ソフトウェア開発プロジェクトの定量的評価方法

7E-2

橋本典子・森田昭憲・弓田龍二・石原義史・小林克己

富士通株式会社

1. はじめに

ソフトウェアの開発では，出荷検査の時点で問題を検出しても手遅れである．そのため，開発の早期段階から監視及び評価を行い，タイムリーな対処を行うことが重要である．

ソフトウェアの品質評価方法には，定性的評価と定量的評価があるが，今回は定量的評価の一手法について紹介する．

2. 前提となる環境

当ソフトウェア事業部では，開発担当部門が開発計画書によって作業計画を報告し，工程完了報告書によって完了状況を報告する制度がある．図1にその概要を示す．過去数年にわたるこれらの制度の運用を通して，開発作業に関するデータが蓄積されてきた．

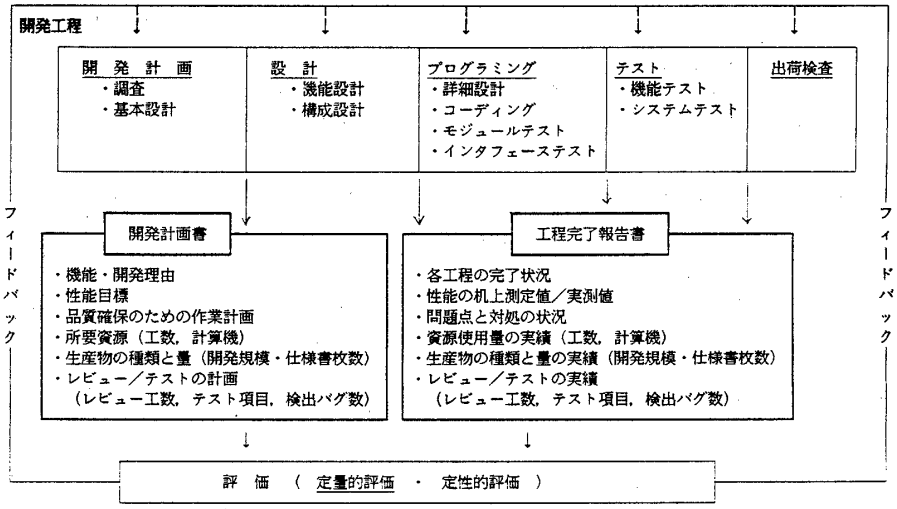

図1．開発計画書／工程完了報告書制度の概要

3. 定量的評価の考え方

開発計画書，工程完了報告書の記載内容の中の数値データには，開発規模（全体量，新規開発規模）・所要資源（工数，計算機）・テスト量（テスト項目数，テストセットの規模）・レビュー量（レビュー工数）・検出バグ数（各工程別）・各工程完了日・性能測定値などがある．これらのデータについて，品質に与える影響を考察し，定量的に評価できる尺度を設定した．表1にその概要を示す．また，これらについて過去の実績データを分析した結果，標準的な値（データの集中する範囲）が存在することが判明した．

同じようなソフトウェアを同じような開発方法で開発した場合，表1の各項目は通常の場合，一定の範囲に納まるものと考えられる．もし，一定の範囲からはずれるものは，何か特別な理由があるのが普通である．その特別な理由は，品質に対して良い方向に働く場合もあるが，悪い方向に働く場合が多い．したがって，一定の範囲からはずれたプロジェクトに対して，その理由を明確にした上で必要に応じて対処を行うことにより，不良品を防止することができると考えられる．

4. 評価方法

設定した基準値と報告される実績値（計画値），または計画値と実績値との離れ具合いによって評価を行う．評価は大きく分けて「悪い」「やや悪い」「良い」の3段階に分類し，基準値の両側6割の分布に入るものを「良い」，残り2割ずつを「やや悪い」「悪い」と評価する．（図2）品質への影響のしかたから判断し，片側のみ「悪い」「やや悪い」の評価を行う場合もある．

そして，本当に問題のあるプロジェクトは複数の評価項目の結果が悪くなると考えられ，組合せて評価を行うことが重要である．その方法として，品質への影響度に応じて各項目に重みづけをすることによって総合点を算出して総合評価を行うことにした．

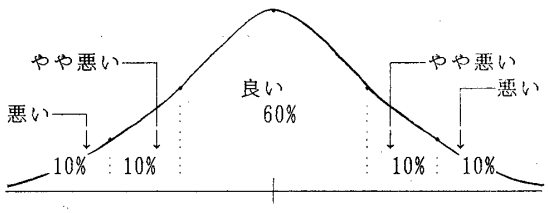

図2．個々の評価尺度に対する評価方法

Quantitative evaluation methods of projects in software development

Noriko Hashimoto, A.Morita, R.Yumita, Y.Ishihara, K.Kobayashi

FUJITSU LIMITED

5. 総合点の妥当性

以上のような仮定に基づく評価の妥当性を，テスト工程終了後の総合点と出荷検査時での評価との関係によって検証した．総合点が一定値以上のものを「良いもの」，一定値に満たないものを「悪いもの」と分類し，出荷検査で検出されたバグ数と比較した．その結果，総合点の良いものの多くは，出荷検査で検出されたバグ数が基準値以下であった．それに対して，総合点が悪いものの大半は，基準値を越えるバグ数が検出されていることが判明した．（図3）

同様のことを個々の評価項目ごとにも実施した一例を図4に示す．これによると，総合評価の場合と同様の傾向がみられるものの，あまり顕著な差は表れていない．この検証により，一つの項目に対する評価だけでは評価の精度は低いが，総合点で評価すると，より精度の高い評価が行えることが確認できた．

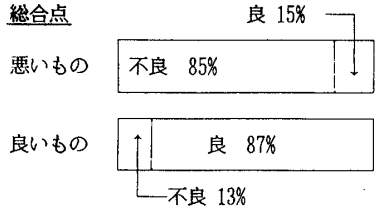

図3．総合点による評価結果と出荷検査での評価結果

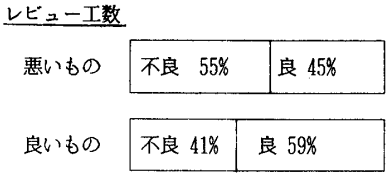

図4．単独評価項目による評価と出荷検査での評価結果

(注1)「不良」は基準値を越えるバグ数が検出されたもの．
(注2) 検証用のプロジェクトは，過去にまだこの評価方法が行われていず，途中段階で何の対策もとられていなかったものを対象とした．

6. 評価結果の活用

プロジェクトの評価は，計画・設計・プログラミング・テストの各工程完了後に提出される報告書のデータに基づいて行なわれる．そして，その工程完了時点での評価を行うことにより，次工程に進めるかどうかの判断材料にすることが可能となる．評価結果の悪い場合には，その原因の分析を行い，できるだけ早期に必要な処置を行うように開発担当部門に要求している．

なお，各項目の評価と総合点の算出をツール化し，グラフ出力することにより，その時点での品質把握を一目で行なうことを可能とした．

7. おわりに

今回の取組みにより，開発の途中段階で品質を評価する手法の基礎ができた．今後は，この評価の精度をより高めるためにさらに研究に取り組んでいきたい．

表1．定量的評価の考え方と評価尺度

評価項目	監視する狙い	品質への影響	評価尺度
検出バグ数・全検出数	バグ検出の十分性について確認する．	少なすぎる⇨レビュー不足，テスト不足 ⇨バグ残存⇨品質不良 多すぎる⇨設計，プログラミングの方法に問題⇨品質不良	プログラミング工程以降の全検出バグ数 / 基準値
・早期工程での検出比率	早い段階でのバグ検出により品質の早期確保に努める．	低すぎる⇨後工程にバグを持ち越す⇨後工程を圧迫 ⇨テスト不足⇨品質不良	プログラミング工程のみの検出バグ数 / プログラミング工程以降の全検出バグ数
レビュー工数	レビューの十分性について確認する．	設計レビューが少なすぎる⇨レビュー不足⇨機能漏れ，不完全な仕様⇨品質不良 ソースレビューが少なすぎる⇨レビュー不足 ⇨バグ残存⇨品質不良	レビュー工数 / 新規規模（K行）
テスト項目数	テストの十分性について確認する．	少なすぎる⇨テスト不足⇨バグ残存⇨品質不良	テスト項目数 / 全体規模（K行）
生産性	無理のない開発を行う	高すぎる⇨必要な作業の漏れ（レビュー不足，テスト不足） ⇨バグ残存，不完全な仕様⇨品質不良 低すぎる⇨コストが高い	新規規模（K行） / 工数
開発規模	計画値とのずれから異常を発見する．	予定より小さい⇨プログラムの機能漏れ ⇨ユーザ要求との不一致⇨品質不良 予定より大きい⇨後工程圧迫，資源不足 ⇨必要な作業の漏れ（テスト不足）⇨品質不良	新規規模実績（K行） / 新規規模予定（K行） 新規規模実績 － 新規規模予定
工程遅延	途中段階の遅れによる後工程の圧迫を防止する．	工程遅れ⇨後工程圧迫，資源不足 ⇨必要な作業の漏れ（テスト不足）⇨品質不良	各工程完了実績日 － 各工程完了予定日

【参考文献】苅田ほか：「基本ソフトウェアの信頼性と保全性」昭和57年度電子通信学会総合全国大会予稿