

ソフトウェア開発における 管理プロセス記述の試み

渡辺淳志[†] 井上克郎[†] 鳥居宏次[‡]

[†] 大阪大学基礎工学部

[‡] 奈良先端科学技術大学院大学

[†] 〒 560 大阪府豊中市待兼山 1-1

[‡] 〒 630-01 奈良県生駒市高山町 8916-5

あらまし

ソフトウェアの開発において管理作業は最も重要な作業の一つである。しかし、従来のソフトウェア開発プロセスの記述では製品に直接かかわるような開発作業は記述されるが、それに付随する管理作業はほとんど記述されない。

本研究では管理作業に重点をおいたプロセス記述方法について提案し、それに基づいたプロセス記述例を示す。また提案した方法を用い、CMM(Capability Maturity Model)に基づいた開発プロセスの記述を行なう。

和文キーワード プロセス、管理作業、CMM(Capability Maturity Model)

A description method of management processes in software development

Atushi Watanabe[†] Katsuro Inoue[†] Koji Torii[‡]

[†] Faculty of Engineering Science, Osaka University

[‡] Advanced Institute of Science and Technology, Nara

[†] 1-1 Machikaneyama, Toyonaka, Osaka 560, Japan

[‡] 8916-5 Takayama, Ikoma, Nara 630-01, Japan

Abstract

Knowing Management processes is one of the most important issues in the software development. Reseraches on the software process mainly focus on the product-producing activities, and not on the management activities.

In this paper, we propose a way of process description for various management activities, and show some examples of such description. In addition, we describe the management processes which are based on CMM(Capability Maturity Model) using our process description method.

英文 key words process, management activity, CMM(Capability Maturity Model)

1 まえがき

近年ソフトウェアプロセスの研究が盛んに行なわれている。ソフトウェア開発にともなう作業は要求定義、仕様作成、設計、コーディング、テスト等、要求される製品の直接/間接的な作成に関するもの(これらの作業をここでは開発作業と呼び、またその系列を開発プロセスと呼ぶ)とそれ以外の資源分配、スケジュール調整、品質保証等、作成作業の制御に関するもの(管理作業、管理プロセス)に分けられよう。

しかし、今までの多くの研究の中心は開発プロセスの記述についてであり、管理プロセスについてはそれほど研究が行なわれてはいない。

この原因としては、ソフトウェア開発における管理対象が組織やプロジェクト毎に異なることが多く、一般的なモデルの作成が難しい点、また開発作業と管理作業の両方をプロセス記述することでプロセス図が複雑になり過ぎ、実用に適さなくなる恐れがある等が挙げられる。

管理作業について、開発作業や管理作業自身を一定の「演算対象」とする関数とみなしていわゆる高階関数を用いて、開発プロセスと管理プロセスを同一レベルで厳密に記述しようとした試みはあるが[3]、意味定義が比較的複雑であり、また直観的に全体を把握するのが困難である。

ここでは、管理作業を高階なものとして開発作業と分離するのではなく、各管理作業もある種の生産物を生成する作業として開発作業と同じレベルで扱うことにする。ある管理作業に制御される開発作業は、その管理作業が作った生産物を入力の一つとして受けとることによって制御が行なわれると考える。

これらの点を踏まえ、本稿では管理プロセスの記述方法について提案する。各管理作業の対象を資源管理に関するもの、作業管理に関するもの、生産物管理に関するものの3つに分類する。同時に、それらを各管理作業を単位作業クラス、プロジェクトクラス、組織クラスと呼ぶ3つの粒度に分類する。

ここではこれらの分類を基に管理プロセスについての記述例を示す。

また、CMM(Capability Maturity Model)のレベル2及びレベル5のGoalを実現するために必要な管理プロセスの記述を行なう。

2 ソフトウェア開発プロセス

2.1 管理作業とは

本研究ではソフトウェアの開発におけるプロセスを(図1)のような形で考える。すなわち、生産物を基に作業を行ない新たな生産物を生み出す形である。生産物は作業によって生み出されるものとして定義される。

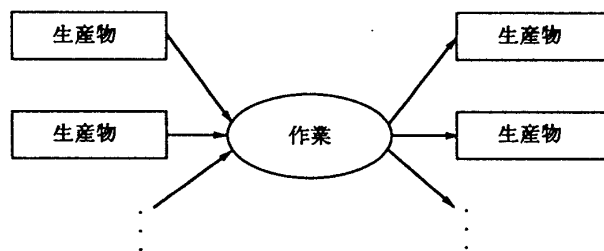


図1：ソフトウェア開発モデル

作業は大きく分類すると開発作業と管理作業の2つに分けることが出来る。開発作業はユーザに渡す製品を作成する作業であり、管理作業は製品そのものではないが製品の作成に必要な生産物を作成する作業である。

例えば、ユーザに渡すドキュメントの作成は開発作業である。一方、開発スケジュールのドキュメントの作成は管理作業である。

開発作業には要求定義や仕様作成、設計、コーディング、テスト等が含まれる。この場合、要求定義書、仕様書、設計書、ソースプログラム等が開発作業に対する入出力生産物となる。

一方、管理作業にはスケジュール管理や品質保証作業、作業者の割り当てといった作業が含まれる。そして、スケジュール表、品質に関するメトリクスデータ、作業の割り当て表などが管理作業の入出力生産物になる。

2.2 管理作業の分類

ソフトウェアの管理作業については従来様々な分類がなされてきた。

例えば[5]では開発管理の分類例として管理対象を工程管理、製品管理、原価管理、品質管理の4つに分類している。

また、一般に広く用いられている管理対象の分類としてはQCD(Quality, Cost, Delivery)が挙げられる。すなわち管理対象をソフトウェア開発における重要な管理項目である品質、コスト、スケジュールの3点に分類する方法である。

これ以外にも[1]における4MすなわちMan, Money, Machine, Materialによる分類等がある。

本稿ではソフトウェア開発における管理項目を資源、作業、生産物の3つに分類する(表1)。これはソフトウェアの開発が要求に対して資源を用いて作業を行ない、生産物を作成するものであると考えることで管理の対象が資源、作業、生産物の3つに分類できると考えたからである。なお、要求は管理の対象とはならないと考え、今回は管理対象の分類から除いた。

分類	管理項目
資源	人、設備、技術等
作業	スケジュール、公約、標準プロセス等
生産物	ソース、ドキュメント、品質等

表1：ソフトウェア開発における管理対象の分類

以下、資源、作業、生産物の各々の管理項目についてより詳細な分類を行なう。

1. 資源

資源の範囲に入る管理項目としては、人、設備、技術等が挙げられる。

人に対する管理作業としては、作業者の雇用、解雇、配置、トレーニング等が挙げられる。設備に対する管理作業についても人と同様のプロセスと考えることができる。すなわち、設備の購入、廃棄、配置、改良等である。また、技術に関しては新規技術の導入とそれに付随する管理作業等が考えられる。

2. 作業

作業の範囲に入る管理項目としては、スケジュール、公約、標準プロセス等が挙げられる。

スケジュールに対する管理作業としては、計画の作成、進捗の監視・評価、計画の調整等が考えられる。公約はプロジェクト実行のために定められる作業仲間での取り決めである。管理作業としては公約の決定、変更等が挙げられる。標準プロセスは組織における標準的な作業の進め方である。管理作業としては標準プロセスの作成、変更等が考えられる。

3. 生産物

生産物の範囲に入る管理項目としては、ソース、ドキュメント、品質等が挙げられる。

ソースに対する管理作業としてはソースの評価や構成管理、再利用のためのソースの集合の管理等が挙げられる。ドキュメントの管理作業については管理のためのドキュメントの作成やその評価、ドキュメントの集合の管理等がある。品質については品質管理作業が挙げられる。品質はソースやドキュメントなど全ての生産物に付随するものだが生産物における最も重要な管理要素のため一つの管理項目とした。

2.3 作業の粒度について

従来、管理作業があまりプロセス記述されなかった原因の一つとして、開発作業と管理作業の両方のプロセスを記述することでプロセス図が複雑になり過ぎ、記述や理解が困難になる点が挙げられる。

この問題を解決するために、本稿ではプロセスの記述を単位作業クラス、プロジェクトクラス、組織クラスの3つの粒度に分けて行なう。

3つ粒度の関係は(図2)で示される。すなわち、組織クラスにおいて記述されるプロジェクトを表す作業をより精密に記述するレベルがプロジェクトクラスであり、プロジェクトクラスにおいて記述される任意の作業と生産物の関係をより精密に記述するレベルが単位作業クラスである。

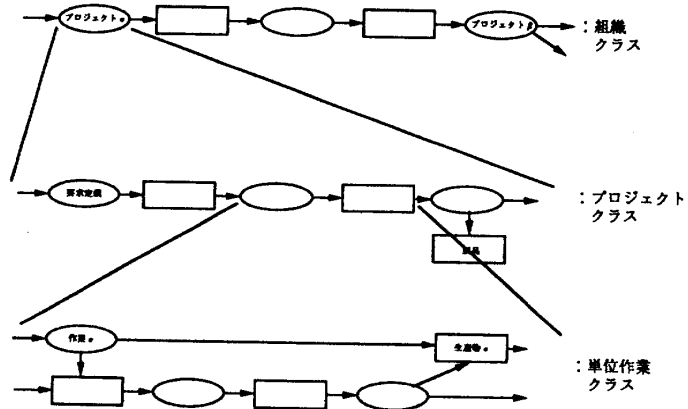


図2：各レベル間の関係

以下、各クラスについて簡単に説明を行なう。

1. 組織クラス

組織クラスは一つのプロジェクトが終了してから次のプロジェクトを開始するまでの間に行なわれるプロセスを記述するために使用される。

このクラスにおける管理作業は例えば生産物に関して言えば再利用のためのソース集合の管理等が挙げられる。

2. プロジェクトクラス

プロジェクトクラスは一つのプロジェクトにおいて行なわれるプロセスを記述するために使用される。

このクラスにおける管理作業は例えば生産物に関して言えばプロジェクトにおける品質目標の作成等が挙げられる。

3. 単位作業クラス

単位作業クラスはプロジェクトにおける任意の一つの作業とその生産物に付随するプロセスを記述するために使用される。

このクラスにおける管理作業は例えば生産物に関して言えばソースの構成管理等が挙げられる。

3 管理プロセスの記述

3.1 プロセスの記述方式

2章における分類を基に管理プロセスを記述する。

(図1)に示すように作業を楕円で表し、生産物を長方形で表す。

作業と生産物の間には有向辺が存在する。この作業と生産物の集合及びその有向辺で構成される有向グラフをプロセスと呼ぶ。ただし、一つの作業はそれだけでプロセスであるが、一つの生産物はプロセスではない。生産物は作業とのつながって初めてプロセスとなる。

作業から生産物への辺は生産物が作業によって生成されることを表し、生産物から作業への辺は生産物が作業において使用されることを表す。作業から作業への辺はなく、生産物から生産物への辺もない。

3.2 プロセス記述例

以下、いくつかのプロセス記述例を示す。

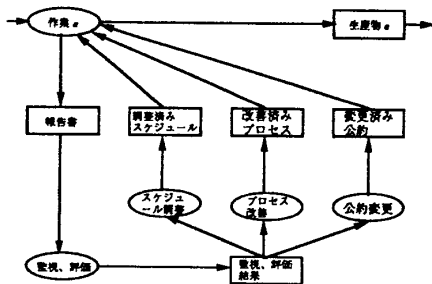


図3：作業（単位作業クラス）

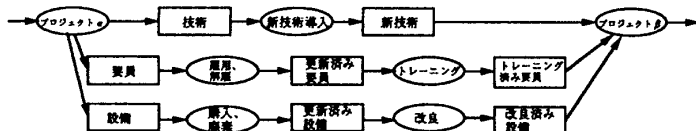


図4：資源（組織クラス）

例えば(図3)は単位作業クラスにおける作業に対する管理作業を表している。すなわち、作業αの実行時に報告書を基に監視・評価作業を行なう。また、監視・評価結果を基にスケジュール調整、プロセス改善、公約変更を行ない、その結果を作業にフィードバックする。

また、(図4)は組織クラスにおける資源に対する作業を表している。すなわち、プロジェクトα終了によって発生する要員、設備に対してそれぞれ更新、改良を行なう。また、プロジェクトαにおいて用いられていた技術に対して、必要であれば新技術を導入する。

これらの記述による利点は、組織における標準的な管理プロセスの記述を単位作業クラス、組織クラスにおいて定義しておけばプロジェクト管理者はプロジェクトクラスのプロセスを記述するだけで詳細な管理プロセスの記述を含んだ開発プロセスの記述をしたことになる点である。

なお、これらの記述はあくまで管理プロセスの記述の一例に過ぎない。

4 CMMに依ったプロセス記述の例

4.1 CMM

CMM(Capability Maturity Model)はソフトウェアを開発する会社や組織のプロセスがどれだけ成熟しているかを評価する指標である。CMMによって組織の現状を評価することが出来、また組織を改善するためにも役立つ。

CMMは米国国防総省の要請に基づきCarnegie Mellon UniversityのSEI(Software Engineering Institute)のHumphryらが中心となって1991年にV.1.0が作成された。現在は1993年に作られたV.1.1[6]が使用されている。

CMMではプロセスの成熟度をアンケート等を用いてInitial, Repeatable, Defined, Managed, Optimizedの5段階に分類する。

4.2 CMMレベル2に基づいたプロセスの記述例

(図5(a), (b), (c))はあるソフトウェア開発プロジェクトにおけるプロセスを単位作業、プロジェクト、作業の各クラスについて記述したものである。

このプロセス図には主に開発作業のみが含まれており、CMM レベル1に相当するプロセスであると考えられる。

(表2)はCMM レベル2で実現せねばならない目標(Goal)である。

この目標を実現するために必要と思われるプロセスを(図5(a),(b),(c))のそれぞれに付加したものが(図6(a),(b),(c))である。各作業や生産物の下の数字はその作業や生産物がCMMのどの目標を実現するために必要なのかを表している。

4.3 CMM レベル5に基づいたプロセスの記述例

同様な方法で、CMM レベル3,4,5についてプロセスの記述を行なった。

ここでは紙面の関係でCMM レベル5のプロジェクトのプロセス記述のみを示す(図7(a),(b),(c))。

CMMではGoalとして、組織や作業が達成しなければならない目標が記述されており、各クラスの記述にそれが現れている。

これらのGoalはプロセス記述においてある程度解釈の余地がある。また、今回のプロセス記述の基とし

(1) 要求マネージメント (ゴール1) ソフトウェアに配置されるシステム要求がソフトウェアエンジニアリングとマネージメントのための基準を見積もるために制御される。 (ゴール2) ソフトウェアの計画、製品、作業がソフトウェアに配置されるシステム要求と一致している。
(2) ソフトウェアプロジェクトの計画 (ゴール1) ソフトウェアの見積もりが計画や後の調査に使えるように文書化されている。 (ゴール2) ソフトウェアプロジェクトの作業と公約が計画され文書化されている。 (ゴール3) 関係するグループと個人がプロジェクトに関する公約に同意している。
(3) ソフトウェアプロジェクトの追跡と監視 (ゴール1) 実際の結果と実行が計画に反しているかどうか追跡される。 (ゴール2) 調整作業が行なわれ、実際の結果と実行が計画から外れた時にメ切がマネージメントされる。 (ゴール3) 関係するグループと個人による同意がなされない限り、ソフトウェアの公約の変更を行なわない。
(4) ソフトウェアの下請けマネージメント (ゴール1) 主契約者が適任なソフトウェア下請けを選ぶ。 (ゴール2) 主契約者と下請けが互いの公約に同意している。 (ゴール3) 主契約者と下請けが進行中のコミュニケーションを維持している。 (ゴール4) 主契約者が下請けの作業結果と実行が公約に反していないかどうか追跡できる。
(5) ソフトウェア品質保証 (ゴール1) ソフトウェア品質保証作業が計画されている。 (ゴール2) ソフトウェアプロダクトと作業への執着(標準, 手続き, 要求)が客観的に変更される。 (ゴール3) 影響されるグループと個人がソフトウェア品質保証作業と結果を知らされる。 (ゴール4) プロジェクトにおいて解決出来ない難しい問題がシニアマネージメントによって処理される。
(6) 構成マネージメント (ゴール1) ソフトウェア構成マネージメントの作業が計画されている。 (ゴール2) 選ばれたソフトウェアの仕事のプロダクトが識別され、制御され、利用される。 (ゴール3) 識別されたソフトウェアの仕事のプロダクトへの変更が制御される。 (ゴール4) 影響されるグループと個人がソフトウェアの基準の状態と内容を知っている。

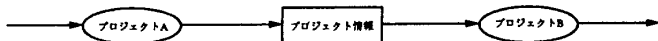


図5(a)：CMMレベル1における組織クラスでのプロセス

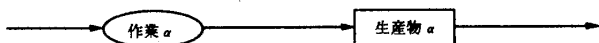


図5(c)：CMMレベル1における単位作業クラスでのプロセス

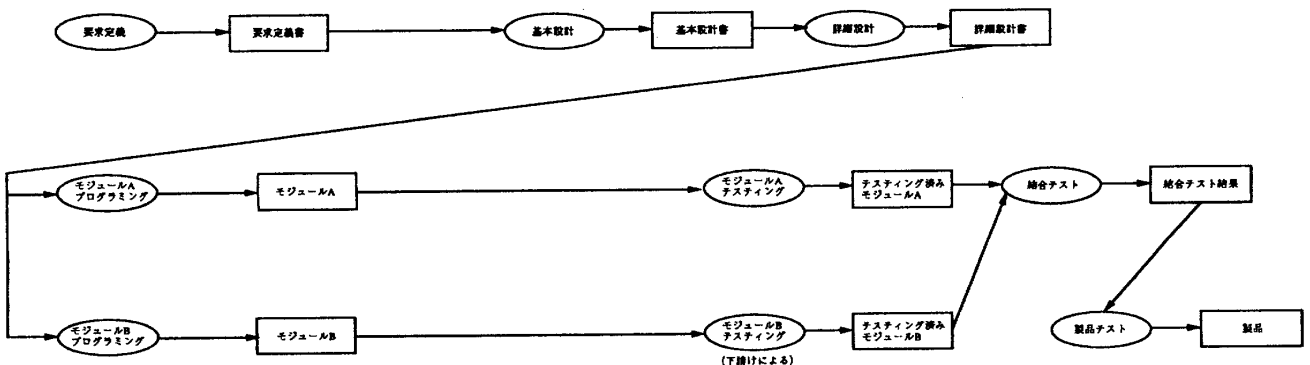


図5(b)：CMMレベル1におけるプロジェクトクラスでのプロセス

たレベル1のプロセスもあくまでプロセスの一例に過ぎない。よって今回のレベル2, レベル5のプロセス記述もあくまでCMMに基づいたプロセス記述の一例に過ぎない。

しかし, CMMの各レベルにおいて必要な作業をプロセス図として記述することで, CMMの理解, 及び組織におけるプロセス改善に役立つであろう。

また, プロセス図を用いることでCMMの手薄な点, 例えば設備に関する管理作業について記述されていないといった点を把握することが出来, CMMの改良や新しいプロセス指標の作成にも役立つものと思われる。

なお, 管理対象を資源, 作業, 生産物のそれぞれに分割してプロセス記述することが出来るが, ここでは簡略化のためまとめて記述した今後より詳細な解析を行なう場合等は, 分けて書く必要がある。

5 まとめ

本論文では管理作業の記述を行なうためにソフトウェア開発のモデルを基に管理作業を資源, 作業, 生産物の3つに分類し, さらに3つの粒度に分類してプロセス記述の例を示した。また, CMMに基づいた管理プロセスの記述の例を示した。

今後の課題としては, 今回のプロセス記述を実際に行なわれたプロジェクトに対して適用しプロセスの記述を行なうことや, 今回のプロセス記述を基にした管理支援システムの作成等を考えている。

参考文献

- [1] 東基衛, 細谷僚一, 高橋宗雄:“ソフトウェアマネジメント概説”, 情報処理, Vol.33, No.8, pp.894-905(1992).
- [2] 後藤太樹, 佐伯元司:“ソフトウェアプロセスの形式的記述とその記述をもとにしたプロジェクト管理支援”, ソフトウェア科学会ソフトウェアプロセス研究会, SP93-1-9, pp.75-84(1993).
- [3] M.Suzuki, K.Katayama:“Meta-Operations in the Process Model HFSP for the Dynamics and Flexibility of Software Processes”, PROCEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON THE SOFTWARE PROCESS, pp.202-217, (1991).
- [4] Watts S.Humphrey, “Managing the Software Process”, Software Engineering Institute, Addison-Wesley, 1989.
“ソフトウェアプロセス成熟度の改善”, 日本電気ソフトウェアプロセス研究会訳, 日科技連, 1991.
- [5] 藤野喜一, 花田收悦:“ソフトウェア生産技術”, 電子情報通信学会, 1985.
- [6] Mark. C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles V. Weber: “Capability Maturity Model for Software, Version 1.1”, CMU/SEI-93-TR-25, February 1993.
- [7] Edward Yourdon: “Decline & Fall of the American Programmer”, A Simon & Schuster Company, 1992.
“ソフトウェア管理の落とし穴: アメリカの事例に学ぶ”, 松原友夫訳, トッパン, 1993.