

階層モデルによるソフトウェア開発環境

貫井春美 三原幸博

(株式会社東芝 システム・ソフトウェア技術研究所)

6L-7

1. はじめに

ソフトウェアの開発は、開発規模の拡大と開発分野のエキスパート化に伴い、開発の形態が個人単位の開発からグループ単位での開発へと移行してきている。

グループ単位でのソフトウェア開発は、単に作業を分担するだけでなく、いつ・だれが・どこで・なにを・どのようにするかという方法論を定義することが重要となる。

本稿では、グループ単位でのソフトウェア開発を“人”を中心に考察し、IMAPシステムに於けるソフトウェア開発作業モデルを提案する。

2. ソフトウェア開発のための資源

2-1. 構成要素

ソフトウェア開発プロジェクトは様々な資源により構成される。構成要素の主なものとして以下が考えられる。

- (1)人 設計者, プログラマ, 管理者等その他関連する人々
- (2)装置 ホストコンピュータ, ミニコンピュータ, ワークステーション, その他入出力機器等
- (3)サービス (ツール)
 - 開発支援ツール (設計・テスト支援等),
 - 管理支援ツール (進捗・成果物管理等),
 - その他支援サービス
- (4)情報 ドキュメント, ソースプログラム, 設計関連資料, 版管理情報等の開発のに関連する情報

このような資源を、目的・意志・用途の一致したもののによってグループ化し、ソフトウェアプロジェクトを構成する。こうすることにより、個々の資源が関連付けられ、作業の関連が明確になる。さらに、グループの構造がプロジェクト構成を表し全体の把握が容易になる。

2-2. 人資源のグループ化

人資源のグループ化は、開発目的の一致したもの、

意志の一致したものを基準に行う。これには、データ共有の効率化・機密保護・コミュニケーションの効率化というねらいがある。

グループ化した人資源の各構成要素は大きく以下のように分類される。

- ・グループリーダー
- ・グループメンバ

すなわち、グループは一人のリーダーと一人以上のメンバーから構成される。また、開発規模により複数のグループからさらにグループが階層的に構成される場合もある。

Fig1に人資源のグループ化の概念を示す。

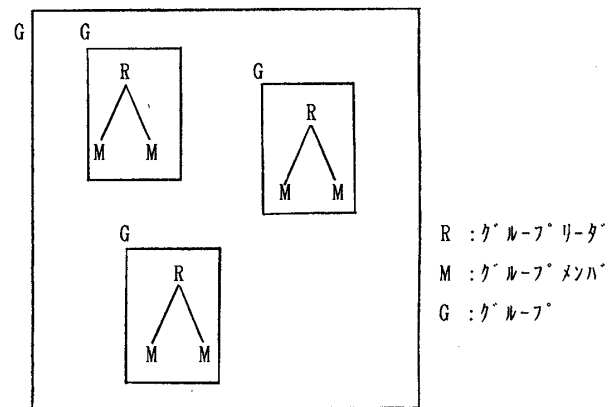


Fig1.人資源のグループ化

3. グループでのソフトウェア開発

ソフトウェアのライフサイクル[1]に従い開発工程を個人単位で作業する場合、各作業間のつながりは個人のみが知ればよい。しかし、ソフトウェア開発プロジェクトは、多くの人資源により構成され、これらの人資源はそれぞれの作業を独立して行っているのではなく他のメンバーと共同して進めている。Fig2にソフトウェアモジュール設計をグループ単位で行った時の作業分析を示す。

3-1. 部品化再利用の促進

グループリーダーは、グループ共通データの管理を行

Software Development Environment Based On Hierachy Model

Harumi Nukui Yukihiro Mihara

TOSHIBA CORPORATION

う。Fig2での共通パッケージ（ライブラリ、インクルードファイル等）はグループ共通のデータである。リーダはこれを作成してグループメンバーに共通利用させる。さらに、共通パッケージの履歴管理を行いグループメンバーが常に最新のデータを利用できる環境をつくっておく必要がある。こうすることにより、部品化再利用が促進され品質の向上につながる。

3-2. コミュニケーション

開発中のモジュールに仕様変更が発生した時、個人単位の仕事では変更情報（変更日、変更内容）を記録しておけばよい。しかし、グループ単位の開発では、変更を記録すると同時にグループ内の関連するメンバーに通知することが必要となる。（Fig2参照）すなわち、グループリーダは誰に通知すべきかを常に把握していなければならない。また、確実に通知されたかを確認することも必要となる。

このようにソフトウェア開発プロジェクトでは各メンバーのコミュニケーションが円滑に行われ、グループ内での情報交換が容易に行われなくてはならない。さらに、グループだけでなく、プロジェクトに関係ある他部門との情報交換も十分に行えなければならない。

メンバー間で交換（コミュニケーション）すべき情報として以下を考える。

- ・作業の指示、終了報告
- ・仕様の変更
- ・会議、打ち合せ等の開催通知とそれらの議事録

リーダ グループ メンバ

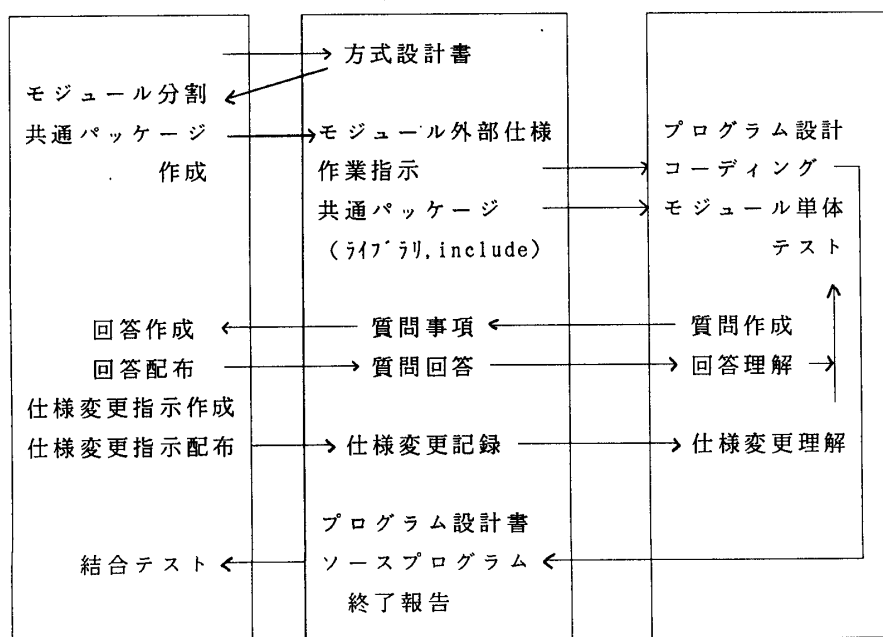


Fig2. モジュール設計・製造の流れ

- ・進捗情報の報告
- ・仕様書、ソースプログラム、資料等の回覧

3-3. 人資源の分散

グループ単位での開発では、人資源が以下のように分散していることがある。

(1) 地理的分散

各人資源の物理的距離が離れている。管理者および担当者がどこで作業しているか意識する必要がある。

(2) 時間的分散

各人資源の作業時間が異なる。この場合作業指示や仕様変更の通知がリアルタイム性に欠けるという問題がある。

このような場合も作業モデルとしては、Fig2に示したような作業の流れが適用できる。しかし、作業指示や仕様変更の通知のようなコミュニケーションが複雑となるため、さらに深く作業モデルに関する検討が今後の課題となる。

4. まとめ

グループ単位のソフトウェア開発を前提としたソフトウェア開発の作業モデルに関して述べてきた。ここで示した作業モデルの特徴として以下があげられる。

- ・部品化再利用の促進
- ・コミュニケーションの円滑化
- ・分散開発への適用

5. おわりに

ソフトウェア生産性向上のアプローチとして

- (1) 開発方法論の確立
- (2) 開発支援環境の充実
- (3) 開発支援ツールの充実

の3つが考えられる。

本稿では(1)を中心に述べてきたが、今後、これを支援する環境およびツールに関する考察を継続する。

<参考文献>

- [1] 大筆他「IMAPシステム(1)~(10)」情処学会第31回全国大会 4F-1~4F-10
- [2] 竹内他「ソフトウェアエンジニアリングに関する調査」日本電子工業振興協会