プログラム測定によるテスト工程管理方法の提案

2N－6

古城路子 山田淳 小笠原秀人
(株) 東芝 研究開発センター

1 はじめに

ソフトウェア大規模化・複雑化，開発環境の分散化にともない，ソフトウェアの品質・信頼性の確保に対する要求が増大してきている．品質保証についての国際標準であるISO9000s認証でも，十分な工程管理や品質管理の計画と実施証明・定量的管理を行なえる品質システムが必要となってきている．本報告では，自動的に計測可能な「ソースプログラム構造の変化量」を利用した，テスト工程の定量的管理方法を提案し適用事例を示す．

2 現状の問題点

テスト工程の限られた時間内に全てをテストすることは困難である．そこでこの限られた時間内では，効率的なテスト（不具合検出率の高いテスト）が重要となり，そのための指標が必要である．

また，不具合報告書をもとにした信頼度成長を，より精度良くモニターするために，クロスチェック用の指標を増やす必要がある．

3 プログラム測定によるテスト工程管理方法

現状の問題を解決するため，自動的に計測することが可能な「ソースプログラム構造の変化量」を利用してテスト工程の管理をサポートする方法を提案する．「ソースプログラム構造の変化量」を「不具合修正」の代用尺度とみなしてモニタリングすることにより，テスト工程における効率的なテスト計画の立案と，信頼度成長（プログラム安定度）の把握などが可能となる．

プログラム測定に利用したツールは，「ソフトウェア品質評価支援ツールESQUT」（以下ESQUT）である．ESQUTは，「ステップ数」「条件文数」「ループ数」「引数数」「コメント数」といったメトリクスを計測する．ESQUTをテスト工程で，複数回にわたって適用することにより，「ソースプログラム構造の変化量」の時間的推移の情報を得ることができる．以下に運用ルーチンを示す．

1. Plan：テスト開始前に基準値・適用間隔の設定
2. Do：ESQUT自動計測・計測結果編集
3. Check：編集情報をもとに定期的にウォークスルー(以下WT)実施(問題分析・進捗状況把握)
4. Action：WT結果をもとに品質改善アクション

Quantitative management method for testing software with program structure measurement
Michiko Kojo,Atsushi Yamada,Hideto Ogasawara
Rsesearch and Development Center,TOSHIBA Corporation
70,Yanagi-cho,Saiwai-ku,Kawasaki,Kanagawa 210,Japan

4 適用事例

4.1 分析データ

この方法を実際のプロジェクトに適用し，評価を行なった．対象ソフトウェアは通信システムを構成するサブシステムで，対象規模は最終時点で3Kstep～256Kstep 中央値94Kstep(13機能) である．

ESQUTの定期的計測では，1回/1週ベースで計22回(5カ月間)の計測を行なった．また，テスト開始時点，テスト1/3終了時点(6週目)，テスト2/3終了時点(13週目)の計3回，プログラムの複雑度の計測も行なった．複雑度の基準値としては，ステップ数300，条件文数30，ループ数10と設定した．

4.2 変化量の推移

図1，図2，図3はステップ数，条件文数，ループ数のそれぞれの変化量の推移を示した図である．横軸は1週間毎，縦軸は変化量の累積値をプロットしている．いずれも初期45 %，中期45 %，後期10 %，程度の増加傾向があり，増加や安定が同時期に起こっており，推移の傾向が類似している．また，全体の増加率は，ステップ数29 %，条件文数36 %，ループ数33 %と，条件文の増加率がもっとも高い値を示している．

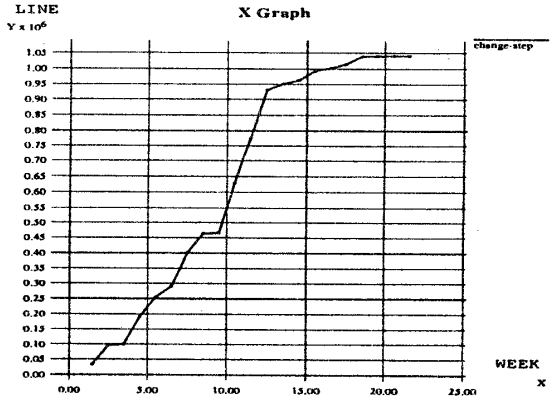

図1: プログラム構造の変化量の推移（ステップ数）

4.3 プログラム複雑度

全モジュール数のうち，テスト2/3終了時点（13週目）での基準値を超過した複雑なモジュール（以下基準値超過モジュール）の占める割合は各機能別で1.1 %～7.2 %（中央値2.9 %）であるのに対し，全変更量のうち基準値超過モジュールから検出された変更量の占める割合は図4のようであった．これは基準値超過モジュールに不具合が含まれる可能性が高いことを示している．特に条件文数，ループ数といったプログラム構造の複雑度が増すと，危険であることがわかる．

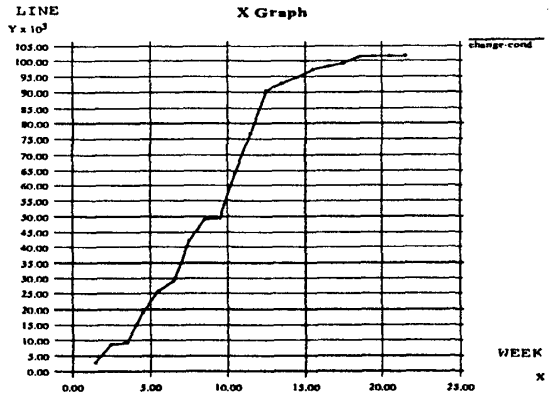

図 2: プログラム構造の変化量の推移（条件文数）

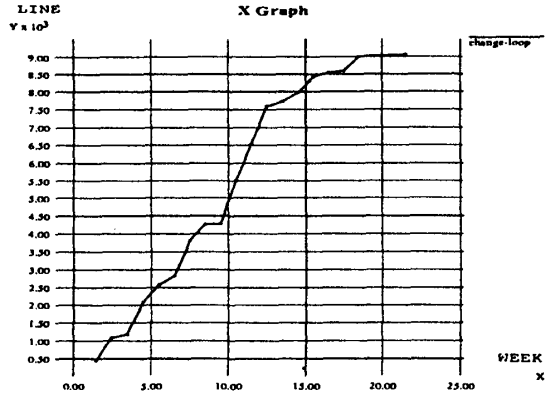

図 3: プログラム構造の変化量の推移（ループ数）

基準値超過モジュール一覧表などをもとに，定期的にWT を行なうことにより効率的なテストをすることが可能となる．

また，時間の経過とともにステップ数（規模）やモジュール数，また基準値超過モジュールは増加傾向がある．テスト開始時点での基準値超過モジュールからの変更の占める割合よりも，テスト 1/3 終了時点（6 週目），テスト 2/3 終了時点（13 週目）で計測した基準値超過モジュールからの変更の占める割合が高くなっていく傾向がある．したがって，テスト開始時点での基準値オーバーモジュールだけではなく，テスト期間内に数回計測する方がより効果的である．

4.4 信頼度成長

不具合曲線による信頼度成長分析の場合，検出された不具合の時間推移をデータとして用いるので，不具合報告が遅れて提出された場合は，集計作業・入力作業をやり直さなければならず，その結果不具合曲線は新たに引き直される．この事例では，図 1の 5〜10 週の時点ではプログラム構造が大きく変化している．しかしながら，10 週目時点までに集計された不具合報告数は，最終時点で集計された 10 週目時点のトータル不具合報告数の約 50 %に過ぎず，タイムラグがあった．つまり 10 週目時点で信頼度成長曲線を描いても適正

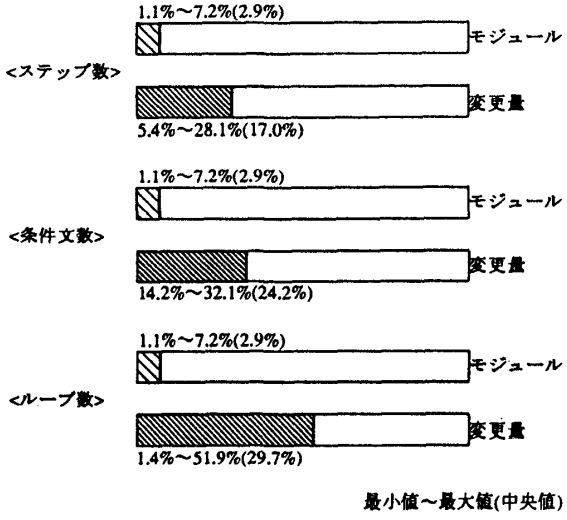

図 4: 基準値超過モジュールからの変更率

に状況を把握することは難しいということがわかる．

不具合報告とプログラム構造の変化量をクロスチェックすることにより，信頼度の成長状況の把握がより精度良く，リアルタイムに行なえる．また，不具合報告書の集計値が少ないにも関わらず変更が多く入っている場合には，不具合報告書提出のフォローができる．

5 まとめ

プログラム測定によるテスト工程管理の効果を以下にまとめる．

1. 測定は自動収集なので少ない負荷で計測が可能
2. 変化量が極端に多いモジュール・基準値超過モジュールの把握・追跡によるテスト対象の選定と重点レビューが可能
3. リアルタイムで定量的な信頼度推定とのクロスチェック・進捗管理・現状把握が可能
4. 「変化量の時間的変化の収束と安定化」＝「不具合の収束」とみなすことにより，リリース時期の判断材料の 1 つとなる
5. プログラム構造の変化をもとに不具合原因の種類の推定が可能

＜参考文献＞

1. 山田　茂：「ソフトウェアマネジメントモデル入門」共立出版,1993
2. 石井　康雄：「ソフトウェアの検査と品質保証」日科技連,1986
3. ロジャー・S・プレスマン：「ソフトウェア・エンジニアリング序説」TBS 出版会,1983
4. 宮本　勲：「ソフトウェア・エンジニアリング」TBS 出版会,1982
5. 菅野文友監修「ソフトウェア品質管理事例集」日科技連,1990