

# 机上デバッグの徹底による、ソフトウェア品質の向上

4L-3

佐野木茂・近藤篤史

三菱電機関西コンピュータシステム（株）

エンジニアリングシステム部 公共情報システム第二課

## 1. はじめに

ソフトウェアを製作する過程において、その品質を向上させる手段としては、従来よりレビューや単体試験というものがある。しかし、実際のソフトウェア開発においては、種々の制約により、これらが満足には行われていない。そのため、組み合わせ試験・システム試験といった、開発の後期の段階で抽出される障害の中に、本来であれば単体試験段階で抽出されなければならないもの（コーディングミスなどの”うっかりミス”等）も多く含まれているのが現状である。

今回、某システム開発プロジェクトにおける品質向上施策の一環として、単体試験前のソースプログラムの机上での十分なチェックを徹底させることで、ソフトウェア製作の早期の段階での品質向上を図った。本報告では、その「机上デバッグ」施策について述べる。

## 2. 机上デバッグの運用

過去のシステム開発において、前に述べたような、試験前のプログラムの机上でのチェックを義務付けている例はなく、また、統一された机上でのチェックの基準やルールというものも無かったので、今回のシステム開発では、品質管理規定のなかに、「机上デバッグ運用方法」を盛り込み、メンバー全員への徹底を図った。

図2. 1に実施方法および運用ルートを示す。

対象は、製作する全てのモジュールとし、それぞれのモジュールのコンパイル完了時に机上デバッグを行い、不具合があった場合は、その内容を「机上デバッグ報告書」に記入するようにした。そして、不具合修正後、再度机上デバッグを実施し、このプロセスを不具合検出0件になるまで繰り返し行い、品質管理者へ報告するようにした。

「机上デバッグ報告書」は、当該モジュールに対してコードレビューを行う際に、レビューにも配布し、参考資料として活用を図った。

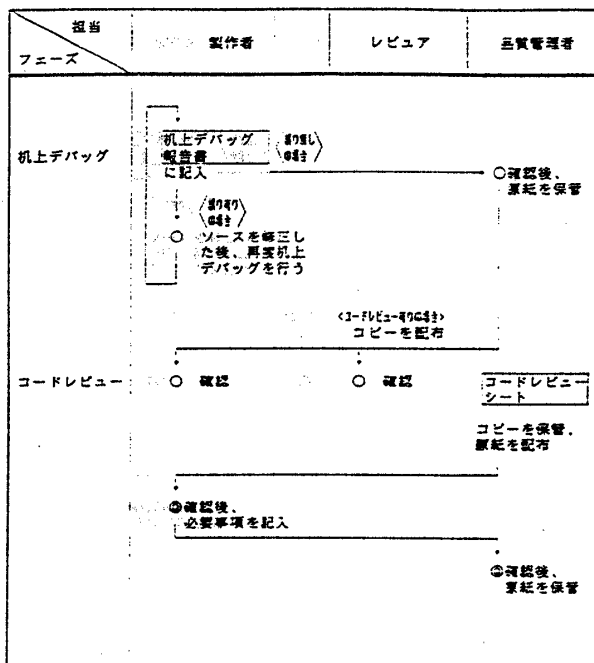


図2. 1 机上デバッグの実施方法および運用ルート

Improvement of quality of software by thoroughness in debugging on desk

Shigeru Sanoki, Atsushi Kondo

Mitsubishi Electric Computer Systems (Kansai) Corporation

### 3. 机上デバッグ結果の評価

今回のシステム開発で、机上デバッグにより検出した不具合の内容について分析した結果を、図3. 1に示す。

机上でのチェックにより最も多く検出されたものは「コーディングミス」。続いて、引数の不正、エラー処理抜けなど、いわゆる「うっかりミス」と言われるものが上位を占めた。また、データ宣言・参照エラーがこれに次いで多いが、これは今回開発言語であるFORTRAN に不慣れな者がメンバーに多かったことが影響していると思われる。

一方で、排他方法の不正などに代表される、本来組み合わせ試験以降で抽出されるような項目に関しては、やはり机上でのチェックでは検出しにくいことも分かった。

次に、過去の同様システムの開発で、組み合わせ試験以降に発生した障害のうち比率の高かった項目が、今回同フェーズでどのように変化したかを、表3. 1に示す。

過去のシステム開発で多かった項目が、今回は、いずれも比較的強く押えられていることがわかる。コーディングミスは、それでも20%近くを占めているが、これは前述したように、メンバーが若年層中心であり、自身でのチェックのみでは十分にミスを検出できなかったものと考えられる。

一方、S/W間のI/F不正や排他方法の不正など、実際にマシンを稼働させてはじめて明らかになってくる類の障害が、以前に比べて多く検出されるようになった。

### 4. まとめ

今回、この「机上デバッグ」を徹底させることで、従来のシステム開発に比べて大きく改善された点として、以下2点が挙げられる。

- (1) 製作の初期の段階での品質が向上したこと、特に、「うっかりミス」と言われる類の初歩的エラーを早い段階で検出できたこと。
- (2) 仕様もれ等、ソフトウェア設計上重要な手落ちを、早い段階でたたき出したことで、製作の手戻りを最小限に押えられたこと。

今後、今回の結果を踏まえて、机上でのチェック項目を見直し、マシンを使用した試験の項目の抽出の仕方を工夫して、ソフトウェア開発の初期の段階での品質向上を一層推し進めて行きたい。

最後に、今回のデータ収集に協力していただいた某システム開発プロジェクトのメンバーに感謝の意を表して、この報告を締めたいと思う。

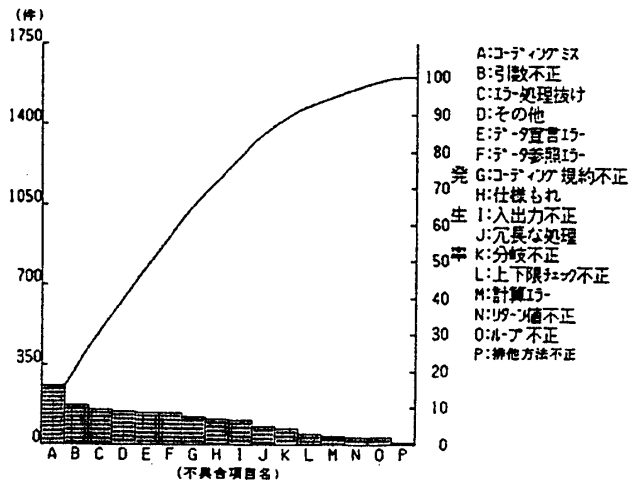


図3. 1 机上デバッグ不具合項目 パレート図

障害原因	過去のプロジェクト	今回のプロジェクト
コーディングミス	32.9%	19.4%
仕様書の不備	22.7%	14.1%
前工程仕様の理解不足	20.2%	4.0%
:	:	:
S/W間のI/F不正	4.9%	9.4%
データベース不正	2.1%	11.4%
排他制御不正	2.0%	5.3%

表3. 1 組み合わせ試験以降の障害内容