

技術報告書
1993年2月
CMU/SEI-93-TR-24
ESC-TR-93-177

ソフトウェア能力成熟度モデル 1.1版



Mark C. Paulk
Bill Curtis
Mary Beth Chrissis
Charles V. Weber

無制限の配布は著作権により制限される

ソフトウェアエンジニアリング研究所
カーネギーメロン大学
ペンシルベニア州、ピッツバーグ市 15213





Software Engineers Association

Marusho-bldg. 3-12 Yotsuya, Shinjuku-ku, Tokyo 106-0004, JAPAN

E-mail:sea@sea.or.jp, Voice:+81-3-3356-1077, FAX:+81-3-3356-1072

<http://www.iiijnet.or.jp/sea/>

Preface to the Japanese Edition

At the request of the Center for Information Systems Engineering, we have reviewed this official Japanese edition of Capability Maturity Model® for Software, Version 1.1, and believe it to be an accurate and faithful translation of the original English edition.

Sincerely,

Vice Chairperson
Software Engineers Association
Mitsuhiro Takahashi

Verifier
Tomoo Matsubara
IEEE Software Soapbox Editor
Cutter Consortium Faculty Member



Software Engineers Association

Marusho-bldg. 3-12 Yotsuya, Shinjuku-ku, Tokyo 106-0004, JAPAN

E-mail:sea@sea.or.jp, Voice:+81-3-3356-1077, FAX:+81-3-3356-1072

<http://www.ijnet.or.jp/sea/>

日本語版への序文

情報システムエンジニアリングセンターからの要請により、我々はこの公式日本語版ソフトウェア能力成熟度モデル[®] 1.1 版のレビューを実施した、そして、原典の正確かつ忠実な翻訳であると信ずる。

以上

[英語版原本に署名]

[英語版原本に署名]

SEA 常任幹事
高橋光裕

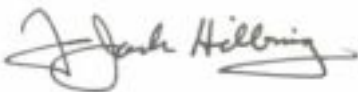
検証者
松原友夫
IEEE Software Soapbox 編集者

Permission to Reproduce and Distribute Copyrighted Materials

Permission to reproduce, in whole or in part, the volume of materials released by Carnegie Mellon University (CMU) under the title Capability Maturity Model® for Software, Version 1.1 is granted under the following conditions:

1. This letter must be reproduced with each copy.
2. All copies must include the copyright notice.
3. The materials are not used for commercial gain.
4. The materials are not to be distributed beyond your organization. Refer such questions to the Center for Information Systems Engineering (CISE).
5. The materials are to be used in a manner consistent with the framework and methodology advanced by the CISE.
6. NO WARRANTY. THIS MATERIAL FURNISHED BY CARNEGIE MELLON UNIVERSITY (CMU) AND THE CENTER FOR INFORMATION SYSTEMS ENGINEERING (CISE) IS FURNISHED ON AN "AS IS" BASIS. CMU MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CMU DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Sincerely,



F. Jack Hilbing
Director

著作物の複製と再配布の許諾

カーネギーメロン大学(CMU)が公開しているソフトウェア能力成熟度モデル® 1.1版の著作の一部または全部を複製することの許諾を、以下の条件の下に与える。

1. この文章が各々の複写と共に複製されなければならない。
2. 全ての複写は本著作権表示を含まなければならない。
3. 本文書の商用利用はしない。
4. 本文書は貴組織を越えての再配布はできない。そのような質問は情報システムエンジニアリングセンター(CISE)に照会せよ。
5. 本文書は、CISE が進める枠組みや方法論と首尾一貫した方法で使用する。
6. 無保証。この文書はカーネギーメロン大学(CMU)と情報システムエンジニアリングセンター(CISE)によって供給され、かつ “ そのまま ” を原則に供給されている。CMU は、用途または販売可能性への適合の保証、独占性あるいは本文書の利用から得られる結果を含む、限定されないいかなる事柄に関しても、明示または暗示されたいかなる種類の保証もしない。CMU は、特許、商標、または著作権侵害がないことに関するいかなる種類のいかなる保証も行わない。

以上

[英語版原本に署名]

F. Jack Hilbing
理事

日本語版に寄せて

真面目にソフトウェアの開発の改善を推進しようとしている組織や人にとって、または、すでに CMM(R)の概念に沿って非公式に成熟度評価を行ってきた組織にとって、かねてから期待されていた "Capability Maturity Model for Software(R), Version 1.1 (CMU/SEI-93-TR-24)" 及び "Key Practices of the Capability Maturity Model(R), Version 1.1 (CMU/SEI-93-TR-25)" の日本語への翻訳が、SEI 公認の公式日本語版として完成し、このたび出版の運びとなったことは、大きな朗報でしょう。

日本の品質意識の高い企業は、まだ、ソフトウェアプロセスという言葉が普及するずっと以前から、品質という側面から、それぞれの流儀で実質的にプロセス改善をやってきました。しかし、当時は、ハードウェアで成功した品質管理のやり方をもとにして、それを一部改良したものが多いようでした。

私は、かつて機械工場で工場挙げての改善に加わり、その後ソフトウェア組織に移り、ソフトウェア品質・生産性の改善に携わり、現在は、いろいろなソフトウェア組織を覗く機会があるコンサルタントという仕事をしています。これらの経験を通して、ソフトウェア組織には、一般に共通した典型的な現象が見られます。

例えば、「賽の河原」現象ともいうべき同じ問題の繰返しです。これは、開発に携わる人たちが、あるプロジェクトでどんなに苦労しても、喉元過ぎれば熱さを忘れて、次の、またはその後のプロジェクトでも同じ古典的なトラブルを繰返し、結局何年経っても状況は変化しない現象です。最も頻度の高い古典的な問題には、要求仕様がはっきりしないまま開発を開始して後で大量の仕様変更を要求される羽目になる、チームの編成が必要な数とスキルが充足しないままプロジェクトが走り出して後で混乱を招く、納期のプレッシャーからテスト不十分のままシステムを出荷してバグの修正に追まられる、外注に依存した部分の品質が悪くて作業の足を引っ張る、などが含まれるでしょう。

こうしたトラブルは、解決策がチームや組織の経験として学習されていないか、原因が分かっている場合でも、プロジェクトチームの権限を越える対策が必要なのに、それをあきらめていることが多いようです。この奥に潜む原因の一つは、組織の上から下まで物理的に実在する「物」を通してトラブルや原因の共通認識に立つことが容易な、従って物に注目すれば改善が容易な、有形物の開発プロセスの改善に慣れ親しんできたハードウェア指向の精神構造ではないかと私は感じています。

ハードウェアの世界では起こりにくい問題が、ソフトウェアではなぜ繰返し起こるかが、ソフトウェアプロセス改善の原点でなければなりません。ハードウェア指向の品質システムをソフトウェアに適用することの効果をすべて否定するわけではありませんが、それには限界があるばかりか、有形物流のプロセス改善でよしとする精神構造は、組織における人やチームの役割を軽視することで、しばしば害を及ぼします。

ソフトウェアのプロセス改善のためには、ハードウェアで成功したこと以上のことをやらねばならないのです。ソフトウェアプロセスの改善は、チームを構成する人が組織の仕組みの中で行う活動の改善ですから、組織全体としての取り組みが必要不可欠です。

こうしたことは、地道にソフトウェア品質の改善を行ってきた過程である程度分かっていたと思いますが、私を含めて、日本人は個々の組織でよいことをやっても、それを体系化して広く産業界に普及するのは苦手なので、改善は改善意識の高い組織の内部に留まっていて、それを産業全体のものにすることはできませんでした。

幸い、米国カーネギーメロン大学、ソフトウェア工学研究所の Watts Humphrey を始めとする優れたメンバーの努力によって、ハードウェア指向の品質管理とは異なる視点で、ソフトウェアの特性に着目したソフトウェアのためのプロセス改善への取り組み方法が CMM として体系化されました。その後、最初のモデルは、米国らしいオープンな議論からの定期的なフィードバックというかたちで、あるいは、プロセス成熟評価の産業への普及の過程で作られた膨大な補助資料や文献の充実というかたちで、整備された体系として成熟しました。

かつて私は、頻繁に日本のソフトウェア組織を訪ねたドイツの GMD の研究所長であった Dr. Gerhard Goos から、「日本のソフトウェア組織は、同じ企業の中でさえ、ソフトウェアプロセスや開発環境に、なぜ大きな格差があるのですか？また、程度の低い組織でも、レベルの低さに気づかずに、それを得々として説明するのはなぜでしょう？」と問われたことがあります。日本の組織の閉鎖性、開発者の非流動性、または、企業や組織間の技術交流が場はあっても実質的に行われていない、などがそのときの私の答えでした。しかし、これからは違ってくるでしょう。我々は、CMM という実質的に世界共通の客観的な尺度を、より利用しやすい形で持ったのですから。

私が、これを機会に心から期待するのは、これが「点取り」や「レッテル買い」のために使われるのではなく、現場での地道な改善に有効に利用されることです。

この文書が、真剣にソフトウェアプロセス改善に取り組んでいる、または取り組もうとしている日本の方々に勇気づけることができれば幸いです。

最後に、この CMM の翻訳とレビューという膨大な作業をボランティア - でやってくださった方々に、厚く感謝致します。

(R) Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office.

1999 年 1 月

フリーコンサルタント

松原友夫

日本語版作成の経緯

1990 年代に入り、日本国内のソフトウェア企業はソフトウェアプロセス改善に積極的に取り組むようになり、その過程で ISO-9001 取得がブーム化した。同じ頃、ソフトウェア技術者協会(SEA)のプロセス改善分科会 SPIN(Software Process Improvement Network) にて、長期的な視野に立ったソフトウェアプロセス改善には CMM のような多段階評価の方が適しているとの認識の元に、CMM の日本国内への普及が話題となった。その際、CMM のような外国生まれの技術を日本国内へ普及するにあたって我々は、CMM 普及のための第一歩として、CMM を開発した原著者等の意図通りに正確に理解することのできる日本語版のテキスト、つまり「日本人のための CMM 基本文書」の存在が不可欠と考えた。

日本語版の役割は、単に言語としての英語の得手不得手の壁を取り払うだけではない。文化や企業風土が異なる日米間では、一見等価に見える言葉でも、概念や意味内容、文脈の中での使われ方などに大きな違いがあるという場合も決して少なくない。プロセス評価手法である CMM について言えば、このような微妙な解釈のずれが誤評価などの重大な問題を引き起こさないとも限らない。このような問題を防止するためには、日本におけるソフトウェア開発の現状やプロセス改善について熟知している人々が、CMM の本質、つまり原著者等の意図を十分に理解し、原著を字面だけでなく意味的にも忠実に翻訳することが必要である。

これらの経緯から、SEA とカーネギーメロン大学(CMU)との間で翻訳出版契約を結び、CMU との間で合意された厳格な翻訳手順に従って作業が進められた。翻訳作業は、SEA-SPIN 有志による CMM 翻訳チームと検証チームとが、約 2 年間に渡っての翻訳と検証の作業を繰り返すことにより CMM の理解を深め、かつ翻訳精度を高めていった。その結果、フランス語版に続く第 2 番目の公式外国語版として CMU の正式な認定を得た CMM v1.1 公式日本語版として出版するに至った。

末筆ながら、本書の出版に当たり、数多くの方々からのご支援とご協力に感謝する。

本文書の翻訳チームメンバである坂本 啓司、塩谷 和範、田中 一夫、中村 淳、端山 毅、各氏の長期間にわたる翻訳作業への貢献に感謝する。また、数百ページにもわたる原文の初期翻訳に当たっては、高木 徳生、朝見 昇、新原直樹、田中 慎一郎、中川 明彦、前田 高明 各氏をはじめ、幾多の方々からご支援とご協

力を得た。さらに、検証作業においては、検証チームを代表する乗松 聡氏の CMM に対する深い知識と長期間に渡る検証作業への貢献に感謝する。特に、松原 友夫氏、岸田 孝一氏にはプロセス改善への深い見識のもとに文書の総合レビューを実施して頂き、完成に向けてご尽力いただいた。この他にも、SEA-SPIN メンバーを始め、テスト参加者など数多くの方々がレビューなどを通じてご助力をいただいたことに深く感謝する。

ソフトウェア技術者協会（SEA）
CMM グループ代表世話人
高橋光裕

-- メモ --

目次

謝辞	v
読者へ	vii
この文書の目的は何か?	viii
誰がこの文書を読むべきか?	viii
この文書はどのように編成されているか?	ix
他のCMM成果物は何か?	x
さらに多くの情報を受け取るにはどうするか?	xi
1 プロセス成熟度の枠組み	1
1.1 未熟なソフトウェア組織対成熟したソフトウェア組織	1
1.2 プロセス成熟度の基礎となる基本的なコンセプト	3
1.3 能力成熟度モデルの概観	4
2 ソフトウェアプロセス成熟度の5段階	7
2.1 成熟度レベルの行動的特徴	9
2.1.1 レベル1 - 初期レベル	10
2.1.2 レベル2 - 反復できるレベル	10
2.1.3 レベル3 - 定義されたレベル	11
2.1.4 レベル4 - 管理されたレベル	12
2.1.5 レベル5 - 最適化するレベル	13
2.2 成熟度レベルの理解	14
2.2.1 初期レベルの理解	15
2.2.2 反復できるレベルと定義されたレベルの理解	15
2.2.3 管理されたレベルと最適化するレベルの理解	16
2.3 ソフトウェアプロセスの可視性	19
2.4 プロセス能力および実績の予測	22
2.5 成熟度レベルの飛び越し	25
3 作業面から見たCMMの定義	27
3.1 成熟度レベルの内部構造	28
3.2 成熟度レベル	30
3.3 キープロセスエリア	30
3.4 コモンフィーチャ	37
3.5 キープラクティス	39
4 CMMを使う	43
4.1 ソフトウェアプロセスアセスメントとソフトウェア能力審査手法	44
4.2 ソフトウェアプロセスアセスメントとソフトウェア能力審査の違い	47
4.3 プロセス改善におけるCMMの他の用途	49

目次

5	CMMの将来の方向性	51
5.1	CMM が取り扱わないもの	51
5.2	当面の活動	51
5.3	長期的な活動	52
5.4	結論	53
6	参考文献	55
	付録A：キープロセスエリアのゴール	59
A.1	レベル2のキープロセスエリア： 反復できるレベル	59
A.2	レベル3のキープロセスエリア： 定義されたレベル	61
A.3	レベル4のキープロセスエリア： 管理されたレベル	62
A.4	レベル5のキープロセスエリア： 最適化するレベル	63

図の一覧

図2.1	ソフトウェアプロセス成熟度の5段階	8
図2.2	The Juran Trilogy Diagram: 品質計画,品質制御, および品質改善	17
図2.3	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	20
図2.4	成熟度レベル毎のプロセス能力	23
図3.1	CMMの構造	29
図3.2	成熟度レベル毎のキープロセスエリア	31
図3.3	CMM構造の編成: キープラクティスの例	40
図4.1	ソフトウェアプロセスアセスメントとソフトウェア 能力審査の共通ステップ	45

図の一覧

謝辞

ソフトウェア能力成熟度モデルの記述を最初に作り出したグループは、献身的に多くの時間を費やして、そのモデルと特徴を論じ、ついでそのモデルをCMM v1.0として文書化した。このグループのメンバーはMark Paulk、Bill Curtis、Mary Beth Chrissis、Edward Averill、Judy Bamberger、Tim Kasse、Mike Konrad、Jeff Perdue、Charlie Weber、およびJim Withey等である。

この文書は、SEIのソフトウェアプロセスプログラムの最初の責任者であるWatts Humphreyのビジョンに基づいている。この文書が完成するまでにはいくつもの草稿があった。Jim Withey、Mark PaulkおよびCynthia Wiseが1990年に初期の草稿を作り出し、Watts Humphreyがこの文書の第2稿を提供した。Mark Paulkはその後この文書を引き継ぎ、完成まで主導的役割を果たした。Mary Beth ChrissisとBill Curtisの協力を得て、Markはこの文書を1991年8月にCMM v1.0としてまとめた。Mark Paulkはその文書の改訂版、CMM v1.1を作り出した。それがこの技術報告書である。

さまざまな段階において、何人かの人々がこの文書で表明されているコンセプトに貢献した。例えば、Joe Besselman、Marilyn Bush、Anita Carleton、Marty Carlson、Betty Deimel、Suzie Garcia、Richard Kauffold、Steve Masters、Mary Merrill、Jim Over、George Pandelios、Jane Siegel、およびCharlie Weberらである。

我々はTodd Bowman、Dorothy Josephson、Debbie Punjack、Carolyn Tady、Marcia Theoret、Andy Tsounos、およびDavid Whiteらの管理面での支援に感謝する。Mary Beth Chrissis、Suzanne Couturiaux、およびBill Pollakらの編集面での支援に感謝する。American Institutes for ResearchのRenne Dutkowskiからは文書のデザインについて助言を頂いた。

謝辞

読者へ

1986年11月にソフトウェアエンジニアリング研究所（SEI）は、Mitre社の援助を受けて、組織的なソフトウェアプロセス改善に効果があるであろうプロセス成熟度の枠組みを開発し始めた。この取り組みはソフトウェア契約者の能力をアセスメントする手法を連邦政府に提供するという要請に応えて開始された。1987年9月に、SEIはプロセス成熟度の枠組みの概要[Humphrey87a]と成熟度質問票[Humphrey87b]を公表した。SEIは、ある組織のソフトウェアプロセスのどの分野に改善が必要であるかを特定するための簡単なツールとして成熟度質問票を供給するつもりであった。不幸にも成熟度質問票は、プロセス成熟度問題を見定めるための手段としてより、むしろ『モデル』と見なされることが多過ぎた。

ソフトウェアプロセス成熟度の枠組みと暫定版の成熟度質問票による4年間の経験を経て、SEIはソフトウェアプロセス成熟度の枠組みをソフトウェア能力成熟度モデル（CMM）に発展させた[Paulk91、Weber91]。CMMはソフトウェアプロセスアセスメントで得られた知識と、産業界と政府両方からの広範囲のフィードバックに基づいている。成熟度の枠組みを詳細化することによって、プロセス改善プログラムを確立するためのより効果的なガイダンスを組織に対して提供するモデルが生まれた。

最初のリリースであるCMM v1.0は1991年から1992年にかけてソフトウェアコミュニティによってレビューされ使われた。CMM v1.0のワークショップが1992年4月に開催され、およそ200人のソフトウェア専門家が出席した。このCMM v1.1はそのワークショップからのフィードバックと、その後のソフトウェアコミュニティからのフィードバックの結果である。

CMMは、成熟度質問票を含むソフトウェアプロセス改善に有用なツールのセットを系統的に構築するための基盤である。忘れてはならない本質的なポイントは、質問票ではなくモデルがソフトウェアプロセス改善の基盤であることである。この文書は読者にCMM v1.1を紹介することを意図したものである。

この文書の目的は何か？

この文書はソフトウェアの能力成熟度モデルの技術的な概観を提供しており、バージョン1.1を反映している。特にこの文書は、5つのレベルを持つプロセス成熟度の枠組み、CMMを構成する構造的コンポーネント、CMMを実際に使う方法、そしてCMMの未来の方向性について説明している。この文書は、CMMを理解するための最良な情報源のひとつとなり、また、SEIが提唱しているソフトウェアプロセス成熟度に対するいくつかの誤解を解消するだろう。

SEIはモデルを洗練し拡大するために産業界および政府と共に働いてきた。ソフトウェア組織は、成熟度質問票に対してより、CMMに焦点を合わせることを推奨されている。SEIは、この焦点を奨励するためのプロセス成果物を開発してきたし、今も開発し続けている。CMM v1.1は、この文書[Paulk93a]と、『能力成熟度モデルのキープラクティス1.1版』[Paulk93b]で構成されている。『能力成熟度モデルのキープラクティス1.1版』は、CMMの各レベルのキープラクティスを説明している。この文書は、ソフトウェアプロセス成熟度の基盤となっている原理を説明しており、ソフトウェアプロセス成熟度を改善するための指針としてCMM v1.1をソフトウェア組織が使用することを支援する目的で書かれている。

誰がこの文書を読むべきか？

この文書は、CMMとそれに関連する成果物の入門的解説を提供する。したがって、CMMについて学ぶことに興味を持っている人は誰でもこの文書を読むべきである。しかしながらこの文書は、読者がソフトウェアの開発や保守についての若干の知識と経験、およびソフトウェアコミュニティが今日直面する問題についての理解を持っていると想定している。

このドキュメントは以下のような用途で使うことができる：

- ソフトウェアを開発または保守するための効果的なプロセスの一部であるキープラクティスを理解したい人によって
- CMMで次の成熟度レベルを達成するために必要とされるキープラクティスを確認したい人によって
- 効果的にソフトウェアを開発する能力を理解し改善したい組織によって
- 特定のソフトウェア組織と契約して作業を実施させる際のリスクを知りたい調達組織または外部発注元によって
- 成熟度質問票の質問を作るための基盤としてSEIによって
- ソフトウェアプロセスアセスメントあるいはソフトウェア能力審査を行うチームの準備をするインストラクタによって

この文書はどのように編成されているか？

この文書には以下の5つの章がある：

第1章

CMMを理解しその背後にある動機づけと目的を理解するために必要なコンセプトを定義する。

第2章	CMMの5つのレベルとそれらの基盤となる原理を説明する。
第3章	CMMがどのようにキープロセスエリアとして構造化され、コモンフィーチャによって構成され、キープラクティスの言葉で説明されているかを述べる。
第4章	CMMがソフトウェアプロセスアセスメント、ソフトウェア能力審査、およびプロセス改善プログラムのためのガイダンスを如何に提供しているかについてハイレベルの概観を述べる。
第5章	CMMとその関連成果物の将来の方向について述べて結論とする。

他のCMM成果物は何か？

この文書は単独で読むこともできるが、他の成果物への出発点となるように作られている。この文書と関連成果物は、読者がCMMを理解し使用するのを助けるためのものである。すべてのCMMベースの成果物は、モデルから系統的に引き出されたものであり、また今後も引き出されていくものである。この文書が書かれた時点においては、これら成果物の大部分が暫定版の試行的なテストとリリースというさまざまな段階にあり、最終形式で利用可能な状態ではない。

CMMベースの成果物セットにはいくつかの診断ツールが含まれており、それらは組織におけるソフトウェアプロセスの強み、弱み、およびリスクを明らかにするために、ソフトウェアプロセスアセスメント¹とソフトウェア能力審査²を行うチームによって使われる。恐らくこれら診断ツールの中で最も良く知られているのは成熟度質問票であろう。ソフトウェアプロセスアセスメントとソフトウェア能力審査の手法とトレーニングは同じくCMMに基づいている。

これらの成果物のユーザは、自分たちのソフトウェアプロセスの成熟度を改善することに熱心なコミュニティを形成している。SEIは、モデルとその関連成果物の質を高めるためにソフトウェアコミュニティと協働するであろう。

さらに多くの情報を受け取るにはどうするか？

CMMのトレーニングや、ソフトウェアプロセスアセスメントとソフトウェア能力審査をどのように実施するかを含めて、CMMとその関連成果物に関する情報を得るための連絡先は以下の通りである。

SEI Customer Relations
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
(412) 268-5800
Internet: customer-relations@sei.cmu.edu

¹ ソフトウェアプロセスアセスメントは、トレーニングを受けたソフトウェア専門家チームが行う査定であり、組織のソフトウェアプロセスの現状を判断し、組織が直面する優先順位の高いソフトウェアプロセス関連課題を判断し、そしてソフトウェアプロセス改善に対する組織的支援を獲得する。

² ソフトウェア能力審査は、トレーニングを受けた専門家チームが行う査定であり、ソフトウェア作業の実施に対して適格な契約者を特定したり、あるいは実施中のソフトウェア開発に使われているソフトウェアプロセスの状態をモニターする。

この文書や『能力成熟度モデルのキープラクティス1.1版』のような、SEIの技術報告書はDefense Technical Information Center(DTIC)、National Technical Information Service (NTIS) およびResearch Access Inc. (RAI) から直接入手可能である。これらの文書を入手するための連絡先は以下の通り：

RAI: Research Access Inc.
 3400 Forbes Avenue
 Suite 302
 Pittsburgh, PA 15213
 Telephone: (800) 685-6510
 FAX: (412) 682-6530

NTIS: National Technical Information Service
 U.S. Department of Commerce
 Springfield, VA 22161-2103
 Telephone: (703) 487-4600

DTIC: Defense Technical Information Center
 ATTN: FDRA Cameron Station
 Alexandria, VA 22304-6145
 Telephone: (703) 274-7633

SEIの技術報告書はインターネットでも入手可能である。インターネット上のUNIXマシンからanonymous FTPを使う方法は以下の通り：

```
ftp ftp.sei.cmu.edu3
login: anonymous
password: <your user id or any string>
cd pub/cmm
get READ.ME
get <files>
quit
```

ファイルREAD.MEは、どのようなファイルが入手可能であるかについての情報を含んでいる。他のSEI出版物も類似の方法で利用可能である。

3 SEIのftpマシンのアドレスは128.237.2.179である。

1 プロセス成熟度の枠組み

新しいソフトウェアの方法論と技術を適用することで生産性と品質を向上するという果たされぬ約束に20年を費やした後、産業界と政府組織は、基本的な問題がソフトウェアプロセスを管理する能力の不足であることを実感しつつある[DoD87]。規律の無い混沌としたプロジェクトの大混乱の中では、良い手法とツールがあってもその利益が実感されることはない。多くの組織で、プロジェクトはしばしば過度に遅れ、計画された予算の2倍もが費やされる[Siegel90]。このような事例では、プロジェクトがこれらの問題を避けるために必要なインフラと支援を組織が提供していない場合が多い。

しかしながら規律の無い組織であっても、個別のソフトウェアプロジェクトでは優秀な結果を出すこともある。このようなプロジェクトが成功するとき、一般に、それは献身的なチームの英雄的な努力によるものであり、成熟したソフトウェアプロセスを用いて組織が証明済みの手法を反復しているわけではない。組織全体にわたるソフトウェアプロセスが無い場合、結果を反復できるか否かは、次のプロジェクトに同じ個人を割り当てることができるか否かに完全に依存している。もっぱら特定の個人が割り当てられるか否かにかかっている成功は、組織にわたる長期的な生産性と品質の改善に対する基盤となるものではない。継続的な改善は、ソフトウェアのエンジニアリングと管理の効果的なプラクティスのために、プロセスインフラを構築する努力に焦点を当て続けることによってのみ実現できるものである。

1.1 未熟なソフトウェア組織対成熟したソフトウェア組織

プロセス改善について道理にかなった目標を設定するためには、未熟なソフトウェア組織と成熟したソフトウェア組織の間の相違について理解する必要がある。一般に、未熟なソフトウェア組織では、ソフトウェアプロセスがプロジェクト進行中に実務者と彼らのマネージャによって即席で作られる。たとえソフトウェアプロセスが明記されていたとしても、厳格には従わず、徹底もされない。未熟なソフトウェア組織は受け身的であり、マネージャはいつも目の前の危機を解決すること（消火活動としてよく知られている）に集中している。現実的な見積もりに基づいていないためにスケジュールと予算はいつも超過して

プロセス成熟度の枠組み

しまう。厳しい期限が課せられると、スケジュールを合わせるために製品の機能充足性と品質がしばしば妥協されることになる。

未熟な組織では、成果物品質を判断するためや、成果物またはプロセスの問題を解くための客観的基準がない。そのため、成果物品質を予測することは困難である。プロジェクトが予定より遅れてくるとレビューやテストのような品質を高めるための活動がしばしば削減されるかまたは省略される。

一方、成熟したソフトウェア組織では、ソフトウェアの開発と保守のプロセスを管理するための能力を組織全体で持っている。ソフトウェアプロセスは既存要員と新たに加わった者の両方に正確に伝達され、作業活動は計画されたプロセスに従って実行される。遵守すべきプロセスは使用に適しており

[Humphrey91b]、作業が実際に実施される方法と首尾一貫している。これらの定義されたプロセスは必要に応じて更新され、そして制御されたパイロットテストと費用対効果分析のいずれかまたは両方を通して改善策が構築される。定義されたプロセスでは、役割と責任はプロジェクト中でまた組織にわたって明確である。

成熟した組織では、マネージャがソフトウェア成果物の品質と顧客満足度をモニターする。成果物品質を判断するためや、成果物とプロセスの問題を分析するための客観的かつ定量的な基準がある。スケジュールと予算は過去の実績に基づいており、現実的なものであり、コスト、スケジュール、機能充足性と成果物品質の期待される成果は通常達成される。一般に、関係者のすべてがそうすることの価値を理解しているので、規律あるプロセスに首尾一貫して従っている。そしてプロセスを支援するために必要なインフラが存在している。

未熟な、そして成熟したソフトウェア組織についてのこれらの観測結果を利用するためには、ソフトウェアプロセス成熟度の枠組みを構築する必要がある。この枠組みは場当たりの混沌としたプロセスから成熟した規律あるソフトウェアプロセスへの進化の道筋を説明している。この枠組みが無いと、継続した改善を支援するために必要な基盤が確立されないため、改善プログラムは効果的でないことが明らかになることがある。ソフトウェアプロセス成熟度の枠組みはソフトウェアプロセス、ソフトウェアプロセス能力、ソフトウェアプロセス実績、ソフトウェアプロセス成熟度といった次節で定義されるコンセプトを統合することによって生まれる。

1.2 プロセス成熟度の基礎となる基本的なコンセプト

Webster辞典によると、『プロセス』とは『目的または結果を達成するための一連の処置、変化、あるいは機能といった、何かを作り出すための作用系』である。IEEEはプロセスを『与えられた目的のために実施される一連のステップ』[IEEE-STD-610]と定義している。『ソフトウェアプロセス』は、ソフトウェアおよび関連成果物（例えば、プロジェクト計画、設計文書、コード、テストケース、およびユーザマニュアル）の開発と保守に使用する活動、手法、プラクティス、および変換作業の集合として定義することができる。組織が成熟するにつれて、ソフトウェアプロセスはより良く定義され、組織を通じて一層首尾一貫して実施されるようになる。

『ソフトウェアプロセス能力』は、あるソフトウェアプロセスに従うことによって達成できると期待される結果の範囲を表している。組織のソフトウェアプロセス能力は、その組織が取り掛かる次のソフトウェアプロジェクトで予想される最も可能性が高い成果を予測するひとつの手段を提供する。

『ソフトウェアプロセス実績』は、ソフトウェアプロセスに従うことによって達成された実際の結果を表わす。したがって、ソフトウェアプロセス実績は、達成された結果に焦点を合わせており、ソフトウェアプロセス能力は、期待される結果に焦点を合わせている。ある特定のプロジェクトの実績は、そのプロジェクトの特質や実施される環境によるので、必ずしもその組織のプロセス能力の全てを反映しないだろう。すなわちプロジェクトの能力はその環境によって制約を受ける。例えば、引き受けたアプリケーションや技術に急激な変更が

プロセス成熟度の枠組み

あると、プロジェクト要員は学習過程に置かれるため、プロジェクトの能力、および実績が、組織のプロセス能力に達しないことがある。

『ソフトウェアプロセス成熟度』は、特定のプロセスが明示的に定義され、管理され、計測され、制御され、そして効果的であることの程度を示している。成熟度は能力向上の可能性を意味し、組織のソフトウェアプロセスの豊かさと、組織全体を通じてプロジェクトで適用される一貫性の両方を示している。成熟した組織では、通常文書化とトレーニングを通して、そのソフトウェアプロセスが組織全体でよく理解されており、そしてプロセスは絶えずそのユーザによってモニターされて、改善されている。成熟したソフトウェアプロセスの能力は既知である。ソフトウェアプロセス成熟度が意味することは、組織のソフトウェアプロセスの結果として生じる生産性と品質は、そのソフトウェアプロセスを使用することによって達成される規律の首尾一貫した増大を通して、時間をかけて改善され得ることである。

ソフトウェア組織は、ソフトウェアプロセス成熟度が向上するにつれて、方針、標準および組織構造という形でそのソフトウェアプロセスを制度化する。『制度化』には、ビジネスの手法、プラクティス、手順を支援するインフラと企業文化の構築が必然的に伴い、最初にそれらを定義した人たちが去った後にも持続するようになる。

1.3 能力成熟度モデルの概観

しばしばソフトウェア技術者とマネージャが非常に詳細に問題を知っているにもかかわらず、どの改善活動が最も重要であるかについて彼らの意見が合わないことがある。改善のための系統だった戦略無しには、最初に何の改善活動に着手するべきかについて、経営層と専門要員の間にコンセンサスを得ることは難しい。プロセス改善努力で永続的な結果を達成するためには、組織のソフトウェアプロセス成熟度を段階的に高めるような進化の道筋を設計する必要がある。ソフトウェアプロセス成熟度の枠組み[Humphrey87a]では、それぞれの段階における改善が次の段階で着手するべき改善の基盤を形成するように段階を並べている。したがって、ソフトウェアプロセス成熟度の枠組みから引き出された改善戦略は、継続的なプロセス改善へのロードマップを提供する。それは

進歩への指針として、組織の欠陥を明らかにするものであり、トラブルを抱えたプロジェクトに応急処置を提供するように意図されたものではない。

ソフトウェア能力成熟度モデルは、いかにしてソフトウェアの開発と保守のプロセスを制御するか、また、いかにしてソフトウェアの優れたエンジニアリングと管理の文化へと高めるかについて、ソフトウェア組織に対してガイダンスを提供する。CMMIは、ソフトウェア組織が現在のプロセス成熟度を決定し、ソフトウェアの品質とプロセスの改善に最も重大ないくつかの課題を特定することによって、ソフトウェア組織がプロセス改善戦略を選択することを導くよう作られている。組織は、限定された活動だけに集中し、その完遂に向けて積極的に取り組むことによって、組織全体のソフトウェアプロセスを着実に改善し、ソフトウェアプロセス能力を継続的かつ永続的に増大させることができる。

CMMの段階構造は過去60年間にわたって存在した成果物品質の原理に基づいている。1930年代、Walter Shewartは統計的品質管理の原理を広めた。彼の原理はさらに発展させられ、W. Edwards Deming[Deming86]とJoseph Juran[Juran88、Juran89]の業績の中で有用性が実証された。これらの原理は、ソフトウェアプロセスを定量的に制御するために、プロジェクトの管理とエンジニアリングの基盤を確立する成熟度の枠組みとしてSEIによって取り入れられ、継続的なプロセス改善のための基盤となった。

品質原理を取り入れた成熟度の枠組みはPhilip Crosbyが書いた『*Quality is Free*』という本[Crosby79]によって最初に示唆された。Crosbyの品質管理成熟度格子では、品質プラクティスを採用する際の5つの進化段階を記述している。この成熟度の枠組みは、IBMにいたWatts Humphreyの指導の下でRon Radiceと彼の同僚によってソフトウェアプロセスに適用された[Radice85]。Humphreyは1986年にこの成熟度の枠組みをSEIにもたらし、成熟度レベルのコンセプトを加えて、ソフトウェア産業界全体が現在使用している基盤を開発した。

プロセス成熟度の枠組み

Humphreyによる成熟度の枠組みの初期バージョンは、SEIの技術報告書 [Humphrey87a、Humphrey87b]、論文[Humphrey88]、および『ソフトウェアプロセス成熟度の改善』[Humphrey89]という本のなかで記述されている。暫定版の成熟度質問票[Humphrey87b]は、組織のソフトウェアプロセスの成熟度を描写するためのツールとして1987年に公表された。1987年にソフトウェアプロセスアセスメントとソフトウェア能力審査というふたつの手法がソフトウェアプロセス成熟度を評価するために開発された。1990年から、SEIは政府と産業界の多くの人々の助けを借り、ソフトウェアプロセス改善への数年間にわたる適用の経験に基づいてモデルをさらに拡大し洗練してきた。

2 ソフトウェアプロセス成熟度の5段階

継続的なプロセス改善は、革命的な刷新というよりは、むしろ数多くの小さな進化ステップを基盤としている[Imai86]。CMMは、5つの成熟度レベルとして進化ステップを体系付けた枠組みを規定しており、継続的なプロセス改善のための連続的な基盤を築いている。これら5つの成熟度レベルは、組織におけるソフトウェアプロセスの成熟度を計測し、その能力を評価するための尺度を定義している。これらのレベルは、組織における改善努力の優先順位付けにも役立つ。

『成熟度レベル』は、成熟したソフトウェアプロセスを達成する途上の整った形で定義された進化の段階である。各成熟度レベルはそれぞれ、継続的なプロセス改善の基盤における層を規定している。それぞれのレベルは、プロセスゴールの集合で構成される。これらのゴールが達成されれば、ソフトウェアプロセスの重要なコンポーネントが安定する。成熟度の枠組みの各レベルを達成することで、ソフトウェアプロセスの各コンポーネントを確立し、結果として組織のプロセス能力が増大する。

CMMは図2.1のように5つのレベルで編成され、ソフトウェアプロセスをより成熟させるための改善処置が優先順位付けされている。図2.1の矢印は、成熟度の枠組みの各段階で、組織によって制度化されるプロセス能力のタイプを示している。

ソフトウェアプロセス成熟度の5段階

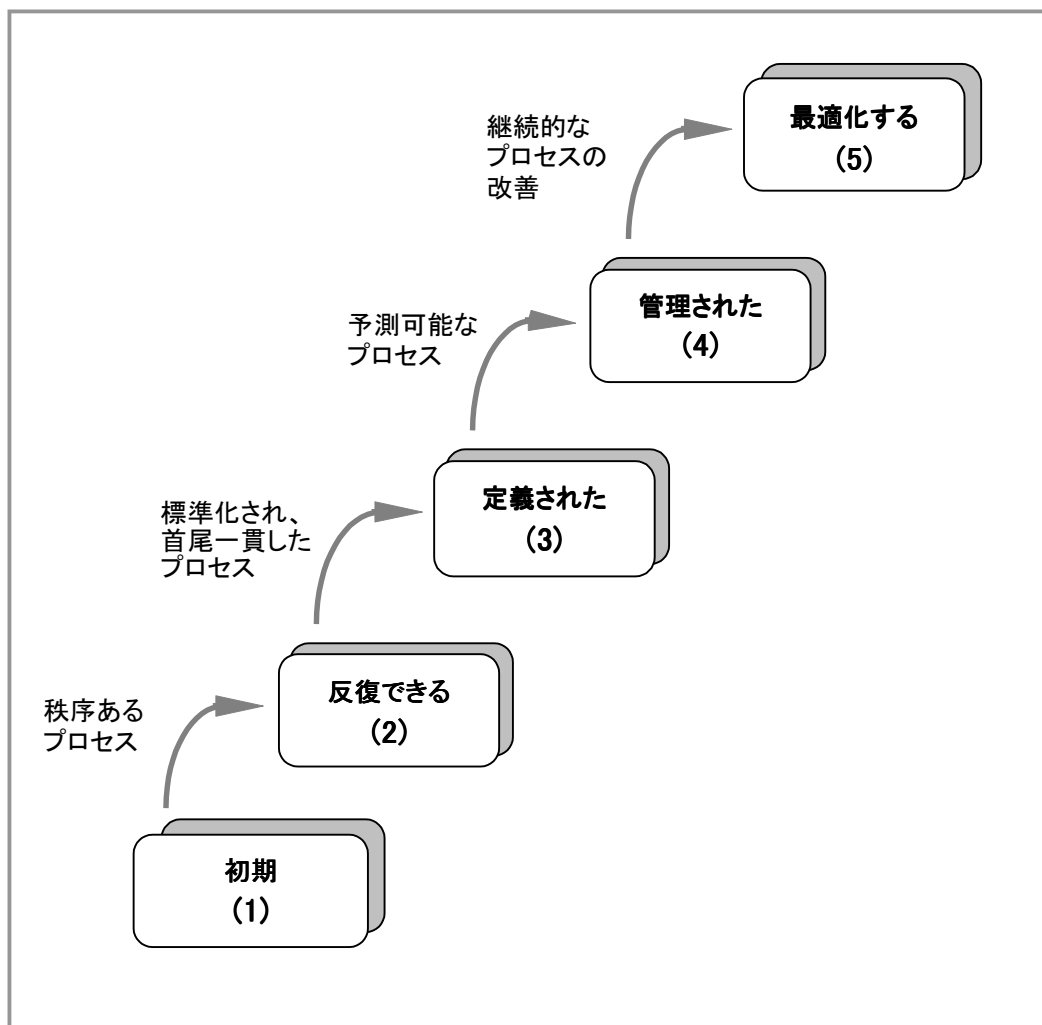


図2.1 ソフトウェアプロセス成熟度の5段階

成熟度レベル5段階の各レベルにおける主なプロセス変更の特徴を以下に示す：

- 1) 『初期』 ソフトウェアプロセスは場当たりの、時には混沌としたものと特徴付けられる。ほとんどのプロセスは定義されておらず、成功は個人の努力に依存する。

- 2) 『反復できる』コスト、スケジュール、機能充足性を確認するために、基本的なプロジェクト管理プロセスが確立されている。同様のアプリケーションのプロジェクトに関しては、以前の成功経験を反復するためのプロセス規律がある。
- 3) 『定義された』管理およびエンジニアリングの活動に対するソフトウェアプロセスが、「組織の標準ソフトウェアプロセス」として文書化、標準化、そして統合化されている。ソフトウェアの開発と保守において、承認されテラーリングされたバージョンの「組織の標準ソフトウェアプロセス」をすべてのプロジェクトが使用する。
- 4) 『管理された』ソフトウェアプロセスおよび成果物品質に関する詳細な計測結果が収集されている。ソフトウェアプロセスも成果物も、定量的に理解され制御される。
- 5) 『最適化する』革新的なアイデアや技術の試行、およびプロセスからの定量的フィードバックによって、継続的なプロセス改善が可能になっている。

2.1 成熟度レベルの行動的特徴

成熟度レベル2から5は、ソフトウェアプロセスを確立し改善するために組織によって実施される活動、各プロジェクトによって実施される活動、およびプロジェクト横断的に結果として生じるプロセス能力によって特徴付けることができる。レベル1の行動的特徴付けは、上位の成熟度レベルでのプロセス改善と比較する基盤を確立するために含まれている。

2.1.1 レベル1 - 初期レベル

初期レベルでは、組織がソフトウェアの開発と保守のために安定した環境を提供しないのが典型的である。組織が健全な管理を欠いている場合、役に立たない計画活動とリアクション型の責任体制によって、良いソフトウェアエンジニアリングプラクティスの利点が損なわれてしまう。

危機に直面すると、典型的なプロジェクトは計画された手順を放棄し、コーディングとテストに戻る。プロジェクトの成功は、卓越したマネージャと熟練し効果的なソフトウェアチームがいるかどうかにかかっている。有能で力のあるソフトウェアマネージャは、ソフトウェアプロセスの手抜きをしようという圧力に打ち勝てることもある。しかし、彼らがプロジェクトを去ったとき、これらの安定的影響力は彼らとともに消え去ってしまう。たとえ強固なエンジニアリングプロセスであっても、健全な管理なくしては不安定性を克服することは不可能である。

レベル1の組織のソフトウェアプロセス能力は予測不能である。なぜならソフトウェアプロセスが、作業の進捗につれて、たえず変更され、修正され続けるからである（プロセスは場当たりのものである）。この場合、スケジュール、予算、機能充足性および成果物の品質は一般的に予測できない。実績は、個人の能力に依存しており、各個人の先天的なスキル、知識、および動機によって変動する。安定したソフトウェアプロセスの実績はほとんど無く、実績は組織の能力よりもむしろ個人の能力によって予測される。

2.1.2 レベル2 - 反復できるレベル

反復できるレベルでは、ソフトウェアプロジェクト管理の方針とその方針を履行するための手順が確立されている。新しいプロジェクトの計画とその管理は、類似プロジェクトの経験に基づいている。レベル2を達成することの目的は、ソフトウェアプロジェクトの効果的な管理プロセスを制度化することである。具体的なプロセス実装はプロジェクト毎に相違があるとしても、効果的な管理

プロセスは、以前のプロジェクトで成功した実践を組織が反復できるようにする。効果的なプロセスは、実践され、文書化され、徹底され、トレーニングされ、計測され、改善可能なものと特徴付けられる。

レベル2組織のプロジェクトでは、基本的なソフトウェア管理を導入している。現実的なプロジェクトコミットメントは、以前のプロジェクトの観察結果と、現在のプロジェクトの要件が基盤になっている。プロジェクトのソフトウェアマネージャは、コスト、スケジュール、および機能充足性を確認し、コミットメントを果たす上での問題は、発生時点で明らかにされる。ソフトウェア要件およびその要件を満足させる作業成果物にはベースラインがあり、その一貫性は制御されている。ソフトウェアプロジェクト標準が定義され、組織はこれらに忠実に従うことを確実なものにする。強い顧客供給者関係を確立するために、ソフトウェアプロジェクトは、外注先がある場合は外注先とともに進められる。

レベル2組織のソフトウェアプロセス能力は、規律ある状態と要約できる。ソフトウェアプロジェクトの計画と進捗確認は安定しており、過去の成功事例は反復することが可能である。プロジェクトのプロセスは、プロジェクト管理体制の効果的な制御のもとにあり、以前のプロジェクトの実績に基づいた現実的な計画に従っている。

2.1.3 レベル3 - 定義されたレベル

定義されたレベルでは、組織全体でのソフトウェアの開発と保守の標準プロセスが文書化されている。これにはソフトウェアのエンジニアリングと管理の両方のプロセスが含まれ、これらのプロセスは首尾一貫したものとして統合化される。この標準プロセスは、CMMでは「組織の標準ソフトウェアプロセス」と表されている。ソフトウェアマネージャと技術要員は、より効果的に動けるよう、レベル3で確立されたプロセスを利用する（そして適宜変更する）。組織はそのソフトウェアプロセスを標準化する際には、ソフトウェアエンジニアリングの効果的なプラクティスを活用する。組織のソフトウェアプロセス活動の責任を負っているグループ（ソフトウェアエンジニアリングプロセスグループ（SEPG）などと呼ばれる）が存在する[Fowler90]。要員およびマネージャが要求される知識や技能を習得し割当てられた役割を遂行できるように、全組織的なトレーニングプログラムが履行されている。

ソフトウェアプロセス成熟度の5段階

プロジェクトは、「組織の標準ソフトウェアプロセス」をテーラリングし、プロジェクトの特徴に合わせて、独自の定義されたソフトウェアプロセスを作成する。CMMでは、このテーラリングされたプロセスを「プロジェクトの定義されたソフトウェアプロセス」と呼んでいる。定義されたソフトウェアプロセスは、整った形で定義されたソフトウェアのエンジニアリングと管理のプロセスが一体となって統合された集合から成っている。整った形で定義されたプロセスは、開始基準、インプット、作業を実施するための標準と手順、（ピアレビューのような）検証メカニズム、アウトプット、および完了基準を含んでいる。ソフトウェアプロセスが整った形で定義されているので、管理層はすべてのプロジェクトの技術的進捗をよく見通せる。

レベル3組織のソフトウェアプロセス能力は、標準と首尾一貫性と要約できる。ソフトウェアのエンジニアリングと管理の両方の活動が安定し反復できる。確立された製品ラインにおいて、コスト、スケジュール、機能充足性は制御下にあり、ソフトウェア品質は確認される。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活動、責任、および役割に対する組織全体の共通の理解を基盤としている。

2.1.4 レベル4 - 管理されたレベル

管理されたレベルでは、組織は、ソフトウェアの成果物とプロセスの両方に対して定量的品質目標を設定している。生産性および品質は、組織的計測プログラムの一部として、重要なソフトウェアプロセス活動について全プロジェクト横断的に計測される。「組織のソフトウェアプロセスデータベース」は、「プロジェクトの定義されたソフトウェアプロセス」から入手可能なデータを収集し、分析するために使われる。レベル4では、ソフトウェアプロセスは、整った形で定義され首尾一貫した計測を備えるようになる。これらの計測は、プロジェクトのソフトウェアプロセスと成果物を評価するための定量的基盤になる。

プロジェクトは、その成果物とプロセスに対する制御を達成し、許容可能な定量的境界内に収めるためにプロセス実績の変動幅を小さくする。プロセス実績の意味ある変動を、（特に確立した製品ラインでは）ランダムな変動（ノイズ）と区別することが可能になる。新しいアプリケーション領域に対する学習過程に伴うリスクは既知のものになり、注意深く管理される。

レベル4組織のソフトウェアプロセス能力は、予測可能と要約できる。プロセスが計測され、計測可能な限界内で遂行される。このレベルのプロセス能力では、このような限界の定量的範囲内で、プロセスと成果物の品質がどんな傾向かを組織が予測できるようになる。これらの限界を越えたとき、この状況を正すための処置がとられる。ソフトウェア成果物は予測可能な高品質なものである。

2.1.5 レベル5 - 最適化するレベル

最適化するレベルでは、組織全体が継続的なプロセス改善を重視している。組織には、欠陥の発生を予防することを目標として、プロセスの弱みと強みを先を見越して特定する手段が備わっている。ソフトウェアプロセスの有効性を示すデータは、組織のソフトウェアプロセスに対する変更提案や新しい技術の費用対効果分析に使われる。最高のソフトウェアエンジニアリングプラクティスによる革新的な改善が特定され、組織全体に移転される。

レベル5の組織のソフトウェアプロジェクトチームは、欠陥原因を決定するために欠陥の分析を行う。ソフトウェアプロセスは、既知の欠陥の再発を防止するために評価され、その教訓は他のプロジェクトにも広められる。

レベル5組織のソフトウェアプロセス能力は、継続的な改善と特徴付けられる。レベル5の組織では、プロセス能力の範囲を改善する努力を怠らず、これによってプロジェクトの実績を向上させる。既存プロセスの漸進的進歩と、新しい技術および手法による革新の両方によって改善が行われる。

2.2 成熟度レベルの理解

CMMIは、個々の成熟度レベルにおける組織を特徴付ける重要な（もしくはキーとなる）属性について記述しており、その意味で記述的モデルである。政府との大規模契約プロジェクトを遂行する組織に期待される規範的な行動について、詳細なプラクティスが特徴付けているという意味で、規範的モデルである。CMMIは適度なレベルの抽象概念で書かれており、ある組織がソフトウェアプロセスをどのように実装するかについて過度に制約することを意図するものではない。すなわち、ソフトウェアプロセスに本質的な特徴は通常どうあると期待されているかを記述しているに過ぎない。

CMMIを適用するいかなる状況においても、キープラクティスは合理的に解釈されるべきである。もしもビジネス環境が、大規模組織のものと比べて大幅に相違がある場合、知識のある専門家の判定のもとで、CMMIは適切に解釈されなければならない。

CMMIは処方箋ではないので、組織に対して改善の方法を教えたりはしない。CMMIはそれぞれの成熟度レベルに達する特定の手法を処方するのではなく、各々の成熟度レベルにおける組織について記述している。レベル1からレベル2までいくのに、数年を要することがあり、他のレベルに移行するには通常、2年ほどかかる。

ソフトウェアプロセス改善は、組織の戦略的計画および事業目標、その組織的構造、その使用技術、その社会的文化、およびその管理体制などの周辺状況のなかで発生する。CMMIはTQM運動のプロセス面に焦点を置いている。優れたプロセス改善は、ソフトウェアプロセスの範囲外の観点への対応も伴う（例えば、プロセス改善が実装され制度化されることを可能にするための組織文化の改変に伴う人的問題など）。

2.2.1 初期レベルの理解

レベル1の組織は、場当たりの混沌としたプロセスを持つとしばしば特徴付けられるが、たとえ予算やスケジュールが超過するにしても、彼らは、しばしば稼動可能な成果物を開発する。レベル1の組織における成功は、組織内の人々の能力と英雄的な活躍に依存する。有能な人々を選抜し、雇用し、育成し、かつ／または保持することは、すべての成熟のレベルにおいて共通の重要な課題である。ただし、多くの場合それらはCMMの範疇ではない。

2.2.2 反復できるレベルと定義されたレベルの理解

プロジェクトが拡大し複雑化するにつれて、争点は技術的課題から、組織的、管理的課題---プロセス成熟度の焦点[Siegel90,DoD87,GA0-92-48]---にシフトする。プロセスは人々が効果的に作業できるようにするものであり、最高の要員の教訓を文書化されたプロセスとして組み込み、(通常トレーニングを経て)これらのプロセスを効果的に実施するのに必要なスキルを磨き、そして作業を実施している人々から学んで継続的に改善することにより、一層効果的に作業できるようになる。

レベル2を達成するには、管理層は、規律あるソフトウェアプロセスを達成するために、自らのプロセスに焦点を当てなければならない。レベル2はレベル3の基盤を提供する。なぜなら、レベル3における技術的、組織的課題に取り組む前に、プロセスを改善するために対処する管理に焦点を当てているからである。管理層は、プロジェクト管理プロセスを文書化し、それに従うことによって、レベル2達成のリーダーとしての地位を確立する。

ソフトウェアプロセス成熟度の5段階

レベル2組織のプロセスは、プロジェクト毎に相違があってもよい。レベル2達成のための組織的要件は、適切な管理プロセスを確立するためにプロジェクトを導く方針が存在することである。文書化された手順は、トレーニングとソフトウェア品質保証の援助を伴って、組織全体で制度化されうる首尾一貫したプロセスの基盤を提供する。

レベル3は、レベル2のプロジェクト管理基盤の上に、ソフトウェアプロセス全体の定義、統合、および文書化によって構築される。この場合の統合とは、ひとつのタスクのアウトプットが、次のインプットに滑らかにつながることを意味している。タスク間にある不整合は、プロセスを実施する際に偶然見つかるよりも、むしろそれらはソフトウェアプロセスの計画段階で取り上げられる。レベル3の難しさのひとつは、過度に硬直することなく、作業を行う権限を各自に与えることである [Humphrey91b]。

2.2.3 管理されたレベルと最適化するレベルの理解

成熟度レベル4および5はソフトウェア産業では、あまり知られていない領域である。レベル4と5のソフトウェアプロジェクトや組織は、ほんのわずかの例があるだけである [Humphrey91a、Kitson92]。レベル4および5の特徴についての一般的な結論を出せるほどの例はない。これらのレベルの特徴は、他の産業からの類推と、数少ないソフトウェア産業の実例をもとに定義されている。

レベル4および5の特徴の多くは、図2.2で示すように統計的过程制御のコンセプトに基づいている。Juran Trilogy Diagram [Juran88] はプロセス管理の第一目標を示している。

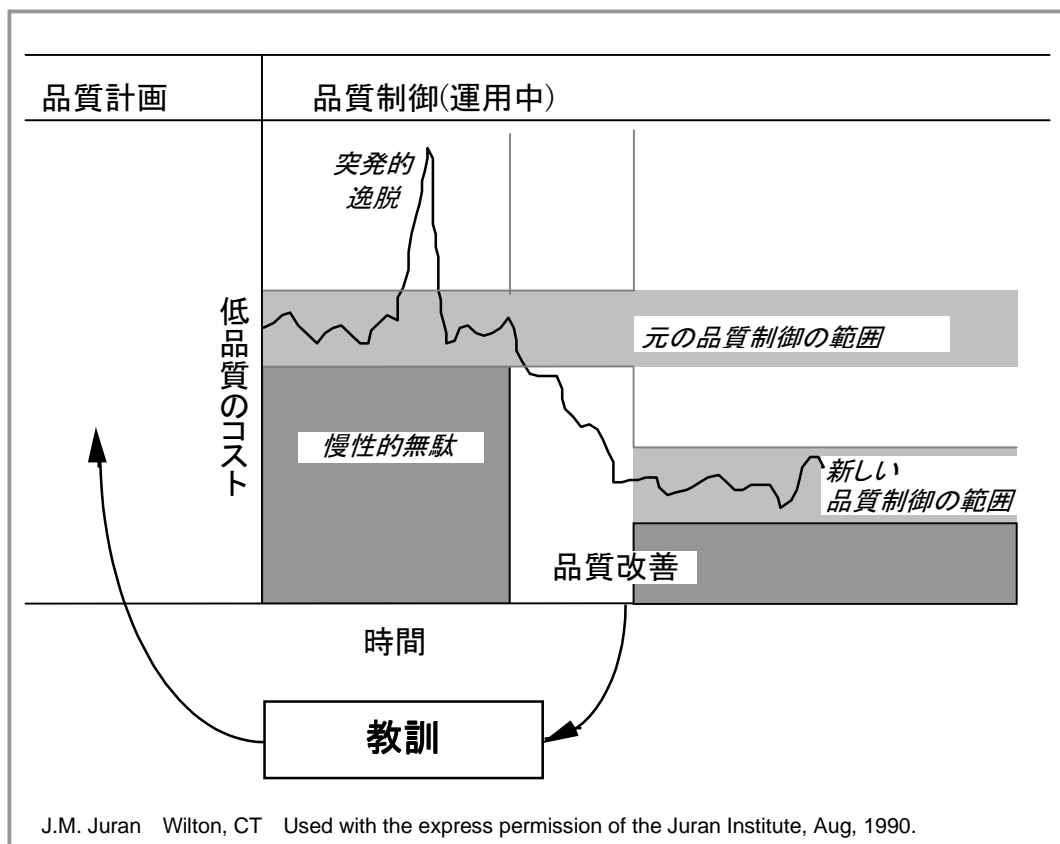


図2.2 The Juran Trilogy Diagram: 品質計画，品質制御，および品質改善

Juranは品質管理を3つの基本的管理プロセスに分類している[Juran88]。品質計画の目的は、作業部隊、つまりソフトウェア作成者に、顧客のニーズに合致する成果物を作成する手段を提供することである。作業部隊は成果物を作成するが、品質不足のためにいくつかの手戻りが必要になる。プロセスがそのように計画されているため、この無駄は慢性的なものである。事態の悪化を防ぐためには品質制御を実行する。図2.2のようにプロセス中で散発的に発生する異常値は、消火活動を表している。慢性的無駄は改善の機会を与える。その機会を捉えることは、品質改善と言われる。

ソフトウェアプロセス成熟度の5段階

第一の責務、すなわちレベル4の焦点は、プロセス制御にある。ソフトウェアプロセスは管理されているので、品質制御の範囲のもとで安定して機能している。慢性的無駄は必然的に存在し、計測結果の中には制御されるべき異常値もあるだろうが、システムは全体として一般的に安定している。このことは、変動の特殊原因を制御するコンセプトの出番を示している。プロセスが安定し計測されているので、例外的な状況が発生した場合には、変動の『特殊原因』を特定し、取り組むことが可能になる。

第二の責務、すなわちレベル5の焦点は、継続的プロセス改善である。ソフトウェアプロセスは品質を改善するために変更され、品質制御の範囲は移動する。慢性的無駄を減らす新たな実績のベースラインが確立される。そのようなプロセス改善における教訓は、将来のプロセス計画に適用される。ここで、変動の共通原因を取り上げるというコンセプトが、前面に現れる。どんなシステムの中にも、単にランダムな変動によって、手戻りという形の慢性的無駄がある。無駄は許容できないものであり、無駄を除去するための組織的な努力はシステムの変更に至り、無駄の発生を防止するために非効率性の『共通原因』を変更することによって、プロセスを改善する。

CMMの最も高い成熟度レベルに到達する組織には、非常に信頼できるソフトウェアを予想されたコストおよびスケジュールの範囲内で生産できるプロセスがあると予期される。より高い成熟度レベルに対する理解を深めることによって、既存のキープロセスエリアは洗練され、他のものがモデルに追加されるかもしれない。CMMは製造業で示唆されたプロセスのアイディアに由来しているが、ソフトウェアプロセスは、製造プロセスにあるような複製課題によって支配されているわけではない。ソフトウェアプロセスは設計課題によって支配され、知識集約的な活動である [Curtis88]。

2.3 ソフトウェアプロセスの可視性

ソフトウェアエンジニアは、プロジェクトの状況および実績に関する直接の情報を持っているため、プロジェクトの状態に対する詳細な見通しを持っている。しかしながら、大規模プロジェクトでは、通常、責任を負ってきた分野の個人的経験からしか見通せない。上級マネージャのように直接状況を把握できないプロジェクトの外部者は、プロジェクトのプロセスに対する可視性を欠き、進捗状況をモニターするために定期的レビューに依存する。図2.3は、プロセス成熟度の各レベルにおいて、管理層に与えられるプロジェクトの状況と実績の可視性のレベルを示している。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。

ソフトウェアプロセス成熟度の5段階

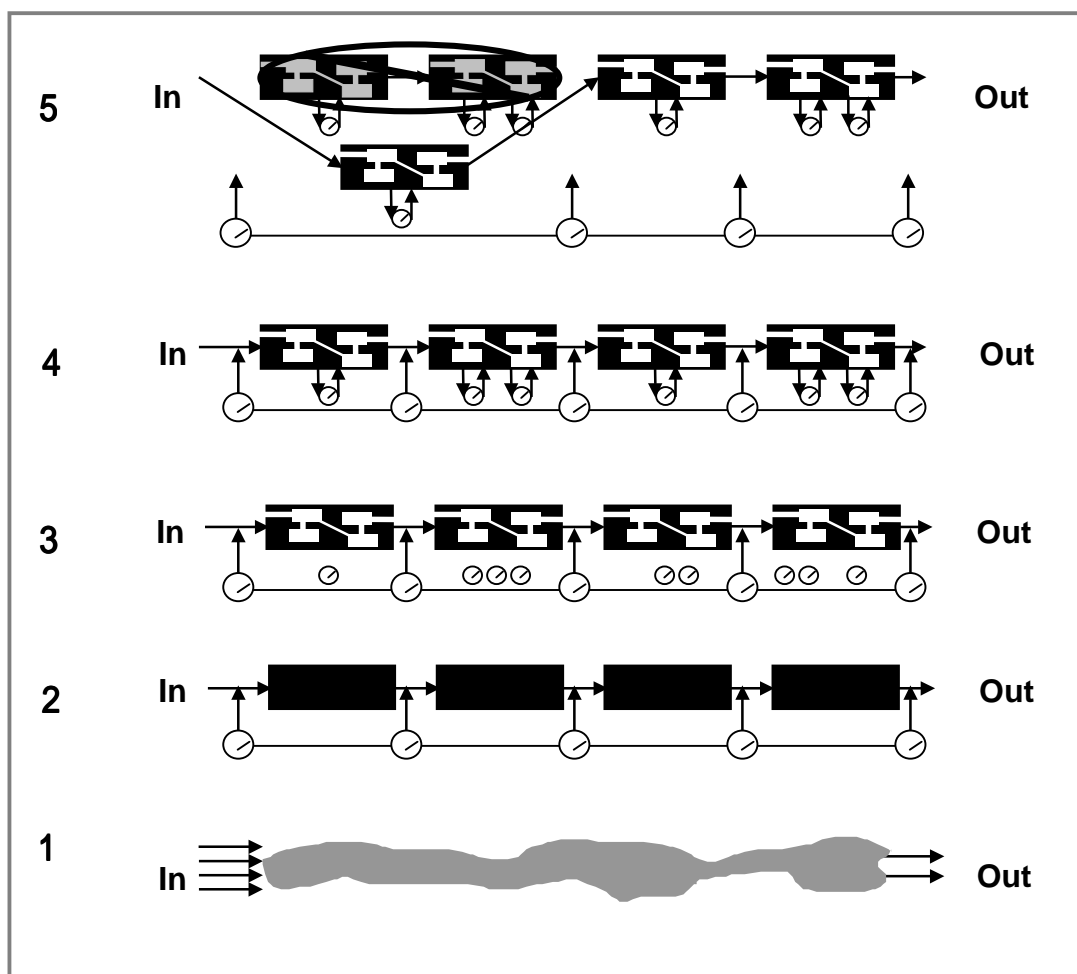


図2.3 各成熟度レベルにおける管理層から見たソフトウェアプロセスに対する可視性

レベル1では、ソフトウェアプロセスは無定型の存在 ---ブラックボックス--- 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。活動のステージ分けが貧弱に定義されているために、プロジェクトの進捗と活動の状況を確認することはマネージャにとって非常に難しい。⁴

⁴ これは90-90ルール状況である：プロジェクトの90%の期間は90%完成の状態である。

ソフトウェアプロセス成熟度の5段階

要件は制御されずにソフトウェアプロセスに流れ込み、成果物ができあがる。ソフトウェアによく通じていないマネージャにとっては、ソフトウェア開発は魔術に見える。

レベル2では、顧客の要件および作業成果物は制御され、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）。管理層は、たとえば箱の中で何が起きているかについて詳細を知らずとも、プロセスの成果物およびプロセスが機能しているかどうかを確かめるチェックポイントが特定され明らかになっている。管理層は問題が発生したときに対応する。

レベル3では、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。内部構造は、「組織の標準ソフトウェアプロセス」が特定のプロジェクトに適用された様子を表している。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。管理層は、先を見越してリスクに備えた準備をする。定義されたプロセスがプロジェクトの活動に大きな可視性を与えるので、プロジェクト外部の人々が正確かつ迅速に最新状況を得ることが可能になる。

レベル4では、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。マネージャは進捗と問題を計測できる。彼らは意思決定のための客観的で定量的な基盤を持っている。プロセスの変動が小さくなるにしたがい、成果を予測する能力は、確実により精確になる。

レベル5では、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。規律ある変化が日常になり、非効率的または欠陥を生じ易い活動は特定され、置き換えられ、あるいは改定される。見通しは、既存プロセスに対してだけでなく、

ソフトウェアプロセス成熟度の5段階

プロセスの潜在的な変更効果へと広がっていく。マネージャは変化の影響と効果性を定量的に見積り、確認することが可能になっている。

2.4 プロセス能力および実績の予測

組織のソフトウェアプロセス成熟度は、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る。レベル1組織のプロジェクトでは、コスト、スケジュール、機能充足性、および品質目標値の達成結果に大きなばらつきが見られる。図2.4で説明しているように、組織のソフトウェアプロセスが成熟するにつれて、狙った目標に合わせる上での3つの改善点が観察される。

第一に、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。例えば、同サイズのプロジェクト10件が5月1日を納品予定として狙っていたとすると、プロジェクトの平均の納品日は組織の成熟によって5月1日に近づいていく。レベル1の組織では、当初予定していた納期を大幅に外すのに対して、レベル5の組織はかなりの正確性をもって納期に合わせるができるようになる。これは、レベル5の組織は、注意深く組み立てられたソフトウェアプロセスを既知のパラメータに収まる範囲で用いており、目標日の選定は、自分たちのプロセスに関して持っている大量のデータと、そのプロセスの適用における自分たちの実績とに基づいているからである（これは図2.4の中の、目標値ラインの右側の曲線下の面積で示されている）。

ソフトウェアプロセス成熟度の5段階

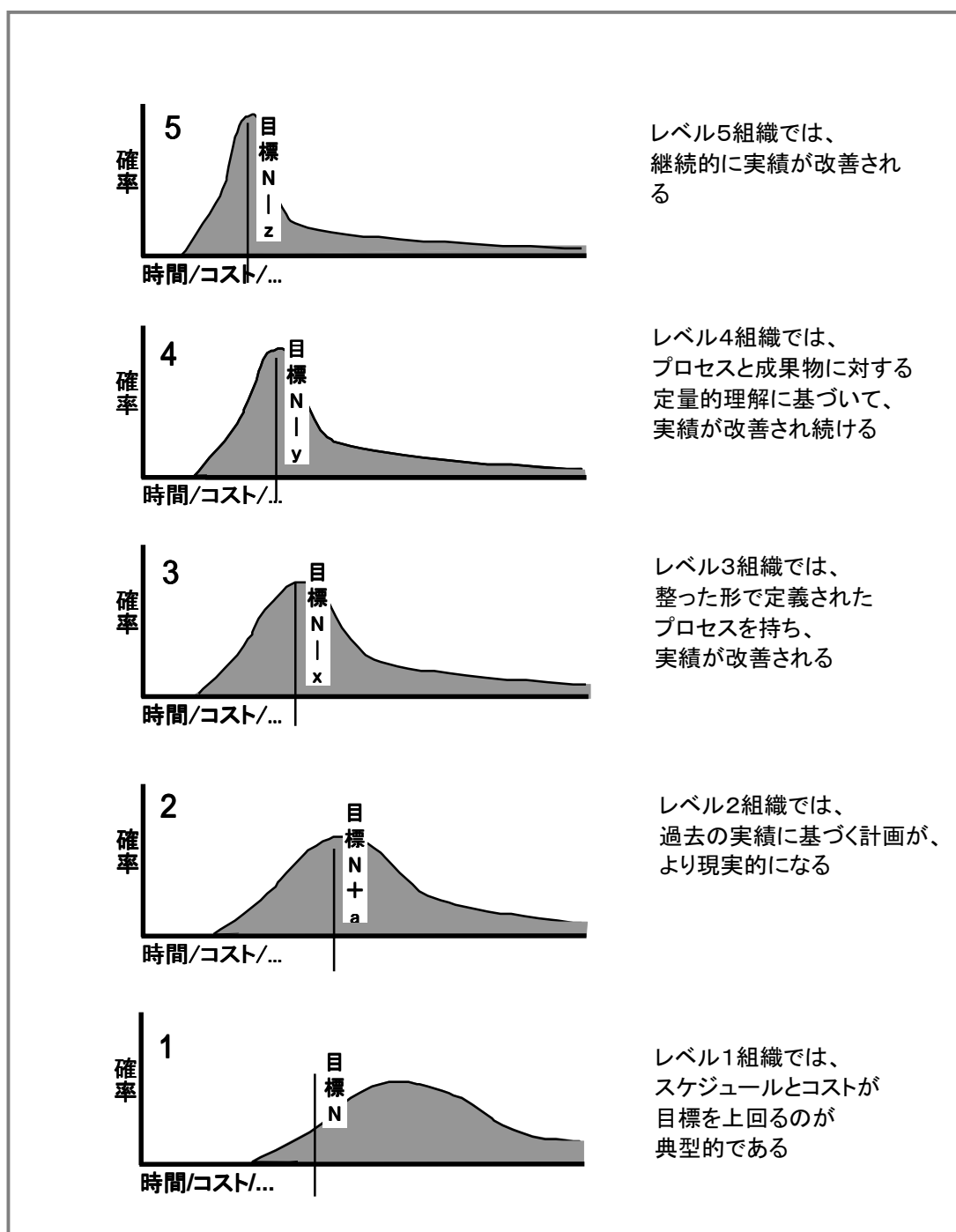


図2.4 成熟度レベル毎のプロセス能力

ソフトウェアプロセス成熟度の5段階

第二に、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。例えば、レベル1の組織では、同様のサイズのプロジェクトでも納期予想は難しく、変化も大きい。一方、レベル5の組織では、より小さい変化の範囲内で納品される。ほとんどすべてのプロジェクトは、制御されたパラメータの範囲でコスト、スケジュール、機能充足性、および品質に関する組織のプロセス能力に近づいてくるので、最高レベルの成熟度ではばらつきが小さくなる（これは図2.4において、曲線の下部分がどれほど目標値ラインの近くに集中しているかで示されている）。

第三に、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。すなわち、ソフトウェア組織が成熟するにつれ、コストは減少し、開発時間は短縮され、生産性および品質が向上する。レベル1では、誤りを訂正するための手戻りが多いために、開発時間が極めて長くなってしまう。対照的にレベル5の組織は、絶え間無いプロセス改善を行い、プロセス効率向上および手戻り削減のために欠陥予防技術を使って、開発時間の短縮を可能にする。（これは図2.4の中で、起点から目標値ラインまでの距離によって説明される）。

図2.4が示すプロジェクトの結果予測における改善は、手戻りといった形のノイズがソフトウェアプロセスから取り除かれ、ソフトウェアプロジェクトの成果がより正確に予測できるようになることを表している。先例のないシステムの場合、新しい技術およびアプリケーションが変動性を増大させてプロセス能力を低下させるため、この絵が複雑になる。先例のないシステムの場合であっても、より成熟した組織の管理とエンジニアリングのプラクティスの特徴は、成熟度の低い組織よりも開発サイクルの早い段階で問題点を明らかにし、取り組むことに役立つ。欠陥のより早い検出が、後ろのフェーズでの手戻りを削減し、プロジェクトの安定性と実績に寄与する。リスク管理は、成熟したプロセスではプロジェクト管理の必須な部分である。ある場合には、成熟したプロセスとは、『失敗した』プロジェクトをソフトウェアライフサイクルの早い段階で確認し、失敗に伴う支出を最小化することを意味している。

2.5 成熟度レベルの飛び越し

CMMの成熟度レベルはそのレベルにある組織の特徴を表している。それぞれのレベルは、次のレベルで効果的かつ効率的にプロセスを実装するための基盤を作る。しかしながら組織は、実際のレベルよりもっと高い成熟度レベルで記述されたプロセスを、有益に使うことが可能である。要件分析、設計、コーディング、テストのようなエンジニアリングプロセスは、CMMではレベル3まで論じられていないが、レベル1でもこれらの活動を実施しなければならない。レベル1やレベル2の組織でもピアレビュー（レベル3）、パレート分析（レベル4）、または新しい技術の試行（レベル5）を有益に実施することが可能であろう。レベル1からレベル2に移るにあたって組織がとるべきステップを規定する際に、レベル3の特徴であるソフトウェアエンジニアリングプロセスグループを設立することがしばしば推奨される。計測はレベル4の焦点ではあるが、それは下位の成熟度レベルでも欠けてはならない部分である。

しかしながら、これらのプロセスは、適切な基盤ができるまで、完全なポテンシャルに到達することはできない。例えば、プロジェクトが火急の場合であっても、ピアレビューが首尾一貫して実装されていなければ、ピアレビューは完全に効果的とはいえない。成熟度レベルは、そのレベルの顕著な問題について記述している。レベル1の顕著な問題は管理である。ソフトウェアプロジェクトの計画と管理で生じる困難のために、他の問題は目立たなくなる傾向がある。

レベルを飛び越すことは反生産的である。なぜなら、各々のレベルでは、次のレベルを達成するのに必要な基盤を形成するからである。CMMIは、組織が優秀なソフトウェアエンジニアリング文化を確立するために進化していく過程で通過するレベルを明確にしている。正しい基盤のないプロセスは、まさにプロセスが最も必要とされる時期に、--ストレスのある時に--失敗する。そして将来の改善の基盤になるものも残らない。

ソフトウェアプロセス成熟度の5段階

レベル1の組織が、反復できるプロセス（レベル2）を確立する前に、定義されたプロセス（レベル3）を実装しようとする、プロジェクトマネージャがスケジュールおよびコストの圧力によって押しつぶされてしまうため、通常失敗に終わってしまう。これが、エンジニアリングプロセスより先に管理プロセスに重点を置くことの根本的な理由である。エンジニアリングプロセスを定義し実装することは、管理プロセスよりも容易であるように見えるかもしれない（特に技術者にとって）。しかし管理の規律が無い場合、エンジニアリングプロセスは、コストとスケジュールの圧力のために犠牲にされてしまう[Humphrey88]。

定義されたプロセスの基盤を持たない組織が、管理されたプロセス（レベル4）を実装しようとする、定義されたプロセスなしでは計測を解釈するための共通基盤を築けないため、通常失敗してしまう。個々のプロジェクトでデータを収集することはできるが、複数のプロジェクトにまたがって有意な計測値はほとんどなく、ソフトウェアプロセスの組織的理解を深めることもない。計測されているプロセスに違いがあるため、定義されたプロセスなくして、意味のある計測をすることは困難である。

管理されたプロセス（レベル4）の基盤を持たずに、最適化するプロセス（レベル5）を実装しようとする、変化の影響に対する理解力が乏しいために失敗しがちである。統計的に狭い境界（プロセスの計測における小さい変動）内でプロセスを制御しないと、データの中に多くのノイズがありすぎて、ある特定のプロセス改善の効果を客観的に判断することができない。合理的で見識のある判断のための定量的な基盤がほとんど無い状態では、意思決定はあたかも宗教的論争へと失墜しかねないのである。

プロセス改善のための努力は、組織の事業環境をふまえ、組織の必要性に焦点を当てるべきである。より高い成熟度レベルのプロセスを実装する能力は、成熟度レベルを飛び越せることを意味しているわけではない。

3 作業面から見たCMMの定義

CMMは、ソフトウェアプロセス能力を向上しようとするソフトウェア組織にとって、推奨される改善の道筋を示す枠組みである。CMMの利用上の詳細は、その多様な使われ方を支援するように作成されている。少なくとも以下の4通りの用途を支援するようになっている。

- アセスメントチームは、組織の強みと弱みを識別するためにCMMを使用するであろう。
- 審査チームは、仕事を依頼する契約先の選定リスクを特定したり、契約をモニターするためにCMMを使用するであろう。
- マネージャや技術要員は、その組織のソフトウェアプロセス改善プログラムの計画と実装に必要な活動を理解するためにCMMを使用するであろう。
- SEPG等のプロセス改善グループは、その組織のソフトウェアプロセスを定義し、改善するための指針としてCMMを使用するであろう。

CMMの多様な使われ方からして、成熟度レベルの構造から実際のプロセスに対する勧告が導き出せるくらい十分に詳細なレベルまで、CMMは分解記述されていなければならない。また、この分解記述は、プロセスとその構造を示しており、ソフトウェアプロセス成熟度とソフトウェアプロセス能力の特徴を表している。

3.1 成熟度レベルの内部構造

各成熟度レベルは構成部品に分解されている。レベル1を除いて、各成熟度レベルの分解は、図3.1に示すように、各レベルの概要からキープラクティスによる利用上の定義に及ぶ。各成熟度レベルはいくつかのキープロセスエリアから構成されている。各キープロセスエリアはコモンフィーチャと呼ばれる5つの節で編成されている。コモンフィーチャはキープラクティスを特徴付けており、それらが全体として対処されれば、キープロセスエリアのゴールが達成されるようになっている。

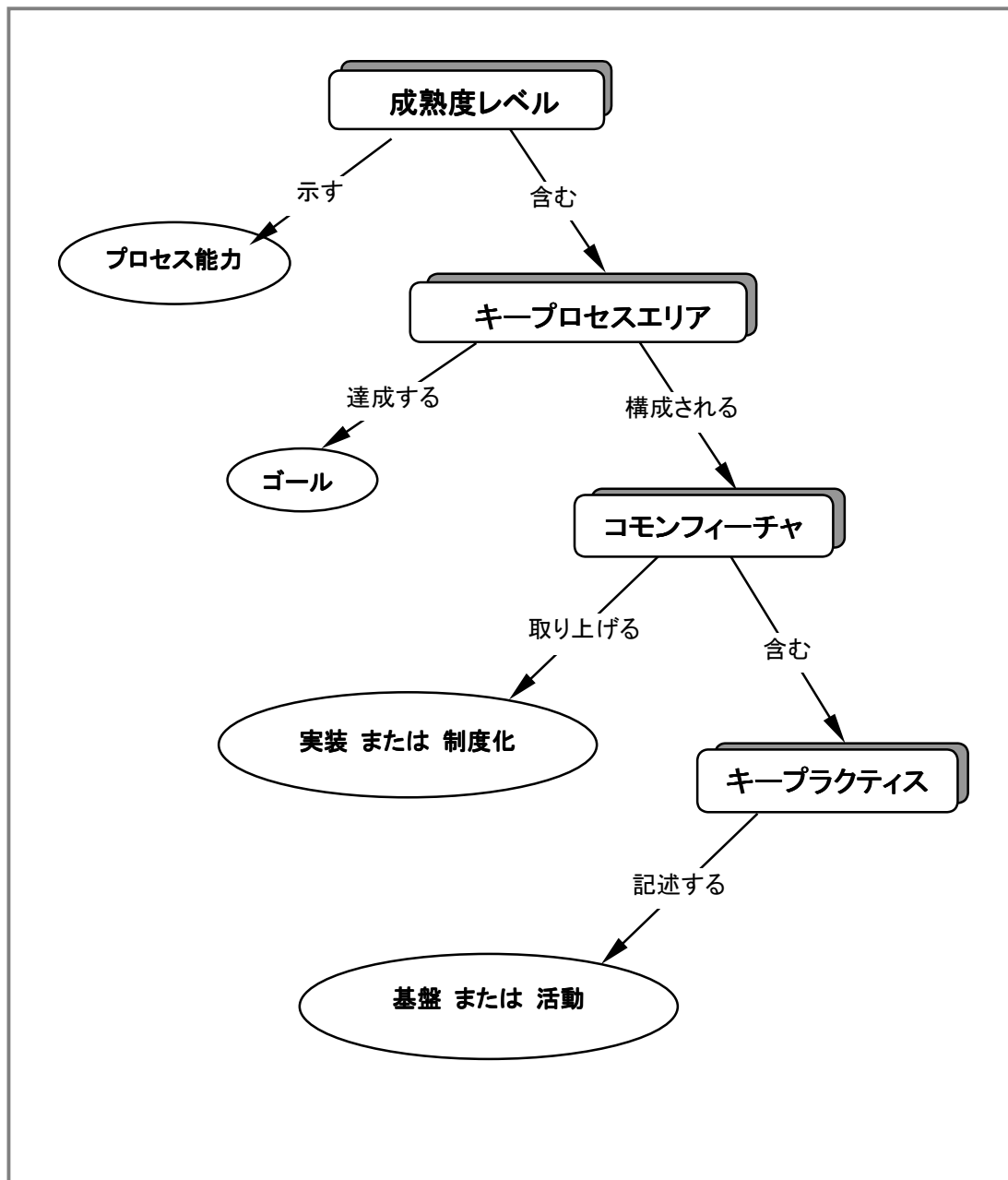


図3.1 CMMの構造

3.2 成熟度レベル

成熟度レベルは、成熟したソフトウェアプロセスを達成する途上の整った形で定義された進化の段階である。各成熟度レベルは、図2.1で表したように、プロセス能力のレベルを示す。例えば、レベル2では組織のプロセス能力は、健全なプロジェクト管理による制御が確立することにより、場当たりの状態から規律ある状態に向上している。

3.3 キープロセスエリア

レベル1を除いて、各成熟度レベルはいくつかのキープロセスエリアに分解されており、それら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領域を示している。キープロセスエリアは、成熟度レベルを達成するために取り組まねばならない問題を特定している。

各『キープロセスエリア』は、関連するひとまとまりの活動を特定しており、それら全体が実施されれば、プロセス能力の確立に重要と考えられるゴールの集まりが達成される。各キープロセスエリアは、図3.2に示すようにひとつの成熟度レベルに属するように定義されている。あるキープロセスエリアのゴール達成の道筋は、アプリケーションドメインや環境の違いによってプロジェクト毎に違って構わない。しかし、そのキープロセスエリアを組織として満足するためには、キープロセスエリアのすべてのゴールを達成しなければならない。あるキープロセスエリアのゴールがプロジェクト横断的かつ継続的に達成されたとき、その組織は、そのキープロセスエリアによって特徴付けられるプロセス能力を制度化したとすることができる。

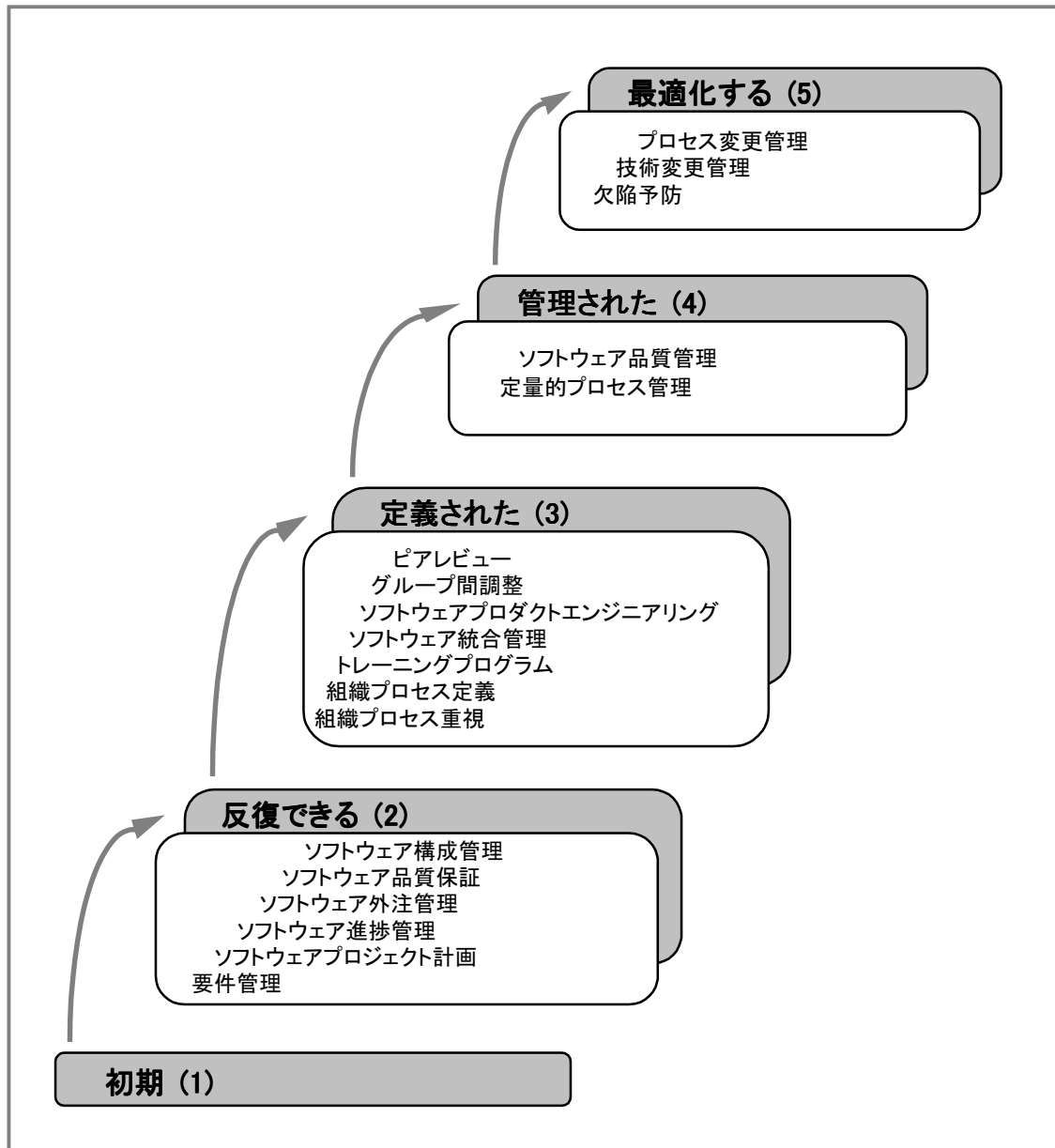


図3.2 成熟度レベル毎のキープロセスエリア

作業面から見たCMMの定義

『キー』という形容詞は、成熟度レベルの達成にキーでないプロセスエリア（やプロセス）があることを意味している。CMMは、ソフトウェアの開発や保守に関わるすべてのプロセスエリアを詳細に記述しているわけではない。プロセス能力のキー決定要因として特定のプロセスエリアが明らかにされ、それらがCMMの中に記述されている。

他の問題もプロセス実績に影響するが、組織のソフトウェアプロセス能力の改善に効果的なものがキープロセスエリアとして識別されている。これらは成熟度レベル達成の要件と考えることができる。図3.2は、各成熟度レベルのキープロセスエリアを表している。ある成熟度レベルを達成するには、そのレベルのキープロセスエリアを満足しなければならない。あるキープロセスエリアを満足するには、そのキープロセスエリアのすべてのゴールを満足しなければならない。『ゴール』は、そのキープロセスエリアのキープラクティスを要約しており、組織やプロジェクトがそのキープロセスエリアを効果的に実装したかどうかを決定するためにも利用できる。ゴールは、各キープロセスエリアの範囲、境界、意図を表している。⁵

各キープロセスエリアで実行される特定のプラクティスは、その組織が高レベルのプロセス成熟度を達成するにつれて発展するであろう。例えば、レベル2のキープロセスエリアである『ソフトウェアプロジェクト計画』の中で記述されているプロジェクト見積り能力は、レベル3、4、5で追加利用できるようになるプロジェクトデータを取り扱うように発展させなければならない。レベル3の『ソフトウェア統合管理』は、定義されたソフトウェアプロセスを利用してプロジェクトが管理されるようになって、レベル2の『ソフトウェアプロジェクト計画』と『ソフトウェアプロジェクト進捗管理』が発展したものである。

CMMのキープロセスエリアは、どのようにして組織が成熟するかを記述するひとつの方法である。これらのキープロセスエリアは、ソフトウェアのエンジニアリングと管理の長年の経験や、ソフトウェアプロセスアセスメントやソフトウェア能力審査の5年以上の経験に基づいて定義された。

⁵ 各キープロセスエリアのゴールのリストについては、付録Aを参照。

レベル2のキープロセスエリアは、ソフトウェアプロジェクトの関心事に焦点を当てており、基本的なプロジェクト管理制御を確立することに関連している。レベル2の各キープロセスエリアの説明は以下の通り：

- 『要件管理』の目的は、顧客とソフトウェアプロジェクトの間で、そのプロジェクトが取り上げる顧客要件に関しての共通理解を確立することである。この顧客との合意は、ソフトウェアプロジェクトを（『ソフトウェアプロジェクト計画』で記述されているように）計画し、（『ソフトウェアプロジェクト進捗管理』で記述されているように）管理する基盤になる。顧客との関係の制御は、（『ソフトウェア構成管理』で記述されているような）効果的な変更制御プロセスに従うことに依存する。
- 『ソフトウェアプロジェクト計画』の目的は、ソフトウェアエンジニアリングを実施し、またソフトウェアプロジェクトを管理するための妥当な計画を確立することである。これらの計画は、（『ソフトウェアプロジェクト進捗管理』で記述されているような）ソフトウェアプロジェクトの管理のために必要な基盤である。現実的な計画なしには、効果的なプロジェクト管理は実装できない。
- 『ソフトウェアプロジェクト進捗管理』の目的は、ソフトウェアプロジェクトの実績がソフトウェア計画から著しく逸脱した時に、管理層が効果的な処置をとることができるように、実際の進捗状況に対する十分な可視性を与えることである。
- 『ソフトウェア外注管理』の目的は、適格なソフトウェア外注先を選定し、それらを効果的に管理することである。『ソフトウェア外注管理』は、『ソフトウェア品質保証』や『ソフトウェア構成管理』で必要とされる調整に加えて、基本的な管理制御に対する『要件管理』、『ソフトウェアプロジェクト計画』、および『ソフトウェア進捗管理』の関心事を適宜組み合わせ、外注先に制御を適用する。

作業面から見たCMMの定義

- 『ソフトウェア品質保証』の目的は、ソフトウェアプロジェクトで用いられているプロセスならびに開発中の成果物に関し、適切な可視性を管理層に提供することである。『ソフトウェア品質保証』は、ソフトウェアのエンジニアリングと管理の大半のプロセスにおいて、必須な部分である。
- 『ソフトウェア構成管理』の目的は、プロジェクトのソフトウェアライフサイクルの全般にわたって、ソフトウェアプロジェクトの成果物の一貫性を確立し維持することである。『ソフトウェア構成管理』は、ソフトウェアのエンジニアリングと管理の大半のプロセスにおいて、必須な部分である。

レベル3のキープロセスエリアは、プロジェクトと組織の両方の課題を取り上げている。レベル3では、効果的なソフトウェアのエンジニアリングと管理のプロセスを、組織がプロジェクト横断的に制度化する基盤を確立する。レベル3の各キープロセスエリアの説明は以下の通り：

- 『組織プロセス重視』の目的は、組織の全体的なソフトウェアプロセス能力を改善するソフトウェアプロセス活動について、組織の責任を確立することである。『組織プロセス重視』活動の主要な成果は、『組織プロセス定義』で記述されている一群のソフトウェアプロセス資産である。『ソフトウェア統合管理』で記述されているように、これらの資産はソフトウェアプロジェクトが利用する。
- 『組織プロセス定義』の目的は、プロジェクト横断的にプロセス実績を改善するのに利用できるソフトウェアプロセス資産を開発し維持し、そして組織に累積的で長期的に利益をもたらす基盤を提供することである。これらの資産は、『トレーニングプログラム』で記述されているトレーニングなどのメカニズムによって制度化し得る安定した基盤を提供する。

- 『トレーニングプログラム』の目的は、個人がスキルと知識を身につけることにより、各自の役割を効果的かつ効率的に遂行できるようにすることである。トレーニングは組織の責任であるが、ソフトウェアプロジェクトは、必要なスキルを明確化するべきであり、その必要性がプロジェクト固有のものである場合は、必要なトレーニングを実施するべきである。
- 『ソフトウェア統合管理』の目的は、ソフトウェアのエンジニアリング活動と管理活動とを一体化して、定義されたソフトウェアプロセスに統合することである。定義されたソフトウェアプロセスは、「組織の標準ソフトウェアプロセス」や関連したプロセス資産からテラリングされたものである。これらについては、『組織プロセス定義』キープロセスエリアに記述されている。このテラリングは、『ソフトウェアプロダクトエンジニアリング』で記述されているように、プロジェクトの事業環境や技術的必要性に基づいている。『ソフトウェア統合管理』は、レベル2の『ソフトウェアプロジェクト計画』と『ソフトウェア進捗管理』が発展したものである。
- 『ソフトウェアプロダクトエンジニアリング』の目的は、整った形で定義されたエンジニアリングプロセスを首尾一貫して実施することである。このプロセスは、すべてのソフトウェアエンジニアリング活動を統合し、正しくかつ首尾一貫したソフトウェア成果物を効果的かつ効率的に作成する。『ソフトウェアプロダクトエンジニアリング』は、要件分析、設計、コーディング、およびテストなど、プロジェクトの技術的活動を記述する。
- 『グループ間調整』の目的は、プロジェクトが顧客のニーズをより効果的かつ効率的に満足させるため、ソフトウェアエンジニアリンググループが積極的にその他のエンジニアリンググループと連携する手段を確立することである。『グループ間調整』は、『ソフトウェア統合管理』の分野横断的な側面であり、ソフトウェアエンジニアリングの範囲外にまで及ぶ。ソフトウェアプロセスを統合するだけでなく、ソフトウェアエンジニアリンググループと他のグループとの連携も調整され、制御されなければならない。
- 『ピアレビュー』の目的は、早期に効率よくソフトウェア作業成果物から欠陥を取り除くことである。ピアレビューには、ソフトウェア作業成果物や予防され得る欠陥について理解を深めるという重要な付随的效果

作業面から見たCMMの定義

がある。『ピアレビュー』は、重要かつ効果的なエンジニアリング手法であり、『ソフトウェアプロダクトエンジニアリング』の一環として実施される。Faganスタイルインスペクション[Fagan86]、構造化ウォークスルー、あるいは他の同様のいくつかのレビュー手法[Freedman90]を用いて実装することができる。

レベル4のキープロセスエリアは、ソフトウェアプロセスと構築中のソフトウェア作業成果物の両方に対する定量的理解を確立することに焦点を当てている。このレベルのふたつのキープロセスエリア、すなわち『定量的プロセス管理』と『ソフトウェア品質管理』の間には、緊密な相互依存関係がある。レベル4の各キープロセスエリアの説明は以下の通り：

- 『定量的プロセス管理』の目的は、ソフトウェアプロジェクトのプロセス実績を定量的に制御することである。ソフトウェアプロセス実績は、ソフトウェアプロセスに従うことによって達成された実際の結果を表わす。このキープロセスエリアの焦点は、計測可能な安定したプロセスにおける変動の特殊原因を特定し、一時的な変動を引き起こす事情を適宜正すことである。『定量的プロセス管理』は、『組織プロセス定義』、『ソフトウェア統合管理』、『グループ間調整』、『ピアレビュー』のプラクティスに対して、包括的計測プログラムを付加する。
- 『ソフトウェア品質管理』の目的は、プロジェクトのソフトウェア成果物の品質を定量的に理解し、特定の品質目標を達成することである。『ソフトウェア品質管理』では、『ソフトウェアプロダクトエンジニアリング』で記述されたソフトウェア作業成果物に対して、包括的な計測プログラムを適用する。

レベル5のキープロセスエリアは、組織とプロジェクトの両方が、継続的で計測可能なソフトウェアプロセス改善を実装するために取り組まねばならない課題を扱っている。レベル5の各キープロセスエリアの説明は以下の通り：

- 『欠陥予防』の目的は、欠陥の原因を特定し、その再発を予防することである。ソフトウェアプロジェクトは、欠陥を分析し、その原因を突き止め、『ソフトウェア統合管理』で記述されているように、その定義されたソフトウェアプロセスを変更する。汎用的価値のあるプロセス変更は、『プロセス変更管理』で記述されているように、他のソフトウェアプロジェクトにも移転される。
- 『技術変更管理』の目的は、新しい技術（ツール、手法、およびプロセス）を特定し、『プロセス変更管理』で記述されているように、整然とした方法でそれらを組織に導入することである。『技術変更管理』の焦点は、変化し続ける世界の中で革新を効率的に遂行することである。
- 『プロセス変更管理』の目的は、組織で使用されているソフトウェアプロセスを継続的に改善していくことであり、その意図は、成果物の開発において、ソフトウェア品質の改善、生産性の向上、およびサイクルタイムの短縮である。『プロセス変更管理』は、『欠陥予防』による漸進的改善と『技術変更管理』による革新的改善を利用し、それらの改善を組織全体で活用できるようにする。

3.4 コモンフィーチャ

便宜上、キープロセスエリアはコモンフィーチャで構成されている。コモンフィーチャは、キープロセスエリアの実装と制度化が効果的で、反復でき、永続するかどうかを示す属性である。5つのコモンフィーチャは下記の通りである：

作業面から見たCMMの定義

実施のコミットメント 『実施のコミットメント』は、プロセスが確立され持続することを確実なものにするために組織がとらなければならない処置を記述している。『実施のコミットメント』は、組織方針や上級管理層の主催者としての態度などを確立することである。

実施能力 『実施能力』は、ソフトウェアプロセスを十分に実装するために、プロジェクトまたは組織に存在しなければならない前提条件を記述している。『実施能力』は、資源、組織構造、トレーニングなどである。

実施される活動 『実施される活動』は、キープロセスエリアを実装するのに必要な役割や手順を記述している。『実施される活動』は、計画と手順を確立し、作業を実施し、それを確認し、そして必要に応じた是正処置をとることである。

計測と分析 『計測と分析』は、プロセスの計測と計測値の分析の必要を記述している。『計測と分析』は、『実施される活動』の状況や有効性を判断するための計測の例などを含んでいる。

履行検証 『履行検証』は、確立されたプロセスを遵守して活動が実施されることを確実なものにするための処置を記述している。『検証』は、管理層やソフトウェア品質保証によるレビューや監査を含んでいる。

『実施される活動』コモンフィーチャにおけるプラクティスは、プロセス能力を確立するために何が実装されなければならないかを記述している。他のプラクティスは全体として、『実施される活動』コモンフィーチャで記述されているプラクティスが組織内で制度化されるための基盤を形成している。

3.5 キープラクティス

各キープロセスエリアは、そのゴールの満足に寄与するキープラクティスによって記述されている。『キープラクティス』は、キープロセスエリアの効果的な実装と制度化に大きく寄与する基盤と活動を記述している。

各キープラクティスは、一文で表現されており、多くの場合、例や補足説明などの詳細記述が付け加えられている。これらのキープラクティスを、トップレベルキープラクティスとも言い、キープロセスエリアに対する基本方針、手順、活動を記述している。詳細記述部分は、サブプラクティスとも言われる。図3.3では、『ソフトウェアプロジェクト計画』のキープロセスエリアを例にキープラクティスの背後にある構造を示した。

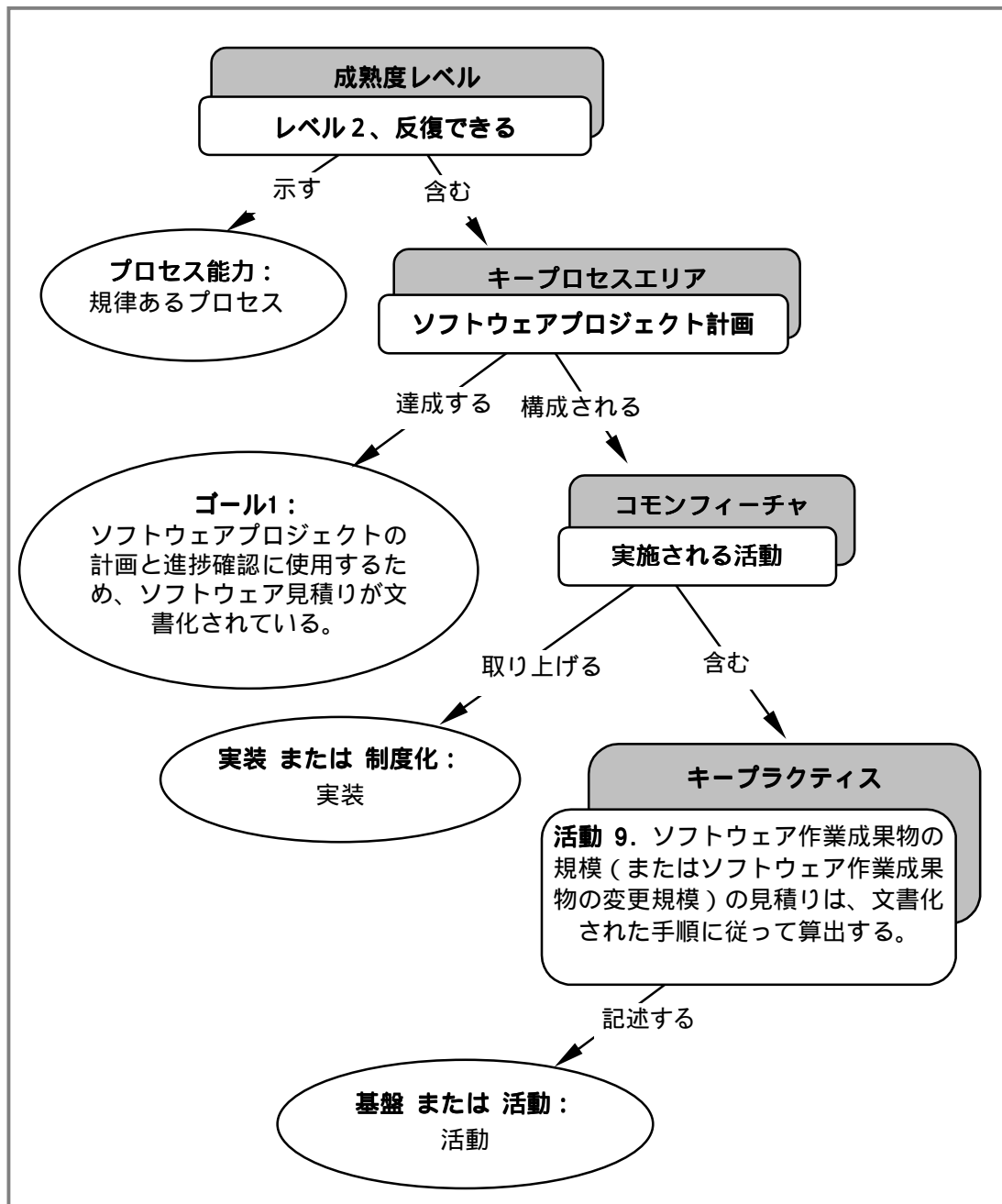


図3.3 CMM構造の編成： キープラクティスの例

図3.3で図示したように、プロジェクトを計画し進捗確認するために計画を文書化するというゴールを首尾一貫して達成することを確実なものにするためには、ソフトウェア規模を見積るための文書化された手順を組織が確立しなければならない。もし、このような見積りが文書化された手順に依らずに作成されると、規模見積りの前提がはっきりしないので、そのばらつきが大きくなる。このような手順に望まれる具体的な記述としては、過去の規模データの利用、前提の文書化、見積り値のレビューなどがある。このような基準は、合理的な規模見積り手順に従ったかどうかの判定に使える。

キープラクティスは『何』がなされるかを記述したものであって、ゴールの達成を『どのように』するかを強制していると解釈してはならない。別のプラクティスがキープロセスエリアのゴールを達成することもあるだろう。キープラクティスは、キープロセスエリアのゴールが、異なるやり方であっても、効果的に達成されたかどうかを判断するために合理的に解釈されるべきものである。キープラクティスは、『能力成熟度モデルのキープラクティス1.1版』[Paulk93b]に載っており、その解釈のガイダンスも一緒に載っている。

4 CMMを使う

CMMは、成熟したソフトウェア組織の特徴を記述しており、一般に利用可能な一連の基準を確立している。これらの基準は、組織により開発と保守のプロセスを改善するために、あるいは、政府または企業により、特定の会社とのソフトウェアプロジェクト契約のリスクを評価するために使われる。

この章では、ソフトウェアプロセスアセスメントとソフトウェア能力審査に焦点を当てる。SEIで開発されたこれらふたつの手法は、ソフトウェアプロセスの組織による実行の成熟度を査定するものである。

- 『ソフトウェアプロセスアセスメント』は、組織のソフトウェアプロセスの現状を判断し、組織が直面する優先順位の高いソフトウェアプロセス関連課題を決定し、そしてソフトウェアプロセス改善に対する組織的支援を獲得することに使われる。
- 『ソフトウェア能力審査』は、ソフトウェア作業の実施に対して適格な契約者を特定したり、また、実施中のソフトウェア開発に使われているソフトウェアプロセスの状態をモニターするために使われる。

アセスメントまたは審査を開催する読者には、本概観だけでは十分で無い。これらの手法を通じてCMMを適用したい者は、アセスメントと審査のためのトレーニングについてさらに情報を求めるべきである。

CMMは、ソフトウェアプロセスアセスメントとソフトウェア能力審査の共通基盤である。しかしながら、これら手法の目的は極めて異なっており、使われている具体的な手法にも著しい違いがある。両手法はモデル（CMM）とそれから派生する成果物に基づいている。CMMは、CMMベースの成果物と組み合わせることにより、手法の首尾一貫した信頼できる適用を可能にしている。

4.1 ソフトウェアプロセスアセスメントとソフトウェア能力審査手法

ソフトウェアプロセスアセスメントは、組織自身のソフトウェアプロセス内において改善の優先順位を見極めることに重点を置いている。アセスメントチームは、CMMを指針として使い、所見を見極め優先順位を付ける。これらの所見は、CMMのキープラクティスによる指針とともに、（例えばソフトウェアエンジニアリングプロセスグループによって）組織のための改善戦略を立てるために使われる。

ソフトウェア能力審査は、高品質のソフトウェアを期限通りに予算内で構築するために、特定のプロジェクトや契約に関連するリスクを見極めることに重点を置いている。調達プロセスの中で、入札者に対してソフトウェア能力審査を実施することがある。CMMIによって構造化された審査の所見は、特定の契約者を選定する際にリスクを見極めるために使用されることがある。審査はまた、契約者のソフトウェアプロセスの潜在的な改善事項を見極めるために、現在の契約について、彼らのプロセス実績をモニターするために実施されることがある。

CMMIは、ソフトウェアプロセスアセスメントおよびソフトウェア能力審査を実施する際に参照する共通の枠組みを確立する。ふたつの手法は目的が異なっているが、これらはソフトウェアプロセス成熟度を査定するための基盤としてCMMIを使っている。図4.1にアセスメントと審査について共通ステップの概要を示す。

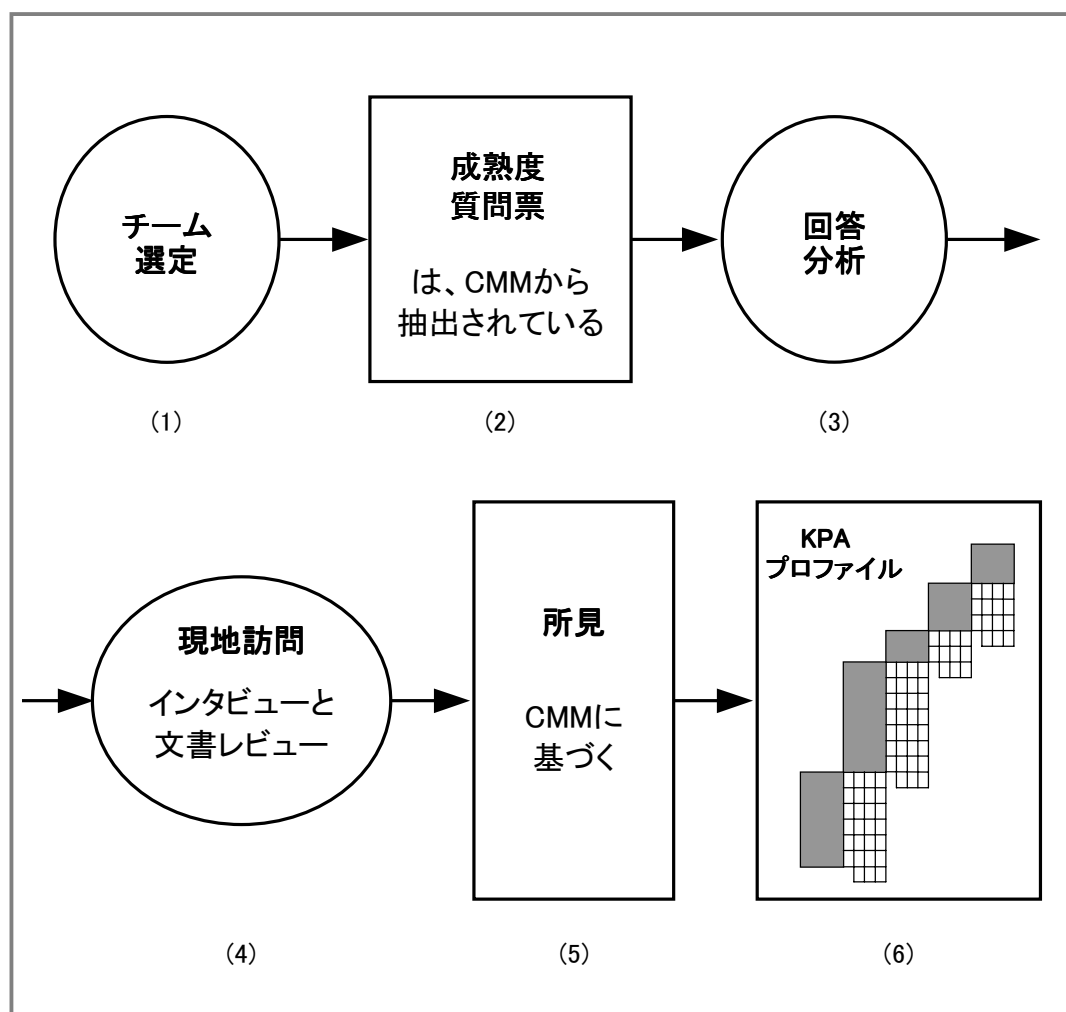


図4.1 ソフトウェアプロセスアセスメントとソフトウェア能力審査の共通ステップ

第1ステップはチームを選定することから始まる。このチームは、アセスメントまたは審査の手法の詳細および、CMMの基本的コンセプトについてトレーニングを受けるべきである。チームメンバは、ソフトウェアのエンジニアリングと管理の十分な知識がある専門家であるべきである。

第2ステップは、アセスメントまたは審査の対象となる対象組織の代表者に、成熟度質問票と他の診断手段を完成してもらうことである。この活動が完了したら、アセスメントまたは審査チームは、質問と回答を照らし合わせ、更なる探求が必要な領域を見極めるために、回答分析を実施する（第3ステップ）。その調査対象領域は、CMMのキープロセスエリアに対応している。

これでチームは、アセスメントまたは審査するために現地を訪れる（第4ステップ）準備が整う。回答分析結果を手掛かりにして、チームはその現地が従っているソフトウェアプロセスを理解するために、インタビューやドキュメントのレビューを開催する。CMMのキープロセスエリアとキープラクティスは、チームメンバの質問、聞き取り、レビュー、そしてドキュメントとインタビューから受け取った情報の統合の指針になる。チームは、現地のキープロセスエリアの実装が、関連したキープロセスエリアのゴールを満足させるかどうかの決定に、専門家の判定を適用する。⁶ CMMのキープラクティスと現地でのプラクティスの間に明らかな違いがあるときには、チームはそのキープロセスエリアの判断の根拠を文書化しなければならない。

現地調査期間の最後に、チームはその組織のソフトウェアプロセスの強みと弱みを見極めた所見の一覧を作成する（第5ステップ）。ソフトウェアプロセスアセスメントでは、所見はプロセス改善のための勧告の基盤になる。ソフトウェア能力審査では、所見は調達機関により実施されるリスク分析の一部になる。

最後に、チームはその組織がキープロセスエリアのゴールを満足しているか否かを示すキープロセスエリアプロファイルを作成する（第6ステップ）。あるキープロセスエリアに関連する所見があったとしても、ゴールの達成を妨げる大きな課題が特定されない場合は、そのキープロセスエリアが満足されていると判断することができる。

⁶ その会社の知的所有権または機密事項に関連するかもしれない場合には、これらの判定を完全な情報無しに行わねばならないかもしれない。

まとめてとして、ソフトウェアプロセスアセスメントおよびソフトウェア能力審査手法の双方は、

- 現地訪問の足がかりとして成熟度質問票を使う。
- 現地調査の指針となる地図としてCMMを使う。
- CMMのキープロセスエリアによって強みと弱みを特定する所見を作成する。
- キープロセスエリア内のゴール達成分析に基づきプロファイルを作成する。
- キープロセスエリアプロファイルおよび所見により、適切な対象者にこれらの結果を発表する。

4.2 ソフトウェアプロセスアセスメントとソフトウェア能力審査の違い

これらの類似にもかかわらず、また同一手法の連続する適用においてさえも、ソフトウェアプロセスアセスメントやソフトウェア能力審査の結果は異なることがある。ひとつの理由は、アセスメントまたは審査の範囲が異なる場合があることである。第1に、調査される組織が決定されなければならない。大企業では、いくつかの異なる『組織』の定義が可能である。その範囲は、共通の上級管理層、共通の地理的位置、プロフィットセンタとコストセンタの指定、共通のアプリケーション領域、または他の考慮要素に基づいてよい。第2に、同一に見える組織であっても、選定したサンプルプロジェクトが範囲に影響を与えるかもしれない。アセスメントチームが会社のある部門全体の範囲に基づく所見を導いて、その部門をアセスメントしたとしよう。その後で、その部門のある製品ラインが審査され、同じチームがもっと狭い範囲に基づく所見に到達するかもしれない。状況を理解すること無しの結果の比較には問題がある。

ソフトウェアプロセスアセスメントとソフトウェア能力審査は、動機、ねらい、成果、および結果の所有権において異なっている。これらの要因は、インタビューの進め方、質問の範囲、収集される情報、および成果の形成に、かなりの違いを生じさせる。アセスメントと審査の手法は、用いる詳細な手順を吟味す

ればきわめて異なる。アセスメントのトレーニングは、審査を実施するチームのための準備にはならないし、逆もまた同様である。

ソフトウェアプロセスアセスメントは、開かれた協力的な環境で実施される。その成功は、組織を改善しようとする管理層と専門要員の双方からのコミットメントに依存している。そのねらいは、問題を浮かび上がらせ、マネージャと専門要員が組織を改善するのを助けることにある。質問票は、アセスメントチームが成熟度レベルの課題に焦点を当てるのに役立つ。しかし、組織のソフトウェアプロセスを理解するための道具として重要視されるのは、構造化された、あるいは構造化されていないインタビューである。組織が直面しているソフトウェアプロセス上の課題の特定以外に、改善に対するBuy-in(納得して支援すること)、組織全体のプロセスへの重視、およびアクション計画実行への動機と熱意は、アセスメントの最も価値ある成果である。

一方ソフトウェア能力審査は、もっと監査指向の環境で実施される。チームの勧告は契約者の選定や報奨の設定に影響するので、そのねらいは金銭的な考慮に結び付けられている。主眼は、組織によって実際に実施されたソフトウェアプロセスを明らかにする文書化された監査の証拠である。

しかしながらこれは、ソフトウェアプロセスアセスメントとソフトウェア能力審査の結果が比較できてはならないと言っているのではない。両手法はCMMに基づくため、比較と相違のポイントは明白かつ説明可能であるべきだ。

4.3 プロセス改善におけるCMMの他の用途

ソフトウェアエンジニアリングプロセスグループなど、ソフトウェアプロセスを改善しようと努めている人々にとって、CMMは、アクション計画の作成、アクション計画の実施、およびプロセスの定義の各領域で具体的な価値を持っている。ソフトウェアエンジニアリングプロセスグループのメンバは、彼らのソフトウェアプロセスの課題と事業環境の知識を備えており、アクション計画作成時に、彼らの現在のプラクティスをCMMのキープロセスエリアのゴールと比較できる。キープラクティスは、企業の目標、経営上の優先順位、プラクティスの実績レベル、各プラクティスの組織での実装の価値、およびプラクティスを実装するための組織の能力との関連の中で、その文化に照らして吟味されるべきである。

ソフトウェアエンジニアリングプロセスグループは、次に、どの改善項目が必要か、どのように変化をもたらすか、および必要なBuy-in（納得して支援すること）をどのように獲得するかを決定しなくてはならない。CMMはこれらの活動を援助する上で、プロセス改善の議論の起点を与え、一般に受け入れられているソフトウェアエンジニアリングプラクティスと根本的に異なる前提を表面化させる助けになる。アクション計画の実施において、ソフトウェアエンジニアリングプロセスグループは、実際に使用するアクション計画の各部分を構築し、ソフトウェアプロセスを定義するために、CMMとそのキープラクティスを使用できる。

5 CMMの将来の方向性

より高いレベルのソフトウェアプロセス成熟度を達成することは漸進的であり、継続的なプロセス改善への長期的なコミットメントが必要である。継続的なプロセス改善の基盤およびそれを指向する文化をソフトウェア組織に築くためには、10年またはそれ以上かかるだろう。10年もの長きにわたるプロセス改善プログラムは、米国のほとんどの企業にとって異質のもののように映るだろうが、ソフトウェア組織に成熟をもたらすためには、この水準の努力が要求される。この期間は、プロセス成熟度の大きな向上を達成した米国の自動車産業のように、他の産業における経験に照らし合わせても首尾一貫している[Gabor90]。

5.1 CMMが取り扱わないもの

CMMは、銀の弾丸[Brooks87]ではないし、成功するプロジェクトにとって重要なすべての課題を取り上げているものでもない。例えば、現在CMMは、特定のアプリケーションドメインにおける専門技術について取り上げていないし、特定のソフトウェア技術も支持しないし、また有能な人材を選抜し、雇用し、動機付けし、そして保持するための方法も示唆していない。これらの課題はプロジェクトの成功に決定的であり、その中の幾つかはこれまで他の分野で分析されてきている[Curtis90]が、それらはまだCMMへ統合されていない。CMMは、特にソフトウェアの管理およびエンジニアリングのプロセス上の課題を取り上げ、整然とした規律ある枠組みを提供するために開発された。

5.2 当面の活動

CMMに関するチュートリアルおよびコースは、米国中の主要なコンファレンスおよびセミナーで開催されており、CMMおよびそれに関連した成果物に対する十分な意識をソフトウェア産業界が持つことを確実なものにしようとしている。CMMをベースにしたツール(例：成熟度質問票)、ソフトウェアプロセスアセスメントのトレーニング、およびソフトウェア能力審査のトレーニングは、CMMを具体化するために開発され改訂され続けている。

CMM開発活動の当面の焦点は、小さなプロジェクトならびに小さな組織向けのCMMといったテーラリングされたバージョンのCMMに向うであろう。CMM v1.1は、大規模で、政府と契約するような組織の規範的なプラクティスについて記述している。したがって、これらのプラクティスは、このテンプレートとは異なる組織のニーズに応じてテーラリングされなければならない。

5.3 長期的な活動

今後2、3年の間、CMMはソフトウェアプロセスアセスメント、およびソフトウェア能力審査での使用を通して、広範なテストを受け続けるであろう。CMMをベースにした成果物およびトレーニング教材は、適宜開発され改訂され続ける。CMMは、改良され続ける生きた文書である。しかし、CMM v1.1は少なくとも1996年までは、ベースラインとして残ることが予想されている。これは、安定性に対するニーズと継続的な改善に対するニーズとの間で、適切で現実的なバランスを提供するためである。

CMMの次のバージョンであるバージョン2では、SEIは全体のモデルを改良することに注力するだろう。モデルの全レベルが改訂されるかもしれないが、レベル4-5に力点をおくことになるだろう。現在、レベル2-3のためのキープロセスエリアは、全レベルの中で最も完全に定義されている。レベル4-5にあるとアセスメントされた組織はほとんど無い[Humphrey91a, Kitson92]ので、そのような組織の特徴は、レベル2-3に比べてよく知られていない。SEIが、レベル4-5を理解し達成しようと努力している組織と緊密に作業を進める中で、これらふたつのレベルのためのプラクティスがより洗練されていくだろう。また、CMMは技術および人的資源についての課題に取り組み解決していくために、多次元なモデルになるかもしれない。

SEIはまた、ソフトウェアプロセスアセスメント、改善、および能力審査に向けた国際標準を構築するために、国際標準化機構(ISO)と共に作業を進めている。ISO標準の開発はCMM v2に影響を与えるし、SEIのプロセスに関しての作業はISOの活動にも影響を与えるだろう。

5.4 結論

継続的な改善は、ソフトウェアプロセスに適用されたのと全く同様に、成熟度モデルそのものならびにプラクティスにも適用される。CMMの変更によるソフトウェア関連業界への潜在的な影響は、今後注意深く検討されるであろう。しかし、CMM、成熟度質問票、およびソフトウェアプロセスアセスメントとソフトウェア能力審査手法は、ソフトウェアプロセス改善の経験が増加するにつれて、進化し続けるだろう。SEIは、この進化の努力を続ける上で、産業界、政府、および学界と緊密に作業を進めるつもりである。

CMMIは、ソフトウェア成果物の管理と開発について、規律ある首尾一貫した方法で改善するための概念的な構造を提供する。しかし、ソフトウェア成果物がうまく構築されること、あるいはすべてのソフトウェアエンジニアリングの問題が十分に解決されることを保証するものではない。CMMIは、成熟したソフトウェアプロセスのためのプラクティスを特定すると共に、その標準的水準(そしてあるケースにおいては最高水準)の例を提供する。しかしこのことは、網羅的または強制的であることを意図していない。CMMIは、効果的なソフトウェアプロセスの特徴を特定している。しかし、成熟度の高い組織は、プロセスと同様に、人的側面や技術面を含め、成功するプロジェクトに必須なすべての課題に取り組んでいる。

6 参考文献

- Brooks87 F.P. Brooks, "No Silver Bullet: Essence and Accidents of Software Engineering," *IEEE Computer*, Vol. 20, No. 4, April 1987, pp. 10-19.
- Crosby79 P.B. Crosby, *Quality is Free*, McGraw-Hill, New York, NY, 1979.
- Curtis90 B. Curtis, "Managing the Real Leverage in Software Productivity and Quality," *American Programmer*, Vol. 3, No. 7, July 1990, pp. 4-14.
- Deming86 W. Edwards Deming, *Out of the Crisis*, MIT Center for Advanced Engineering Study, Cambridge, MA, 1986.
- DoD87 *Report of the Defense Science Board Task Force on Military Software*, Office of the Under Secretary of Defense for Acquisition, Washington, D.C., September 1987.
- Fagan86 M.E. Fagan, "Advances in Software Inspections," *IEEE Transactions on Software Engineering*, Vol. 12, No. 7, July, 1986, pp. 744-751.
- Fowler90 P. Fowler and S. Rifkin, *Software Engineering Process Group Guide*, Software Engineering Institute, CMU/SEI-90-TR-24, ADA235784, September, 1990.
- Freedman90 D.P. Freedman and G.M. Weinberg, *Handbook of Walkthroughs, Inspections, and Technical Reviews, Third Edition*, Dorset House, New York, NY, 1990.

参照文献

- Gabor90 A. Gabor, *The Man Who Discovered Quality*, Random House, New York, NY, 1990.
- GAO-92-48 *Embedded Computer Systems: Significant Software Problems on C-17 Must Be Addressed*, GAO/IMTEC-92-48, May 1992.
- Humphrey87a W.S. Humphrey, *Characterizing the Software Process: A Maturity Framework*, Software Engineering Institute, CMU/SEI-87-TR-11, ADA182895, June 1987.
- Humphrey87b W.S. Humphrey and W.L. Sweet, *A Method for Assessing the Software Engineering Capability of Contractors*, Software Engineering Institute, CMU/SEI-87-TR-23, ADA187320, September 1987.
- Humphrey88 W.S. Humphrey, "Characterizing the Software Process," *IEEE Software*, Vol. 5, No. 2, March, 1988, pp. 73-79.
- Humphrey89 W.S. Humphrey, *Managing the Software Process*, Addison-Wesley, Reading, MA, 1989.
- Humphrey91a W.S. Humphrey, D.H. Kitson, and J. Gale, "A Comparison of U.S. and Japanese Software Process Maturity," *Proceedings of the 13th International Conference on Software Engineering*, Austin, TX, 13-17 May 1991, pp. 38-49.

- Humphrey91b W.S. Humphrey, "Process Fitness and Fidelity,"
*Proceedings of the Seventh International Software Process
Workshop*, 16-18 October 1991.
- IEEE-STD-610 ANSI/IEEE Std 610.12-1990, "IEEE Standard Glossary of
Software Engineering Terminology," February 1991.
- Imai86 M. Imai, *Kaizen: The Key to Japan's Competitive Success*,
McGraw-Hill, New York, NY, 1986.
- Juran88 J.M. Juran, *Juran on Planning for Quality*, Macmillan, New
York, NY, 1988.
- Juran89 J.M. Juran, *Juran on Leadership for Quality*, The Free
Press, New York, NY, 1989.
- Kitson92 D.H. Kitson and S. Masters, *An Analysis of SEI Software
Process Assessment Results: 1987-1991*, Software
Engineering Institute, CMU/SEI-92-TR-24, July 1992.
- Paulk91 M.C. Paulk, B. Curtis, M.B. Chrissis, et al, *Capability
Maturity Model for Software*, Software Engineering
Institute, CMU/SEI-91-TR-24, ADA240603, August 1991.
- Paulk93a M.C. Paulk, B. Curtis, M.B. Chrissis, and C.V. Weber,
Capability Maturity Model for Software, Version 1.1,
Software Engineering Institute, CMU/SEI-93-TR-24,
February 1993.

参考文献

- Paulk93b M.C. Paulk, C.V. Weber, S. Garcia, M.B. Chrissis, and M. Bush, *Key Practices of the Capability Maturity Model, Version 1.1*, Software Engineering Institute, CMU/SEI-93-TR-25, February 1993.
- Radice85 R.A. Radice, J.T. Harding, P.E. Munnis, and R.W. Phillips, "A Programming Process Study," *IBM Systems Journal*, Vol. 24, No.2, 1985.
- Siegel90 J.A.L. Siegel, et al., *National Software Capacity: Near-Term Study*, Software Engineering Institute, CMU/SEI-90-TR-12, ADA226694, May 1990.
- Weber91 C.V. Weber, M.C. Paulk, C.J. Wise, and J.V. Withey, *Key Practices of the Capability Maturity Model*, Software Engineering Institute, CMU/SEI-91-TR-25, ADA240604, August 1991.

付録 A：キープロセスエリアのゴール

各キープロセスエリアのゴールを成熟度レベル毎に以下に示す。

A.1 レベル2のキープロセスエリア：反復できる段階

要件管理

- ゴール1 ソフトウェアのエンジニアリングと管理に使用するベースラインを確立するため、「ソフトウェアに割り当てられたシステム要件」が制御されている。
- ゴール2 ソフトウェアの計画、成果物、および活動が、「ソフトウェアに割り当てられたシステム要件」と首尾一貫した状態に保たれている。

ソフトウェアプロジェクト計画

- ゴール1 ソフトウェアプロジェクトの計画と進捗確認に使用するため、ソフトウェア見積りが文書化されている。
- ゴール2 ソフトウェアプロジェクトの活動とコミットメントが、計画され文書化されている。
- ゴール3 影響を受けるグループおよび個人が、ソフトウェアプロジェクトに関連する各自のコミットメントに合意している。

ソフトウェアプロジェクト進捗管理

- ゴール1 ソフトウェア計画に照らして、実際の結果と行動の進捗が確認されている。
- ゴール2 実際の結果と行動がソフトウェア計画と著しく乖離する場合には、是正処置がとられ、決着まで管理されている。
- ゴール3 ソフトウェアに関するコミットメントの変更は、影響を受けるグループおよび個人によって合意されている。

キープロセスエリアのゴール

ソフトウェア外注管理

- ゴール1 外部発注元は、適格なソフトウェア外注先を選定している。
- ゴール2 外部発注元とソフトウェア外注先は、双方のコミットメントについて合意している。
- ゴール3 外部発注元とソフトウェア外注先は、継続的に連絡を取り合っている。
- ゴール4 外部発注元は、ソフトウェア外注先のコミットメントに照らし、その実際の結果と行動の進捗を確認している。

ソフトウェア品質保証

- ゴール1 ソフトウェア品質保証活動が計画されている。
- ゴール2 適用される標準、手順、および要件に対して、ソフトウェア成果物と活動の忠実性が客観的に検証されている。
- ゴール3 ソフトウェア品質保証の活動および結果が、影響を受けるグループと個人に対して通知されている。
- ゴール4 ソフトウェアプロジェクト内部で解決できない非遵守に関する課題が、上級管理層によって取り上げられている。

ソフトウェア構成管理

- ゴール1 ソフトウェア構成管理活動が計画されている。
- ゴール2 選択されたソフトウェア作業成果物が特定され、制御され、そして利用可能である。
- ゴール3 特定されたソフトウェア作業成果物に対する変更が制御されている。
- ゴール4 ソフトウェアベースラインの状況と内容が、影響を受けるグループと個人に通知されている。

A.2 レベル3のキープロセスエリア：定義された段階

組織プロセス重視

- ゴール1 ソフトウェアプロセスの開発活動と改善活動が組織横断的に調整されている。
- ゴール2 プロセス標準に比較して、使用されているソフトウェアプロセスの強みと弱みが特定されている。
- ゴール3 組織レベルでのプロセスの開発活動と改善活動が計画されている。

組織プロセス定義

- ゴール1 組織のための標準ソフトウェアプロセスが開発され維持されている。
- ゴール2 ソフトウェアプロジェクトによる「組織の標準ソフトウェアプロセス」の使用に関する情報が、収集され、レビューされ、そして利用可能になっている。

トレーニングプログラム

- ゴール1 トレーニング活動が計画されている。
- ゴール2 ソフトウェアの管理と技術の役割を遂行するために必要なスキルと知識を身につけるトレーニングが提供されている。
- ゴール3 ソフトウェアエンジニアリンググループとソフトウェア関連グループの個人は、自らの役割を遂行するために必須なトレーニングを受けている。

ソフトウェア統合管理

- ゴール1 「プロジェクトの定義されたソフトウェアプロセス」は、「組織の標準ソフトウェアプロセス」をテラリングしたものである。

キープロセスエリアのゴール

ゴール2 「プロジェクトの定義されたソフトウェアプロセス」に従って、プロジェクトが計画され管理されている。

ソフトウェアプロダクトエンジニアリング

ゴール1 ソフトウェアを作成するために、ソフトウェアエンジニアリングのタスクが、定義され、統合され、そして首尾一貫して実施されている。

ゴール2 ソフトウェア作業成果物は、相互に首尾一貫性が保たれている。

グループ間調整

ゴール1 顧客の要件は、影響を受けるすべてのグループによって合意されている。

ゴール2 エンジニアリンググループ間のコミットメントは、影響を受けるグループによって合意されている。

ゴール3 エンジニアリンググループは、グループ間の課題を特定し、進捗を確認し、そして解決する。

ピアレビュー

ゴール1 ピアレビュー活動が計画されている。

ゴール2 ソフトウェア作業成果物の欠陥は、特定され取り除かれている。

A.3 レベル4のキープロセスエリア：管理された段階

定量的プロセス管理

ゴール1 定量的プロセス管理の活動が計画されている。

ゴール2 「プロジェクトの定義されたソフトウェアプロセス」のプロセス実績が、定量的に制御されている。

ゴール3 「組織の標準ソフトウェアプロセス」のプロセス能力が、定量的な表現で理解されている。

ソフトウェア品質管理

ゴール1 プロジェクトのソフトウェア品質管理の活動が計画されている。

ゴール2 ソフトウェア成果物の計測可能な品質目標とその優先順位が定義されている。

ゴール3 ソフトウェア成果物の品質目標を達成するまでの実際の進捗が、定量化され管理されている。

A.4 レベル5のキープロセスエリア：最適化する段階

欠陥予防

ゴール1 欠陥予防活動が計画されている。

ゴール2 欠陥の共通原因が探し出され、特定されている。

ゴール3 欠陥の共通原因に優先順位が付けられ、系統的に除去されている。

技術変更管理

ゴール1 技術変更の導入が計画されている。

ゴール2 新技術は、品質と生産性に対する効果を判断するために評価されている。

ゴール3 適切な新技術が、正規のプラクティスの中へ組織横断的に移転されている。

プロセス変更管理

ゴール1 継続的なプロセス改善が計画されている。

キープロセスエリアのゴール

- ゴール2 組織のソフトウェアプロセス改善活動に対して、組織全体が参加している。
- ゴール3 「組織の標準ソフトウェアプロセス」と「プロジェクトの定義されたソフトウェアプロセス」は、継続的に改善されている。

CMM[R] v1.1 公式日本語(SEA)版の変更要求手続き

CMU/SEIとの合意の元に、SEA-SPIN CMM[R] v1.1 (TR24/25) 公式日本語版への変更要求は、

- (1)このページを通じてのみ受け付け、SEA-CMMグループが取りまとめを行います。
- (2)日本語版特有の問題は、SEA-CMMグループが処置します。
- (3)CMM[R]そのものに対する問題は、SEAからCMU/SEIに回付します。

- SEA-SPIN 公式日本語版 変更要求 - ソフトウェア能力成熟度モデル 1.1版

成果物:CMM[R] v1.1 (SEI-93-TR-24およびSEI-93-TR-25のSEA公式日本語版の双方)

SEA-CMM 受付番号: _____ (SEA-SPIN-CMM 変更管理グループにて採番)

提出組織名: _____
提出者氏名: _____ 連絡先TEL番号: _____
電子メール: _____ 連絡先FAX番号: _____

提出日: _____ 課題: _____

変更箇所: ☐ TR24 ☐ TR25

章節番号: _____ キープロセスエリア: _____
プラクティスおよび番号、他: _____
図番号: _____、表番号: _____

変更提案:

変更理由:

送信

取りやめ

上記項目について洩れなく記入した後、「送信」ボタンをクリックして下さい。
入力文字としては、半角カタカナは扱えません。全角にて入力ください。
変更要求は、変更管理グループにてまとめて審議し、公開PDF版に反映させます。原則として、個別連絡はいたしません。場合によっては変更要求の疑問点について、変更管理グループからの質問メールが届くこともあるかと思えます。その時には、なにとぞご協力くださるようお願いいたします。

[R] Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office.

この変更要求ページへの質問、コメント、提言などは次の宛先にお送り下さい。 cmm-cr-request@sea.or.jp
[SEA-SPIN Page](#) に行く。 || [SEA Home Page](#) に行く。

-- メモ --