件



[図2 品質比較]

人時/ks



[図3 工数比較]

・メトリクス導出/計測作業のシステム化

5. おわりに

ソフトウェア品質向上のためには、メトリクスやこれを効果的に運用するための方法論の研究が重要であり、今後上記の問題について研究を進めていく。一方で、開発担当者およびユーザがより品質に対する意識を高めることが重要であると考えられる。その意味からも、今後、広く一般に対して高品質ソフトウェアに対する教育が重要になると思われる。

[参考文献]

1. Murine, G.E and others(1983):"Applying Software Quality Metrics",ASQC Quality Congress Transactions
2. Murine, G.E and others(1984):"Measuring Software Product Quality",Quality Progress. 17,[5],16-20
3. "ソフトウェア開発・システムの文書化標準化調査研究報告書":日本規格協会 1989