

## 現場作業に密着したテーラリングを実現する

### 「標準プロセスの構造及びプロセス定義」

#### Standard process and process architecture which enable to tailor for development site closely

主査	： 阪本 太志	東芝デジタルメディアエンジニアリング(株)
副主査	： 三浦 邦彦	矢崎総業(株)
副主査	： 中森 博晃	パナソニック ファクトリーソリューションズ(株)
リーダー	： 内山 哲三	ビジネスキューブ・アンド・パートナーズ(株)
研究員	： 岩田 英成	T I S (株)
	宇野 俊治	S C S K (株)
	久保 敬二郎	富士通九州ネットワークテクノロジーズ(株)
	田中 宏明	(株)東京ビジネスソリューション
	松尾 美貴	富士通九州ネットワークテクノロジーズ(株)

#### 1. 研究概要

現在、CMMI<sup>®</sup>[01]や ISO/IEC 15504 などのプロセス改善モデルの普及により、標準プロセスを構築しているソフトウェア開発組織が増えて来ている。その中で、組織内のプロジェクトは、組織標準プロセスをテーラリングし、プロジェクトのプロセスを定義する事が求められているが、「標準プロセス - プロジェクトのプロセス」間、または「プロジェクトのプロセス - プロジェクトで計画された業務」間に大きな乖離が生じている。

このため、プロジェクト実績に基づき、プロセスの分析及び評価を行い、組織的なプロセス改善を継続的に行う事が困難になっている場合が多い。

本研究では、標準プロセスとプロジェクトの関係をより明確にするための、「標準プロセスの構造及びプロセス定義方法」について考察を行った。

#### Abstract

Currently, software development organizations which establish standard process increase by Process Models (CMMI<sup>®</sup>[01] or ISO/IEC-15504, etc.) spread. Projects in the organization have to define project's process tailored from organization's standard process, but there are big gaps between standard process and project's process or between project's process and planned project's tasks.

Therefore, it is too difficult to conduct organizational process improvement continuously based on project's performance.

This research focuses on standard process architecture and process definition method to clarify relationship between standard process and projects.

#### 2. テーマ選定の理由

本分科会のメンバーは、ソフトウェア開発者、品質保証担当、そして SEPG<sup>SM</sup>で構成されており、それぞれの立場で、組織の標準プロセス及びテーラリングに対し、以下のような課題を持っている。

(1)ソフトウェア開発者

- エンハンス開発が主流になっている中で、新規開発を想定した標準プロセスに従った開発を行う事が難しい。
- 短納期開発が増えている中で、重厚な組織標準プロセスに準拠する事が出来ない。
- 組織標準プロセスと実際のプロセスに大きな乖離があるが、テーラリングに多くの時間を費やしたくない。

(2)品質保証担当

- 実際のプロジェクトは、イテレーション型やアジャイルに傾倒した開発を行っているため、「ウォーターフォール型の組織標準プロセスには従わない」と主張する。
- ウォーターフォール型の組織標準プロセスで定義されたタイミングで、工程移行判定が実施されなくなっている。

(3)SEPG<sup>SM</sup>

- CMMI<sup>®</sup>[01]やISO/IEC 15504などの国際規格準拠を目指すあまり、プロジェクトにとって有意性のない成果物の作成を求めてしまっている。
- 組織標準プロセスで定義されたタスクを変更・追加・削除する事をテーラリング基準としているが、プロジェクトではテーラリング基準や標準プロセスそのものを無視した開発を行っている。
- 多種多様なプロジェクトの実績を、画一的な組織標準プロセスに従って分析・評価を行っても、バラつきが大きすぎて有効なデータを得る事が出来ない。

これらの課題は、組織標準プロセスのあり方とテーラリング手法が、現在のソフトウェア開発スタイルに適合出来ていない事が原因であると考え、上記課題の解決策を研究する事をテーマとして選定した。

3. 活動目標

組織標準のあり方とテーラリング手法を研究するにあたり、下記目標を設定した。

- (1)現場に負荷をかけない、より実態に近い組織標準プロセスの構造を定義する。
- (2)イテレーションやアジャイルなど、多種多様な開発スタイルに柔軟に対応出来る「組織標準プロセスの構造」と「テーラリング手法」を考える。
- (3)十分な品質保証が出来るソフトウェア開発組織の醸成を可能にする。

4. 活動内容

活動は、定例会8回に、臨時会3回を追加開催し、会合をメインに行った。

表 1. 活動内容

日程	内容
2013年5月	定例会：分科会メンバーの課題共有
2013年6月	定例会：研究テーマの洗い出し
2013年7月	定例会：研究テーマの選定
2013年7月	臨時会：各社の標準プロセスの共有、共通課題の特定
2013年9月	定例会：研究テーマの範囲、研究アプローチの決定
2013年10月	定例会：標準プロセス構造検討、テーラリング指針定義
2013年11月	定例会：静的プロセス構造の定義、動的プロセス構造の定義
2013年11月	臨時会：組織標準プロセスの構造とテーラリングの概念図作成
2013年12月	定例会：報告書草稿作成
2013年1月	臨時会：報告書レビュー
2014年1月	定例会：報告書まとめ

## 5. 研究成果及び考察

### 5.1 現状整理と課題の認識

#### (1)現状整理

われわれは本テーマを議論する中で、現在の標準プロセスの主流となっているウォーターフォール型モデルが、実際のプロジェクトの開発スタイルに適用しづらくなっている事に着目した。

##### ①開発スタイルの進化

IT技術の進歩に伴い、開発環境を共有していた時代に比べると現場の開発環境は大きく飛躍し、開発者個人の端末を用いて設計・開発・テストが一貫して行えるような状態にある。また、ソフトウェアが実現出来る機能範囲の拡大により、ユーザーからの要求が高度化している。さらに、高機能を短納期に開発するために、アジャイル開発を代表とする、小さい工程を反復して開発を行うイテレーション型、分業制による並行開発などの新しい開発スタイルを取り入れるプロジェクトが増えてきた。

##### ②組織標準プロセスの不遵守

開発スタイルの進化に従って、ウォーターフォール型モデルを前提とした組織標準プロセスそのものが適用しづらくなっており、開発現場は、組織標準プロセスやテラリング基準を無視する傾向にある。

#### (2)問題点と課題

開発スタイルの進化、組織標準プロセスの不遵守により、以下の問題点、課題が考えられる。

##### ①柔軟な組織標準プロセスの必要性

今後も新たな開発スタイルが増えてくる度に組織標準プロセスを一から作り上げる事は現実的に厳しい。このため、どんな開発スタイルでも適用出来る柔軟な組織標準プロセスが必要である。

##### ②プロジェクト特性に適合したテラリング基準の確立

CMMI<sup>®[01]</sup>やISO/IEC 15504などの国際標準に適合させようとするあまり、テラリング幅が狭い画一的な標準プロセスとなり、実態にあったテラリングが実施出来ない。小規模・短納期などのプロジェクトによっては、品質を担保する証跡を残すための対応、膨大且つ重厚な成果物の承認など、通常の開発業務に加え、必要以上に品質を証明するための業務が重くのしかかっている。このため、プロジェクトに必要以上に負荷をかけない、実態に合ったプロセス定義が必要である。

##### ③組織的改善活動の仕組み確立

①②に起因して、開発現場は、組織標準プロセスやテラリング基準を適用する事が困難となっている。これにより、プロジェクトの成果や実績が組織標準プロセスの実績になっておらず、それらを組織標準プロセスやテラリング基準の改善に活かす事が出来ない。このため、開発スタイルごとに実績を残しフィードバック出来る仕組みが必要である。

### 5.2 対策案検討

前述した課題を解決するために考案した「組織標準プロセスの構造とテラリングの概念」を図1に示す。

まず、様々な開発スタイルに対応可能とするため、各プロセスを部品化し「静的プロセス」として定義する。次に、「静的プロセス」を開発スタイルに合わせて選択し、時系列を考慮して構築したライフサイクルを「動的プロセス」と定義する。プロジェクトは、「動的プロセス」を選択するための指針として、「選択指針」を利用する。選択指針とは、プロジェクトがライフサイクルを選択するための指針である。また、定義された「動的プロセス」が適切でない場合、独自に定義する事も可能とする。

プロジェクトは、「テーラリング指針」に基づいて、選択したライフサイクルをプロジェクト特性に合わせてテーラリングする。

これらの適用結果であるプロジェクト実績情報を収集し、上記構成要素の改善に活かす。

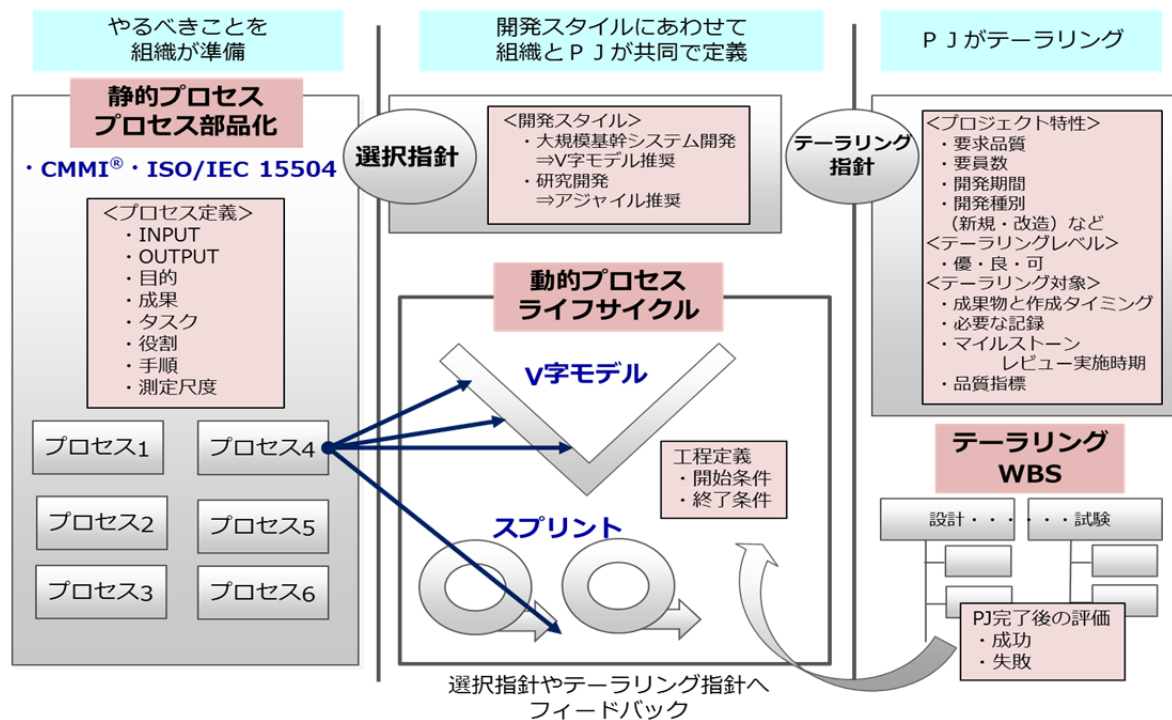


図1. 組織標準プロセスの構造とテーラリングの概念

### 5.2.1 静的プロセスと動的プロセスの定義による柔軟な組織標準プロセスの構築

組織標準プロセスを準備するというと、とかく開発スタイル別(たとえば、ウォーターフォール用・アジャイル用など)に定義しがちだが、これでは、開発スタイルが増えるたびに標準を用意する必要があり、煩雑である。よって、今後新たに増えるであろう、開発スタイルにも柔軟に対応出来る仕組みの検討を行った。まず、代表的な開発スタイルであるウォーターフォールとアジャイル開発のプロセスを比較し、その違いを検討してみた。例えば、どちらも開発作業において品質分析を行うという面では同じだが、どういうタイミングで行うか(工程ごと、リリースごとなど)に違いがある。つまり「やるべき事」は共通であり、実施タイミングに違いがある。この「やるべき事」を「静的プロセス」として定義し、実施タイミングを「動的プロセス」として定義する。この考え方をを用いると組織標準プロセスを、どの様な開発スタイルにも適用する事が出来る。

#### (1) 静的プロセス(やるべき事)の定義

「静的プロセス」は、個々のプロセスの目的および成果(ゴール)を明確にし「やるべき事」を部品化する事にした。部品化により時系列(実施タイミング)的な概念は排除されている。部品化するプロセスには、特定のライフサイクルを意識していない CMMI® V1.3<sup>[01]</sup>の SG(固有ゴール)を採用した。つまり、SG相当の目的単位にプロセスが部品化される。

#### (2) 動的プロセス(ライフサイクル)の定義

「静的プロセス」を、作業の時系列に並べる事によりプロジェクトのライフサイクル、すなわち「動的プロセス」が定義される。この際、SG単位に定義された「静的プロセス」は、開発の時系列、つまり工程単位に定義される。

プロジェクトは、「選択指針」に従い、「動的プロセス」を選択する(図2:参照)。プロジェクトの開発スタイルが定義された「動的プロセス」に適合しない場合は、SEPG<sup>SM</sup>と共に「動的プロセス」を定義する事も可能とする。この仕組みを導入する事により、多種多様な開発に対応可能なライフサイクルを定義する事が出来る。プロセス部品と、ライフサイクルの関係例を図2に示す。

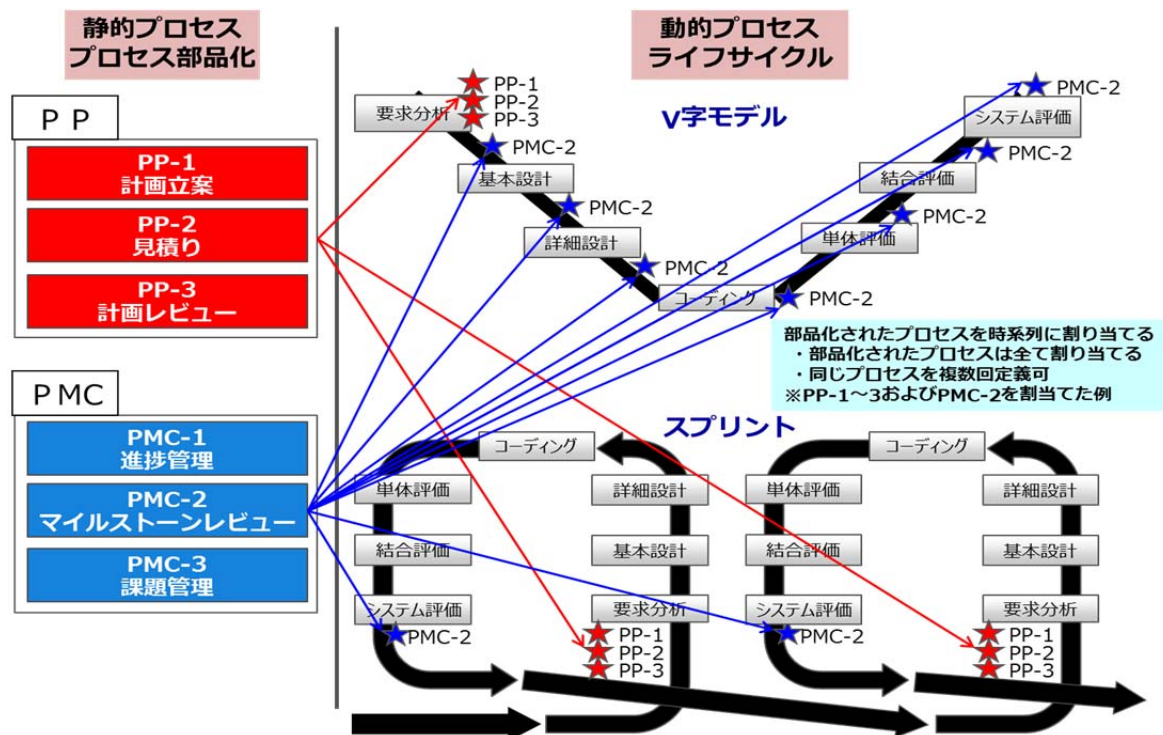


図2. プロセス部品とライフサイクルの関係例

### 5.2.2 テーラリングのパターン化によるテーラリング基準の構築

プロジェクトには、大規模、小規模、ウォーターフォール、アジャイルなど、多種多様な開発スタイルが存在する。このため、品質保証スタイルも画一的ではなく、プロジェクトの開発スタイルに合わせた形で「動的プロセス」として定義される。しかし、このままではプロジェクトへの負荷が必要以上に高くなるため、個々のプロジェクト特性(開発期間、人員規模など)に合わせ、適切に調整出来る様にテーラリングする必要がある。

テーラリングの程度(テーラリングレベル)、テーラリング対象の組み合わせが多い程、プロジェクト特性に合わせた品質保証が可能だが、適用するには高い力量が求められる。力量を必要としないテーラリングが実施出来る様に、テーラリングレベルと、テーラリング対象を最小限に絞り込み、プロジェクト特性に応じて選択する事とした。

「テーラリングで考慮すべきプロジェクト特性」、「テーラリングレベル」について検討した結果を以下に記載する。

#### (1) テーラリングで考慮すべきプロジェクト特性

研究員のこれまでの開発経験、開発現場の意見を元に抽出した「テーラリングで考慮すべきプロジェクト特性」を表2に示す。

一般的にテーラリングで考慮すべきプロジェクト特性として考えられる「Quality: 要求品質」、「Cost: 人員数(工数)」、「Delivery: 開発期間」に加え、「開発種別(新規・改造)」、「技術的難易度」を定義する事により、これまで以上にプロジェクト特性に適合したテーラリングが出来る仕組みとした。

表2. テーラリングで考慮すべきプロジェクト特性例

プロジェクト特性	定義	テーラリングパターン選択基準
要求品質	・開発物に要求される品質 ・開発物の適用システムに求められる品質 ・顧客要求品質 ・適用認定規定 ISO/IEC 15504、CMMI <sup>®</sup> 、Automotive SPICE <sup>®</sup> 等	高: 人命・財産・社会インフラ: 影響大
		低: 人命・財産・社会インフラ: 影響小
人員数(工数)	・開発を行う為、開発期間内に必要な人員の数 ※開発量(規模)、コスト(金額)と、同じ扱い	15名以上
		6名以上、15名未満
		5名以下
開発期間	・開発要件決定後から納品までの日数	1年以上
		4ヶ月以上～1年未満
		3ヶ月以下
開発種別 (新規・改造)	・開発形態: 新規開発/エンハンス開発	新規
		エンハンス
技術的難易度	・開発全体の技術難易度 ・開発物適用システム(業界) ・開発言語 ・レガシ開発or業界の新技術適用開発 ・類似開発物開発経験有無 ・開発システムの形態	高: 業界新技術適用、類似開発経験無し
		中: 業界新技術適用、類似開発経験有り
		低: レガシ開発、開発経験あり

(2) 簡単にテーラリングを行うためのテーラリングレベル設定

テーラリングレベル、テーラリング対象が、多種多様であればあるほどプロジェクト特性に合ったテーラリングが可能である。しかし、適用するにはテーラリングに対する力量が必要である。多くのプロジェクトでは力量が不足しており、有効なテーラリングが行えていない。テーラリングレベル、テーラリング対象を絞り込む事により、力量に頼らないプロジェクト特性に合った有効なテーラリングを行う事が可能となる。具体的には、テーラリングレベルを優・良・可の3つ、テーラリングレベルを特徴づける要素(テーラリング対象)として、以下の4つに絞込み、定義する事とした。

- **成果物**：成果物として何を作成するのか、いつ成果物を作成するのか
- **品質記録**：品質記録として何を残すのか、いつ品質記録を作成するのか
- **品質保証監査(工程移行判断)**：工程移行判断は、いつ実施するのか
- **測定と分析**：何を収集し、どの様な観点で分析するのか

テーラリングレベル、テーラリング対象例を表3に示す。

表3. テーラリングレベル、テーラリング対象例

テーラリングレベル	テーラリング対象	テーラリング内容
優(標準)	成果物	標準プロセス準拠
	品質記録	標準プロセス準拠
	品質保証監査(工程移行判断)	標準プロセス準拠
	測定と分析	標準プロセス準拠
良	成果物	標準プロセスで定義されている成果物相当を作成する。作成時期は、プロジェクトで定義する
	品質記録	標準プロセスで定義された品質記録相当をエビデンスとして残す(代替・まとめ可能)。
	品質保証監査(工程移行判断)	複数の工程をまとめて品質保証監査を実施する
	測定と分析	標準プロセスで定義された品質データを収集する。品質指標、分析内容は、プロジェクトで定義する
可	成果物	作成する成果物、作成時期を、プロジェクトで定義する
	品質記録	エビデンスとして残す品質記録をプロジェクトで定義する
	品質保証監査(工程移行判断)	最終工程完了時期にだけ品質保証監査を実施する
	測定と分析	収集する品質データ、品質指標、分析内容をプロジェクトで定義する

(3) テーラリングのパターン化

組織は、表2のプロジェクト特性の組み合わせ毎に、表3のテーラリングレベルを決定するためのマトリクスを作成する(表4参照)。

表4. テーラリングパターン選択マトリクス：例

テーラリングパターン	開発期間	人員数(工数)	要求品質	開発種別 (新規・改造)	技術難易度	テーラリングレベル		
Pattern: 1	1年以上	15名以上	高(人命・財産・社会インフラ・影響大)	新規	高	優(標準)		
Pattern: 2					中	優(標準)		
Pattern: 3					低	優(標準)		
Pattern: 4				エンハンス	高	優(標準)		
Pattern: 5					中	優(標準)		
Pattern: 6			低(人命・財産・社会インフラ・影響小)	新規	高	優(標準)		
Pattern: 7					中	良		
Pattern: 8					低	良		
Pattern: 9					エンハンス	高	優(標準)	
Pattern: 10						中	良	
Pattern: 11			6名以上、15名未満	高(人命・財産・社会インフラ・影響大)	中	優(標準)		
Pattern: 12					低	優(標準)		
Pattern: 13					中	可		
Pattern: 14			3ヶ月以下	5名以下	低(人命・財産・社会インフラ・影響小)	新規	低	可
Pattern: 15							高	優(標準)
Pattern: 105	エンハンス	低			中	可		
Pattern: 106					低	可		
Pattern: 107	高	中	高	優(標準)				
Pattern: 108			低	可				

(4) テーラリングの流れ

プロジェクトは「選択指針」に従ってライフサイクルを選択後、自身のプロジェクト特性を表2を参照して評価する。評価結果を元に、表4で定義されたテーラリングレベルを適用し、プロジェクトのプロセスを決定する。プロジェクトは、表3の内容に従ってテーラリングを実施する。これによって、テーラリング力量に影響されずにプロジェクト特性を考慮したテーラリングが実現出来る。

5.2.3 改善活動のサイクル

ライフサイクルの選択指針、およびテーラリング指針を、より実態にあったものに成熟させるためには、プロジェクトがテーラリングした結果であるプロセスの変更内容、変更理由、品質実績などのテーラリング実績を INPUT 情報として組織的改善活動を行う必要がある。<sup>[02]</sup>改善の活動のサイクルについて検討した結果を以下に記述する。

(1) テーラリング実績に基づく改善活動の流れ

組織標準プロセスの改善活動のサイクルは、「テーラリング指針」の改善と「動的プロセス・選択指針」の改善に大別される。テーラリング結果として定義されたプロジェクトのプロセスは、プロジェクトによって実行され、テーラリング実績が記録される。SEPG<sup>SM</sup>は、プロジェクトのテーラリング実績を収集・分析し、「テーラリング指針」の改善を行う。このサイクルを継続的に行う事で、「テーラリング指針」の精度を上げていく。また、新たなライフサイクルの標準化、選択指針の見直しなど、「動的プロセス」の改善に必要な情報も、蓄積されたテーラリング実績から分析する。この改善サイクルを継続して行う事で、プロジェクトの実態に適合した組織標準プロセスを醸成させていく。

「テーラリング指針」と「動的プロセス・選択指針」の改善サイクルを図3に示す。

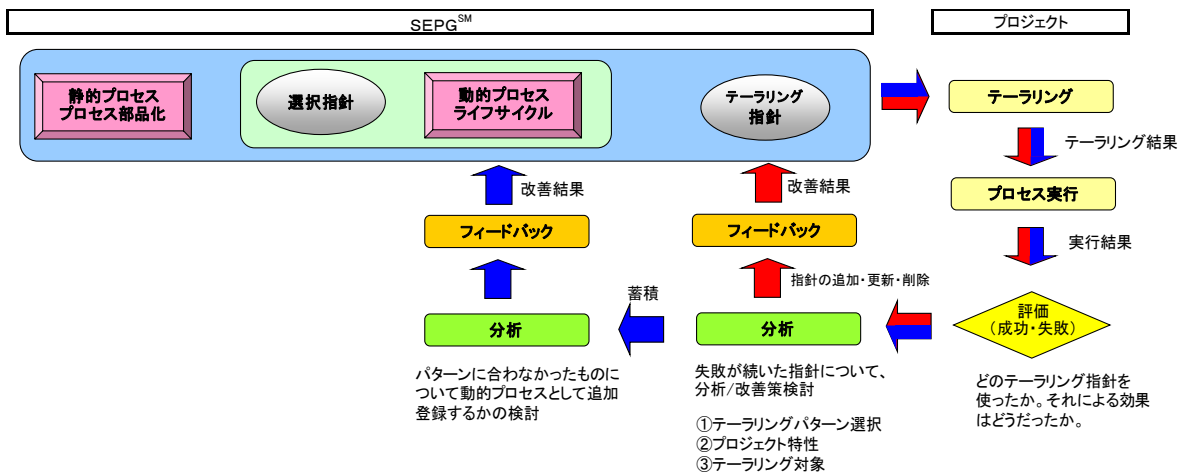


図3. 「テーラリング指針」と「動的プロセス・選択指針」の改善サイクル

### 5.3 期待される効果

本研究で検討した仕組みを実施する事によって、以下の効果が期待される。

- 昨今の急激なライフサイクルの多様化に対応する事が可能となる。
- 組織標準プロセスが適用出来ないプロジェクトを最小限に抑える事が可能となる。
- プロジェクトは、必要以上の負荷が排除された効率的で、かつ有用なプロセスを実施する事が可能となる。
- 記録されたテーラリング実績を元に組織標準プロセスが改善される事により、組織標準プロセスの醸成が進み、プロジェクトを成功に導く事が可能となる。

### 5.4 今後の課題

#### (1) 静的プロセスの構成要素の定義

「静的プロセス」は CMMI<sup>®</sup> v1.3<sup>[01]</sup>の SG を採用し、プロセス単位にその目的と成果物、及び INPUT/OUTPUT、タスクや手順を定義する事としているが、CMMI<sup>®</sup>[01]の定義そのものに近い文書になる可能性がある。定義内容の精査が今後の課題である。

#### (2) 動的プロセスの具体性の程度

部品化された「静的プロセス」を組み合わせて「動的プロセス」を定義するが、プロジェクトの実態を反映しすぎると、「動的プロセス」の数が増加し制御が困難になる一方、汎用的な表現ではテーラリング指針の定義が困難になる事が予想される。「動的プロセス」を定義する基準の整備が今後の課題である。

#### (3) テーラリングで考慮すべきプロジェクト特性の定義

プロジェクトの力量に頼らないテーラリングを行うため、テーラリングの対象範囲やレベルを絞ったテーラリング指針とした。ただし、定義したプロジェクト特性は我々研究員の経験から特定したものであり、実運用で PDCA サイクルを回し、改善していく必要がある。

#### (4) 現場での試行

今回は、仕組みの検討に想定以上の時間を要したため、現場での試行が出来なかった。しかし、現状を打破するための壮大な仕組みであり、今回定義した仮説の検証及び精緻化を行うためには、現場での試行は欠かせない。継続案件として、現場での試行を検討する。

## 6. 感想

- 研究の序盤においては、各社の持つ標準プロセスやテーラリングに関わる様々な用語の定義が各研究員間で異なっており、共通認識を確立する事に時間を要した。
- テーラリングに特化した文献が無く、自分たちでテーラリングモデルを開発しなければならず、検討に時間を要した。

## 7. 参考文献

[01] CMU/SEI-2010-TR-033、JASPIC CMMI V1.3 翻訳研究会訳

『開発のための CMMI 1.3 版』 2010 年

[02] ソフトウェアプロセス成熟度の改善、日科技連出版社、1991/09 発行

\* CMMI and Capability Maturity Model are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

(SM) SEPG is a service mark of Carnegie Mellon University.

Automotive SPICE<sup>®</sup> is a registered trademark of the Verband der Automobilindustrie e.V. (VDA).



8. 付録

テーラリングパターン選択マトリクス例を付録1に示す。

高品質、大規模、高い新規性等、厳格なプロセスの実装が必要なプロジェクト特性を多く含む場合は、基本的にテーラリングレベル：優を適用する。しかし、Pattern:106の様に、高品質が求められず、小規模、エンハンス開発であっても開発難易度が高い場合は、テーラリングレベル：優を適用した方が、成功するパターンも存在する。

テーラリングパターン選択マトリクスを作成する際は、以下の点を考慮する。

- ①全てのプロジェクト特性を考慮する
- ②品質に大きな影響を与えるプロジェクト特性に着目する

付録1. テーラリングパターン選択マトリクス例

テーラリングパターン	開発期間	人員数(工数)	要求品質	開発種別 (新規・改造)	技術難易度	テーラリングレベル
Pattern: 1	1年以上	15名以上	高(人命・財産・社会インフラ:影響大)	新規	高	優(標準)
Pattern: 2					中	優(標準)
Pattern: 3				低	優(標準)	
Pattern: 4				高	優(標準)	
Pattern: 5				中	優(標準)	
Pattern: 6				低	優(標準)	
Pattern: 7			高	優(標準)		
Pattern: 8			中	良		
Pattern: 9			低	良		
Pattern: 10			高	優(標準)		
Pattern: 11			中	良		
Pattern: 12			低	良		
Pattern: 13		高	優(標準)			
Pattern: 14		中	優(標準)			
Pattern: 15		低	優(標準)			
Pattern: 16		高	優(標準)			
Pattern: 17		中	良			
Pattern: 18		低	良			
Pattern: 19		高	優(標準)			
Pattern: 20		中	良			
Pattern: 21		低	良			
Pattern: 22		高	優(標準)			
Pattern: 23		中	可			
Pattern: 24		低	可			
Pattern: 25		高	優(標準)			
Pattern: 26		中	良			
Pattern: 27		低	良			
Pattern: 28		高	優(標準)			
Pattern: 29		中	良			
Pattern: 30		低	良			
Pattern: 31		高	優(標準)			
Pattern: 32		中	可			
Pattern: 33		低	可			
Pattern: 34		高	優(標準)			
Pattern: 35		中	可			
Pattern: 36		低	可			
Pattern: 37	高	優(標準)				
Pattern: 38	中	優(標準)				
Pattern: 39	低	優(標準)				
Pattern: 40	高	優(標準)				
Pattern: 41	中	優(標準)				
Pattern: 42	低	優(標準)				
Pattern: 43	高	優(標準)				
Pattern: 44	中	優(標準)				
Pattern: 45	低	優(標準)				
Pattern: 46	高	優(標準)				
Pattern: 47	中	良				
Pattern: 48	低	良				
Pattern: 49	高	優(標準)				
Pattern: 50	中	優(標準)				
Pattern: 51	低	優(標準)				
Pattern: 52	高	優(標準)				
Pattern: 53	中	優(標準)				
Pattern: 54	低	優(標準)				
Pattern: 55	高	優(標準)				
Pattern: 56	中	良				
Pattern: 57	低	良				
Pattern: 58	高	優(標準)				
Pattern: 59	中	可				
Pattern: 60	低	可				
Pattern: 61	高	優(標準)				
Pattern: 62	中	優(標準)				
Pattern: 63	低	優(標準)				
Pattern: 64	高	可				
Pattern: 65	中	可				
Pattern: 66	低	可				
Pattern: 67	高	優(標準)				
Pattern: 68	中	可				
Pattern: 69	低	可				
Pattern: 70	高	優(標準)				
Pattern: 71	中	優(標準)				
Pattern: 72	低	可				
Pattern: 73	高	優(標準)				
Pattern: 74	中	優(標準)				
Pattern: 75	低	優(標準)				
Pattern: 76	高	優(標準)				
Pattern: 77	中	良				
Pattern: 78	低	良				
Pattern: 79	高	優(標準)				
Pattern: 80	中	良				
Pattern: 81	低	良				
Pattern: 82	高	優(標準)				
Pattern: 83	中	可				
Pattern: 84	低	可				
Pattern: 85	高	優(標準)				
Pattern: 86	中	優(標準)				
Pattern: 87	低	良				
Pattern: 88	高	優(標準)				
Pattern: 89	中	良				
Pattern: 90	低	良				
Pattern: 91	高	優(標準)				
Pattern: 92	中	可				
Pattern: 93	低	可				
Pattern: 94	高	優(標準)				
Pattern: 95	中	優(標準)				
Pattern: 96	低	可				
Pattern: 97	高	優(標準)				
Pattern: 98	中	良				
Pattern: 99	低	可				
Pattern: 100	高	優(標準)				
Pattern: 101	中	可				
Pattern: 102	低	可				
Pattern: 103	高	優(標準)				
Pattern: 104	中	可				
Pattern: 105	低	可				
Pattern: 106	高	優(標準)				
Pattern: 107	中	可				
Pattern: 108	低	可				