

保守段階でも適用可能なソフトウェア障害処理票管理システム*

5C-9

櫛山 淳雄† 埴 未帆† 大木 淳‡

† 日本電気株式会社

‡ (株)NEC 情報システムズ

1 はじめに

ソフトウェア保守は複雑なプロセスである。ユーザに出荷されたソフトウェア製品は、ユーザからの障害報告や改善要求に基づき、修正とリリースが繰り返される。特に、複数のユーザを有するソフトウェア製品の場合、ユーザからは非同期にさまざまな障害や改善要求が報告され、ある障害や改善要求がどこに処置されるべきかあるいは、どこに処置されたかを把握しなければならない。本論文では、保守段階における障害管理の方式と、それに基づく障害処理票管理システムについて述べるとともに、システムの適用結果から本提案方式並びにシステムの有効性について論じる。

2 関連研究

ソフトウェア開発における障害管理については、いくつかの研究事例や製品がある。障害管理はプロセスに基づいたソフトウェア開発環境が実際に支援する対象として [2]、あるいは、CSCW システムのソフトウェア開発プロセスへの適用事例 [5] として取り上げられている。これらの研究で取り上げられる障害管理プロセスは比較的単純なものであり、本論文で述べるような実際の保守における複雑な障害管理プロセスを扱っていない。一方、障害管理を支援する製品もいくつか提供されている (例えば、[1], [6])。これらのツールに共通する機能として、障害票を記述する GUI、報告された障害情報のデータ管理、レポート作成機能等がある。そして、これらの製品にはある範囲のカスタマイズ機能が用意されており、規定された障害管理プロセス (状態遷移モデルで表現) のもとでデータ項目の追加ができるもの [6] や障害管理プロセス自身のカスタマイズを行えるもの [1] がある。障害管理プロセス自身をカスタマイズ可能な製品は、複雑なプロセスを記述することができるが、それは非常に労力のかかる作業である。また、これらのツールは、複数の派生版を有する製品を対象にした保守を想定していない。

3 保守段階での障害管理プロセスへの要件

本論文では、パッケージ製品として販売され、かつ、そのパッケージ製品をベースに、個別ユーザからの障害報告や機能改善要求に対応した派生版を提供しているソフトウェアプロジェクトの保守プロセスを対象とする。

この種のプロジェクトでは、ある特定ユーザからの障害報告や改善要求はそのユーザ向けに作成される製品版で処置されるとともに、それらの内容を調査し、他の特定ユーザへ修正版を提供したり、次期パッケージのリリース時に処置したりしている。従って、製品の版管理は複雑である。図 1 に製品版の派生関係の一例を示す。

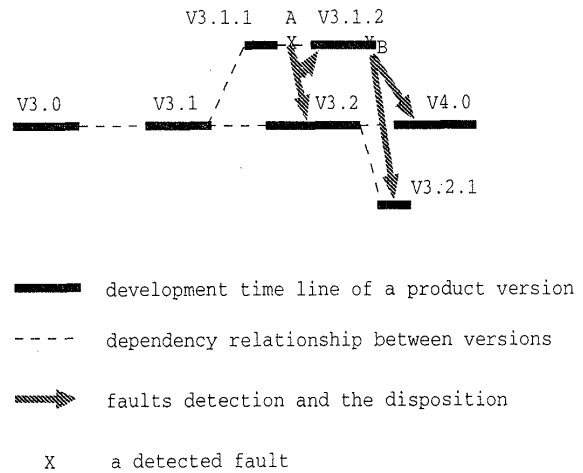


図 1: 製品版の派生関係

この時、製品のユーザへの提供時期が同一であるとは限らないため、開発者が行ったある障害のある版への処置が他の版で処置漏れが生じないようにしなければならない。例えば図 1 では、V3.1 から V3.1.1 開発と V3.2 開発が枝分かれしていることを示しているものとする。ここで、V3.1.1 に対して、その製品出荷後に検出された障害 A はそれが V3.1 以前に作り込まれたものであった場合¹、V3.1.1 に後続する V3.1.2 と、V3.1.1 と並行して開発された V3.2 に対して処置される必要がある。

* A problem report management system for software maintenance, Atsuo Hazeyama †, Miho Hanawa †, Atsushi Ohki ‡, † NEC Corporation, ‡ NEC Information Systems Ltd.

¹ 障害の発生した機能が開発された製品版の情報は別途保持している。

以上のことから、各障害報告に対して、それが他のどの版で発生しているのかを把握し、各版での処置状況をトレースできる仕掛けが必要となる。新規開発の場合の障害管理では、製品全体の構成木は枝分かれしていないので、このような複雑なトレースを行う必要はなかった。

トレース結果は一覧形式で公開するが、この時に、プロジェクト管理上は、各製品版ごとに処置されるべき障害報告の一覧が必要である(各製品版で処置されなければならない障害の処置漏れを防ぐため)。一方、開発者の側からは、ある障害報告がどの版で処置済みでどの版で未処置かが一目でわかるように、障害処理票ごとの処置状況を示す一覧が求められる。

4 保守段階でも適用可能な障害処理票管理システム

本章では、前章で述べた保守段階における障害管理の要件を考慮した障害処理票管理システムについて述べる。本システムは、著者らがこれまでに開発したソフトウェア開発プロセスにおける障害処理票管理システム [3], [4] を機能強化したものである。紙面の都合上、本稿では、複数の製品版での障害処置状況を管理する機能についてのみ述べる。

新規開発のソフトウェアの障害管理システムは、現在開発中のものに対する障害処置状況を追跡するだけで良かった [3], [4]。しかし、保守フェーズにおいては1件1件の障害報告が、複数の製品版に対して処置しなければならない場合があるため、処置状況に関する情報を、処置するべきすべての製品版に対して保持しなければならない。そこで、各障害報告では、障害の発生した製品版並びにその障害を処置するべきすべての製品版に対して、製品版の情報、処置状況を示す情報、処置が行われた日付を3つ組みにして管理する。例えば、ある障害がV2.1で検出されV2.1では1998/10/1に処置されたが、それをV1.4でも処置しなければならず、V1.4ではまだ未処置の場合には、以下のように記述する。

Status: V2.1, Modified, 1 Oct. 1998; V1.4, OPEN;

このように製品の版情報とその処置状況を時系列で管理することにより、例えばある製品版で制限事項化されていた障害が後続のどの版で処置されたのかを追跡することもできる。

管理レポートとして、各製品版ごとに処置されるべき障害報告の一覧と、ある障害報告(ある障害処理票)が、どの版で処置済みでどの版で未処置かを示す一覧を提供している。

5 システムの適用評価

本論文で述べた障害管理方式を実現した障害処理票管理システムを、製品の機能強化を進めている実ソフトウェアプロジェクトに適用した。その結果本システムの適用により、ある製品版では、他の製品版からの障害報

告により処置するべき障害が数十件も潜在していることが明らかになった。このことは本論文で述べた方式に基づく障害管理を行うまでは把握されていなかったことである。このようにして作成された情報に基づいて、プロジェクトリーダー、開発担当者、品質保証技術者が1件1件の障害について、当該の製品版において処置の必要性の有無を検討する場が持たれ、顧客からの障害報告を受ける前に未然に障害に対処した版をリリースすることができた。

また、複数の製品版に対して処置するべき障害処理票の分布を調べた結果、主要な機能強化の開発ではその割合は小さいが、その後の小さな機能強化やバグを処置した版では、複数の製品版に対して処置するべき障害処理票の割合が相対的に大きくなる傾向にあることがわかった。

6 おわりに

本論文では、複数の製品版が並行して開発されているソフトウェアプロジェクトの保守段階における障害管理の方式と、それを実現した障害処理票管理システムについて述べた。そして、実プロジェクトへの適用結果から本方式並びにシステムの有効性について論じた。

本論文で述べた障害管理方式とそれに基づく支援システムは、開発プロセスで適用できることはもちろん、バージョン木が単一である保守プロセスにも適用可能である。

参考文献

- [1] ClearDDT, <http://www.rational.com/products/ccmbu/clearddts>.
- [2] Gruhn V., Urbainczyk J., Software Process Modelling and Enactment: An Experience Report related to Problem Tracking in an Industrial Project, Proceedings of the 20th International Conference on Software Engineering, pp. 13 - 21, IEEE Computer Society Press, Kyoto, Japan, April 1998.
- [3] 樋山, 東出, 埴, 山名, 山崎, 電子メールを用いたソフトウェアプロジェクト管理の実践, 情報処理学会第51回全国大会 3M-5, pp.5-101 ~ 5-102, 1995年9月.
- [4] 樋山, 大木, 埴, WWWと電子メールを活用したソフトウェア障害処理票管理システム, 情報処理学会第56回全国大会 2C-6, pp.1-221 ~ 1-222, 1998年3月.
- [5] Kaplan S. M., Tolone W. J., Carroll A. M., Bogia D. P., Bignoli C., Supporting Collaborative Software Development with ConversationBuilder, Proceedings of the 5th Symposium on Software Development Environments, ACM SIGSOFT Software Engineering Notes pp. 11 - 20, Virginia, USA, Dec. 1992.
- [6] PR-Tracker, <http://www.halcyon.com/software/prtracker.html>.