

ソフトウェア保守に於ける生産性向上

3J-4

村田季吉

あさひ銀総合システム(株)

1. はじめに

コンピュータ技術の発展に伴い、その利用技術や適用範囲も急速に拡大してきた。システムは年々、大規模・複雑化しておりソフトウェアの需要が増え続く限り、それらの保守の必要度もましてくる。情報システム部門にとり最も手間のかかる業務が、既存システムの保守である。

ソフトウェアの開発については、ワークステーション機能の発達により、ホストの負荷軽減や開発の効率化を図るために分散開発手法が考案されておりこれに基づいた様々な開発支援ツールが提供されてきている。しかし、保守作業については有効な支援ツールは整備されておらず、現状では保守担当者への負担は増えるばかりである。このような環境の中ですでに使用されている開発・保守支援ツールが保守に対しても有効利用できるならば、現状抱えている保守の問題を解決する糸口となるであろう。

2. 現状分析

現状での保守に於ける問題点を以下に述べる。

- ・影響範囲の調査に時間がかかる。
- ・既存ドキュメントが保守に適していない。
- ・個人の知識、技術、経験に依存し要員が固定化する傾向があり、ノウハウが資産として継承されない。
- ・保守時にドキュメントが更新されていない。
- ・テスト項目の洗い出し、テストデータの作成に厳密な基準がない。
- ・テストが不完全で、新たなバグが発生している。
- ・プログラムの標準化や構造化がされていない。

3. 研究テーマの設定

現在すでに利用している開発・保守支援ツールを保守で活用するための改善点や不足機能などについて以下の4つの項目を研究テーマとした。

(1) 現行の開発・保守支援ツールの改善点

開発・保守支援ツールを用いて作られたプログラムを保守する場合に問題となる項目を明らかにするとともに、その改善策を提唱し保守支援ツールとしてさらに有効となるよう検討する。

(2) 保守用ドキュメント

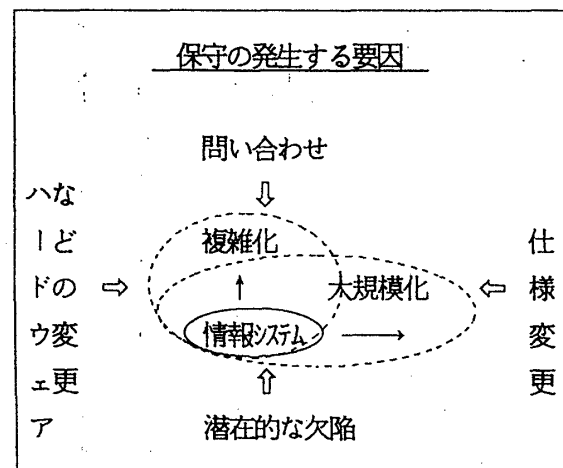
開発・保守支援ツールの特長であるドキュメント性に着目し、保守で使用するドキュメントや要望の面から検討し、より保守をし易くするドキュメントとその出力方法について検討する。

(3) テスト支援機能

テスト項目やルート、テストデータの生成と検証を自動で行い人手に頼る部分を極力排除する事により信頼性を高める支援方法について検討する。

(4) 導入・移行

開発・保守支援ツール導入後の保守作業効率の向上という点に着目し、導入・移行作業のためのポイントや留意点について検討する。



Productivity Improvement of Software Maintenance

Kiyoshi Murata

Asahi Bank Systems Co., Ltd.

4. 現行の開発・保守支援ツールの改善点

(1) 操作性の向上

- ・エディタの編集機能の改善
- ・ホスト側エディタの操作性と機能の改善

(緊急対応が求められる保守においては、分散による対応とは限っていない。)

(2) 理解性の向上

- ・日本語による表記をより業務表現化して、抽象度を高め、仕様書の解析を容易にする。
- ・記述位置を自動的に整形することにより、仕様書の見やすさが向上し、コーディング規約の標準化を図る。

(3) 保守性の向上

- ・記述規約のチェック機能を追加し、規約外の記述箇所を容易に検出させる。
- ・ワークステーション機種間の機能を統一させる
- ・抽象化構文や日本語名標の自動変更や自動生成

5. 提案ドキュメント

(1) エラーメッセージ説明書

トラブル時の早期対応を図ることができる。

(2) プログラム変更履歴書

修正内容の確認や過去の変更内容を把握できる

(3) 判定項目説明書

プログラム機能を容易に解析することができる

(4) デシジョン・ブルリスト

仕様書修正に伴う設計書を自動で出力できる。

(5) クロスリファレンスリスト

各仕様書の関連や影響調査の効率化を図れる。

(6) セクション構造図変更履歴書

プログラム構造の変更履歴が把握できる。

(7) 入出力項目関連図

データ項目の関連や使用方法が把握できる。

6. テスト支援機能

(1) テストケースの設定

既存のテスト支援ツールを分析し、さらに効果的なツールとすべく検討した。

- ・ステップ単位の設定機能により全ケースを網羅

- ・見やすいテスト仕様書によりテスト漏れを防止
- ・テスト仕様書を画面より一括編集し効率を図る
- ・検証時に活用できるテストケースデータの連携

(2) テストの実施

- ・テストデータの自動作成は既存ツールで効率化
- ・テストの実行では、有効な支援は皆無である。

(3) テスト結果の検証

- ・データ項目の内容変化を追跡しデバッグを支援
- ・テストケースと結果の連携による網羅率の把握
- ・見やすい検証リストにより検証作業を支援する

7. 導入・移行

(1) 既存資産の移行

移行手順に従い、工程毎の事前検討事項を調査

(2) 作業環境の整備

作業環境で必須となるWSや関連ソフトを検討

(3) 作業標準の決定

導入後の各種規約や作業標準を作成する。

(4) 導入・移行および管理体制の整備

導入・移行のために必要な要員や体制の整備

(5) 保守運用管理

保守及び運用管理方法やセキュリティを検討

8. 評価

今回の研究成果に基づいて、保守作業比率から工数比を推計して見ると、現行の開発・保守支援ツールでもCOBOLと比較した場合、保守に於ける作業効率の向上が認められる。さらに今回、提案した改善策が実現されることにより、より一層の効果が期待できる。ただし、あらかじめ留意点など考慮しておく必要がある。

9. おわりに

保守支援ツールにより保守作業は変革し、その生産性は従来の方法より向上するが、本質的な問題を解決する訳ではない。保守とは地道な作業であるが情報社会にとって非常に重要な作業である。その意味でももっと広く社会に保守の重要性を認識していただく必要がある。