

## 第7章

# AgileとCMM

## 効率UPと品質改善は両立できるか

Agile特集の最後の章では、Agile開発方法論の今後考えられる現場導入の可能性として、CMM（Capability Maturity Model：能力成熟度モデル）との併用を考えてみたいと思います。

現在SEI-SW-CMM（以後単にCMMと記述）は、世界中の企業に、生産性の向上と品質改善で注目されている「**組織の成熟度モデル**」です。CMMは、Agile開発方法論とはまったく異なるアプローチで、組織の成熟度を高めていくことを意図しています。事実、一般にはCMMはHeavy weightなモデルとして位置づけられており、感覚的にはAgileの対極に位置するように感じられます。

では、本当にAgileとCMMは対極な存在で、両者の併用は成り立たないのでしょうか。本章では、このやや野心的なテーマを取り上げて少し考えてみます。

では文末に掲載する参考文献やSEIのWebサイト（図1）を参照してください。

### CMMとは？

CMMは、高い品質・生産性・競争力を実現するためのモデルで、組織成熟度のレベルに対応した「ゴール」と、推奨される“キープラクティス”（KP：Key Practice）と呼ばれる「ベストプラクティス」から構成されています（図2）。

CMMは、単なる「机上の学問的なモデル」ではなく、成功した多くのプロジェクトで実践されたBest Practiceから構成されています。CMMは、“進化型”（動的）のフレームワークであり、“前進的”な標準を定義しています。

ただし、あくまでモデルであるため、具

### CMMの概要

具体的にCMMとAgile開発方法論の併用を考えていく前に、CMMを簡単に整理しておきましょう。ただし、ここでは最低限必要な紹介にとどめますので、CMMの詳細について

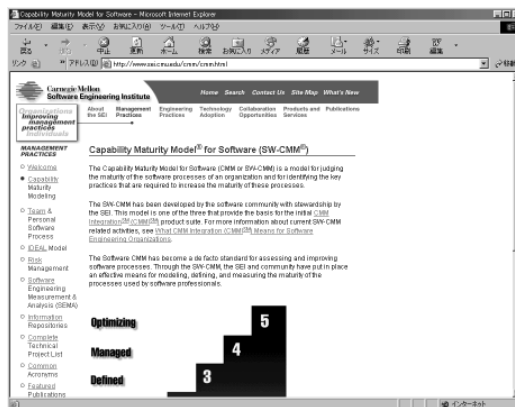


図1 SEIによるCMMのWebサイト  
(<http://www.sei.cmu.edu/cmm/cmm.html>)

# 特集1 ● Agileなソフトウェア開発

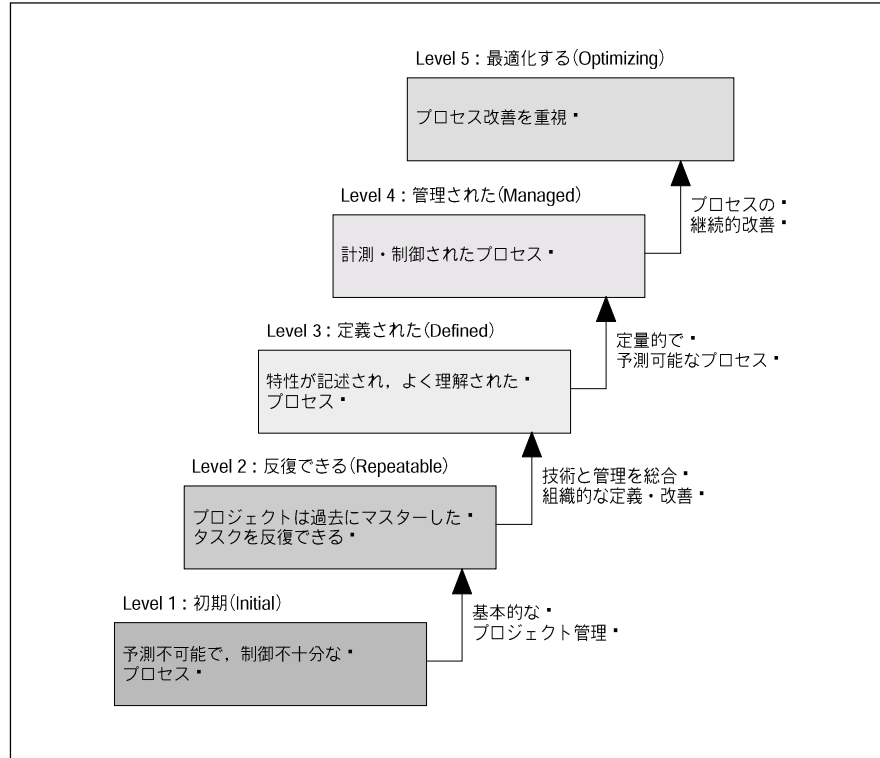


図2 SW-CMMモデルのレベル

体的な開発プロセスが記述されているわけではありません。各企業、組織あるいはプロジェクトは、適切なプロセスを実装する必要があります。これは、「その組織でどのような活動を通じてKPを実現するか」をプロセス定義することを意味します。つまり、その組織がおかれている開発システムの特徴、ビジネス環境、企業文化を考慮して、プロセス定義が必要となるのです。

## ソフトウェア開発改善モデル～IDEALモデル

CMM自体は、組織成熟度のレベルと、KPA (Key Process Areas) <sup>注1)</sup> ごとのベストプラクティスであるKPが記述されているだけです (図3)。具体的にどのようにして

組織がCMMの各レベル相当の成熟度を高めていけばよいのかについては、直接言及されていません。

各企業、組織あるいはプロジェクトは、適切なプロセスを定義すると同時に、どのように品質改善活動を進めていくのかを計画する必要があります。SEIでは、「IDEALモデル」という「品質改善活動のモデル」を発表しており、CMMを用いて成熟度を向上させて行くプロセスが提供されています。

ただし、IDEALモデルはCMM専用の品質改善活動ではありません。IDEALモデルはCMMを前提としているわけではないため、ISOやその他のモデルを利用して品質改善活動を行う場合にも利用可能です。

注1) CMMでは合計18エリアが存在します。KPAとは、組織がソフトウェアプロセスを改善するために、焦点を当てるべき「鍵となる」プロセス領域のことです。

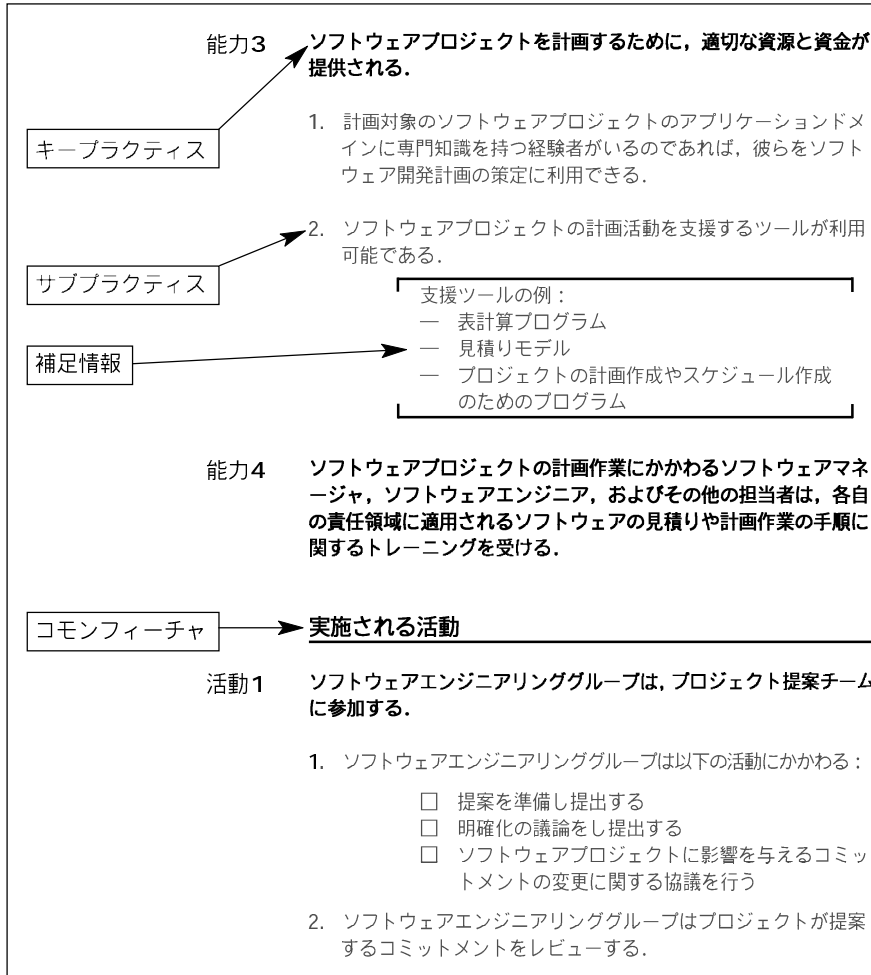


図3 CMMのキープラクティス (KP) の例 (「能力成熟度モデルのキープラクティス 1.1版」より)

IDEALモデルは全部で5つのフェーズから構成されており、品質改善活動を進めていく具体的な内容になっています (図4, 図5)。

### ○ CMMは官僚的か?

CMMについて、「官僚的」とか「厳格過ぎる活動が要求される」などといった批判を耳にすることがあります。おそらく、「CMMは明確に定義された作業プロセスノ

管理プロセスを要求しており、そのプロセスに厳密に従わなければならない」という印象があるからでしょう。また、プロセスの手順が文書化され、開発成果物についても客先に納入するという点も、「作業上の中間成果物も文書化が要求されている」というように受け止められているようです。

以上は必ずしも間違いではありませんが、CMMの正しい解釈とはいえません。そこで、ここではCMMの基本的な解釈につい

# 特集1 ● Agileなソフトウェア開発

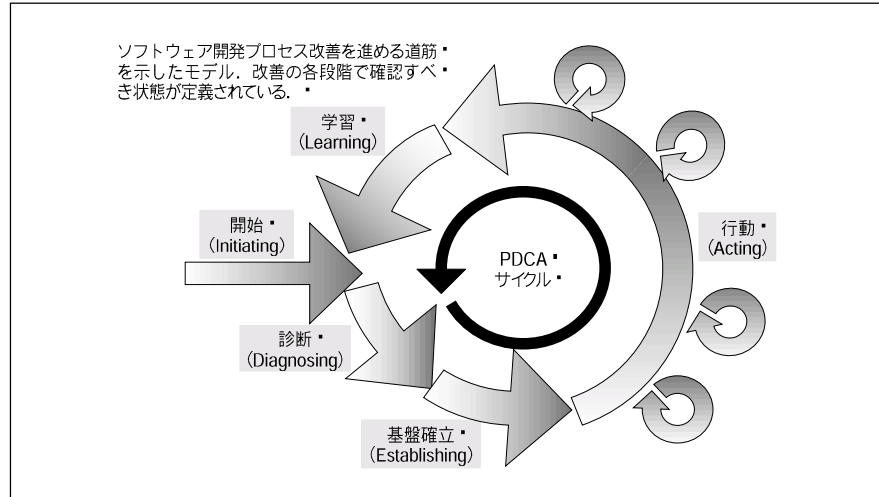


図4 SEI IDEALモデル

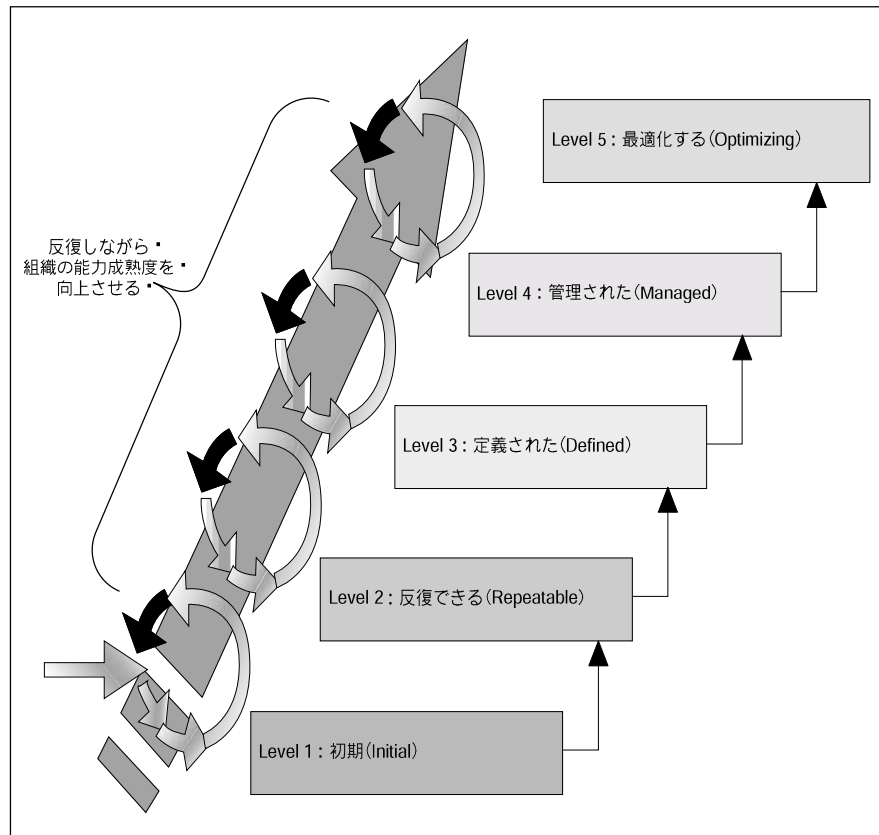


図5 IDEALモデルとCMMの2つのモデルによる品質改善 (CMU/SEI-93-TR-24 (カーネギー・メロン大学) より)

CMMの解釈	解説
レベル確認は、各KPAに記述されているゴールを満たしていること。 各KPAに記述されているKPは、推奨されるBest Practiceであり、「典型的な活動例」である。	KPは、あくまで推奨される活動である。KPAのゴールに記述されている状態を達成可能であれば、組織やプロジェクトが独自に活動を定めてもよい。これをAlternative Practiceと呼ぶ。
各KPAが組織やプロジェクトでの活動に当てはまらない場合は、「適用外」(Not Applicable : NA)とする。	組織やプロジェクトにとって実施する必要のないKPAやKPは、NAとなる。たとえば、Level 2に「外注管理 : SCM」が存在するが、開発を外注に委託していなければ、このKPAは対象外である。
CMMはモデルであり、具体的なプロセスは組織、プロジェクトが検討し、実装しなければならない。	ソフトウェア開発を行っている企業、組織では、対象とするジャンル、開発の規模、開発期間、開発条件や環境などがそれぞれ異なる。よって、組織の文化や目標、開発システムの特徴に合わせてプロセス定義をする必要がある。

表1 CMMの解釈

で確認してみたいと思います (表1)。

確かに、CMMのKPの中では、作業手順書や各種計画書の文書化を要求する記述が多く見られます。しかしながら、必ずしもドキュメントを作成して、厳密な手順に従うことを要求しているというわけではありません。この点は、組織がCMMをどのように利用し、生産性向上、品質改善を実施するかの方針次第といえます。

つまり、ある組織やプロジェクトが品質や生産性の向上を、「明確な作業の手順化」と「成果物のドキュメント化」によって実現するという戦略を選択するならば、伝統的な開発方法論に近いものになるでし

ょうし、官僚的な開発プロセスに近くなることは頷けます。大規模な開発では、ある程度の手順化とドキュメント化が必要になるのはやむをえないことでしょう。しかし、重要な点は、それが官僚的か否かということよりも、効果を出すために正しい戦略であるかどうかということです。

逆にいえば、開発規模も小さくなく、開発者同士が直接話し合えるような規模、開発環境であれば、作業手順などを文書化する必要性も低減されると思います。CMMのKPを遵守しながらも、効果的にKPAごとに要求されているゴールの状態に組織があれば、それで十分効果的な開発プ

## コラム：CMMの誤解

SEI-SW-CMMは、ISOと違って「認証」という仕組みではありません。CMMでは、組織の成熟度を「判定」するために「CBA-IP1」というアセスメントを実施しますが、「認証」の仕組

みではないため、「取得」「合格」という表記は正しくありません。

しかし、多くの雑誌や新聞などで、「〇〇企業がCMMのレベル2を取得」とか「〇〇企業がCMMのレベル2に合格」と記

述されているのを目にします。認証ではありませんので、この場合「CMMのレベル2に到達した」とか「CMMのレベル2に相当の活動が確認された」と書くのが適切でしょう。 ■

# 特集1 ● Agileなソフトウェア開発

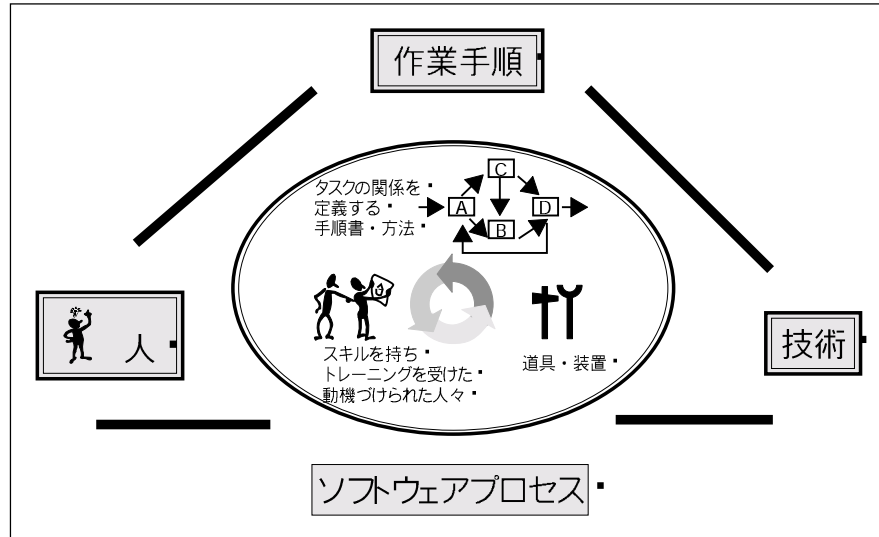


図6 CMMのソフトウェアプロセスの概念(『SEI公認CMM入門トレーニングテキスト』より)

プロセスであり、目指すレベルの組織成熟度と判定されることでしよう。

このように、CMMが官僚的か否かは、組織あるいはプロジェクトの品質改善の方針とプロセス定義に大きく依存することを知らせていただきたいと思います。

人が作業に与える影響や仕事へのやり甲斐、人と人との約束プロセスなど、人間の持つ心理的な部分についても重要視しています。

SEIも、ソフトウェア開発を図6のように位置づけており、“人”を大変重要視しています。この点は、Agile開発方法論の多くが人を中心に考えて重要視していることと変わらないといえるでしょう。

## CMMの価値体系

CMMの概要についての確認が終わりましたので、次にCMMがどのような価値体系に基づいているかについて、考えてみたいと思います。

### 人の重視

CMMの生みの親的存在であるWatts S. Humphreyは、“人”や人が集まって形成するチームやプロジェクトあるいは組織のことを、非常に重視しているということです。Humphreyの著書『ソフトウェアプロセス成熟度の改善』(参考文献14)では、

### プロセスの重視

CMMがプロセスを重視するのは、多くのソフトウェア開発現場では開発プロセスが存在せず、効果的な開発が行われていないという調査事実に基づいています。

開発プロセスが存在しないか、あるいは存在したとしても未熟なものである場合には、過去の開発資産やメトリクスなどのデータを活かすことなく開発が行われています。

CMMの狙いは、作業手順や成果物をプロセスとして再利用することで、それまで担当者ごとにアドホック(場当たりの)な作業が進められていた開発現場に秩序をも

「ソフトウェアプロセス」の定義	ソフトウェア開発プロセス…ソフトウェアおよび関連成果物の開発と保守の使用する活動、手法、プラクティス及び変換作業の集合（「SEI公認CMM入門トレーニングコーステキスト」より）
ソフトウェア開発上の問題とプロセス	①ソフトウェア開発における諸問題の85％はプロセス（仕組み）に起因しており、ソフトウェアの品質は、それを開発・保守するプロセスに大きく依存する。成果物（プロダクト）だけに意識が行くと、問題の根本的な原因がつかみにくい ②良い方法についての知識を逃す
プロセスを重視する理由	①成果を繰り返せる ②プロジェクトの傾向が予測できる ③成果物の特性が予測できる
CMMが解決しようとしている問題	①システム開発の「コスト・スピード・品質」に満足できない、予測できない ②ユーザーニーズに追いつけない ③新技術の導入が難しい ④開発現場の士気、教育、環境が十分でない
CMMによる開発プロセス改善の価値	①TQC/TQMの概念を（主にソフトウェア）開発に適用した ②品質を直接の問題としてとらえる ③品質向上が生産性（やスピード）の向上につながる ④成熟度の概念に基づく体系的アプローチを重視 ⑤プロジェクト管理、標準化、測定データ重視、継続的改善 ⑥組織、チームとしての取り組みを重視
目指すゴール	①ビジネス上の競争力強化 ②顧客満足度の向上 ③生産性（QCD：Quality/Cost/Delivery、品質／コスト／納期）の向上

表2 CMMの考え方

たらずことにあります。また、リスクやシステムの要件変更、ビジネスニーズの変化を積極的に予見することで、リスクを回避したり、事前に効果的な対応策を講ずることを狙っています。

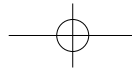
プロセスを繰り返し利用することによって、計画の効果的な見積り方法のノウハウや当初の計画からのずれを、各種メトリクスへフィードバックし、次回のプロジェクトで利用することを狙っています。結果として、開発プロセスはより効果的なものへと発展していき、各種メトリクスの精度も向上していくことになります。CMMの基本的な考え方を表2に示します。



## Agile開発とCMM

CMMは、一般に大規模開発向きとされています。この点については、CMMの歴史上の経緯からみて事実でしょう。では、少人数の開発に絶対向かないかといえば、必ずしもそうは思えません。事実、5～10人程度の開発チームで開発規模も小さく、開発期間の短いプロジェクトが多い組織でも、CMMを用いた品質改善活動は行われており、大きな効果を挙げている例も数多くあります。

したがって、CMMのモデルを参考にし



# 特集1 ● Agileなソフトウェア開発

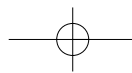
方法論	SEI-SW-CMM	XP	Scrum
価値体系の軸			
開発者の役割	スキル、役割、モチベーションなどを重視する。ただし、各種メトリクスは属人性を極力排除する。	スキル、役割、モチベーションなどを非常に重視する。人が中心。プロセスや技法は手段に過ぎない。基本的に分析からテストまで同じ人が担当。	スキル、役割、モチベーションなどを非常に重視する。人やチームが中心。プロセスや技法は手段に過ぎない。
プロジェクト管理の定義	CMMをモデルとして提唱。組織やプロジェクトでCMMをベースのプロセス定義化。	具体的なプロジェクト管理はあまり明確になってない。	Scrumはプロジェクト管理方法論。独特の管理プロセスを持ち、プロジェクトを推進するプロジェクトリーダーとして、ScrumMasterが存在する。
エンジニアリングプロセスの定義	CMMでは定義されていない。利用組織、プロジェクトで定義する。	特に決められていないが、基本的な作業の流れは存在する。	特に決められていない。システムの特徴や開発状況に合わせて別途検討する。
開発者間のコミュニケーションの位置づけ	重視する。定義したプロセスに従う。	口頭によるコミュニケーションを非常に重視する。	口頭によるコミュニケーションを非常に重視する。
ドキュメントなどの成果物の位置づけ	計画や作業手順の文書化を推奨。成果物の文書化も組織やプロジェクトの方針によって有無が決まる。	可能な限り作成しない。成果物は作業のオーバーヘッドを生む。動作するシステム、ソースコードが重要。	可能な限り作成しない。成果物は作業のオーバーヘッドを生む。動作するシステム、ソースコードが重要。
技法の位置づけ	定義していない。	ヘアプログラミング、リファクタリング、テストファースト、計画ゲーム。	あまり限定していない。
再利用性の考慮	プロセスや各種成果物、メトリクスのデータの再利用を狙っている。	原則考慮しない。	原則考慮しない。
開発規模	中規模～大規模。ただし、小規模にも適用可能。	小規模。	小規模。
対応分野	当初は防衛システム。ただし、あらゆる分野で利用され汎用的。	汎用的。予見的開発が困難な分野。ただし、ミッションクリティカルはあまり意識していない。	開発組織の変更が多く、成熟度が期待できない分野。ITが代表的。
Toolの位置づけ	定義していない。	特に規定しない。ただし、テストToolやリファクタリングToolを利用する。	特に規定しない。
変化に対する適応性	重視している。プロセスを常に最適になるように進化させていく。	非常に意識しており、変化に強い。	非常に意識しており、変化に強い。
方法論のタイムスコープ	1つのプロジェクトから組織の成熟度までカバー。	基本的にプロジェクト単位。	基本的にプロジェクト単位。
方法論の目的	最適化された手順とメトリクスを用い、属人性を排除したシステムティックなアプローチによる生産性と品質向上。	質の高いシステムを効率よく開発する。システムを期日までに確実に完成させる。変化に対応し、本質的な作業に注力する。	システムを効率よく開発する。システムを期日までに確実に完成させる。変化に対応し、本質的な作業に注力する。

表3 CMMとAgile方法論の価値体系の比較

て組織の開発プロセスや管理プロセスを定義すれば、生産性や品質を向上できる組織は、決して少なくはないはずです。当然、少数数のプロジェクトでは、大規模開発のプロジェクトのような活動とはかなり異なる点も多いかもしれません。この場合には、

CMMを額面通りに適用しても効率的ではないでしょう。

さらに、いずれかのAgile開発方法論を採用しながら、CMMを併せて利用するには、CMMについてよく理解し、CMMのKPAごとのゴールを満たす活動を、Agile





主要な検討すべき項目	内容
目的とゴール	開発方法論やCMMを導入する目的は？ 目標となる状態は何か？
価値体系	システムの特徴、ビジネス環境、人や組織の持つ価値観にとって、開発方法論や活動自体が適切なものか。
人、Tool、予算などの資源	開発方法論や品質改善活動を行う場合に、人、Tool、予算などの資源があるか。無理な計画、導入は破綻する。
組織の体制	開発方法論や品質改善活動によっては、組織の体制が重要となる。必要に応じて組織編成やグループ編成が必要となる。
支援	開発方法論や品質改善活動の導入や活動はいろいろなことで、人、組織、企業の支援が必要である。特に、上級管理者の支援がないと活動に苦勞する。

表4 CMMの解釈の一部

開発方法論の活動の中でどのように実現するかを検討する必要があります。具体的には、そのAgile方法論内で、どのようにしてKPを実現するかということです。

CMMはあくまでベストプラクティスから構成されるモデルであり、開発プロセスを定義していないため、このようなAgile開発方法論との併用も可能になります。ただし、この場合多少なりとも、余分な作業やドキュメント開発が増えることは仕方ないでしょう。

表3は、CMMとAgile開発方法論の価値体系を比較したものです。濃い灰色の価値体系の軸は価値観が似ている軸で、薄い灰色は、濃い灰色ほどではありませんが価値観が近いと考えられる軸を表しています<sup>注2</sup>。また、白い部分は少し距離があると思われる軸です。ただし、この表は筆者個人の主観的な要素が多分に含まれていると考えてください。異論もあるかもしれません。

以上の比較表についてはいろいろな見方や意見があると思いますが、各Agile開発方法論の間でも価値観がまったく同等ではないことを考慮すれば、一般に受けるイメージほどの距離はないのでは、という気がします。そうであれば、CMMとAgile方法論の併用の可能性が期待できますし、お

互いに補完し合う関係としての利用も可能かもしれません。

可能性がある反面、CMMとAgile開発方法論の併用については、まだ多くの検討事項が存在しており、実のところ今ここで明確な指針と結論を出すことは早計です。すでに記述したように、外注管理、システムティックな構成管理、高い精度のメトリクスについては、Agile開発方法論に効果的に取り込む方法を実現しなければなりません。対象とする開発システムについても、もう少し細かく整理する必要があるでしょう<sup>注3</sup>。

## 何が目的なのか

多くの場合、開発方法論の欠陥や問題点が揶揄されますが、実際には導入する側に問題がある場合も少なくありません。

企業、組織、プロジェクトなどが、Agile開発方法論、Heavyweight開発方法論、CMMなどを導入する際には、何を目的、ゴールにするかははっきりすることが重要です。そして、目的やゴールに則した導入計画や改善計画が必要です。この部分を見落とすと、方法論の導入や改善計画は頓挫するでしょう。

注2) 組織やプロジェクトのCMM/開発プロセスの導入目的や導入方針によって、距離が変化すると思われます。ここでは、「組織やプロジェクトの導入方針次第で併用の可能性が期待できる」という意味でとらえてください。

注3) 現在までのところ、開発システムやビジネスドメインについて、「IT分野」と「政府請負型システム」というように、極端に二極化して議論されることが多いようです。しかし、現実にはそう簡単に分類できるのではなく、「開発規模」や「ビジネス環境変化の程度」、「製品のシリーズ化の傾向」、「組織内における開発システム群の類似度」、「組織の安定度」など、さまざまな事情を考慮して整理する必要があるでしょう。

## 特集1●Agileなソフトウェア開発

表4に、特に重要と思われる事項の一部をピックアップしまとめてみました。



### 最後に

Agile開発方法論を中心に、価値体系を1つの視点として、開発方法論・管理プロセスについて紹介してきました。

重要なことは、XPやScrum, RUPやCMMなどは、いずれも方法論やモデルに過ぎないということです。近年、方法論やモデルが話題になるにつれ、方法論やモデルの利用自体が目的化する傾向が見られる気がします。

すでに記述したように、重要なのは、組織として方法論を採用する目的を明確にすることです。そして、それに沿った導入計画や改善計画を立案することです。「何のためにそれを導入しようとしているのか」をはっきりさせ、それに基づく戦略を立てることが重要であり、方法論はあくまで目的達成のための手段であることを再認識する必要があります。

最後になりましたが、この記事を書くにあたってメタボリクスの山田正樹さん、SONYの小野剛さん、横河電機の井上健さん、中村淳さんには有益な情報提供で

支援いただきました。この場を借りてお礼を申し上げます。

今回の記事が、開発方法論やプロセスを導入する際の何らかのヒントになれば幸いです。 ■

### ■参考文献

#### 開発方法論関連

1. 『Agile Software Development with Scrum』, Ken Schwaber, MekeBeedle 著, PRENTICE HALL
2. 『XPエクストリーム・プログラミング検証編』, Giancarlo Succit他著, 小野剛その他訳, ピアソン・エデュケーション
3. 『XPエクストリーム・プログラミング入門 ソフトウェア開発計画の究極の手法』, Kent Beck 著, 長瀬嘉秀監訳, 飯塚麻理香訳, ピアソン・エデュケーション
4. 『XPエクストリーム・プログラミング実行計画編』, Kent Beck他著, 長瀬嘉秀監訳, 永田渉・飯塚麻理香訳, ピアソン・エデュケーション
5. 『Agile Modeling Effective Practices for eXtreme Programming and the Unifeid Process』, Scott.W.Ambler, WILEY.
6. 『ラビッド・アプリケーション・デベロップメント (1) (2)』, James Martin 著,

## CMM日本語版の入手方法

日本では、SEI発行のテクニカル・レポート24 (TR-24) とテクニカル・レポート25 (TR-25) が、SEAのCMM翻訳分科会の献身的な活動で翻訳され、利用可能となっています。下記のURLからもダウンロード可能です。 ■

<http://www.ijnet.or.jp/seaf/CMM/index.html>

- 能力成熟度モデルのキーブラクティス 1.1版
- ソフトウェア技術者協会CMM研究会
- 公式日本語版
- ISBN:4-916227-05-0



芦沢真佐子他訳, リックテレコム

7. 『インフォメーション・エンジニアリング統合化CASEのための方法論 (1)』, James Martin 著, 三菱CC研究会IEタスクフォース訳, プレンティスホール/トッパン
8. 『インフォメーション・エンジニアリング統合化CASEによる計画と分析 (2)』, James Martin 著, 三菱CC研究会IEタスクフォース訳, プレンティスホール/トッパン
9. 『インフォメーション・エンジニアリング統合化CASEによる設計と製作 (3)』, James Martin 著, 三菱CC研究会IEタスクフォース訳, プレンティスホール/トッパン
10. 『ラショナル統一プロセス入門 第2版』, フィリップ・クルーシェテン著, 藤井拓監訳, 日本ラショナルソフトウェア(株) 訳, ピアソン・エデュケーション
11. 『UMLによる統一ソフトウェア開発プロセス オブジェクト指向開発方法論』, Ivar Jacobson他著, 藤井拓監修, 日本ラショナルソフトウェア(株) 訳, 翔泳社
12. 『Agile Software Development』, Alistair Cockburn, Addison-Wesley
13. 『Adaptive Software Development』, Jim Highsmith, Dorset House

CMM関連

14. 『The Practice of Capability Maturity Model, Version 1.1』, Software Engineering

Institute (SEI), SEI-93-TR-025

15. 『ソフトウェアプロセス成熟度の改善』, Watts S. Humphrey 著, 藤野喜一 監訳, 日科技連出版社 1991.9
16. 『能力成熟度モデル1.1版』, Mark C. Paulk 他著, ソフトウェア技術者協会訳, CMU/SEI-93-TR-24 1993.2 (ダウンロード <http://www.ijnet.or.jp/sea/CMM/index.html>)
17. 『能力成熟度モデルのキープラクティス1.1版』, Mark C. Paulk 他, ソフトウェア技術者協会訳, CMU/SEI-93-TR-25 1993.2 (ダウンロード <http://www.ijnet.or.jp/sea/CMM/index.html>)
18. 『IDEAL: A User's Guide for Software Process Improvement』, Bob McFeeley, CMU/SEI-96-HB-001 1996.2, カーネギー・メロン大学 ソフトウェア工学研究所 (Software Engineering Institute, <http://www.sei.cmu.edu/>)
19. 『ソフトウェア能力成熟度モデル CMMガイドブック』, Kenneth M. Dymond 著 西村高志他訳 日刊工業社
20. 「ソフト開発プロセス改善手法と事例研究 品質や生産性を劇的に向上」, 日経コンピュータ 1997.3.17号
21. 「ソフト開発のプロセス導入効果と実践方法米モトローラの事例に学ぶ」, 日経コンピュータ 1997.12.22号

## コラム : CMMとAgile

Agile Modelingのメーリングリスト([www.agilemodeling.com](http://www.agilemodeling.com) から[agilemodeling@topica.com](mailto:agilemodeling@topica.com)へ参加)でも, CMMとAgileの併用が可能かという話題が登場しています。CMMとAgile

の両方の導入を考えている方には参考になると思います。

また, 下記のURLには, Agile開発方法論の1つであるXPに対する, オブジェクト指向方法論やCMMの専門家たちによるコメン

トが掲載されています。興味がある方は参照してみてください。■

<http://www.computer.org/seweb/Dynabook/Commentator.htm>