

## 適応型ソフトウェアプロセスに関する一考察

3W-1

菅原広行、山口智治、福田高秋、白石智

NTT ネットワークサービスシステム研究所

### 1. はじめに

近年、ビジネスにおけるソフトウェアの重要性は更に増加し、ソフトウェアそのものがビジネスの中心となっており、要求仕様への即応、タイムリーなリリース、急速な進化・発展が要求されている。ソフトウェア生産性や品質を抜本的に改善するため、ソフトウェア開発プロセスが重要な課題として研究されており、waterfall model (Royce, 1970) 以降、プロトタイプング、スパイラル・モデル、インクリメンタル開発プロセス、Iterative enhancement、RAD (Rapid Application Development)、Cleanroom 手法、CMM、PSP 等、様々な提案がなされてきている[1-6]。

一方、今日のソフトウェア開発は先述の要求に加えて、アーキテクチャ、分析・設計方針、開発言語、開発環境・形態等が多様化してきており、一様なソフトウェア開発プロセスでは、これらの状況に対応できなくなっている。また、最適なプロセスを選択し、開発計画を立てることは、プランナーの経験に拠るところが大きいと言える。

そこで、本稿では、開発対象や開発プロジェクトの開発条件に応じて、その都度適切な開発プロセスを適用することによって、開発効率や品質の最適化を実現することを目的とした、適応型のソフトウェア開発プロセスを提案する。また、適応型プロセス・システムの構想を示す。

### 2. 適応型プロセス

#### 2.1. 基本コンセプト

多様化する開発条件に応じて、プロセス・コンポーネントを選択あるいは組合せて最適なプロセスをコーディ

ネートする。また、開発状況に応じて、そのプロセスに沿うように開発作業を自律的に補正していく(図1)。

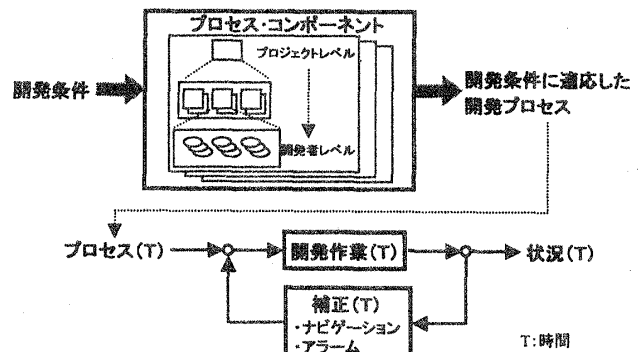


図1: 適応型ソフトウェア・プロセス

#### 2.2. 適応型プロセスの概要

- (1) 適用する開発プロダクトの開発期間、開発規模、リソース、予算、適用するソフトウェア技術、開発ツール等の開発条件に応じて、最適な開発プロセスをコーディネートする。プロセスをコーディネートする対象は、プロジェクト、エキスパートグループなどの開発グループ(例えば、アーキテクト、デザイナー、インテグレータ、テスター等)、そして開発者個人である。
- (2) 実際の開発に際して、開発作業を通じて開発データを取得し、その状態に応じて、開発プロセスをナビゲートする。ナビゲートする対象はプロジェクト、開発グループ、開発者個人である。
- (3) 実際の開発経験をフィードバックして、プロセス・コーディネーションの部品となるプロセスコンポーネント(プロセス事例モデル)を改善する。

#### 2.3. 適応型プロセス・システム

図2に適応型プロセスのシステム構想を示す。

本システムを使用した場合の主な手順及び機能を以下

#### Adaptive Software Process

Hiroyuki Sugawara, Tomoharu Yamaguchi, Takaaki Fukuda, and Satoshi Shiraishi

{sugawara.hiroyuki, yamaguchi.tomoharu, fukuda.takaaki, shiraishi.satoshi}@nslab.ntt.co.jp

NTT Network Service Systems Laboratories

3-9-11 Midori-Cho Musashino-Shi, Tokyo 180-8585, Japan

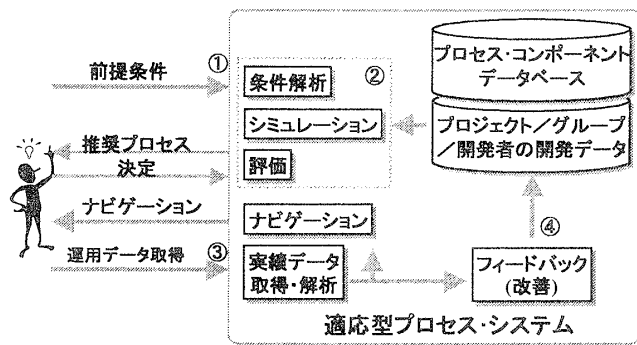


図2: 適応型プロセス・システム

に示す。

① 前提となる開発条件設定

開発プロセスを決定する前提条件を設定する。具体的な項目を図3に挙げる。

② シミュレート ⇒ 最適プロセスをコーディネート

前提条件を過去の開発事例や提案モデルで構成されるプロセス・コンポーネントにあてはめ、開発要求に最も合致した開発プロセスをコーディネートする。

③ 実開発のナビゲーションと開発実績データ取得

実開発の作業実績データを取得しながら、先にコーディネートした開発プロセスに沿って作業をナビゲートする。

④ 実績データによるデータベースの更新

実績データを元にプロセス・コンポーネント(モデル)及びプロジェクト/グループ/開発者データを見直す。例えば、その対象ドメインでの生産性や開発者のスキル等。

2.4. 期待される効果

適用型プロセスを適用することにより、以下の効果が期待できる。

- 開発条件を満足する最適な開発計画を立案できる。
- 円滑な進捗・品質管理を実現できる。
- 開発プロセス事例(モデル)を再利用できる

3. 課題と今後のとりくみ

本稿では、適用型プロセスの基本コンセプトを提案し、そのシステム構想を示した。しかし、その実現においては多くの課題がある。以下に、主な課題を挙げると共に、今後の取組み方針を示す。

(1) プロセス・コンポーネント・データベースの構築

本システムの実現にあたっては、プロセス・コンポーネ

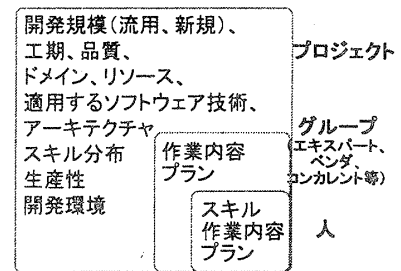


図3: プロセス決定の前提条件項目

ント・データベースを如何に構築するかが最も重要な課題となる。先に挙げたプロセス決定の前提条件をパラメータとして、開発条件との整合性や開発効率等の定量的な比較、判断を可能とするプロセスのモデル記述方式を確立する必要がある。更に、デザイン・パターン[6]に見られるように、記述テンプレートの統一的なフォーマット作成が必要と考える。

(2) 実績作業データ取得とナビゲーション方式

現状の市販プロセス管理支援システムは、開発者や管理者が実績データを別途投入するが、本システムでは、開発に携わる各人の作業(アクション)を通じて実績データの取得を図る。また、ナビゲーションについても、開発者に次の作業を指示し、且つある程度アクションを制限できるものでなくてはならない。従って、開発者のアクション(コマンド)を解釈、作業状態を把握し、次の作業を指示するエージェントによる実現を検討する。

4. まとめ

本稿では、適用型プロセスの基本コンセプト及びそのシステム構想に関して報告し、実現に向けた課題を挙げた。

参考文献

[1] B. W. Boehm, "A Spiral Model of Software Development and Enhancement," ACM Software Eng. Notes, Aug. 1986.  
 [2] V. Basili et al., "Iterative Enhancement: A Practical Technique for Software Development," IEEE Trans. Software Eng., 1975.  
 [3] H. D. Mills et al., "Cleanroom Software Engineering," IEEE Software, Sept. 1987.  
 [4] M. C. Paulk et al., "capability Maturity Model for Software, Ver. 1.1," Software Engineering Institute, CMU/SEI-93-TR-25.  
 [5] W. Humphrey, "Using A Defined and Measured Personal Software Process," IEEE Software, May 1996.  
 [6] M. Aoyama, "Agile Software Process and Its Experience," ICSE, April 1998.  
 [7] E. Gamma et al., "Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.