

NAIST-IS-DT0261016

博士論文
2005年3月

プロセス改善のための成果物観点による
ソフトウェアプロセスモデル化手法

田中 康

奈良先端科学技術大学院大学
情報科学研究科
情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
博士(工学)授与の要件として提出した博士論文である.

田中 康

審査委員: 松本 健一 教授
西谷 紘一 教授
飯田 元 助教授

プロセス改善のための成果物観点による ソフトウェアプロセスモデル化手法

内容梗概

ソフトウェア工学の技術的な進展の一方で、スケジュールの遅れ、ソフトウェア品質に基づく事故、そしてコストの見積もり間違いなど、ソフトウェア開発プロジェクトの進め方に起因する失敗が重要な課題となっている。このような問題に対して「ソフトウェアプロセス」という概念が登場し、ソフトウェアプロセスの観点から、ソフトウェア開発が抱える問題を解決するための様々な技術の提案や議論が行われてきたが、依然として、ソフトウェア開発プロジェクトの失敗原因は改善されていない。特に、失敗原因を改善するために定義されたソフトウェアプロセスが実際の開発プロジェクトで利用されないために、改善が実現しないといった問題を抱えている。

本論文では、定義されたソフトウェアプロセスが実プロジェクトで利用されない問題を、プロセスの記述方法、すなわちプロセスのモデル化方法の観点から分析し、プロセスのモデルが現実の開発活動を適切に表現していないことに根本的な原因があると予測した。つまり、従来のプロセスモデルには、高度に一般的で抽象的なプロセスの流れを示すことができるものや、あるいは特定のプロジェクトのプロセスの中身を非常に具体的に記述することのできるものは存在するが、これらは、計画策定や進捗管理に直接活用できないか、あるいは、複数のプロジェクト間で繰り返し活用できないといったように、具体性と一般性のいずれかに短所が存在する。プロセス改善のための参照モデルとして現在広く利用されている CMM(Capability Maturity Model)や CMMI(Capability Maturity Model Integration)では、定義したプロセスをプロジェクトが利用するために「制度化」というプロセス遵守のための規律を定義している。しかし、このような「制度化」につとめても、従来のプロセスモデル化手法を用いて定義したプロセスが現実の開発活動を適切に表現していないため、実際の開発プロジェクトで利用されなくなってしまう、さらに改善の効果を期待することも困難になる。そこで、現実の開発活動を適切に表現し、プロセスの改善および開発プロジェクトでの利用に適したプロセスモデルとして PReP(Product Relationship Process)モデルを開発した。さらに、PReP モデルのプロジェクトの管理およびプロセス改善への利用方法を定義し、そのための支援環境、および組織への導入モデルの定義を行った。

従来のモデル化の方法の多くは、タスク(作業)の時間的順序関係によってプロセスをモデル化している。一方、PReP モデルは、開発プロセスの中で作成され管理される実体である成果物に着目したプロセスのモデル化方法であり、成果物の入出力の関連構造に着目したモデル化方法を特色としている。モデル化の方法として、成果物と成果物間の関連の種別を分類し、その表記方法を定義した。

PReP モデルでは、最終成果物を作るために必要となる成果物の入出力の関連構造が、適用する技術と管理方法とによって一意に決まるため、同等のプロファイルを持つプロジェクト間では、90～100%の再利用率となり、定義したプロセスの再利用性が高いことが分かった。さらに、調査の結果、モデルの理解性、改善への適用性、プロジェクト管理への利用性において優れているとの評価を得た。また、PRePモデルでは、成果物を“作業者に割り当てられた作業の成果物”と定義することによって、モデルの粒度とプロジェクトの管理粒度が一致するために、プロジェクトの計画策定や進捗管理への適用性が高い。特に、プロジェクト管理のためのスケジュール作成に必要な内容と構造とになるため、記述したモデルからプロジェクト管理で使用するガントチャートへの自動変換が可能であることを確認した。

現在、筆者の所属する企業の開発組織では、本論文で定義したプロジェクト管理とプロセス改善への利用方法と、組織への導入モデルに従い、PRePモデルの導入と適用を行っている。プロセスの記述と利用は順調に進んでおり、次の段階の課題として、最適なプロセスに関する議論が行われている。また、本論文で定義したプロセス中心型環境の基本アーキテクチャを基にした様々な応用技術の研究に関する発展が今後見込まれる。

キーワード

ソフトウェアプロセス、プロセスモデル、プロジェクト管理、プロセス改善、プロセス再利用、プロセスパターン、プロセス中心型環境

Product-Based Process Modeling Method for Software Process Improvement

Overview

While the technical advancements continue in Software Engineering, mistakes and failures such as schedule delays, accidents due to software quality and wrongly estimated costs which are caused by how software development projects are managed remain as critical issues. The concept of “software process” appeared on the scene in order to counter these problems. Since then, numerous debates have taken place and various technical solutions have been proposed but, root causes for these failures in software development projects still remain unimproved. In particular, there is a problem of being unable to implement improvement because the software processes that have been defined for fixing the root causes, are not being used in the development projects.

This paper examines this problem of software processes not being used in the real projects from the viewpoints of how these processes are described. In other words, the problem is examined from the process modeling methodology aspect, and has predicted that the fundamental cause of the problem lies in how process models do not adequately describe real-life development activities. In the traditional models, there are ones that can show a highly generic, abstract process flow, and there are ones that can describe in great detail the contents of a particular process from a particular project. These models, in general, have shortcomings either in the level of details or in the level of generality. For example, some models cannot be directly applied to planning and progress tracking and oversight, or, they cannot be reused among multiple projects.

Widely used in the present time as reference models for process improvement, CMM (Capability Maturity Model) and CMMI (Capability Maturity Model Integration) models have defined a discipline called “Institutionalization” that stipulates compliance to defined processes to facilitate the use of those defined processes in projects. Nevertheless, the defined processes are discarded because they are defined using the traditional process modeling methods and therefore, do not adequately describe real-life development activities. This also has resulted in inability to expect and/or predict return on improvement activities. I have developed PReP (Product Relationship Process) Model as a process model especially designed for use in development projects as well as in process improvement in order to address this issue. In addition, I defined how the Model is to be

applied to project management and process improvement. I also defined a model for deploying the Model in an organization, and the support environment.

Most conventional process models describe a process as a relationship in time order. On the other hand, PReP describes a process as input-output relationships between products to be produced and managed in development process. So I defined types and notations of relations between products.

I found that the PReP had a high reusability, because results showed that reuse rate became 90-100% between projects that has similar profiles. That is owing to the mechanism of PReP in which the relationship of input-output structure between products to get final products is defined uniquely according to the technology to be applied and the method for managing a project. Furthermore, I was able to establish that PReP was easy to understand, and is excellent when applied to improvement programs and project management. In PReP, the level of details of a process model is equal to the level of details in arrangement for managing a project because, defining "product" as "the product allocated to a worker". So PReP had a feature that could be widely applied to project planning and progress monitoring. In particular,, I verified that I can convert process model to Gantt-chart used for project management automatically by the mechanism that the details of description of a process model has enough data and structure for generating a schedule.

Now I am applying PReP for project management and process improvement program in my organization by applying method defined in this thesis. I can recognize that process modeling activity and using a defined process goes smoothly, and then PReP is going to be used for improving a process. And process-centered environment based on the architecture defined in this thesis is expected as an application of PReP.

Keywords:

software process, process model, project management, process improvement, reuse process, process pattern, process-centered environment

研究業績

学術論文誌

1. 田中康, 飯田 元, 松本 健一: 成果物間の関連に着目した開発プロセスモデル PReP: (PReP: Product-Based Modeling Method for Software Process,) 情報処理学会論文誌, 2004年「社会人学生」論文, 2005. (採録) .

国際会議(査読有り)

1. Yasushi Tanaka, Riki Utsumi: Reuse of software process assets with knowledge management system, Proceedings of the 2002 International Symposium on Empirical Software Engineering (ISESE 2002), Vol. II - Poster and Research Demonstration Sessions, pp.17-18, 10/2002.
2. Hajimu Iida and Yasushi Tanaka: A Compositional Process Pattern Framework for Component-based Process Modeling Assistance, Proceedings of the 1st Workshop on Software Development Patterns (SDPP'02).. Technical Report TUM-I0213, Munich University of Technology), pp.57-64, 12/2003.
3. Hajimu Iida, Yasushi Tanaka, Ken'ichi Matsumoto.: Daibutsu-den: A Component-based Framework for Organizational Process Asset Utilization, Proceedings of Profes2002, Lecture Notes in Computer Science 2559/2002, pp. 207-219, Springer-Verlag, ISSN: 0302-9743, 2002.

研究会発表

1. 田中康: 形骸化しないさせないプロセス記述方法, SEPG Japan 2004, Proc, pp.31-42, 2004.
2. 田中康: SPI と技術論, 37th SEA-SPIN Meeting, March, 2002.
3. 後藤徹平, 飯田元, 松本健一, 田中康: プロセス資産の再利用を支援するプロセスモデルの提案とそれに基づくプロセス中心型開発環境の試作, 信学技報 SS2002-34, pp.43-48, 2003.1
4. 粕淵健郎, 飯田元, 松本健一, 田中康: ソフトウェアプロセスのグラフモデル表現に基づくプロジェクト事例共有システムの開発, 信学技報 SS2002-34, pp.49-54, 2003.1
5. 岡井洋樹, 松本健一, 飯田 元, 田中 康: コンポーネント指向のプロセスモデルに基づくソフトウェアプロセス測定・分析支援手法の提案, 情報処理学会・研究会報告「ソフトウェア工学」 No.140-010, pp71-78, 2003.3

目次

1 章 はじめに.....	1
1.1 本研究の背景.....	1
1.2 本研究の目的.....	2
1.3 本研究の位置づけ.....	3
1.4 論文の構成.....	4
2 章 プロセスモデル.....	6
2.1 ソフトウェアプロセスとプロセスモデル.....	6
2.2 プロセス改善のためのプロセスモデルの要件.....	7
3 章 プロセスのモデル化方法.....	8
3.1 プロセスモデルの抽象度による分類.....	8
3.2 プロセスの記述方法による分類.....	10
3.2.1 タスク観点のプロセスモデル.....	10
3.2.2 成果物観点でのプロセスモデル.....	11
3.3 プロセス改善に適したプロセスモデルとは.....	12
4 章 成果物の関連によるプロセスモデル PReP.....	13
4.1 PReP モデルの概要.....	13
4.1.1 PReP モデルの特徴.....	13
4.1.2 PReP モデルの構成.....	16
4.2 構造モデルとその記述方法.....	18
4.2.1 構造モデルの構成.....	20
4.2.2 PReP モデルでの「成果物」の定義.....	23
4.2.3 成果物の表記.....	24
4.2.4 成果物の関連の表記.....	26
4.2.5 関連表記での多重度の修飾子.....	28
4.2.6 その他の表記要素.....	29
4.3 構造モデルの記述手順.....	30
4.4 工程モデルとその記述方法.....	33
4.4.1 工程モデルの表記要素.....	35
4.4.2 工程モデルの記述手順と規則.....	36
4.5 スケジュールの導出.....	38

4.6	管理・支援等のプロセスの記述.....	40
4.7	成果物関連によるプロセスモデルの意味.....	40
5	章 PReP モデルの利用.....	41
5.1	PReP モデルによるプロセスの再利用	41
5.1.1	作業モデルの再利用.....	42
5.1.2	構造モデルの再利用.....	43
5.1.3	工程モデルの再利用.....	44
5.2	標準プロセス定義への利用.....	45
5.3	プロジェクト計画策定での利用	47
5.3.1	プロジェクトのプロセスの定義.....	48
5.3.2	スケジュールの定義.....	48
5.4	プロジェクトの実行と管理での利用	49
5.5	PReP モデル使用時のプロセス実績の収集	49
5.6	PReP モデルにおけるプロセスの評価と改善.....	50
5.7	プロセスパターンから見た PReP モデルの利用.....	51
5.7.1	プロセスパターンとは.....	51
5.7.2	PReP モデルでのプロセスパターン.....	51
6	章 PReP モデルの有効性の考察	53
6.1	モデル化方法の検証対象.....	53
6.2	モデル化方法の適用結果.....	53
6.3	PReP モデルの有効性の考察	56
6.3.1	現実の開発活動のモデル化.....	56
6.3.2	モデルの利用観点からの評価.....	57
6.3.3	モデル化の粒度	58
6.3.4	再利用性.....	58
6.3.5	改善への適用例.....	58
7	章 プロセス中心型環境への適用	60
7.1	ソフトウェア開発支援環境.....	60
7.2	PReP モデルに基づくプロセス中心型環境.....	61
7.2.1	論理アーキテクチャ	62
7.2.2	各レイヤーにおけるサービスについて	64
7.2.3	想定するエンティティ.....	68
7.3	プロセス中心型環境での PReP モデルのデータ形式.....	71
8	章 プロセス再利用の導入モデル	75
8.1	プロセス改善から見たプロセス再利用の位置付け.....	75
8.1.1	成熟度レベルとプロセス改善の関係.....	75

8.1.2	プロセスの再利用から見た成熟度レベルの構造	79
8.1.3	成熟度レベルに応じたプロセス再利用の実装方法	80
8.2	プロセス再利用の導入モデル	82
9 章	まとめ・今後の発展	84
9.1	PReP モデルの特長と研究成果	84
9.2	今後の発展	85
9.2.1	PReP モデルのプロセス改善への適用事例研究	85
9.2.2	PReP モデルを用いたプロセス中心型環境の研究	85
	謝辞	87
	参考文献	88
	付録 A 成果物と関連の種別の組み合わせによる ガントチャートへの展開	A-1
	付録 B PReP モデルの DTD 定義	B-1

図表目次

図

図 1 EPMs (Entity Process Models) によるプロセスの記述例.....	12
図 2 成果物の関連によるモデルの基本構成 図 3 時間的順序関係に基づくプロセスモデルの例.....	15
図 4 PReP モデルの構成.....	16
図 5 構造モデルで記述したエンジニアリングプロセスの例.....	19
図 6 構造モデルの構成.....	21
図 7 PReP モデルでの成果物の表記.....	25
図 8 PReP モデルでの成果物の関連の表記.....	26
図 9 同期関連および入力と同期の合成関連の利用.....	27
図 10 多重度の修飾子.....	28
図 11 その他の表記要素.....	29
図 12 ソフトウェアプロセスの範囲の記述例.....	30
図 13 同期関連の正しい記述方法.....	31
図 14 工程モデルの記述例.....	34
図 15 工程モデルの構成.....	34
図 16 工程モデルの定義規則に基づいた記述例.....	37
図 17 成果物と関連の種別に対応したスケジュール展開の基本パターン.....	39
図 18 CMM, CMMI でのプロセスの再利用による改善サイクル.....	42
図 19 PReP モデルを用いたプロセスアーキテクチャ記述の例.....	46
図 20 標準プロセスを使用したプロジェクトのプロセス定義とスケジュール策定の例.....	47
図 21 構造モデル中に見られるプロセスパターン (モジュール実装).....	52
図 22 プロトタイピングのパターン.....	52
図 23 事例1の構造モデル記述例.....	55
図 24 事例1の工程モデル記述例.....	55
図 25 事例1の工程モデルからスケジュールデータへの出力例.....	56
図 26 プロセス上の原因の特定の例.....	59
図 27 PReP モデルをプロセスモデルとして使用するプロセス中心型環境の論理的アーキテクチャ.....	63
図 28 工程モデルの XML 記述での DTD 構造図.....	72
図 29 Microsoft-Visio で描いた工程モデルから Microsoft-Project へのスケジュールデータの変換.....	73
図 30 Microsoft Visio で描いた PReP モデル.....	74
図 31 PReP XML から Microsoft-Project へ変換を行った結果.....	74

図 32 成熟度レベル毎のプロセス能力 (CMU/SEI-93-TR-24 より)	77
図 33 成熟度レベルとプロセス改善の関係	78
図 34 プロセスの再利用から見た成熟度レベルの構造	80
図 35 各成熟度レベルにおける管理層から見たソフトウェアプロセスに対する可視性.....	81
図 36 プロセス再利用の導入に関連するプロセス領域.....	83

表

表 1 ソフトウェアプロセスモデルの抽象度	9
表 2 構造モデルの持つ属性	22
表 3 成果物の持つ属性	22
表 4 作業モデルの持つ属性	22
表 5 代表的な成果物の例.....	24
表 6 モデル化方法の適用結果.....	54
表 7 被験者 34 名による PReP モデルと従来モデルとの有効性の比較評価結果	57
表 8 Entity Layer に配置したサービス	67
表 9 CMM, CMMI での各成熟度レベルの定義.....	79

1 章 はじめに

1 章では、本研究の背景と目的を述べた上で、ソフトウェアプロセス研究における本研究の位置づけを整理する。

1.1 本研究の背景

1960 年代半ば、メインフレームコンピュータが普及するとともに起こったソフトウェア危機^{1,2)}を解決する手段として、1968 年、NATO(北大西洋条約機構)の技術委員会の主催によりドイツで開催されたソフトウェア工学会議の準備会議において、ミュンヘン工科大学の Bauer により「ソフトウェア工学」という概念が提案されたと言われている。その後、要求分析、ソフトウェア仕様、設計法、検証・保守といった、ソフトウェアライフサイクルの各開発段階の明確化の研究が進められ、データの抽象化、構造化、モジュール化などの方法論や各種技法の研究、さらにソフトウェア生産の自動化を目指す CASE(Computer Assisted(Aided) Software Engineering)などの開発環境が提案され現在に至っている³⁾。

ソフトウェア工学の技術的な進展の一方で、ソフトウェアの開発規模の増大に伴い、見積り の 不正確さ、納期の遅れ、ソフトウェア品質に基づく事故など、ソフトウェア開発プロジェクトの管理の側面での失敗が問題となってきた^{4,5)}。大規模ソフトウェア開発プロジェクトの管理という、ソフトウェア開発が抱えるようになった新たな課題に対し、「ソフトウェアプロセス」という概念が登場し、広く議論されるようになったのは 1984 年にイギリスの Egham で行われたソフトウェアプロセス国際ワークショップ(ISPW: International Software Process Workshop)からのことであると言われている^{6,7)}。そして 1987 年、ソフトウェア工学国際会議の第 9 回大会において米国の Osterweil が、「ソフトウェアプロセスもソフトウェアである(Software Process Are Software Too)」という論文⁸⁾を発表して大きな反響を呼び、ソフトウェアプロセスの概念が急速に広まった。

今日、製造、流通、販売、そしてサービスといった産業全般にわたって、ソフトウェアが果たす役割は益々広がっている。ソフトウェアを開発する企業にとっては、高いソフトウェア品質が市場競争における重要な要素となり¹⁰⁾、また、日本の製造業を代表する組み込み系ソフトウェアの開発を行っている企業にとっても、ソフトウェア開発の技術力が企業の競争力を上げるための重要な課題となっている^{11,12,13)}。このような産業構造のソフトウェア比重の増大に伴い、ソフトウェア開発の技術力が産業全体に大きく影響を及ぼすようになりつつあり、この傾向は今後益々増してくると予想される¹⁴⁾。その一方で、ソフトウェアの大規模化に伴ったソフトウェア開発のコストやリスクも増大する傾向にあり、ソフトウェアプロジェクトの失敗は、企業のビジネスに大きく影響をする重要な課題となっ

ている。1960年代半ばにメインフレームコンピュータの普及とともに起こったソフトウェア危機は、ソフトウェア規模の増大、企業競争の激化に伴う品質や生産性競争として形を変えて現在に至っている¹⁵⁾。

このようなソフトウェアの品質や生産性の課題を解決するために、米国では、米国国防総省(DoD:Department of Defense)のソフトウェア調達モデルとして開発されたCMMが、ソフトウェア開発プロセス改善のための参照モデルとして利用され、主にソフトウェアプロセスの管理面における適用の研究が急速に進んだ。一方、急激な規模の拡大や企業間の競争力として重要となったソフトウェア開発に対して、品質・コスト・納期(QCD:Quality Cost Delivery)に関する課題を抱えていた日本の企業においても、1998年頃より、CMMによるプロジェクトの管理技術に着目したプロセス改善の議論がはじまった^{16) 17)}。2002年4月には、プロセス改善に関する議論の場として日本SPIコンソーシアム(JASPIC)が設立された。現在、より高い生産性の実現や市場の要求の変化に対して、より柔軟な対応能力が求められており、プロセス改善はますます重要な課題となっている^{18) 19)}。

しかし、例えばCMMを用いたプロセス改善では、ビジネス目標に合わせてプロセスを最適化することができるレベル5(CMMでは、組織のソフトウェア開発能力のレベルが、能力の低いレベル1から5段階で定義されている)の能力を獲得するためには、平均で7年10ヶ月を要することが報告されており²⁰⁾、プロセスの改善には多大なコストと時間が必要となる。この7年10ヶ月という時間は、現在の企業間の競争の状況からは非常に長い時間であり、実際の開発現場では、短期間での改善効果が強く求められている。また、CMMおよびCMMIでは、組織の標準となるプロセスを定義し、開発プロジェクトが定義したプロセスを再利用し、その結果を定量的に評価することによってプロセスを改善する仕組みが定義されているが、定義された標準プロセスが開発プロジェクトで利用されずに形骸化してしまい、プロセスの改善が非常に困難になる問題が多く報告されている^{21, 22, 23)}。このような状況の中、プロセス改善を確実に、短期間に、そして少ないコストで実現する方法が求められている。

1.2 本研究の目的

ソフトウェアの開発プロセスを改善するためには、ソフトウェア開発で行われている活動を外在化すること、すなわちモデル化が必要である。プロセスをモデル化することにより、ソフトウェア開発活動を客観的に理解することができ、開発方法を評価し、問題を特定することができ、改善した開発の進め方を実際の活動に適用することができる。本研究では、現在、プロセスの改善を困難にしている原因として、このプロセスの外在化の方法に着目した。

従来プロセスのモデル化方法には、ウォーターフォールモデル^{24) 25)}やスパイラルモデル²⁶⁾など、目的に応じた様々なモデルが提案されてきた。例えばウォーターフォールモデルやスパイラルモデルは抽象度の高い概念的なレベルのモデルを与えるものであり、プロセスを概観するために有効であった。また、プログラミング言語や状態遷移図、形式文法等を用いたプロセスモデルは、

個別の作業に対する標準的な作業工程を記述することができ、また、作業の自動化の研究などが行われた。しかし、概念的なレベルのプロセスモデルでは、開発プロジェクトの概観を理解するためには役に立つが、計画策定時において見積もりを立て、プロジェクトの評価を行うためには抽象度が高すぎ、そのため、進捗の管理にも直接活用することができなかった。一方で、個別の作業に対する標準的な作業工程を記述したプロセスモデルは、変動の無い非常に局所的な作業を記述する場合には再利用が可能であるが、範囲を広げると、現実の活動は常に最適化のために変動しているため、即座に記述したものと乖離し、利用できなくなってしまう。その結果として、定義したプロセスが開発プロジェクトの管理やプロセスの改善に利用されず形骸化してしまうといった問題を抱えていた^{21, 22, 23)}。このように、従来のプロセスのモデル化方法は、プロセス改善に用いるには十分な方法ではなかった。

そこで、本研究は、ソフトウェアプロセスの改善に適したプロセスのモデル化の方法を特定し、開発することを目的とし、プロセスのモデル化方法および記述したプロセスモデルの利用方法に関する研究を行った。研究は、産学協同の研究として行った。筆者が企業と大学に所属する利点を生かし、大学側での研究成果を企業側の実際の開発プロジェクトに適用し、その結果を評価することによってモデルをさらに洗練させるという連携をとった。プロセスモデルの有効性の評価を行うためには、実際の生きた開発プロジェクトでの評価が必要であり、産学連携の研究として意味のあるものであった。

1.3 本研究の位置づけ

ソフトウェアプロセス国際ワークショップ (ISPW) に代表されるソフトウェアプロセスに関する国際会議では、主にソフトウェアプロセスの表現方法や記述方法、そして記述したソフトウェアプロセスをどう利用するかなどに関しての研究や議論が行われている。このようなソフトウェアプロセスに関する研究は、概ね以下のように分類される³⁾。

1. ソフトウェアプロセスを人間の活動や計算機上の作業を含めて、どう捉えるか、考えるか
2. ソフトウェアプロセスをどのように表現するか、また、そのための言語やグラフの提案や、それらの能力の比較
3. ソフトウェアプロセスをどう利用するか、特に(2)で得られたものを用いて開発作業をどう効率化するか

上記分類にあてはめると、本研究は、ソフトウェアプロセスの表現方法とその利用方法として、上記分類の2および3を対象とした研究として位置づけることができる。

1.4 論文の構成

本論文は下記に示す構成となっている。

1章 はじめに

1章では、本研究の背景と目的を述べた上で、ソフトウェアプロセス研究における本研究の位置づけを整理する。

2章 プロセスモデル

2章ではソフトウェアプロセスおよびプロセスモデルの定義を確認したあと、プロセス改善のためのプロセスモデルの役割を整理し、プロセスモデルが満足すべき要件を確認する。本章で述べるプロセスモデルの要件は、PReP モデルの有効性の検証に用いる。

3章 プロセスのモデル化方法

3章では、プロセスのモデル化方法をモデルの抽象度と記述方法から分析し、従来のプロセスモデルの問題点を特定する。その上で、プロセスの改善に適したプロセスのモデル化の方法を方向付ける。

4章 成果物の関連によるプロセスモデル PReP

4章では PReP モデルの概要と記述方法に関して述べる。PReP モデルは、開発のライフサイクルで作成される成果物間の関連構造を中心にプロセスをモデル化する方法であり、「構造モデル」、「工程モデル」、「作業モデル」という3つのモデルで構成される。まず、これら3つのモデルから構成される PReP モデルの概略と、このプロセスモデルの中心となる成果物の定義と種類について説明したあと、記述規則と記述手順を示す。

5章 PReP モデルの利用

5章では、プロセス改善への PReP モデルの利用に関して述べる。プロセス改善への利用は、プロセスの再利用を中心とした標準プロセスの定義、定義したプロセスの計画策定への利用、プロジェクトの実行と管理への適用、プロセス実績の評価と改善が考えられる。

6章 PReP モデルの有効性の考察

6章では、PReP モデルのプロセスのモデル化方法の有効性の検証を行った結果と、その考察を述べる。モデル化の有効性に関しては、「2.2 プロセス改善のためのプロセスモデルの要件」で述べた Humphrey の提案するプロセスモデルの要件と比較して考察を行った。モデル化の有効性の評価を行うために、前章で述べたプロセスモデルの記述が行われるプロジェクト計画の策定段階(プロジェクトのプロセスの定義、ライフサイクルの定義、スケジュールの定義)に対してモデルを適用しその結果を評価した。

7章 プロセス中心型環境への適用

プロセスモデルの利用方法やプロセスの利用を支援するための環境は、プロセスのモデル化方法に大きく依存する。その意味で、PReP モデルでは、プロセスモデルの記述方法を定義するだけで

はなく、プロセスの再利用モデルと、プロセス再利用を支援するための支援環境のアーキテクチャも定義する。プロセスモデルをもとに、プロセスの管理、実行改善などを支援するための開発環境は「プロセス中心型環境」と呼ばれており、7章では、PRePモデルを適用する際のプロセス中心型環境の基本アーキテクチャ案について述べる。

8章 プロセス再利用の導入モデル

プロセスモデルを再利用し、プロジェクト管理やプロセス改善を実際に実現するためには、「7章 プロセス中心型環境への適用」で示した支援環境に加えて、組織の状態や文化、そして制約に対して、どのような手順でプロセスモデルの利用の仕組みを導入するかについての指針が必要である。この導入の指針は、プロセスのモデル化方法の指針と大きく関係するものである。8章では、PRePモデルをプロセスモデルとして使用する場合のプロセスモデルの利用の仕組みの実現と、プロセス改善の実現のための導入モデルに関する基本的な案を述べる。

2 章 プロセスモデル

2 章ではソフトウェアプロセスおよびプロセスモデルの定義を確認したあと、プロセス改善のためのプロセスモデルの役割を整理し、プロセスモデルが満足すべき要件を確認する。本章で述べるプロセスモデルの要件は、PReP モデルの有効性の検証に用いる。

2.1 ソフトウェアプロセスとプロセスモデル

ソフトウェアの開発・保守・管理において、ソフトウェアプロセスに注目する傾向は、ビジネスプロセス・リエンジニアリング、ワークフロー管理²⁷⁾、VLSI(高集積度回路)設計、そしてソフトウェア工学などの分野で顕著に見られている²⁸⁾。このような分野において、ソフトウェアプロセスは、単に「プロセス」と略されて用いられる場合や、「ソフトウェア開発プロセス」、「作業工程」など、様々な呼ばれ方がされている。ソフトウェアプロセスという名称に対して、現時点で明確な同意が得られた定義がなされているとは言えない。IEEE の“A Compilation of IEEE Standard Computer Glossaries (610)”では、ソフトウェアプロセスは「与えられた目的のために実施される一連のステップ」と定義されているが²⁹⁾、古くは OS 等の研究分野でも用いられた言葉でもあり、そこでの先入観から誤解を生じることもある。ソフトウェアプロセス改善の分野での議論においても、プロジェクト管理や教育、供給者の選定など、ソフトウェアの開発から保守、管理に至るまでの幅広い見方をする場合と、特定の開発作業の内容に関する非常に限られた行為が対象となる場合がある。前章で示したソフトウェアプロセスに関する研究分野の分類においても、ソフトウェアプロセスを人間の活動や計算機上の作業を含めて、どう捉えるか自体が研究の対象ともなっている。

CMM では、ソフトウェアプロセスを、「ソフトウェアならびに関連する生産物を開発・保守するために使われる一連の活動、手法、プラクティス、変換」と定義している⁴⁾。本研究では、CMM での定義のように、ソフトウェアプロセスを、ソフトウェアを生産し維持するための活動、またそれらの管理方法の特徴づけるものとして、比較的幅の広い意味として捉えることとする。

プロセスモデルとは、このソフトウェアプロセスを外在化し記述したものである。一般に、ある対象をモデル化するには、その目的に応じて異なったモデルが得られる。プロセス改善を行うためにも同様に、プロセス改善に適したプロセスモデルが必要である³⁰⁾。プロセスをモデル化することにより、ソフトウェアプロセスを理解することができ、評価し問題を特定し、改善を行うことができる¹⁾。本研究では、プロセス改善を目的とした場合の最も有効なプロセスのモデル化の方法を得ることが目的である。

2.2 プロセス改善のためのプロセスモデルの要件

プロセス改善とは、組織の開発活動でのベストプラクティス(多くの事例でうまく行った方法)をプロセスとして定義し、定義したプロセスを個々の開発プロジェクトに適用し、プロジェクトの実施結果から適用したプロセスを評価し、さらに改善する継続的な活動である。プロセス改善を実現するためには、定義したプロセスモデル上で、プロセスの問題点を正確に特定できること、モデル化したプロセスを使用して開発活動を制御できることなどが必要になる。Humphrey らは、記述されたプロセスはプロセス改善における重要な要素であり、次に示す目的のもとに使用されるとしている^{30, 31)}。

1. プロセスに関する効果的なコミュニケーションの実現

プロセスの利用者、プロセス定義者、プロジェクト管理者、そして開発者間のプロセスの理解と実行に関する基礎を提供し、個人の開発活動を支援する

2. プロセスの再利用の促進

プロセスの定義には非常に大きなコストがかかる。一方で、個々の開発プロジェクトが自分達のプロセスを独自に開発するための時間および作業者の十分な割り当てはない場合が多く、再利用によるプロセス開発効率の向上が必要となる

3. プロセス改善の支援

正確で理解しやすく、拡張と再利用が可能なプロセスモデルは、プロセスの学習のための有効な手段であり、プロセス改善のための重要な要素となる

4. プロジェクトのプロセスの管理

プロジェクトを効果的に管理するためには、計画に関する明確な理解とプロジェクトの状況に対する正確な把握が要求される。プロセスモデルは、プロジェクトの状況を正確に把握するための基準と手段を提供する基礎となる

さらに Humphrey らは、上記で示したプロセスの利用目的を満足するプロセスモデルの要件として下記の3項目を定義している³¹⁾。

1. 現実に行われている、または行われるべき活動をモデル化できること
2. プロセスのモデル化と改善を行うために十分であるとともに柔軟で理解が容易であること
3. 必要とする粒度でのプロセスの洗練が可能であること

標準プロセスが形骸化する大きな原因としては、従来のプロセスモデルが、上記の要件を満足しておらず、そのために、Humphrey らの示したプロセス改善への利用が行われていないことが考えられる。すなわち、適切なプロセスのモデル化の方法こそが、プロセスの利用にとって重要な課題である²¹⁾。本研究では、現状のプロセスモデルに対して、モデル化方法の分類と整理に立ち返って再評価することで、Humphrey らの定義した要件を満足するプロセスモデルの要件を確認した。次章では、まずプロセスモデルを抽象度の観点から分析し、次にプロセスのモデル化方法の整理と分析から、プロセス改善に適したモデル化の方向を特定した結果について述べる。

3 章 プロセスのモデル化方法

3 章では、プロセスのモデル化方法をモデルの抽象度と記述方法から分析し、従来のプロセスモデルの問題点を特定する。その上で、プロセスの改善に適したプロセスのモデル化の方法を方向付ける。

3.1 プロセスモデルの抽象度による分類

プロセスの記述には、これまで数多くの方法が試みられてきた³²⁾。第6回ソフトウェアプロセスワークショップ(6th ISPW)では、18 種類のプロセスモデルが、記述モデルの主な目的、記述表現、記述要素、動作制御の方法、そして基礎となる言語による分類等によって比較検討された³³⁾。また、提案されるプロセスモデルの方法や得失を議論するための共通評価基準として、仮想的なプロジェクトが設計され提案された(この例題は、提案された会議名から「ISPW6 の問題」とも呼ばれる)³⁴⁾。このような、プロセスモデルの分類として、Humphreyらによるプロセスモデルの抽象度による分類がある。

Humphrey らは、ソフトウェアプロセスモデルの抽象度を、表 1 に示すように、「Universal」、
「Worldly」、「Atomic」の 3 段階のレベルで定義した³⁰⁾。Universal レベルはプロセスの概念を提供する抽象度の高いレベルである。Universal レベルのプロセスモデルとしては、“Waterfall model”^{24, 25)}、“Spiral model”²⁶⁾、“Iterative enhancement”³⁵⁾等があり、これらのモデルは、プロセスを概観するために有効である。

Universal レベルとは逆に、抽象度が低く特定の作業手順をモデル化するレベルを Atomic レベルと定義している。Atomic レベルは作業手順を詳細にモデル化し、標準化された手順を提供する。Atomic レベルには、Pascal, ADA といった手続き的プログラミング言語や、関数型、論理型言語や属性文法等の宣言的プログラミング言語を用いた記述方法。またベトリネット、状態遷移図、形式文法等を用いたモデル化の方法があり、作業手順の記述の他、作業の自動化のためのモデルとして有効である。また、標準プロセスの定義においても、ETVX³⁶⁾などの Atomic レベルのプロセスモデルは、局所的なプロセスの記述として標準作業手順の定義に有効であると考えられる。

一方、Universal レベルと Atomic レベルとの間の抽象度で、実際の開発活動をモデル化するレベルを Worldly レベルと定義している。Worldly レベルのプロセスモデルとしては、これまでに EPMs(Entity Process Models)³¹⁾が提案されている。Worldly レベルは、実際に行われている開発活動の実体をモデル化するレベルであり、開発プロジェクト全体で行われる活動の順序を示し、それぞれの活動の必要条件とその結果を定義する³⁰⁾。Worldly レベルのモデルは、開発プロジェ

外の計画を詳細に立て、進捗管理の基準となるモデルを提供するために有効であり、さらに、プロセス改善の検討や、プロセスの再利用にも適しており、「2.2 プロセス改善のためのプロセスモデルの要件」で示したプロセスモデルの要件を満足するには、Worldly レベルを中心としたプロセスモデルが必要であると予測できる。では、Worldly レベルを中心としたプロセスモデルの記述に適したモデル化の方法とはどのようなものだろうか。次節では、この問題を、プロセスの記述方法の視点から検討する。

抽象度	プロセスモデルの抽象度	モデルの利用	モデルの例
高	Universal レベル	プロセス概念の理解	Waterfall model, Spiral model 等のタスク観点でのモデル
	Worldly レベル	実際の開発活動の理解・改善・管理・再利用	EPMs
低	Atomic レベル	特定の作業手順の記述 作業の自動化	プログラミング言語, ETVX, ペトリネット等の、タスク観点でのモデル

表 1 ソフトウェアプロセスモデルの抽象度

3.2 プロセスの記述方法による分類

プロセスモデルの分類には、記述方法の分類として、表現形式による分類(自然言語、プログラミング言語、グラフなど)、基礎とする言語による分類(LISP, Ada, ペトリネット)などの観点があるが³³⁾、そもそも、ソフトウェアプロセスをどのような観点からモデル化するか分類方法として、タスク観点と成果物観点の分けがある^{3,33)}。ソフトウェアプロセスを、作業行為に着目してモデル化する方法がタスク観点によるモデル化であり、作成される作業成果物に着目したモデル化の方法が成果物観点でのモデル化方法である。Worldly レベルのプロセスモデルの記述として、タスク観点でのプロセスモデルと、成果物観点でのプロセスモデルの得失を、前節で述べたプロセスモデルの抽象度とモデル化方法との関係から比較し、プロセス改善に適したプロセスのモデル化の方法を方向付ける。

3.2.1 タスク観点のプロセスモデル

これまでに提案されているプロセスモデルの多くは、プロセスをタスクの観点から定義するものである。タスク観点によるプロセスモデルとは、意味を持つひとまとまりの作業を「タスク」として定義し、定義したタスク間の時間的順序関係を記述するものである。表 1 のモデルの例に示したように、Universal レベルのプロセスをモデル化する Waterfall model, Spiral model 等はタスク観点でのプロセスモデルである。また、Atomic レベルをモデル化するプログラミング言語や ETVX, およびペトリネット等を用いたモデルもタスク観点でのプロセスモデルである。

プロセスをモデル化する際、プロセスを、ある入力に対して出力を得る変換であると捉え、変換の処理とその順序関係をモデル化の観点に置くイメージは受け入れられやすい。ソフトウェア自体が処理の順序関係であると捉えるならば、ソフトウェアプロセスに関しても、処理の順序関係としてのタスクの観点でのモデル化は自然な発想であると言える。プロセスが固定的で、変動が無いものであれば、処理の時間的系列として表現したタスク観点によるプロセスモデルは、変動の無いモデルとして記述が容易となる。しかし、現実の開発活動における作業の関係は非常に複雑であり、しかも容易に、かつ頻繁に変動するものである。そのため、実際の活動をタスクの時間的な順序関係としてモデル化しようとすると、複数のタスクが関係し合った非常に複雑なモデルになってしまう。さらに、現実の開発活動においては、状況の変化に応じてタスクの順序の見直しが行われるため、定義されたプロセスモデルが、すぐに現実の活動と乖離してしまう^{30, 31)}。前節において、プロセス改善で使用されるプロセスモデルは、Worldly レベルを中心としたプロセスモデルが適切であると述べたが、Worldly レベルのプロセス、すなわち現実の開発活動では、タスクの観点によってプロセスを記述することは原理的には可能であっても、複雑さおよび現実の活動の変動性の理由から非常に困難なものになると考えられる。

記述上の複雑さとは別の、タスク観点による現実の作業のモデル化の弊害として、作業の分断化と固定化の問題がある。一般的に、生産的活動において作業者は、様々な状況の変化に対応して、柔軟に作業方法を修正し、状況に適応しようとする。このような、作業者の柔軟性は、開発の

効率と技術レベルを上げるうえで阻害してはならない要素である。1960年代の日本において、いわゆる「流れ作業」という生産管理方法に対して議論が行われた。そこでは、タスク(作業)の固定化は、製造プロセス全体の管理をしやすくする一方で、労働従事者の柔軟性を阻害し、その結果、プロセス全体の効率が、プロセスを構成する作業要素のうち、最も低い効率の作業に引き下げられてしまうことになるといった、作業分断化問題が議論された³⁷⁾。また、作業分断化と固定化は、技術を労働者から奪うものであり、基本的な人権にかかわる問題であることが警告された³⁷⁾。ソフトウェア開発は、製造業における組立を中心とした作業よりも、より柔軟性が求められ知識集約型の作業である。そのため、作業の分断化と固定化につながるタスク観点でのプロセスの記述と、作業者の管理は、記述が難しいだけでなく適切でないと言える。

このように、タスク観点でのプロセスモデルは、**Universal**のレベルのモデルとしてプロセスを記述する場合は、プロセスの概観を与えるわかりやすいモデルを提供する。また、**Atomic**のレベルの変動の無い、固定的、局所的な標準作業を定義する場合には効果的な方法である。しかし、**Worldly**レベルのプロセス、すなわち、現実の開発活動をモデル化する場合は、作業の時間的な順序関係が大きく変動するためにモデル化が難しくなる。さらに、作業の固定化は全体の作業効率の低下をも招く。そのため、**Worldly**レベルのプロセスのモデル化方法として、タスク観点でのプロセスモデルは適していないと言える。

3.2.2 成果物観点でのプロセスモデル

タスクと異なる観点によるプロセスのモデル化の方法として、タスクによって作成される成果物に着目した方法がある。成果物とは、「作業の成果としての実体であり、振る舞いを持つもの」である。例えば要件や仕様、実行コードの他に、検証済みのテスト環境なども成果物と考えられる。成果物観点でのプロセスモデルは、成果物の集合とその要素間の関連性として定義される。

タスク観点によるプロセスモデルは、タスク間の時間的順序関係の変動に対応して変化する。例えば、タスク A が終了した後タスク B を実行すると計画していたものが、ある時点で、タスク A と B を並行して進めることになったり、複数人で分担して作業を進めたりと、計画の変更は実際の開発作業では頻繁に起こりうるものである。それに対して、成果物観点によるモデルでは、タスクの変動に影響されない、要素間の関連によってプロセスを記述できると予測される。さらに、**Worldly**レベルでのプロセスモデルは、現実の開発の活動をモデル化するものであるため、現実の開発活動で取り扱われる成果物に着目することは、**Worldly**レベルのプロセスモデルを記述する方法として適していると予測できる^{30,31)}。しかし、既存のプロセスのモデル化の方法として、成果物観点でのプロセスモデルの提案は非常に少ない。Humphreyらは、Jackson System Development³⁸⁾でのエンティティ指向の考え方をプロセスモデルに適用した EPMs (Entity Process Models)³¹⁾を提案している。EPMs でのプロセスモデル例を図 1 に示す。EPMs は、図 1 に示すように、開発ライフサイクル中で作成される実体 (Entity) の状態遷移を詳細にモデル化するプロセスモデルである。しかしその一方で、一つひとつの成果物の状態遷移の記述に力点を置いているため、各成果物間の関連や制約を記述すると非常に複雑なモデルとなる。その結果、プロジェクト全体を示すプロセ

モデルとしては見通しが悪く、ライフサイクルを表現しにくいものとなり、プロジェクトの管理とプロセスの改善のためのプロセスモデルとしては冗長でかつ難解なモデルであると言える。

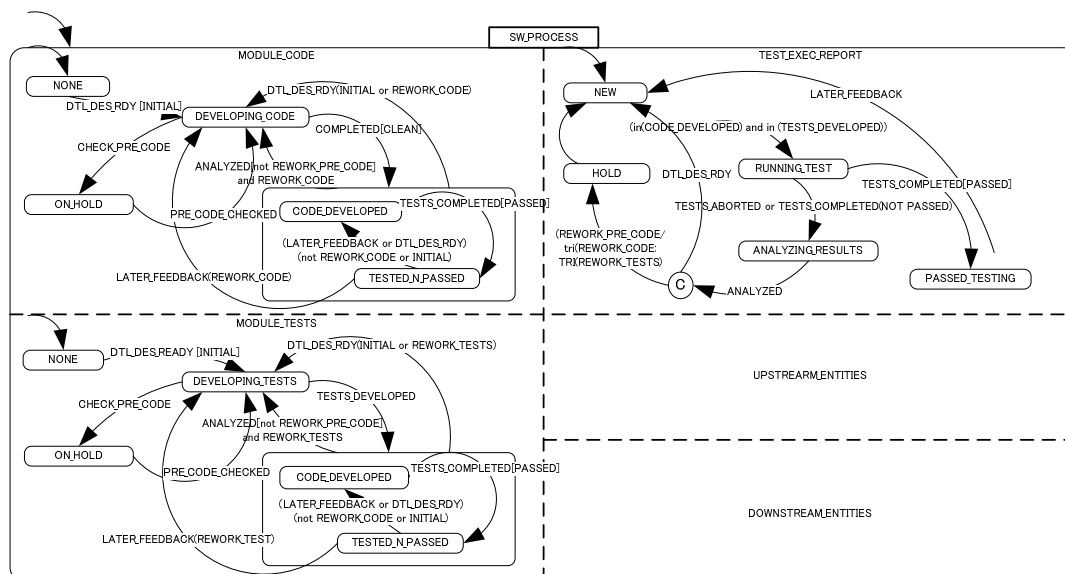


図 1 EPMs(Entity Process Models)によるプロセスの記述例

3.3 プロセス改善に適したプロセスモデルとは

前節で行った、プロセスのモデル化方法に対する抽象度と記述方法の分析から、プロセスの改善に適したプロセスモデルとして、Worldly レベルでのプロセスモデルを中心とした成果物観点でのプロセスモデルが有効であると予測できる。しかし、従来のプロセスモデルの中に、我々の目的に合致したモデル化方法が無かった。そのため、Worldly レベルのプロセスモデルを中心とした成果物観点でのプロセスモデルとして、PReP(Product Relationship Process)モデルを開発した。次章では、PReP モデルの構成および記述と利用の方法について述べ、プロセスモデルとしての有効性を、「2.2 プロセス改善のためのプロセスモデルの要件」で記述したプロセスモデルに対する要件と比較して評価を行う。

4 章 成果物の関連によるプロセスモデル

PReP

4 章では PReP モデルの概要と記述方法に関して述べる。PReP モデルは、開発のライフサイクルで作成される成果物間の関連構造を中心にプロセスをモデル化する方法であり、「構造モデル」、「工程モデル」、「作業モデル」という 3 つのモデルで構成される。まず、これら 3 つのモデルから構成される PReP モデルの概略と、このプロセスモデルの中心となる成果物の定義と種類について説明したあと、記述規則と記述手順を示す。

4.1 PReP モデルの概要

“PReP”とは「**Product Relationship Process**」の略であり、その名称が示すように、開発のライフサイクルで作成される成果物間の関連を描くことによってプロセスをモデル化する方法である。成果物間の関連によって開発プロセスの活動をモデル化することにより、プロセス改善への効果的な利用が可能なプロセスモデルを提供することを目的としている。対象は、ソフトウェアの開発プロジェクトであり、小規模(4~5名)から中・大規模(約 50~100 名程度)、開発期間が 1~3 ヶ月から約 1 年程度の、一般的なソフトウェア開発および組み込み系ソフトウェア開発を想定しているが、より広い適用も可能であると考えられる。

4.1.1 PReP モデルの特徴

成果物観点によるプロセスモデルでは、一般に、成果物の集合とその要素間の関連性によってプロセスをモデル化する。PReP モデルは、開発プロジェクトにおいて作業担当者に割り当てられ、プロジェクトとして管理する成果物を成果物の集合とし、各成果物間の入出力の関係を成果物の要素間の関連性として記述するモデルである。また、プロセスモデルの抽象度としては、Worldly レベルの抽象度のプロセスの記述方法を特徴とし、Universal レベルと Atomic レベルのプロセスモデルを統合している。このようなモデル化の方法に基づく PReP モデルは以下の特徴を持つ。

PReP モデルの特徴 1: 成果物の関連による静的な構造モデル

PReP モデルの基本概念は、図 2 に示すように、ある成果物(図 2 での成果物 B)を作成するために必要となる成果物(図 2 での成果物 A)を特定し、それらの成果物間の入出力の関連を記述するというものである。開発ライフサイクルをモデル化する場合は、最終成果物を始点として、そこから必要となる成果物の系列を逆に遡ることによってプロセスモデルを得る。

PReP モデルは、他のプロセスモデルと違い、成果物の入出力の関連構造のみをモデル化したものである。そして、タスク観点によるプロセスモデルに現れる時間的な順序関係の概念を成果物に内包することによって、時間的な順序関係が現れない静的な関連構造のモデルを得ることができる。例えば、図 2 に示す PReP モデルでの矢印線は、「成果物 B の作成に成果物 A が入力として必要である」という静的な関連構造を示している、一方、他のプロセスモデルの場合、矢印線は一般的に時間的な順序関係を示す。例えば図 3 に示すように、一般的なタスク観点でのモデル(図 3 の a)では、矢印線はタスク i を行った後でタスク ii を行うことを示し、データフロー図(図 3 の b)では、処理 iii の結果、データ iv が生成されることを示す。さらに EPMs(Entity Process Models)31(図 3 の c)では、実体 v の状態が実体 vi に時間的に遷移することを示している。

PReP モデルの特徴 2: 現実の活動をモデル化

PReP モデルが、成果物の関連の静的構造をモデル化していることは、現実の活動、すなわち Worldly レベルをモデル化する方法として重要な特長である。例えば、図 3 の a に示したタスクによるプロセスモデルの場合、タスク i が完全に終了することを待ってタスク ii が開始されることを示している。しかし、現実の開発活動では、このようなウォーターフォールでタスクが進行することは少なく、状況に応じてタスク i が終了しない状態でタスク ii が開始される場合や、両タスクが並行して進められる場合、もしくは、タスク i の実効が始まっていない状態でタスク ii が着手されるといった場合もある。このように現実の開発活動をタスクの観点で正確にモデル化しようとするとなかなか複雑なものになってしまう。そのため、一般的にタスクによるプロセスモデルでは、現実の開発の状況の複雑さに影響されない、すなわち抽象度の高い Universal レベルでのプロセスモデルか、もしくは、時間的な順序関係の変動が無い、固定的で局所的な標準作業工程のプロセスモデルの記述になってしまう。そのために、タスク観点では、Worldly レベルのプロセスモデルを得ることが困難となるのである。

成果物の入出力の関連構造のモデルでは、入力側および出力側の成果物を適切な粒度で定義することにより、現実の開発プロセスをモデル化することが可能である。また逆に、現実の開発プロセスを適切にモデル化できた状態で、プロセスモデルを構成する成果物が適切な粒度によって定義されたと言える。そして、プロジェクトの管理では定義された成果物の粒度に合わせて、見積もり、リソースの割り当て、スケジュールを定義することによって、適切な管理粒度によって、開発の計画と、進捗の追跡を行うことができる。さらに、プロセスの実行結果を、管理粒度と対応した成果物の粒度で評価することができ、プロセス上のボトルネックを成果物の粒度で知ることが可能となる。

PReP モデルの特徴 3: 高い再利用性

ある成果物を作成するために必要となる成果物は、出力側成果物の作成に適応する技術によってきまる。そのため、PReP モデルで記述される成果物入出力関係の系列は、開発プロジェクト全体に適用される技術によって決定される。また、プロセスモデルを構成する成果物の集合は、作業担当者への割り当て粒度、すなわち、プロジェクトの管理粒度から決まる。すなわち、PReP モデルは、適応技術と管理方法によって一意に構造が決まる。そのため、同じ技術と同じ管理方法が適用される開発プロジェクト間では、定義したプロセスの再利用性が高いと考えられる。

PReP モデルの特徴 4: 統計的生産管理手法の適用

CMM および CMMI の基本概念は、日本の製造業における統計的生産管理手法を基にしている。製造工程での固定された生産設備に対応した固定的なプロセスに対して、ソフトウェア開発では、人による知識集約型の作業が中心となる流動的なプロセスとなり、生産設備のようにプロセスを固定してモデル化することが難しい。生産設備では、各工程の順序関係が固定されているため、プロセスモデルに変動が無く管理しやすい。さらに、独立して定義した各工程を単位として、プロセス効率を評価することも容易である。しかし、生産設備と同様のモデルをソフトウェア開発に持ち込もうとすると、実際の開発は各タスクの関係が流動的に関係しあって進行しているため、時間的順序関係によるモデル化は原理的には可能であっても、非常に複雑なものとなり、管理および改善に利用可能なモデルを得ることが難しくなってしまう。一方、成果物の入出力の関連構造では、前述のように静的な構造のモデルを得ることができ、製造工程における固定された生産設備に相当するプロセスモデルを定義することが可能である。そのため、製造業における統計的生産管理手法の適用が可能となると考えられる。一般的にプロセスをモデル化する場合、ソフトウェア開発プロセスを製造工程の概念、または、プログラムの概念として把握し、処理の系列としてプロセスをモデル化する傾向がある。しかし、ソフトウェア開発の場合、各処理の関係の流動性および複雑性から、工程の時間的順序関係ではなく、成果物の入出力関連による静的な構造によってプロセスをモデル化することによって、統計的生産管理モデルを実際の開発の管理と改善に適応することが可能となるのである。

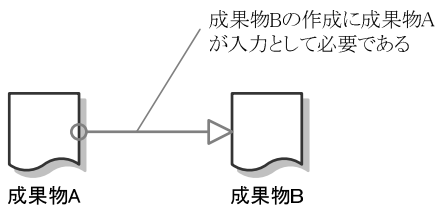


図 2 成果物の関連によるモデルの基本構成

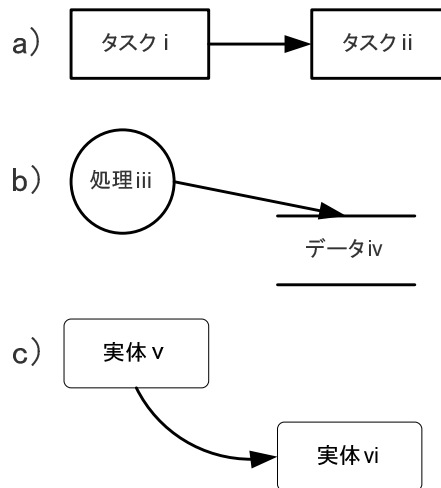


図 3 時間的順序関係に基づくプロセスモデルの例

4.1.2 PReP モデルの構成

PReP モデルは、図 4 に示すように、構造モデル、工程モデル、そして作業モデルの 3 つのモデルで構成される。構造モデルは、PReP モデルの特徴である成果物間の入出力の静的な関連をモデル化するものである。また、工程モデルは、構造モデルに、工程や反復、マイルストーンと言った開発ライフサイクルを定義するための情報を付加したものである。そして、構造モデルおよび工程モデルを構成する個々の成果物には、成果物を作成するための標準作業手順の情報として作業モデルが関連付けられる。

それぞれのモデルをプロセスモデルの抽象度で定義すると、構造モデルは開発ライフサイクル中で実際に作成される成果物の関連によるモデルであることから **Worldly** レベルのモデルであると言える。構造モデルは、最終成果物を作成するために必要となる入力成果物の関連が構造化されたモデルである。しかし実際の開発ライフサイクルでは、主にプロジェクトのリスクを低減する目的から、あらかじめ設定された完成度で成果物が一旦リリースされたり、設定した完成度を確認するまで指定された工程内のプロセスが反復されたりする。このような実際の開発ライフサイクルをモデル化するものが工程モデルである。さらに工程モデルでは各工程を定義するために、例えば、要件定義フェーズ、概要設計フェーズと言った、開発ライフサイクルでの各工程を指し示す名称が与えられる。これらの工程の概念は、開発ライフサイクルでのプロセスの概観を与える意味から、**Universal** レベルの抽象度のプロセスモデルであると言える。また、各成果物に関連付けられる作業モデルは、対応する作業成果物の標準作業工程のモデルを提供することから、**Atomic** レベルのプロセスモデルであると言える。

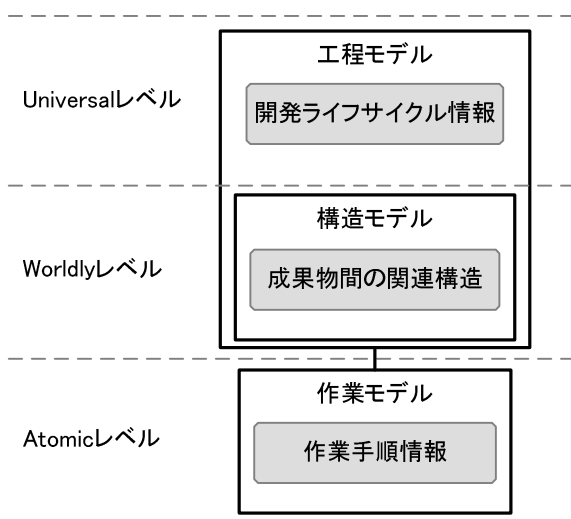


図 4 PReP モデルの構成

PREP モデルでは、構造モデルおよび工程モデルの記述方法に対して、独自の記述方法を提供する。一方、作業モデルは、Atomic レベルでの標準作業工程をモデル化する方法として、プロセスの利用目的と環境に適した記述方法であれば良く、その選択に対する制約を設けていない。例えば、自然言語を中心に記述したい場合は、ETVX などが考えられ、グラフ表現を行いたい場合は、IDEF0³⁹および一般的なフロー図等が考えられる。あるいは、プロセス中心型設計環境への適用を考えるのであれば、XML による形式記述なども考えられる。以下では、PREP モデルにおいて記述方法を特定している構造モデルと工程モデルの記述方法(Notation)と、記述手順に関して述べる。

4.2 構造モデルとその記述方法

構造モデルは、最終成果物を作成するために必要となる成果物の入力に関連を記述したモデルである。図 5 に構造モデルで記述したエンジニアリングプロセスの例を示す。例では、右側に記述された3つの成果物(統合成果物, 検証関連成果物 A および B)が最終成果物であり、これら成果物を作成するために必要となる成果物の入出力の関連が示されている。成果物の関連の構造は、最終成果物を作成するために適用する技術と、プロジェクトの管理方法から決まるものである。そのため、構造モデルは、組織またはプロジェクトが用いる技術および管理方法の改善への利用が考えられる。

構造モデルを構成する各成果物には、その成果物を作成するための標準作業工程の情報が結びつく。標準作業工程は作業モデルとして記述される。作業モデルはひとつの成果物に複数定義することが可能である。つまり、ある成果物を作成するためのコストや得られる成果物の品質などに応じて、作成方法が幾通りか存在する。実際のプロジェクトでは、プロジェクトの目的や制約に応じて適切な作業モデルが選択されることになる。成果物に適用可能な作業モデルの実績を記録、測定することによって、各作業モデルの効率や品質を評価することができ、成果物作成方法に関する知識の共有を行うことができる。

成果物観点でのモデルの特徴として、開発プロジェクトが適用する技術の変更や開発規模の大きな変更等がない限り、成果物間の関連が開発ライフサイクルを通して変動し無いことがあげられる。そのため、定義したプロセスの再利用性が高く、組織の標準プロセスの定義に適していると考えられる。一方、個々の成果物を作成するための手順として関連付けられる作業モデルには、プロジェクト毎に異なった方法を選択することが可能であり柔軟性を持つ。PReP モデルは、このように、標準化と柔軟性という特長を持つモデルであるとも言える。なお、入力として必要な成果物に変更が生じるような作業モデルの変更は、成果物の関連の構造自体に変更を及ぼす規模の適用技術の変更であり、構造モデルが変更され、新しく定義されることになる。

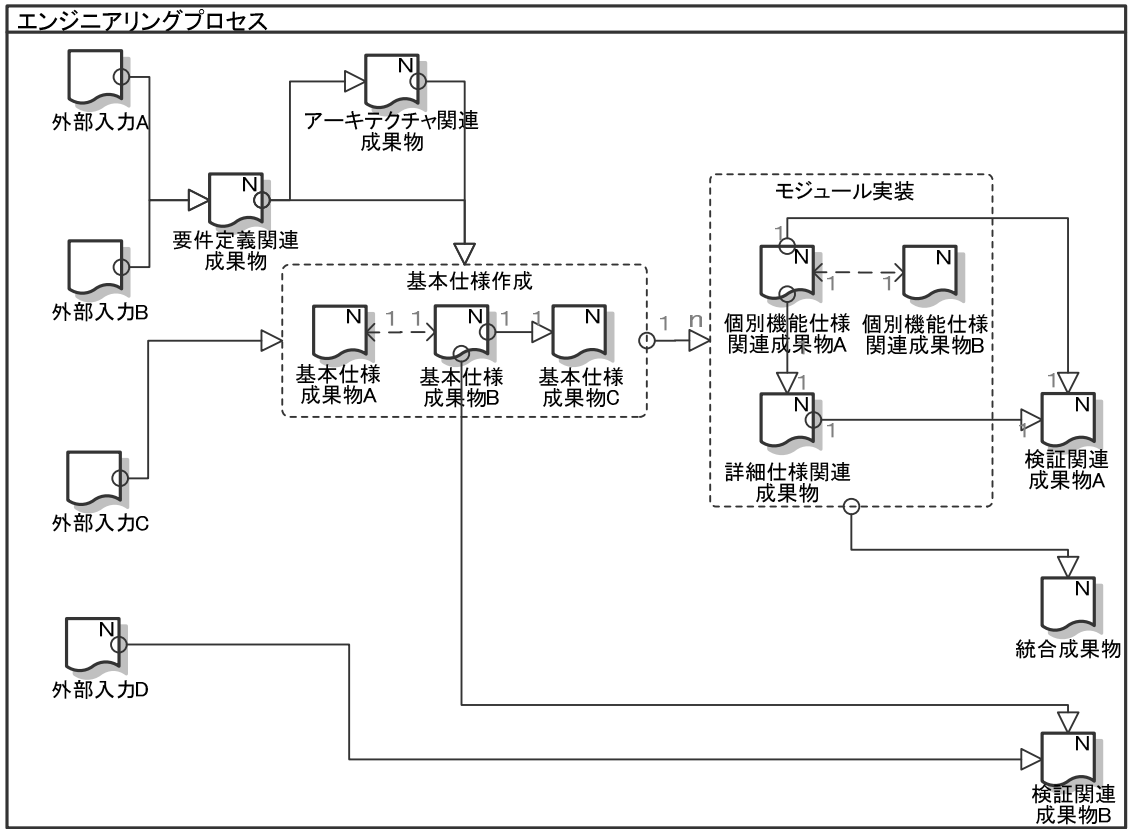


図 5 構造モデルで記述したエンジニアリングプロセスの例

4.2.1 構造モデルの構成

図 6 に構造モデルの構成を示す。図 6 に示すように構造モデルは、複数の成果物の入力関連で定義され、また各成果物にはひとつ以上の作業モデルが関連付く構成を持つ。構造モデル、成果物、作業モデルは図 6 に示す属性を持ち、それぞれの属性の定義を表 2、表 3、表 4 に示す。

構造モデル全体が保有する属性

構造モデル全体としては、管理用 ID、構造モデルの名称とその概要、包含する成果物、プロジェクトの規模見積の属性を持つ。プロジェクトの規模見積は、COCOMO II⁴⁰⁾やユースケースポイント法⁴¹⁾などによる統計的類推手法によるプロジェクトの粒度での見積もりモデル(成果物単位での積み上げ式の見積もりではなく、プロジェクト全体をトップダウンに見積もるモデル)に基づく見積もり結果を格納する属性である。この属性はプロジェクトのプロファイル(特徴となる属性とその値)を表すものであり、プロジェクトの計画、進捗管理そしてプロセス実績の評価の際、成果物単位での見積りと共に利用される。

見積もりに関する属性

成果物単位での見積りは、成果物と作業モデルに割り振られた属性によって実現する。まず成果物は規模の見積り値をもつ。一方で、成果物に結びついた作業モデルは工数・期間・コスト・品質の見積り式を持つ。成果物の持つ規模の見積り値が作業モデルの見積り式に代入されることによって工数・期間・コスト・品質の見積りの値が決定され、その値は成果物が保持するとしている。このことによって、作業モデルを変更することによって同じ規模の成果物を作成する場合の工数・期間・コスト・品質の見積りの値が変わり、プロジェクトの目的および制約に適した作業モデルを選択することができる。

その他の成果物属性

成果物が持つその他の属性として、管理用 ID、成果物名称と概要がある。また、情報の多重度に関する分類属性として、通常成果物または並列成果物の属性を持つ(4.2.3 参照)。さらに多重度で定義した情報単位の属性として、例えば、ユースケース単位、機能単位、モジュール単位などのデータの型が定義される。

成果物を作成するための作業モデルが複数あるときには、複数の作業モデルが成果物に結び付けられる。この成果物と作業モデルとのリンク情報は、後述するように作業モデル側が持つ。成果物側では、工数等の標準的な見積り値を定義するため、標準作業モデル属性として、標準工程とする作業モデルがひとつ定義される。また、プロセスを再利用する際の利用知識として、過去の適用事例からの情報が成果物ごとに管理される。利用知識としては、例えば、ベストプラクティスとして参考となる成果物事例や成果物特有のリスクに関する知識などが考えられる。再利用に関しては、「5.1 PRoP モデルによるプロセスの再利用」で詳しく述べる。

その他の作業モデル属性

作業モデルが持つその他の属性としては、管理用 ID, 作業モデルの名称と概要を持つ。そして、標準作業工程に関する情報を持つ。標準作業工程の情報は、先に述べたように、ETVX や IDEF0 および一般的なフロー図等の他、プロセス中心型環境を構築する際には、プログラム言語や XML による記述など、標準的な作業を伝達するための最適な記述形態がとられる。

成果物と作業モデルとのリンク情報は、成果物側で利用可能な作業モデルを管理するのではなく、作業モデル側がリンク先の成果物を特定する。このことによって、個々の作業モデルを独立して管理し、再利用することが可能となる。さらに、各作業モデルは、その作業を遂行するために必要となるリソース要件を持つ。構造モデルをもとに計画を作成する場合に、定義された要件を満足するリソースが各作業に割り当てられる。また、作業モデルを再利用する際の利用知識として、顕在化したリスクや作業実施ノウハウなどの情報が管理される。再利用に関しては、「5.1 PReP モデルによるプロセスの再利用」で詳しく述べる。

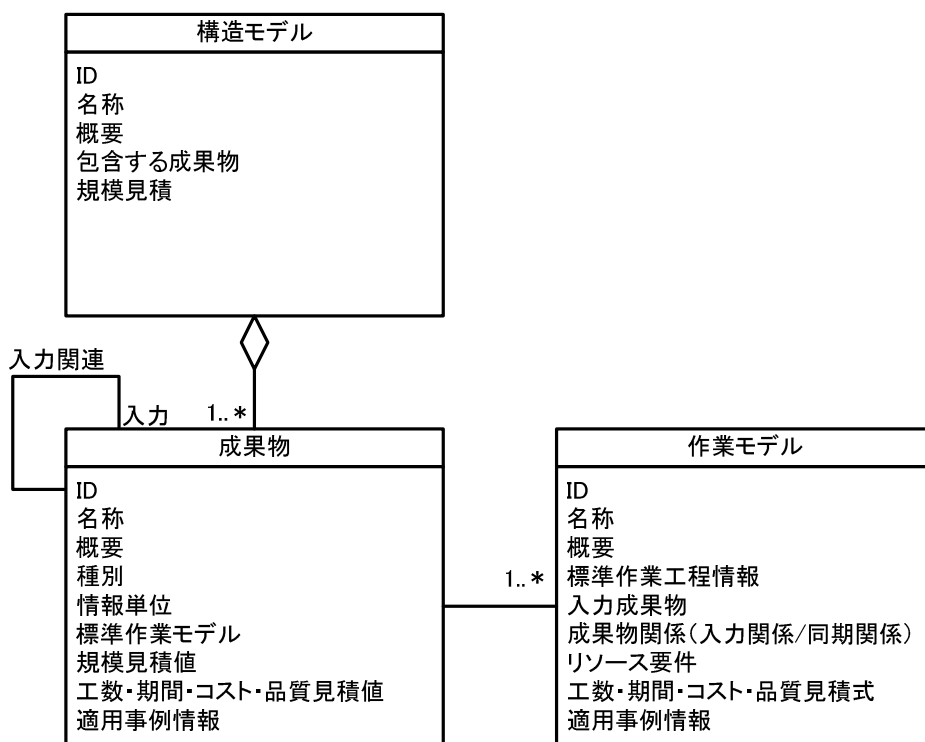


図 6 構造モデルの構成

属性	説明
ID	管理用 ID
名称	構造モデルの名称
概要	構造モデルの概要
包含する成果物	モデルが有する成果物
規模見積	プロジェクトの規模見積

表 2 構造モデルの持つ属性

属性	説明
ID	管理用 ID
名称	成果物名称
概要	成果物概要
種別	通常成果物／並列成果物
情報単位	情報単位の型
標準作業モデル	組織の標準とされる作業モデル ID
サイズ見積値	成果物の規模見積もり
工数・期間・コスト・品質見積値	成果物規模を入力として作業モデルの見積もり式から算出される見積値
適用事例情報	成果物に関する事例や知識

表 3 成果物の持つ属性

属性	説明
ID	管理用 ID
名称	作業モデル名称
概要	作業モデル概要
標準作業工程情報	標準作業工程情報および情報へのリンク
入力成果物	作業モデルに必要な成果物
成果物関連	成果物の関連情報
リソース要件	作業モデルに必要なリソースの要件(能力)
工数・期間・コスト・品質見積式	成果物規模を入力として、工数、品質等を算出するための見積式
適用事例情報	作業モデルを再利用する際の適用事例や知識

表 4 作業モデルの持つ属性

4.2.2 PReP モデルでの「成果物」の定義

成果物観点でのモデルでは、開発プロセスから適切な成果物を特定することが重要である⁸⁹⁾。一般的に、ソフトウェアの開発ライフサイクルで作成される成果物の種類には、表 5 に示すように、最終成果物である実装セット、導入・運用セット、さらに企画書や要件定義書などのように、開発プロジェクトの中および外に対して情報を共有するために作成する成果物、そして、評価・検証用成果物がある⁴²⁾。また、計画書やソースコードと言った成果物の他に、表 5 に示すように、検証用の環境や、構成管理環境なども開発プロジェクトの中で作成される成果物である。

PReP モデルでは、成果物の定義を「各作業者に割り当てられた作業の成果としての実体であり、かつ振る舞いを持つもの」とした。この定義によって、プロセスモデルを構成する成果物の粒度が、作業者に割り当てた成果物の粒度と一致することによってプロジェクトの管理粒度と対応することになる。プロセスモデルで扱う成果物の粒度をプロジェクトの管理粒度に対応させることは、実際に行われている開発活動の実体をモデル化する方法として適しているだけでなく、プロジェクトの管理粒度に合致した計画の作成や、定義したプロセスに従った開発活動の実行と進捗の監視と制御にも適していると考えられる。例えば、能力の高い作業者が複数の成果物からなるプロセスを引き受ける場合には、その作業者からの最終の成果物を管理することになり、通常の作業者が分担して作業を進める場合には、個々の作業者が受け持つ成果物を細かに管理することになる。また、プロジェクトが階層構造をもって管理される場合には、各管理レベルのプロセスモデルで定義される成果物は、それぞれの管理レベルで管理される成果物によってモデル化されることになる。なお、ここで定義される作業者は個人とは限らず、複数の作業者とその責任者から構成される場合もある。さらにプロジェクト外から成果物が入力される場合は、組織やプロジェクトが作業者となる。

分類	目的	例
情報共有のための成果物	「何を作るか」の情報共有	・企画書 ・要件定義書
	「どう作るか」の情報共有	・設計書 ・アーキテクチャ説明書
	プロジェクトの進め方の情報共有	・プロジェクト計画書 ・構成管理計画書 ・品質管理計画書
実装・設計成果物	実装セット	・各種設計書, 仕様書 ・ソースコードベースライン ・設計環境 ・コンパイル時関連ファイル ・コンパイル環境 ・コンポーネント実行ファイル
導入成果物	導入・運用セット	・統合製品実行ファイルベースライン ・実行時関連ファイル ・運用用成果物
評価・検証成果物	Verification 用の成果物	・単体評価・検証仕様書 ・単体評価・検証環境 ・単体評価・検証結果
	Validation 用の成果物	・統合評価・検証仕様書 ・統合評価・検証環境 ・統合評価・検証結果

表 5 代表的な成果物の例

4.2.3 成果物の表記

PReP モデルでは, 成果物が持つ情報の多重度により「通常成果物」と「並列成果物」の 2 通りの種別を定義する. さらに, 定義するプロセスの外部からの成果物, 工程モデルで用いられるマイルストーン通過判断となる成果物がある. これらの成果物は, PReP モデルのグラフ表現としての表記では, 成果物の名称を伴ったアイコンを用いた表記を基本とする. 各成果物の定義と表記を以下に述べる.

通常成果物

通常成果物とは, 1 つの情報単位から構成される成果物である. PReP モデルの表記では, 通常成果物は, 図 7 の a) に示すように, 成果物の名称を伴ったアイコンを用いて表現される.

並列成果物

並列成果物は, 同種の独立した複数の情報から構成される成果物であり, 例えば機能単位, モジュール単位, ユースケース単位など, 情報の単位ごとに独立して扱うことのできる情報の集合である. このような情報の単位は, 成果物の種類(型)として, 表 3 で示した成果物の属性の中の情報単位の属性として定義される. 並列成果物は, それぞれ個別の情報単位ごとに並行開発を行うこと

が可能であり、また、一部の独立した情報ができあがった時点で、それを入力とする成果物の開発に着手できる。並列成果物は、図 7 の d) に示すように、通常成果物の表記に「N」の表記を加えて表現される。

外部成果物

定義するプロセスの外部プロセスから入力される成果物がある。成果物の持つ属性としては、表 3 で定義した属性のうち、適用作業モデル、サイズ見積値、工数・期間・コスト・品質見積値が不要となる。PReP モデルの表記では図 7 の c) に示すように、成果物を示すアイコンの図形枠を点線表記とする。

マイルストーン成果物

後述の工程モデルにおいて、マイルストーン通過判断となる成果物として「マイルストーン成果物」が定義される、PReP モデルの表記では図 7 の d) に示すように、「M」の表記を加える。詳しくは、後述の「4.4.2 工程モデルの記述手順と規則」で述べる。

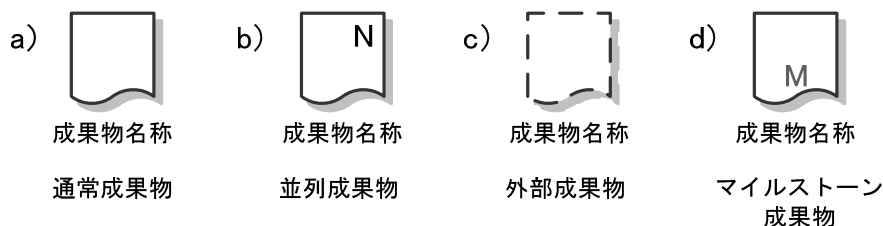


図 7 PReP モデルでの成果物の表記

4.2.4 成果物の関連の表記

成果物の入出力の関連には、図 8 に示すように、「入力関係」、「同期関係」、「入力と同期の合成関係」の 3 種類の関連を定義している。それぞれの定義は以下のとおりである。

入力関係

始点にある成果物の情報に基づいて終点の成果物が作成される関係。終点にある成果物が始点にある成果物の情報を必要とする関係にあることを示しており、図 8 の a) に示すような実線の矢印で表す。

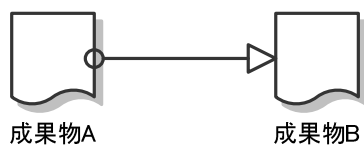
同期関係

始点にある成果物と、終点にある成果物とが相互に情報を必要とする関係。この関係にある成果物は、お互いに情報を交換しながら並行に作成される。図 8 の b) に示すような双方向の破線矢印で表す。同期関連は、図 9 の a) に示すように、成果物 A と成果物 B とが、成果物 C への入力となる条件下で、成果物 A と成果物 B とが協調して作成され、かつ入出力の関係には無い場合に、成果物 A と B との関係の定義に用いられる。

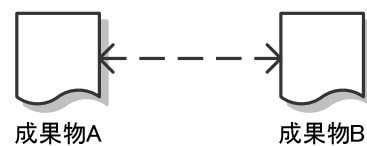
入力と同期の合成関係

始点にある成果物の情報に基づいて終点の成果物が作成され、かつそれぞれが同期を取って作成される関係。始点にある成果物が完成しなくとも終点にある成果物の作成を開始できるが、始点にある成果物が完成しないと終点にある成果物も完成しないことを示す。図 8 の c) に示すような破線の矢印で表す。入力と同期の合成関連は、図 9 の b) に示すように、成果物 A と成果物 B とが、成果物 C への入力となる条件下で、かつ成果物 A と成果物 B とが入力関連である場合、ともに成果物 C への入力となるために協調して作成される、この場合に、成果物 A と B との関係の定義に用いられる。

a) 入力関連



b) 同期関連

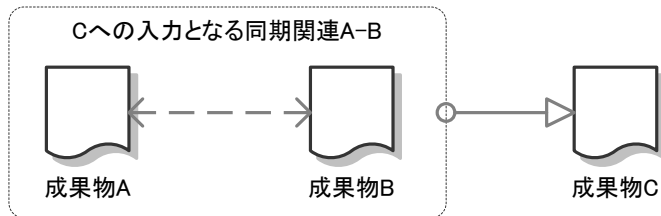


c) 入力と同期の合成関連



図 8 PReP モデルでの成果物の関連の表記

a) 同期関連の利用



b) 入力と同期の合成関連の利用

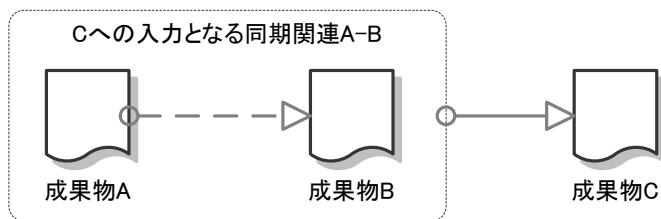


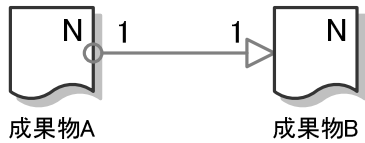
図 9 同期関連および入力と同期の合成関連の利用

4.2.5 関連表記での多重度の修飾子

成果物の関連の表記には, 図 10 に示すように, 「1」または「n」の修飾を付けることができる. 「1」の修飾子は, 図 10 の a) に示すように, 入力側の並列成果物を構成する 1 単位情報が出力側の並列成果物の 1 単位情報への入力となることを示す. この場合, 入力側成果物の 1 情報単位ができた時点で, 出力側成果物の作成を開始することができ, 分散開発の可能性を示している. 成果物間の関連と多重度の組み合わせによる作成開始の制約に関しては, 「4.5

スケジュールの導出」で詳しく述べる。また、「n」の修飾子は、図 10 の b) に示すように、成果物、プロセス要素グループ、集約プロセス等が複数あり、その数がプロセスモデルを実際のプロジェクトに適応する際に決定する場合に、構造モデル上で多重度を表す場合にも用いられる。

- a) 並列成果物Aを構成する1単位情報が並列成果物Bの1単位情報への入力となる



- b) 成果物、プロセス要素グループ、集約プロセス等が複数あり、その数が実際のプロジェクト(インスタンス)で決定する場合、構造モデル(クラスに相当)では「n」の表記で多重度を表す

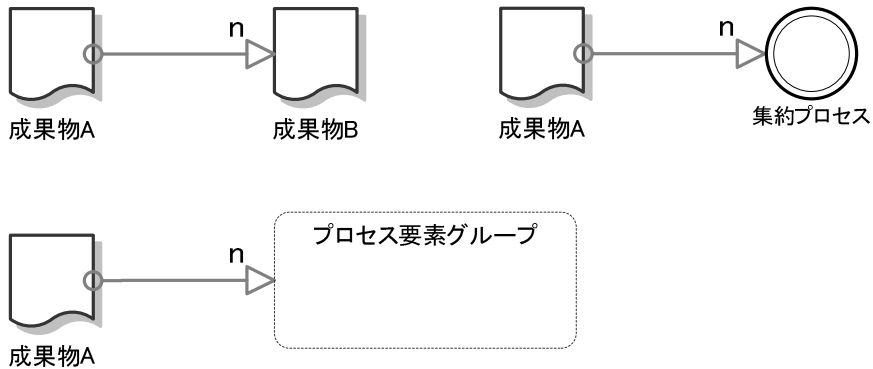


図 10 多重度の修飾子

4.2.6 その他の表記要素

PReP モデルでのその他の表記要素として、図 11 に示すように「外部プロセス」、「集約プロセス」、「プロセス要素グループ」がある。

外部プロセス(図 11 の a)

定義するプロセスの外部のプロセスを示す。開発プロセスに対する企画のプロセスや、進捗管理から出された工数が分析プロセスへ渡されるなどの場合に使用される。

集約プロセス(図 11 の b)

主に、プロジェクトが階層化されて管理される場合に、下位のプロセスを表記する場合に用いされる。モデルを記述する際の紙面の制約および視認性、理解性のための表記要素である。また、プロセス間の関係構造(プロセスアーキテクチャ)の記述にも用いられる(5.2 参照)。

プロセス要素グループ(図 11 の c)

定義したプロセスに対して、プロセス理解のために意味のある一連のプロセス系列に名前をつけるものであり、工程モデルの工程名とともに、Universal レベルでのプロセスモデルの概念を与える。

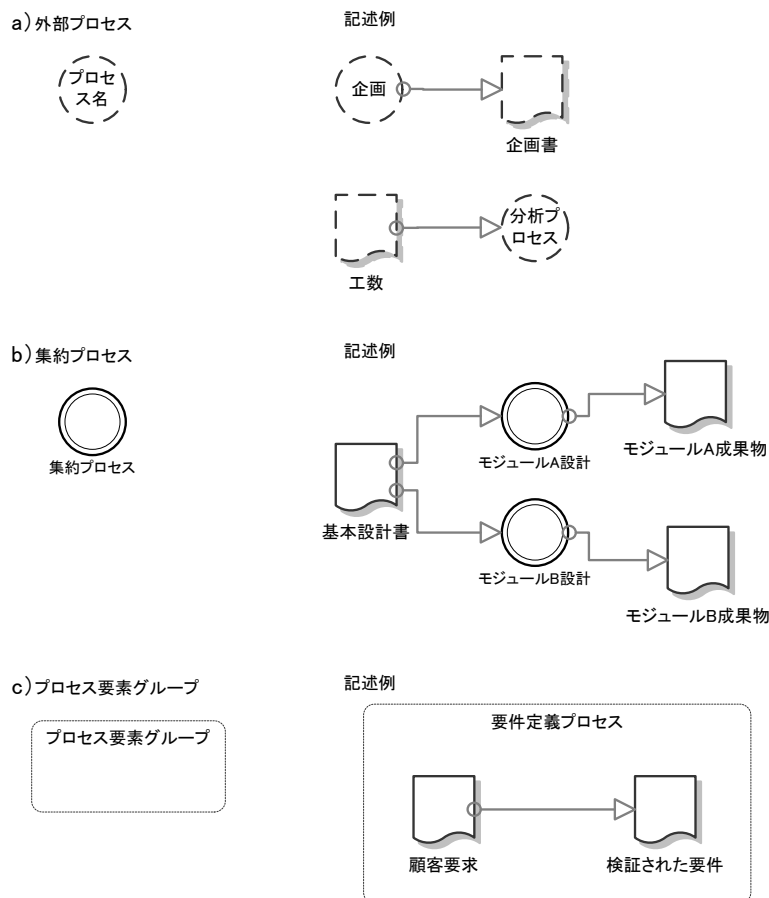


図 11 その他の表記要素

4.3 構造モデルの記述手順

前節で示した成果物と、成果物間の関連の表記による、構造モデルの記述手順を以下に示す。なお、標準プロセスを定義する場合は、下記に示した構造モデルの定義手順に加えて、プロセスアーキテクチャの定義などの主に前段階での作業が必要となる。詳しくは、「5.2 標準プロセス定義への利用」に述べる。

手順 1: ソフトウェアプロセスの範囲の定義

プロセスモデルの記述にあたって、記述するソフトウェアプロセスの範囲を定義する必要がある。ソフトウェアプロセスの範囲を決定するためには、対象とする開発プロジェクトのライフサイクルを定義し、その外部とのインタフェースを定義する必要がある。PReP モデルでは、例えば、図 12 に示すように外部プロセス(企画)とそこからの入力(企画書)が定義され、最終成果物が後工程へ出力される。

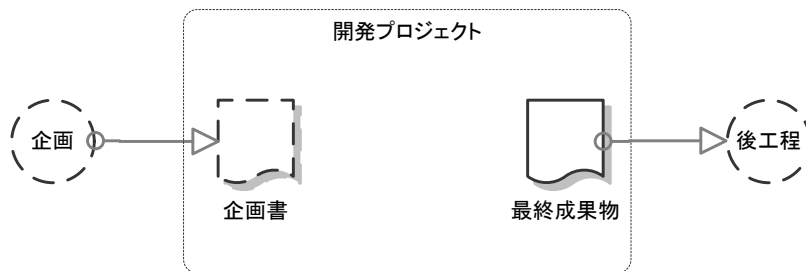


図 12 ソフトウェアプロセスの範囲の記述例

手順 2: 最終成果物の特定

ソフトウェアプロセスの範囲の定義と合わせて、最終成果物を特定し定義する。実際の開発プロジェクトでは、この最終成果物がいまいである場合も多く、また、あらためて納入先との合意を得る作業も必要になる場合もある。最終成果物が何かを特定する作業は、ソフトウェアプロセスの範囲の定義と合わせて重要な作業である。

手順 3: プロジェクトとして管理すべき成果物の特定

最終成果物を作るために必要な成果物であり、さらに開発プロジェクトとして管理すべき成果物を特定する。作業方法としては、まず、現状の開発プロジェクトで管理している成果物を洗い出し、次の成果物間の関連を特定する手順と合わせて精緻化する。定義する成果物の粒度は、開発プロジェクトとして担当者および担当グループに割り振り、その進捗を管理する粒度であり、開発プロジェクトの規模やリスクによって決まる管理の方法に依存する。最適な粒度の成果物の定義は、開発プロジェクトの管理方法の最適化と一致する。また、この段階で、構造モデルで記述する個々の成果物の名称等の属性を定義し、プロセスを利用する関係者が同一の成果物を指し示すことができるようにすることも重要である。

手順 4: 成果物間の関連を特定し記述する

手順 3 で特定された成果物に対して、成果物間の入出力の関連を記述する。この作業は、手順 3 と並行して行われる場合が多い。成果物間の関連の記述では、特に、図 9 で示した同期関連、および入力と同期の合成関連の定義方法を工程モデルでの工程内のイテレーションと区別することがひとつのポイントとなる。例えば、図 13 に示すように、成果物 A と B が行きつ戻りつ作成される状況で、成果物 C への入力が成果物 B のみである場合は、成果物 A と B は同期関係ではなく、図 13 の a) に示すように、工程内のイテレーションとして記述する。一方、成果物 A と B がともに成果物 C への入力となる場合は、成果物 A と B は同期関係として記述される。同様のことが、入力と同期の合成関連にもあてはまる。

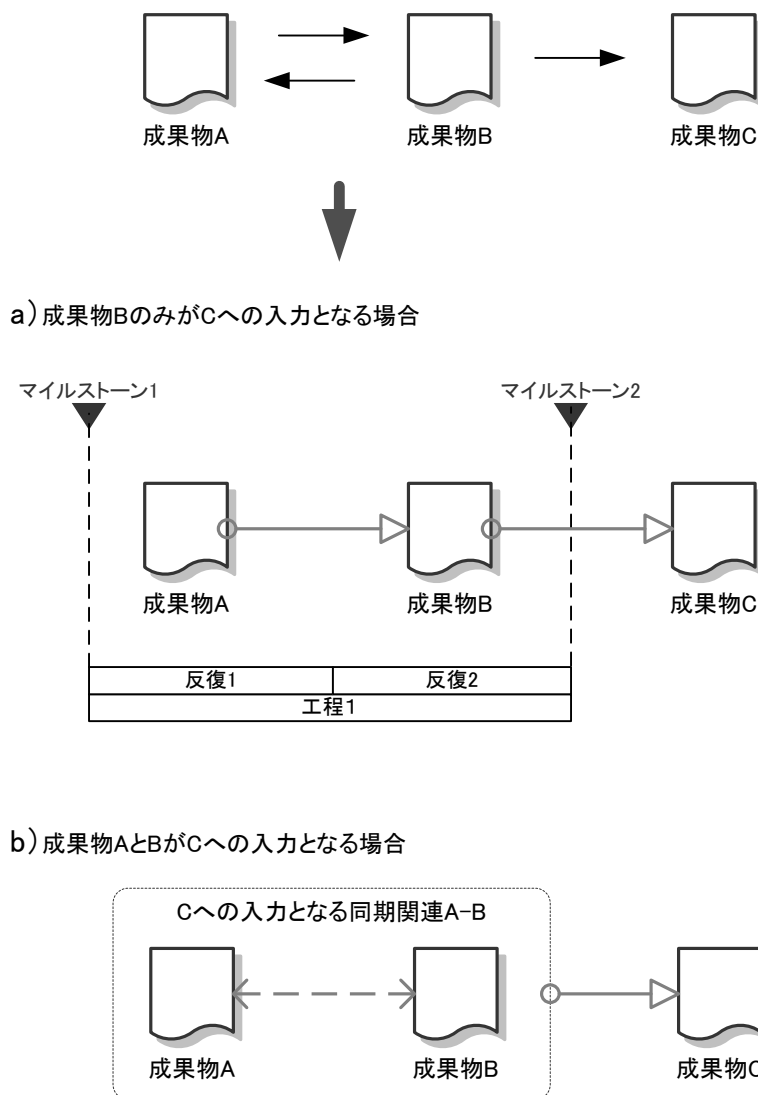


図 13 同期関連の正しい記述方法

手順 5: 必要に応じてプロセスをグルーピングする

定義したプロセスに対して、必要に応じて、グルーピングを行う。例えば、要件定義、アーキテクチャ設計、モジュール実装と言ったように、あるひとまとまりの意味を持った成果物の関連のグループに名前をつける。このグルーピングは、定義した構造モデルに対して、Universal レベルでの概念を与え、また、サブプロセスとして再利用の単位ともなる。

手順 6: 各成果物に対して作業モデルを定義する

上記の手順 5 までで、構造モデルの記述ができたことになり、次に構造モデルの各成果物に対して、作業モデルを定義する。PReP モデルでは、作業モデルの記述に対しての規定は設けていない。これは、定義したプロセスモデルの利用環境および目的、組織内での既存のプロセスモデル資産の利用などによって適切な記述方法が変わるためである。よく用いられる記述方法には ETVX³⁶⁾モデルがある。

4.4 工程モデルとその記述方法

工程モデルは、構造モデルで記述した成果物の関連に対して、開発ライフサイクルを定義するモデルである。図 14 にエンジニアリングプロセスの工程モデルの例を示す。例では、上部にマイルストーンとしての各種会議体が、下部に工程の定義と各工程の反復回数が定義されている。

開発プロジェクトのプロセスは構造モデルに開発ライフサイクルの情報が付加された工程モデルとして定義される。開発プロジェクトの工程モデルは、プロジェクトの最終成果物、適用する技術や管理方法などから、プロセスの雛形となる構造モデルが選択され、選択された構造モデルに対して、開発プロジェクトの目的、条件、制約およびリスクに応じて、工程モデルが定義される。構造モデルで各成果物が複数の作業モデルを保持していた場合には、工程モデルを定義する時点で、プロジェクトの目的、条件、制約およびリスクに応じた最適な作業モデルが選択されることになる。さらに、プロジェクトのリスクを低減するために、実際の開発では、同種の成果物が何回かに分けて作成され、検証・評価される場合がある。開発プロジェクトの規模が大きくなり開発期間が長くなる場合や、技術的リスクにより数回の検証が必要となる際には、例えば成果物の第1版、第2版というように、同種の成果物が何回かに分けて作成される。この場合、作成されるそれぞれの成果物が異なる成果物として定義され、工程モデル上に展開される。

工程モデルは、要件開発、概要設計といった複数の工程の系列として構成される。各工程は、一連のプロセスまたは一連のプロセスの反復として定義される。プロジェクトのリスクや制約、前提条件等の違いによっては、同じ構造モデルから異なった工程モデルが得られることとなる。例えば、要件定義におけるリスクの異なるプロジェクトでは、要件定義の工程内の反復回数が異なった工程モデルが考えられる。工程モデルは、図 15 に示すように、使用する構造モデルに対して、マイルストーン、工程、反復を定義したものとなり、定義した工程モデルを識別するための管理用 ID と、名称および概要が付加される。

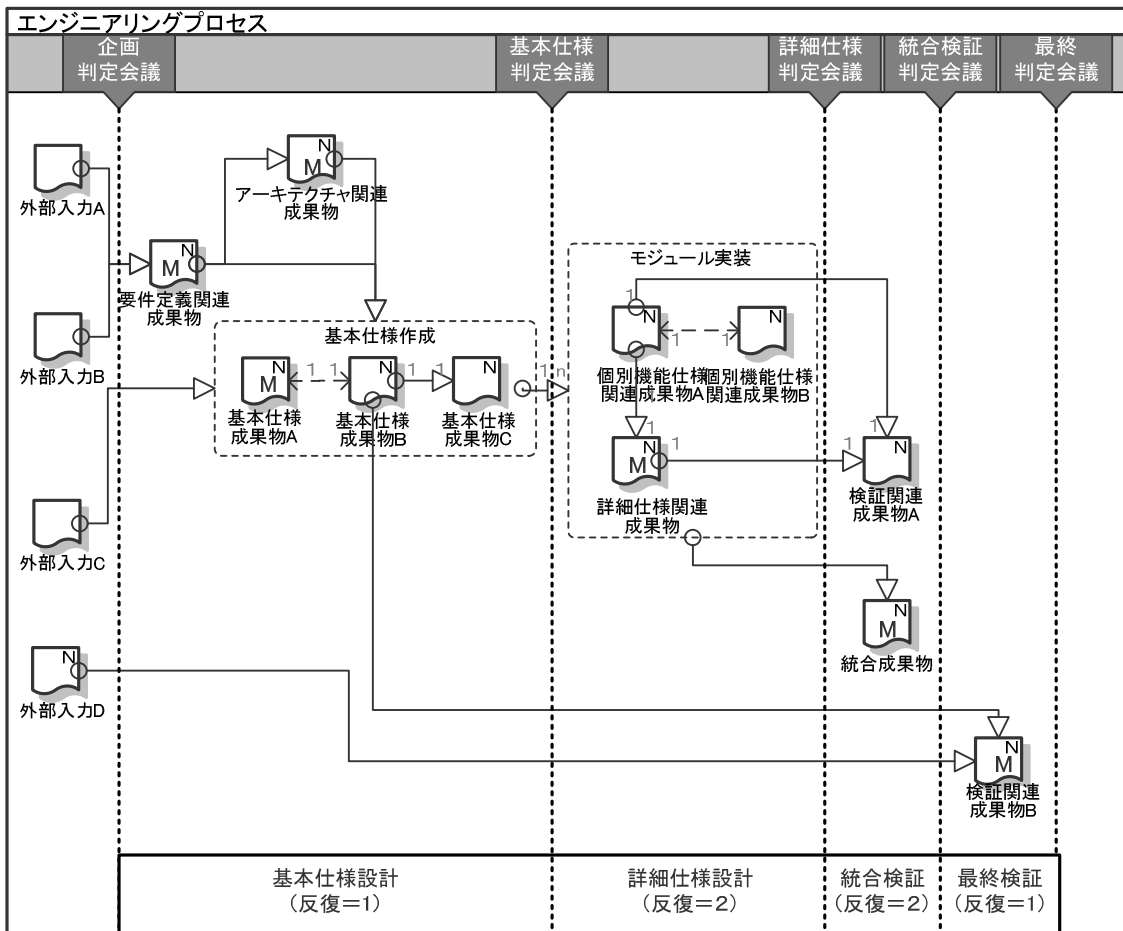


図 14 工程モデルの記述例

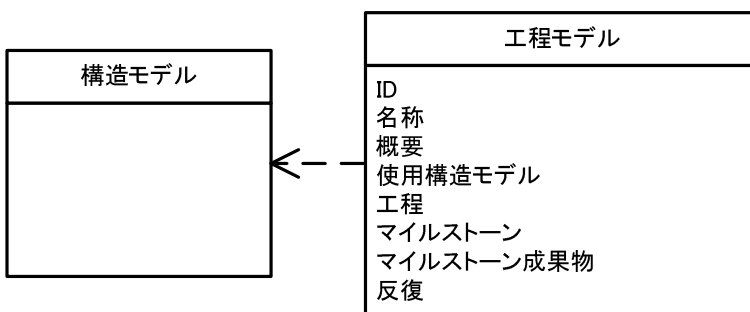


図 15 工程モデルの構成

4.4.1 工程モデルの表記要素

工程モデルの表記で用いられる、工程、マイルストーン、マイルストーン成果物、反復の定義を下記に示す。

工程

マイルストーン成果物に対して、リスク低減の観点から段階的に完成度や内容の確認を取りながら開発を進めていく際の区分け。

マイルストーン

次の工程の開始についてビジネス的判断の見地から合意が形成される場。主に会議体が定義される。マイルストーンでは、マイルストーン成果物(次項参照)を評価することによって、次工程を開始して良いかが判断される。

マイルストーン成果物

開発プロジェクトにとって重要であり、進捗と内容の評価や承認等を行うべき成果物。PReP モデルが成果物に着目したプロセスモデルである特徴から、ライフサイクルを定義する際に構造モデル上で特定する成果物である。

反復

各工程の実施方法として、工程内の反復回数と反復終了時の成果物(主にマイルストーン成果物)の完成度や反復の目的等が定義される。例えば、図 14 において、詳細仕様設計工程では 2 回の反復が定義されているが、この場合、詳細仕様設計工程で定義される一連の作業が 2 回繰り返される。各反復の終了条件は、主にマイルストーン成果物の完成度によって定義される。

4.4.2 工程モデルの記述手順と規則

工程モデルでは、構造モデルに対して、マイルストーン成果物を特定したうえで、成果物の開発段階の区切りを定義することで工程が定義され、各工程では反復が定義される。具体的な定義手順を下記に示す。また、マイルストーンの設定には、下記に示すようにいくつかの規則がある。

手順 1: マイルストーン成果物の特定

構造モデルで定義された成果物から、マイルストーンとなる成果物を特定する。図 16 の例では、「M」の記号のついた成果物(A, C, E, F, J)がマイルストーン成果物として特定されたものである。マイルストーン成果物の特定は、手順 2 に先立って行うが、並行して行う場合もある。

手順 2: 工程とマイルストーンの設定

定義したマイルストーン成果物に対して工程を設定し、工程の終端をマイルストーンとして定義する。マイルストーンの位置は、以下の規則によって定めることができる。

規則 1: 工程に対してマイルストーン成果物が 1 つである場合、その直後がマイルストーンとなる(図 16 の例では成果物 A と成果物 J の直後)。

規則 2: 工程に対してマイルストーン成果物が複数ある場合は、最も下流に位置するマイルストーン成果物の直後がマイルストーンとなる(図 16 の例ではマイルストーン成果物 E と F の直後)。また、マイルストーン成果物以外の成果物がどの工程に含まれるかは次の規則 3, 4 に従う。

規則 3: マイルストーン成果物の直接の入力成果物(図 16 の例での成果物 B, D, I)は、マイルストーン成果物と同一の工程に含まれる。

規則 4: マイルストーン成果物の直接の入力成果物でない成果物(図 16 の例での成果物 G, H)は、それらの出力となる成果物と同じ工程に含む。これは、当該成果物の進捗をその出力となる成果物の側で考慮する必要があるためである。図 16 の例では、G は I の直接の入力であるので I と同じ工程に含まれ、H は G の直接の入力であるので G と同じ工程に含まれる。

手順 3: 反復の設定

各工程の実施方法として、工程内の反復回数を設定する。図 16 の例では、工程 1 に 2 回の反復が定義されている。反復の回数は、プロジェクトごとに異なり、その都度設定される。

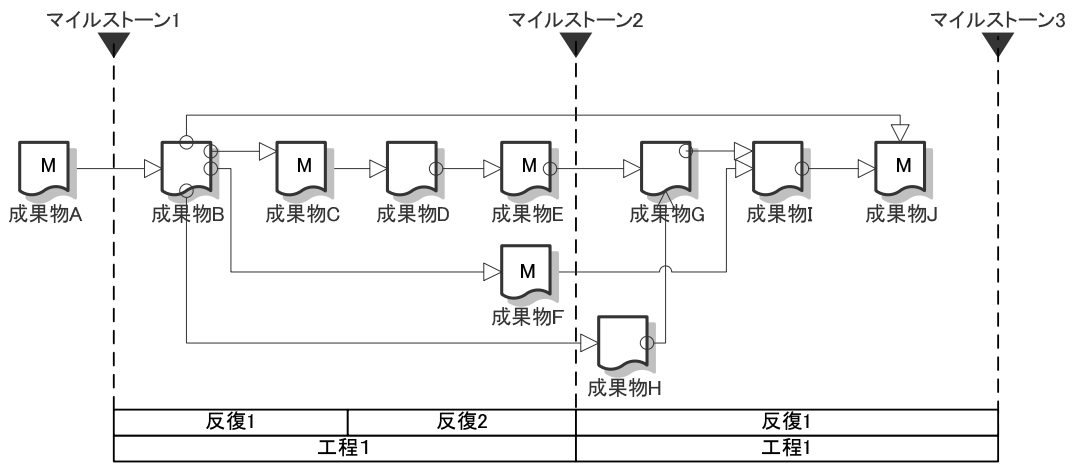


図 16 工程モデルの定義規則に基づいた記述例

4.5 スケジュールの導出

工程モデルに対して、個々の成果物の担当者を定め、カレンダー情報を付加することによって、スケジュールのための基本情報を定義することができる。PReP モデルでは、成果物と関連の種類から入力成果物と出力側の成果物との作成開始タイミングを定義することが可能である。図 17 に入力成果物と出力側の成果物との作成開始タイミングの基本パターンを示す。成果物と関連の種類に対応して「順次開発型」、「分散順次開発型」、「並行開発型」、「終了基準のある並行開発型」の 4 種類の展開方法が定義できる。なお、成果物と関連の種類別の組み合わせによるガントチャートへの展開例を「付録 A」に示す。

順次開発型(図 17 の a)

1 つの情報単位から構成される通常成果物間の入出力関係では、入力成果物の完成を待ってから出力成果物の作成が開始される。

並行開発可能型(図 17 の b)

入力成果物が並列成果物で、出力成果物が通常成果物の場合は、入力成果物の 1 情報単位が作成された時点で、出力成果物の作成を開始することができる。

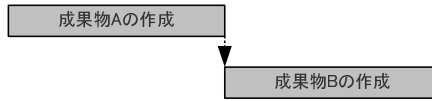
分散可能順次開発型(図 17 の c, d)

入力および出力成果物が共に並列成果物である場合は、入力成果物の一つの情報単位の作成が終わった段階で、出力成果物の作成を開始することができる。それぞれの情報単位毎に複数の作業者を割りあてた分散作業が可能であり、スケジュールを決定する際には、作業者への分散を決定する必要がある。一人の担当者に複数の情報単位を割り当てる場合は、その担当者において、割り当てられた情報に対する順次開発型となる。

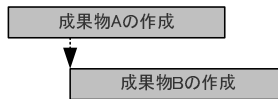
終了基準がある並行開発型(図 17 の e, f)

入力成果物と出力の成果物が並行開発型でさらに同期関係にある場合は、出力成果物の完了が入力成果物の完了条件となる。

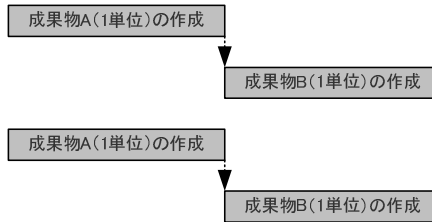
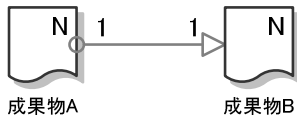
a) 順次開発型



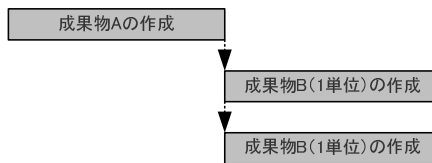
b) 並行開発可能型



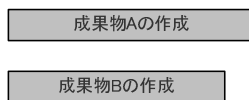
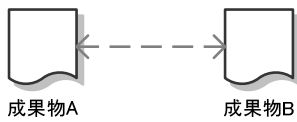
c) 分散可能順次開発型



d) 分散可能順次開発型



e) 終了基準がある並行開発可能型



f) 終了基準がある並行開発可能型

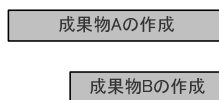


図 17 成果物と関連の種別に対応したスケジュール展開の基本パターン

4.6 管理・支援等のプロセスの記述

PReP モデルは開発プロジェクトの最終成果物を作成するまでの成果物の関連として、要件定義、設計・実装、評価・検証といったエンジニアリングのプロセスだけではなく、プロジェクトの計画策定、進捗管理、リスク管理などのプロジェクト管理プロセスや、構成管理、品質保証、計測などのプロジェクト支援プロセス、組織の標準プロセスの定義・管理やトレーニングなどのプロセス管理プロセスにも適用が可能である。これらのプロセスは、開発ライフサイクルとは別に、主に条件駆動型のプロセスとして定義される。プロセスが駆動する条件としては、変更管理のプロセスのように、設定した条件を満たすイベントが発生した場合に駆動するプロセス、進捗管理のように、あらかじめ設定した日時に駆動するプロセスなどが考えられる。

4.7 成果物関連によるプロセスモデルの意味

これまで説明してきたように、PReP モデルでは、構造モデルおよび工程モデルの記述においてタスクの概念を排除し、成果物の関連構造のみによるプロセスのモデル化に特徴を持っている。成果物は客観的に認識可能な外在化された実体であり、その関連構造は、適用する開発技術とプロジェクトの管理方法から決まる。PReP モデルの重要な特徴は、この成果物という実体の関連構造によって記述されたプロセスのモデル(構造モデル)の下での活動(作業モデル)に対しては、モデル化の方法を規定していないところにある。つまり、入力成果物から出力成果物が作られる作業のモデルは、組織の成熟度、制約、目的、そして作業者に合わせて最適化されることを望んでいる。例えば、標準的な作業工程の手順を細かく定義する場合もあり、使用されるツールを特定するのみである場合もあり、または、ノウハウのみが交換される場合も考えられる。このように、作業モデルを規定しないと言うことは、3.2.1 でも議論をしたように、作業者の柔軟性を最大限に生かすモデルとすることを目的としているからである。

入力成果物から出力成果物を作る作業モデルに対しては、様々な状況の変化に対応する作業者の柔軟性と、作業者の能力を十分に引き出すことができるモデルが求められる。一方、成果物の関連の構造は上位の管理主体である組織の外在化された知識である。この観点から、PReP モデルは、構造モデルで外在化される組織の知識と、構造モデル下での作業者のレベルの知識とを分離し、それぞれのレベルの知識の外在化と共有を可能にするモデルであるとも言える。

5 章 PReP モデルの利用

5 章では、プロセス改善への PReP モデルの利用に関して述べる。プロセス改善への利用は、プロセスの再利用を中心とした標準プロセスの定義、定義したプロセスの計画策定への利用、プロジェクトの実行と管理への適用、プロセス実績の評価と改善が考えられる。

5.1 PReP モデルによるプロセスの再利用

プロセスの再利用の目的のひとつには、Humphrey が「プロセスの定義には非常に大きなコストがかかるため、個々の開発プロジェクトが自分達のプロセスを独自に開発する時間および作業者の十分な割り当てはない³⁾」と述べているように、コスト的な側面がある。すなわち、プロジェクトの計画を策定し、進捗を管理し、プロジェクトのメンバーとプロジェクト実行に関する共通理解を得るためにはプロセスモデルが必要である。しかし、プロセスモデルを記述するためのコストは大きく、プロセスのモデル化作業に十分な資源の割り当てがない場合も多い。そのため、効率よくプロセスをモデル化する方策としてプロセスの再利用が必要となる。

もうひとつの目的は、ベストプラクティスの再利用である。すなわち、組織内で良いと保障されたプロセスの再利用である。さらに再利用の結果、改善点が発見された場合は、改善されたプロセスが次のベストプラクティスとして利用されるといった再利用のサイクルが回され、プロセスが継続的に改善される。CMM および CMMI では、図 18 に示すように、ベストプラクティスを組織の標準プロセスとして定義し、再利用する活動が定義されている⁴⁾。すなわち、組織の標準プロセスを改善のためのプロセス基盤として定義し管理し再利用できるものとする(図 18 の 1)。プロジェクトは標準プロセスを再利用し、プロジェクトの計画を作成し、開発活動を実施し管理する(図 18 の 2)。このとき、標準プロセスのテンプレートとしての再利用は、Humphrey の述べる、プロセス定義のコスト的な目的とも重なる。そして、プロジェクトでのプロセスの適用結果を測定し(図 18 の 3)、その結果を用いて標準プロセスを評価し、改善点を特定し、改善されたプロセスを新たな標準として定義する(図 18 の 4)。

プロセスモデルの再利用は、図 18 における、標準プロセスの定義と、開発プロジェクトでの標準プロセスの適用に関連する。そこでは、再利用される内容と再利用性の高さがプロセスモデルの有効性として重要である。PReP モデルでは、図 4 の PReP モデルの構成に示したプロセスモデルの抽象度の各レベルに対応したプロセスの再利用が可能である。以下に、抽象度の低い「作業モデル」のレベルから順を追って、PReP モデルでの再利用の内容と再利用性について述べる。

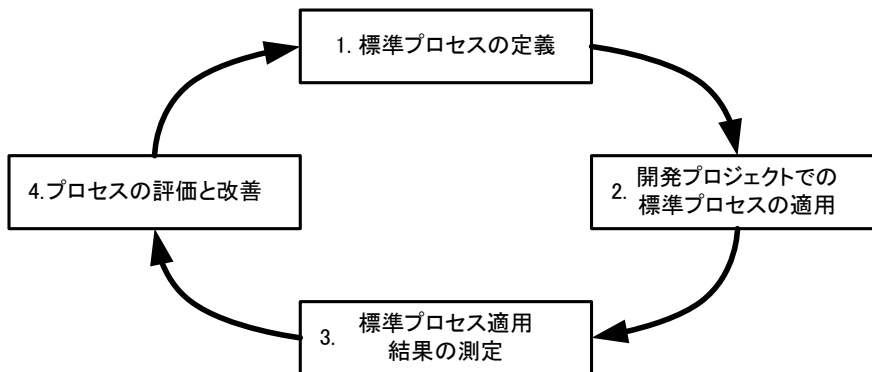


図 18 CMM, CMMI でのプロセスの再利用による改善サイクル

5.1.1 作業モデルの再利用

PReP モデルでは、プロセスモデルを構成する個々の成果物が、成果物を作成するための作業手順情報として作業モデルを保有する。成果物に結び付く作業モデルのレベルでは、下記の再利用が考えられる。

適切な作業モデルの選択と再利用

成果物を作成するための作業モデルは、コストや得られる成果物の品質などに応じて複数定義することが可能である。過去のベストプラクティスから、品質の高い成果物が得られた作業方法や、低いコストで済んだ方法などが蓄積され再利用される。実際のプロジェクトにプロセスを適用する場合には、定義された複数の作業モデルの中から、プロジェクトの目的や制約から適切なモデルが選択され利用される。

見積値／実績値の再利用

作業モデルは「4.2.1 構造モデルの構成」に示したように、成果物の規模の見積り値を入力として、成果物に工数等の見積もり値を返すための見積もり式を持つ。各作業モデルは、成果物単位の見積もり精度を上げるために、見積り式、見積り属性およびその値を保持・管理する。見積り属性には、平均工数やタスクを実施するために必要となる能力などが考えられる。見積り式と値は、成果物単位で収集されたプロジェクトの実績をベースとして更新される。これにより、統計的な類推に基づいた成果物単位での見積りが可能となる。この見積り結果と次項目で説明するリスクや有効性についての知識の再利用とあわせてプロジェクトに適した作業モデルが選択される。

リスクに関する知識の再利用

作業モデルでは、作業を実施した結果、顕在化したリスクが適用事例情報として管理されている。顕在化したリスクを参照することにより、プロジェクトに適した作業モデルの選定および作業の実施において、定性的な判断材料となる。

有効性に関する知識の再利用

プロジェクトで実際に利用された作業モデルに対し、作業担当者による評価結果を適用事例情報として管理する。適用結果に対する定性的評価は、プロジェクトに適した作業モデルの選定において、定性的な判断材料となる。

5.1.2 構造モデルの再利用

構造モデルは、適用する技術と管理方法によってプロセスが決定される。すなわち、適用技術と管理方法の変更がなければ、プロセスモデルは変動せず安定している。そのため、CMM および CMMI における組織の標準プロセスの定義としての利用が有効である(標準プロセスの定義に関しては、「5.2 標準プロセス定義への利用」を参照)。開発プロジェクトのプロセスを定義する際には、プロジェクトの最終成果物、適用する技術、管理方法などから、プロセスの雛形となる構造モデルが選択される。選択された構造モデルに対して、プロジェクトへの適合が行われ、開発プロジェクトの目的、条件、制約およびリスクに応じて、開発ライフサイクル情報が追加された工程モデルが定義される。構造モデルを標準プロセスとして利用する場合、構造モデル全体を再利用する他に、要件定義や検証プロセスのように、ある目的をもったサブプロセスを定義し再利用することも可能である。構造モデルを構成する成果物単位での再利用も含めて、下記の再利用項目が考えられる。

構造モデルのプロセス／サブプロセスの再利用

作成する最終成果物に対して、適用する技術とプロジェクトの管理方法から決定される構造モデルは、同じ技術と管理方法を適用するプロジェクト間での再利用性が高いといえる。また、構造モデル全体を再利用する他に、要件定義プロセス、検証プロセスのようなサブプロセスの単位で適用技術と管理方法が同等の場合、サブプロセス単位での再利用が可能となると考えられる。これらの再利用は、プロセスモデルでのプロセスパターンとしての再利用と考えられる。PReP モデルにおけるプロセスパターンへの適用の可能性に関しては、「5.7 プロセスパターンから見た PReP モデルの利用」で述べる。

プロセス／サブプロセス単位での見積および実績値

プロセスおよびサブプロセスの単位で見積もりと実績の値を管理し再利用することによって、新たに定義するプロセスに対して、統計的な類推に基づいた見積が可能となる。

成果物単位での規模の見積および実績値

成果物の規模(プログラム行数やページ数)に対する規模の見積もりと実績の値を管理し再利用することによって、新たに定義するプロセスに対して、成果物の規模に関する統計的な類推に基づいた見積が可能となる。

成果物に関するリスク

成果物に関するリスク要因として、変更頻度、技術的課題などの成果物特有のリスクが、プロジェクト計画および実行時に利用できる。

成果物のベストプラクティス

品質の高い成果物を公開し共有することで、仕事の質の向上を支援することができる。

プロセス／サブプロセスの利用事例の知識

プロセスおよびサブプロセスの適用結果に関する評価を適用事例情報として管理することにより、プロジェクトに適した標準プロセスおよびサブプロセスの選定における定性的な判断材料となる。

5.1.3 工程モデルの再利用

工程モデルは、プロジェクトのリスク、条件および制約に応じて構造モデルに開発ライフサイクルの情報が付加されたものである。組織内の多くのプロジェクトが、リスク、条件、制約において同等である場合には、組織の標準プロセスとして、開発ライフサイクルの情報を含んだ工程モデルを再利用することが考えられる。工程モデルのレベルでは、次の内容が再利用される。

開発ライフサイクルの再利用

工程モデルレベルでプロセスモデルを再利用することにより、プロジェクトのリスクを低減するための開発ライフサイクルの再利用が可能となる。

開発ライフサイクルに対するリスクに関する知識

工程モデルで定義された開発ライフサイクルに対して、過去に顕在化したリスクに関する情報がプロジェクト計画および管理時に利用される。

5.2 標準プロセス定義への利用

CMM および CMMI では、組織の標準となるプロセスを定義し管理することを要求している。組織のそれぞれの開発プロジェクトが管理された組織の標準プロセスを共通に利用することにより、成果物のスムーズな統合を実現し、さらに適用した標準プロセスを評価することによってプロセス上の問題点を特定し、改善に結びつける。PReP モデルは、再利用性の高さ、成果物の粒度での改善点の特定などの特徴から、組織の標準プロセスの定義に適している。PReP モデルによる組織の標準プロセスの定義には、主に構造モデルが利用される。標準プロセスの定義の手順は、ソフトウェアの開発プロセスに類似している部分があり、下記に、構造モデルを使用した標準プロセス定義の流れを示す。

手順 1: 開発プロジェクトの範囲の定義

「4.3 構造モデルの記述手順」で示した「手順 1: ソフトウェアプロセスの範囲の定義」と同様に、標準プロセスが対象とする開発プロジェクトの範囲(スコープ)を、外部プロセスとのインターフェースを含めて定義する。また、組織内での開発プロジェクトに、最終成果物の種類やライフサイクル、そして規模に関していくつかの種別がある場合には、標準プロセスが対応する複数の標準として定義される。標準プロセスの種類が多くなると定義および管理コストがかかり、逆に、ひとつの標準が複数の種類の開発プロジェクトをカバーすると、個々の開発プロジェクトでの適用コストがかかるため、標準プロセスの種類は、標準の管理と、個々のプロジェクトでの利用の両方の観点から最適な種別を定義することになる。また、定義された標準プロセスに対して、対応する開発プロジェクトの種別のプロファイル(プロジェクトの特徴を示す属性とその値)が与えられる。

手順 2: 標準プロセスの定義要件と制約の定義

標準プロセスの定義にあたって、定義した標準プロセスが満足すべき要件と制約をまとめる。要件と制約には、組織内ですでに用いられている標準や規約、外在化されていない共通の規則やノウハウ、法的な規約などが考えられる。

手順 3: プロセスアーキテクチャの定義

ソフトウェア開発と同様に、要件と制約の定義に続いて、作成する標準プロセスのアーキテクチャの設計を行う。プロセスのアーキテクチャとは、標準プロセスを構成するプロセスの要素間の関係であり、PReP モデルでは、集約プロセス間の関係を入出力となる成果物を用いて定義する。

図 19 に PReP モデルを用いたプロセスアーキテクチャの記述例の一部を示す。図 19 の例では、プロセス要素である、開発チームでの進捗の測定プロセスから出力された進捗報告が、プロジェクト管理での進捗管理プロセスへ入力されている。進捗管理で問題が生じた場合には、再計画要件が出力され、再計画プロセスへ入力される。作成された再計画は、開発チームのエンジニアリングプロセスへの入力となっている。

プロセスアーキテクチャで定義されるプロセス要素の粒度は、標準プロセスを用いる管理者や設計者にとって普遍的な行為の集合となる。例えば、図 19 の事例では、進捗の測定という行為の

集合の中身の実際の測定の方法は、そこで用いるツールや環境、または組織の制約とによって最適化が行われる。しかし、「進捗の測定」という概念は普遍性が高い場合と判断された結果、プロセス要素として定義された。このプロセス要素は、標準プロセスの構造を文書とみなしたときの、いわば大見出しに相当するものであり、利用する際に最も参照しやすい粒度で定義されるものである。

このプロセスアーキテクチャは、プロジェクト管理、開発エンジニアリング、開発支援プロセス（構成管理、品質保証など）、プロセス管理（標準プロセスの評価や更新など）の観点から定義される。

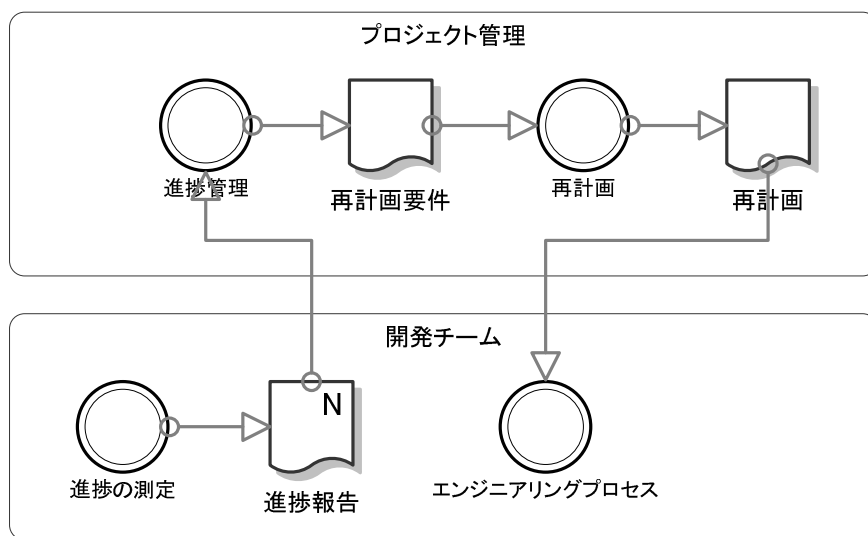


図 19 PReP モデルを用いたプロセスアーキテクチャ記述の例

手順 4: プロセスの定義

プロセスアーキテクチャができあがると、集約プロセスで定義された個々のプロセス要素の中身を、構造モデルで記述することによって、標準プロセスが定義される。構造モデルの定義方法は、前述の「4.3 構造モデルの記述手順」に示した。

5.3 プロジェクト計画策定での利用

図 18 に示したプロセスの再利用による改善サイクルにおいて、PReP モデルを利用した標準プロセスの定義と再利用内容に関して前節で述べた。この節では、定義した標準プロセスの開発プロジェクトでの利用に関して述べる。

標準プロセスを開発プロセスで利用する場合、図 20 に示したような、標準プロセスを使用したプロジェクト計画策定でのプロジェクトのプロセスの定義と、スケジュール策定のプロセスが考えられる。プロセス資産として管理された組織の標準プロセスや過去のリスクを元に、プロジェクトのプロセスを定義し、プロジェクトの体制を定義し、スケジュールを策定する。PReP モデルを利用することによって、標準プロセスの利用から、プロジェクトのプロセスの定義、そしてスケジュールの定義を一貫して行うことができる。以下に、図 20 で示したプロジェクト計画策定でのプロセスの利用の流れに沿って、PReP モデルの利用方法を述べる。

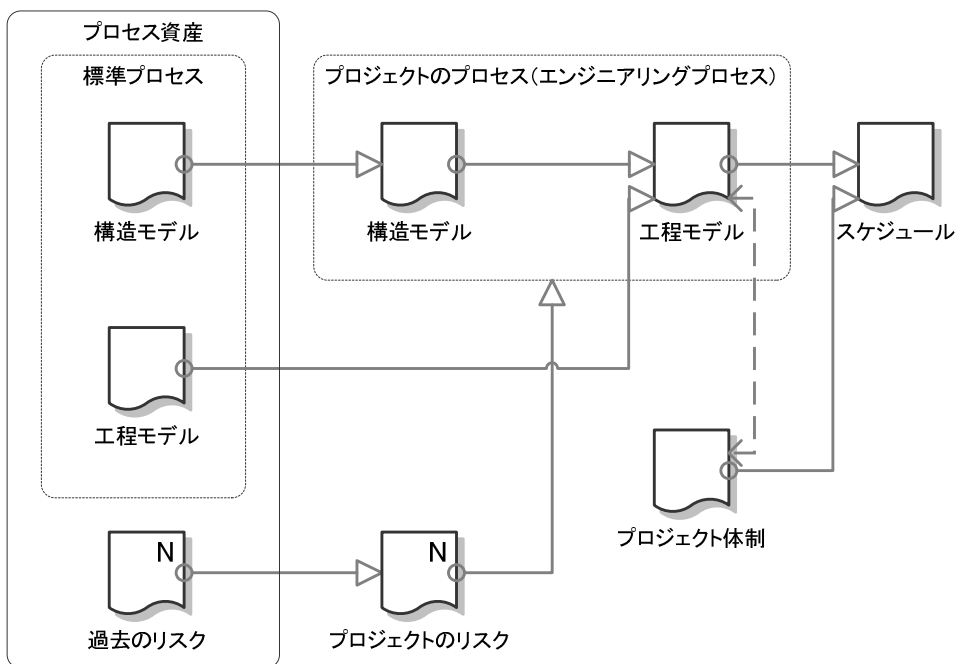


図 20 標準プロセスを使用したプロジェクトのプロセス定義とスケジュール策定の例

5.3.1 プロジェクトのプロセスの定義

開発プロジェクトのプロセスを定義する場合、まず、プロジェクトの最終成果物、適用する技術、管理方法などから、プロセスの雛形となる構造モデルが選択される。選択された構造モデルに対して、プロジェクトへの適合が行われ、プロジェクトのプロセス(構造モデル)として定義される。

選択された構造モデルをプロジェクトへ適合する際には、プロセス資産として管理された過去のリスクから、同種のプロジェクトのプロセスにおいて過去に顕在化したリスクが検討され、プロジェクトのリスクとして定義され利用される。次に、定義したプロジェクトの構造モデルをもとにライフサイクルが定義され、工程モデルが定義される。また、工程モデルを構成する成果物に複数の作業モデルの選択肢がある場合は、プロジェクトに適用する作業モデルが選択される。開発ライフサイクルの決定と、作業モデルの選択によって、プロジェクトに必要となるリソースが決定され、並行してリソース計画としてのプロジェクトの体制が定義される。

5.3.2 スケジュールの定義

工程モデルを定義し、各成果物に対するリソースの割り当てが決まると、スケジュールを決定するために必要な情報が決定される。PReP モデルでは、工程モデルが定義された時点で、スケジュールを定義するための基本情報がそろえることになる。そのため、適切な変換を行うことによって、工程モデルからスケジュールデータの出力が可能となる。スケジュールを定義するためには、出力されたスケジュールデータに対して、カレンダー情報を付加し、リソースの競合を解消する。そして、図 17 に示した成果物と関連の種別に対応したスケジュール展開の基本パターンに従って、配分するリソースと、納期等のプロジェクトの制約を満足するように、分散開発や並行開発の調整を行う。なお、工程モデルからスケジュールデータへの変換に関して、PReP モデルの XML 形式での定義に基づき、Microsoft Visio と Microsoft Project を利用したデータ変換の検証と、支援環境の検討を行った。詳しくは「7.3 プロセス中心型環境での PReP モデルのデータ形式」に述べる。

5.4 プロジェクトの実行と管理での利用

プロジェクト実行時には、定義したプロジェクトのプロセスとスケジュールをもとにプロジェクトの実行と進捗が制御される。プロジェクト実行時には、担当者は割り当てられた作業のプロセス上の位置づけや目標を知るために工程モデルを参照し、標準手順等の情報を得るために担当する成果物の作業モデルを参照する。

プロジェクトの進捗管理では、スケジュールおよび成果物の見積もり値と実績とが比較されて進捗が監視される。もし、計画との乖離が大きくなり、是正処置が必要になった場合には、作業者の割り当ての見直しや、ライフサイクルの見直しなどによってスケジュールが再計画される。

構造モデル、工程モデル、そしてスケジュールは、プロジェクトの実行と進捗の監視において、それぞれ独立した情報として取り扱うことができ、再計画を効率的に行うことが可能である。例えば、構造モデルは適用技術や規模等の大きな変更が無い限り変動は無い。一方、工程モデルは、定義した構造モデルに対して、開発プロジェクトのリスクを低減するように変更される。そのため、開発ライフサイクルの途中でリスクが変動した場合や、計画の見直しが必要となった場合には、必要に応じて工程モデルが変更される。この場合でも、構造モデルには変動が無いため、工程、マイルストーン、および反復の計画の見直しのみで再計画を行うことができる。さらに、開発プロジェクトの作業者の状況が変化した場合や、開発期間の見直しの必要性が生じた場合、作業者の割り当ての見直しや、分散開発、並行開発の見直しを行うことによって、同一の工程モデルに対してスケジュールの変更を行うことができる。

5.5 PReP モデル使用時のプロセス実績の収集

PReP モデルは、成果物を中心としたプロセスモデルである。そのため、プロセス実績の収集も成果物単位で行われる。プロセス実績の収集は、図 18 に示す「標準プロセス適用結果の測定」にあたる。基本的な実績の集計要素としては、成果物の規模、成果物作成にかかった工数および期間、成果物の品質が考えられる。成果物の規模実績値は、プロセスを再利用する際の規模見積もりのためのデータとして利用が可能である。また、成果物の規模と成果物作成にかかった工数から、適応した作業モデルの効率(単位工数あたりの作成規模)のデータを収集することができ、見積もり式の定義のためのデータとして利用が考えられる。また、成果物ごとでのリスクの進捗データから、顕在化したリスクやリスクへの対応方法などが再利用される。このような標準プロセスの適用実績は、次項に示すプロセスの評価と改善のために用いられる。

5.6 PReP モデルにおけるプロセスの評価と改善

図 18 で示した「プロセスの評価と改善」は、収集したプロジェクトの実績データをもとに、行われる。PReP モデルでは成果物単位で実績データが集約され、定義された成果物の粒度で改善すべきプロセスが検討される。例えば、プロジェクト実績の収集データから、成果物に割り当てた作業モデルの効率を求めることができ、ボトルネックとなっていた成果物を知ることができる。また、成果物間の関連から、入力側の成果物の作業モデルの問題を特定したり、顕在化したりリスクから、プロセス上の改善点を特定することが可能であり、下記に示すような改善点の特定と改善方法が考えられる。改善されたプロセスは、標準プロセスとして定義され、再利用される。

1. 成果物単位での効率実績から、効率の低い成果物を特定する。その上で、低効率の成果物の作業モデルに着目し、作業モデル上の問題点を特定し改善する
2. 作業モデルで定義したリソース能力の定義に着目し、定義内容が適切であったかを評価する。リソース能力の定義の見直し、または必要な技術的トレーニングをプロジェクトの計画に盛り込む
3. 管理すべき成果物の粒度に着目し、適切な管理粒度であったかを評価する。粒度が荒く、進捗の把握と是正が適切に管理できなかった、または、粒度が細かすぎて、担当者の能力の自由度を狭めてしまった等の場合に粒度の見直しを行う
4. 効率の低い成果物の入力となった成果物に着目し、適切な情報が入力されていたかを評価する。入力側成果物の作業モデルや使用テンプレートなどに対して、適切な情報が出力されたかを評価し改善する
5. 入力成果物の変更頻度に着目し、変更頻度が高かったどうかを評価する。変更頻度が高かった場合には、入力成果物の作業モデルを見直す。もしくは、外部からの入力成果物の場合には、外部とのインターフェースの安定性を確保するための方法を検討する
6. プロセスの構造の一部または全体に対して適応技術に着目し、プロジェクトに適用したプロセスの構造の一部または全体の変更を検討する
7. 工程モデルで定義した開発ライフサイクルに着目し、プロジェクトのリスクを十分に吸収することができずにリスクが顕在化した場合には、開発ライフサイクルの定義を見直す

5.7 プロセスパターンから見た PReP モデルの利用

CMM および CMMI では、開発プロジェクトのためのプロセスモデルの定義作業に、SEPG (Software Engineering Process Group)と呼ばれる、ソフトウェアプロセスに関する専門の担当者が割り当てられる。プロセスモデルを定義する作業を行うためには、他の技術と同様に、ソフトウェアプロセスに関する様々な経験や知識が必要になる。しかし、ソフトウェアプロセスに関する専門の技術者を設置するためのコスト上の問題、さらには、プロジェクト管理者が自分のプロジェクトのプロセスを容易にモデル化し把握し、管理したいという要求から、プロセスの記述やカスタマイズに関する知識を比較的容易に再利用できるような仕組みが求められている。このような要求に対して、プロセスに関する具体的な解や知識をテンプレートとしてカタログ化し、再利用を可能にする試みとして、「プロセスパターン」という概念が提案され、研究が行われている^{43, 44, 45, 46)}。

5.7.1 プロセスパターンとは

プロセスパターンとは、ソフトウェアプロセスモデルに対してパターン化による再利用を進めようという試みであり、Gamma らによるソフトウェアデザインパターンの手法^{47, 48)}同様に、Alexander が建築分野で提案したデザインパターンの概念^{49, 50)}をソフトウェアプロセス分野に応用したものである。プロセスパターンに関する厳密な定義は困難であるが³⁾、プロセスをパターン化して再利用するためには、プロセスの記述方法、すなわちプロセスモデルに大きく依存すると考えられる。

プロセスパターンを記述するための言語は、1) 読みやすく、2) 記述内容の意味が正確に記述されており、3) 検索が容易なように適切な分類が行われていることが重要である³⁾。パターン記述言語の例としては、オブジェクト指向によるソフトウェア開発プロセスパターン記述言語^{49, 50)}や、UML のクラス図、様々なダイアグラムなどが検討されている。

5.7.2 PReP モデルでのプロセスパターン

PReP モデルで記述された構造モデルで現れる成果物の関連が、プロセスパターンの記述に利用可能であり、かつ有効であると考えられる。例えば、図 21 には、実際のプロセスモデルの記述の結果、構造モデル中に見られたプロセスパターンの例として「モジュール実装」の例を示す。この「モジュール実装」のパターンは、他のモジュール実装のプロセスの記述に再利用された。開発プロジェクトでは、モジュールごとに開発が分担される場合が多いが、各モジュール内での開発プロセスはほぼ同様のプロセスモデルとなり、プロセスのパターンとしての再利用が可能になると考えられる。さらに、図 21 で示したモジュール実装のパターンにおいて、点線で囲まれたモジュール間 I/F 仕様からモジュール内部使用を作成する部分では、プロトタイプが作られているが、これは、図 22 に示すプロトタイプピングのパターンが現れている。

PReP モデルは、成果物の関連の構造のモデルとして、その構造が適用する技術および管理方法で決定されるため、同種の成果物の作成において、適用する技術と管理方法が同じであれば、記述されたプロセスがパターンとして再利用できる可能性が高いと考えられる。その意味からも、プ

プロセスパターンの記述方法として PReP モデルが有効であると考えられる。PReP モデルを使用したプロセスパターンの利用方法は、「7章 プロセス中心型環境への適用」示すプロセスの利用環境とあわせて、今後の研究課題として重要であると考えている。

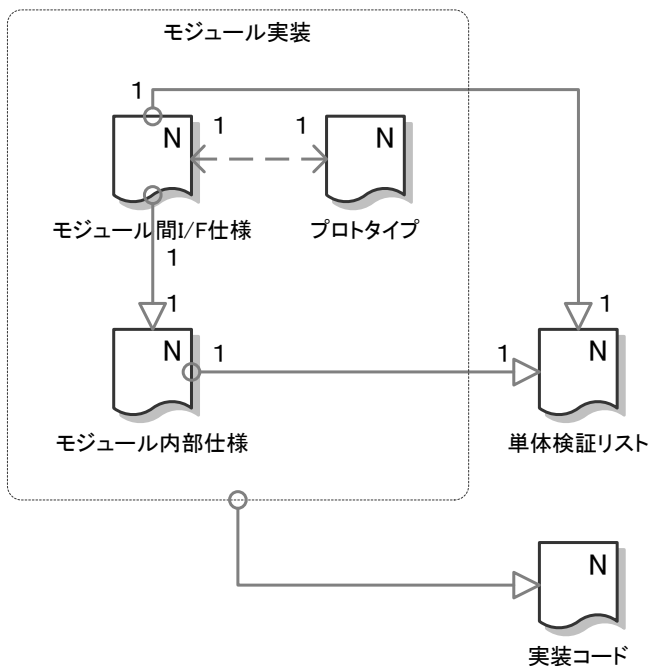


図 21 構造モデル中に見られるプロセスパターン(モジュール実装)

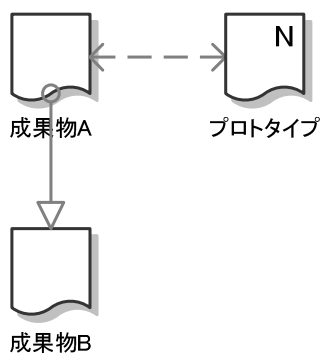


図 22 プロトタイピングのパターン

6 章 PReP モデルの有効性の考察

6 章では、PReP モデルのプロセスのモデル化方法の有効性の検証を行った結果と、その考察を述べる。モデル化の有効性に関しては、「2.2 プロセス改善のためのプロセスモデルの要件」で述べた Humphrey の提案するプロセスモデルの要件と比較して考察を行った。モデル化の有効性の評価を行うために、前章で述べたプロセスモデルの記述が行われるプロジェクト計画の策定段階(プロジェクトのプロセスの定義、ライフサイクルの定義、スケジュールの定義)に対してモデルを適用しその結果を評価した。

6.1 モデル化方法の検証対象

モデル化方法の検証として、実際の開発プロジェクトを対象に適用試験を行った。適用対象は、組み込み系ソフトウェア開発と LSI の論理設計である。今日の LSI のハードウェア設計は、設計記述言語を用いて論理回路を設計する論理設計と、LSI チップ上の物理的な回路パターンを設計するレイアウト設計とに分けることができる。論理設計工程はソフトウェア開発とプロセス上同等の性質を持つ開発であるため、モデル化方法の検証対象として、ソフトウェアプロセスとして扱い、検証対象とした。検証プロジェクト数は、組み込み系ソフトウェア開発プロジェクトが 13 事例、LSI の論理設計プロジェクトが 6 事例である。組み込み系ソフトウェア開発事例の内訳は、既存開発資産の再利用が多くを占める派生展開型開発が 3 事例と、新規開発要素の高い開発が 10 事例であった。

6.2 モデル化方法の適用結果

モデル化方法の適用結果を表 6 に示す。事例1～3 は、組み込み系ソフトウェアの派生展開型開発、事例 4～13 は組み込み系ソフトウェアの新規開発である。また、事例 14～19 は LSI の論理設計プロジェクトである。プロセスモデル作成にかかった延べ人数は、1 人から 6 名、作業延べ時間は、0.5 時間から 24 時間であった。また、定義したプロセスの再利用の関係を表 6 にある「元になったプロセス定義」の列に示した。例えば、事例 2, 3 は、事例 1 で作成したプロセスモデルを再利用して定義した。

適用 8 事例のうち、事例 1 でのプロセスモデル記述結果を示す。図 23 は構造モデルの記述事例である。各成果物の名称は、企業内情報のため一般的な名称に変更してある。図 23 の例に示すように、4 種類の外部入力から、基本仕様関連の成果物、個別の機能仕様、そして検証関連の成果物と最終成果物を出力するまでのプロセスを、構造モデルを用いてモデル化することがで

きた。また、図 24 には、工程モデルを用いて開発ライフサイクルの定義を行った例を示す。工程モデルでは、マイルストーンの定義、工程の定義、工程内の反復の定義を行った図 24 に示すように、マイルストーンとして、「基本仕様判定会議」等の会議体が定義された。これらの会議は実際に行われている会議体であり、そこではマイルストーン成果物として定義した成果物の確認と承認が行われており、実際に行われている活動を工程モデルのマイルストーンとして対応付けることができた。さらに、図 24 で示した工程モデルのうち、最初の工程である「基本仕様設計工程」のスケジュールへの出力例を図 25 に示す。この例では Microsoft Visio で作成した工程モデルから、成果物の属性、成果物間の関連のデータを XML 形式で出力し、そのデータを Microsoft Project のデータ形式に変換したものである。その後、Microsoft Project 上で作業者の割り当て、カレンダー情報の付加、分散作業設定、並行作業での作成開始時期の定義等を行った。

上記で示したように、PReP モデルを使用して、プロセスの定義、ライフサイクルの定義を行い、ライフサイクル情報からスケジュールの基本情報を出力し、スケジュールを定義することができた。事例 1 と同様にして適用を行った 19 事例の結果から、PReP モデルの有効性を次に考察する。

開発分野	開発種別	事例番号	作業人数 (延人)	作業時間 (延h)	元になった プロセス	再利用ノード数/ 全ノード数	再利用率 (%)
組み込み系ソフトウェア開発	派生展開型開発 (3 事例)	1	6	24.0	—	15(全ノード数)	—
		2	4	6.0	事例 1	15/16	94
		3	4	6.5	事例 1	15/16	94
	新規開発 (10 事例)	4	5	7.5	事例 1	15/20	75
		5	4	7.0	事例 4	19/20	95
		6	1	3.0	—	24	—
		7	1	0.5	事例 6	24	100
		8	1	1.0	—	11	—
		9	1	0.5	事例 8	11	100
		10	1	1.0	—	26	—
		11	1	0.5	事例 10	26	100
		12	1	1.0	—	24	—
		13	1	0.5	事例 13	24	100
LSI 論理設計	新規開発 (6 事例)	14	5	10.5	—	29(全ノード数)	—
		15	5	10.5	事例 14	29/29	100
		16	3	5.0	事例 14	17/17	100
		17	2	6.0	事例 14	29/42	69
		18	2	3.0	事例 14	29/36	81
		19	2	3.0	事例 14	29/36	81

表 6 モデル化方法の適用結果

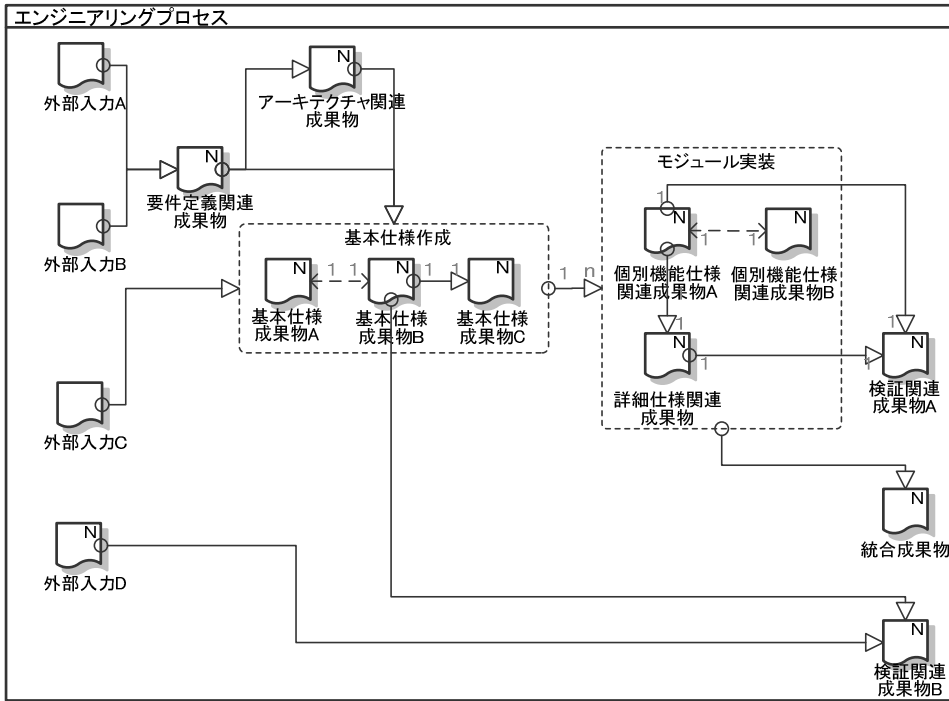


図 23 事例1の構造モデル記述例

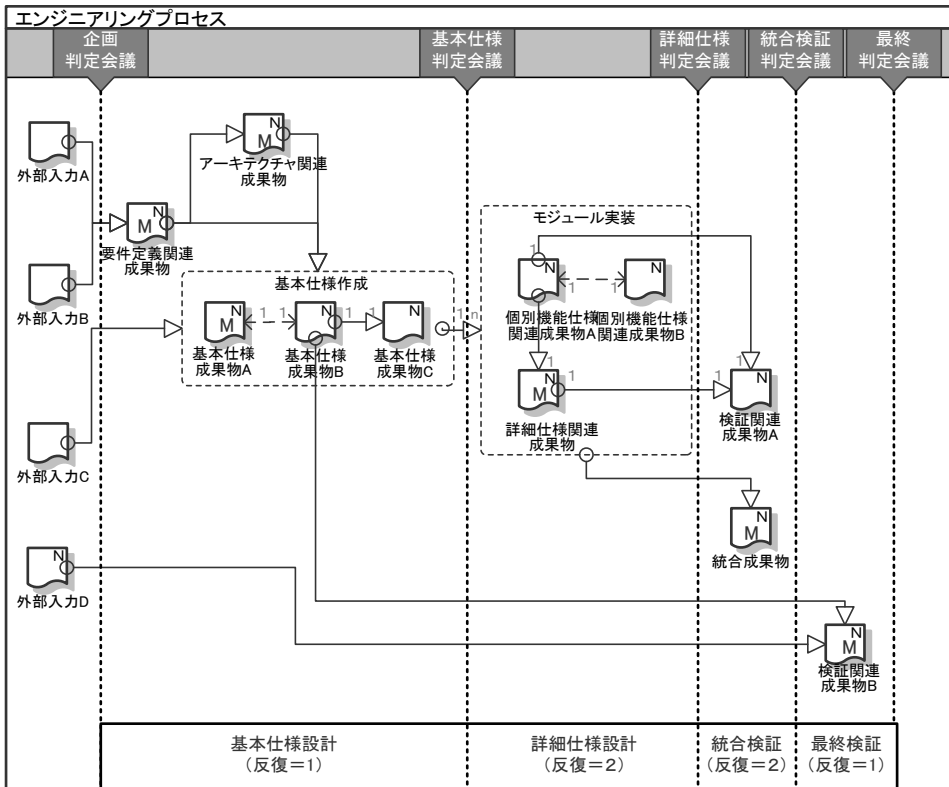


図 24 事例1の工程モデル記述例

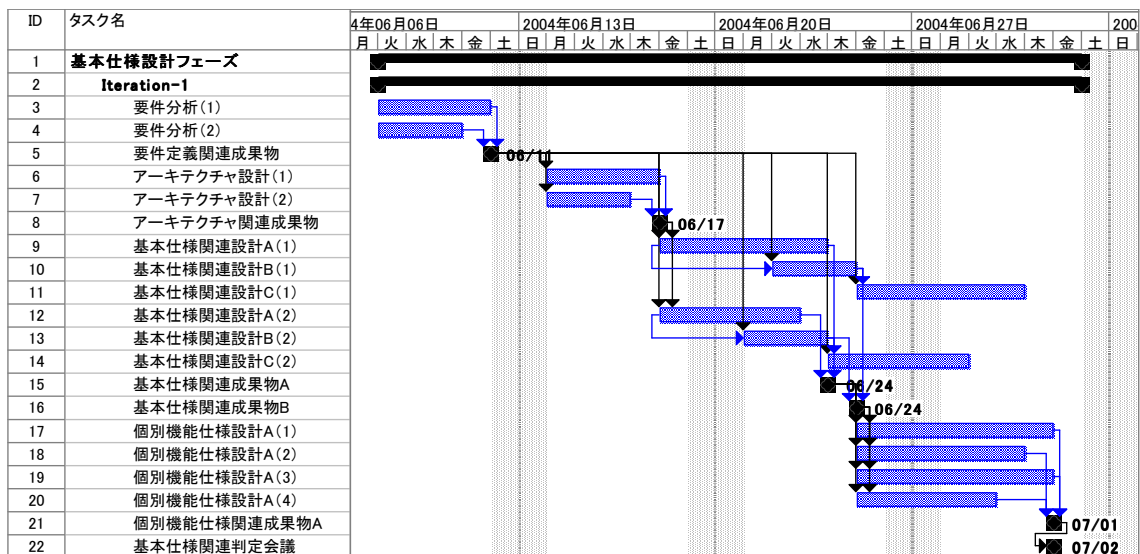


図 25 事例 1 の工程モデルからスケジュールデータへの出力例

6.3 PReP モデルの有効性の考察

適用事例を通じて、PReP モデルを用いてプロジェクトのプロセスの定義、ライフサイクルの定義、スケジュールの出力が行えることを確認した。また、有効性の評価として、これらの事例における作業数と作業時間、プロセスモデルのノード数(同等機能の部分の比較)、再利用ノード数を調べた。さらに、従来のタスク観点でのモデルに対する PReP モデルの有意性を検証するために、PReP モデルの利用者 34 名を対象に、Humphrey によるプロセスモデルに対する 4 つの利用の観点にもとづいた 5 段階評価の調査を行った。

6.3.1 現実の開発活動のモデル化

事例 1 では、以前に 2 名の担当者が開発プロジェクトと並行して、従来のタスク観点でのプロセスモデルの定義を試みていた。しかし、実際の開発活動と適切に対応付けされたプロセスモデルを定義することができず、抽象度の高いレベルで開発プロジェクト全体の概観を示した Waterfall model を定義するに終わっていた。それに対し、PReP モデルによるプロセスモデルでは、図 23、図 24 で示したように、担当者に割り当てられた実際の成果物をもとにプロセスモデルを定義することができた。

実際のモデル化の作業では、まず、各担当者に割り当てられた成果物を洗い出した。洗い出された成果物の粒度はプロジェクトの管理粒度に合わせて定義した。抽出した成果物間の関連を記述することによって構造モデルを作成することができた。定義したプロセスモデルは、現実の開発活動で実際に作成された成果物によってモデル化されており、Worldly レベルのプロセスモデルであると言える。また、適用事例の結果から、プロセス定義作業の延べ時間は 0.5 時間から 24 時間であった。従来モデルによる事例 1 のプロセス記述を試みた際、2 名の担当者が 3 ヶ月を要して

記述を完了できなかった事実と比較しても、非常に短時間でモデル化が可能となったといえよう。特に、実際の適用事例 9, 11, 13 では、定義したプロセスの再利用により 0.5 時間程度で定義を行うことができた。

6.3.2 モデルの利用観点からの評価

Humphrey のプロセスモデルに対する 4 つの利用観点にもとづいた設問による 5 段階評価を、従来のタスク観点でのモデルと PReP モデルそれぞれに対して行った。設問は、1) コミュニケーション: 定義したプロセスが開発要員間でプロセスを理解し議論をする助けになるか、2) 再利用: 定義したプロセスが次回プロジェクトの計画や見積時に再利用可能であるか、3) 改善支援: 定義したプロセスがプロセス改善の助けになるか、4) プロジェクト管理: 定義したプロセスがプロジェクト計画の作成や、進捗管理に役立つか、の 4 問である。評価は PReP モデルの利用者 34 名により行われた。提案方法と従来法との比較に同じ被験者による 5 段階評価を用いたため、評価結果に対して Wilcoxon の符号付順位和検定⁵¹⁾によるノンパラメトリックな検定を行った結果、表 7 に示すように 4 項目全ての利用観点において PReP モデルによる記述が従来のタスク観点での記述に比べて優れている(有意水準 5%)との評価を得た。また、個別のコメントとしては、「プロジェクトの全体のプロセスを見通し良く検討することが可能になった」、「担当している成果物のプロセス上の位置づけが明確になった」、「定義したプロセスを再利用する意味がある」、「プロセス改善や次回の計画に役立つので、成果物単位での規模と工数の計測を行いたい」等の報告を受けた。

評価基準		5 段階評価 平均点	p 値
コミュニケーション	PReP	4.46	2.43e-005
	従来	3.18	
再利用	PReP	4.25	1.76e-006
	従来	2.21	
改善支援	PReP	3.90	2.31e-006
	従来	2.25	
プロジェクトプロセスの管理	PReP	4.18	5.32e-006
	従来	2.64	

表 7 被験者 34 名による PReP モデルと従来モデルとの有効性の比較評価結果

6.3.3 モデル化の粒度

表 6 中, 事例 16 (LSI 論理設計への適用事例) は, 能力の高い少人数の開発者による短期間の開発プロジェクトであった。事例 16 のプロセスモデルは, 29 の成果物による事例 14 のプロセスモデルから中間成果物を省略したもので, 17 の成果物による, より管理粒度の荒いプロセスである。このように PReP モデルは, プロジェクトの管理粒度に応じた粒度でプロセスをモデル化することができる方法であると言える。

6.3.4 再利用性

表 6 に事例間の再利用の関係と再利用率を示した。同じ領域の開発プロジェクト間 (事例 1→2・3, 6→7, 8→9, 10→11, 12→13, 14→15・16) では, 元となるプロセスモデルはそのまま利用されるか, もしくは成果物を追加して利用された。今回の適用事例では, 追加された成果物は, ある成果物の検証用のプロトタイプであった。また, 事例 4, 5, 17, 18, 19 では上流工程, テスト工程等のプロセスが追加され, 共通するプロセス領域はそのまま再利用された。

以上の適用結果から, PReP モデルは現実の開発活動を適切にモデル化していること, プロセスモデルの理解が容易であること, 管理粒度に合わせた粒度でのモデル化が可能であること, 再利用性が高いことを確認することができた。2.2 で議論したプロセスモデルの要件と比較して, PReP モデルはプロセスのモデル化の要件を満足する方法であると考えられる。

6.3.5 改善への適用例

適用事例での改善の一例として, プロセス上の原因となった成果物の特定の例を挙げる。本事例では, 図 26 に示す「検証関連成果物 A」の作成効率において問題が発生していた。従来のプロセス定義では, 効率低下となった成果物を作成するプロセスとして“検証関連成果物の作成”というタスク観点でのプロセスが定義されていた。そして, 当該タスクを中心とした手順等の見直しが検討されていたが, 効果的な改善が見られていなかった。しかし, PReP モデルでの構造モデルの記述を行い, 問題となっていた成果物の入力となる成果物の原因に起因する問題を調べることにより, 効率低下が発生していた成果物の入力となっていた「個別機能仕様関連成果物 A」の作成がモジュール間で一貫性が取れていなかったことが根本原因であることが判明した。このように, 構造モデルを記述することによって, プロセス上での問題の本質原因を, 入出力の関連から検討することが可能であった。このような PReP モデルでのプロセス上の問題点の分析は, プロセス全体を最適化するための TOC (Theory Of Constrains) 理論⁵²⁾への展開も考えられる。

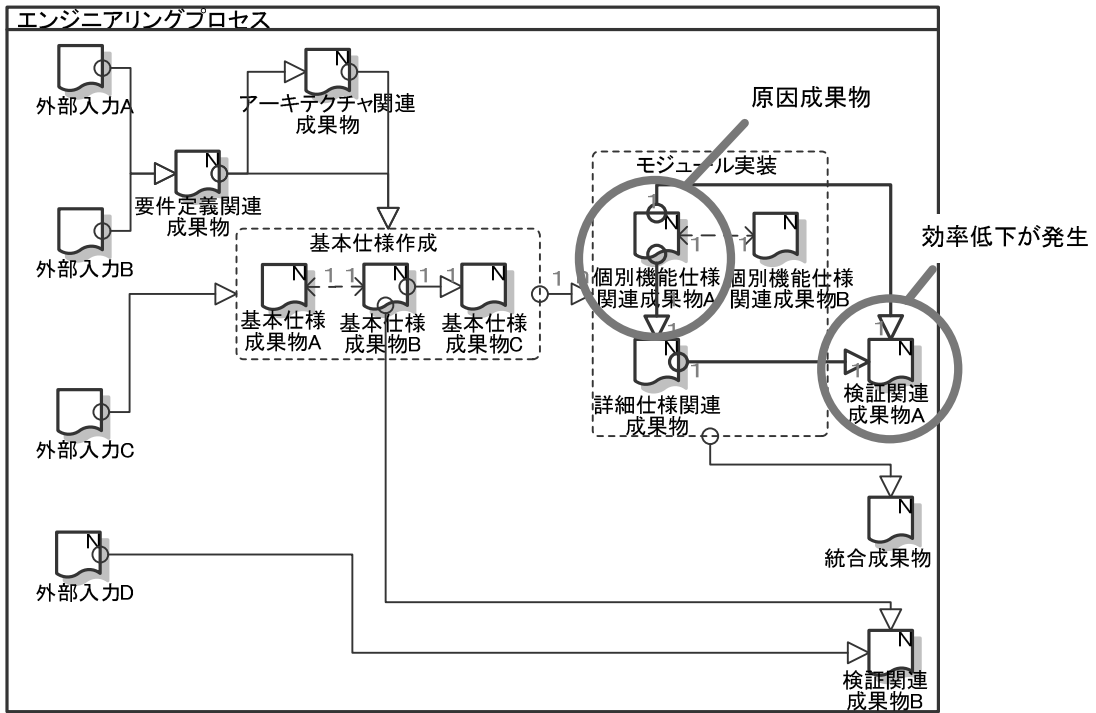


図 26 プロセス上の原因の特定例

7 章 プロセス中心型環境への適用

プロセスモデルの利用方法やプロセスの利用を支援するための環境は、プロセスのモデル化方法に大きく依存する。その意味で、PReP モデルでは、プロセスモデルの記述方法を定義するだけでなく、プロセスの再利用モデルと、プロセス再利用を支援するための支援環境のアーキテクチャも定義する。プロセスモデルをもとに、プロセスの管理、実行改善などを支援するための開発環境は「プロセス中心型環境」と呼ばれており、7 章では、PReP モデルを適用する際のプロセス中心型環境の基本アーキテクチャ案について述べる。

7.1 ソフトウェア開発支援環境

要件の分析と定義プロセス、実行コードの開発プロセス、プロジェクト計画の策定プロセス、進捗の監視、変更管理プロセスなど、定義したプロセスに従った活動を支援するための技術には、商用アプリケーションの一部に組み込まれているものや、要件変更を追跡するためのツールなど、専用のプロセス支援ツールとして開発されているものがある。また、ワードプロセッサ、表計算ソフトといった汎用的なツールがプロセス管理に利用されている例も多い。しかし、これらのツールが支援するものは、多くの場合、断片的な開発プロセスの一部分に限られており、定義したプロセスモデルを適切に連携させ自動化するためには、プロセス支援のための機能を適切にまとめあげた支援環境が必要となる。プロセス支援だけではなく、ソフトウェア開発に関わる様々な支援技術を適切に応用して組み合わせた環境は、ソフトウェア開発環境(SEE: Software Engineering Environment)、または統合開発環境(IDE: Integrated Development Environment)と呼ばれている。このようなソフトウェア開発環境の実現すべき機能を概念レベルで体系的に整理したモデルとして、米国 NIST (National Institute of Standards and Technology) と ECMA (European Communication Manufactures Association) とによって提案されている ECMA リファレンスモデルがある⁵³⁾。このモデルでは、ソフトウェア開発環境の機能要素として以下の 6 種類のサービスを規定している。

1. **オブジェクトマネージメントサービス**: オブジェクト(生産物)を保持・管理する機能
2. **プロセスマネージメントサービス**: プロセス管理全般に関する機能
3. **コミュニケーションサービス**: ツール間でのデータのやり取りやイベントなどの授受機能
4. **ユーザインタフェースサービス**: 支援環境内での統一されたユーザインタフェースを提供
5. **オペレーティングシステムサービス**: オペレーティングシステムに関する機能
6. **ポリシー施行サービス**: セキュリティやアクセス権アンドに関する機能

この中で、プロセス中心型環境では、プロセスマネージメントサービスが特に関連し、このプロセスマネージメントサービス部では、さらに以下のサービスの提供が定義されている。

1. **プロセス開発サービス**: プロセスを定義し、記述し、管理する作業を支援する機能
2. **プロセス実働サービス**: プロセスを実働する機能
3. **プロセス可視化サービス**: 定義したプロセスや実働中のプロセスの状態をわかりやすく表示する機能
4. **プロセス監視サービス**: 実働中のプロセスの状態を監視して、何らかの管理を行う機能
5. **プロセス処理(トランザクション)サービス**: 一連の細かな複数のプロセスからなるプロセス構成や実働を調整し、支援する機能
6. **プロセスリソースサービス**: プロセスの実働に先立って、作業の担い手などのリソースの割り当てを、与えられた制約下で行うための支援機能

PReP モデルを開発組織へ適用・導入するための支援環境の基本アーキテクチャ案について、上記に示した ECMA リファレンスモデルに従って検討を行った結果を次に述べる。

7.2 PReP モデルに基づくプロセス中心型環境

ソフトウェア開発に関連する支援環境の中で、プロセスの支援を中心とした様々な作業の制御や支援の機能を統合した環境は、プロセス中心型環境 (PCE: Process Centered Environment), あるいはプロセス中心型ソフトウェア開発環境 (PSEE: Process-centered Software Engineering Environment) と呼ばれている。プロセス中心型環境として、プロセスの自動実行, 作業の誘導, 開発ツールの統合化, 開発者間の協調作業支援など様々なツールが提案されている⁵⁴⁾, しかし, ECMA リファレンスモデルで定義されているプロセスマネージメントサービス部の機能を統合的に提供する環境として広く利用されているものは現時点ではない。

プロセス中心型環境の動作原理やアーキテクチャは、その環境がどのようなプロセスモデルやプロセス記述言語に基づいているかといった、プロセスの記述方法に大きく左右される³⁾。すなわち、プロセスモデルの目的やモデル化の観点、そしてモデル化の方法は、プロセス中心型環境の仕様、つまり構築する環境を通して何を支援するか引き継がれる。PReP モデルを適用するプロセス中心環境では、プロジェクトの管理とプロセス改善が主な目的となる。さらに、プロセス中心型環境が効果的に機能し利用されるためには、プロセス支援に関する機能を系統立てて利用できるような仕組みが必要である。そのためには、プロセス支援のための機能をどのように利用するか、また組織のプロセスのレベルや状態、そして文化に対してどのような手順で導入するかについての指針も含めた仕組みとして提供されなければならない。ECMA リファレンスモデルでのプロセスマネージメントサービス部の規定は、その意味で、プロセス支援のための機能の系統立てた利用の仕組みとして定義されている。

以下では、PReP モデルと、「5章 PReP モデルの利用」で示した PReP モデルによるプロセス

モデルの再利用モデルに従ったプロセス中心型環境の基本アーキテクチャ仕様について説明する。基本アーキテクチャ仕様では、支援環境で想定されるアクターやサービス、エンティティを洗い出し、論理的なアーキテクチャとして表すことでシステムの概要を明らかにした。また、ここで説明する支援環境を含め、プロジェクト管理とプロセス改善活動を、組織のプロセスのレベルや状態、そして文化に対してどのような手順で導入するかについての指針に関しては、「8章 プロセス再利用の導入モデル」で述べることとする。

7.2.1 論理アーキテクチャ

図 27に、PRePモデルをプロセスモデルとして使用した、プロセス中心型環境の論理的アーキテクチャを示す。プロセス中心型環境の論理アーキテクチャは、以下の4つのパートで構成される。なお、利用者に対するサービスは、「5章 PRePモデルの利用」で述べたPRePモデルを基にしたプロセスの再利用に基づいて定義した。

1. **Actor:** プロセス中心型環境で想定されるアクターを配置
2. **Boundary Layer:** アクターと相互作用を行うサービスを、ユーザの視点から想定した機能単位で配置
3. **Control Layer:** 処理の流れを制御するサービスを配置
4. **Entity Layer:** 目的別に、プロジェクト、プロセス資産、システム管理の領域に分かれており、それぞれの領域に対して想定されるエンティティと、エンティティを管理するサービスを配置

さらに本アーキテクチャでは、プロジェクトの作業領域としてのワークスペースという概念と、プロジェクトの活動結果を公開する領域としてのリリーススペースという概念を定義した。プロジェクトの計画やプロジェクト活動実績といったプロジェクトに関連するエンティティはワークスペースに、組織の標準プロセスやプロジェクトで効果があった活動実績などのエンティティはリリーススペースに該当する。

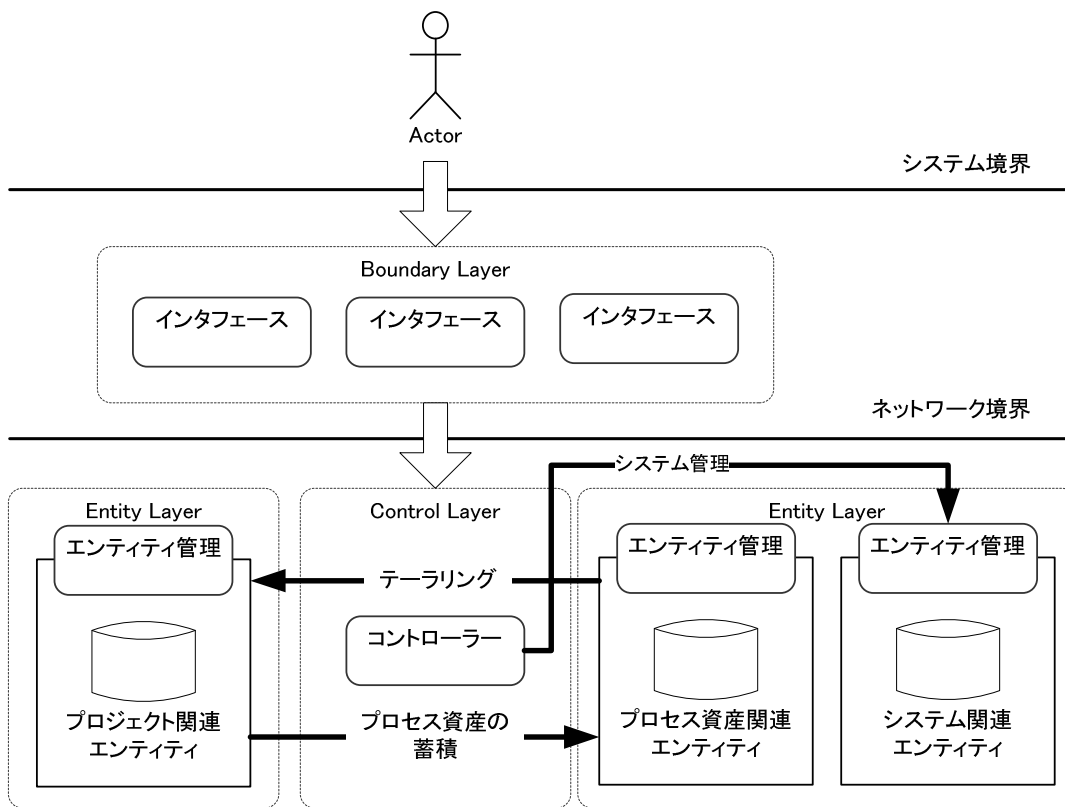


図 27 PReP モデルをプロセスモデルとして使用するプロセス中心型環境の論理的アーキテクチャ

7.2.2 各レイヤーにおけるサービスについて

論理的なサービス群を、図 27 に示す Boundary, Control, Entity の3種類のレイヤーに分類して説明する。

Boundary Layer

Boundary Layer に配置したサービスは、以下の役割と特徴を持つ。

- システム外部とのインタフェースの担当
- システム外部の変更依存し変化を吸収する(システム外部の変更による影響を、他のレイヤーに与えない)

プロセス中心型環境では、主にシステムを利用するユーザの視点から検討し、システムに想定される複数の機能を以下に示すように4種類に分類して、Boundary Layer におけるサービスとして配置する。

1. Project Designer:

プロジェクトプランの策定を行うためのインタフェース群で、主としてプロジェクトリーダーが利用する。

<想定される機能>

- ・プロセス概要定義
- ・プロセス資産検索, 検索結果一覧
- ・構造モデル作成(プロジェクトプロセスの作成)
- ・工程モデル作成(マイルストーン設定, 反復計画作成)
- ・スケジュール作成
- ・クリティカルパス表示
- ・進捗の監視
- ・属性情報設定(成果物, タスク, プロセスの属性)
- ・ユーザ検索, ユーザ検索結果一覧
- ・担当者役割一覧

2. Project Finder:

プロジェクトの実施・管理・計測支援を行うインタフェース群で、主としてプロジェクトリーダーやプロジェクトメンバーなど実際のプロジェクト活動に関わるユーザが利用する。

<想定される機能>

- ・担当成果物, タスク, 入力成果物の確認
- ・プロジェクトメンバーとのコミュニケーション
- ・進捗報告
- ・進捗確認
- ・クリティカルパス表示
- ・プロジェクト実績値(予実差異, 集計値, 生産性など)表示

- ・プロジェクト評価登録

3. Process Manager:

プロセスコンポーネント管理を行うインタフェース群で、主として SEPG など組織のプロセス資産を管理する役割を持つユーザが利用する。

<想定される機能>

- ・メトリクス表示
- ・プロセス評価表示
- ・プロセス資産更新

4. System Administrator:

プロセス中心型環境の管理を行うインタフェース群で、主としてシステム管理者が利用する。想定される機能を以下に示す。

- ・システムユーザ管理
- ・リリーススペース管理
- ・ワークスペース管理

Control Layer

Control Layer に配置したサービスは、以下の役割、特徴を持つ。

- ユースケースの制御を担当する
- ユースケースの変更に依存し、ユースケースの変更の影響を他のレイヤーに与えない

Control レイヤーにおけるサービスとして下記を配置する。

1. ProjectStatus Controller:

プロジェクトの開始や終了などプロジェクトの状態を Control するサービスを提供する。

<想定されるサービス>

- ・プロジェクトの開始
- ・プロジェクト計画の承認, プロジェクト活動の開始
- ・プロジェクトの終了

2. ProjectPlanning Service:

プロジェクト計画の策定, プロジェクトの再計画をするためのサービスを提供する。

<想定されるサービス>

- ・プロジェクト情報の登録
- ・プロセス資産の検索
- ・プロジェクトへの適合を行ったプロジェクトの定義されたプロセス (PDSP: Project Defined Software Process) の登録
- ・マイルストーンの登録
- ・反復計画の登録
- ・規模, 工数, 期間, コストの見積

- ・マスタースケジュールの登録
- ・クリティカルパス情報の取得
- ・成果物ごとの担当者割り当て, 承認ルート設定
- ・プロジェクト計画の出力
- ・プロジェクト計画に対する承認依頼の発行

3. ProjectActivity Service:

プロジェクト活動の支援サービスを提供する.

<想定されるサービス>

- ・成果物, タスク, プロセス情報の取得
- ・関係者とのコミュニケーション機能
- ・進捗の報告, 確認

4. ProjectAssessment Service:

プロジェクト実績の評価を支援するサービスを提供する.

<想定されるサービス>

- ・プロジェクト実績値の集計(工数, 期間, コスト, スケジュールの予実差異の算出)
- ・テーラリング(プロジェクトへのプロセスの適合)情報の取得
- ・生産性, 欠陥密度の算出
- ・プロジェクト実績評価の登録

5. OrganizationProcessManagement Service:

組織プロセスの管理サービスを提供する.

<想定されるサービス>

- ・プロジェクトを横断したタスク, 成果物, プロセス, プロジェクト属性の分析結果
(工数, 期間, コストの予実差異や生産性, 欠陥密度など)を取得する
- ・組織の標準プロセスや組織の標準プロセス候補の登録, 更新, 削除を行う

6. SystemManagement Service:

ワークスペース, リリーススペースやシステムユーザの管理サービスを提供する.

<想定されるサービス>

- ・システムユーザの登録
- ・システムユーザ情報の取得
- ・ワークスペース, リリーススペースの作成
- ・ワークスペース, リリーススペース情報の取得

Entity Layer

Entity Layer に配置したサービスは、以下の役割、特徴を持つ。

- システムが保持すべきデータを操作(CRUD)する
- システム外部やビジネスルールの変更に影響されない

プロセス中心型環境では、「7.2.3 想定するエンティティ」に記述されたエンティティに対して、エンティティを操作する管理サービスを配置している。表 8 に、サービスとエンティティの関係を表した。表の右に示したエンティティ名の内容に関しては、「7.2.3 想定するエンティティ」に記述した。

サービス名	エンティティ名
ProjectPlan Manager	プロジェクトプラン
ProjectActivityData Manager	プロジェクト活動データ
ProcessAssets Manager	組織の標準プロセス グッドプラクティス プロジェクトプラクティス トライアルメソッド プラクティス実績
User Manager	ユーザ情報
Workspace Manager	ワークスペース管理
ReleaseSpace Manager	リリーススペース管理
Project Outline Manager	プロジェクト概要
Resource Manager	要員情報

表 8 Entity Layer に配置したサービス

7.2.3 想定するエンティティ

プロセス再利用の観点からエンティティを検討すると、組織標準プロセスやプロジェクトで使用された標準プロセスの候補となりうる“うまくいった事例”など、再利用の対象となるプロセスを保管するエンティティと、保管されているプロセスを利用して作成したプロジェクト計画やプロジェクト実績など、プロジェクトの管理と実行に必要な情報を保管するエンティティが考えられる。前者のエンティティが含まれる領域を「プロセス資産領域」とし、後者のエンティティが含まれる領域を「プロジェクト領域」として定義する。また、システム的な観点から検討するとシステムを管理するためのエンティティも必要になることがわかる。これらのエンティティが含まれる領域は、システム管理の領域として定義する。次に、上記のプロジェクト、プロセス資産、そしてシステム管理それぞれの領域に含まれるエンティティを説明する。

プロジェクト領域のエンティティ

プロジェクト領域には、プロジェクト計画の策定、プロジェクト活動を行うために必要なエンティティが分類される。

プロジェクトプラン:

プロジェクトのプロセスや見積情報、スケジュールなどプロジェクト計画に関する情報全般がプロジェクトごとに保管されるエンティティ。エンティティに含まれるデータ例を下記に示す。

プロジェクト概要	プロジェクトの目的および背景、プロジェクトの作業範囲、プロジェクトの目標、納品物、成功要因および前提条件、制約、プロジェクト活動上の行動基準(プロジェクトルール)など
成果物	プロジェクトで作成する成果物の情報。属性としては、表 3 に示したものが定義される
作業モデル	成果物ごとに設定された作業モデルの情報。属性としては、表 4 に示したものが定義される
サブプロセス	計画されたプロジェクトプロセスの中で、サブプロセスとして定義された情報
プロセス間関連情報	計画した成果物間の関連およびサブプロセスの関連情報
工程情報	マイルストーンを通過するための条件、評価基準、マイルストーンに必要な成果物など
反復情報	各フェーズ内の反復情報。反復終了時の成果物の完成度や反復計画の目的など
承認ルート情報	成果物の承認ルートの情報など
プロジェクト特性	プロジェクトの規模やプロジェクトの目的および背景、プロジェクトの作業範囲、プロジェクトの目標、納品物、成功要因および前提条件、制約、プロジェクト活動上の行動基準(プロジェクトルール)などのプロジェクトのプロファイル情報
マスタースケジュール	並行タスク、分散タスクなどの設定情報や各タスクの期日
テーラリング記録	標準プロセスやグッドプラクティスのテーラリング記録
文書情報	各成果物が物理的にどの文書に含まれるかあらわした情報

プロジェクト活動データ:

実績工数や進捗データなど、プロジェクト活動を行うことにより発生するデータがプロジェクトごとに保管されるエンティティ。エンティティに含まれるデータの例を下記に示す。

成果物	プロジェクトで作成した成果物の情報
作業モデル	プロジェクトで成果物ごとに設定した作業モデルの情報
サブプロセス	実施したプロジェクトプロセスの中で、サブプロセスとして定義されているグループの情報
プロセス間関連情報	実施した成果物間の関連およびサブプロセスの関連情報
再計画情報	再計画日や再計画の内容
コミュニケーションログ	プロジェクトメンバーのコミュニケーションログ
プロジェクト評価情報	フェーズ評価やイテレーション評価

プロセス資産領域のエンティティ

プロセス資産領域には、組織がプロセス資産として管理する必要があるエンティティが分類される。

組織の標準プロセス:

組織標準プロセスが保管されるエンティティ。

エンティティに含まれるデータの例を下記に示す。

組織標準プラクティス	組織標準の成果物、タスク、プロセス、役割など
標準属性値	タスクの平均工数、工数算出式や成果物のリスクポイントなどプロジェクトを横断してプラクティスの実績を分析、評価した結果、SEPG によって設定される値

グッドプラクティス:

組織標準プロセスの候補となるプロセスが保管されるエンティティ。

エンティティに含まれるデータの例を下記に示す。

グッドプラクティス	プロジェクトで利用された実績があり、SEPG がプロジェクトプロセスから組織の標準プロセスの候補として選択したプロセスや成果物、タスク
標準属性値	タスクの平均工数、工数算出式や成果物のリスクポイントなどプロジェクトを横断してプラクティスの実績を分析、評価した結果、SEPG によって設定される値

プロジェクトプラクティス:

プロジェクトで使用した実績のあるプラクティスで、グッドプラクティスにも組織の標準プロセスにも含まれないプラクティスが保管されるエンティティ。

トライアルメソッド:

プロジェクトでの利用実績はないが、トライアルとして利用するために組織プロセス管理者 (SEPG など) が作成したプラクティスが保管されるエンティティ。

プラクティス実績:

プロジェクトで利用されたプラクティスの実績値や評価が保管されるエンティティ。エンティティに含まれるデータの例を下記に示す。

実績値	プロセス, タスクごとの工数, 期間, コストの予実差異や成果物ごとの規模の予実差異
評価情報	実績値や生産性, 欠陥密度をもとにした成果物, プロセス, タスクの評価情報
平均工数	実績値をもとにしたプロセスやタスクごとの平均工数

システム管理領域のエンティティ

システム管理領域には, システムとして管理する必要があるエンティティが分類される。

ユーザ情報:

システムを利用することができるユーザの情報が保管されるエンティティ。

エンティティに含まれるデータの例を下記に示す。

認証情報	ユーザ ID, パスワード
単価	見積時に利用するプロジェクトメンバーの基本単価
能力	プロジェクトにアサインする際に参考とする能力

ワークスペース管理:

プロジェクトワークスペースの管理情報が保管されるエンティティ。エンティティに含まれるデータの例を下記に示す

プロジェクト情報	ワークスペースに関連付けられたプロジェクトの情報
プロジェクトメンバー情報	システムの利用権限情報(Project Designer, Project Finder など)や 単価, 能力など
プロジェクトのステータス	プロジェクトの開始, 終了など
公開リリーススペース情報	参照可能なリリーススペースの情報

リリーススペース管理:

リリーススペースの管理情報が保管されるエンティティ。エンティティに含まれるデータの例を下記に示す。

リリーススペース情報	リリーススペースの概要説明
管理者情報	リリーススペースにリリース可能なユーザの情報

プロジェクト概要:

プロジェクト識別番号, 開始予定日, 完了予定日, 納品予定物, 予算など組織内のプロジェクトを管理するための情報が保管されるエンティティ。

要員情報:

組織内の要員(社員や関連会社の社員など)の人事管理情報が保管されるエンティティ。

7.3 プロセス中心型環境での PReP モデルのデータ形式

「4.2 構造モデルとその記述」では PReP モデルを人が理解し利用するためのダイアグラムによる記述方法を示したが、PReP モデルをプロセス中心型環境で利用するためには、適切なデータ形式で定義する必要がある。このデータ形式として XML による記述方法の検討を行った。図 28 に工程モデルの XML 記述における DTD の基本構造を示す。構造モデルは工程モデルの構造から、工程の情報を差し引いたものとなる。また、図 28 に示した各要素の定義と DTD の記述例を「付録 B」に示した。

PReP モデルでは、工程モデルが定義された段階で、スケジュールを定義するために必要となる基本データが揃うため、例えば、工程モデルからガントチャートへの変換が可能である。この機能は、図 27 で示したプロセス中心型環境での Boundary Layer のサービスとして、スケジュール作成や進捗の監視機能への適用が考えられる。工程モデルからスケジュールデータへの変換に関して、図 29 に示すように、Microsoft Visio から Microsoft Project への変換を行った事例をあわせて示す。この変換では、図 30 に示した Microsoft Visio で描いたモデルを XML 形式で保存し、保存したファイルを PReP モデルの XML 形式に変換。変換した PReP モデルの XML データをさらに Microsoft Project の XML 形式に変換した。その結果図 31 に示すように、スケジュールの基本データを出力できることを確認した。出力されたデータに、実カレンダー情報、リソース情報を付加し、さらに「4.5

スケジュールの導出」で示した並行開発, 分散開発による調整を行うことにより, プロジェクトで使用できるスケジュールを得ることができる.

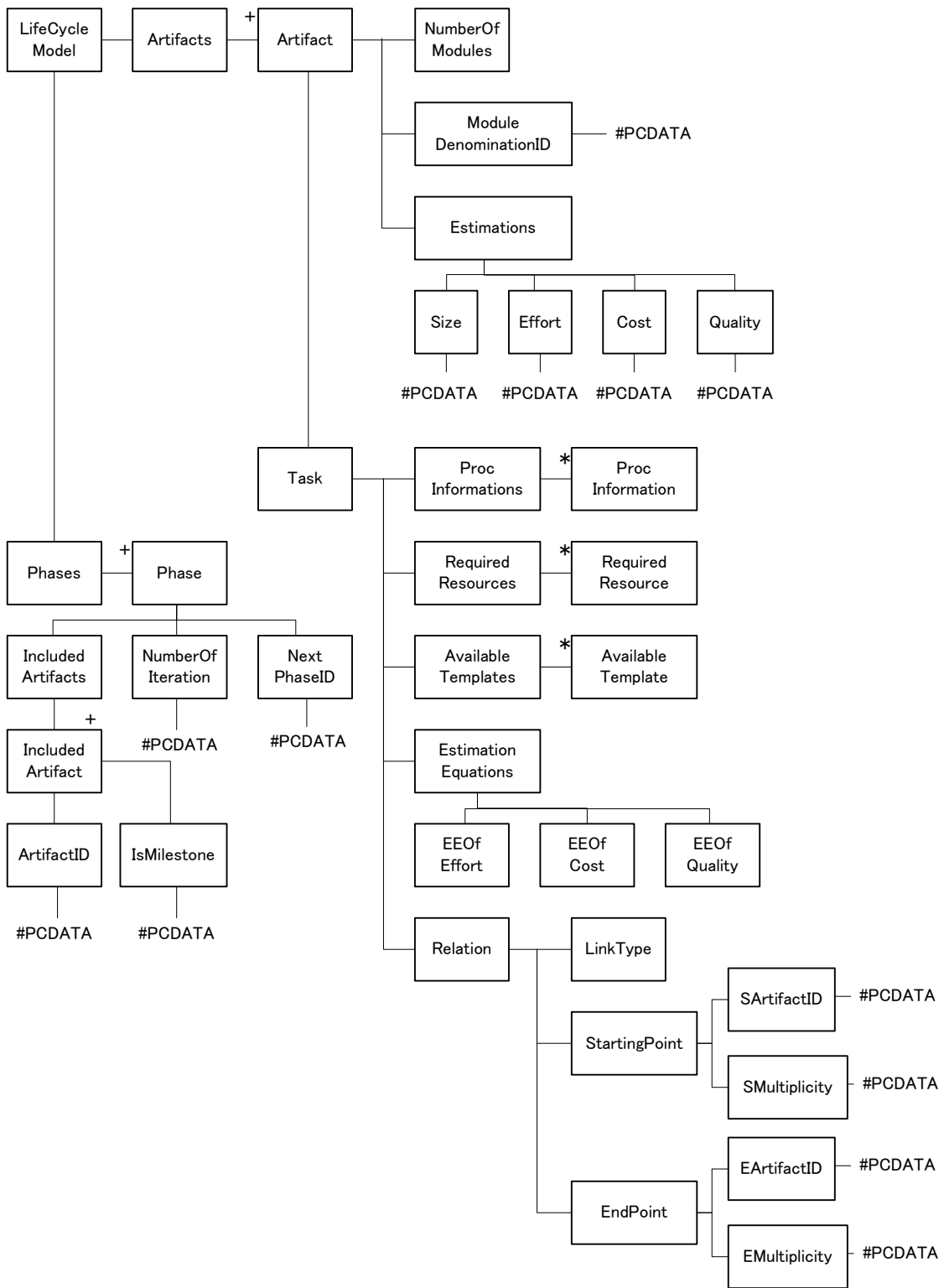


図 28 工程モデルの XML 記述での DTD 構造図

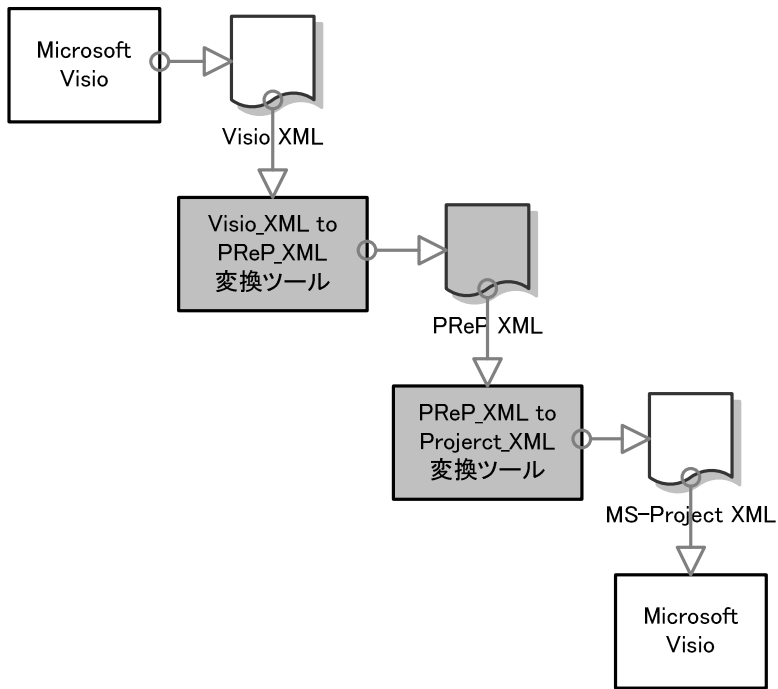


図 29 Microsoft-Visio で描いた工程モデルから Microsoft-Project へのスケジュールデータの変換

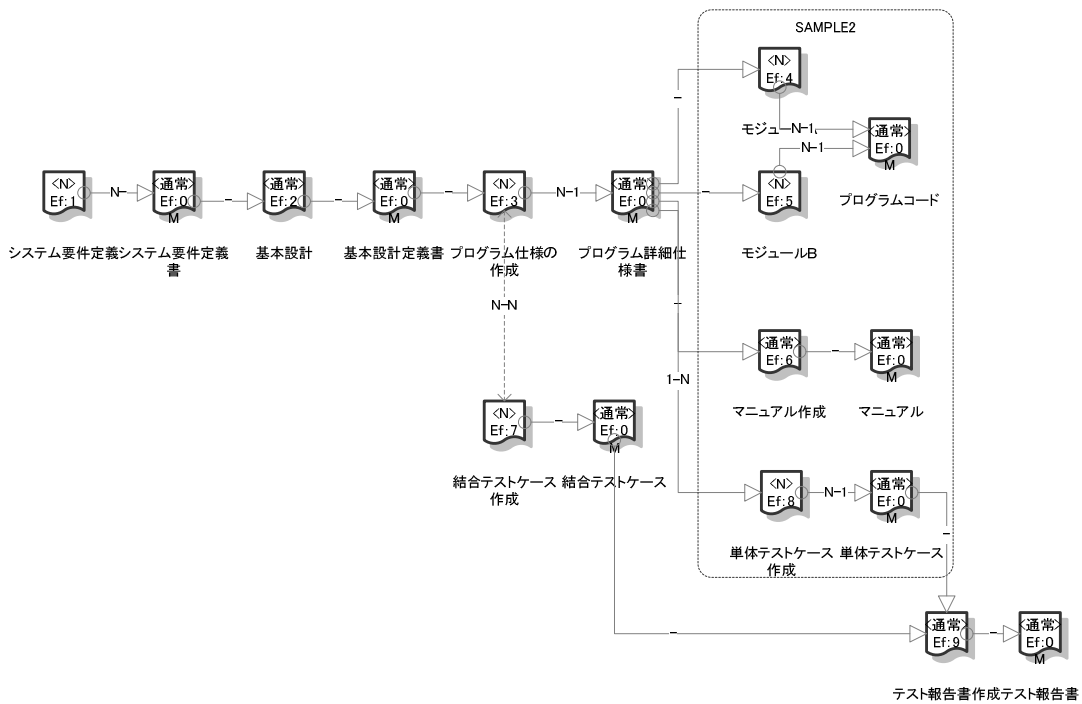


図 30 Microsoft Visio で描いた PRoP モデル

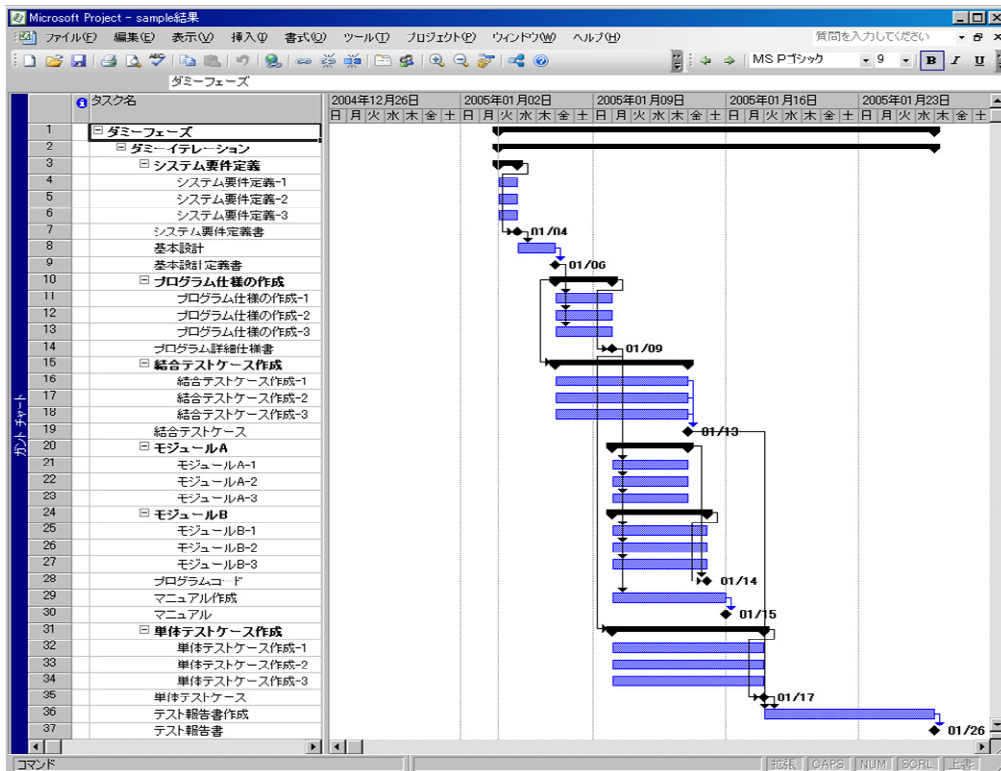


図 31 PRoP XML から Microsoft-Project へ変換を行った結果

8 章 プロセス再利用の導入モデル

プロセスモデルを再利用し、プロジェクト管理やプロセス改善を実際に実現するためには、「7 章 プロセス中心型環境への適用」で示した支援環境に加えて、組織の状態や文化、そして制約に対して、どのような手順でプロセスモデルの利用の仕組みを導入するかについての指針が必要である。この導入の指針は、プロセスのモデル化方法の指針と大きく関係するものである。8 章では、PReP モデルをプロセスモデルとして使用する場合のプロセスモデルの利用の仕組みの実現と、プロセス改善の実現のための導入モデルに関する基本的な案を述べる。

8.1 プロセス改善から見たプロセス再利用の位置付け

CMM および CMMI (段階表現) では組織の能力成熟度のレベルが定義され、これを参照モデルとしたプロセス改善が行われている。しかし、正確には、能力成熟度のレベルとプロセスの改善とは、基本的には独立した概念である。8 章では、まず、能力成熟度のレベルとプロセス改善の関係について整理を行い、その上で、成熟度モデルの背後にあるプロセスの再利用の意味を再定義し、その上で、プロセス改善の観点におけるプロセス再利用の導入方法を確認する。

8.1.1 成熟度レベルとプロセス改善の関係

ソフトウェア開発組織でのプロセス改善の方法として、CMM や CMMI に代表されるモデルベースでのプロセス改善の方法がある。CMM および CMMI (段階表現) では、能力成熟度のレベル (以下成熟度レベル) が定義されている。この「成熟度レベル」とは、実際のソフトウェア開発組織の能力とその特徴を調査し分類した結果、組織の能力が 5 段階の特徴で分類できたとするものである。これを 5 段階の能力レベルとして、各レベルを特徴づける属性をプロセスによって定義している。

5 段階の能力レベルとして、レベル 1 は「初期の段階」で成熟度が低く、その組織でのソフトウェア開発は適切に管理されておらず、常に問題が発生し、プロジェクトはいわゆる“火消し”作業に追われている。一方、レベル 5 の組織では、ビジネス目標にあわせてプロセス改善の必要性が発生した場合、プロセスを適切に、かつ低いリスクで改善することができるとしている。成熟度モデルでは、レベルごとに、そのレベルを特徴付ける組織のプロセスとその特徴が定義されている。この各レベルのプロセスの状態をプロセス改善のためのモデルとして利用する方法がモデルベースのプロセス改善である。

モデルベースのプロセス改善の基本概念は、例えばレベル 2 の組織で特徴的に見られるプロセスの状態と同等の状態が見られるようになれば、その組織はレベル 2 の組織と同等の開発能力

を持っているであろうと予測することである。そのため、モデルベースでのプロセス改善では、まず目標とするレベルを改善の目標の状態として定義する。そして、目標とするレベルのプロセスの状態として参照モデルが定義しているプロセス状態と、現在の組織のプロセス状態とのギャップを分析し、そのギャップを埋める活動を改善の活動とする。また、組織の成熟度レベルは、レベル1から5までの段階を順番に踏んで行くであろうと予測しており、改善活動も、下のレベルから上がっていくことになる。事実、モデルでの各レベルの定義も、ひとつ前のレベルが次のレベルを実現するための条件としてモデル化されている。

プロセス成熟度とプロセス改善

CMM の定義では、プロセス成熟度は、プロジェクトが目標を達成する能力を予測する手助けとなり、組織のソフトウェアプロセスが成熟するにつれて、図 32 に示すような、狙った目標に合わせる上での次の3つの改善点が観察されるとしている⁴⁾。

1. 成熟度が増すにつれ、プロジェクトの狙った結果と、現実の結果の相違が減少してくる
2. 成熟度が高まると、目標値を中心とする実績値の変動が減少する
3. 組織の成熟度が高まるにつれて、目標値も改善されていく

上記の改善点の1と2は、プロセスの予測に対する実績値の誤差が予測かつ管理可能になるということである。すなわちプロセスの予測能力向上が成熟度向上と比例する。一方、改善点の3は、経験的に、成熟度が上がるに連れてプロセスも改善される傾向にあることを述べている。

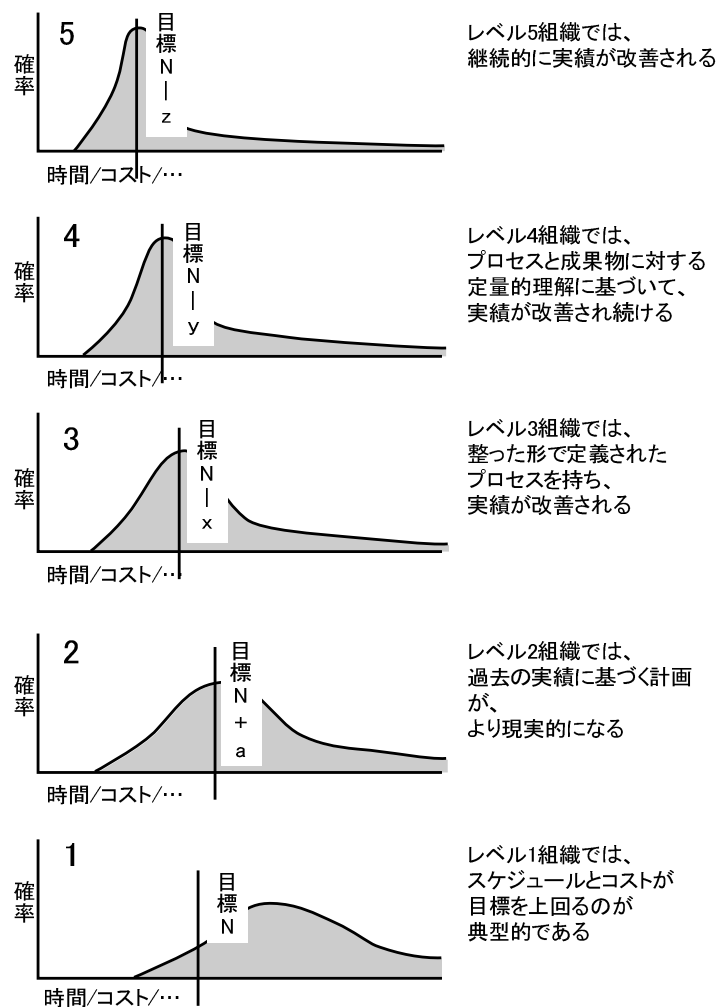


図 32 成熟度レベル毎のプロセス能力(CMU/SEI-93-TR-24 より)

一方で、CMMの定義では、レベル5の組織になってはじめて、プロセスを適切に、かつ低いリスクで改善することができるとしている。さらに、能力成熟度は原文では、「Capability Maturity」であり、Capabilityとは、図32でレベルが上がるに従って分散が少なくなっていることからわかるように、ある目標値に対しての結果の予測能力と管理能力が上がることである。このCapability Maturityが上がることによって、はじめてプロセスを適切に管理し、改善することができるようになる。つまり、本来の意味でのプロセス改善が行えるレベルはレベル5の状態であり、レベル1から5までの成熟度とプロセスの改善とは、本来独立したものである。

図32を、プロセス成熟度とプロセスの改善とに分けて描いたものが図33である。縦軸にプロセスの成熟度レベルを、横軸に改善の状態を示した。縦軸のプロセスの成熟度レベルで見ると、成熟度レベルが上がると、図32にも示されているように、プロセスの予測に対する実績値の誤差(バラツキ)が少なくなりプロセスの管理・予測能力が上がる。一方、横軸の改善の状態で見ると、ある成熟度に応じた管理・予測能力の元での改善活動は、成熟度に応じた評価能力(ばらつきの

上での評価結果)の元で可能となる。実際、成熟度を上げる活動を進める過程で、その時点での組織でのベストプラクティスがプロセスとして定義され、さらにレベル2でのプロジェクト管理プロセスが制度化されることにより、プロジェクトの管理能力が上がる結果、定義したベストプラクティスが再利用できるようになり、図 33 で示した曲線のように、成熟度レベルが上がるとともに、プロジェクトの実績も良くなる場合が多い。つまり、レベル 5 を待たなくても、プロセスの再利用を適切に行うことにより、組織の成熟度レベルに応じた改善が可能となると考えられる。

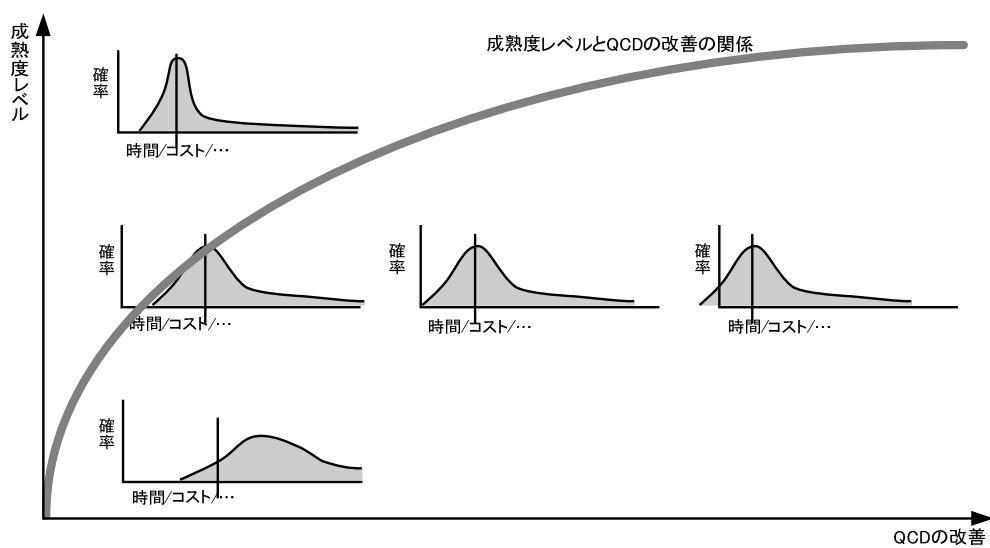


図 33 成熟度レベルとプロセス改善の関係

8.1.2 プロセスの再利用から見た成熟度レベルの構造

CMM および CMMI の段階表現での各成熟度レベルの定義を表 9 に示す。CMM と CMMI とでは、各レベルの表現が多少異なるが、基本構造は同等である(CMMI が CMM をより洗練したものであると解釈し、以降は基本的に CMMI のモデルをもとに議論を進める)。レベル 2 では、個々のプロジェクトに対してプロセス管理の基本能力が求められており、この管理された状態によって、同等のプロジェクトにおいてプロセスを反復できる能力を持つようになる。レベル 3 では、一つひとつのプロジェクトでのプロセスの管理から、組織レベルのプロセスの管理能力に発展し、組織は、組織の標準となるプロセスを定義し、個々のプロジェクトが利用することが求められる。さらに、レベル 4 では、個々のプロジェクトが実施したプロセスを定量的に評価することが求められている。

能力成熟度レベル	CMM	CMMI(段階表現)
レベル 1	初期	初期
レベル 2	反復できる	管理された
レベル 3	定義された	定義された
レベル 4	管理された	定量的に管理された
レベル 5	最適化する	最適化している

表 9 CMM, CMMI での各成熟度レベルの定義

表 9 で示した成熟度レベルは、組織の能力のレベルを段階的に表現したものであるが、プロセスの再利用という観点からそれぞれのレベルの関係を、図 18 で示した CMM, CMMI でのプロセスの再利用による改善サイクルとあわせると、図 34 のように解釈することができる。すなわち、レベル 3 では、組織の標準プロセスを定義し、それを管理・運用できる状態が求められている。これは、組織としてプロセスを再利用するための基盤を作る活動である。この組織の標準プロセスを各プロジェクトが利用できるための基礎能力がレベル 2 で定義されている“管理された(反復できる)”状態であり、個々のプロジェクトは、定義されたプロセスに従ってプロジェクトを管理し遂行できる能力を持つ。さらにレベル 4 は再利用したプロセスの結果を定量的に測定することができるようになる状態である。また、CMM および CMMI では、再利用のサイクルを回すための制度化と SQA(プロセス QA)の仕組みによるプロジェクトの遵守という方法が定義されている。

このように、レベル 2 から 4 の能力を獲得することによって、組織はプロセスの再利用と再利用結果の定量的な評価を行えるようになり、その結果、レベル 5 で定義されている“プロセスの最適化”が可能となる。つまり、プロセス成熟度レベルを、プロセスの再利用の観点で見直すと、その構造はプロセスを再利用するフレームの段階的構築というモデルと解釈することができる。またこの意味から、成熟度の段階は、プロセスの再利用の実装の段階と考えることができる。

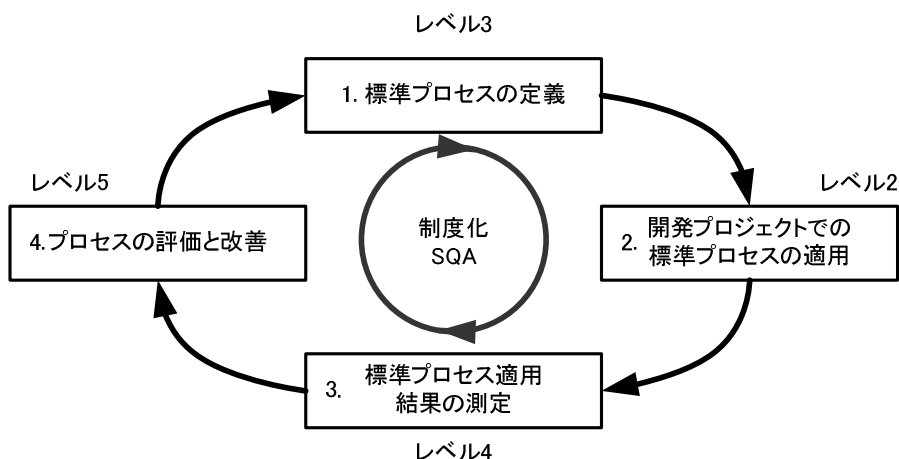


図 34 プロセスの再利用から見た成熟度レベルの構造

8.1.3 成熟度レベルに応じたプロセス再利用の実装方法

成熟度のレベルをプロセスの再利用能力の段階的な実装の順序と定義した場合、その導入モデルはどのようになるであろうか。CMM/CMMI(段階表現)で定義されている、“成熟度を上げる改善活動が段階的に進むこと”に関して考えてみる。もともと CMM/CMMI のモデルは、世の中にあるソフトウェア開発プロジェクトをサンプリングし、クラスタリングした結果である。開発組織が成熟度レベルの段階を踏んでレベル4を実現するためには、レベル3の状態が必要であり、レベル3を実現するためにはレベル2の能力を獲得している必要がある。そのため、開発組織の成熟度が、この段階を踏んで成長していくことも納得できる。しかし逆説的ではあるが、もともと各レベルが次のレベルへの条件になっているというモデルであり、このモデルを参照モデルとした改善では、モデルに従って段階的に実装していくことが順当になる。

図 35 は、「ソフトウェア能力成熟度モデル(TR24)」⁴⁾の「2.3 ソフトウェアプロセスの可視性」の項で使用されている図である。この図は、各成熟度レベルでのソフトウェアプロセスに対する可視性をモデル化したものである。CMM では、各成熟度レベルでのソフトウェアプロセスに対する可視性を、次のように説明している。

- 各後続の成熟度レベルは、レベルが高いほど、ソフトウェアプロセスに対するより良い可視性を提供している
- レベル 1:ソフトウェアプロセスは無定型の存在であり、プロジェクトのプロセスに対する可視性は限定されている
- レベル 2:基本的なプロジェクト管理プラクティスが確立される。これらの管理制御は、定義された時点においてプロジェクトの可視性を与える。ソフトウェアを構築するプロセスは、ブラックボックスの連続のように見え、活動が箱と箱の間を流れる通過点において管理層に可視性を与える(プロジェクトマイルストーン)
- レベル 3:箱の内部構造、すなわち「プロジェクトの定義されたソフトウェアプロセス」での

タスクが見えるようになる。マネージャとエンジニアは、プロセスにおける各自の役割および責任を理解し、適切な詳細さのレベルで、彼らの活動が相互にどのように作用するかを理解している

- レベル4: 定義されたソフトウェアプロセスに計測の仕組みが備わり、定量的に制御されている
- レベル5: 生産性および品質を改善するために、ソフトウェア構築の新しい、あるいは改善された方法が、制御された形でたえず試されている

図 35 で表現された各成熟度レベルでのソフトウェアプロセスに対する可視性を、プロセスの粒度の違いとしても見ることができる。つまり、適切な粒度による表現が可能なプロセスモデルを再利用することで、組織の成熟度レベルに応じた改善が可能となると考えられる。つまり、前節で議論をしたように、組織の成熟度レベルを、プロセスの改善とは独立したプロセス再利用の実装の段階として考えることができ、プロセスの再利用の仕組みを実装した上で、適切な粒度での表現が可能なプロセスモデルを使用することにより、組織の成熟度レベルに応じたプロセスの改善を実現することができると考えられる。PReP モデルは、「6.3.3 モデル化の粒度」で見たように、適切な粒度による表現が可能であり、ここでのプロセスモデルに利用できると考えられる。つまり、PReP モデルに基づくプロセス再利用の仕組みの実装モデルと、その上でのプロセスの可視性のレベルに応じた PReP モデルの利用を行うことによって、プロセス改善を実現することが可能であると予測できる。次節では、PReP モデルに基づくプロセスの再利用の導入モデルに関して述べる。

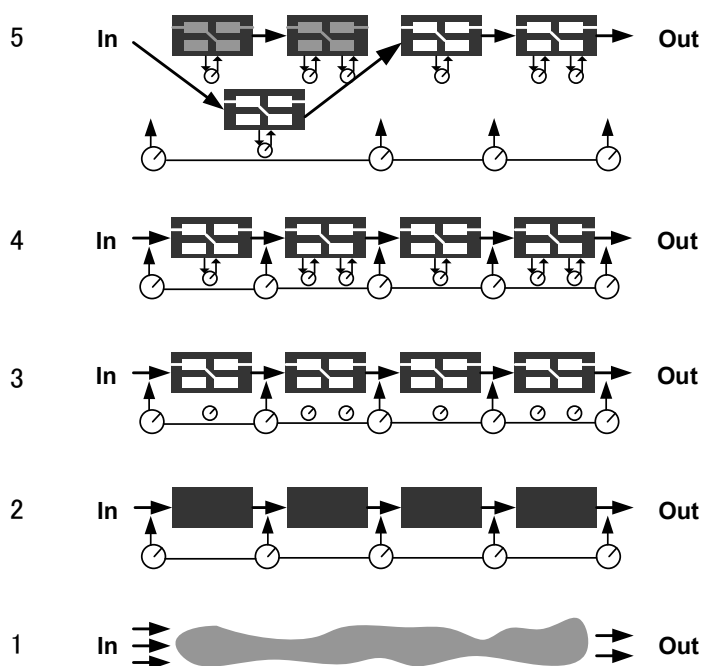


図 35 各成熟度レベルにおける管理層から見たソフトウェアプロセスに対する可視性
(ソフトウェア能力成熟度モデル(TR24), 2.3 ソフトウェアプロセスの可視性, 図 2.3 より)

8.2 プロセス再利用の導入モデル

組織へのプロセス再利用の仕組みの導入を、CMMIで定義されているプロセス領域に基づき定義する。CMMIの段階表現では、組織の成熟度レベルの段階に対応するプロセス領域が定義されているが、以下で述べるプロセス領域の導入モデルは、段階表現とは異なり、組織へのプロセス再利用の仕組みの導入モデルである。このモデルは、CMMIの連続表現の利用モデルとして捉えることもできる。

図36にプロセスの再利用を実現するためのプロセス領域を、図18で示したCMM、CMMIでのプロセスの再利用による改善サイクルと対応させて示した。標準プロセスの定義を実現するためには、OPD(組織プロセス定義)が必要である。また、定義したプロセスを開発プロジェクトで適用するためには、PP(プロジェクト計画策定)、PMC(プロジェクトの監視と制御)、IPM(統合プロジェクト管理)、QPM(定量的プロジェクト管理)、そしてOT(組織トレーニング)の実現が必要となる。さらに、プロジェクトへのプロセスの適用結果を測定するためには、MA(測定と分析)が必要となり、測定結果に基づいたプロセスの評価と改善では、OPF(組織プロセス重視)、OPP(組織プロセス実績)と、OID(組織改革と展開)が必要となる。これらのプロセス領域を実装することによって、プロセスの再利用サイクルを実現することができる。

プロセスの再利用サイクルの上で、再利用され、最適化の対象となるプロセス領域にはエンジニアリングプロセス領域があり、さらに支援の領域とプロジェクト管理の一部のプロセスとして下記がある。つまり、PRePモデルに基づくプロセス再利用の仕組みを実装するために図36に示すプロセス領域の実装が必要であり、その上でのプロセスの可視性のレベルに応じて下記に示すプロセス領域をPRePモデルによってモデル化し再利用することによって、プロセス改善を実現することが可能となる。

再利用対象となるプロセス領域

エンジニアリングプロセス領域

- REQM:要件管理
- RD:要件開発
- TS:技術解
- PI:成果物統合
- VER:検証
- VAL:妥当性検証

支援プロセス領域

- CM:構成管理
- PPQA:プロセスと成果物の品質保証
- DAR:決定分析と解決
- OEI:統合のための組織環境
- CAR:原因分析と解決

プロジェクト管理プロセス領域

- SAM: 供給者合意形成
- RSKM: リスク管理
- IT: 統合チーム編成
- ISM: 統合供給者管理

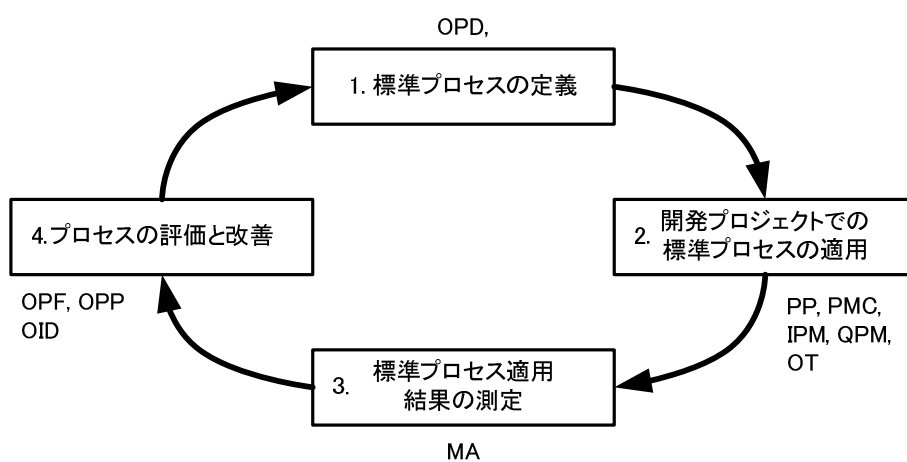


図 36 プロセス再利用の導入に関連するプロセス領域

9 章 まとめ・今後の発展

本研究は、スケジュールの遅れ、ソフトウェア品質に基づく事故、そしてコストの見積もり間違いなど、ソフトウェア開発プロジェクトの進め方に起因する失敗を背景とした中で、失敗原因を改善するために定義されたソフトウェアプロセスが、実際の開発プロジェクトで利用されないために、改善が実現されないといった課題を解決すべき問題とした。そして従来のプロセスのモデル化方法が、現実の開発活動を適切に表現していないことに根本的な原因があると予測し、現実の開発活動を適切に表現し、プロセスの改善に適したプロセスモデルの開発を取り上げた。その結果、本研究の成果として、現実の開発活動を適切に表現し、プロセスの改善に有効なプロセスモデルである PReP (Product Relationship Process) モデルを得た。従来のモデル化の方法の多くは、タスク(作業)の時間的順序関係によってプロセスをモデル化しているのに対し、PReP モデルは、開発プロセスの中で作成され管理される実体である成果物に着目し、モデルから、時間的な順序関係を排除した成果物の入出力の関連構造による静的な構造のモデルを特色としており、下記に示す特長を持つプロセスモデルである。さらに、プロセスの記述方法(Notation)の定義だけでなく、利用方法、支援環境の基本アーキテクチャ、組織への導入モデルの定義を含む体系として定義した。

9.1 PReP モデルの特長と研究成果

高い再利用性

PReP モデルのモデル化方法を評価したところ、成果物の入出力の関連構造によるプロセスモデルが、適用する技術と管理方法とによって一意に決まるため、同等のプロファイルを持つプロジェクト間で、90~100%の再利用率となり、再利用性が高いことがわかった。

高い理解容易性

利用者に対する評価調査を行ったところ、従来のプロセスモデルと比べて、理解容易性が高いことがわかった。また、改善への適用、プロジェクト管理への利用可能性において優れていることが評価の結果からわかった。

プロジェクト計画策定への利用可能性

成果物を、作業者に割り当てられた作業成果物と定義することによって、プロジェクト管理のためのスケジュール作成に必要となる内容と構造とになるため、記述したモデルからプロジェクト管理で使用するガントチャートへの自動変換が可能であることを確認した。

9.2 今後の発展

PReP モデルを基にした適用事例研究として、下記に上げる研究が考えられ今後の発展が期待できる。

9.2.1 PReP モデルのプロセス改善への適用事例研究

プロセスパターン事例の研究(ベストプラクティス)

PReP モデルで記述したプロセスから、パターンとして広く利用可能なプロセスを定義する研究。現在プロセスパターンの分野では、自然言語による Alexandrian Form を用いた開発パターンや組織プロセスパターンの研究が行われているが、自然言語を用いたモデルに代わるプロセスパターンの定義と利用の可能性が考えられる。

プロセス実績分析方法に関する研究

現在のプロセス実績の測定方法、すなわちメトリクスの研究では、主にプロジェクト全体の振る舞いを統計的に分析する方法が主体である。PReP モデルを用いたプロセス実績の評価では、プロジェクト全体に対するトップダウン的な評価ではなく、個々の成果物からなる積み上げによる実績の評価を行うことができ、製造業における統計的生産管理手法の適用の可能性が考えられる。

見積もり手法の研究

プロセス実績の評価と同様に、COCOMO II などによるトップダウン的な見積もりに対して、成果物観点でのボトムアップ型の見積もり手法の研究が考えられる。ボトムアップ型の見積もりを行うことにより、再計画時の見積もりや、進捗の監視をより正確に行うことが可能となると予測できる。

9.2.2 PReP モデルを用いたプロセス中心型環境の研究

PReP モデルを用いたプロセス中心型環境の基本アーキテクチャを、7章に示したが、7章で定義した支援サービスとして、次のような支援環境のインタフェースの開発が考えられる

プロセスモデリング支援

PReP モデルによるプロセスの記述支援インタフェース。構造モデル、および工程モデルの記述を支援する。

プロセス実績シミュレーション

上記のプロセスモデリングの支援に加えて、プロセス実績のデータベースと連動したプロセスのシミュレーションを行うことによるプロセスの最適化支援や、リソース計画の支援が考えられる。また、スケジュールデータの出力なども考えられる。

成果物を媒介とした進捗管理および協調作業支援環境

定義されたプロジェクトのプロセスをもとに、成果物を媒介とした協調作業支援が考えられる。例えば、担当する作業に必要な入力成果物の作業担当者とのコミュニケーション支援や、プロジェクト中での成果物を中心としたコミュニケーションの度合いからの、プロジェクトのコミュニケーション

リスクの判断などが考えられる。

プロジェクト監視支援

定義されたプロジェクトのプロセスをもとに、成果物を媒介とした進捗の監視が考えられる。例えば、成果物ごとの進捗状況を概観できるインタフェースなどである。

プロセス実績評価の支援

成果物単位での実績の測定と収集の支援と合わせた、プロセス実績の評価支援が考えられる。例えば、成果物観点でプロセスを記述することにより、成果物ごとの工数や完成度の進捗に加えて、外部から購入する成果物などの購入実績の進捗なども一貫して取り扱うことができることから、EVM(Earned Value Management)への利用が考えられる。

謝辞

本研究を行うにあたり、終始御指導を賜りました松本健一教授、飯田元助教授に深謝致します。飯田元助教授には、研究に関するご指導に加え、論文の作成においても多大なご指導とご助言、ご協力を頂きました。

本論文の審査委員として加わっていただきました本学の西谷紘一教授には、本論文を完成させるにあたり、貴重なご指導および、論文の価値を高めるための再考を促す機会を与えていただきました。

本研究を開始したときの松本研究室のメンバーであった後藤徹平氏、粕淵健郎氏、岡井洋樹氏からは、本研究を立ち上げる過程で重要となる領域の議論に参加していただきました。

研究の過程で一貫してご協力をいただいた松本研究室の大杉直樹氏からは、本研究の内容に関する貴重なアドバイスを頂きました。

本研究の参加メンバーである松本研究室の岡本圭司氏、村上弘氏、秋永知宏氏、上野秀剛氏、大岡徹也氏、木村隆洋氏、引地一将氏、本村拓也氏には、本研究に関しての多大な協力を頂きました。

最後に、本研究を進めるにあたってご協力をいただいた松本研究室の諸兄、また、社会人学生としての会社の業務と学生生活の両立を支えていただいたソニー株式会社の同僚の方々に深く感謝いたします。

参考文献

- 1) W. W. Gibbs: Software's Chronic Crisis, Scientific American, September, pp.86-95, 1994.
- 2) 高橋 秀俊: ソフトウェア危機 (The Software Crisis), 情報処理学会会誌 Vol.10, No.6, pp.373-374, 1969
- 3) 井上克郎, 松本健一, 飯田元.: ソフトウェアプロセス, 共立出版, pp.1-5, pp.3, pp.27, pp.51, pp.199, 2000.
- 4) M. Paulk, B. Curtis, M. Chrissis, C. Weber. : Capability Maturity Model for Software (Version 1.1), CMU/SEI-93-TR-024, 1993.
- 5) Software Hell, Business Week, http://www.businessweek.com/1999/99_49/b3658015.htm, Dec. 6, 1999
- 6) W. Royce: Managing the Development of Large Software Systems, Proc. 9th International Conference on Software Engineering, CA, pp.328-338, April 1987.
- 7) A. Fuggetta, A. Wolf (Ed): Software Process, International Conference on Software Engineering, Proceedings of the Conference on The Future of Software Engineering, pp.25-34, 2000.
- 8) L. Osterweil: Software Process Are Software Too, Proc. 9th International Conference on Software Engineering, Monterey CA, pp.2-12, April 1987.
- 9) CMMI Product Team : CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, (Version 1.1), CMU/SEI-2002-TR-011/TR-012, 2002.
- 10) S. H. Kan, V. R. Basili and L. N Shaprio: Software quality: an overview from the perspective of total quality management, IBM System Journal, Vol.33, No.1, pp.4-19, 1994.
- 11) 日経エレクトロニクス, ソフトウェアが明暗分ける, 1998年1月号, 1998.
- 12) 今井 正治: システムレベルデザイン:1.システムレベルデザインに向けて, 会誌「情報処理」 Vol.45 No.5, pp.451-455, 2004.
- 13) 日経エレクトロニクス: ソフト危機を救うSoC設計 松下の決断, 2004年10月11日号(no.884), 2004.
- 14) 松原 友夫, :ソフトウェア産業にもデフレがやってくる, Deflation Will Come Soon in Japan's Software Industry, 情報処理学会会誌 Vol.44, No.4, pp.373-,pp.375-385, 2003
- 15) 日経エレクトロニクス: ソフト危機, 日経エレクトロニクス, 1998.3-23, no.712, pp.139-158, 1998.
- 16) 松原友夫・乗松聡: 日本におけるCMM導入を考える, 共立出版 bit, Vol.30, No3, pp 2-8, 1998.3
- 17) 情報処理学会会誌「情報処理」: ソフトウェア管理技術の最新動向を探る, Vol.44 No.4, 2003.4.

- 18) 伊藤 昌夫: SPIとプログラマー:天国あるいは地獄?, SPI Symposium 2001 Fall: 日本版ソフトウェアプロセス改善を考える, 2001.
- 19) 赤坂 幸彦: プロセス改善:成功と危機, SPI Symposium 2001 Fall: 日本版ソフトウェアプロセス改善を考える, 2001.
- 20) CMM / CMMI Maturity Profile, <http://www.sei.cmu.edu/sema/profile.html>
- 21) 田中康: 形骸化しないさせないプロセス記述方法, SEPG Japan 2004, Proc, pp.31-42, 2004.
- 22) 後藤雅史: 組織文化の標準プロセス定着への利用, SEPG Japan 2004, Proc, pp.55-65, 2004.
- 23) ジャケイ愛: 本質を重視したプロセス定義, SEPG Japan 2004, Proc, pp.43-53X, 2004.
- 24) Royce, Winston W.: Managing the Development of Large Software Systems: Concepts and Techniques, Proceedings of IEEE WESCON, page 1-9, IEEE, August, 1970.
- 25) Royce, Winston W.: Managing the Development of Large Software System, Proceedings of the 9th International Conference on Software Engineering, page 328-338, IEEE, August, 1987.
- 26) Boehm, Barry W.: A Spiral Model of Software Development and Enhancement, ACM software Engineering Notes 11(4), pp.14-24, August, 1986.
- 27) J. Verity: Getting Work to Go with the Flow, Business Week, June 21, pp.156-161, 1993.
- 28) 岸田孝一(監修): ソフトウェアプロセスのトレンド, 海文堂出版, 1997.
- 29) Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries (610-1990)
- 30) Watts S. Humphrey. : Managing the Software Process, Addison-Wesley, ISBN-0201180952, 1989. 藤野喜一(監訳): ソフトウェアプロセス成熟度の改善, 日科技連, pp.268-274, 298-300, 1991.
- 31) Watts S. Humphrey, Marc I. Kellner. : Software Process Modeling: Principles of Entity Process Models, CMU/SEI-89-TR-002, 1989.
- 32) B. Curtis, M. Kellner, J. Over: Process Modeling, Communication of ACM, Vol.35, No.9, pp.75-90, September 1992.
- 33) M. Kellner, D. Rombach: Session summary, Comparisons of Software Process Descriptions, Proc, 6th International Software Process Workshop, IEEE Computer Society Press, pp.7-18, Hakodate, Japan, October 1990.
- 34) M. Kellner, et. Al.: Software Process Modeling Example Problem, Proc, 6th International Software Process Workshop, IEEE Computer Society Press, pp.19-29, Hakodate, Japan, October 1990, also in Proc. 1st International Conference on Software Process, IEEE Computer Society Press, pp.176-186, Redondo Beach, CA, October 1991.
- 35) Basili, V. R., and A. J. Turner. : Iterative Enhancement: A Practical Technique for Software Development, IEEE Transaction on Software Engineering, SE-1(4), pp.390-396, December, 1975.
- 36) Radice, R.: A Programming process architecture, IBM Systems Journal, 24, no. 2. 1985.
- 37) 星野芳郎: 星野芳郎著作集第2巻 技術論II, 勁草書房, pp.66-92, 1978
- 38) Jackson, Michael A.: System Development, Prentice-Hall International, Englewood Cliffs, NJ

1983.

39) David A. Marca and Clement L. McGowan, SADT: Structured Analysis and Design Techniques, McGraw-Hill, New York, NY, 1988.

40) Barry W. Boehm, et.: Software Cost Estimation With Cocomo II, Prentice Hall, 2000.

41) G. Karner, "Use Case Points - Resource Estimation for Objectory Projects", Objective Systems SF AB(Rational Software), 1993.

42) Perdita Stevens, Rob Pooley: Using UML: Software Engineering with Objects and Components Revised Edition, Addison Wesley, 1999.

43) Hajimu Iida: Pattern-Oriented Approach to Software Process Evolution, IWPSE'99, pp.55-59, 1999.

44) Hajimu Iida and Yasushi Tanaka: A Compositional Process Pattern Framework for Component-based Process Modeling Assistance, Proceedings of the 1st Workshop on Software Development Patterns (SDPP'02).. Technical Report TUM-I0213, Munich University of Technology) ,pp.57-64, 12/2003.

45) Hajimu Iida, Yasushi Tanaka, Ken'ichi Matsumoto.: Daibutsu-den: A Component-based Framework for Organizational Process Asset Utilization, Proceedings of Profes2002, Lecture Notes in Computer Science 2559/2002, pp. 207-219, Springer-Verlag, ISSN: 0302-9743 (2002).

46) Scott W. Ambler: Process Patterns: Building Large-Scale Systems Using Object Technology, SIGS Books, Cambridge University Press, 1998.

47) Scott W. Ambler: More Process Patterns: Developing Large-Scale Systems Using Object Technology, SIGS Books, Cambridge University Press, 1999.

48) E. Gamma, R. Helm, R. Johnson and J. Vlissides: Design Patterns ? Elements of Reusable Object-Oriented Software, Addison Wesley Professional Computing Series, Addison Wesley, 1994 [本井田真一, 吉田和樹(監訳), デザインパターン, ソフトバンク, 1995].

49) C. Alexander: A Pattern Language, Oxford University Press, 1977 [平田翰那(訳), パタン・ランゲージ, 鹿島出版会, 1984].

50) C. Alexander: The Timeless Way of Building, Oxford University Press, 1979 [平田翰那(訳), 時を超えた建設の道, 鹿島出版会, 1994].

51) 田中豊, 垂水共之(編): Windows 版 統計解析ハンドブック ノンパラメトリック法, 共立出版, pp.60-63, 1999.

52) Eliyahu M. Goldratt, Eli Schragenheim, Carol A. Ptak: Necessary but Not Sufficient: A Theory of Constraints. North River Pr. , 2000.

53) European Communication Manufacturers Association, Reference Model for Frameworks of Software Engineering Environments, 3rd ed., Technical Report TR/55, 1993.

54) Hans-Ulrich Kobialka. :Implementing Support for Software Processes in a Process-centered Software Engineering Environment, GMD Research Series, 1988.

付録 A

成果物と関連の種別の組み合わせによる ガントチャートへの展開

果物と関連の種別の組み合わせによるガントチャートへの展開事例を示す。下記事例では、Microsoft Project への出力を想定し、開始、終了の依存関係を定義している。

下記事例は次の構成となっている。

1	展開ルール名
2	関連モデル図
3	タスクの条件および制約 先行タスク： 先行タスクへの依存タイプ： 先行タスクとの間隔： タスクの開始・終了制約：
4	ガントチャートの例 * 下記の例で成果物が並列成果物の場合、「N」の値は3としてある。

成果物と関連の種別の組み合わせによるガントチャートへの展開例

1	通常成果物+通常成果物の入力関連																																																																																														
2	<p>成果物A 成果物B</p>																																																																																														
3	先行タスク：成果物 A 先行タスクへの依存タイプ：終了-開始 先行タスクとの間隔：0 タスクの開始・終了制約：できるだけ早く																																																																																														
4	<table border="1"> <thead> <tr> <th rowspan="2">ID</th> <th rowspan="2">タスク名</th> <th rowspan="2">期間</th> <th colspan="7">4年11月14日</th> <th colspan="7">2004年11月21日</th> <th colspan="7">2004年11月28日</th> </tr> <tr> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th><th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th><th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>成果物A</td> <td>2日</td> <td>■</td><td>■</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>4</td> <td>成果物B</td> <td>2日</td> <td></td><td></td><td>■</td><td>■</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </tbody> </table>				ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日							月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	3	成果物A	2日	■	■																			4	成果物B	2日			■	■																
ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日																																																																														
			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日																																																																								
3	成果物A	2日	■	■																																																																																											
4	成果物B	2日			■	■																																																																																									

1	通常成果物+通常成果物の同期関連																																																																																														
2	<p>成果物C 成果物D</p>																																																																																														
3	先行タスク：成果物 C 先行タスクへの依存タイプ：開始-開始 先行タスクとの間隔：0 タスクの開始・終了制約：できるだけ早く																																																																																														
4	<table border="1"> <thead> <tr> <th rowspan="2">ID</th> <th rowspan="2">タスク名</th> <th rowspan="2">期間</th> <th colspan="7">4年11月14日</th> <th colspan="7">2004年11月21日</th> <th colspan="7">2004年11月28日</th> </tr> <tr> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th><th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th><th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>成果物C</td> <td>2日</td> <td>■</td><td>■</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>6</td> <td>成果物D</td> <td>1日</td> <td>■</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </tbody> </table>				ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日							月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	5	成果物C	2日	■	■																			6	成果物D	1日	■																			
ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日																																																																														
			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日																																																																								
5	成果物C	2日	■	■																																																																																											
6	成果物D	1日	■																																																																																												

1	通常成果物+通常成果物の入力同期合成関連																																																																																																																																																																																					
2																																																																																																																																																																																						
3	<p>先行タスク：成果物 E</p> <p>先行タスクへの依存タイプ：開始-開始</p> <p>先行タスクとの間隔：成果物 E の期間-成果物 F の期間が 0 以上の場合その値を間隔として指定，それ以外の場合 0</p> <p>タスクの開始・終了制約：できるだけ早く</p>																																																																																																																																																																																					
4	<table border="1"> <thead> <tr> <th rowspan="2">ID</th> <th rowspan="2">タスク名</th> <th rowspan="2">期間</th> <th colspan="7">4年11月14日</th> <th colspan="7">2004年11月21日</th> <th colspan="7">2004年11月28日</th> </tr> <tr> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th> <th>日</th><th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th> <th>日</th><th>月</th><th>火</th><th>水</th> </tr> </thead> <tbody> <tr> <td>9</td> <td>成果物E</td> <td>2日</td> <td colspan="7">[Bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>10</td> <td>成果物F</td> <td>3日</td> <td colspan="7">[Bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th rowspan="2">ID</th> <th rowspan="2">タスク名</th> <th rowspan="2">期間</th> <th colspan="7">4年11月14日</th> <th colspan="7">2004年11月21日</th> <th colspan="7">2004年11月28日</th> </tr> <tr> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th> <th>日</th><th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th> <th>日</th><th>月</th><th>火</th><th>水</th> </tr> </thead> <tbody> <tr> <td>7</td> <td>成果物E</td> <td>2日?</td> <td colspan="7">[Bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>8</td> <td>成果物F</td> <td>1日</td> <td colspan="7">[Bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> </tbody> </table>				ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日							月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	9	成果物E	2日	[Bar]																					10	成果物F	3日	[Bar]																					ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日							月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	7	成果物E	2日?	[Bar]																					8	成果物F	1日	[Bar]																				
ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日																																																																																																																																																																					
			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水																																																																																																																																																																			
9	成果物E	2日	[Bar]																																																																																																																																																																																			
10	成果物F	3日	[Bar]																																																																																																																																																																																			
ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日																																																																																																																																																																					
			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水																																																																																																																																																																			
7	成果物E	2日?	[Bar]																																																																																																																																																																																			
8	成果物F	1日	[Bar]																																																																																																																																																																																			

1	並列成果物+通常成果物の入力関連																																																																																																																																																																				
2																																																																																																																																																																					
3	<p>先行タスク：成果物 G (N=3)</p> <p>先行タスクへの依存タイプ：終了-開始</p> <p>先行タスクとの間隔：0</p> <p>タスクの開始・終了制約：できるだけ早く</p>																																																																																																																																																																				
4	<table border="1"> <thead> <tr> <th rowspan="2">ID</th> <th rowspan="2">タスク名</th> <th rowspan="2">期間</th> <th colspan="7">4年11月14日</th> <th colspan="7">2004年11月21日</th> <th colspan="7">2004年11月28日</th> </tr> <tr> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th> <th>日</th><th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th> <th>日</th><th>月</th><th>火</th><th>水</th> </tr> </thead> <tbody> <tr> <td>12</td> <td>成果物G</td> <td>2日</td> <td colspan="7">[Bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>13</td> <td>成果物G-1</td> <td>2日</td> <td colspan="7">[Bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>14</td> <td>成果物G-2</td> <td>2日</td> <td colspan="7">[Bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>15</td> <td>成果物G-3</td> <td>2日</td> <td colspan="7">[Bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>16</td> <td>成果物H</td> <td>2日</td> <td colspan="7">[Bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> </tbody> </table>				ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日							月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	12	成果物G	2日	[Bar]																					13	成果物G-1	2日	[Bar]																					14	成果物G-2	2日	[Bar]																					15	成果物G-3	2日	[Bar]																					16	成果物H	2日	[Bar]																				
ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日																																																																																																																																																				
			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水																																																																																																																																																		
12	成果物G	2日	[Bar]																																																																																																																																																																		
13	成果物G-1	2日	[Bar]																																																																																																																																																																		
14	成果物G-2	2日	[Bar]																																																																																																																																																																		
15	成果物G-3	2日	[Bar]																																																																																																																																																																		
16	成果物H	2日	[Bar]																																																																																																																																																																		

1	並列成果物+通常成果物の同期関連																																																																																																																																																																								
2																																																																																																																																																																									
3	<p>先行タスク：成果物 I (N=3)</p> <p>先行タスクへの依存タイプ：開始-開始</p> <p>先行タスクとの間隔：0</p> <p>タスクの開始・終了制約：できるだけ早く</p>																																																																																																																																																																								
4	<table border="1"> <thead> <tr> <th rowspan="2">ID</th> <th rowspan="2">タスク名</th> <th rowspan="2">期間</th> <th colspan="7">4年11月14日</th> <th colspan="7">2004年11月21日</th> <th colspan="7">2004年11月28日</th> </tr> <tr> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> </tr> </thead> <tbody> <tr> <td>17</td> <td>成果物</td> <td>2日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>18</td> <td>成果物I-1</td> <td>2日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>19</td> <td>成果物I-2</td> <td>2日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>20</td> <td>成果物I-3</td> <td>2日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>21</td> <td>成果物J</td> <td>1日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> </tbody> </table>				ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日							月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	17	成果物	2日	[Blue bar]																					18	成果物I-1	2日	[Blue bar]																					19	成果物I-2	2日	[Blue bar]																					20	成果物I-3	2日	[Blue bar]																					21	成果物J	1日	[Blue bar]																				
ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日																																																																																																																																																								
			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日																																																																																																																																																		
17	成果物	2日	[Blue bar]																																																																																																																																																																						
18	成果物I-1	2日	[Blue bar]																																																																																																																																																																						
19	成果物I-2	2日	[Blue bar]																																																																																																																																																																						
20	成果物I-3	2日	[Blue bar]																																																																																																																																																																						
21	成果物J	1日	[Blue bar]																																																																																																																																																																						

1	並列成果物+通常成果物の入力同期合成関連																																																																																																																																																																																																																																																																																																																																													
2																																																																																																																																																																																																																																																																																																																																														
3	<p>先行タスク：成果物 K (N=3)</p> <p>先行タスクへの依存タイプ：開始-開始</p> <p>先行タスクとの間隔：， 間隔：成果物 K-#の期間-成果物 L の期間が 0 以上の場合その値を間隔として指定，それ以外の場合 0 (ただし，#には数字 1~3 が入ることとする)</p> <p>タスクの開始・終了制約：できるだけ早く</p>																																																																																																																																																																																																																																																																																																																																													
4	<table border="1"> <thead> <tr> <th rowspan="2">ID</th> <th rowspan="2">タスク名</th> <th rowspan="2">期間</th> <th colspan="7">4年11月14日</th> <th colspan="7">2004年11月21日</th> <th colspan="7">2004年11月28日</th> </tr> <tr> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> </tr> </thead> <tbody> <tr> <td>22</td> <td>成果物K</td> <td>2日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>23</td> <td>成果物K-1</td> <td>2日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>24</td> <td>成果物K-2</td> <td>2日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>25</td> <td>成果物K-3</td> <td>2日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>26</td> <td>成果物L</td> <td>1日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th rowspan="2">ID</th> <th rowspan="2">タスク名</th> <th rowspan="2">期間</th> <th colspan="7">4年11月14日</th> <th colspan="7">2004年11月21日</th> <th colspan="7">2004年11月28日</th> </tr> <tr> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> </tr> </thead> <tbody> <tr> <td>27</td> <td>成果物K</td> <td>2日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>28</td> <td>成果物K-1</td> <td>2日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>29</td> <td>成果物K-2</td> <td>2日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>30</td> <td>成果物K-3</td> <td>2日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>31</td> <td>成果物L</td> <td>3日</td> <td colspan="7">[Blue bar]</td> <td colspan="7"></td> <td colspan="7"></td> </tr> </tbody> </table>				ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日							月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	22	成果物K	2日	[Blue bar]																					23	成果物K-1	2日	[Blue bar]																					24	成果物K-2	2日	[Blue bar]																					25	成果物K-3	2日	[Blue bar]																					26	成果物L	1日	[Blue bar]																					ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日							月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	27	成果物K	2日	[Blue bar]																					28	成果物K-1	2日	[Blue bar]																					29	成果物K-2	2日	[Blue bar]																					30	成果物K-3	2日	[Blue bar]																					31	成果物L	3日	[Blue bar]																				
ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日																																																																																																																																																																																																																																																																																																																													
			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日																																																																																																																																																																																																																																																																																																																							
22	成果物K	2日	[Blue bar]																																																																																																																																																																																																																																																																																																																																											
23	成果物K-1	2日	[Blue bar]																																																																																																																																																																																																																																																																																																																																											
24	成果物K-2	2日	[Blue bar]																																																																																																																																																																																																																																																																																																																																											
25	成果物K-3	2日	[Blue bar]																																																																																																																																																																																																																																																																																																																																											
26	成果物L	1日	[Blue bar]																																																																																																																																																																																																																																																																																																																																											
ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日																																																																																																																																																																																																																																																																																																																													
			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日																																																																																																																																																																																																																																																																																																																							
27	成果物K	2日	[Blue bar]																																																																																																																																																																																																																																																																																																																																											
28	成果物K-1	2日	[Blue bar]																																																																																																																																																																																																																																																																																																																																											
29	成果物K-2	2日	[Blue bar]																																																																																																																																																																																																																																																																																																																																											
30	成果物K-3	2日	[Blue bar]																																																																																																																																																																																																																																																																																																																																											
31	成果物L	3日	[Blue bar]																																																																																																																																																																																																																																																																																																																																											

1	通常成果物＋並列成果物の入力関連																																																																																																																																	
2																																																																																																																																		
3	<p>先行タスク：成果物 S</p> <p>先行タスクへの依存タイプ：終了－開始</p> <p>先行タスクとの間隔：0</p> <p>タスクの開始・終了制約：できるだけ早く</p>																																																																																																																																	
4	<table border="1"> <thead> <tr> <th>ID</th> <th>タスク名</th> <th>期間</th> <th>4年11月14日</th> <th>2004年11月21日</th> <th>2004年11月28日</th> </tr> <tr> <th></th> <th></th> <th></th> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th> <th>日</th><th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th> <th>日</th><th>月</th><th>火</th><th>水</th> </tr> </thead> <tbody> <tr> <td>54</td> <td>成果物S</td> <td>2日</td> <td>■</td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td> </tr> <tr> <td>55</td> <td>成果物T</td> <td>2日</td> <td></td><td>■</td><td>■</td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td> </tr> <tr> <td>56</td> <td>成果物T-1</td> <td>2日</td> <td></td><td>■</td><td>■</td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td> </tr> <tr> <td>57</td> <td>成果物T-2</td> <td>2日</td> <td></td><td>■</td><td>■</td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td> </tr> <tr> <td>58</td> <td>成果物T-3</td> <td>2日</td> <td></td><td>■</td><td>■</td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td> </tr> </tbody> </table>	ID	タスク名	期間	4年11月14日	2004年11月21日	2004年11月28日				月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	54	成果物S	2日	■																	55	成果物T	2日		■	■															56	成果物T-1	2日		■	■															57	成果物T-2	2日		■	■															58	成果物T-3	2日		■	■																	
ID	タスク名	期間	4年11月14日	2004年11月21日	2004年11月28日																																																																																																																													
			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水																																																																																																															
54	成果物S	2日	■																																																																																																																															
55	成果物T	2日		■	■																																																																																																																													
56	成果物T-1	2日		■	■																																																																																																																													
57	成果物T-2	2日		■	■																																																																																																																													
58	成果物T-3	2日		■	■																																																																																																																													

1	通常成果物＋並列成果物の同期関連																																																																																																																																	
2																																																																																																																																		
3	<p>先行タスク：成果物 U</p> <p>先行タスクへの依存タイプ：開始－開始</p> <p>先行タスクとの間隔：0</p> <p>タスクの開始・終了制約：できるだけ早く</p>																																																																																																																																	
4	<table border="1"> <thead> <tr> <th>ID</th> <th>タスク名</th> <th>期間</th> <th>4年11月14日</th> <th>2004年11月21日</th> <th>2004年11月28日</th> </tr> <tr> <th></th> <th></th> <th></th> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th> <th>日</th><th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th> <th>日</th><th>月</th><th>火</th><th>水</th> </tr> </thead> <tbody> <tr> <td>59</td> <td>成果物U</td> <td>2日</td> <td>■</td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td> </tr> <tr> <td>60</td> <td>成果物V</td> <td>2日</td> <td></td><td>■</td><td>■</td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td> </tr> <tr> <td>61</td> <td>成果物V-1</td> <td>2日</td> <td></td><td>■</td><td>■</td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td> </tr> <tr> <td>62</td> <td>成果物V-2</td> <td>2日</td> <td></td><td>■</td><td>■</td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td> </tr> <tr> <td>63</td> <td>成果物V-3</td> <td>2日</td> <td></td><td>■</td><td>■</td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td> </tr> </tbody> </table>	ID	タスク名	期間	4年11月14日	2004年11月21日	2004年11月28日				月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	59	成果物U	2日	■																	60	成果物V	2日		■	■															61	成果物V-1	2日		■	■															62	成果物V-2	2日		■	■															63	成果物V-3	2日		■	■																	
ID	タスク名	期間	4年11月14日	2004年11月21日	2004年11月28日																																																																																																																													
			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水																																																																																																															
59	成果物U	2日	■																																																																																																																															
60	成果物V	2日		■	■																																																																																																																													
61	成果物V-1	2日		■	■																																																																																																																													
62	成果物V-2	2日		■	■																																																																																																																													
63	成果物V-3	2日		■	■																																																																																																																													

1	通常成果物+並列成果物の入力同期合成関連																																																																																																																																																																																																																																																																																																																																										
2	<p>成果物W 成果物X</p>																																																																																																																																																																																																																																																																																																																																										
3	<p>先行タスク：成果物 W</p> <p>先行タスクへの依存タイプ：開始-開始</p> <p>先行タスクとの間隔：成果物 W の期間-成果物 X-#の期間が 0 以上の場合その値を 間隔として指定，それ以外の場合 0（ただし，#には数字 1~3 が入ることとする）</p> <p>タスクの開始・終了制約：できるだけ早く</p>																																																																																																																																																																																																																																																																																																																																										
4	<table border="1"> <thead> <tr> <th rowspan="2">ID</th> <th rowspan="2">タスク名</th> <th rowspan="2">期間</th> <th colspan="7">4年11月14日</th> <th colspan="7">2004年11月21日</th> <th colspan="7">2004年11月28日</th> </tr> <tr> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> </tr> </thead> <tbody> <tr> <td>64</td> <td>成果物W</td> <td>2日?</td> <td colspan="7">■</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>65</td> <td>成果物X</td> <td>1日?</td> <td colspan="7">■</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>66</td> <td>成果物X-1</td> <td>1日?</td> <td colspan="7">→■</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>67</td> <td>成果物X-2</td> <td>1日?</td> <td colspan="7">→■</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>68</td> <td>成果物X-3</td> <td>1日?</td> <td colspan="7">→■</td> <td colspan="7"></td> <td colspan="7"></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th rowspan="2">ID</th> <th rowspan="2">タスク名</th> <th rowspan="2">期間</th> <th colspan="7">4年11月14日</th> <th colspan="7">2004年11月21日</th> <th colspan="7">2004年11月28日</th> </tr> <tr> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th><th>日</th> </tr> </thead> <tbody> <tr> <td>64</td> <td>成果物W</td> <td>2日?</td> <td colspan="7">■</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>65</td> <td>成果物X</td> <td>1日?</td> <td colspan="7">■</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>66</td> <td>成果物X-1</td> <td>1日?</td> <td colspan="7">→■</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>67</td> <td>成果物X-2</td> <td>1日?</td> <td colspan="7">→■</td> <td colspan="7"></td> <td colspan="7"></td> </tr> <tr> <td>68</td> <td>成果物X-3</td> <td>1日?</td> <td colspan="7">→■</td> <td colspan="7"></td> <td colspan="7"></td> </tr> </tbody> </table>	ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日							月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	64	成果物W	2日?	■																					65	成果物X	1日?	■																					66	成果物X-1	1日?	→■																					67	成果物X-2	1日?	→■																					68	成果物X-3	1日?	→■																					ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日							月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日	64	成果物W	2日?	■																					65	成果物X	1日?	■																					66	成果物X-1	1日?	→■																					67	成果物X-2	1日?	→■																					68	成果物X-3	1日?	→■																				
ID	タスク名				期間	4年11月14日							2004年11月21日							2004年11月28日																																																																																																																																																																																																																																																																																																																							
		月	火	水		木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日																																																																																																																																																																																																																																																																																																																				
64	成果物W	2日?	■																																																																																																																																																																																																																																																																																																																																								
65	成果物X	1日?	■																																																																																																																																																																																																																																																																																																																																								
66	成果物X-1	1日?	→■																																																																																																																																																																																																																																																																																																																																								
67	成果物X-2	1日?	→■																																																																																																																																																																																																																																																																																																																																								
68	成果物X-3	1日?	→■																																																																																																																																																																																																																																																																																																																																								
ID	タスク名	期間	4年11月14日							2004年11月21日							2004年11月28日																																																																																																																																																																																																																																																																																																																										
			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	木	金	土	日																																																																																																																																																																																																																																																																																																																				
64	成果物W	2日?	■																																																																																																																																																																																																																																																																																																																																								
65	成果物X	1日?	■																																																																																																																																																																																																																																																																																																																																								
66	成果物X-1	1日?	→■																																																																																																																																																																																																																																																																																																																																								
67	成果物X-2	1日?	→■																																																																																																																																																																																																																																																																																																																																								
68	成果物X-3	1日?	→■																																																																																																																																																																																																																																																																																																																																								

1	並列成果物+並列成果物の入力関連																																																																														
2																																																																															
3	<p>先行タスク：成果物 AE</p> <p>先行タスクへの依存タイプ：終了-開始</p> <p>先行タスクとの間隔：0</p> <p>タスクの開始・終了制約：できるだけ早く</p>																																																																														
4	<table border="1"> <thead> <tr> <th>ID</th> <th>タスク名</th> <th>期間</th> <th>4年11月14日</th> <th>2004年11月21日</th> <th>2004年11月28日</th> </tr> <tr> <td></td> <td></td> <td></td> <td>月</td><td>火</td><td>水</td><td>木</td><td>金</td><td>土</td> <td>日</td><td>月</td><td>火</td><td>水</td><td>木</td><td>金</td><td>土</td> <td>日</td><td>月</td><td>火</td><td>水</td> </tr> </thead> <tbody> <tr> <td>96</td> <td>成果物AE</td> <td>3日</td> <td colspan="3">[Blue bar]</td> <td></td> </tr> <tr> <td>97</td> <td>成果物AE-1</td> <td>1日</td> <td>[Blue bar]</td> <td></td> <td></td> </tr> <tr> <td>98</td> <td>成果物AE-2</td> <td>2日</td> <td>[Blue bar]</td> <td>[Blue bar]</td> <td></td> </tr> <tr> <td>99</td> <td>成果物AE-3</td> <td>3日</td> <td>[Blue bar]</td> <td>[Blue bar]</td> <td>[Blue bar]</td> </tr> <tr> <td>100</td> <td>成果物AF</td> <td>3日</td> <td>[Blue bar]</td> <td>[Blue bar]</td> <td>[Blue bar]</td> </tr> <tr> <td>101</td> <td>成果物AF-1</td> <td>1日</td> <td>[Blue bar]</td> <td></td> <td></td> </tr> <tr> <td>102</td> <td>成果物AF-2</td> <td>1日</td> <td>[Blue bar]</td> <td></td> <td></td> </tr> <tr> <td>103</td> <td>成果物AF-3</td> <td>1日</td> <td>[Blue bar]</td> <td></td> <td></td> </tr> </tbody> </table>	ID	タスク名	期間	4年11月14日	2004年11月21日	2004年11月28日				月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	96	成果物AE	3日	[Blue bar]				97	成果物AE-1	1日	[Blue bar]			98	成果物AE-2	2日	[Blue bar]	[Blue bar]		99	成果物AE-3	3日	[Blue bar]	[Blue bar]	[Blue bar]	100	成果物AF	3日	[Blue bar]	[Blue bar]	[Blue bar]	101	成果物AF-1	1日	[Blue bar]			102	成果物AF-2	1日	[Blue bar]			103	成果物AF-3	1日	[Blue bar]					
ID	タスク名	期間	4年11月14日	2004年11月21日	2004年11月28日																																																																										
			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水																																																												
96	成果物AE	3日	[Blue bar]																																																																												
97	成果物AE-1	1日	[Blue bar]																																																																												
98	成果物AE-2	2日	[Blue bar]	[Blue bar]																																																																											
99	成果物AE-3	3日	[Blue bar]	[Blue bar]	[Blue bar]																																																																										
100	成果物AF	3日	[Blue bar]	[Blue bar]	[Blue bar]																																																																										
101	成果物AF-1	1日	[Blue bar]																																																																												
102	成果物AF-2	1日	[Blue bar]																																																																												
103	成果物AF-3	1日	[Blue bar]																																																																												

1	並列成果物+並列成果物の同期関連																																																																														
2																																																																															
3	<p>先行タスク：成果物 AE</p> <p>先行タスクへの依存タイプ：開始-開始</p> <p>先行タスクとの間隔：0</p> <p>タスクの開始・終了制約：できるだけ早く</p>																																																																														
4	<table border="1"> <thead> <tr> <th>ID</th> <th>タスク名</th> <th>期間</th> <th>4年11月14日</th> <th>2004年11月21日</th> <th>2004年11月28日</th> </tr> <tr> <td></td> <td></td> <td></td> <td>月</td><td>火</td><td>水</td><td>木</td><td>金</td><td>土</td> <td>日</td><td>月</td><td>火</td><td>水</td><td>木</td><td>金</td><td>土</td> <td>日</td><td>月</td><td>火</td><td>水</td> </tr> </thead> <tbody> <tr> <td>104</td> <td>成果物AG</td> <td>3日</td> <td colspan="3">[Blue bar]</td> <td></td> </tr> <tr> <td>105</td> <td>成果物AG-1</td> <td>1日</td> <td>[Blue bar]</td> <td></td> <td></td> </tr> <tr> <td>106</td> <td>成果物AG-2</td> <td>2日</td> <td>[Blue bar]</td> <td>[Blue bar]</td> <td></td> </tr> <tr> <td>107</td> <td>成果物AG-3</td> <td>3日</td> <td>[Blue bar]</td> <td>[Blue bar]</td> <td>[Blue bar]</td> </tr> <tr> <td>108</td> <td>成果物AH</td> <td>1日</td> <td>[Blue bar]</td> <td></td> <td></td> </tr> <tr> <td>109</td> <td>成果物AH-1</td> <td>1日</td> <td>[Blue bar]</td> <td></td> <td></td> </tr> <tr> <td>110</td> <td>成果物AH-2</td> <td>1日</td> <td>[Blue bar]</td> <td></td> <td></td> </tr> <tr> <td>111</td> <td>成果物AH-3</td> <td>1日</td> <td>[Blue bar]</td> <td></td> <td></td> </tr> </tbody> </table>	ID	タスク名	期間	4年11月14日	2004年11月21日	2004年11月28日				月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	104	成果物AG	3日	[Blue bar]				105	成果物AG-1	1日	[Blue bar]			106	成果物AG-2	2日	[Blue bar]	[Blue bar]		107	成果物AG-3	3日	[Blue bar]	[Blue bar]	[Blue bar]	108	成果物AH	1日	[Blue bar]			109	成果物AH-1	1日	[Blue bar]			110	成果物AH-2	1日	[Blue bar]			111	成果物AH-3	1日	[Blue bar]					
ID	タスク名	期間	4年11月14日	2004年11月21日	2004年11月28日																																																																										
			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水																																																												
104	成果物AG	3日	[Blue bar]																																																																												
105	成果物AG-1	1日	[Blue bar]																																																																												
106	成果物AG-2	2日	[Blue bar]	[Blue bar]																																																																											
107	成果物AG-3	3日	[Blue bar]	[Blue bar]	[Blue bar]																																																																										
108	成果物AH	1日	[Blue bar]																																																																												
109	成果物AH-1	1日	[Blue bar]																																																																												
110	成果物AH-2	1日	[Blue bar]																																																																												
111	成果物AH-3	1日	[Blue bar]																																																																												

1	並列成果物+並列成果物の入力同期合成関連																																																																																																																																																																																																	
2	<p>成果物AI 成果物AJ</p>																																																																																																																																																																																																	
3	<p>先行タスク：成果物 AI</p> <p>先行タスクへの依存タイプ：開始-開始</p> <p>先行タスクとの間隔：成果物 AI-#の期間-成果物 AJ-#の期間が 0 以上の場合その値を間隔として指定，それ以外の場合 0（ただし，#には数字 1~3 が入ることとする）</p> <p>タスクの開始・終了制約：できるだけ早く</p>																																																																																																																																																																																																	
4	<table border="1"> <thead> <tr> <th rowspan="2">ID</th> <th rowspan="2">タスク名</th> <th rowspan="2">期間</th> <th colspan="5">4年11月14日</th> <th colspan="5">2004年11月21日</th> <th colspan="3">2004年11月28日</th> </tr> <tr> <th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th> <th>日</th><th>月</th><th>火</th><th>水</th><th>木</th><th>金</th><th>土</th> <th>日</th><th>月</th><th>火</th><th>水</th> </tr> </thead> <tbody> <tr> <td>112</td> <td>成果物AI</td> <td>3日</td> <td colspan="3">■</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>113</td> <td>成果物AI-1</td> <td>1日</td> <td>■</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>114</td> <td>成果物AI-2</td> <td>2日</td> <td>■</td> <td>■</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>115</td> <td>成果物AI-3</td> <td>3日</td> <td>■</td> <td>■</td> <td>■</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>116</td> <td>成果物AJ</td> <td>3日</td> <td colspan="3">■</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>117</td> <td>成果物AJ-1</td> <td>1日</td> <td>■</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>118</td> <td>成果物AJ-2</td> <td>1日</td> <td>→■</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>119</td> <td>成果物AJ-3</td> <td>1日</td> <td>→■</td> <td>■</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	ID	タスク名	期間	4年11月14日					2004年11月21日					2004年11月28日			月	火	水	木	金	土	日	月	火	水	木	金	土	日	月	火	水	112	成果物AI	3日	■																	113	成果物AI-1	1日	■																	114	成果物AI-2	2日	■	■																115	成果物AI-3	3日	■	■	■															116	成果物AJ	3日	■																	117	成果物AJ-1	1日	■																	118	成果物AJ-2	1日	→■																	119	成果物AJ-3	1日	→■	■															
ID	タスク名				期間	4年11月14日					2004年11月21日					2004年11月28日																																																																																																																																																																																		
		月	火	水		木	金	土	日	月	火	水	木	金	土	日	月	火	水																																																																																																																																																																															
112	成果物AI	3日	■																																																																																																																																																																																															
113	成果物AI-1	1日	■																																																																																																																																																																																															
114	成果物AI-2	2日	■	■																																																																																																																																																																																														
115	成果物AI-3	3日	■	■	■																																																																																																																																																																																													
116	成果物AJ	3日	■																																																																																																																																																																																															
117	成果物AJ-1	1日	■																																																																																																																																																																																															
118	成果物AJ-2	1日	→■																																																																																																																																																																																															
119	成果物AJ-3	1日	→■	■																																																																																																																																																																																														

付録 B

PReP モデルの DTD 定義

PReP モデルの DTD 記述例

図 28 に示した工程モデルの XML 記述での DTD 構造図での DTD 記述例を示す.

```
<!-- LifeCycleModel -->
<!ELEMENT LifeCycleModel (Artifacts, Tasks, Phases, Relations)>
<!ATTLIST LifeCycleModel ID CDATA #REQUIRED
                Name CDATA #REQUIRED
                Outline CDATA #IMPLIED
                Link CDATA #IMPLIED>

<!-- Artifacts -->
<!ELEMENT Artifacts (Artifact+)>

<!-- Artifact -->
<!ELEMENT Artifact (NumberOfModule, ModuleDenominationID, Estimations)>
<!ATTLIST Artifact ID CDATA #REQUIRED
                Name CDATA #REQUIRED
                Outline CDATA #IMPLIED
                Link CDATA #IMPLIED>

<!-- NumberOfModules -->
<!ELEMENT NumberOfModules (#PCDATA)>

<!-- ModuleDenominationID -->
<!ELEMENT Denomination (#PCDATA|Value)*>
<!-- Value -->
<!ELEMENT Value (#PCDATA)>

<!-- Estimations -->
<!ELEMENT Estimations (Cost, Effort, Quality, Size)>
<!ELEMENT Cost (#PCDATA)>
<!ELEMENT Effort (#PCDATA)>
<!ELEMENT Quality (#PCDATA)>
<!ELEMENT Size (#PCDATA)>

<!-- Task -->
<!ELEMENT Task (RelatedArtifact, RequiredResources, StepInformations, AvailableTemplates,
TEstimations)>
<!ATTLIST Task ID CDATA #REQUIRED
                Name CDATA #REQUIRED
                Outline CDATA #IMPLIED>
```

```

                Link                CDATA #IMPLIED>

<!-- RequiredResources -->
<!ELEMENT RequiredResources (RequiredResource*)>
<!-- RequiredResource -->
<!ATTLIST RequiredResource ID                CDATA #REQUIRED
                             Name              CDATA #REQUIRED
                             Outline           CDATA #IMPLIED
                             Link              CDATA #IMPLIED>

<!-- ProcInformations -->
<!ELEMENT ProcInformations (ProcInformation*)>
<!-- ProcInformation -->
<!ATTLIST ProcInformation ID                CDATA #REQUIRED
                             Name              CDATA #REQUIRED
                             Outline           CDATA #IMPLIED
                             Link              CDATA #IMPLIED>

<!-- AvailableTemplates -->
<!ELEMENT AvailableTemplates (AvailableTemplate*)>
<!-- AvailableTemplate -->
<!ATTLIST AvailabeTemplate ID                CDATA #REQUIRED
                             Name              CDATA #REQUIRED
                             Outline           CDATA #IMPLIED
                             Link              CDATA #IMPLIED>

<!-- EstimationEquations -->
<!ELEMENT EstimationEquations (TCost, TEffort, TQuality)>
<!-- EEOfcost -->
<!ELEMENT EEOfcost (#PCDATA)>
<!-- EEOfeffort -->
<!ELEMENT EEOfeffort (#PCDATA)>
<!-- EEOfquality -->
<!ELEMENT EEOfquality (#PCDATA)>

<!-- Phases -->
<!ELEMENT Phases (Phase+)>
<!-- Phase -->
<!ELEMENT Phase (NumberOfIteration, IncludedArtifacts, MilestoneTypeID, NextPhaseID)>
<!ATTLIST Phase ID                CDATA #REQUIRED
                 Name              CDATA #REQUIRED
                 Outline           CDATA #IMPLIED
                 Link              CDATA #IMPLIED>

<!-- NumberOfIteration -->
<!ELEMENT NumberOfIteration (#PCDATA)>

<!-- IncludedArtifacts -->
<!ELEMENT IncludedArtifacts (IncludedArtifact+)>
<!-- IncludedArtifact -->
<!ELEMENT IncludedArtifact (ArtifactID, IsMilestone, AppliedTask)>
<!-- ArtifactID -->
<!ELEMENT ArtifactID (#PCDATA)>
<!-- IsMilestone -->

```

```

<!ELEMENT IsMilestone (#PCDATA)>

<!-- NextPhaseID -->
<!ELEMENT NextPhaseID (#PCDATA|Value)*>

<!-- Relations -->
<!ELEMENT Relations (Relation*)>
<!-- Relation -->
<!ELEMENT Relation (LinkType, StartingPoint, EndPoint)>
<!ATTLIST Relation  ID          CDATA #REQUIRED
                    Name       CDATA #REQUIRED
                    Outline     CDATA #IMPLIED
                    Link        CDATA #IMPLIED>
<!-- LinkType -->
<!ELEMENT LinkType (#PCDATA)>

<!-- StartingPoint -->
<!ELEMENT StartingPoint (SArtifactID, SMultiplicity)>
<!-- SArtifactID -->
<!ELEMENT SArtifactID (#PCDATA)>
<!-- SMultiplicity -->
<!ELEMENT SMultiplicity (#PCDATA)>

<!-- EndPoint -->
<!ELEMENT EndPoint (EArtifactID, EMultiplicity)>
<!-- EArtifactID -->
<!ELEMENT EArtifactID (#PCDATA)>
<!-- EMultiplicity -->
<!ELEMENT EMultiplicity (#PCDATA)>

```

DTD での各要素の定義

図 28 および前項で示した DTD の各要素の定義を下記の表記に従って示す。

要素名:(要素名)

親要素 : (親要素)
子要素 : (値以外の子要素)
定義回数: (必要な定義回数)
要素説明: (要素の説明)
内容の型: (子要素を除いた内容の型)
内容説明: (子要素を除いた内容の説明)
属性名 : (属性名)
属性の型: (属性の型)
属性説明: (属性の説明)

要素名:LifeCycleModel

親要素 : なし
子要素 : Artifacts, Phases
定義回数: 1回
要素説明: 本スキーマの最上位構造
内容の型: なし
内容説明: なし
属性名 : ID, Name, Outline, Link
属性の型: ID → CDATA #REQUIRED (必須文字列)
Name → CDATA #REQUIRED (必須文字列)
Outline → CDATA #IMPLIED (暗黙文字列)
Link → CDATA #IMPLIED (暗黙文字列)
属性説明: ID → 要素特定に用いられる一意な識別子, 最上位の親要素の Name 属性から
当該要素の属性 Name まで”. ”でそれらを結んだ表記を用いる
Name → 要素に与えられる名称
Outline → 要素の概要を示す
Link → XML のリンク機能を用い, 要素に対するリソースとの関連をこの属性で取り
扱う

要素名:Artifacts

親要素 :LifeCycleModel

子要素 :Artifact

定義回数:1回以上

要素説明:複数の成果物を取り纏める要素

内容の型:なし

内容説明:なし

属性名 :なし

属性の型:なし

要素名:Artifact

親要素 :Artifacts

子要素 :NumberOfModules, ModuleDenominationID, Estimations, Task

定義回数:1回以上

要素説明:成果物/タスクに対応する要素. 当該要素の再利用に必要な値を含む

内容の型:なし

内容説明:なし

属性名 :ID, Name, Outline, Link

属性の型:ID -> CDATA #REQUIRED (必須文字列)

Name -> CDATA #REQUIRED (必須文字列)

Outline -> CDATA #IMPLIED (暗黙文字列)

Link -> CDATA #IMPLIED (暗黙文字列)

属性説明:要素 PReP の属性説明を参照

要素名:NumberOfModules

親要素 :Artifact

子要素 :なし

定義回数:1回

要素説明:成果物のモジュール分割数を内容とする.

内容の型:#PCDATA (文字列)

内容説明:成果物のモジュール分割数を内容とする. 1 以上の正数若しくは”N”で表記する

属性名 :なし

属性の型:なし

属性説明:なし

要素名:ModuleDenominationID

親要素 :Artifact

子要素 :なし

定義回数:0 回以上

要素説明:単位名称の ID を内容とする

内容の型:#PCDATA(文字列)

内容説明:なし

属性名 :なし

属性の型:なし

属性説明:なし

要素名:Estimations

親要素 :Artifact

子要素 :Cost, Effort, Quality, Size

定義回数:1回

要素説明:当該成果物に対する見積もり値を取り纏める要素

内容の型:なし

内容説明:なし

属性名 :なし

属性の型:なし

属性説明:なし

要素名:Cost

親要素 :Estimations

子要素 :なし

定義回数:1回

要素説明:当該成果物の作成に必要な, 利用者によって見積もられたコストの値を内容とする

内容の型:#PCDATA(文字列)

内容説明:当該成果物の作成に必要な, 利用者によって見積もられたコストの値

属性名 :なし

属性の型:なし

属性説明:なし

要素名:Effort

親要素 :Estimations

子要素 :なし

定義回数:1回

要素説明:当該成果物の作成に必要な,利用者によって見積もられた工数の値を内容とする

内容の型:#PCDATA(文字列)

内容説明:当該成果物の作成に必要な,利用者によって見積もられた工数の値

属性名 :なし

属性の型:なし

属性説明:なし

要素名:Quality

親要素 :Estimations

子要素 :なし

定義回数:1回

要素説明:当該成果物に対して利用者によって見積もられた品質の値を内容とする

内容の型:#PCDATA(文字列)

内容説明:当該成果物に対して見積もられた品質の値

属性名 :なし

属性の型:なし

属性説明:なし

要素名:Size

親要素 :Estimations

子要素 :なし

定義回数:1回

要素説明:当該成果物に対して利用者によって見積もられた規模の値を内容とする

内容の型:#PCDATA(文字列)

内容説明:当該成果物に対して利用者によって見積もられた規模の値

属性名 :なし

属性の型:なし

属性説明:なし

要素名:Task

親要素 :Artifact

子要素:RequiredResources, AvailableTemplates, ProcInformations, EstimationEquation, Relations

定義回数:0 回以上

要素説明:成果物に適用する標準タスクを内容とする

内容の型:#PCDATA(文字列)

内容説明:成果物に適用する標準タスク

属性名 :ID, Name, Outline, Link

属性の型:ID -> CDATA #REQUIRED (必須文字列)

Name -> CDATA #REQUIRED (必須文字列)

Outline -> CDATA #IMPLIED (暗黙文字列)

Link -> CDATA #IMPLIED (暗黙文字列)

属性説明:要素 PReP の属性説明を参照

要素名:RequiredResources

親要素 :Task

子要素 :RequiredResource

定義回数:1回

要素説明:タスクを適用した場合に成果物の作成に必要なリソース要件を取り纏める要素

内容の型:なし

内容説明:なし

属性名 :なし

属性の型:なし

属性説明:なし

要素名:RequiredResource

親要素 :Task

子要素 :なし

定義回数:0 回以上

要素説明:タスクを適用した場合に成果物の作成に必要なリソース要件を属性値から指定する

内容の型:#PCDATA(文字列)

内容説明:なし

属性名 :ID, Name, Outline, Link

属性の型:ID -> CDATA #REQUIRED (必須文字列)

Name -> CDATA #REQUIRED (必須文字列)

Outline -> CDATA #IMPLIED (暗黙文字列)

Link -> CDATA #IMPLIED (暗黙文字列)

属性説明:ID -> タスクを適用した場合に成果物の作成に必要なリソース要件へのID

Name -> タスクを適用した場合に成果物の作成に必要なリソース要件の名称

Outline -> タスクを適用した場合に成果物の作成に必要なリソース要件の概要

Link -> 要素 PReP の属性説明を参照

要素名:ProcInformations

親要素 :Task

子要素 :ProcInformation

定義回数:1回

要素説明:タスクに関連する作業手順を取りまとめる要素

内容の型:#PCDATA(文字列)

内容説明:なし

属性名 :なし

属性の型:なし

属性説明:なし

要素名:ProcInformation

親要素 :ProcInformations

子要素 :なし

定義回数:0 回以上

要素説明:タスクに関連する作業手順を属性値から指定する要素

内容の型:なし

内容説明:なし

属性名 :ID, Name, Outline, Link

属性の型:ID -> CDATA #REQUIRED (必須文字列)

Name -> CDATA #REQUIRED (必須文字列)

Outline -> CDATA #IMPLIED (暗黙文字列)

Link -> CDATA #IMPLIED (暗黙文字列)

属性説明:ID -> タスクに関連する作業手順へのID

Name -> タスクに関連する作業手順の名称

Outline -> タスクに関連する作業手順の概要

Link -> 要素 PReP の属性説明を参照

要素名:AvailableTemplates

親要素 :Task

子要素 :AvailableTemplate

定義回数:1回

要素説明:タスクを適用した場合に成果物の作成に利用可能なテンプレートを取り纏める要素

内容の型:なし

内容説明:なし

属性名 :なし

属性の型:なし

属性説明:なし

要素名: AvailableTemplate

親要素 : AvailableTemplates

子要素 : なし

定義回数: 0 回以上

要素説明: タスクを適用した場合に成果物の作成に利用可能なテンプレートを属性値から指定する要素

内容の型: なし

内容説明: なし

属性名 : ID, Name, Outline, Link

属性の型: ID → CDATA #REQUIRED (必須文字列)

Name → CDATA #REQUIRED (必須文字列)

Outlin → CDATA #IMPLIED (暗黙文字列)

Link → CDATA #IMPLIED (暗黙文字列)

属性説明: ID → タスクに利用可能なテンプレートへのID

Name → タスクに利用可能なテンプレートの名称

Outline → タスクに利用可能なテンプレートの概要

Link → 要素 PReP の属性説明を参照

要素名: EstimationEquation

親要素 : Task

子要素 : EEOfCost, EEOfEffort, EEOfQuality

定義回数: 1回

要素説明: タスクによって計算される, 成果物作成に必要な見積もり値を取りまとめる要素

内容の型: なし

内容説明: なし

属性名 : なし

属性の型: なし

属性説明: なし

要素名: EEOfCost

親要素 : Task

子要素 : なし

定義回数: 1回

要素説明: タスクによって計算される, 目標とする成果物作成に必要なコストの見積もり値を内容とする

内容の型: #PCDATA (文字列)

内容説明: タスクによって計算される, 目標とする成果物作成に必要なコストの見積もり値

属性名 : なし

属性の型: なし

属性説明: なし

要素名 : EEOfEffort

親要素 : Task

子要素 : なし

定義回数: 1回

要素説明: タスクによって計算される, 目標とする成果物作成に必要な工数の見積もり値を内容とする

内容の型: #PCDATA (文字列)

内容説明: タスクによって計算される, 目標とする成果物作成に必要な工数の見積もり値

属性名 : なし

属性の型: なし

属性説明: なし

要素名 : EEOfQuality

親要素 : Task

子要素 : なし

定義回数: 1回

要素説明: タスクによって計算される, 目標とする成果物作成に必要な品質の見積もり値を内容とする

内容の型: #PCDATA (文字列)

内容説明: タスクによって計算される, 目標とする成果物作成に必要な品質の見積もり値

属性名 : なし

属性の型: なし

属性説明: なし

要素名 : Relation

親要素 : Task

子要素 : LinkType, StartingPoint, EndPoint

定義回数: 1回

要素説明: 成果物間の関連の情報を子要素として取りまとめる要素

内容の型: なし

内容説明: なし

属性名 : ID, Name, Outline, Link

属性の型: ID -> CDATA #REQUIRED (必須文字列)

Name -> CDATA #REQUIRED (必須文字列)

Outline -> CDATA #IMPLIED (暗黙文字列)

Link -> CDATA #IMPLIED (暗黙文字列)

属性説明: ID -> 関連への ID

Name -> 関連の名称. (始点成果物 ID + “>>” + 終点成果物 ID)

Outline -> 関連の概要

Link -> 要素 PReP の属性説明を参照

要素名:LinkType

親要素 :Relation

子要素 :なし

定義回数:1回

要素説明:成果物間の関連の種別に関する内容を含む要素

内容の型:#PCDATA(文字列)

内容説明:成果物間の関連の種別を内容として数字で表現する。(1:入力関係, 2:同期関係, 3:
入力と同期の合

成関係)

属性名 :なし

属性の型:なし

属性説明:なし

要素名:StartingPoint

親要素 :Relation

子要素 :SArtifactID, SMultiplicity

定義回数:1回

要素説明:成果物間の関連の, 始点側に関する情報を取りまとめる要素

内容の型:なし

内容説明:なし

属性名 :なし

属性の型:なし

属性説明:なし

要素名:SArtifactID

親要素 :StartingPoint

子要素 :なし

定義回数:1回

要素説明:成果物間の関連の, 始点側の成果物の(属性) ID を内容とする要素

内容の型:#PCDATA

内容説明:成果物間の関連の, 始点側の成果物の(属性) ID

属性名 :なし

属性の型:なし

属性説明:なし

要素名:SMultiplicity

親要素 :StartingPoint

子要素 :なし

定義回数:1回

要素説明:成果物間の関連の, 始点側の多重度を内容とする要素

内容の型:#PCDATA

内容説明:成果物間の関連の, 始点側の多重度. (“1”又は”n”で表記)

属性名 :なし

属性の型:なし

属性説明:なし

要素名:EndPoint

親要素 :Relation

子要素 :EArtifactID, EMultiplicity

定義回数:1回

要素説明:成果物間の関連の, 終点側に関する情報を取りまとめる要素.

内容の型:なし

内容説明:なし

属性名 :なし

属性の型:なし

属性説明:なし

要素名:EArtifactID

親要素 :EndPoint

子要素 :なし

定義回数:1回

要素説明:成果物間の関連の, 終点側の成果物の(属性) ID を内容とする要素.

内容の型:#PCDATA

内容説明:成果物間の関連の, 終点側の成果物の(属性) ID

属性名 :なし

属性の型:なし

属性説明:なし

要素名:EMultiplicity

親要素 :EndPoint

子要素 :なし

定義回数:1回

要素説明:成果物間の関連の, 終点側の多重度を内容とする要素

内容の型:#PCDATA

内容説明:成果物間の関連の, 終点側の多重度. (“1”又は”n”で表記)

属性名 :なし

属性の型:なし

属性説明:なし

要素名: Phases

親要素 : LifeCycleModel

子要素 : なし

定義回数: 1回

要素説明: ライフサイクルモデル展開時の、フェーズを取りまとめる要素

内容の型: なし

内容説明: なし

属性名 : なし

属性の型: なし

属性説明: なし

要素名: Phase

親要素 : Phases

子要素 : NumberOfIteration, IncludedArtifacts, NextPhaseID

定義回数: 1回以上

要素説明: ライフサイクルモデル展開時の、フェーズに関する情報を取りまとめる要素

内容の型: なし

内容説明: なし

属性名 : ID, Name, Outline, Link

属性の型: ID → CDATA #REQUIRED (必須文字列)

Name → CDATA #REQUIRED (必須文字列)

Outline → CDATA #IMPLIED (暗黙文字列)

Link → CDATA #IMPLIED (暗黙文字列)

属性説明: ID → Phase のID

Name → (Phase の名称)

Outline → Phase の概要

Link → 要素 PReP の属性説明を参照

要素名: NumberOfIteration

親要素 : Phase

子要素 : なし

定義回数: 1回

要素説明: フェーズのイテレーション回数を内容とする

内容の型: #PCDATA

内容説明: フェーズのイテレーション回数.

属性名 : なし

属性の型: なし

属性説明: なし

要素名:IncludedArtifacts

親要素 :Phase

子要素 :IncludedArtifact

定義回数:1回

要素説明:当該フェーズに含まれる各成果物の(非再利用)情報を取りまとめる要素を子要素とする要素

内容の型:なし

内容説明:なし

属性名 :なし

属性の型:なし

属性説明:なし

要素名:IncludedArtifact

親要素 :IncludedArtifacts

子要素 :ArtifactID, IsMilestone

定義回数:1回以上

要素説明:当該フェーズに含まれる成果物の(非才利用)情報を取りまとめる要素

内容の型:なし

内容説明:なし

属性名 :なし

属性の型:なし

属性説明:なし

要素名:ArtifactID

親要素 :IncludedArtifact

子要素 :なし

定義回数:1回以上

要素説明:当該成果物の(属性) ID を内容とする要素

内容の型:#PCDATA

内容説明:当該フェーズに含まれる当該成果物の ID を内容とする

属性名 :なし

属性の型:なし

属性説明:なし

要素名: IsMilestone

親要素 : IncludedArtifact

子要素 : なし

定義回数: 1回

要素説明: 当該成果物がマイルストーン成果物か否かの判定項目を内容とする要素

内容の型: #PCDATA

内容説明: 当該成果物がマイルストーン成果物か否かの判定項目を内容とする. マイルストーンである場合は”1”そうでない場合は”0”を指定する.

属性名 : なし

属性の型: なし

属性説明: なし

要素名: NextPhaseID

親要素 : Phase

子要素 : なし

定義回数: 1回

要素説明: 当該フェーズの次フェーズへの ID を内容とする要素

内容の型: #PCDATA (文字列)

内容説明: 当該フェーズの次フェーズの ID を含む. 属性”ID”の表記方法に倣う

属性名 : なし

属性の型: なし

属性説明: なし

プロセス改善のための成果物観点によるソフトウェアプロセスモデル化手法

2005年3月

本論文は奈良先端科学技術大学院大学情報科学研究科に博士(工学)授与の要件として提出した博士論文である。

田中 康

奈良先端科学技術大学院大学

情報科学研究科 情報システム学専攻
