品質評価・管理のための手法とツールの実践的適用

Quality Evaluation and Management Tools for Software Development

小笠原　秀人　H. Ogasawara
田中　武志　T. Tanaka

ソフトウェアの大規模・複雑化に伴い，設計・作成工程とテスト工程で利用する品質評価と品質管理のための手法やツールに対する高度化の要求が増している。

当社は，これらの要求にこたえて品質評価ツール“ESQUT”(Evaluation of Software Quality for User's viewpoinT) と信頼性推定ツール“SQATP”(Software Quality Assurance Tool Package) を開発した。ESQUTは，設計・作成工程での成果物の品質評価ツールで，測定値の変化が大きい不安定なプログラムの検出，進捗(ちょく)把握，問題のプログラムの絞込みなどが容易となった。また，SQATPはテスト工程で発生する不具合情報に基づいてソフトウェアの信頼性を推定するツールで，プログラムの品質やテスト内容の評価，その他に有効である。

Due to the recent trend toward larger and more complex software, there is an increasing need among software developers for tools for software quality evaluation and management.

We have developed two new tools for software quality evaluation and management. The first of these, called ESQUT (evaluation of software quality from user's viewpoint), can measure the metrics of C and COBOL source code and allows detailed design documents to be expressed by tree-structured charts. For example, the number of lines of code (LOCs) is one of the C source code metrics. The second is called SQATP (software quality assurance tool package). SQATP can estimate software reliability based on detected failures during the testing phase.

This paper presents these tools and describes examples of their application to actual projects.

1 まえがき

ソフトウェアが大規模かつ複雑になり，品質評価と管理が重要になっている。複雑化するソフトウェアには従来のレビューによる品質作込みや，主観的な進捗報告に基づいた品質評価や管理だけでは対応できず，プログラムの品質評価や品質管理のための手法とツールの高度化が不可欠になっている。

ソフトウェアの品質評価・管理には，ソフトウェアの開発過程で行う方法と，ソフトウェアがある程度完成してからテストするという形で行う方法とがある。

ここでは，当社が開発した設計・作成工程で成果物の品質を評価する品質評価ツール“ESQUT”と，テスト工程で発生する不具合情報に基づきソフトウェアの信頼性を推定する信頼性推定ツール“SQATP”を紹介し，その適用例と効果を述べる。

2 ソフトウェアの開発過程での品質作込み

一般に，ソフトウェアの品質評価は最終のテスト工程に頼っている場合が多い。しかし，本来品質はプロセスで作り込まれるもので，品質を意識した設計・作成作業が必要である。

当社では，設計・作成工程で品質を作り込むために，ソフトウェア開発の成果物や作業の質を定量化する品質メトリクスを活用する方法をとっている。

2.1 ソフトウェアの品質作込み

設計・作成工程では，定期的に設計書やプログラムの品質メトリクスを計測し，複雑さや保守性の観点から成果物の品質を評価する。開発スケジュールに従って定期的に品質メトリクスを計測し，その値が基準となる品質メトリクス値（品質基準値）に達していない場合には，警告などを出す。一度作成されたプログラムの構造を変更するのは困難であるが，これによって①早い段階でプログラム構造に問題がないか，②構造を変更すべきかどうかなどが検討でき，タイミングよく品質作込みができる。

2.2 ソフトウェア品質評価ツール ESQUT

ESQUTは，TFF (Technical description Formula for Fifty steps/module design) 設計書とC，COBOLのソースコードの品質を定量的に計測するツールである。TFF設計書は，木構造チャートで記述された設計書であり，プログラムの処理手順やアルゴリズムを表したものである。TFF設計書

表1．ESQUTで計測する品質メトリクス
Metrics of ESQUT quality evaluation tool

TFF設計書	Cソースコード	COBOLソースコード
タスクレベル	ファイルレベル	大きさの観点
・タスク数 ・モジュール数	・モジュール数 ・総ステップ数 ・インクルードファイル数	・ステップ数 ・1段落のステップ数 ・1文のステップ数の最大値
モジュールレベル	モジュールレベル	制御構造の観点
・処理ボックス数 ・条件ボックス数 ・モジュール引数数 ・ループ数	・ステップ数 ・条件文数 ・ループ数 ・引数数 ・コメント文数	・条件判定のステップ数に対する比率 ・条件分岐のネストの深さの最大値 ・プログラムのコール数のステップ数に対する比率
		データの使われかたの観点
		・“代入”かつ“参照”に使われたデータの比率

とC，COBOLの品質メトリクスを**表1**に示す。

表1に示した品質メトリクスは，“処理の複雑さ”や“保守性”の観点から導き出されている。品質メトリクスの基準値は，開発するソフトウェアの種類（オペレーティングシステム(OS)，アプリケーションなど）によって異り，過去に開発した類似プログラムを定量化した結果と，不具合情報や開発経験に基づいて定める。品質基準値を満たしていないプログラムは，“処理が複雑”な作りになっており，不具合を含む可能性が高いことが適用結果から明らかになっている(1)。また，保守性の観点からも良い品質とはいえない。

図1にESQUTの計測結果の帳票出力例を示す。

プログラムが品質基準値を満たしていない場合，レビューによってモジュール分割の可能性やプログラム構造の変更の可能性が検討される。このような活動を実施するために，ESQUTは品質メトリクスの計測のほかに次の機能をもつ。

(1) 利用者が設定する計測スケジュールに従って自動的に品質メトリクスを計測する。

(2) 計測された結果から，総ステップ数の推移などの進捗情報を提供する。

ファイル名	モジュール名	ステップ数	条件文数	ループ数	引数数	コメント数
jfx	main	151	16	1	0	43
	fx_cmd	196	*34	10	1	61
	fx_load	93	*24	5	0	25
	fxloadr	85	18	3	3	24
	fxloadc	70	13	2	3	22
	fx_dump	135	*31	6	0	22
	fxdumpr	185	*29	6	5	48
	fxdumprv	103	17	5	4	24
	fxdumpc	183	*29	6	5	50
	fxdumpcv	84	13	4	4	24
	fx_list	*245	*31	4	0	32
	fx_dele	56	13	4	0	9
	fxdelemf	84	9	5	1	10
	fx_init	95	14	*12	0	10
	fx_help	28	0	0	0	0
	fx_sys	13	0	0	0	3
	fxfsrch	84	17	3	4	15
	fxhash	39	7	2	2	3
	fxdirget	108	14	7	2	27
	fxalcblk	39	5	3	1	7
	fxcnvstr	20	0	1	3	1

＊は，品質基準値を満たしていない計測結果。

図1．ESQUT計測結果出力例　モジュールごとに品質メトリクスを計測し，帳票形式で出力した例。
Example of ESQUT measurement results

(3) 時間を置いて計測された二つの計測結果の比較情報（新規，修正，削除モジュール）を提供する。

(4) 品質基準値を満たしていない計測結果だけを抽出する。

これらの機能によって，測定値の変化が大きい不安定なプログラムの検出，進捗把握，および問題を含む可能性の高いプログラムの絞込みが容易になる。

3 テスト工程での品質評価・管理

テスト工程では，品質の早期見極めと，テスト結果のフィードバックによるスケジュールの調整やテスト方法の改善が重要である。品質を早い段階で評価するためには，テストの実施状況から対象ソフトウェアの品質を推定することが必要であり，テスト結果の効率的なフィードバックには，不具合情報のデータベース化が有効である。

3.1 ソフトウェア信頼性推定ツール SQATP

テストの実施状況からソフトウェアの品質を評価するために，従来からソフトウェア信頼性推定と呼ばれる残存不具合数を予測する方法が利用されている。

SQATPは，テスト工程の途中で検出された不具合数から，残存不具合数を推定するツールである。このツールは，表計算ソフトウェアを利用して作成した不具合情報から，残存する不具合数を自動的に推定する。残存不具合数の推定はすでに提案されて，適用実績の多いソフトウェア信頼度成長モデルを用いている。このツールでは，指数モデル，遅延S字モデル，HGDM (Hyper-Geometric Distribution software reliability groth Model) モデルの3種類が選択できる(2)。

管理者は，SQATPで推定した残存不具合数と，テスト作業の内容および検出された不具合情報とを照らし合わせて，プログラムの品質やテストの内容を評価する。

3.2 不具合情報のデータベース化

不具合の分析結果は，テストの状況や品質を評価するための重要な情報である。当社でも，不具合情報のデータベース化を推進しており，開発作業の弱点分析や品質作込みの不十分な機能の絞込みなどに利用している。これらの分析結果は，開発技術の改善・成長につなげている。

4 適用結果

実際のプロジェクトへESQUTとSQATPを適用した事例と効果について述べる。

4.1 ESQUTの適用例と効果

大規模な通信制御装置のプログラム開発にESQUTを適用し，品質基準値の効果を確認した。過去のプログラム開発の実績データから，品質基準値を次のように設定した。

(1) ステップ数／モジュール　300

(2) 条件文数／モジュール　30

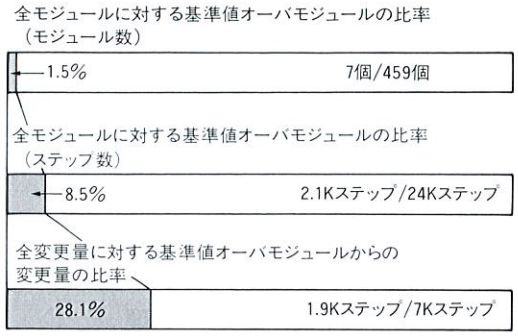

図2．品質基準値オーバモジュールからの変更量の比率　品質基準を満たしていないものは，品質基準値内のモジュールより変更されやすいことを示している。

Ratio of detected volume of variations from quality limitation value over module

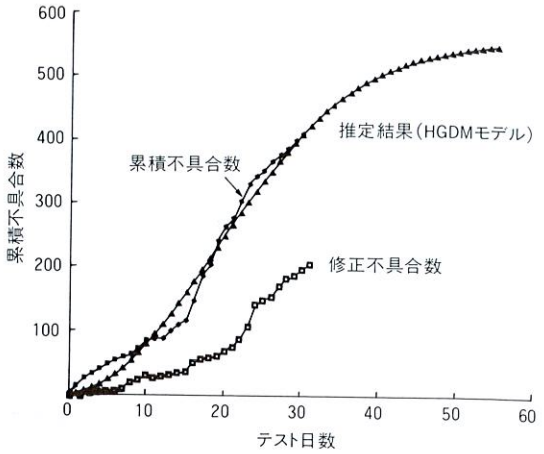

図3．残存不具合数の推定結果　テスト開始から30日経過した時点で，残存不具合数を推定した結果。

Estimation of nondetected failures

(3)　ループ数／モジュール　　10

あるサブシステムを対象に，単体テストから総合テスト終了までの間で，1週間単位で品質メトリクスを計測し，品質基準値を満たしていないモジュールの変更量を評価した結果を**図2**に示す。テスト工程での変更は，不具合あるいは機能変更などがあったことを意味する。

全モジュール459個のうち，品質基準値を満たしていないモジュールは7個であった。これら7個の総ステップ数は，全モジュールの総ステップ数の約8.5％に相当する。単体テスト以降に発生した総変更量（1週間単位の追加，修正，削除の行数の累積値）は約7Kステップであり，そのうちの約28％が基準値オーバモジュールからの変更であった。

これまでの多くの適用から，全体の開発量の10％程度を品質基準値オーバモジュールとして検出したとき，品質基準値オーバモジュールから検出される不具合数あるいは変更量は全体の約30～40％を占める，という結果が得られている。したがって，設計・作成工程で定期的に基準値オーバモジュールを抽出し，重点レビューをすることによって，早い段階から効率よく不具合を検出できるようになる。また，テスト工程では，基準値オーバモジュールに対するテスト項目密度を高くすることによって，プログラム構造を意識したテスト項目の設定ができる。

4.2　SQATPの適用例と効果

大規模な通信制御装置のプログラム開発にSQATPを適用し，残存不具合数の推定を行った例を**図3**に示す。

総合テストが約半分（30日）消化した時点で，残存不具合数を推定した。30日消化した時点では，約400件の不具合が検出されている。この時点では，テストが収束するまでに約30日程度かかり，総不具合数は約550件に達するとHGDMモデルで推定している。実際には，総合テスト終了段階で約500件の不具合が検出され，ほぼ60日間で総合テストが終了した。テスト中期以降での残存不具合数の推定精度が良いことは，今までの適用事例からも明らかになっている。テスト工程でSQATPを利用した結果，次の効果が明らかになった。

(1)　テストスケジュールの見直しや，機械・人員の割当てをするための情報として，有効に活用できる。

(2)　推定結果から，品質レベルを早期に評価できるため，品質目標値（不具合密度など）やテストケース・テストデータの見直しをするための判断材料となる。

5　あとがき

設計・作成工程で成果物の品質を評価するツール“ESQUT”と，テスト工程で発生する不具合情報に基づいてソフトウェアの信頼性を推定するツール“SQATP”について，その概要，適用例と効果について述べた。

ソフトウェアの品質をより確かなものとするための手法やツールを確立するため，ESQUT，SQATPを含めたツールと品質評価・管理の手法開発を今後さらに推進する。

文　献

(1)　小笠原秀人，他：メトリクスを利用したソフトウェア品質管理の提案と実践，情報処理学会サマーワークショップ・イン・立山，pp.27-34 (1995)

(2)　山田　茂：ソフトウェア信頼性評価技術，HBJ出版 (1989)

小笠原　秀人　Hideto Ogasawara

1990年入社。ソフトウェア品質評価・管理ツールの研究・開発に従事。現在，研究開発センター システム・ソフトウェア生産技術研究所。

Systems & Software Engineering Lab.

田中　武志　Takeshi Tanaka

1994年入社。ソフトウェア品質評価・管理ツールの研究・開発に従事。現在，研究開発センター システム・ソフトウェア生産技術研究所。

Systems & Software Engineering Lab.