

Rational Unified Process - より成熟したプロセスを実現する ための手法

**Annie Kuntzmann–Combelles, Q-Labs
France**

**Philippe Kruchten, Rational Software
Canada**

Rational Software、Q-Labs ホワイト・ペーパー

TP 178, 2/01

目次

要約	1
効率性は十分か	1
改善に向けたロードマップ	2
組織が SPI プログラムを開始するために必要な項目	2
CMM モデル (参考資料 [6])	2
IDEAL 手法 (参考資料 [4])	2
プロセス	2
RATIONAL UNIFIED PROCESS	3
ISO 文書	3
SPI に向けた RUP の使用	3
RUP はプロジェクト・レベルで CMM 要求と一致	4
ライフ・サイクルの問題	4
プロジェクト開発構想を取得し、ビジネス・モデルを開発	6
プロジェクトの組織化と計画	6
統一変更管理 (UCM: Unified Change Management) ポリシーの配置	7
プロセスの選択とカスタマイズ	7
リスク管理	9
測定	9
RUP は組織レベルで CMM と一致	10
プロセスの改善	10
組織においてプロセスとツールを実装するためのステップ	11
プロセス実装プロジェクトはフェーズに分割可能	12
要員とスキル	13
RUP 実践的考察: RUP の活用方法	14
結論	17
著者の連絡先情報	18
参考資料	19

要約

このホワイト・ペーパーは、成熟した組織、つまりレベル 3 の開発ユニットが例示しなければならない主要な概念と Rational Unified Process (RUP) コンポーネントがこうした要求にどのように適合するかを強調します。

高度なプロジェクト成熟度と組織成熟度の両方を取り上げます。さらに、セクション 4 では、RUP の使用を開始するために有効ないくつかのヒントを提供し、さまざまな環境で早期に採用した人々が得た主なメリットを報告します。

効率性は十分か

今日、企業は高度なパフォーマンスを達成するための方法を継続して探求しなければなりません。今後の課題は世界規模での競争です。以下のような徴候は、世界市場で競争するとき一層鮮明になってきます。

- **性能のニーズ:**非常に多くのバグが検証段階で見つかったが、納期目標を達成するために不安定な製品バージョンが市場に送り出される
- **効率性のニーズ:**プロジェクトは予算と計画を超過する
- **マーケット・シェアの損失:**競争相手はパフォーマンスでも品質でも勝る
- **有能な要員の欠如:**離職率は高く、新規技術者の雇用は非常に困難
- **最小のリスクで発展的なテクノロジーを統合するニーズ:**製品を支えるテクノロジーは流行を追う。チームは常に課題に応えるだけの経験を持っているとはかぎらない

より一般的に、開発される製品のすべての側面を最適化するという強いニーズがあります。

したがって、製品を開発するためのプロセスが結果に重大な影響を与えるという広く受け入れられている前提の下で、最も一般的なシナリオはソフトウェア・プロセスを検討して製品を改善することになるでしょう。

この種の状況に直面している組織はおそらく以下のようなソリューションを考慮するでしょう。

ソリューションの最初の部分は市場に対するソフトウェアの価値を理解することです。ソフトウェアが製品とプロジェクトにとって競争優位の鍵であり続ける場合、それらを規定して実装するプロセスは時間の経過と共に注意深く定義、文書化、最適化されなければなりません。経営層の上部はいまだにソフトウェアは問題を発生させ、納期が大幅に遅れると確信しているように見受けられます。つまり、ソフトウェアの付加価値分析は行われていないのです。

ソリューションの第 2 の部分は以下のようにプロセスの主な問題を修正するために十分な投資を行うことです。

- プロセスを評価して改善の機会を見つける
- 一からやり直さない:他者の経験から学び、突破口を見出す
- すべてのプロジェクトにおいて管理と技術の両面を考慮する
- 将来必要になるスキルを管理する。学習し、共有し、成長する

組織を生産性の高いチームに変えて行くためにいくつかのロードマップがあります。最良の決定は、可能なものを変更し、変更できないものを受け入れ、成功したソフトウェア・コミュニティによって検証済みのものを可能な限り借用することです。能力成熟度モデル (CMM) (参考資料 [6]) と Rational Unified Process (RUP) (参考資料 [11]) は、適切な製品開発プロセスに向けた進展を加速できる、広く普及した強固なツールの例です。ツールという用語を使用した理由は、これら 2 つのコンポーネントがソフトウェアの実践原則を変更し、改善するための効果的なサポートである一方、ユーザーの組織環境に対する最良の実践原則を定義できるものだからです。

次のセクションでは、これらの 2 つのコンポーネントである CMM と RUP、プロセスと提供製品を改善するために考慮しなければならないその他の主要なコンポーネントについても説明します。

改善に向けたロードマップ

過去 10 年間、ソフトウェア・コミュニティではプロセスに対する意識が高まり、ソフトウェア製品をその生産プロセスから独立して評価できないことがわかりました。その結果として、多数の組織が、ソフトウェア・プロセス改善 (SPI) 構想として知られるソフトウェア開発プロセス向上のためのプログラムを作成しました。ほとんどの場合、ソフトウェア開発は組織内の 1 つまたは複数のユニットまたは部署に集中し、独立した専門分野とみなされています。SPI プログラムを立ち上げると、こうした状況は変化し、ソフトウェア・チームは組織の他の分野と適切に統合されます。

組織が SPI プログラムを開始するために必要な項目

- プログラムを開始するための刺激とアクティビティの改善をコーディネートする主要な担当者。言い換えると、改善プロジェクトに関する開発構想と目的を明確にするため。
- プロジェクトと組織の実践原則を評価し、成功の阻害要因となる不足部分を識別するための参照モデル。ソフトウェア工学研究所 (SEI) で開発された CMM モデルは何千もの組織で使用されている実際の参照であり、SPICE フレームワーク (ISO 15504: 参考資料 [9]) は近々、CMM が準拠する公式標準になると思われる参照です。
- 改善プログラムを結果の成功に至るまで管理する手法。SEI で定義された IDEAL (参考資料 [4]) フレームワークは有力であることが実証されています。
- 各プロジェクトを成功させるために適用可能な開発プロセス。RUP は受け入れられる市販のソリューションです。

CMM モデル (参考資料 [6])

ソフトウェア工学研究所 (SEI) 能力成熟度モデル (CMM) は効果的なソフトウェア・プロセスの要素を説明するフレームワークです [6]。CMM は、未成熟な初期レベルから成熟し統制されたプロセスへの、発展的な改善経路を表します。CMM モデルは、ソフトウェア開発と保守機能を拡張するために示されている多くの主要なプロセス領域において、推奨される実践原則のセットを提供します。CMM は開発者に対し、開発と保守プロセスを管理し、卓越したソフトウェア開発と保守の文化に向けて進化する方法をガイドします。

IDEAL 手法 (参考資料 [4])

ソフトウェア・プロセス改善は、ソフトウェア作業の整理と実行方法を進化させるための体系的、協調的、かつ長期にわたる手法です。改善手法として IDEAL 手法というものがあります。これは SEI により定義された SPI の統合アプローチです。

IDEAL は、初期 (Initiating)、診断 (Diagnosing)、構築 (Establishing)、実施 (Acting)、活用 (Leveraging) という 5 つのフェーズを定めます。これらの各フェーズでは以下のように特定のアクティビティに専念します。

- 実現または支援されるビジネスの目標と目的を規定する (**初期**)
- 関連する標準または参照モデルに照らして組織の現在の状態を見極める (**診断**)
- 選択した方法を実装するための計画を立てる (**構築**)
- 利用可能なすべてのものを集めて組織のニーズに合った「最善の推測」ソリューションを作成する。例えば、既存のツール、プロセス、知識、スキルを活用してソリューションを実行する (**実施**)
- IDEAL を実装するために用いられたプロセスについて、学習した教訓をまとめる (**活用**)

プロセス

プロセスをまったく持たない組織もありますし、経験に基づくプロセスを持っている組織もあります。また、OPEN や Rational Unified Process (RUP) のように、市販のプロセスもあります。OMG グループは近々、一般的なプロセス・モデルを提案するかもしれません。「プロセス」という用語はソフトウェア開発コミュニティで広く使用されていますが、ソフトウェア・プロセスのコンポーネントに関する仮定は非常に異質です。SEI の定義では、ソフトウェアや関連製品を開発、保守するために使用する一連のアクティビティ、方法、実践原則、変換となっています。プロセスは、プロジェクトが定義された目的に一致することを確認する主要な要素です。

この定義にはアクティビティ、テクノロジー（つまり方法とツール）、人が含まれることに注意してください。これら 3 つは最も重要なコンポーネントです。

Rational Unified Process

RUP は長期のプロジェクト経験から恩恵を得る既存のプロセス・フレームワークの一例です（参考資料 [10]、[11]）。

RUP は、システムのアーキテクチャーを定義する初期バージョンを素早く開発することにより、高リスクの領域にごく初期の段階から対応することを重視します。RUP はプロジェクトの開始時に要求が確定していることを前提としませんが、プロジェクトの進行と共に要求を洗練することができます。RUP は変更を予測し、適応します。このプロセスは文書も「儀式」も重視せず、ソフトウェア開発に関連する退屈な作業の多くを自動化する役目を果たします。主な焦点はソフトウェア製品そのものとその品質に当てられ、エンド・ユーザーの満足度と投資収益率目標の達成度が測定されます。

RUP は一般的なため、サイズと適用分野の両方において、多様なソフトウェア製品やプロジェクト向けにカスタマイズ可能で、人、プロセス、ツール（または方法）という 3 つの領域を専門に扱います（参考資料 [10]、[11]）。

ISO 文書

最後に、この全体像を完成させるため、ISO 9001、ISO 12207、ISO 15504 (SPICE: 参考資料 [9]) といった ISO 文書について述べます。これらは、組織が自らのソフトウェア開発ノウハウを他の組織と比較可能な参照資料です。ISO ドキュメントは、前に述べた他のコンポーネントが準拠する一般的なフレームワークを形成します。

ISO 15504 (SPICE ともうい) はソフトウェア・プロセスを分析するためのもう 1 つの参照モデルです。完全に実現されている多数の評価モデル (CMM はその中の 1 つ) や多数の評価方法 (SEI により定義される方法はその中の 1 つ) が ISO 15504 の規範的な部分にマップできることを仮定しています。

このホワイト・ペーパーの残りでは、ソフトウェア・コミュニティの事実上標準である CMM に焦点を当てます。CMM は、何百というソフトウェア・プロセス改善構想をサポートする完全に実現されたモデルです。説明する内容はすべて SPICE に準拠します。

SPI に向けた RUP の使用

前のセクションでは、CMM 参照モデル、改善プログラムの完了に至るまでの管理方法 (IDEAL)、競争力を高め、ソフトウェアの世界規模の基準を満たすために適用可能な開発プロセスについて説明しました。これらの主要な要素を考慮し、RUP の概念を CMM 要求に対応付け、RUP の潜在能力を反映させて、特定のソフトウェア性能を実現します。組織にとって、ソフトウェア性能とは以下のいずれかを意味します。

- 適時に成功するプロジェクトの適切な判断
- ビジネスにおける成功と生き残り
- リスクの取得とプロジェクト出力の管理
- 製品の適切な品質の確保
- 製品における先進的なソフトウェア・テクノロジーの統合

ソフトウェア性能は以上の説明を高度に組み合わせたものですが、ソフトウェア性能の一部として考慮しなければならない以下のような概念があります。

- プロジェクト・レベルの考慮点
ライフ・サイクルの問題: プロジェクトのライフ・サイクルは製品の実装と保守に限定されません。プロジェクトのビジネス、財務、戦略に対処する管理的な側面が存在します。

プロセスの選択とカスタマイズ: 市販製品および/または過去の経験を反映する標準プロセスは多様なプロジェクトの目的に合致しません。効率性は、特定のプロジェクト向けに適切な開発プロセスを定義することにより生まれます。

リスク管理:どんなプロジェクトもリスクに直面します。有能な組織はリスクを予測し、プロジェクトの再構成を決定して、リスク発生の影響を最小限にとどめます。

測定:ソフトウェア組織における長期にわたる進展の鍵は履歴データを収集し、ソフトウェアの品質と生産性を分析することです。それぞれのプロジェクトについて、一定の履歴データを収集することは時間を要する作業です。作業、スケジュール、プログラム・サイズ、実装される機能、欠陥の数に関するデータは、将来のプロジェクトを計画し、予測を向上させるうえで確固とした基礎になります。パフォーマンスの予測は成熟の兆候の1つです。

- **組織レベルの考慮点 (プロジェクトの集合として)**
プロセスの改善:性能は、組織が過去、特に他者の過ちから学習し、こうした教訓をプロセスの発展に変換できることを示します。改善反復サイクルは改善を定量化するために計測が定義されている場合にのみ完了します。

要員とスキル:組織の能力はプロセスを適用し、製品を開発する人々の能力に密接に結び付いています。

RUP はプロジェクト・レベルで CMM 要求と一致

CMM は特定の成熟レベルに達した組織におけるプロジェクトの状態を説明します。

レベル 2 組織のプロジェクトには、基本的なソフトウェア管理が導入されています。現実的なプロジェクトの遂行事項は、以前のプロジェクトで観察された結果と、現在のプロジェクトの要求に基づきます。ソフトウェア・プロジェクトの管理者はソフトウェアのコスト、スケジュール、機能を追跡します。遂行事項を満たす作業の中で、問題は発生した時点で識別されます。ソフトウェアの要求とそれを満たすために開発される製品にはベースラインがあり、その完全性が管理されます。ソフトウェア・プロジェクト標準が定義され、組織はそれが忠実に守られていることを確認します。ソフトウェア・プロジェクトは場合によって請負業者と協業し、顧客側と供給側との間に強力な関係を築きます。

定義されたレベル (レベル 3) で、ソフトウェアの開発と保守に関する組織横断的な標準プロセスが文書化されます。プロジェクトでは組織の標準ソフトウェア・プロセスをカスタマイズし、プロジェクトに固有の特性に合わせて、独自の定義されたソフトウェア・プロセスを開発します。このカスタマイズされたプロセスを、CMM では、プロジェクトの定義されたソフトウェア・プロセスと呼びます。定義されたソフトウェア・プロセスには、十分に定義されたソフトウェアの開発と管理プロセスが、矛盾のない統合されたセットとして含まれます。十分に定義されたプロセスは、作業実行のための準備基準、入力、標準、手順、ピア・レビューのような検証メカニズム、出力、完了基準などで特徴付けられます。経営陣によるすべてのプロジェクトでの技術的進行の理解は、ソフトウェア・プロセスが十分に定義されているためです。

ライフ・サイクルの問題

CMM で、ライフ・サイクルの問題は複数の主要プロセス領域 (KPA: Key Process Area) で扱われます。CMM の概念を組織に初めて提示すると、一般に、「ソフトウェア・プロセス」という用語が「ソフトウェア・ライフ・サイクル」と間違えて解釈されます。プロジェクト・ライフ・サイクルには、開発の問題と同じくらいビジネスと財務の問題が含まれます。レベル 2 の主要プロセス領域とレベル 3 の主要プロセス領域の一部はプロジェクト・ライフ・サイクルに関連します。内容を以下に示します。

- **要求管理目標 1:**ソフトウェア開発と管理で使用するベースラインを確立するために、ソフトウェアに割り当てられたシステム要求が管理される。
- **ソフトウェア・プロジェクト計画目標 1:**ソフトウェア・プロジェクトを計画し追跡するために、ソフトウェア見積もりが文書化される。
- **ソフトウェア・プロジェクト計画目標 2:**ソフトウェア・プロジェクトのアクティビティと遂行事項が、計画され文書化される。
- **ソフトウェア・プロジェクト追跡目標 1:**ソフトウェア計画に対して、実際の結果と作業の進捗が追跡される。

- ソフトウェア・プロジェクト追跡目標 2:ソフトウェアに関する遂行事項の変更が、影響を受けるグループと個人によって同意される。
- ソフトウェア構成管理目標 1:選択されたソフトウェア製品が識別、管理されて、利用可能である。
- ソフトウェア構成管理目標 2:識別されたソフトウェア製品に対する変更が管理される。
- ソフトウェア構成管理目標 3:影響を受けるグループと個人に、ソフトウェア・ベースラインのステータスと内容が通知される。
- ソフトウェア製品開発目標 1:ソフトウェアを作成するために、ソフトウェア開発タスクが定義、統合され、一貫して実行される。
- ソフトウェア製品開発目標 2:ソフトウェア製品が、相互に矛盾しない。

言い換えると、CMM は以下のことを要求します。

1. 開発されるアプリケーションのビジネス環境を考慮し、プロジェクト要求の優先度を設定し、プロジェクトの機能と機能外の内容に関する必要なすべての決定を行うために、ベースラインが暗黙的に定義される。
2. プロジェクト計画が作成され、現実的で信頼できるアクティビティ、ロールと責務、マイルストーン、成果物の一覧が設定される。プロジェクト計画はリスクを考慮する。
3. プロジェクト計画が監視され、機能内容やプロジェクト環境/組織に大きな逸脱が発生する場合にプロジェクト・チームが対応する。
4. プロジェクト内のすべての変更が識別され、分析される。ベースラインとプロジェクト計画の中で、さらに決定がなされ、報告される。
5. プロジェクトの最高のパフォーマンスを確保するために方法とツールが選択される。

Rational Unified Process は、これらの要求を以下のように方向づけフェーズの主要な作業として実現します。

- 前述のリスト中の項目 1 を一致させることにより、プロジェクト開発構想を取得し、ビジネス・モデルを開発する。
- プロジェクトを組織し、計画する。
- 潜在的なリスクを見積もる。(この項目は前述のリスト中の項目 2 に関連する。)
- 前述のリスト中の項目 3 と 4 と同様の統一変更管理ポリシーを展開する。

方向づけフェーズで優先する目的は、プロジェクトのライフ・サイクル目標に関して、利害関係者全員の同意を得ることです。方向づけフェーズは、プロジェクトを進める前にビジネスや要求のリスクを検討しなければならない新規の開発作業にとって、特に重要です。既存システムを拡張するプロジェクトでは、方向づけフェーズは短いフェーズになります。ただし、新規のシステム開発と同様に、プロジェクトを実施する価値があること、また、実現可能であることを確認することが重要です。

方向づけフェーズの終わりで、プロジェクトのライフ・サイクル目標が考察され、プロジェクトを進行するか取りやめるかの決定がなされます。

レビューを行う必須の成果物は以下のとおりです。

- プロジェクト開発構想書
- 開発企画書
- リスク・リスト (リスク管理のセクションを参照)
- ソフトウェア開発計画書
- 反復計画書
- 開発個別定義書 (プロセスの選択とカスタマイズのセクションを参照)

これらの成果物は、このセクションの冒頭で列挙した CMM 主要プロセス領域のいくつかの目標を満たします。

プロジェクト開発構想を取得し、ビジネス・モデルを開発

ビジネス・モデルは、ビジネス・ワーカーの内部的な視点からビジネス・ユースケースを定義します。このモデルは、ビジネスの現場で働く人々や、彼らが扱ったり使用したりするもの（ビジネスのクラスやオブジェクト）が相互に関係するしきみを静的、動的の両面で定義し、期待される結果を生成します。このモデルは、ビジネスの領域において果たされるロールと、その有効な責務にも重点を置きます。モデルのクラスのオブジェクトは、全体としてすべてのビジネス・ユースケースを実行する能力が備わっていなければなりません。

Rational Unified Process は、ビジネス・モデルに基づき、開発構想書の作成というタスクを識別します。このタスクの目的は以下のとおりです。

- どの問題を解決すべきかについて、同意を得ること。
- システムの利害関係者を識別すること。
- システムの境界を定義すること。
- システムの主要な基本要件を記述すること。

プロジェクトの組織化と計画

ソフトウェア・プロセスがプロジェクトの特性の影響を受けるのと同様に、プロジェクトの組織もその影響を受けます。ここに提示するデフォルトの構造（図 1 を参照）は、以下のような要因の影響を反映するように調整します。

- ビジネス状況
- ソフトウェア開発作業の規模
- 斬新さの度合い
- アプリケーションのタイプ
- 現在の開発プロセス
- 組織的な要因
- 技術面と管理面の複雑性

これらの要因は、RUP においてプロセスの識別として考慮され、プロジェクト構造の選択に影響します。

プロジェクト構造は主に以下の 2 つから定義されます。

1. 各反復の長さ
2. 反復の数

反復は、いわば完全なミニプロジェクトであり、すべての主要なワークフローを経て、多くの場合、完全ではないが実行可能なシステムが生み出されます。そのシステムは 1 つのリリースと言えます。

反復の数を定めるには、リスク、サイズ、複雑さによってさまざまなケースが考えられます。

まったく新しい領域に製品を提供しようとする場合、方向付けフェーズにいくつかの反復を追加して、概念を統合したり、部門をまたがる顧客やエンド・ユーザーに種々の実物見本を提示したり、提案の要求に対する確固とした応答を示したりする必要があります。

製品が大規模かつ複雑で、長期間にわたって開発される場合、作成フェーズに 3 つ以上の反復を計画する必要があります。

プロジェクトが続く間、プロジェクト計画において把握された作業分解構造をサポートするために、組織は発展していきます。これを図 1 に示します（参考資料 [7]）。

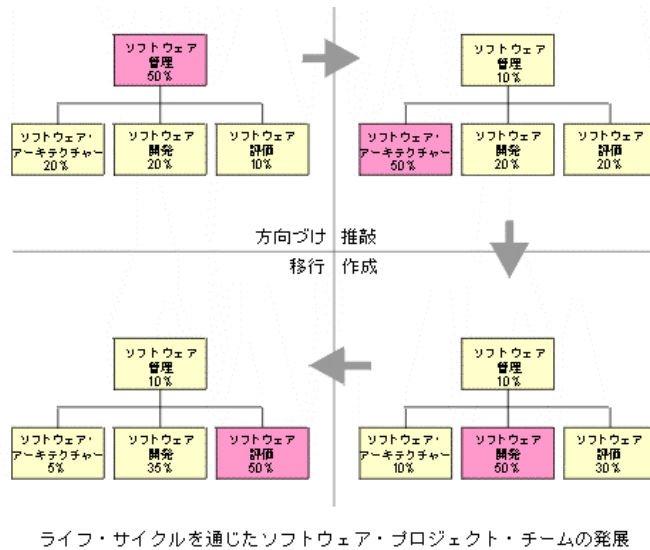


図 1:ソフトウェア・プロジェクト・チームの発展

この発展は、以下のように各フェーズにおける一連のアクティビティーの違いを強調しています。

- 方向付けチーム:計画作りに焦点を絞った組織であり、計画にすべての関係方面の同意を得られるようにするために、ほかのチームから多くの協力を得ます。
- 推敲チーム:アーキテクチャーに焦点を絞ったチームであり、その中のソフトウェア・アーキテクチャー・チームがプロジェクト推進の原動力となり、安定的なアーキテクチャーのベースラインを構築するために、ソフトウェア開発チームとソフトウェア評価チームの協力を得ます。
- 作成チーム:大部分のアクティビティーをソフトウェア開発チームとソフトウェア評価チームで行う、バランスの取れた組織です。
- 移行チーム:顧客指向の組織であり、使用状況のフィードバックが配置アクティビティーの原動力となっています。

この発展の過程で、メンバーをチーム間で移動させることが、プロジェクトに知識と能力を保持し続ける助けとなります。

統一変更管理 (UCM: Unified Change Management) ポリシーの配置

統一変更管理は、ソフトウェアシステム開発での変更をその要求からリリースまで管理する Rational の手法です。UCM は開発のライフ・サイクルにわたり、要求の変更、設計モデル、文書化、コンポーネント、テスト・ケース、ソース・コードを管理する方法を定義します。

UCM モデルの大きな特徴の 1 つは、プロジェクトの進行と変更中の成果物の計画と追跡に同じアクティビティーを使用することにあります。UCM モデルは、プロセスとツールの両方により構成されます。Rational ClearCase® と Rational ClearQuest® は UCM の基礎テクノロジーです。ClearCase は、システム成果物とプロジェクト管理成果物の両方を含む、ソフトウェア・プロジェクトにより生成されるすべての成果物を管理します。ClearQuest は、プロジェクトのタスク、障害、拡張要求 (総称的にアクティビティーと呼ばれる) を管理し、プロジェクトの進行を追跡するために必要なグラフ作成とレポート作成ツールを提供します。

プロセスの選択とカスタマイズ

成熟レベル 3 は、組織全体を対象にする 3 つの KPA とプロジェクトの組織、管理、開発を対象とする 4 つの KPA を含みます。管理と開発の両方のアクティビティーに関するソフトウェア・プロセスは文書化して標準化され、組織規模のソフトウェア・プロセスに統合されます。すべてのプロジェクトは、ソフトウェアの開発と保守のために、文書化されて承認済みの

組織のプロセス・バージョンを使用します。組織プロセス重視 (OPF)、組織プロセス定義 (OPD)、トレーニング・プログラム (TP) の各 KPA により、組織はレベル 2 におけるプロジェクトの最善の実践原則を識別し、それらを組織の標準として文書化することができます。これらの KPA は、プロジェクト、過去のプロジェクトで学んだ教訓、将来に向けた組織の使命と開発構想により識別されるニーズに基づくスキル管理にも焦点を当てます。しかし、特定のプロジェクトが、成功の実現、リスクの管理、パフォーマンスの向上を行うために、標準プロセスを採用するガイドラインがなければ、成熟はレベル 3 要求を満たしません。

したがって、ほとんどの主要評価担当者と CMM コミュニティにとって、レベル 3 は統合ソフトウェア管理 (ISM) KPA によって特徴付けられます。この KPA はソフトウェア開発と管理作業を一貫した定義済みソフトウェア・プロセスに統合するという目的を持っています。この定義済みプロセスは、顧客のニーズと制約、市場の要求、ビジネス戦略を満足させるのに役立つ標準プロセスの最適なカスタマイズです。

ISM は CMM モデルにおける 2 つの目標により定義されます。最初の目標はカスタマイズを含み、2 番目の目標はプロジェクト管理を必要とします。ISM におけるアクティビティ 10:「プロジェクトのソフトウェア・リスクは文書化された手順によって識別、評価、記録、管理される」はレベル 3 組織にとって第 2 の主要指標です。リスク管理はレベル 2 で要求されますが、レベル 3 組織はリスクの予測に成功しており、企業家的決定システムを例示することができます。後に示すリスク管理セクションでこうした側面を説明します。

RUP では、開発個別定義書の主要概念と環境の作業分野がこれらの要求を実現します。開発個別定義書は Rational Unified Process 製品の静的な構成です。つまり、特定のプロジェクト、製品、組織向けにカスタマイズされたソフトウェア開発のビジネス・プロセスです。開発個別定義書は何をどのように行うかということに焦点を当てます。また、従うべきプロセスの概要も提供するため、プロジェクトのメンバー全員が理解可能です。

Rational Unified Process は、任意の開発個別定義書で変更、カスタマイズ、追加、削除される可能性があるプロセスの構成を列挙します。

- **作業分野**
ソフトウェア・プロジェクトで、分析/設計、実装などの主要な作業分野のいずれかを完全に省くことはめったにありません。
- **成果物**
プロジェクトは、生成、更新、納品する成果物によってより大きく異なります。
- **タスク**
タスクは、少なくとも次のような 2 つの理由で変更される場合があります。まず、成果物を入力として使用し、成果物を出力として生成または更新するタスクはこれらの成果物の変更による影響を受けます。特に、ある成果物、または成果物中のある情報要素が不要になった場合、対応するステップが削除または大きく変更される場合があります。また、特定のアプリケーション・ドメインや開発の専門分野 (設計のステップ、プログラミング言語、自動化コード生成ツール、測定テクニックなど) に関係する、特定の技術、手法、ツールを導入するために、タスクが変更される場合もあります。

プロセス・エンジニアはプロセスの構成に責任を負い、開発プロセスの概観を決定し、開発組織 (チーム、プロジェクト、または会社) に開発個別定義書を「導入」して、開発者に使用方法を教えます。

プロセス・エンジニアが開発個別定義書を設定したら、プロジェクト管理者がそれをインスタンス化し、所定のプロジェクトについて実行します。これをプロセスの制定と呼ぶことがあります。

プロジェクトが進行するに従って、プロセスそのものの最中に学習した内容を、プロセス・エンジニアがプロセスを改良するためのフィードバックとして使用します。

リスク管理

リスク管理は CMM で広くカバーされており、主にレベル 2 ですが、レベル 3 まで管理することもできます。主要プロセス領域ソフトウェア・プロジェクト計画、ソフトウェア・プロジェクトの追跡と監視、統合ソフトウェア管理という目標を達成するために具体的なリスク・アクティビティが列挙されています。CMM 要求は多くの組織で観察される実践原則に従います。レベル 2 で、プロジェクトは一般に定期的にリスクを識別して評価します。しかし、事前対応策はあまりありません。レベル 2 において、リスク管理は十分にカバーまたは理解されておらず、プロジェクトのフォローアップと初期のリスク判別に問題があります。

これに対して、レベル 3 の組織では、リスクが識別、評価、軽減され、チームはかなり適切な予測を行います。最も先進的なチームの場合、定量的測度を使用して決定を行います。

CMM ベースの評価に関する長年の経験とソフトウェア・プロセスの改善は、リスクの管理においてソフトウェア・ユニットが直面する困難を浮き彫りにします。プロジェクトの管理と測定にあらゆる労力が注ぎ込まれているにもかかわらず、リスクの管理はあいかわらず弱点の 1 つとなっています。

RUP と反復的手法を使用することはプロジェクト・リスクの管理を成功させるために役立ちます。Rational Unified Process 全体の原動力がリスク管理です。

リスクは反復の基になります。リスクを限度内にとどめるかまたは軽減するために、種々のリスクに対する計画を反復的に立てます。リスク軽減戦略の有効性を評価するために、リスク・リストを定期的にレビューします。その結果、プロジェクト計画と以降の反復計画に改訂が及びます。

リスクを管理する鍵は、リスクが顕在化する（問題や障害が発生する）以前に、対処策を決定することです。例えば、大陸を横断する飛行機の進行方向をほんの数度変えただけで、飛行機が到着する場所は大きく違ってきます。それと同様に、早い段階からリスクを管理すればするほど、事後になって対処するよりもコストが安く苦痛が少なくなります。

以下にリスクの 4 つのカテゴリーを示します。

1. リソース面でのリスク
 - 組織
 - 資金
 - 人
 - 時間
2. ビジネス面でのリスク
3. 技術面のリスク
 - 見通しに関するリスク
 - 技術的なリスク
 - 外部依存リスク
4. スケジュール面でのリスク

リスク管理計画書とリスク・リストは RUP のプロジェクト管理作業分野で識別される 2 つの成果物です。これらはリスクの識別と評価を推奨します。

測定

測定は CMM の一般的な機能の 1 つです。一般的な機能は、レベル 2 からレベル 5 までの各主要プロセス領域について存在し、実践原則が制度化される時期を示します。メトリックスは非常に重要なので SW-CMM の新バージョン (CMMI フレームワーク、統合 CMM システム・エンジニアリング、ソフトウェア開発に組み込まれている) では、各主要プロセス領域に対する測定ベースの目標があります。

CMM で鍵となるのはプロセスを測定して、レベル 2 における妥当性とレベル 3 における有効性を判断することです。レポート作成からメトリックスに基づく分析と実行への移行は一般に困難を伴いますが、プロジェクトの成熟度が増したことの現れです。小規模にかつ早い段階で、プロジェクトの定量的管理に成功すると、測定のメリットを受け、理解することができます。

こうした一般的な要求に加えて、ソフトウェア・プロジェクト計画、ソフトウェア・プロジェクトの追跡と監視、統合ソフトウェア管理などの KPA は固有の測定要求を持ちます。これらはプロジェクト見積もりとプロジェクト管理を可能にするプロジェクト・データに関連しています。

RUP はメトリックスを使用するためのガイダンスを提供します。測定計画書と測定成果物は方向づけフェーズで生成しなければなりません。

プロジェクト測定の成果物はプロジェクトのメトリックス・データを格納します。ここには、常に最新の利用可能な測定結果が保存されています。また、根本的なデータから算出された派生メトリックスも含み、プロシージャやアルゴリズムなど派生メトリックスの算出方法に関する情報も格納されています。目標（機能、品質など）に対する進捗、支出、その他のリソース消費など、プロジェクト・ステータスのレポートはプロジェクト測定を使用して作成されます。プロジェクト・ステータスをさらに頻繁に、または継続的に表示するには、ソフトウェアのデータを自動的に収集するエージェントを使用してリアルタイムにプロジェクト・ステータスを表示する、Rational Project Dashboard 手法などを使用します。

Rational 文書は、最初は単純な初期のメトリックス・セットを編集します。プロジェクトの特定の側面のメトリックスには以下の項目があります。

- 規模と複雑さの進捗
- 要求や実装、規模、複雑さの変更率に関する安定性
- 変更範囲のモジュール化の度合い
- エラーの数とタイプから見た品質
- エラー発生率の成熟度
- プロジェクトの費用対計画の費用から見たリソース

RUP は組織レベルで CMM と一致

レベル 2 組織のソフトウェア・プロセス能力は規則正しくまとめることができます。なぜなら、ソフトウェア・プロジェクトの計画と追跡は安定していて、早期の成功を繰り返すことができるからです。プロジェクトのプロセスはプロジェクト管理システムの有効な管理下にあり、前のプロジェクトのパフォーマンスに基づく現実的な計画に従います。

定義されたレベル（レベル 3）では、組織規模のソフトウェア開発と保守に関する標準プロセスが文書化されます。この中にはソフトウェア開発と管理の両方のプロセスが含まれます。さらに、これらのプロセスは一貫した標準に統合されます。この標準プロセスは CMM 全体を通じ、組織の標準ソフトウェア・プロセス (OSSP) と呼ばれます。レベル 3 で確立されたプロセスは適宜使用され、変更され、ソフトウェア管理者や技術スタッフがより効果的に活動することを助けます。組織は、ソフトウェア・プロセスを標準化するときに、効果的なソフトウェア開発の実践原則を利用できます。組織のソフトウェア・プロセス・アクティビティに責任を持つグループがあり、例えば、SEPG（ソフトウェア開発プロセス・グループ）などがそれにあたります。組織規模のトレーニング・プログラムを実践して、スタッフと管理者が、割り当てられたロールを遂行するために必要な知識や技術を習得できるようにします。

ソフトウェアの開発と管理の両方のアクティビティが安定し、繰り返し可能な状態にあります。確立された製品ラインの中で、コスト、スケジュール、機能は管理され、ソフトウェア品質が追跡されます。このプロセス能力は、定義されたソフトウェア・プロセスのアクティビティ、ルール、責務に対する、共通の理解または組織全体の理解に基づきます。

プロセスの改善

能力は、組織が過去、特に過ちから学び、他者から学び、教訓をプロセスの発展に変換できることを暗示します。改善反復サイクルは改善を定量化するために計測が定義されている場合にのみ完了します。

組織の成熟度が向上すると、標準プロセスが変化します。レベル 2 からレベル 3 への移行は、プロジェクト内のすべての適切な実践原則が制度化され、プロジェクト全体で最良の実践原則を識別するための評価プロセスが設定されて、組織の標準ソフトウェア・プロセス (OSSP) に文書化されることを暗示します。これらは組織プロセス重視 (OPF) 主要プロセス領域の要求です。その後、OSSP は、組織全体のプロジェクトで得られた教訓を基にして洗練されます。プロジェクトは同じ標準を使用するので、経験と学習した教訓は容易に蓄積され、OSSP そのものはさまざまな経験からメリットを受けることができます。

RUP 環境の作業分野は同様の手法を開発します。「組織においてプロセスを実装する」という概念は、開発組織でプロセスとツールを実装するために組織レベルで行うことを説明します。

ソフトウェア開発組織における新しいプロセスの実装は、RUP の 4 つのフェーズ、方向づけ、推敲、作成、移行を使用して記述できます。

組織においてプロセスとツールを実装するためのステップ

CMM の OSSP は組織内の新しいプロセスであり、その定義は RUP の 4 つのフェーズに従うことができます。

それぞれのソフトウェア開発プロジェクトが必要なカスタマイズと共に使用可能な組織規模の開発環境を構築するかどうかの決定は重要ですが、一定の成熟レベルが存在しなければなりません。

組織全体の環境を作成する場合、組織の開発環境を作成するプロジェクトを開始する必要があります。また、そうしたプロジェクトを開始することを決めた場合、このプロジェクト・チームはソフトウェア開発プロジェクト・チームと非常に密接に作業することを明確にしておかなければなりません。RUP はこれを特別なプロジェクトとみなすことも推奨します。これによって CMM 要求も実現されます。

プロセス実装プロジェクトはいくつものフェーズに分かれます。プロジェクトの準備が完了し、プロセスとツールが配置され、組織全体での使用が成功するまで 4 つのステップすべてがフェーズごとに実行されます。

プロセス実装プロジェクトはフェーズに分割可能

4つのフェーズの内容は以下のとおりです。

- フェーズ 1: プロセス実装プロジェクトをスポンサーに売り込む。
- フェーズ 2: 主要なリスクを処理する。
- フェーズ 3: すべてを完了する—テンプレート、ガイドライン、開発個別定義書の例が準備され、トレーニング・カリキュラムが設定される。
- フェーズ 4: 組織全体に配置する。

これらのフェーズは、Rational Unified Process を使用したソフトウェア開発プロジェクトと同様に、それぞれ、方向づけ、推敲、作成、移行と呼ぶ場合があります。

特定のプロジェクトで得られた教訓や特定のテクノロジーによってパフォーマンスを向上させるために標準プロセスが発展する都度、同じプロジェクト・フェーズが定義されます。

RUP は、ソフトウェア・プロセス改善の全体的な背景である組織的な変更の管理という概念も定義します。プロセスの変更を適切に実装するための推奨事項は以下のとおりです。

- 組織のさまざまなレベルで、変更エージェントを指定する。
- 変更を、小さく、合理的で、数値化できるステップとして計画する。
- 組織のレベルに合わせた普通の言葉を使用して、変更を伝達する。

これらの推奨事項は着想の元となっている IDEAL 手法による推奨と同様です。

最後に、RUP は「メンター」という特定のアクターを定義します。メンターとは、プロジェクト・チームに必要なこととそれがいつ必要になるかを教え、導く要員です。メンタリングの典型的な方法は次のとおりです。

- ワークショップ指導者
アクティビティーの中で、グループで行うことが最適なものがあります。例えば、ユースケース・モデリング時にアクターやユースケースを検索することがあります。このようなアクティビティーを行うときに、プロセス専門家であるモデリング指導者が存在すると有益です。
- プロセス専門家
プロセス専門家とは、プロジェクトに対してオンサイト・サポートを行う要員です。プロセス専門家のタスクは、開発者がプロセスやモデルをできる限り使用できるようにすることです。
- プロジェクト管理者サポート
プロセス専門家は、プロジェクト管理者がプロジェクトを計画し進行するための支援を行うことができます。プロジェクト管理者に該当プロセスの経験が浅い、またはない場合があります。
- レビュー担当者
知識を伝達する費用効果の高い方法は、プロセス専門家が各フェーズの結果をレビューすることです。プロセス・レビュー担当者は、プロジェクトで実行されたプロセス適合のレビュー時にも価値をもたらします。

このメンターは CMM の SEPG (ソフトウェア開発プロセス・グループ) とは異なるロールを果たすのでしょうか。いいえ、同じタイプのロールです。

要員とスキル

組織レベルで、CMM は「トレーニング・プログラム」主要プロセス領域を識別します。実際には以下で述べるように「スキル管理」という方が適切かもしれません。成熟した組織は中長期的なスキルのニーズを識別して計画し、適当な時期に取得できるように管理しなければなりません。

トレーニング・プログラムの目的は、個人がそれぞれのロールを効果的かつ効率的に果たすことができるように、スキルと知識を開発することです。トレーニングは組織の責任ですが、ソフトウェア・プロジェクトは必要なスキルを識別し、プロジェクトのニーズが特有な場合は必要なトレーニングを提供しなければなりません。RUP における管理作業分野は、プロジェクトの要員配置時にこれらの問題を考慮します。

トレーニング・プログラム主要プロセス領域には以下の 3 つの目標があります。

- 目標 1: トレーニング・アクティビティーが計画されている。
- 目標 2: ソフトウェア管理と技術的なロールを実行するために必要なスキルと知識の開発に向けたトレーニングが提供される。
- 目標 3: ソフトウェア開発グループとソフトウェア関連グループの個人が、各自のロールを実行するために必要なトレーニングを受けている。

RUP のいくつかの側面はこれらの要求を実現します。ロールは正確に識別され、それらの適正能力が定義されます。ロールは通常、1 人の要員か、またはチームとして働く複数の要員で実現されます。プロジェクト・チーム・メンバーは通常、多くの異なるロールを果たします。人は多くの「帽子」をかぶることが可能のように、多くの異なるロールを実行できます。

ロールとは、人を指すわけではありません。そのロールを担う要員がビジネスにおいて行うべき振る舞いや、担うべき責務を記述したものです。

ほとんどのロールは、組織内部の要員により実現されますが、開発組織外部の人々も重要なロールを果たします。例えば、プロジェクトや開発する製品の利害関係者がこれに該当します。

ロールの各タイプは、チーム・メンバーが遂行しなければならないスキルと知識により慎重に定義されます。これらの定義に基づき、組織は不足しているスキルと、ロールを効率的に実行するために必要なトレーニングの特性を導き出すことができます。

RUP ガイドラインはトレーニングとして提供されるべきロールの適切なセットを設定します。コースの開発者は列挙されているロール・カテゴリーの 1 つです。

RUP でまったく扱われず、CMM で要求されるただ 1 つの側面は、将来に向けた新しいスキルを開発するために、トレーニングを識別、計画、提供することです。組織の戦略が今から 1 年後に電子商取引上の活動に重点を置くことである場合を想像してください。この目標はさらに分解する必要があり、おそらく使用する環境タイプと同じくらい実行する開発タイプに影響を与えるでしょう。チームはこの変更準備しなければならず、必要なトレーニングが定義、開発、または外部調達されなければなりません。レベル 3 におけるトレーニング主要プロセス領域は、これらの問題が慎重に管理、追跡、記録されることを要求します。

- このホワイト・ペーパーのセクション 3 は成熟特性と RUP の基本要件との詳細な対応付けを説明しています。このホワイト・ペーパーは、CMM 参照の各目標と対応する RUP のガイドラインとの間に系統だった対比を作成したといえましょう。これは意図的なものではなく、高度な成熟の主要な兆候に焦点を当てた結果です。
- RUP におけるソフトウェア品質は、一方ではテスト・アクティビティーを通じて、他方では品質保証計画書の作成タスクを通じて処理されます。しかし、ここでの例示は高度な成熟が真に意味するところに関する経験と観察に重点を置きます。つまり、レベル 1 またはレベル 2 組織とくらべてレベル 3 組織では何が異なるかということです。
- 次のセクションでは以上のようなテクノロジーの組み合わせを早期に採用した事例を紹介します。

RUP 実践的考察: RUP の活用方法

もっとも良くある質問は「既に実行した人はいるか」です。

この質問は、前述の概念を管理者や実務者に説明すると何度も繰り返されます。成熟度を上げ、プロセス指向を高め、リスクを管理し、成功することはすべてのソフトウェア・ビジネスにとって魅力的です。しかし、要求が頻繁に変更され、チームがトレーニング不足のテクノロジーを適用する必要があるなど、変化する状況の中で本当に実行できるのでしょうか。

Rational の顧客事例には Computing Devices International 社での経験があります。Computing Devices 社は大手の防衛電子技術、情報企業です。情報は、ほとんどすべての業界において、主要な競争優位性になっています。Computing Devices International 社の場合は優位性は情報だけではありません。同社の使命は、いつでもどこでも、オンデマンドの主幹業務情報を配信するエンドツーエンドのソリューションを提供することです。

同社は高品質のソリューションを一貫して、スケジュールどおりに納品する必要があります。同社の経験から得られた主なメリットには以下のものがあります。

- 反復設計手法による開発プロセスの加速
- 開発期間を 3 年から 1 年 7 カ月へ短縮
- 信頼できるスケジュール通りの納品の実現
- 顧客満足の上昇
- 開発コストを 33% 削減

Computing Devices 社が顧客向けに開発する高度なシステムは、厳しい要求に対して設計されるソフトウェアに大きく依存しています。Computing Devices 社はまず、安全性に関する重要ソフトウェアの作成プロセスに目を向けました。

組織内の複数の事業部は、構造化設計または「ウォーターフォール」手法に基づき、さまざまなツールを使用していました。結果的に、開発プロセスは遅く、一貫した手法が欠けていました。このため、納期遅延や、コストの膨張、ソフトウェア品質が高い基準を満たさないという状況を引き起こしました。Computing Devices 社は対策を実施すること、それも早急に実施することが必要であると気付きました。

Rational のもう 1 つの顧客事例として Skandia-IT 社を紹介します。この組織には、たったの 12 カ月間で 9 つの大規模保険システムがスケジュール通りまたはスケジュールより早く納品されました。

同社は、開発者とコンサルタントに最新の開発プロセスで引き付けることにより新規人材の雇用を加速し、ユースケースを使用した要求の勧誘、システムのモデル化、開発の指示により顧客満足を確保し、レガシー・システムをラッピングしたうえでコンポーネントとして使用することにより柔軟な 3 層アーキテクチャーを迅速に開発しました。これらは Skandia-IT 社が克服した課題の一部です。

Skandia-IT 社の管理者はこう説明します。「私たちが全プロセスを変更したのはいくつかの理由があります。最も重要なことはビジネス全体を再編したということです。製品志向組織から顧客志向組織への変更と同時に開発プロセスを変更しました。新しいシステムは、顧客が電話やインターネットなどを使用して自ら多くの対応を行うことを可能にします。Skandia 社の従業員は、システムの操作方法を学習する代わりに、保険のスペシャリストになるための作業により多くの時間を割くことが可能です。削減された時間は顧客との関係を保つために投資されます」。

Skandia-IT 社は新しいプロセスの導入に多額の投資を行いました。この投資はそれだけの価値がありました。品質は向上し、納期日が予測可能になったほか、熟練したプロジェクト・リーダーや開発者を雇用するのが容易になるという副次的な効果も生まれました。

以上の他にも、プロセスを重視し、RUP を使用して展開を加速する企業の例があります。Q-Labs 社は、インターネット放送ソフトウェア開発の急成長企業である Lysis 社と協業しています。立ち上げ時の環境において、プロセス成熟の高いレベルに注目するプロセス・テクノロジーや方法論は当てはまらない場合があります。この種の組織は本質的にペースが速く、敏感で、革新的なため、製品を早く出荷するニーズとプロセス管理を最大化するニーズとのはざまで、「プロセスというテーマはこうした多様な考え方をどの程度まで吸収するのか」という疑問が生じます。

この企業の特徴としては次のものがあります。

- 急成長しているソフトウェア・ユニットである。
- 開発速度が速い (6 カ月未満)。
- 先進のオブジェクト・テクノロジーを使用して優良な製品を供給する。
- ダイナミックな市場に適応する。

この環境は他のソフトウェア業界とまったく異なるわけではありませんが、時間の尺度は完全に異なり、予測可能性は生死に直結し、チームワークを根本的に変更する必要があります。学習の時間はほとんどなく、最良の選択肢は既に存在する十分実証済みの実践原則から開始することです。これは、Lysis 社で成長を導くために CMM を採用し、プロセスを導入するためにバックグラウンドとして RUP を採用したときに実際に行われたことでした。

ソフトウェアの実践原則を定義するニーズについては疑いの余地がありません。ソフトウェアの実践原則には理解、伝達、実行、管理の機能があります。また、社内外の組織との関係を構築する基礎を形成します。Lysis 社の事例の場合、営業部門や潜在的なソフトウェア開発請負業者との関係が対象になります。定義されるプロセスは常に柔軟でなければなりません。変更は、要員、基盤、リリース・スケジュール、採用されたテクノロジーにおいて急に起こる場合があります。開発プロセスはそれらを吸収する必要があります。迅速で適時の応答が要求されます。短い反復サイクルを持つ反復プロセス、リスク指向型、建設的な管理が推奨されます。製品ラインのコンポーネント全体でプロセスを繰り返すことができません。プロセス定義は、企業が成長するのに伴い、参加する新規要員に通知する際に役立ちます。

CMM において、プロセス定義 (成熟レベル 3) はプロセス再現性 (成熟レベル 2) に続きますが、こうした企業の場合は、RUP などの標準プロセスから開始することが再現性の促進と確立に有用であり、結果として大きなメリットを得ることができます。

Lysis 社で行われた方法は、開発の実践原則を CMM の要求に対して評価することから始まりました。この結果、プログラミング、オブジェクト・テクノロジー、ツール環境に最も関心を持つチームにより、プロセス指向が定着しました。こうした観察に基づき、プロセスを定義するための行動計画が厳しい時間制約を伴って作成されました。SPI をサポートするために経営層により設定されたビジネス目標には以下のものが含まれます。

- ターゲット市場で 1 番になるために開発プロセスを短縮する
- プロジェクト・モードから製品モードへの移行をスピードアップする
- 企業の成長を管理する
- 人々を満足させる
- 開始時の応答性を維持する
- 予測可能性と可視性を向上させる (IPO について、必要な四半期の結果)
- 製品品質の定量的な改善
- コストと潜在的な利益を管理する
- 顧客満足を最適化する
- パートナーとのインターフェースを識別し、管理する (作業の請負部分について)

SPI プログラムそのものは以下のように、RUP フレームワークを使用し、プロジェクトとして作成されました。

- 方向づけ = 評価
- 推敲 = プロジェクト計画 + Web ベースのリポジトリ定義とプロトタイプ化

- 作成 = プロセス定義 + 実装とツール使用
- 移行 = パイロット・プロジェクト + 展開

CMM 主要プロセス領域に対応するために作業パッケージが定義されます。各パッケージは目標、実行するアクティビティ、納入物、レビュー・プロセスを持ちます。一例として、ソフトウェア計画の目標を以下に示します。

行動目標:

- ソフトウェア計画の現在の優良な実践原則を文書化する
- プロジェクトに対する一般的な組織セットを定義する
- 計画のためのロールと責務を定義する
- プロジェクト・レベルの遂行事項プロセスを定義する
- プロジェクト・プロセスの標準セットに関する一般的なプロジェクト・ライフ・サイクルとカスタマイズのガイドラインを定義する
- 標準の作業分解構造 (WBS) を定義する
- 納入物の標準リストを作成する
- 開発計画のテンプレートとガイドラインを定義する
- プロジェクト計画プロセスを作成する
- 計画に必要なスキルとトレーニングを定義する

最初から RUP を使用することで以上の行動をスピードアップすることが可能であり、組織は確実な経験からメリットを得られます。SPI 構想は完成していませんが、いくつかの目に見える成功要因が観察されています。例えば、新しい製品の開発が十分に構造化されたり、計画が立てられ、定期的に見直されたりします。修正作業が実行され、ソフトウェア・チーム外の可視性が大きく向上します。

Rational は E-corporation 社に関する同様の事例について報告しています。E-corporation 社は、世界中の企業にデジタル・ビジネスの最前線を紹介している欧州の会社です。E-corporation 社は戦略的で、芸術的なソフトウェア・スキルを組み合わせ、インターネットや E ビジネスのベンチャー事業の着想、創造、実装を行います。

RUP を使用した E-corporation 社の主なメリットは以下のとおりです。

- チーム・メンバー間の意思疎通が改善され、効率性が向上する。
- 顧客の期待がより正確に実現される。
- 高品質で迅速な要求を満たすためのフレームワークが提供される。
- 反復的開発のリスクが減少し、専門家の知識や業界で認められた最善の実践原則に基づく実証済みプロセスが使用される。

E-corporation 社がすべてのプロジェクトで 1 つの共通プロセスを使用し、作業した結果から得た最大のメリットはコンポーネント、経験、アクティビティの効率的な再利用です。Rational Unified Process は、組織がさまざまなプロジェクトで経験を積む中で、常に微調整を行うことが可能です。すべてのアクティビティは一貫した構造と言語に則って文書化されているので、前のプロジェクトから得た多くの経験を新しいプロジェクトで活用することができます。

E-corporation 社は、すべてのサイトのすべてのプロジェクトで Rational Unified Process を展開し、チームの統一性をさらに強化する予定です。Rational Unified Process は同社の開発チームに、すべての作業を最適化するために実証済みの原則を適用し、意思疎通を簡略化し、すべての要求を満たす品質ソリューションを保証する能力をもたらします。

以上は、さまざまな組織が Rational Unified Process と CMM ガイドラインを組み合わせることによってメリットを得た例です。現在、私たちは、RUP を組織のプロセス資産を適用できる包括的な構造とみなす別の組織と協業しています。

結論

RUP と CMM の対応付けについて説明してきたことは ISO 15504 にも同様に当てはまります。これら 2 つの参照の基礎にある概念は類似し、成熟の特性を説明するために強調してきた主要因は ISO 15504 の能力尺度をサポートします。

このホワイト・ペーパーは、RUP の概念が高度に成熟した組織の目標にどれほど対応するかという点を明確にしてきました。したがって、RUP の採用は CMM の要求を適切に実現する方法です。

Telcordia Technologies 社から発表された最近の記事には、ビジネスにおける高い成熟の定量的結果が示されています。Telcordia 社は最近、成熟レベル 5 を評価し、以下のように報告しています。

- SPI 構想の開始以来、現場の欠陥密度の減少は 94% に達した。
- 1995 年以来、ソフトウェア・リリースの 98% 以上がスケジュール通り納品された。
- コード行のテストにかかるコストは 64% 減少した。

Telcordia 社の品質追及の道のりはこの成熟レベルに達するまで 6 年を要しました。

e-ビジネスや e-コマース分野のソフトウェア特化企業にとって、納品までの時間は重要であり、プロセスを最適化するための作業に 5 年も 6 年も費やすことは不可能です。したがって、競争力の維持に必要な成熟に達するため、考えられるただ 1 つのシナリオは以下のとおりです。

- ソフトウェア・プロセスの価値を理解する
- RUP などの既存フレームワークを採用する
- フレームワークを市場環境の特性に適応させる
- ビジネス上の影響を測定する
- カスタマイズされたプロセス・モデルを学習し、拡張する

RUP モデルは包括的なので、道に迷わないようにすることが重要です。そのため、Rational では「重要な RUP」を定め、モデルの採用時に重視すべき側面を設定しています [11, 12]。デフォルトの RUP を使用することは可能ですが、最初からすべてを実装する必要はありません。重要な基本要件は以下のとおりです。

- 開発企画書
- スケジュール
- 開発構想書
- リスク
- アーキテクチャー
- 変更依頼と欠陥の処理方法
- テスト
- ソフトウェア製品そのもの
- ユーザー・サポート・マニュアル
- プロジェクト評価

RUP の実行はそれ自体が目標ではありません。RUP は、グループとチーム内の困難な点を解決するのに役立ち、競争力のある目標を達成することを支援します。

著者の連絡先情報

Annie Kuntzmann-Combelles, Executive VP
Q-Labs France
28 Villa Baudran 94742 Arcueil cedex
電話 +33 (0)1 49 08 58 00
akc@objectif.fr

Philippe Kruchten, Rational Fellow
Rational Software Canada
pbk@rational.com

参考資料

- [1] K. Pulford, A. Kuntzmann-Combelles, and S. Shirlaw, 1995. *A Quantitative Approach to Software Management—The AMI Handbook*. Addison Wesley Longman.
- [2] Barry W. Boehm, 1996, "Anchoring the Software Process," *IEEE Software*, July 1996, pp.73-82.
- [3] Philippe Kruchten, 1996. "A Rational Development Process," *CrossTalk*, 9 (7), July 1996, p.11-16.
- [4] Robert McFeeley, 1996. *IDEAL: A User's Guide for Software Process Improvement*. Software Engineering Institute, Pittsburgh, PA, CMU/SEI-96-HB-001.
- [5] Steve McConnell, 1997. *Software Project Survival Guide*. Redmond, WA:Microsoft Press. (邦訳:「ソフトウェアプロジェクト サバイバル ガイド」、株式会社アルテア・ジャパン 訳、日経 BP ソフトプレス)
- [6] Mark Paulk, et al. 1993. *Capability Maturity Model for Software, Version 1.1*. Software Engineering Institute, Pittsburgh, PA SEI-93-TR-024.
- [7] Walker Royce, 1998. "Software Project Management: A Unified Framework" Addison Wesley Longman.
- [8] Jas Madhur et al, 1998. 「Rational Unified Process による CMM レベル 2 とレベル 3 の達成」ホワイト・ペーパー、Rational Software
- [9] Alec Dorling et al., 1999. *SPICE, the Theory & Practice of Software Process Improvement*, IEEE Computer Society.
- Philippe Kruchten, 2000. [10] Philippe Kruchten, 2000. *The Rational Unified Process—An Introduction, 2nd ed.*, Addison Wesley Longman. (邦訳:「ラショナル統一プロセス入門 第2版」、日本ラショナルソフトウェア株式会社 訳、藤井 拓 監訳、ピアソン・エデュケーション)
- [11] *Rational Unified Process, version 2000.02.10*, Rational Software Corporation
- [12] Leslee Probasco, 2000. "The Ten Essentials of RUP", *The Rational Edge*, December 2000, <http://www.therationaledge.com>.

Rational®

the software development company

Corporate Headquarters:
Rational Software
18880 Homestead Road
Cupertino, CA 95014
Tel: (408) 863-9900

Toll-free: (800) 728-1212
E-mail: info@rational.com
Web: www.rational.com
International Locations: www.rational.com/worldwide

Rational、Rational ロゴ、Rational Unified Process、Rational Rose は IBM Corporation の商標です。Microsoft、Microsoft Windows、Microsoft Visual Studio、Microsoft Word、Microsoft Project、Visual C++ および Visual Basic は、Microsoft Corporation の米国およびその他の国における商標です。他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。ALL RIGHTS RESERVED. Made in the U.S.A.

© Copyright 2002 IBM Corporation.
内容は予告なく変更されることがあります。