クリーンルーム手法の，バージョンアップ開発への適用

6R－5

森　茂子／佐藤　和夫

日本アイ・ビー・エム（株）大和研究所

1. はじめに

クリーンルーム手法は，高信頼性ソフトウェアの開発が可能な手法として，注目を集め[2)4)]，その適用事例も紹介されている[5)6)7)8)]．しかし，本手法はバージョンアップの場合の対処のしかたには明示的にはふれていない．また公表された事例は，バージョンアップ対象となる製品の設計を，クリーンルーム手法の設計で全て書き出す作業を前提とし，その作業に多大な費用と時間を要するものであった[9)]．

今回，日本の金融機関向ミドルウェアのバージョンアップに際し，クリーンルーム手法を適用するために，現実的な費用と時間で実施可能な設計方法を、新たに提案した．

・既に存在している，保守資料を有効利用しながら，修正，変更を行う．

・修正，変更項目ごとに，関係するデータおよびモジュールを洗いだす．それに対してのみ，クリーンルーム手法を適用する．

・テストは，従来通りの方法で行う．

実際に，従来ウォーターフォール型のプロセスで開発されてきた，上記ミドルウェアでこの提案を実施の結果，信頼性については約５．６倍の改善と予測される結果を得，効果を確認した．生産性は約８％向上し，開発期間は従来と同程度であった．これにより，現実的な費用と時間で実施可能で，しかも高信頼性を得られる，バージョンアップのための設計方法であることを確認した．

2. 保守文書を有効利用する，設計方法の提案

新しい開発方法論を取り入れて、高品質ソフトウェアを作成したい．しかし多くの場合には，それは従来営々として積み重ねてきた保守文書を破棄することを意味し，新開発方法論の採用は費用と時間の面で壁にぶつかる．一方，保守文書の有効利用をしようとすると，次に述べる３つの壁がある．

I. 文書類のマッピング・・・ウォーターフォール型のプロセスでは，外部仕様書，製品設計書，コンポーネント設計書，モジュール設計書，コードとなる[1)]．これとクリーンルーム手法の，機能詳細記述，データ分析/設計/実現，手続き分析/設計/実現とを，どう対応させれば良いか．

II. 開発方法論の変更・・・作成済みの保守文書を生かすということは，いいかえれば仕様を変更しようとすると，既に作成済みの設計，コードのことを念頭に入れながら作業をすることになる．下流が先に存在することは考えないで考案されている開発方法論は，そのことに対応出来るようにどう変更すれば良いか．

III. テストへの影響の考察・・・すでに存在する膨大なテストケースは，生かすべきかそうでないか．

この３つの壁を乗り越えるために，我々の工夫した設計方法を次に述べる．

2.1 文書類のマッピング

a. 保守文書類から，変更に関係するすべての入力，出力，データ，手続きを洗いだす．つまり，仕様書，設計書，コードから変更部分を明らかにしておく．

b. 変更部分についてのみ，クリーンルーム流の文書作成を行う．保守文書類との対応は取らない．

2.2 開発方法論の変更

a. クリーンルーム手法では，複数の機能の設計を別々に行う．ところが，既存設計の変更という形で設計を進めてゆくと，個々の詳細化の過程で矛盾が生じる場合がある．それを避けるために，複数機能の同時設計を行い，同期を取りつつ作業する．

b. 検証という，クリーンルーム独自のレビューを行う際に，新規作成部分で閉じたレビューのみならず，保守文書も利用してトップダウン，ボトムアップ両面からレビューを行う．

2.3 テストへの影響の考察

すべての変更個所を洗い出したつもりでも，抜けがある可能性が残る．ウォーターフォール型の総当たりのテスト方法のほうが，抜けが見つかる可能性が高いと判断し，従来通り

Application of Cleanroom Software Engineering to the Development of a New Software Version

Kazuo Satoh, Shigeko Mori

IBM Japan, LTD

1623-14, Shimotsuruma, Yamato, Kanagawa 242, Japan

のテストプロセスを実施する．これは，クリーンルーム流と比べると，コスト増の要因となる．

3. プロジェクトへの適用結果

金融機関向ミドルウェアAは，従来第X版までウォーターフォール型のプロセスで開発されてきた．第Y版の開発に当たり，今回提案した設計方法を適用して，バージョンアップ作業を行った．

3. 1 信頼性

テスト工程での，欠陥の収斂速度を第X版と第Y版を比較する形で，図1に示す．第Y版では収斂密度が大幅に向上している．また第X版では，別の観点からのテストを開始すると，すなわち新たな局面になると，新らたな欠陥が検出されている．第Y版では，新たな局面になっても，欠陥は検出されていない．これらは，経験則によれば，出荷後の信頼性が大幅に向上していることを示している．

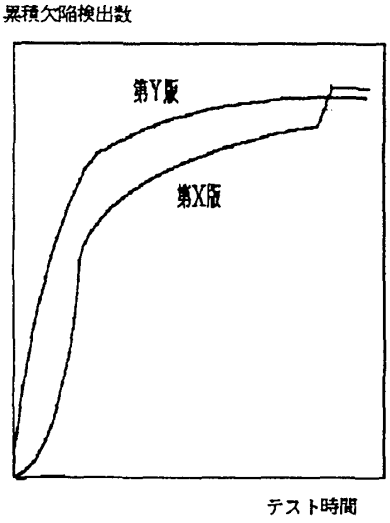

図1　テスト工程での、欠陥検出密度の推移

定量的には，ＩＢＭ社内で古くから用いられてきた，通称サンタテレサモデル[3]で出荷後の信頼性を予測してみると，第X版に比べて第Y版は，約5．6倍の改善がなされている．

3.2 生産性と開発期間

開発の生産性は8．4%改善した．開発期間については従来程度で収まった．より詳細に検討すると，設計工程では，作業量は48．3%から35．8%へとクリーンルーム手法を採用すると減少した．また，前行程では作業量は増加し、後工程が減少した．一方，製造およびテスト工程では，作業量は増加し，総合で8．4%の改善となった．

4. クリーンルーム手法の利点

定性的にみると，信頼性の向上をもたらす要因すなわちこの手法の利点は，次の事項だと考える．本論文で提唱した設計方法を用いることにより，バージョンアップ時にも，この利点を受けられる．

- 開発レビューによる知識の共有
- あいまいさの少ない設計記述による検証の精度の向上

5. おわりに

現実的な費用と時間で実施可能で，しかも高信頼性が得られる，バージョンアップのための設計方法を提唱し，その効果を実際のプロジェクトで確認した．今後の課題としては，(1)保守文書類から変更部分を，より的確に洗いだす方法論の開発および(2)大規模なバージョンアップ時の複数機能の同時設計の効率化−たとえば，有効なＣＡＳＥツールの導入が残っている．

参考文献:

1)R. A. Radice, R. W. Phillips : Software Engineering, Prentce-Hall, New Jersey(1988).

2) 佐藤和夫 : 無故障ソフトウェアを開発するための「クリーンルーム手法」紹介，情報処理，Vol. 35,No.9, pp.836～844(1994).

3) H. Remus, Z. Zilles : Prediction and Management of Program Quality, IBM Technical Report, TR-03.044(1978).

4) 佐藤和夫 : バグ追放と高生産性を両立する新開発手法「クリーンルーム」，日経コンピュータ，1995.3.20, pp.207～214(1995).

5) R. C. Linger, H.D. Mills : A Case Stydy in Cleanroom Software Engineering: The IBM COBOL Structuring Facility, Proceedings of 12th COMPSAC, pp.10 ～ 17, IEEE Computer Society Press, Los Alamitos, CA(1988).

6)L-G. Tann : OS32 and Cleanroom, Proceeding of 1st Annual European Industrial Symposium on Cleanroom Software Engineering(Copenhagen,Denmark), Section 5, pp.1 ～ 40, Q-Labs AB, IDEON Research Park, S-223 70 Lund, Sweden (1993).

7)Grant E.Head/日経エレクトロニクス訳 : クリーンルーム手法をソフト開発に適用し効果を確認，日経エレクトロニクス，1995.2.13, pp.121～137 (1995).

8)P. A. Hausler, R. C. Linger, C. J. Trammell : Adapting Cleanroom Software Engineering with a Phased Approach, IBM Systems Journal, Vol.33, No.1, pp.89～109(1994).

9)百名朝直，西橋幹俊，高橋和明 : ＸＳＣＯＰＥ２−クリーンルーム手法適用事例，第2回クリーンルーム研究会，大和 (1993).