

GS LetterNeo Vol.122

2018 年 9 月

CMMI®の組織プロセス成熟度概念の振り返り(2)¹

静永 誠

はじめに

CMMI (Capability Maturity Model Integration: 能力成熟度モデル統合)は、プロジェクト管理やプロセス管理の能力を段階的にレベル表現できるモデルです。主にシステム開発およびソフトウェア開発で適用されることが多いモデルですが、定義上はサービス業全般を含む、様々な領域へ適用可能なものとされています。

新バージョン(CMMI V2.0)が久し振りにリリースされた機会に、前回(Vol.121)と今回の 2 回に渡って、CMMI の特徴である段階的な成熟度レベルについて、概要と歴史を振り返ります。

本号では CMMI のコンセプトの作成経緯について紹介します。

成熟度レベルの作成経緯

CMMI の前身にあたる Software CMM® (Capability Maturity Model) は、1990 年代に Carnegie Mellon 大学のソフトウェア工学研究所 (Software Engineering Institute、以下 SEI²)により公開されました。CMM の開発の経緯をみると、ソフトウェア関連企業に関する能力評価法の開発という米国連邦政府から要請へ応えるための、SEI での研究がその前身になっています。

Software CMM や CMMI にみられるプロセス成熟

度フレームワークの元々のコンセプトは、1980 年代初めに Watts Humphrey と彼の同僚によって開発されました。Humphrey は、他の業界での品質管理の成功をみて、開発プロセスを継続的に改善する方法として、Shewart-Deming 改善サイクル(Plan-Do-Check-Act)をソフトウェア組織に導入しようとしていました。

しかしながら、Humphrey の組織では、Shewart-Deming サイクルに似た手法を使用して 10 年ほど高度なソフトウェアテクノロジーを導入していましたが、大きな成功は得られませんでした。そこで Humphrey は、継続的改善への障害を体系的に取り除くために、Shewart-Deming サイクルを段階的に導入する必要があると考えます。Deming の原則に基づいた継続的な改善を支援する環境を構築するには、組織が特定の順序で実装上の問題を排除しなければならないという考えから、CMMI の組織プロセス成熟度へと繋がる研究が進められました。[1]

ここでは、CMMI の前身まで含めた開発の歴史を、成熟度レベルの概念が確立されるようすを中心に紹介します。

ソフトウェアへの TQM の適応

SEI のソフトウェアプロセス成熟度に対する研究の底流にある基礎的な前提は、ソフトウェア製品の品質の大部分は、ソフトウェアの構築に使われる、開発および保守のプロセスの品質によって決まるというものでした。

そのような前提のもと、SEI ではプロセス品質の改善に係る研究のなかで、Walter Shewhart の統計的品質管理や、Edwards Deming の Deming サイクルの概念を取り入れ、TQM (Total Quality Management) の原理をソフトウェア開発組織に適応させる研究が行われていました。

ソフトウェア開発のプロセスの品質を CMMI のような成熟度の段階構造で示すフレームワークは、Philip Crosby が著書「Quality is Free」で説明している品質管理成熟度表 (Quality Management Maturity Grid) を、Humphrey がソフトウェアのプロセスに適応することで作成されました。[2]

品質管理成熟度表は、クロスビーが成熟度という概念を適用した実務経験から作成したものであり、管理者が専門的な品質管理の訓練を受けなくても、5 段階に示された品質ポイントをなぞることで品質改善ができるように工夫されたものになります。[3] [4]

ソフトウェアプロセス成熟度のフレームワークと質問票

1986 年、ソフトウェア関連企業に関する能力評価法の開発を米国連邦政府から要請され、SEI は MITRE 社の協力を得ながら、ソフトウェアプロセスの改善を目標とした、プロセスの成熟度に関するフレームワークの開発を開始しました。翌 1987 年、SEI はその成果として、成熟度フレームワークの簡単な定義と、ソフトウェアプロセスの成熟度に関する質問票を発表しました。

この成熟度フレームワークでは、継続的プロセス改善の一連の土台となる 5 段階が識別され、あわせて組織のソフトウェアプロセスの成熟度レベルを測定する順序尺度の定義がされています。この成熟度レベルのコンセプトは、以降の Software CMM および CMMI の進化のなかでも、常に存在し続ける重要なものとなりました。

さらに、この時期の決定事項のなかで重要なものに、成熟度レベルの違いでは、プロジェクトのコミットメントの識別および管理に注力することと、計画の管理に注力することが重大であるとした決定があります。[2]この決定が、成熟度モデルと Crosby の品質管理成熟度表などとの大きな違いのひとつになりました。以降、Software CMM や CMMI の V1.3 までの各バージョンにおいても、成熟度レベルは、まずプロジェクト管理に関する要求をレベル 2 で定義し、技術的な手法はレベル 3 以降で標準化などを要求する内容になっています。

ここで注意が必要な点は、成熟度レベルが 1 や 2 の組織でも、技術的な活動は実施されているということです。成熟度レベルが 1 や 2 の組織では、技術的な活動は場当たりの (ad hoc) に実施されていることが多いと考えられています。そのうえで、組織の成熟度を上げるには、技術的な手法の標準化などよりも、コミットメントの管理などに注力する方が優先される、というフレームワークになっています。

参考までに、成熟度フレームワークの説明のなかで、各段階へ進むために対応すべき重要なポイントとして示されているポイントを表 1 に紹介します。[5]

表 1 成熟度フレームワークのレベル向上で対応すべきポイント

対象レベル	対応すべきポイント
レベル 2	基礎的なプロジェクト制御 (project control) の導入 : <ul style="list-style-type: none"> ● プロジェクト管理 (Project Management) ● 管理者の監視 (Management Oversight) ● 成果物保証 ● 変更管理
レベル 3	<ul style="list-style-type: none"> ● プロセスグループ (開発プロセスの改善に集中して取り組むグループ) の確立 ● プロセスアーキテクチャ (標準プロセスとほぼ同じ) の確立 ● ソフトウェア工学手法の導入
レベル 4	<ul style="list-style-type: none"> ● プロセス測定の最低限の基礎的なセットの確立 ● プロセスデータベースの確立 ● プロセスデータ分析へのリソース確保 ● 品質目標に対する成果物品質の査定
レベル 5	<ul style="list-style-type: none"> ● プロセスデータ収集の自動化 ● 成果物からプロセスへの管理の焦点の移行

「Managing the Software Process」出版と Software CMM のリリース

1989 年、Humphrey は研究成果として「Managing the Software Process」を出版します。[6]この書籍は成熟度フレームワークの内容を詳細に示すとともに、ソフトウェアプロセスにプロセス管理の原理を適用するという動きに大きな影響を与え、以降の Software CMM 開発を促進させるものとなりました。「Managing the Software Process」の内容をベースに、コミュニティ向けのレビュー版のリリースを繰り返した後、プロセス成熟度の正式なモデルとして 1991 年に Software CMM v1.0 がリリースされます。

Software CMM では、正式に、成熟度レベルがキープロセスエリア (key process area) という観点から説明されるようになりました。キープロセスエリアは組織がプロセスを改善するために注力すべきところを指し示すものになります。

Software CMM では、全てのキープロセスエリアは

1 つの成熟度レベルに紐付く形で構成されており、複数の成熟度レベルを跨ないように定義されました。これは、「Managing the Software Process」から Software CMM リリースまでのレビュー期間で加えられた重要な変更のひとつといえます。[2]

表 2 は、キープロセスエリアのレベルでの「Managing the Software Process」と「Software CMM」の対応関係を一部、簡単に示したものです。[2][6]

表中、「Managing the Software Process」の「Software configuration management」の 1 と 2 が、「Software CMM v1.0」では成熟度レベル 2 と 3 の各レベルで、異なるキープロセスエリアが対応付けられています。これは、構成管理に関する内容について、Software CMM では、成熟度レベル 2 の組織が取り組むべき内容が「Software configuration management」でまとめられ、成熟度レベル 3 の組織が取り組むべき内容が「Software product engineering」でまとめられていることを示しています。

また、「Managing the Software Process」では成熟度レベル 3 で扱われるものとして付録の中で取り上げられているプロジェクト管理関連の活動が、「Software CMM v1.0」では成熟度レベル 3 のキープロセスエリアとして定義されています。これは、例えばプロジェクト計画に関する内容について、Software CMM では、成熟

度レベル 2 の組織が取り組むべき内容が「Project planning」でまとめられ、さらに成熟度レベル 3 の組織が取り組むべき内容が「Integrated software management」でまとめられ、キープロセスエリアとして追加されていることを示しています。

表 2 「Managing the Software Process」と「Software CMM」の対応

成熟度レベル 2	
Managing the Software Process	Software CMM v1.0
(付録 A で「プロジェクト管理」の活動の中で取り上げられている)	Requirements management
The project plan	Software project planning
Managing software organization	Software project tracking and oversight
(付録 A で「プロジェクト管理」の活動の中で取り上げられている)	Software subcontract management
Software quality assurance	Software quality assurance
Software configuration management 1	Software configuration management
成熟度レベル 3	
Software engineering process group	Organization process focus
Defining the software process	Organization process definition
Software standard	Software product engineering
Software testing	
Software configuration management 2	
Software inspection	Peer review
(付録 A で「組織」の活動の中で取り上げられている)	Intergroup coordination
(書籍中の各章で取り上げられている)	Training program
(付録 A で「プロジェクト管理」の活動の中で取り上げられている)	Integrated software management

このように、例えばプロジェクト計画を扱う内容であっても、成熟度レベルに沿って異なるキープロセスエリアへと分割した背景には、各キープロセスエリアは成熟度レベルを進めるために特に重要な少数の項目（「vital few issues」）だけを提示する、という決定がありました。それは、組織が一度にあまりにも多くの改善策を取り上げ過ぎる失敗を防げるようにすることを意図したものになります。[2]

Software CMM で強調している組織的な改善は、低い成熟度レベルにあるプロセスエリアは、高いレベルにあるプロセスエリアよりも優先して改善するというものになります。これは、組織単位ではなくプロセスエリア単位で能力レベルを設定する、ISO/IEC 15504-2、3301 シリーズや CMMI の連続表現とは異なる方向性といえます。

Software CMM の開発停止と CMMI の開発

1993 年、小規模な改訂である Software CMM v1.1 がリリースされました。[7]これは、内容や意図自体の変更はされずに、主に語句や表現の明確化および一貫性の向上のための変更が加えられたものになります。

当時、新しいキープロセスエリアを追加するといった、より大規模な改訂への要求については、v2.0 リリースで取り込むという計画になっていました。Software CMM v2.0 の開発は 3 回の大きなレビュー実施をおし、ほぼ完成するところまで進みましたが、1997 年にスポンサーであった米国国防省の意向によって、CMMI 作成に注力するために中止されました。[2]

CMMI は当初、以下の成熟度モデルを統合するものとして開発されました。

- ハードウェア開発なども含めたシステム開発の成熟度モデル(EIA/IS-731)
- 米国国防省で導入が図られていた統合成果物プロセス開発 (integrated product and

process development、例えば国防省プロジェクトでハードウェア、ソフトウェア、運用など様々な専門家グループが参加して開発するとき、グループ間連携を強調する考え方)の成熟度モデル(Integrated Product Development CMM v0.98)

- Software CMM v2C

このように複数のモデルを統合した背景には、Software CMM にみられるような成熟度の枠組みがソフトウェア開発以外でも応用可能であったために、様々な領域で試みられていた CMM に類似した成熟度モデルの開発がありました。それらのモデルは、アーキテクチャ、内容、アプローチなどに差異があり、複数のモデルを適用した場合は改善策を広げていくことが難しく、トレーニング、アプレイザル、改善活動などを実施するときコストが高くつくようになっていました。[8]

この問題を解決するために、米国政府、産業界の代表、および SEI により CMMI プロジェクトが組織され、様々な CMM で扱っている内容を統合するフレームワークとして CMMI の作成が進められました。

CMMI は、他の分野の要素とともに、SW-CMM の主要な要素が取り入れられました。同時に、システム開発の成熟度モデルの統合などを行うなかで、ISO/IEC 15504-2 とも親和性のある連続表現も導入されました。CMMI は最初の正式版が 2000 年にリリースされ、産業界からのフィードバックの反映などをした CMMI V1.1 が、2002 年にリリースされています。

CMMI の適用範囲の拡大

CMMI V1.1 のリリース以降、CMMI のフレームワークは、他の対象領域に適用可能であることが明らかになってきたため、他領域への拡大が図られるようになりました。CMMI V1.2 では「取得のための CMMI(CMMI-AQC)」と「サービスのための CMMI(CMMI-SVC)」が計画され、既存の CMMI は名称を「開発のための

CMMI(CMMI-DEV)」に変更されました。[8]

その結果、2006 年には CMMI-DEV、2007 年には CMMI-AQC、2009 年には CMMI-SVC がそれぞれ V1.2 としてリリースされています。この時点から CMMI は正式に、ソフトウェアおよびシステムの開発だけでなく、サービス業全般 [9]や、各種委託開発における調達活動 [10]にも適用できるモデルという位置づけになりました。従って、CMMI の説明によれば、例えばサービス業のひとつであるコンサルティング会社のプロセス成熟度のレベル判定も、CMMI-SVC を使えば可能ということになります。

CMMI の開発はその後進められ、2010 年には CMMI-DEV、CMMI-AQC、CMMI-SVC のそれぞれで V1.3 がリリースされています。ここではプロセス領域の統合など、モデル構成要素に関しては大きな変更もいくつかなされていますが、成熟度レベルのコンセプト上の大きな変更はないといえます。

CMMI は V2.0 リリースでは構成要素に大きな変更が加えられています。しかしながら、本記事の冒頭でも紹介したように、V2.0 はリリース直後であるため、執筆時点では適用事例はまだ存在せず、詳細が明確ではない点も残っています。従って、V2.0 の変更点については、別の機会に確認できればと思います。

最後に

本記事は、2011 年以來の大きな改訂が CMMI で行われた機会に、CMMI のプロセス成熟度の概念と、そのコンセプトの作成経緯について紹介しました。

CMMI は、その前身まで遡ると 1980 年代に開発され、以降進化を続けているモデルになります。主にシステム開発およびソフトウェア開発で適用されることが多いモデルですが、定義上はサービス業全般を含む、様々な領域へ適用可能なものとされています。

ビジネスにおける継続的なプロセス改善は、業種を問

わず重要なテーマと考えられます。今回の CMMI 新バージョンのリリースを契機に、プロセス改善を進めるツールとして、CMMI に関心を持つ人が増えればと期待しています。

参考資料

1. Curtis, Bill; Hefley, William E.; Miller, Sally A. People Capability Maturity Model (P-CMMR) Version 2.0. Carnegie Mellon University, Software Engineering Institute, 2001.
2. Paulk, Mark. C. A History of the Capability Maturity Model for Software. ASQ Software Quality Professional, Vol.12, No.1, December 2009.
3. Crosby, Philip B. クオリティ・マネジメント:よい品質をタダで手に入れる法. (監訳)小林宏治. 日本能率協会, 1980年.
4. 渡辺昇、平松庸一. 経営品質メカニズムに関する理論モデルの研究－組織成熟度と組織変革の共進化プロセス－. 日本経営品質学会誌オンライン, 2007.
5. Humphrey, Watts S. Characterizing the Software Process: A Maturity Framework. Software Engineering Institute, Carnegie Mellon University, 1987.
6. Humphrey, Watts S. ソフトウェアプロセス成熟度の改善. (監訳)藤野喜一, 日本電気ソフトウェアプロセス研究会. 日科技連, 1991年.
7. Paulk, Mark C; Curtis, Bill; Chrissis, Mary Beth; Weber, Charles V. ソフトウェア能力成熟度モデル 1.1版. (訳)ソフトウェア技術者協会 CMM グループ, 1993年.
8. CMMI の変遷. (オンライン) (引用日:2018年6月29日).
<https://www.daiwa-computer.co.jp/jp/consulting/cmmi-evolution>.
9. CMMI Product Team. CMMI for Service, Version 1.3. Software Engineering Institute, Carnegie Mellon University, 2010.
10. CMMI Product Team. CMMI for Acquisition, Version 1.3. Software Engineering Institute, Carnegie Mellon University, 2010.

¹ Capability Maturity Model、CMM、CMMI は、Carnegie Mellon 大学によって米国特許商標庁に登録されています。

² SEI は、Carnegie Mellon 大学の商標です。

GSLetterNeo Vol.122

2018年9月20日発行

発行者 株式会社 SRA 先端技術研究所

編集者 土屋正人

バックナンバー <http://www.sra.co.jp/gsletter>

お問い合わせ gsneo@sra.co.jp



株式会社SRA

〒171-8513 東京都豊島区南池袋 2-32-8

夢を。



夢を。Yawaraka Innovation
やわらかいのべしょん