

3R-4

オブジェクト指向型のプロダクト管理モデルについて\*

内藤 岳, 石井 一彦, 加藤 保夫†  
NTT ソフトウェア研究所‡

1 はじめに

ソフトウェアの開発プロジェクト内のデータを厳密に管理することが品質の良い生産物を生産する時の本質である。当然、生産物を管理することはソフトウェアの開発での重要な要因であるが 開発プロジェクトでのデータの流れを適切にコントロールする外部環境が生産性と品質の決定要因となる。[1]

つまり、生産物の構成が決定されると開発中その構成の変化が少ないものを管理するバージョン管理だけでなく、そのバージョンの変更を起こす元になるデータを管理しバージョン管理を適切にコントロールする外部環境が必要となる。

そこで、本論文ではソフトウェアのライフサイクルの中の下流工程においてソースコードや実行ファイルのバージョン管理だけでなく、試験工程の発生するバグをも管理の対象とし、このバグを管理することにより試験を適切にコントロールするプロダクト管理モデルを明確にし、オブジェクト指向の考えを用いて、本モデルの管理対象のカプセル化を提案する。

2 バグの発生解決プロセスとプロダクト管理モデル

2.1 バグの発生解決プロセス

バグの発生から解決までには、以下のようなプロセスを含んでいる。

- バグ発生：障害処理票を起票することにより認識される。
- 判定：バグの発生しているモジュールを切り分ける。
- 解析：バグの発生している箇所および原因をつき止める。
- 検討：どのようにバグを訂正するかを考える。
- 修正：実際にソースなどを修正する。
- 確認：バグが取り除かれているか、他に悪い影響を与えていないかをかくにんする。

通常は、このバグの発生から解決までを障害処理票の形で管理している。複数組織での大規模開発における、このプロセスをバグの発生解決プロセスと管理対象の関係に着目してモデル化したのが図1である。

2.2 管理オブジェクト

今までのバージョン管理で管理されていたソースなどの他に障害処理票や開放通知書などの管理情報もオブジェクトとして抽出した。

- ソース：各言語で書かれたテキスト
- オブジェクト：ソースをコンパイルして得られる中間ファイル
- モジュール：オブジェクトの集合体
- 実行形式：実行可能な形式のプログラム
- 障害処理票：バグを管理するためのもの
- 解析状況票：解析の状況を記入したもの
- 試験手順書：試験の手順およびチェックリスト
- 開放通知書：実行形式を提供するときの情報を記入したもの

抽出したオブジェクトのメッセージと動作を表1に示す。

2.3 オブジェクト間の関係

抽出されたオブジェクトには、ソース、実行形式などの従来から管理されていたオブジェクトが持っていた共通の属性を持つ。さらに、今回新しく抽出したオブジェクトには他のオブジェクトの関連を示す属性がある。これを表2に示す。

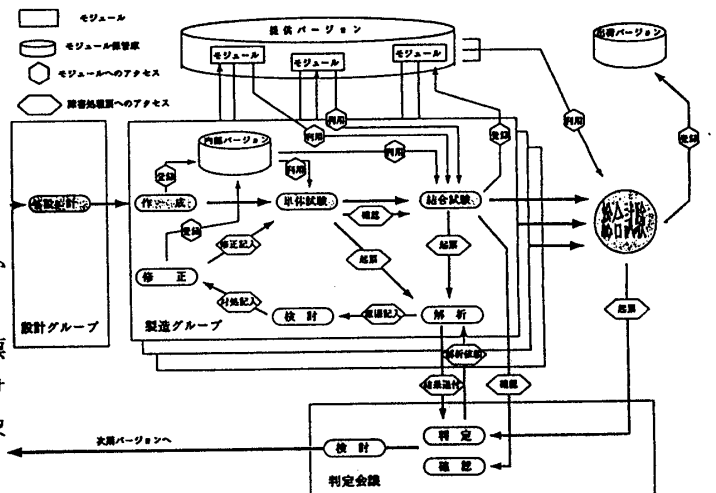


図1 バグの発生解決プロセスとプロダクト管理モデル

\*A Management Model of Software with Object-Oriented Method  
†Takashi Naito, Kazuhiko ISHI, Yasuo KATO  
‡NTT Software Engineering Laboratory

3 考察

今までのソフトウェアのライフサイクルモデルであるウォーターフォールモデルでは、各工程の終了時に次工程へ生産物が確定していることを前提としている。それにしたいが、ツールも各工程毎に作成され工程終了時に必要とする生産物を生成するようになってきている。特に、下流工程である製造、試験、運用保守の工程では、それぞれの工程用のツールとしてコンパイラ、デバッガ、SCCS[2] や rcs[3] のような構成管理ツールなどが用意されている。

しかしながら実用上、製造、試験、運用保守の各工程においては工程の区分は非常に曖昧である。例えば、試験中にバグが出た場合にはバグの箇所を発見するためにデバッガを使用し、ソース修正のためにコンパイラを使用し、ソース管理のための構成管理ツールを使用する。

工程が曖昧だったのは、図からわかるように、バグの存在がソフトウェアのライフサイクルの中で明確に定義されていなかったためである。つまり、いままでバグはソフトウェアのライフサイクル上の正規の生産物ではなく、存在しない方がよいものとみなされてきたからである。

そこで、このバグをソフトウェアの開発上重要な要素としてとらえ、これを明確に管理することによりソフトウェアの製品管理が容易になり、試験工程での遅れ、ネックなどを的確に把握することができる。

さらに、今回新たに抽出した障害処理票などの管理対象は、バグ発生解決プロセスの中で多数のサブプロセスからアクセスを受けている。これらの、管理対象をオブジェクト指向の考え方により整理したことによりサブプロセスからのアクセスを定められた適切なインタフェースを通して行なうことが可能である。

このことにより、プロセス全体の中での管理対象の整合性が一貫して維持できるようになり、プロダクト管理が容易に行なうことができる。また、下流工程における支援環境の一部として、プロダクト管理システムの構築のための工数、時間を削減することができる。

4 さいごに

本論文では、ソフトウェアのライフサイクルに下流工程における管理対象を今までのソースなどの工程生産物だけでなく、バグなどの工程内だけに存在するものまで広げた。このことにより、下流工程での作業のトリガーが明らかにたつた。

また、管理対象をオブジェクト指向の考えによりとらえ、各オブジェクトの共通点を明らかにし、また各オブジェクトの関連を明らかにした。

今後は、これらの結果を元にし、下流工程における支援環境に反映していきたい。

[参考文献]

- [1] M. W. Evans and G. W. Furr, "Automating The Configuration Management Process", COMSAC'86, 1986
- [2] W. F. Tichy, "RCS - A System for Version Control", Software Practice and Experience, 1985
- [3] AT&T, "Source Code Control System", UNIX System V programmer's Guide"

表1 管理オブジェクトの一覧

オブジェクト	メッセージ	動作
実行形式	export modify display	モジュールを link 実行形式ファイル进行操作する (デバッガ) link 情報を見る
モジュール	export modify display	オブジェクトファイルを archive モジュール进行操作する (ダンプ&エディット) archive 情報を見る
オブジェクト	export modify display	ソースファイルを compile オブジェクト进行操作する (ダンプ&エディット) compile 情報を見る
ソース	export update modify display import	自身の任意のバージョンを取り出す。 export したソースファイルを最新の状態にする。 ソースを修正し、登録する。 ソースを見る ソースファイルを登録する。
障害処理票	export update modify display import	自身の任意のバージョンを取り出す。 export した障害処理票を最新の状態にする。 障害処理票を修正し、登録する (ツール起動)。 障害処理票を見る (ツール起動) 障害処理票を登録する。
試験手順書	export update modify display import	自身の任意のバージョンを取り出す。 export した試験手順書を最新の状態にする。 試験手順書を修正し、登録する (ツール起動) 試験手順書を見る (ツール起動) 試験手順書を登録する。
解析状況票	export update modify display import	自身の任意のバージョンを取り出す。 export した解析状況票を最新の状態にする。 解析状況票を修正し、登録する。 解析状況票を見る (ツール起動) 解析状況票を登録する (ツール起動)
開放通知書	export update modify display import	自身の任意のバージョンを取り出す。 export した開放通知書を最新の状態にする。 開放通知書を修正し、登録する (ツール起動)。 開放通知書を見る (ツール起動) 開放通知書を登録する。

表2 オブジェクトの属性

オブジェクト	インスタンス	意味
[共通]	body material exportTool modifyTool displayTool	自分自身 自分自身作り出すためのオブジェクト 自身の作り方 自身の操作方法 自身の表示方法
実行形式	[共通]	
モジュール	[共通]	
オブジェクト	[共通]	
ソース	[共通]	
障害処理票	[共通] status executableFile checkList bugReport fixedFile	自分の属する view 発生した実行形式 試験手順 関連障害処理票 修正した実行形式
試験手順書	[共通] executableFile	被試験対象実行形式
解析状況票	[共通] bugReport modul	障害処理票 モジュール
開放通知書	[共通] descript bugReport executableFile	仕様書 障害処理票 実行形式