

技術報告書
1993年2月
CMU/SEI-93-TR-25
ESC-TR-93-178

能力成熟度モデルの キープラクティス 1.1版



Mark C. Paulk
Charles V. Weber
Suzanne M. Garcia
Mary Beth Chrissis
Marilyn Bush

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.ijnet.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 Key Practices of the Capability Maturity Model[®], Version 1.1, and believe it to be an accurate and faithful translation of the original English edition.

Sincerely,

A handwritten signature in cursive script that reads "M. Takahashi".

Vice Chairperson
Software Engineers Association
Mitsuhiro Takahashi

A handwritten signature in cursive script that reads "Tomoo Matsubara".

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能力成熟度モデル[®]のキープラクティス 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

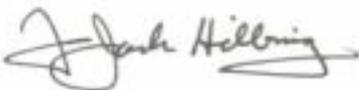
検証者
松原友夫,
IEEE Software Soapbox 編集者

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 Key Practices of the Capability Maturity Model[®], 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

**CENTER FOR INFORMATION
SYSTEMS ENGINEERING**

**Carnegie Mellon
Research Institute**

4516 Henry Street, Suite 205
Pittsburgh PA 15213-3729
Voice: 412-268-1700
FAX: 412-268-6369
www.cise.cmu.edu

著作物の複製と再配布の許諾

カーネギーメロン大学(CMU)が公開している能力成熟度モデル[®]のキープラクティス 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

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

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くありました。

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」現象ともいえる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すべてを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を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。

未筆ながら、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	0-vii
概要	
1 キープラクティス文書について	0-1
1.1 読者へ	0-1
1.2 本書と他の文献との関係	0-3
1.3 本書の構成	0-4
1.4 本書の使い方	0-5
2 能力成熟度モデルの概観	0-7
2.1 能力成熟度モデルの紹介	0-7
2.2 CMMの起源	0-7
2.3 CMMの構造	0-8
2.4 CMMの成熟度レベルの定義	0-12
2.4.1 レベル1 – 初期レベル	0-13
2.4.2 レベル2 – 反復できるレベル	0-14
2.4.3 レベル3 – 定義されたレベル	0-15
2.4.4 レベル4 – 管理されたレベル	0-16
2.4.5 レベル5 – 最適化するレベル	0-17
2.5 CMMのキープロセスエリア	0-18
2.6 キープラクティス	0-26
2.7 ゴール	0-27
2.8 コモンフィーチャ	0-27
3 キープラクティスページの使い方	0-31
4 CMMの解釈	0-35
4.1 キープラクティスの解釈	0-35
4.2 コモンフィーチャの解釈	0-35
4.2.1 実施のコミットメント	0-36
4.2.2 実施能力	0-37
4.2.3 実施される活動	0-39
4.2.4 計測と分析	0-47
4.2.5 履行検証	0-47
4.3 ソフトウェアプロセス定義の解釈	0-50
4.3.1 プロセス定義のコンセプト	0-51
4.3.2 「組織のソフトウェアプロセス資産」に関するコンセプト	0-53
4.3.3 「プロジェクトの定義されたソフトウェアプロセス」 に関するコンセプト	0-58

目次

4.3.4 「プロジェクトの定義されたソフトウェアプロセス」と ソフトウェア開発計画の関係	0-61
4.3.5 ライフサイクルとCMM	0-61
4.3.6 技術とCMM	0-62
4.3.7 文書化とCMM	0-62
4.3.8 プロセスデータの収集と分析	0-63
4.4 組織における構造と役割	0-64
4.4.1 組織における役割	0-64
4.4.2 組織の構造	0-69
4.4.3 独立性と組織の構造	0-73
4.5 専門家の判定の適用	0-74

レベル2のキープロセスエリア

要件管理	L2-1
ソフトウェアプロジェクト計画	L2-11
ソフトウェアプロジェクト進捗管理	L2-31
ソフトウェア外注管理	L2-45
ソフトウェア品質保証	L2-63
ソフトウェア構成管理	L2-75

レベル3のキープロセスエリア

組織プロセス重視	L3-1
組織プロセス定義	L3-11
トレーニングプログラム	L3-25
ソフトウェア統合管理	L3-37
ソフトウェアプロダクトエンジニアリング	L3-59
グループ間調整	L3-85
ピアレビュー	L3-97

レベル4のキープロセスエリア

定量的プロセス管理	L4-1
ソフトウェア品質管理	L4-19

レベル5のキープロセスエリア

欠陥予防	L5-1
技術変更管理	L5-19
プロセス変更管理	L5-33

付録

付録A: 参照文献	A-1
付録B: 用語集	A-3
付録C: キープラクティス抄録	A-27
付録D: 変更履歴	A-67
付録E: 索引	A-73

目次

図の一覧

図2.1	能力成熟度モデルの構造	0-9
図2.2	ソフトウェアプロセス成熟度の5段階	0-13
図2.3	成熟度レベル毎のキープロセスエリア	0-19
図2.4	キープロセスエリアのプロセス区分	0-25
図3.1	キープラクティスの記載例	0-33
図4.1	CMMで使用される概念的なソフトウェアプロセスの枠組み	0-52

図の一覧

謝辞

ソフトウェアエンジニアリング研究所(SEI)は、本文書の作成に取り組まれた多くの方々に感謝したい。1988年以来、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、および産業界と政府からのフィードバックによって得られた知識に基づいて、ソフトウェア能力成熟度モデル(CMM)は、開発され修正され続けてきた。本文書は、1991年8月にリリースされたCMM v1.0の改訂版の一部として作成された。

能力成熟度モデルの概念的枠組みは、Watts Humphreyが開発した。我々がこのモデルを洗練し、プラクティスを発展させるに際して、同氏から助言を頂いた。

本文書が現在の形になるまでには、多くの草稿が作成された。このモデルのためにアイデアを出し、プラクティスをまとめることに多くの時間を提供して下さった方々に感謝したい。特に、Charlie Weber, Mark Paulk, Cynthia Wise, Jim Withey, Mary Beth Chrissis, Suzanne Garcia, およびMarilyn Bushの貢献に感謝する。

その他にも多くの方々からコメント、批判、提案を寄せて頂いた。Ed Averill, Judy Bamberger, Anita Carleton, Bill Curtis, Susan Dart, Lionel Deimel, Peter Feiler, Julia Gale, Jim Hart, Ron Higuera, Purvis Jackson, Tim Kasse, David Kitson, Mike Konrad, Peter Malpass, Mark Manduke, Judah Mogilensky, Warren Moseley, Jim Over, George Pandelios, Bob Park, Jeff Perdue, Dick Phillips, Mike Rissman, Jim Rozum, Christer von Schantzらに感謝する。

CMMの草稿をレビューし、見識溢れるコメントや推奨を提供することに時間と努力を惜しまなかったCMM Correspondence Groupメンバの皆様には、深く感謝する。その中でも特に、1990年3月と1992年4月のCMMワークショップにご参加頂いた方々、またその他の機会を捉えてコメントや推奨を寄せて頂いた方々に心から感謝する。これら実務者の経験なくしては、このモデルを洗練するという今回の取り組みは達成されなかったであろう。

我々に指針を示して下さったCMM Advisory Boardの皆様にも感謝したい。技術的な洞察に加えて、取り組みの焦点を絞るに際して助力頂き、また、産業界や政府のレビューから寄せられた多くのコメントを処理するための処置を評価

謝辞

し計画する際にもご協力頂いた。現在のメンバは、Constance Ahara, Kelley Butler, Bill Curtis, Conrad Czaplicki, Raymond Dion, Judah Mogilensky, Martin Owens, Mark Paulk, Sue Stetak, Charlie Weber, およびRonald Willisの各氏である。CMM v1.0については、Harry Carl, Jim Hess, Jerry Pixton, および Jim Witheyの各氏らにもご協力頂いた。

最後に本文書の作成にご協力頂いた多くの方々に感謝したい。承認プロセスを促進して頂いたJoint Program Officeメンバの方々に感謝する。本文書の編集、デザインにご協力頂いたAmerican Institute for ResearchのGinny RedishとRenee Dutkowskiにも感謝したい。管理面でご支援頂いたCarolyn Tady, David White, およびDebbie Punjack、編集、フォーマット、その他本文書の作成に必要なあらゆる面でご支援頂いたMary Beth Chrissis, Suzanne Couturiaux, およびMike Konradのご努力を高く評価する。

1 キープラクティス文書について

1.1 読者へ

信頼性と実用性が高いソフトウェアを、スケジュール通りかつ予算通りに開発することは、多くの組織にとって困難な取り組みである。また、納期が遅れたり、予算を超過したり、期待通りのものができなかつたりすると、その組織の顧客にとっての問題も引き起こす。このことは、ソフトウェアプロジェクトの規模が年々大きくなり、重要性が増すにつれて、深刻な問題になっている。効果的なソフトウェアのエンジニアリングと管理の実践のために、プロセス基盤を構築することに集中的かつ継続的に努力することで、これらの問題は克服され得る。

プロセス基盤を構築するためには、ソフトウェアを開発する組織が、そのソフトウェアプロセスを実施する能力を評価する方法が必要になる。また、それらの組織がプロセス能力を向上するためのガイダンスも必要である。米国国防総省(DoD)などの顧客にとっても、ある組織がソフトウェアエンジニアリング契約を実施する場合に、その組織の能力を評価するためのより効果的な方法が必要である。外部発注元にとっても、取り引きする可能性のある外注先の能力を評価する方法が必要である。

ソフトウェア開発組織と、DoDや外部発注元などの顧客側の両方を支援するために、ソフトウェアエンジニアリング研究所(SEI)は、ソフトウェア能力成熟度モデル(CMM)を開発した。CMMは、成熟し能力のあるソフトウェアプロセスの特徴を記述している。またCMMでは、未熟で反復不可能なソフトウェアプロセスから、成熟し整った形で管理されたソフトウェアプロセスへの発達を、成熟度レベルという言葉で記述している。

CMMは以下の用途に使用できる：

- ソフトウェアプロセス改善。組織が、そのソフトウェアプロセスに対する変更を計画し、具体化し、そして実施する。

キープラクティス文書について

- ソフトウェアプロセスアセスメント。トレーニングを受けたソフトウェア専門家チームが、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。

- ソフトウェア能力審査。トレーニングを受けたソフトウェア専門家チームが、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。

本書では、CMMの各成熟度レベルに対応するキープラクティスを記述している。本書は、CMMの各レベルにおける成熟度が意味するところを詳細に示しており、ソフトウェアプロセス改善、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に際して使用できる指針である。

CMMのキープラクティスは、政府から大型の契約を受注する組織において正規のプラクティスと期待される事項で表現されている。CMMを適用するいかなる状況においても、プラクティスをどのように適用するかについて、合理的な解釈が加えられるべきである。CMMを解釈するガイドラインは、本書の第4章に盛り込まれている。大規模な契約を受注する組織とは事業環境が大幅に異なる組織にCMMを適用する場合、CMMは適切に解釈されなければならない。十分な知識に基づいてCMMを利用するためには、専門家の判定の役割を認識しなければならない。

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
- 成熟度質問票などプロセス成果物を開発す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

1.2 本書と他の文献との関係

以下のふたつの文献は、CMMの基礎を築いた：

- 『Characterizing the Software Process』 [Humphrey88]
- 『*Managing the Software Process*』 [Humphrey89]

CMMバージョン1.0は、1991年8月に以下のふたつの技術報告書としてリリースされた：

- 『Capability Maturity Model for Software』 [Paulk91]
- 『Key Practices of the Capability Maturity Model』 [Weber91]

キープラクティス文書について

この初期バージョンは1992年に改訂された。CMMの最新版を理解し利用するには、以下のふたつの文献が必要である：

- 『ソフトウェア能力成熟度モデル1.1版』 [Paulk93a]
- 本文書『能力成熟度モデルのキープラクティス1.1版』 [Paulk93b]

『ソフトウェア能力成熟度モデル1.1版』には、CMMの入門的解説、5つの成熟度レベルの記述、CMMの利用上の定義とその構造、組織が成熟度モデルをどのように利用できるかについての議論、およびCMMの将来動向が盛り込まれている。

『能力成熟度モデルのキープラクティス1.1版』には、CMMの各成熟度レベル毎のキープロセスエリアに対応するキープラクティスと、キープラクティスの解釈を補助する情報が含まれている。

成熟度質問票などのプロセス成果物は、CMMのキープラクティスから導き出されたものである。SEIのプロセス成果物で、ソフトウェアプロセス改善、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支援するものには、トレーニングコース、ハンドブック、現地訪問ガイドなどがある。

1.3 本書の構成

この第1章は、CMMと本書の概観である。第2章から第4章の概観は以下の通りである：

- CMMとその構成要素の概観
- キープラクティスフォーマットの使い方の記述
- キープラクティスの使い方と解釈の記述

概観に続いて、CMMのキープロセスエリアに対応するキープラクティスを記述する。実際に適用する際に必要な厳密さを省略して、キープラクティスについて素早く雰囲気を知りたいという読者は、付録Cのキープラクティス抄録を参照されたい。

付録として、本書で引用した参照文献、用語集、キープラクティス抄録、変更履歴、および索引を添付する。

1.4 本書の使い方

CMMについてよく知らない読者は、キープラクティスを使用しようとする前に、まず『ソフトウェア能力成熟度モデル1.1版』[Paulk93a]と、本書の第1章から第4章すべてに目を通すべきである。

CMMとその構造についてすでに知識をお持ちの読者は、キープラクティスの解釈法について説明した第4章から始めても構わない。

キープラクティス文書について

2 能力成熟度モデルの概観

2.1 能力成熟度モデルの紹介

ソフトウェア能力成熟度モデル(CMM)は、効果的なソフトウェアプロセスの主要要素を記述する枠組みである。CMMは場当たりの未熟なプロセスから、成熟し秩序あるプロセスへと進化する改善経路を記述している。

CMMはソフトウェアの開発と保守の計画、エンジニアリング、および管理に関するプラクティスを取り扱う。これらのキープラクティスに従うことによって、コスト、スケジュール、機能充足性、および成果物品質の目標達成に対する組織の能力が改善される。

CMMは基準を確立し、その基準によって反復可能な方法で組織のソフトウェアプロセスの成熟度を判定できるようにし、そしてその成熟度と産業界の実践の状態とを比較できるようにする[Kitson92]。CMMは、組織がそのソフトウェアプロセス改善を計画するためにも利用できる。

2.2 CMMの起源

ソフトウェアエンジニアリング研究所(SEI)は、政府の要求とMITRE社の支援のもとで、成熟度モデルおよび成熟度質問票の初版を開発した。モデルおよび質問票の開発において、SEIは、ソフトウェアプロセスの開発と改善に関与している実務者からの助言を取り入れた。我々の目標は以下のようなモデルを提供することであった：

- 現実の実践に基づく
- 最良の実践状態を反映する
- ソフトウェアプロセス改善、ソフトウェアプロセスアセスメント、またはソフトウェア能力審査を実施している人々のニーズを反映する

能力成熟度モデルの概観

- 文書化されている
- 広く一般に利用可能である

初期バージョンの成熟度モデル以来、ソフトウェアプロセス成熟度に対する新たな知識と見識が獲得されてきた。この見識は以下によって獲得されたものである：

- ソフトウェア関係以外の組織の研究
- 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実施および観察
- モデルに対する変更要求の奨励と分析
- 産業界や政府の代表者との会議やワークショップへの参加
- 産業界や政府のレビューからのフィードバックの奨励

これらの新たに獲得された知識を利用して、能力成熟度モデルとそのプラクティスが改訂され、CMM v1.1ができた。

2.3 CMMの構造

CMMは5つの成熟度レベルで構成されている。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つのまとまりで構成されている。コモンフィーチャはキープラクティスの特徴づけるもので、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。CMMのこの構造を図2.1に示す。

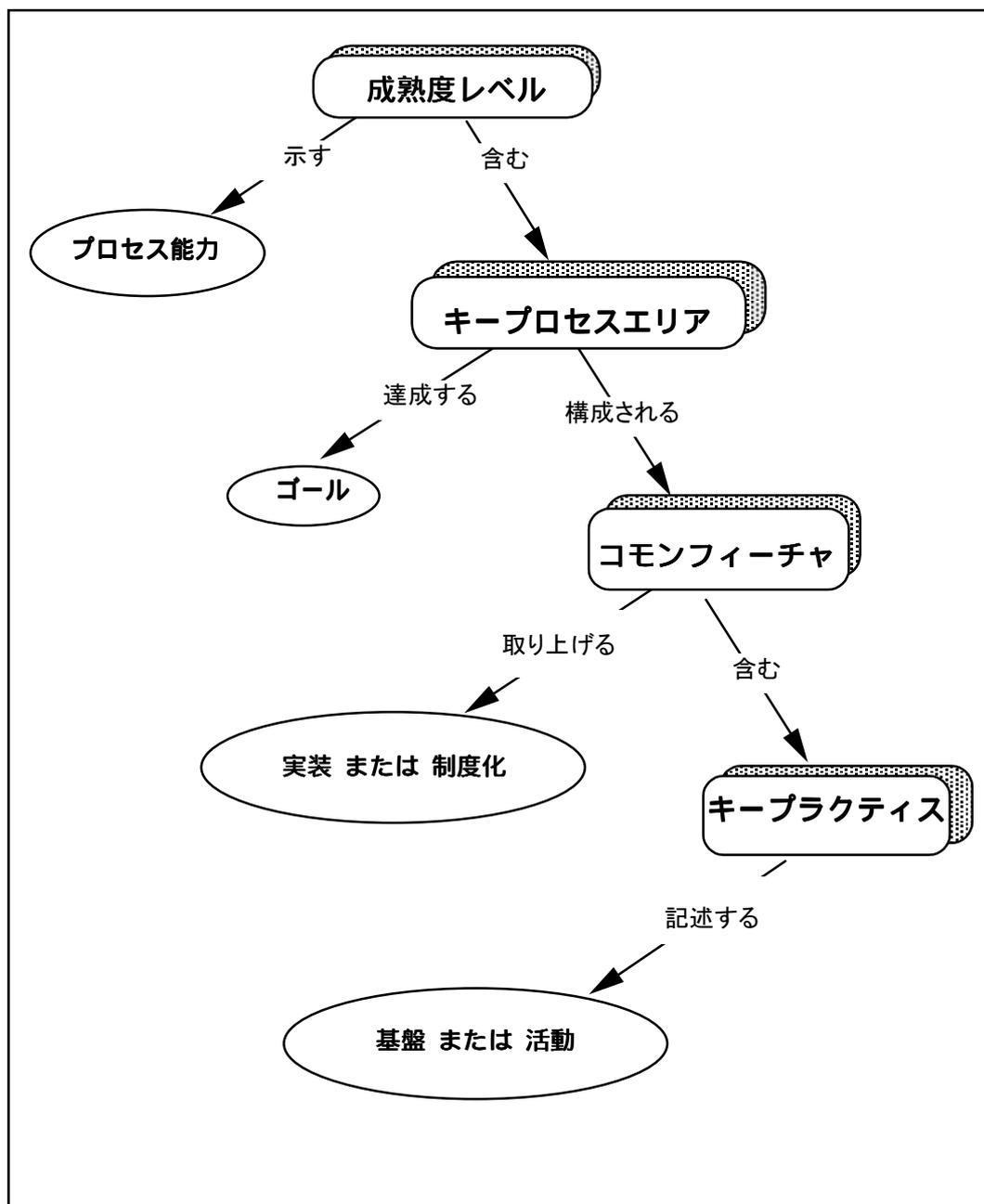


図2.1 能力成熟度モデルの構造

能力成熟度モデルの概観

CMMのコンポーネントは以下のものを含む：

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。5つの成熟度レベルはCMMのトップレベルの構造を提供している。
プロセス能力	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することが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。
キープロセスエリア	各成熟度レベルはキープロセスエリアで構成されている。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、その成熟度レベルでの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、ひとつの成熟度レベルに属するように定義されている。例えば、レベル2のキープロセスエリアのひとつとして、『ソフトウェアプロジェクト計画』がある。

ゴール

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。

ゴールの例を『ソフトウェアプロジェクト計画』キープロセスエリアで見れば、『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』がある。ゴールの解釈に関する詳しい情報については『ソフトウェア能力成熟度モデル1.1版』[Paulk93a]および本書の『4.5 専門家の判定の適用』を参照のこと。

コモんフィーチャ

キープラクティスは、『実施のコミットメント』、『実施能力』、『実施される活動』、『計測と分析』、および『履行検証』という5つのコモんフィーチャに分けられる。コモんフィーチャは、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。

『実施される活動』コモんフィーチャは実装活動を記述している。他の4つのコモんフィーチャは、制度化の要因を記述しており、それらの要因は、組織文化のプロセス部分を形成する。

能力成熟度モデルの概観

キープラクティス 各キープロセスエリアには、そのゴールの満足に寄与するキープラクティスが記述されている。『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最も寄与する基盤と活動を記述している。

例えば、『ソフトウェアプロジェクト計画』キープロセスエリアのプラクティスのひとつに『プロジェクトのソフトウェア開発計画は、文書化された手順に従って策定する。』というものがある。

2.4 CMM成熟度レベルの定義

ソフトウェア作業成果物を開発し保守するソフトウェアプロセスを、組織が確立し改善するにつれて、その組織は成熟度のレベルを高めていく。図2.2はCMMの5段階成熟度レベルを表している。

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。各キープロセスエリアはひとまとまりのゴールで構成されており、これらが満足されたとき、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。成熟度モデル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が制度化され、結果的に組織のプロセス能力全体が向上する。

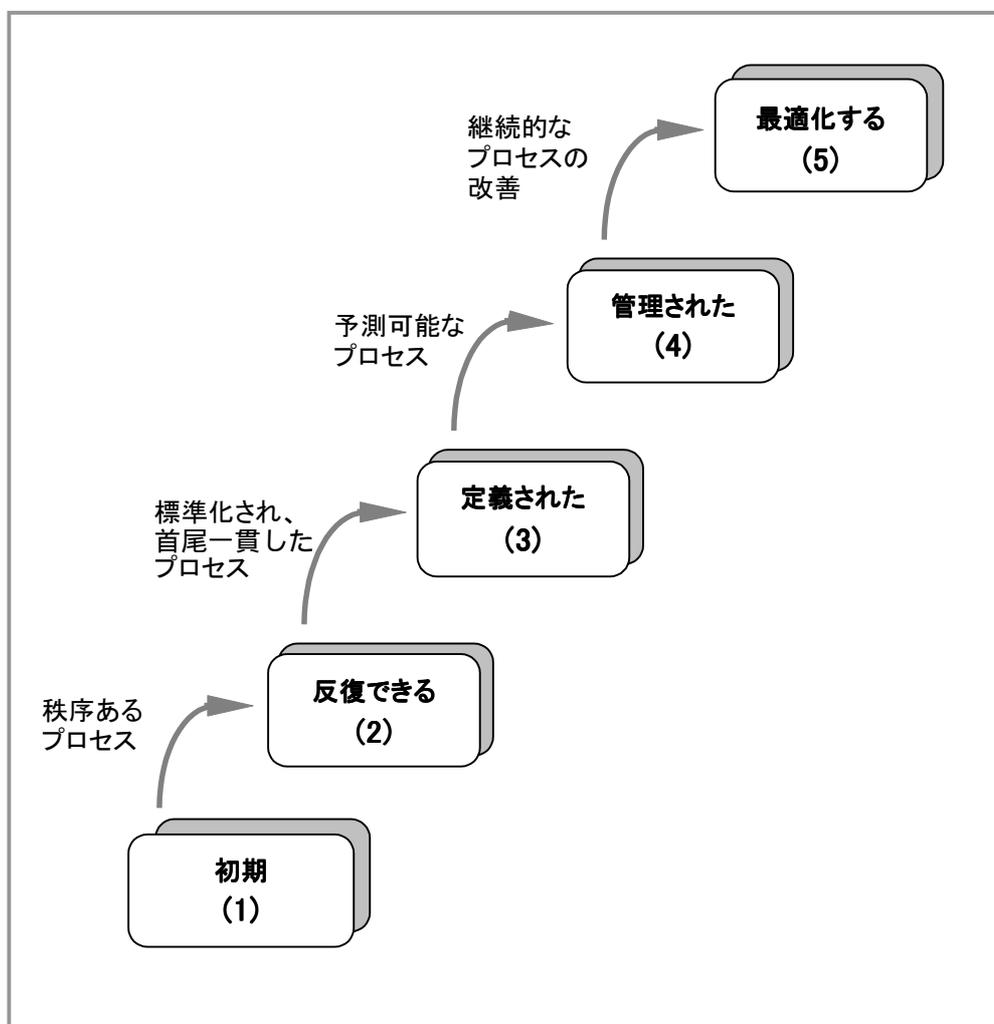


図2.2 ソフトウェアプロセス成熟度の5段階

2.4.1 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の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

能力成熟度モデルの概観

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、コーディングやテストを繰り返す。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。有能で実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プレッシャー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、たえまなく変更され、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。

2.4.2 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ってもよいが、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、秩序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。

2.4.3 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。レベル3で確立されるプロセスは、ソフトウェアマネージャと技術要員のより効率的な活動を支援するために利用される（そして適宜変更される）。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。

能力成熟度モデルの概観

プロジェクトは、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ために、「組織の標準ソフトウェアプロセス」をテーラリングする。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、検証メカニズム（ピアレビューのような）、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。

2.4.4 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、すべてのプロジェクトにとって重要なソフトウェアプロセス活動のために計測される。「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明らかになり、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。

2.4.5 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に重点をおく。組織は、プロセスの弱みと強みを先を見越して特定する手段を持っており、欠陥の発生を予防するゴールがある。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。ソフトウェアエンジニアリングのベストプラクティスを利用する革新は、組織全体で特定され、移転される。

レベル5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ている。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ている。

2.5 CMMのキープロセスエリア

図2.3は、CMMの各成熟度レベルに対するキープロセスエリアを列挙したものである。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、ひとつの成熟度レベルに属するように定義されている。キープロセスエリアは、組織がその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す構成要素である。キープロセスエリアは、その成熟度レベルを達成するために取り組まねばならない課題を特定している。

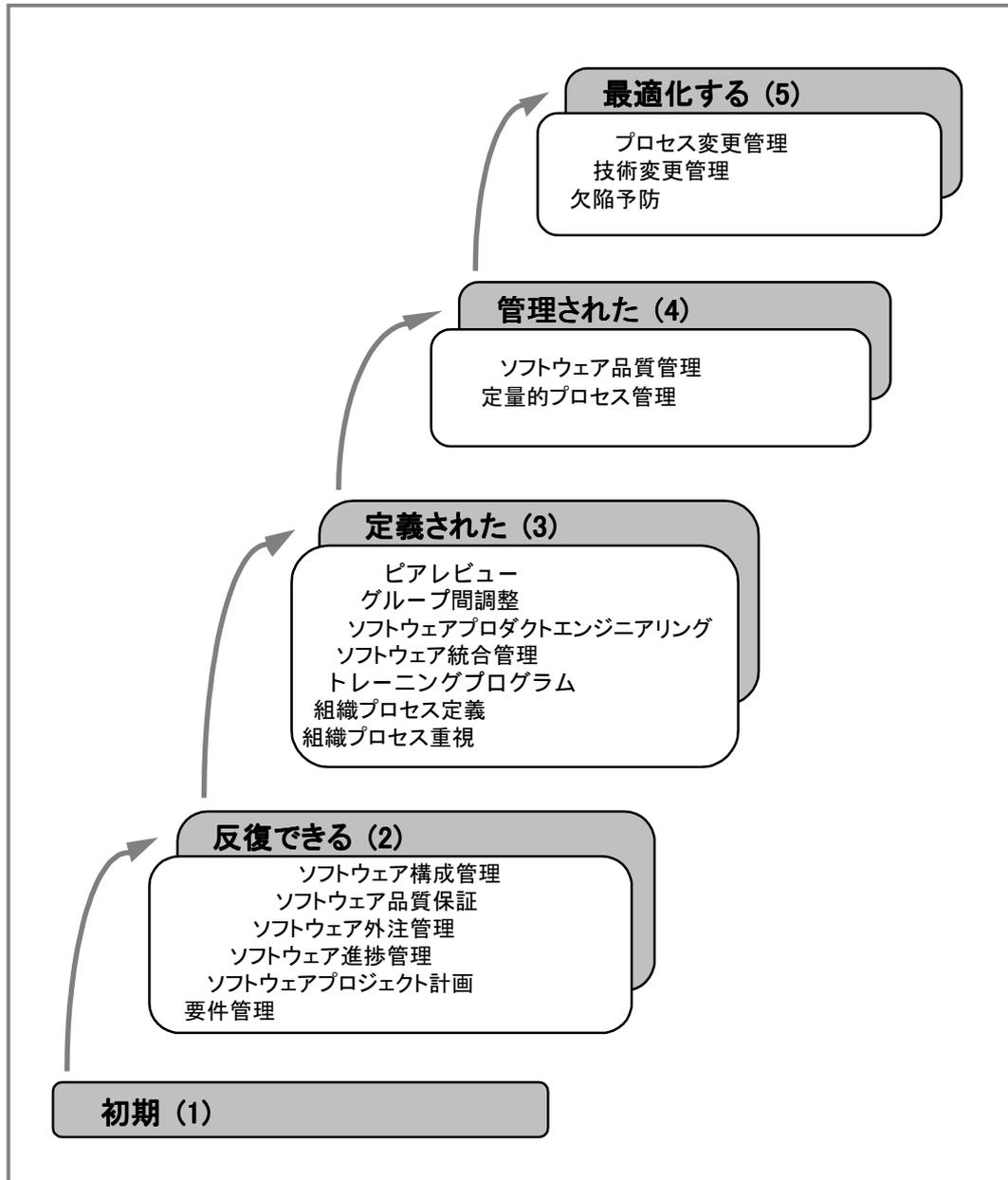


図2.3 成熟度レベル毎のキープロセスエリア

能力成熟度モデルの概観

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。

- 『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。

能力成熟度モデルの概観

- 『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：

能力成熟度モデルの概観

- 『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。

定義からして、キープロセスエリアは、ひとつの成熟度レベルのところで表現されている。しかし、キープロセスエリア間には関係があり、特定の管理や技術の領域における改善は、ひとつのキープロセスエリアに限定される必要はない。図2.4はこれらの関係を表している。組織は、下位レベルの成熟度レベルを達成する前に、上位レベルのキープロセスエリアに取り組んでも良いが、上位の成熟度レベルのキープロセスエリアを達成したとしても、下位レベルのキープロセスエリアに対して焦点を当て続けるように注意を払わなければならない。

プロセス区分 レベル	管理 ソフトウェアプロジェクト 計画、管理等	組織 上級管理層による レビュー等	エンジニアリング 要件分析、設計、 コーディング、テスト等
5 最適化する		技術変更管理	
		プロセス変更管理	欠陥予防
4 管理された	定量的プロセス管理		ソフトウェア品質管理
3 定義された	ソフトウェア統合管理	組織プロセス重視	ソフトウェアプロダクトエンジニアリング
	グループ間調整	組織プロセス定義 トレーニングプログラム	ピアレビュー
2 反復できる	要件管理 ソフトウェア プロジェクト計画 ソフトウェア進捗管理 ソフトウェア外注管理 ソフトウェア品質保証 ソフトウェア構成管理		
1 初期	場当たりのプロセス		

図 2.4 キープロセスエリアのプロセス区分

能力成熟度モデルの概観

キープロセスエリアは図 2.4 の通り、『管理』プロセス、『組織』プロセス、『エンジニアリング』プロセスの三つの大きな区分に分類される。『管理』プロセス区分は、プロジェクト管理活動を指しており、レベル 2 における計画や進捗確認が、レベル 3 では定義されたソフトウェアプロセスによる管理となり、レベル 4 では定量的管理となり、そしてレベル 5 では定常的に変化する環境における革新的な管理へと、進化していく。『組織』プロセス区分は、組織が成熟してきたときのプロジェクト間の責任を指しており、レベル 3 におけるプロセス課題の重視に始まり、レベル 4 ではプロセスの定量的管理を経て、レベル 5 における継続的プロセス管理という環境下において変化の管理にまで達する。『エンジニアリング』プロセス区分は、要件分析、設計、コーディング、およびテストといった技術的活動を指している。これらの活動はすべてのレベルで実施されるものであるが、レベル 3 における規律あるエンジニアリング、レベル 4 における統計的プロセス制御、そしてレベル 5 における継続的に計測される改善へと進化していく。

レベル 4 と 5 では、これらのプロセス区分に跨るキープロセスエリアが存在する。このことは、レベル 4 と 5 に対する理解が深まれば、CMM v2 において新しいキープロセスエリアが特定される潜在的可能性を示している。

2.6 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。

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ラクティス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定するために合理的に解釈されるべきものである。

2.7 ゴール

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。あるキープロセスエリアのキープラクティスを特定の状況下のプロジェクトに適用する際に、その適用の仕方がキープラクティスの合理的な一形態かどうかを決定するために、ゴールが利用できる。同様に、キープロセスエリアを実装する代替案をアセスメントし、または審査するとき、キープロセスエリアの意図を代替案が満足しているどうかを決定するために、ゴールが利用できる。ある組織においてゴールを解釈するための詳しい情報については、『ソフトウェア能力成熟度モデル1.1版』 [Paulk93a] と、本書の4.5『専門家の判定の適用』を参照されたい。

2.8 コモンフィーチャ

各キープロセスエリアのキープラクティスは、5つのコモンフィーチャによって構成されている。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。組織がキープラクティスを利用し易いように、コモンフィーチャによってキープラクティスはグループ化され、配列されている。5つのコモンフィーチャは下記の通りである：

能力成熟度モデルの概観

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、プロジェクトや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。プロジェクトや組織による『実施される活動』は、キープロセスエリアの具体的な実装を記述するので、キープラクティスの中でも最も大きな部分になっている。しかし、他のコモンフィーチャに属すキープラクティスも同様に重要であり、これらは、キープロセスエリアを支援し制度化するために何が必要とされないかを取り上げている。

3 キープラクティスページの使い方

キープラクティスは成熟度レベル別にまとめられ、レベル毎にタブページで区切られている。タブページには、成熟度レベルの記述、その成熟度レベルのキープロセスエリアの一覧、各キープロセスエリアの開始ページ番号が示される。

それぞれのキープロセスエリアでは、次の内容が示される：

- キープロセスエリアの簡単な記述
- キープロセスエリアのゴール
- キープラクティス

キープラクティスは、5つのコモンフィーチャ(『実施のコミットメント』、『実施能力』、『実施される活動』、『計測と分析』、『履行検証』)にグループ化されており、図3.1に示すキープラクティスの例のように、階層形式で表現される。キープラクティスは、次の項目を含む：

キープラクティス キープラクティスは、最上位のキープラクティスとも言われ、そのキープロセスエリアの基本方針、手順、活動を記述する。キープラクティスは太字で表記され、各コモンフィーチャ毎に通し番号が振られている。例えば、『実施される活動』コモンフィーチャの最初のキープラクティスは、『活動1』と表記される。

サブプラクティス サブプラクティスは、下位のキープラクティスとも言われ、トップレベルのキープラクティスの下に位置付けられ、トップレベルのキープラクティスに関して実装されると期待される事項を記述する。サブプラクティスは、キープラクティスが十分に実装されたかどうかを決定するためにも利用できる。

キープラクティスページの使い方

補足情報

補足情報は、キープラクティスの説明に続く括弧内の部分である。補足情報には、例、詳細説明、および他のキープロセスエリアへの参照が含まれる。

キープラクティスに続くサブプラクティスや補足情報が次ページにまたがる場合には、新しいページの先頭にキープラクティス番号を括弧付きで記して、そのページの情報が前ページのキープラクティスの続きであることを示す。

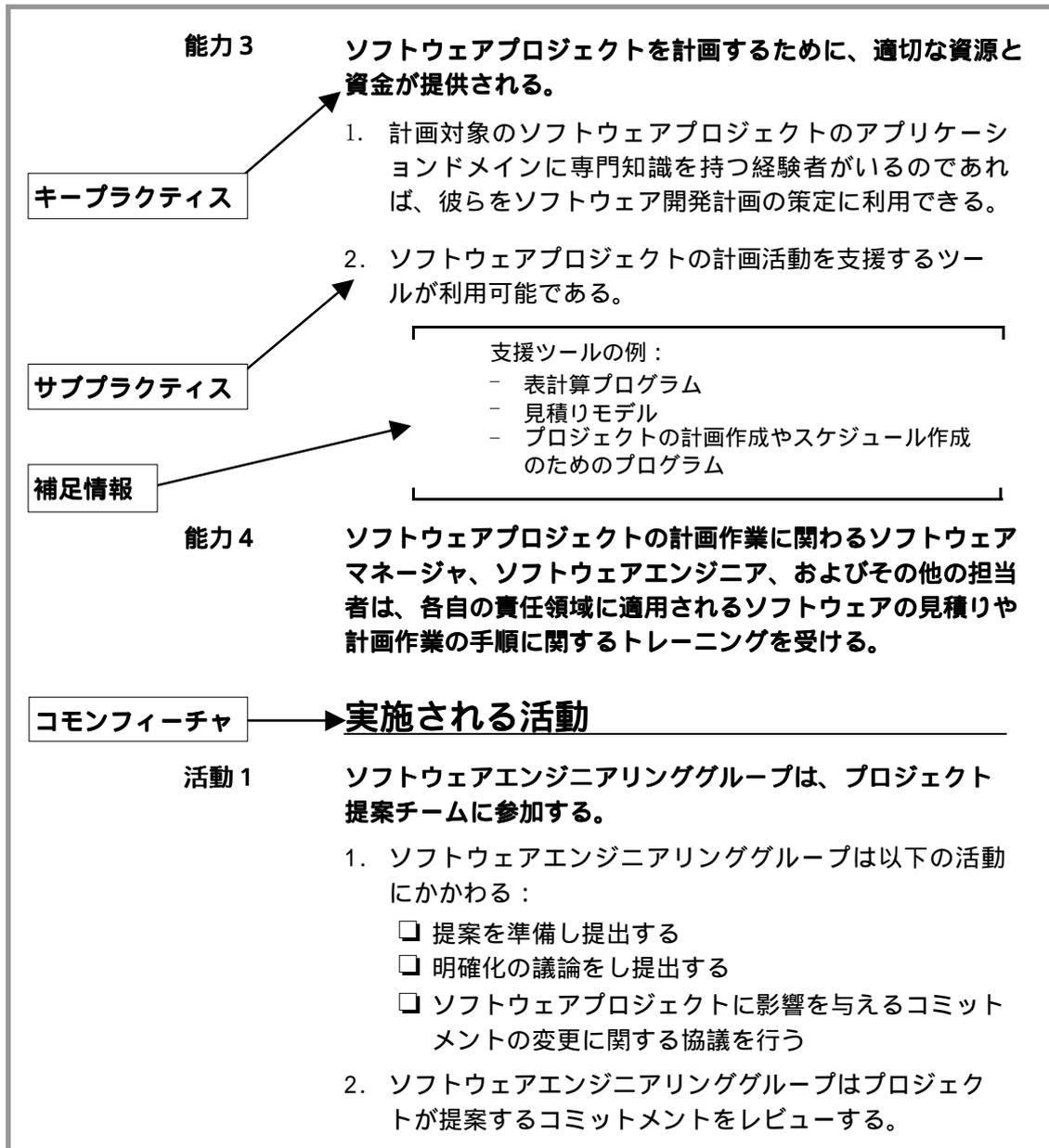


図3.1 キープラクティスの記載例

キープラクティスページの使い方

4 CMMの解釈

4.1 キープラクティスの解釈

キープラクティスを規定することは、特定のソフトウェアライフサイクルモデル、特定の組織構造、特定の責任分担、または開発に対する特定の管理や技術のアプローチを、要求したり支持したりすること意図していない。そうではなくて、効果的なソフトウェアプロセスに不可欠な要素を記述することを意図している。

キープラクティスは、様々なプロジェクトや組織に適用する原則、種々の典型的なソフトウェアアプリケーションにおいて妥当な原則、そしていつまでも妥当であり続ける原則を伝えるためのものである。したがって、ここでは原則を記述するにとどめ、組織の文化とマネージャや技術要員の経験によって、キープラクティスをどのように実装するかは、それぞれの組織に任せる。

キープラクティスは特定の実装方法には依存しないが、キープラクティスをより明解にするために特定の用語や例が首尾一貫して使用されている。この章では、役割、責任、関係、成果物、および活動に対して、CMMで使用されている慣例を記述する。キープラクティスを使用する組織は、これらの慣例をよく理解し、それぞれの組織、プロジェクト、および事業環境に合わせて適切に投影するべきである。

付録Bの用語集には用語が定義されており、本章や他の章で使用されている用語もその中に含まれている。

4.2 コモンフィーチャの解釈

キープラクティスの各コモンフィーチャにおいては、キープロセスエリア相互の連続性と一貫性を保つために、特定の慣用句や慣例が使われている。各コモンフィーチャ毎に構造上主要な慣例を以下に説明する。

4.2.1 実施のコミットメント

方針書

方針書が使用される場合、一般的に方針書は、プロジェクトがそのキープロセスエリアのプラクティスについて明文化された組織方針に従うことに言及する。このことは、組織のコミットメントと、実際に作業を実施するプロジェクトとの結びつきを強調するためである。

方針書に関するサブプラクティスは、そのキープロセスエリアで取り扱う活動であって、特に明文化された方針による制度化に適した活動を要約している。

いくつかのキープロセスエリア(例えば、『組織プロセス重視』)においては、キープロセスエリアの活動の焦点は組織であり、プロジェクトではない。このような場合、方針書は言い換えられており、組織が明文化された方針に従うことに言及する。

リーダーシップ

いくつかのキープロセスエリアでは、リーダーシップの役割(例えば、プロジェクトソフトウェアマネージャ)の割り当てや、キープロセスエリアが成功裡に制度化されるために特に必要な主催者としての態度を示す活動が、『実施のコミットメント』として記述されている。

4.2.2 実施能力

資源と資金

ほとんどのキープロセスエリアには、キープロセスエリアで扱われる活動に対する十分な資源と資金の必要性を反映するキープラクティスが含まれる。サブプラクティスによって記述されたこれらの資源と資金は、一般に三つのカテゴリーに分類される。それは、特別な技能、十分な資金、およびツールの入手である。キープロセスエリアの活動を実施するのに役に立つであろうツールが例として列挙されている。

ここで、『予算』ではなく『資金』を使用しているのは、約束された予算より、実際に配分され使用される資金の方が実際のプロセスにとって関係があることを強調するためである。

トレーニング

CMMでは、トレーニングという用語は、この用語が使われる場合に通常想定されているよりもやや広い意味で使われている。トレーニングとは、専門的な教育と実践によって各個人を熟練させることである。ここでいうトレーニングには、組織内の各個人へ技能や知識を移転するものであれば、公式な手段と同様、非公式な手段も含まれている。教室形式のトレーニングは多くの組織が従業員に技能を身に付けさせるために使用する一般的な方法であるが、CMMには、例えば教育ビデオ、コンピュータ支援による教育、公式なメンタリングや実習のプログラムも含まれている。『トレーニングプログラム』キープロセスエリアは、これらのトレーニング手段に関する具体的なキープラクティスを記述している。

CMMでは、トレーニングを表現するのに一般的にふたつのテンプレートを用いる。レベル2では『トレーニングを受ける』という表現が使用される。レベル3以上では『必修トレーニングを受ける』という表現が使用される。レベル2でのトレーニングは組織全体としての制度化が不十分であることを区別するために、このように異なるテンプレートを用いている。レベル3以上では、『トレーニングプログラム』キープロセスエリアのキープラクティスが、組織のトレーニング活動を統御していると想定している。

すべてのキープロセスエリアにおいて、トレーニングの可能性のある題材は参考例として表現されている。これは、組織の状況の違いに応じて、異なった特定のトレーニングが必要とされるであろうことを示している。

オリエンテーション いくつかのキープロセスエリアには、オリエンテーションについて記述したキープラクティスがある。オリエンテーションという言葉は、移転される技能や知識の程度がトレーニングを通して期待されているよりも浅いことを示すのに広く使われている。オリエンテーションは、そのトピックエリアの実施担当者を監督する人々、または実施担当者と接する人々のために、トピックの概要や紹介を提供するものである。

前提要件 いくつかのキープロセスエリアには、前提要件の必要を表すキープラクティスが含まれている。例えば、ソフトウェア開発計画は『ソフトウェアプロジェクト進捗管理』の前提要件である。別のキープロセスエリアの活動のアウトプットとして期待されているものが前提要件となる場合もある。また、ソフトウェアプロジェクトの範囲の外から得られると期待されていることが前提要件となる場合もある(例えば、「ソフトウェアに割り当てられたシステム要件」は、『要件管理』の前提要件である)。

『キー』プラクティスを強調するというCMMの考え方によると、すべての前提要件がそれぞれのキープロセスエリアに対して列挙されているとは限らない。キープロセスエリアを実装するために特に重要だとわかったものだけが、CMMIに引用されている。

4.2.3 実施される活動

『実施される活動』は、すべてのコモンフィーチャの中で最も構造的な多様性が大きい。これはキープロセスエリアに関する実装活動は、詳細さのレベル、組織の焦点(例えば、プロジェクトや組織)、および計画や文書化の必要性によって異なるからである。以下にいくつか一般性のある部分を示す。

計画の種類

キープラクティスで記述される計画の種類は主にふたつある。すなわち、公式計画(例えば、ソフトウェア開発計画、ソフトウェア品質保証計画、ソフトウェア構成管理計画)と非公式計画(例えば、ピアレビュー計画、リスク管理計画、技術管理計画)である。

非公式計画は、公式計画の一部として(例えば、ピアレビュー計画がソフトウェア開発計画の一部として文書化されることもある)、あるいは、公式計画の付帯計画として(例えば、ピアレビューのスケジュールがソフトウェア開発計画の一章として作成されることもある)文書化されるのが典型的である。公式計画には、それらを作成し、確実に従うという両面で、管理層の強いコミットメントが必要である。請負契約では、通常、これらの計画はその作業について契約する顧客への納入物になる。

公式計画

公式計画について言及している箇所としては、特に計画活動を取り上げているキープラクティスがふたつある。すなわち、文書化された手順に従って計画を策定もしくは改訂することを要求するものと、キーププロセスエリアの活動を計画に基づいたものとすることを要求するものである。

文書化された手順に関わるサブプラクティスは、計画のインプットとして何が必要かを示し、また、その計画に必要なコミットメントや支援を得るために期待される処置などが示される。こうしたサブプラクティスでは、その計画の典型的なレビュー担当者が特定される。また、どのレベルの承認が必要かという点も明示される。

活動の基盤となる計画に言及しているサブプラクティスでは、対象となる計画に想定される内容を記述している。計画の種類や、その計画で扱う一般的なトピックに関する組織的柔軟性の必要性に応じて、計画内容についてどの程度まで詳細に触れるかが変わってくる。

非公式計画

非公式計画は、通常ひとつのキープラクティスで記述される。サブプラクティスには、計画内容に関する情報や、計画の策定と改訂の手順が含まれる。

文書化された手順に従う

タスクや活動の担当者が反復できる方法でそれを実行できるようにし、また、その分野の一般知識を持つ他の人が同じようにそのタスクや活動を学習し、実施できるようにするには、文書化された手順が通常必要である。これはプロセス制度化の一面である。

ひとくちに文書化された手順といっても、手続きを個人的に手書きしたものから、組織の正式な標準処理手続きまで、その公式性や詳細のレベルは多種多様である。この公式性や詳細さのレベルは、そのタスクもしくは活動を誰が行うか(例えば、個人かチームか)、実施の頻度はどの程度か、結果の重要度や利用目的、および結果を受け取る相手によっても左右される。

ソフトウェアに
割り当てられた
システム要件

「ソフトウェアに割り当てられたシステム要件」とは、CMMでは通常『割り当てられた要件』と呼ぶが、システム要件の部分集合であり、システムのソフトウェアコンポーネントにおいて実装される部分を指す。割り当てられた要件はソフトウェア開発計画への主要インプットである。ソフトウェア要件分析の結果、割り当てられた要件が詳細化され改良される。その結果、ソフトウェア要件として文書化される。

顧客要件は単にソフトウェアだけでなくシステム全体に関与する。CMMにおいて顧客要件の検討は、ソフトウェアで実装される顧客要件が中心となる。ハードウェア、ソフトウェアなどへのシステム要件の割り当ては、システム設計全体の一部としてシステムエンジニアリンググループが行うのが典型的である。ソフトウェアプロジェクトに割り当てられるシステム要件は、通常、CMMでは『割り当てられた要件』と呼ばれる。これには技術要件（機能充足性、性能など）と非技術要件（納期、コストなど）が含まれる。

顧客と供給者との関係

顧客はその組織の内部の場合も、外部の場合もあり得る。組織内の顧客の例としてはマーケティング部門がある。組織外の顧客の例としては米国国防省がある。米国国防省との契約の場合がそうであるように、ユーザが顧客と異なる場合もある。CMMは、重要なソフトウェアコンポーネントを伴うシステムを購入する外部顧客の観点で書かれている。

グループ間の境界は、CMMで述べられているように、適切に解釈されなければならない。例えば、ソフトウェアのみの調達の場合、顧客とソフトウェアエンジニアリンググループの間にシステムエンジニアリンググループは介在しないだろう。この場合、顧客要件、システム要件、割り当てられた要件はおそらく同一のものであり、システムエンジニアリンググループの責任は、顧客とソフトウェアエンジニアリンググループが分担することになる。

「進捗を確認し、是正処置を取る」と「管理する」の違い

レベル2の『ソフトウェアプロジェクト進捗管理』においては、『...の進捗を確認し、適宜、是正処置をとる』という表現がキープラクティスの随所で使われる。レベル3の『ソフトウェア統合管理』では、同様のキープラクティスの随所で『...を管理する』という表現が使われている。このように表現が違うのは、レベル2ではプロジェクトのソフトウェアプロセスが完全に定義されていないことを反映している。管理層の処置は問題が起こってから反応する形態になりがちである。レベル3では、プロジェクトのソフトウェアプロセスが完全に定義されており、各種ソフトウェア作業成果物、タスク、および活動の関係が整った形で定義されている。管理層は、よりよく問題を予期できるようになり、事前に発生を防止できるようになる。介入が必要な場合には、ソフトウェアプロセス全体への影響を理解し、そうした介入がさらに効果的に定義され適用される。

「レビューされている」と「ピアレビューを受ける」の実施

レビューでは、コメントや承認を得るため、ソフトウェア作業成果物もしくは作業成果物一式がマネージャ、顧客、エンドユーザー、あるいはその他の関係者に提示される。レビューは、典型的にはタスクの最後に行われる。ピアレビューでは、欠陥を特定するため、ソフトウェア作業成果物もしくは作業成果物一式を開発者の同僚に提示する。ピアレビューには、マネージャ、顧客、およびエンドユーザーは参加しないのが典型的である。ピアレビューは、タスクの進行中に行われる必須な部分である。ピアレビューは、欠陥を早期に取り除いて、成果物の生産性と成果物の品質を向上させるために実施される。レビューを受けるソフトウェア成果物もあれば、ピアレビューを受けるソフトウェア成果物もあり、またピアレビューとレビューの両方を受けるソフトウェア成果物もある。

「構成管理下に置かれる」と「管理され制御されている」の違い

いくつかのソフトウェア作業成果物（例えば、ソフトウェア設計とコード）に対しては、所定の時点でベースラインが確立されるべきである。ベースラインは、公式なレビューを経て合意され、その後の開発の土台として役立てられる。ベースラインが設定された項目については、厳格な変更管理プロセスが適用される。ベースラインは、顧客との交渉に制御と安定性を与える。このような活動を、ベースライン構成管理と呼ぶこともある。このようなソフトウェア作業成果物に対して、『構成管理下に置かれる』という表現が使われる。

開発者によって構成の制御が行われているときは、通常、開発構成管理と呼ばれている。開発構成管理下にある項目のいくつかは、開発の所定の時点でベースライン構成管理下に置かれることがある。『構成管理下に置かれる』という表現は、開発構成管理を拡張したものとして解釈することができる。ただし、妥当な最低限の解釈としては、ベースライン構成管理のみが要求されることである。

例えば見積りやソフトウェア開発計画など、構成管理下に置かなくてもよいいくつかのソフトウェア作業成果物についても、『管理され制御されている』ことは必要とされる。この表現は、ソフトウェア作業成果物を特定し定義するプロセスを特徴付けるために使われる。ベースラインの一部ではなく、したがって構成管理下に置くわけではないが、プロジェクトの秩序ある進行のために管理されなければならない作業成果物に対して使われる。『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

4.2.4 計測と分析

『計測と分析』コモンフィーチャにあるキープラクティスは、『実施される活動』コモンフィーチャのキープラクティスに関する状況を判断するのに必要である基本的な計測プラクティスを記述している。キープロセスエリアの活動にそもそも含まれる計測は、『実施される活動』コモンフィーチャの中に含まれている。

計測に関しての推奨例は補足情報として表現されている。なぜならば、プロジェクト環境が異なれば、計測ニーズや計測アプローチも異なるからである。

4.2.5 履行検証

『履行検証』コモンフィーチャは、一般的に、上級管理層およびプロジェクト管理層による監督に関連したキープラクティスや、ソフトウェア品質保証グループや他の者が実施すると期待されている特定の検証活動を含んでいる。この活動はキープラクティスが適切に実施されていることを検証するためのものである。

定期的な上級管理層の監督

上級管理層による定期的なレビューの主目的は、ソフトウェアプロセス活動に対する認識と見通しを上級管理層に提供することである。活動のレビューは、適切な抽象レベルでタイムリーに行う。レビューの間隔は、組織のニーズに合わせる。例外報告への体制が十分整っていれば、上級管理層のレビュー間隔は長くても構わない。

上級管理層によるレビューの範囲と内容は、どの上級マネージャがレビューに関与するかに依存する。組織のすべてのソフトウェア活動に責任を持つ上級マネージャによるレビューは、全組織を所管する上級役員によるレビューとは異なったスケジュールが設定されるし、異なったトピックスを取り上げることが期待されている。上級管理層によるレビューでは、プロジェクト管理層の監督レビューとは異なるトピックスを扱ったり、同様のトピックスをより高い抽象度で扱うことが期待されている。

定期的およびイベント発生を契機とするプロジェクト管理層の監督

ステージの違いやプロジェクト特性に応じて、プロジェクトが異なる種類のレビューを必要とすることを強調するキープラクティスでは、『定期的、かつイベント発生を契機として』というテンプレートが用いられている。プロジェクト管理層は、ソフトウェア作業状況に関心を払い続けるべきであり、ソフトウェアプロジェクトに重大なイベントが発生した場合に報告を受けるべきである。プロジェクト管理層が参加する例としては、重要な設計レビューのような公式レビューの他、プロセス改善計画の状況やプロセス非遵守課題の解消などプロセス課題に関わるレビューなどがある。

プロジェクトレベルでは、プロジェクト管理層がプロジェクトの作業面により積極的に関与することを反映して、プロジェクト管理層による監督は、上級管理層による監督より詳細なレベルであることが期待されている。

ソフトウェア品質保証活動

ソフトウェア品質保証(SQA)グループによるレビューかつ/または監査として適切と考えられる特定の活動が、キープラクティスとして記述されている。『トレーニングプログラム』や『グループ間調整』キーププロセスエリアのように、SQA検証活動については記述されていない特例もある。これらはソフトウェアプロジェクトと組織の境界にあるキーププロセスエリアであり、そこではSQAグループが権限を持つことが期待されていない。

4.3 ソフトウェアプロセス定義の解釈

ソフトウェアプロセス定義は、より高い成熟度を達成するための基礎となる。本節では、ソフトウェアプロセス定義の諸側面について論じる。これらは、レベル3の『組織プロセス定義』に始まるプロセス定義に関するキープラクティスを利用する上で役立つものである。

CMMにおけるプロセス定義の基本コンセプトとは、「組織の標準ソフトウェアプロセス」のことである。「組織の標準ソフトウェアプロセス」は、基本プロセスの利用上の定義であり、組織におけるソフトウェアプロジェクト横断的な共通ソフトウェアプロセス確立の指針となる。また「組織の標準ソフトウェアプロセス」は、基本的なソフトウェアプロセス要素を記述し、各ソフトウェアプロジェクトは、これらの要素を「プロジェクトの定義されたソフトウェアプロセス」に組み込むことが期待されている。これらソフトウェアプロセス要素間の関係(例えば、順序付け、インタフェース)についても記述する。「組織の標準ソフトウェアプロセス」は、ソフトウェア活動の首尾一貫した実施方法を組織横断的に確立するものであり、長期的な安定と改善に不可欠なものである。

組織レベルでは、「組織の標準ソフトウェアプロセス」は、公式な方法で記述され、管理され、制御され、そして改善される必要がある。プロジェクトレベルでは、「プロジェクトの定義されたソフトウェアプロセス」の有用性と、プロジェクトに対する付加価値に重点が置かれる。「プロジェクトの定義されたソフトウェアプロセス」は、プロジェクトで使われるソフトウェアプロセスの利用上の定義である。また「プロジェクトの定義されたソフトウェアプロセス」は、整った形で特徴付けられ理解されたソフトウェアプロセスのことであり、ソフトウェア標準、手順、ツール、および手法によって記述される。プロジェクトの特徴に合わせるために、「組織の標準ソフトウェアプロセス」をテーラリングして、「プロジェクトの定義されたソフトウェアプロセス」を作成する。

『組織プロセス定義』のキープラクティスは、プロセス定義に対するアプローチを反映する用語で表現されており、安定性と柔軟性の両面を支援する。このアプローチを図4.1に示し、重要要素を以下に記述する。

4.3.1 プロセス定義のコンセプト

プロセス定義作業において、SEI が採ったアプローチを支える基本コンセプトは、成果物を開発し保守するのと同様の方法で、プロセスを開発し保守することである。以下がそれに必要なものである：

- どんなプロセスが記述されるかを定義する要件
- プロセスがどのように定義されるかについて情報を提供するアーキテクチャと設計
- プロジェクトや組織の状況に応じたプロセス設計の実装
- 計測によるプロセス記述の妥当性確認
- プロセスで想定した組織やプロジェクトにおける広範な作業に対するプロセスの展開

成果物開発のアナロジーを用いて、ソフトウェアプロセスの開発や保守の枠組みは、これらのコンセプトをプロセス開発の分野により特化したものに言い換えるように進化した（これはリアルタイム組み込みシステムの開発や管理情報システムの開発で使われる用語が特化していることに似ている）。この枠組みの主要要素を図4.1に示し、以下に簡単に説明する。

プロセスエンジニアリング分野で発展しつつあるプロセス定義のコンセプトについて、詳しくは論文『Software Process Development and Enactment: Concepts and Definitions』[Feiler 92]を参照のこと。

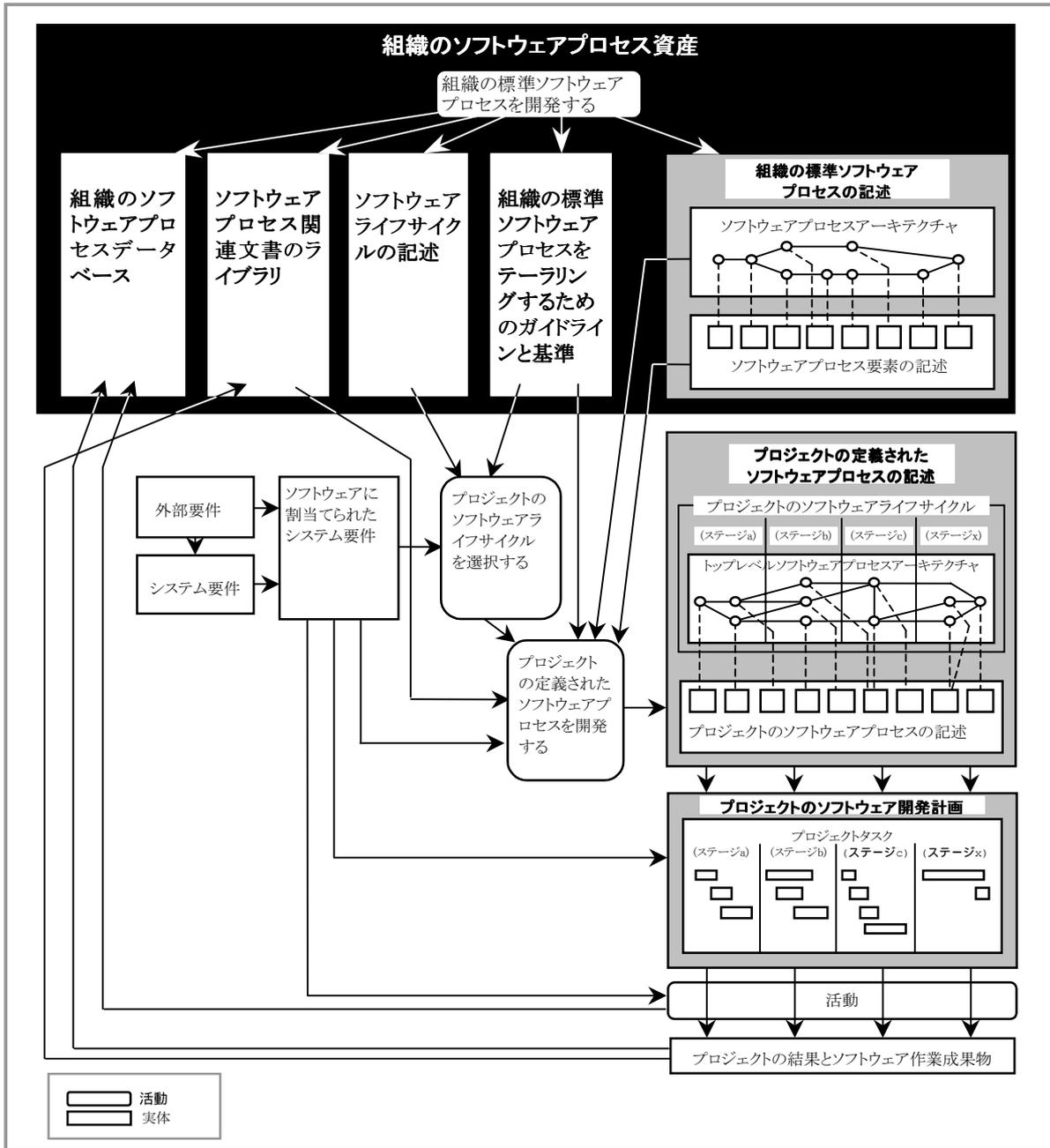


図4.1 CMMで使用される概念的なソフトウェアプロセスの枠組み

4.3.2 「組織のソフトウェアプロセス資産」に関するコンセプト

組織の ソフトウェア プロセス資産

組織は、図4.1に示すソフトウェアプロセス資産を確立し保守する。これらのソフトウェアプロセス資産には以下のものが含まれる：

- 「組織の標準ソフトウェアプロセス」(ソフトウェアプロセスアーキテクチャとソフトウェアプロセス要素を含む)
- 使用が承認されたソフトウェアライフサイクルの記述
- 「組織の標準ソフトウェアプロセス」をテーラリングするガイドラインと基準
- 「組織のソフトウェアプロセスデータベース」
- ソフトウェアプロセス関連文書のライブラリ

ソフトウェアプロセス資産は、「プロジェクトの定義されたソフトウェアプロセス」を開発し、保守し、そして実装しようとするプロジェクトが利用できるようになっている。

組織では、標準ソフトウェアプロセスを確立するアプローチによって、ソフトウェアプロセス資産が様々な方法でまとめられる。例えば、ソフトウェアライフサイクルの記述は、「組織の標準ソフトウェアプロセス」にとって不可欠な部分であるかもしれない。あるいはまた、ソフトウェアプロセス関連文書のライブラリのある部分は、「組織のソフトウェアプロセスデータベース」に格納されるかもしれない。

組織の

標準ソフトウェア
プロセス

「組織の標準ソフトウェアプロセス」は、基本的なプロセスの利用上の定義であり、組織内のソフトウェアプロジェクト横断的な共通ソフトウェアプロセスを確立する際の指針となる。また「組織の標準ソフトウェアプロセス」は、基本的なソフトウェアプロセス要素を記述し、各ソフトウェアプロジェクトは、これらの要素を「プロジェクトの定義されたソフトウェアプロセス」に組込むことが期待されている。これらソフトウェアプロセス要素間の関係(例えば、順序付け、インタフェース)についても記述する。「組織の標準ソフトウェアプロセス」は、ソフトウェアを開発し保守する組織内のプロジェクト間において、共通のソフトウェアプロセスを確立する際の指針となる。

ソフトウェアプロセス要素間の関係は、『ソフトウェアプロセスアーキテクチャ』とも言われている。

「組織の標準ソフトウェアプロセス」は、「プロジェクトの定義されたソフトウェアプロセス」の基盤となる。それは組織のプロセス活動に継続性を与え、組織内で使われるソフトウェアプロセスの計測と長期的な改善の参考になるものである。

ソフトウェア

プロセス

アーキテクチャ

ソフトウェアプロセスアーキテクチャとは、「組織の標準ソフトウェアプロセス」の抽象度の高い記述(要約)である。そこには、「組織の標準ソフトウェアプロセス」におけるソフトウェアプロセス要素間の順序付け、インタフェース、相互依存、およびその他の関係が記述されている。また他の外部プロセス(例えば、システムエンジニアリング、ハードウェアエンジニアリング、および契約管理)とのインタフェース、依存関係、およびその他の関係も記述されている。

ソフトウェア
プロセス要素

ソフトウェアプロセス要素は、ソフトウェアプロセス記述の構成要素である。各プロセス要素は、整った形で定義され、境界が設定され、密接に関係したひと組のタスク(例えば、ソフトウェア見積もり要素、ソフトウェア設計要素、コーディング要素、ピアレビュー要素)を扱っている。プロセス要素の記述は、利用時に記入されるテンプレート、完成される断片、精緻化される抽象概念、あるいは修正をしてまたは修正せずに使用される完全な記述のいずれであってもよい。

使用が承認されたソフトウェア
ライフサイクル
の記述

ソフトウェアライフサイクルとは、ソフトウェア成果物が着想されたときから始まり、ソフトウェアがもうこれ以上使われなくなったときに終わるまでの期間のことである。典型的には、コンセプトステージ、要件ステージ、設計ステージ、実装ステージ、テストステージ、設置と検収ステージ、運用保守ステージ、および場合によっては廃棄ステージがソフトウェアライフサイクルに含まれる[IEEE-STD-610]。

組織では、契約をした顧客向けに、かつ/または市場のユーザ向けに様々なソフトウェアを作成するであろうから、ひとつのソフトウェアライフサイクルがすべての状況に適切ではないだろう。したがって、プロジェクトで使用されるソフトウェアライフサイクルとして、組織が複数を特定してもよい。典型的には、これらソフトウェアライフサイクルは、ソフトウェアエンジニアリングの文献をもとに、その組織向けに手直しされる。またこれらソフトウェアライフサイクルは、「組織の標準ソフトウェアプロセス」と組み合わせられ、「プロジェクトの定義されたソフトウェアプロセス」の作成に使われるべく利用可能になっている。

テーラリングの ガイドラインと 基準

「組織の標準ソフトウェアプロセス」は、プロジェクトによってそのままでは使えないような一般的なレベルで書かれている。ガイドラインは、(1) 使用が承認されたものからソフトウェアライフサイクルを選択し、(2) 「組織の標準ソフトウェアプロセス」や選択されたソフトウェアライフサイクルを、プロジェクトの特徴に合わせてテーラリングし、詳細化する際に、ソフトウェアプロジェクトの指針となるように確立される。

これらのガイドラインと基準は、全プロジェクトが共通の基盤に基づいて、「プロジェクトの定義されたソフトウェアプロセス」を計画し、実装し、計測し、分析し、そして改善することを確実なものとするのに役立つ。

組織のソフトウ ェアプロセスデ ータベース

「組織のソフトウェアプロセスデータベース」は、ソフトウェアプロセスと結果としてのソフトウェア作業成果物に関するデータ、特に「組織の標準ソフトウェアプロセス」に関わるデータを収集し利用可能にするために確立されるデータベースである。データベースには、実際の計測データとそれに関連した情報が含まれたり、参照付けられたりしている。関連した情報は、計測データを理解し、その合理性と適用性を評価するために必要である。

プロセスや作業成果物のデータの例としては、ソフトウェアの規模、工数、およびコストの見積り；ソフトウェアの規模、工数、およびコストの実際のデータ；生産性データ；ピアレビューのカバレッジと効率；およびソフトウェアコード内で発見された欠陥の個数や重大性がある。

ソフトウェアプロセス関連文書のライブラリ

ソフトウェアプロセス関連文書のライブラリは、(1) 現在進められている他のプロジェクトや将来のプロジェクトに潜在的に役立つプロセス文書、特に「組織の標準ソフトウェアプロセス」に関する文書を蓄積するため、(2) それらを組織全体で共有可能にするために確立される。このライブラリには、文書の例や実際の文書の断片が含まれており、将来のプロジェクトにおいて、「組織の標準ソフトウェアプロセス」をテラリングする際に使われることを期待している。これらの例には、「プロジェクトの定義されたソフトウェアプロセス」、標準、手順、ソフトウェア開発計画、計測計画、プロセストレーニング教材などの事項が含まれる。このライブラリは重要な資源であり、プロジェクト開始時点で成功プロジェクトの例を提供することによって、新プロジェクト開始に必要な工数の削減に役立てることができる。

4.3.3 「プロジェクトの定義されたソフトウェアプロセス」に関するコンセプト

「プロジェクトの定義されたソフトウェアプロセス」の記述

「プロジェクトの定義されたソフトウェアプロセス」の記述とは、プロジェクトで使われるソフトウェアプロセスを作業面から見た定義である。「プロジェクトの定義されたソフトウェアプロセス」とは、整った形で特徴付けられ理解されるソフトウェアプロセスであり、ソフトウェア標準、手順、ツール、および手法によって記述される。「プロジェクトの定義されたソフトウェアプロセス」は、プロジェクトの特徴に合わせるため、「組織の標準ソフトウェアプロセス」をテーラリングしたものである。

テーラリングには、組織で承認されたものの中からソフトウェアライフサイクルを選択し、プロジェクトの特徴に合わせて「組織の標準ソフトウェアプロセス」を修正することが含まれる。

「プロジェクトの定義されたソフトウェアプロセス」は、マネージャや技術要員がプロジェクトのタスクや活動を実施することを計画し、実施し、そして改善するための基盤となる。ひとつのプロジェクトで複数の定義されたソフトウェアプロセス(例えば、運用ソフトウェア用とテスト支援ソフトウェア用)を使ったり、また複数の類似プロジェクトでひとつの定義されたソフトウェアプロセスを使うこともできる。

ステージ

ステージとは、ソフトウェア開発作業を分割したものであり、プロジェクトで実行される関連タスクのセットである。ステージは、意味のあるまとまりであり、管理可能な大きさである。ステージは通常、ソフトウェアライフサイクルを小分けにしたものであり、多くの場合、次のステージ着手前に公式レビューを受けることで終結する。

タスク 実施される作業は、タスクにブレイクダウンされる。タスクとは、ソフトウェアプロセスにおける整った形で定義された単位であり、プロジェクトの状況に対する可視的なチェックポイントを管理層に与える。タスクには、開始基準（前条件）と完了基準（後条件）がある。

プロセス定義の文脈において、タスクとは、定義されたプロセスの整った形で定義されたコンポーネントである。すべてのタスクは活動と見なすことができるが、（ある活動がタスクを含むことはあるものの）すべての活動がタスクと見なせるほど整った形で定義されてはいない。このため、レベル2のキープラクティスでは『タスク』という表現は使われておらず、さほど厳密でない『活動』という表現が使われている。

活動 活動とは、ある狙いの達成に向けて、精神的かつ物理的に、実行されるステップや遂行される職務のことである。活動には、マネージャと技術要員が、プロジェクトと組織のタスクを実施するために行うすべての作業が含まれる。

ソフトウェア作業成果物(プロジェクトの結果)

活動やタスクの結果は、主にソフトウェア作業成果物で構成される。ソフトウェア作業成果物とは、ソフトウェアプロセスを定義し、維持し、そして使用する際に作成されるあらゆるもののことであり、顧客やエンドユーザーに納められるものもあればそうでないものもある。ソフトウェア作業成果物には、プロセス記述、計画、手順書、コンピュータプログラム、および関連する文書が含まれる。作業成果物は、プロセス中の次のステップに対するインプットとなったり、あるいはそのソフトウェアプロジェクトについての記録情報として将来のプロジェクトで使用される。

ソフトウェア作業成果物の例としては、計画、見積り、実際の工数データ、是正処置の文書、要件文書がある。ソフトウェア作業成果物の一部であって、顧客やエンドユーザーに納入されるものは、ソフトウェア成果物と言われる。

ソフトウェア成果物

ソフトウェア成果物とは、顧客やエンドユーザーに納入することが指定されているコンピュータプログラム、手順書、および関連する文書とデータの一式、もしくはその中の幾つかのことである。[IEEE-STD-610]

すべてのソフトウェア成果物は、ソフトウェア作業成果物でもある。しかし、顧客やエンドユーザーに納入されないソフトウェア作業成果物は、ソフトウェア成果物ではない。

4.3.4 「プロジェクトの定義されたソフトウェアプロセス」とソフトウェア開発計画の関係

「プロジェクトの定義されたソフトウェアプロセス」の記述は、通常、そのまま実施できるほど十分に具体的ではないであろう。その記述は、役割(誰がタスクを実施するか)や、タスクを実施するために必要なソフトウェア作業成果物の種類といった事項を特定しているのが典型的である。しかし、その役割を担う個人、作られる具体的なソフトウェア作業成果物、あるいはタスクや活動を実施するためのスケジュールを明確化しているわけではない。

プロジェクトのソフトウェア開発計画は、「プロジェクトの定義されたソフトウェアプロセス」(何をどのようにするか)と、どのようにプロジェクトを実施するかの詳細(例えば、誰がどんなスケジュールでどのソフトウェア作業成果物を作成するか)の橋渡しとなる。プロジェクトのソフトウェア開発計画は、ひとつの文書であることもあるし、多数の計画を集めたものをひとつのソフトウェア開発計画ということもある。「プロジェクトの定義されたソフトウェアプロセス」とそのソフトウェア開発計画の組み合わせが、プロセスを実際に実施可能にする。

4.3.5 ライフサイクルとCMM

キープラクティスは、ソフトウェアライフサイクルの選択を制限するものではない。ある特定のソフトウェアライフサイクルを様々な場面で使ってきた人は、そのライフサイクルの要素が、キープラクティスの構成や構造の中にあることに気づくだろう。しかし、ある特定のソフトウェアライフサイクルの使用を勧めたり、妨げたりしようという意図はない。

『ステージ』という用語は、ソフトウェアプロジェクトの作業の定義された分割を指して使われるが、この用語はある特定のソフトウェアライフサイクルに結び付けられるべきものではない。キープラクティスで使われているように、『ステージ』は、厳格に順序付けられているステージを意味しても良いし、重なったり繰返したりするステージを意味してもよい。

4.3.6 技術とCMM

キープラクティスは、プロトタイピングやオブジェクト指向設計、あるいは、ソフトウェア要件、設計、コード、および他の要素の再利用といった特定のソフトウェア技術を必要としたり、またはそれを妨げたりするものではない。

4.3.7 文書化とCMM

キープラクティスは、いくつかのプロセス関連文書を記述しており、それぞれが特定分野の内容を扱っている。キープラクティスで名づけられている文書と、組織やプロジェクトにおける実際の作業成果物との間に1対1の対応関係があることを、キープラクティスは求めている。DoDによって明記されている文書、またはDOD-STD-2167AやIEEEのソフトウェア標準などの標準とも、1対1の対応関係を意図していない。キープラクティスでは、そこで記述されている文書に当てはまる内容が、組織やプロジェクトで文書化される作業成果物の一部分であることのみを求めている。

文書構造の観点から、キープラクティスの中で参照される文書の内容は、より大きな文書の一部であってもよい。例えば、ある組織は、ソフトウェアリスク管理計画の本質的要素を含んだソフトウェア開発計画を持っているだろう。

あるいは、キープラクティスの中で参照される文書の内容が、キープラクティスの中で名付けられている文書一式とは異なる複数の文書に分散していてもよい。例えば、プロジェクトが、ソフトウェア開発計画、ソフトウェア管理計画、およびプロジェクトのWBS(work breakdown structure)という3個の文書を作成して、ソフトウェアプロジェクトのソフトウェアリスク管理計画、ソフトウェア品質保証計画、ソフトウェア開発計画に対する一連のキープラクティスを満足させてもよい。

4.3.8 プロセスデータの収集と分析

プロセスデータの収集と分析に関するキープラクティスは、成熟度レベルの向上につれて進化する。

レベル2では、データは主にプロジェクトの作業成果物の規模、工数、スケジュールに関するものである。また各プロジェクトで別々に定義され、収集され、そして蓄積される。データは、非公式なメカニズムによってプロジェクト間で共有される。

レベル3では、各プロジェクトが、「組織の標準ソフトウェアプロセス」からテーラリングした「プロジェクトの定義されたソフトウェアプロセス」を持っている。それぞれの「プロジェクトの定義されたソフトウェアプロセス」に関するデータは、「組織のソフトウェアプロセスデータベース」に収集され蓄積される。収集され蓄積されるデータは、プロジェクト毎に異なるであろうが、「組織のソフトウェアプロセスデータベース」の中では整った形で定義される。

レベル4では、「組織の標準ソフトウェアプロセス」に基き、組織は標準セットの計測値を定義する。すべてのプロジェクトは、プロジェクト特有のデータと同様に、標準セットの計測データを収集し、「組織のソフトウェアプロセスデータベース」に蓄積する。プロジェクトは、「プロジェクトの定義されたソフトウェアプロセス」のプロセス実績を定量的に理解し安定させるために、このデータを使用する。さらに組織も、「組織の標準ソフトウェアプロセス」に対するプロセス能力ベースラインを確立させるために、このデータを使用する。

レベル5では、データは、技術やプロセスの改善領域を選択し、これらの改善を計画し、そして組織のプロセス能力に対する改善の効果を評価するために使われる。

4.4 組織における構造と役割

CMMIは、組織について特定の構造やモデルから独立であり続けることを目指している。しかし、CMMのプラクティスを表現するためには、組織における構造と役割に関する首尾一貫した用語を使用する必要がある。これらの用語は、ある特定の組織における構造と役割とは異なっていることがある。以下の節では、CMMのキープラクティスを解釈するのに必要な組織、プロジェクト、および役割に関する種々のコンセプトを記述する。

4.4.1 組織における役割

役割は、ひとりまたは複数の個人が負うであろう定義された責任の単位である。以下の役割の記述は、キープラクティスで頻繁に使用される：

マネージャ	マネージャは、その責任範囲内で、タスクや活動を実施する個人に対して、技術面や管理面で統制し制御する役割を遂行する。伝統的なマネージャの役割としては、その責任範囲内で、計画し、資源を割当て、組織化し、統制し、そして制御することなどがある。
-------	--

上級マネージャ 上級マネージャは、組織内の高レベルな管理的役割を遂行する。上級マネージャの主要な管理の焦点は、組織の長期的な活力であり、短期的なプロジェクトや契約上の関心事やプレッシャではない。一般に、エンジニアリングの上級マネージャは、複数のプロジェクトに対して責任を負うことになるだろう。また、上級マネージャは、ソフトウェアプロセスの長期的な改善のために資源(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)を提供し保護する。

CMMにおける上級管理層とは、上記を満足するすべてのマネージャを指し、組織全体の長までも含む。キープラクティスで用いられる上級管理層という言葉は、そのキープロセスエリアや対象となるプロジェクトや組織の状況に応じて適切に解釈されるべきである。キープロセスエリアのゴールを達成するために必須なリーダーシップと監督の役割を発揮しなければならない上級マネージャ達を明確に含めることを意図している。

プロジェクトマネージャ プロジェクトマネージャは、プロジェクト全体に対する事業上の全責任を負う役割を遂行する。プロジェクトマネージャは、ソフトウェア、あるいはハードウェア/ソフトウェアシステムを構築するプロジェクトを統制し、制御し、运营管理し、そして規制する個人である。プロジェクトマネージャは、顧客に対して最終的な責任を負う個人である。

プロジェクト中心の組織構造では、プロジェクトに携わる人々のほとんどがプロジェクトマネージャに報告をあげる。ただし、ある種の機能グループに関しては、マトリクス状の報告関係をとることもある。マトリクス状の組織構造では、事業要員のみがプロジェクトマネージャに報告することになるかもしれない。そしてエンジニアリンググループは、マトリクス状の報告関係を持つことになるだろう。

プロジェクトソフトウェアマネージャ プロジェクトソフトウェアマネージャは、プロジェクトのすべてのソフトウェア活動に対する全責任を負う役割を遂行する。プロジェクトソフトウェアマネージャは、プロジェクトマネージャがソフトウェアのコミットメントに関して任せた個人であり、プロジェクトのソフトウェアに関わるすべての資源を制御する。

プロジェクトのソフトウェアエンジニアリンググループは、プロジェクトソフトウェアマネージャに報告するだろう。ただし、ツール開発など一部の活動ではマトリクス状の報告関係をとることもあるだろう。

大規模プロジェクトの場合、プロジェクトソフトウェアマネージャは、第二、第三、あるいは第四線マネージャであろう。小規模プロジェクトやひとつのプロジェクトしかない部署の場合は、プロジェクトソフトウェアマネージャが、第一線ソフトウェアマネージャであったり、高レベルのマネージャであることがある。

第一線ソフトウェアマネージャ 第一線ソフトウェアマネージャは、ソフトウェア技術者や他の関連要員で構成される単一の組織単位（部署やプロジェクトチーム）の配員や活動に対して直接の管理責任（技術的な指示、人事や給与の管理など）を負う役割を遂行する。

ソフトウェアタスクリーダー ソフトウェアタスクリーダーは、特定のタスクに関する技術チームのリーダーとしての役割を遂行する。ソフトウェアタスクリーダーは、技術的な責任を負い、そのタスクに携わる要員に対して技術的な指示を与える。

ソフトウェアタスクリーダーは通常、そのタスクに携わる他のメンバと同じ第一線ソフトウェアマネージャに報告する。

要員、ソフトウェアエンジニアリング要員、個人 CMMのキープラクティスでは、様々な技術的役割を遂行する個人を表すのに、いくつかの用語が使用されている。要員は、タスクリーダーも含め、ソフトウェア開発、またはソフトウェア構成管理など、割り当てられた職務を完成する責任を負う個人である。ただしこれにマネージャは含まれない。

ソフトウェアエンジニアリング要員は、ソフトウェアタスクリーダーも含め、ソフトウェアの技術者(例えば、アナリスト、プログラマ、エンジニア)であり、プロジェクトでソフトウェア開発や保守を実施する。ただしこれにマネージャは含まれない。

キープラクティスで『個人』という用語が用いられるときは、その用語が現れる文脈によって修飾され、範囲が示される(例えば、『ソフトウェア外注契約の管理を行う個人』)。

システムエンジニアリングやシステムテストのような他のエンジニアリンググループに対しても、似たような役割の範囲が規定されている。

特定のプロジェクトや組織では、これらの役割と個人は1対1に対応しなくてもよい。ひとりで複数の役割を遂行したり、ひとつの役割を複数人で分担することもできる。

例えば、ソフトウェアのみの小規模プロジェクトでは、ひとりが6つの役割を担当することもあるだろう。すなわち、システムエンジニアリング第一線マネージャ、プロジェクトシステムエンジニアリングマネージャ、ソフトウェア第一線マネージャ、プロジェクトソフトウェアマネージャ、プロジェクトマネージャ、およびソフトウェア構成管理マネージャである。

やや大きなプロジェクトでは、ひとりがシステムエンジニアリング第一線マネージャ、プロジェクトシステムエンジニアリングマネージャ、およびプロジェクトマネージャを担当し、もうひとりが第一線ソフトウェアマネージャとプロジェクトソフトウェアマネージャを担当することもあるだろう。これらの2人のマネージャは、同じ第二線の組織に属することもあり、異なる第二線の組織に属することもある。

大規模プロジェクトでは、多くの役割、特に管理層の役割を、別々の個人が遂行することになるであろう。

4.4.2 組織の構造

CMMのキープラクティスを適切に解釈するためには、組織、プロジェクト、およびグループの基本的なコンセプトを理解しなければならない。以下では、CMMにおけるこれらの基本コンセプトを定義する：

組織	組織は、企業やその他の機関(例えば、政府の部局や出先機関)の中の単位であって、多くのプロジェクトを全体として管理している。組織内のすべてのプロジェクトは、共通のトップレベルのマネージャや共通の方針を共有している。
プロジェクト	プロジェクトは、協調作業を必要とする仕事であって、特定の成果物を開発しかつ/または保守することに焦点を当てる。成果物にはハードウェア、ソフトウェア、およびその他のコンポーネントが含まれる。プロジェクトは、独自の資金、費用会計、納入スケジュールを持つのが典型的である。
グループ	グループは、一連のタスクまたは活動に責任を持つ部門、マネージャ、および個人の集まりである。グループは、専任でない個人の場合、異なる部門から割り当てられた専任でない人達の場合、および専任の人達の場合など、さまざまな形態がある。

CMMでよく言及されるグループを以下に記述する：

ソフトウェアエンジニアリンググループ ソフトウェアエンジニアリンググループは、プロジェクトのソフトウェアの開発と保守(要件分析、設計、コーディング、テスト)に責任を持つ個人の集まり(マネージャと技術要員の両方)である。

ソフトウェア品質保証グループ、ソフトウェア構成管理グループ、およびソフトウェアエンジニアリングプロセスグループなど、ソフトウェア関連作業を実施するグループは、ソフトウェアエンジニアリンググループには含まれない。これらのグループは、『その他のソフトウェア関連グループ』のひとつと見なされる。

ソフトウェア関連グループ ソフトウェア関連グループは、ソフトウェアエンジニアリングにおける機能を代表する個人の集まり(マネージャと技術要員の両方)である。ソフトウェア関連グループはソフトウェアの開発や保守をサポートするが、直接の責任は持たない。

ソフトウェアエンジニアリングにおける機能の例として、ソフトウェア品質保証やソフトウェア構成管理がある。

ソフトウェアエンジニアリングプロセスグループ ソフトウェアエンジニアリングプロセスグループは、組織で使用するソフトウェアプロセスの定義、保守、および改善を促進する専門家で構成される。キープラクティスでは、一般化して『組織のソフトウェアプロセス活動に責任を持つグループ』と表現される。

システムエンジニアリンググループ	システムエンジニアリンググループは、次の活動に責任を負う個人(マネージャと技術要員の両方)の集まりである。システム要件を明記する。システム要件をハードウェア、ソフトウェア、およびその他のコンポーネントに割り当てる。ハードウェア、ソフトウェア、およびその他のコンポーネント間のインタフェースを明記する。仕様に準拠することを確実なものにするためにこれらのコンポーネントの設計と開発をモニターする。
システムテストグループ	システムテストグループは、ソフトウェア成果物が要件を満たすかどうかを判断するために、独立したシステムテストを計画し実施する責任を負う個人(マネージャと技術要員の両方)の集まりである。
ソフトウェア品質保証グループ	ソフトウェア品質保証グループは、ソフトウェアプロセスの各ステップと標準が確実に守られるように、プロジェクトの品質保証活動を計画し実装する責任を負う個人(マネージャと技術要員の両方)の集まりである。ソフトウェア品質保証に関する組織の課題については、4.4.3で議論する。
ソフトウェア構成管理グループ	ソフトウェア構成管理グループは、ソフトウェアプロジェクトの公式な構成管理活動を計画し、調整し、そして実装する責任を負う個人(マネージャと技術要員の両方)の集まりである。

トレーニンググループ トレーニンググループは、組織のトレーニング活動を調整し手配する責任を負う個人(マネージャと要員の両方)の集まりである。典型的には、このグループが大半のトレーニングコースを準備し実施し、また他のトレーニング手段の利用に関する調整も行う。

4.4.3 独立性と組織の構造

独立性が必要となるキープラクティスが適切に解釈され従われるように、組織は注意を払わなければならない。特に小型のプロジェクトや組織の場合、これが当てはまる。技術的、組織的バイアスがプロジェクトの品質やリスクに影響を及ぼすことが予想される場合、キープラクティスは独立性を求めている。例えば、以下のふたつのプラクティスは独立性を扱っている：

- SQAグループは、プロジェクトマネージャ、ソフトウェアエンジニアリンググループ、およびその他のソフトウェア関連グループとは独立な上級管理層への報告経路を持つ（『ソフトウェア品質保証』のコミットメント1.2）。
- (システムおよび検収)テストケースとテスト手順は、ソフトウェア開発者から独立したグループが計画し準備する（『ソフトウェアプロダクトエンジニアリング』の活動7.3）。

技術的な考察に基づいて、システムテストや検収テストの独立性が必要である。独立性が確保されていれば、ソフトウェア開発者や保守作業による設計上および実装上の決定が、テスト作業者に不適切な影響を与えなくなることが確実なものとなる。

SQAメンバがプロジェクトのスケジュールやコストの圧力に影響されることなく活動を実施するためには、SQAグループの独立性が必要である。組織的な独

立性なしに、効果的な作業上の独立性を確保することは難しい。例えば、プロジェクトマネージャへ報告する従業員は、たとえ重大な非遵守課題が存在したとしてもテスト活動を中断することを嫌がることもある。

SQAなど独立性が必要な活動については、戦略的事業目標と事業環境の中で、それらの活動をサポートできる組織構造を決定しなければならない。

独立性とは以下の事項を満たすものをいう：

- SQAの役割を実施する担当者に、ソフトウェアプロジェクトに対する上級管理層の『目と耳』としての組織的な自由を与える。
- ソフトウェアプロジェクトの管理層が行う実績査定の対象から、そのプロジェクトに対してSQAの役割を実施する担当者を除外する。
- 上級管理層に、ソフトウェアプロジェクトのプロセスと成果物について客観的な報告が行われているという信頼感を与える。

キープラクティスは、独立性の基準の解釈を許容しているので、このキープロセスエリアのゴールが達成されているかどうかを組織として決定するには、専門家の判定が必要である。

4.5 専門家の判定の適用

広範な状況に適用する妥当な原則の完全な集合を提供するために、一部のキープラクティスについては、意図的に柔軟性を許容する表現を用いている。キープラクティス全体でも、『影響を受けるグループ』、『適宜』、『必要に応じて』など、明確でない表現が使用されている。このような明確でない用語の使用は、キープラクティスの中では一般に最小限に止めており、多くの場面で例が示され、特にその用語の初出時には例を示している。異なる組織、同一組織の異なるプロジェクト、あるいは、同一プロジェクトでもライフサイクルの異

なる時点では、これらの表現が異なる意味を持つことになるだろう。各プロジェクトや組織は、それらの特定の状況に応じて表現を明確化しなければならない。

これらの表現を明確化するには、その表現が使われている総合的な文脈を組織が考慮する必要がある。このような表現に対する特定の解釈がキープロセスエリアのゴールを満足するかどうかを問い直すのが適切である。ゴールが達成されたかどうか決定するには、専門家が判定しなければならない。付録Bの用語集は、キープラクティスにおける表現の解釈の指針となるであろう。

キープラクティスの解釈やそれらがキープロセスエリアのゴールにどう貢献するかを解釈する際にも、専門家が判定しなければならない。一般に、キープロセスエリアでは、組織の規模や成果物に関わらず、すべてのソフトウェア組織で見受けられるべき基本的な行動の集まりを記述している。しかし、CMMのキープラクティスは、プロジェクトまたは組織の事業環境や特定の状況に照らし合わせて解釈しなければならない。この解釈は、CMMだけでなく組織とプロジェクトについての十分な知識に基づいているべきである。キープロセスエリアのゴールは、この解釈を組み立てるための手段を提供する。ある組織があるキープロセスエリアのゴールをキープラクティスと著しく異なる実装で達成する場合は、解釈の根拠を文書化するべきである。文書化された根拠は、あるプラクティスがなぜそのように実装されたのか、アセスメントチームや審査チームが理解する手助けとなる。

専門家の判定を用いるということは、ソフトウェアプロセスにおける『良さ』の問題に行きつくことになる。CMMIは、多くのソフトウェア環境において『妥当な』プロセスに対する最低限の基準を確立しているのであって、ソフトウェアプロセスの『良さ』の要件を提起しているのではない。プロセス管理の狙いは、組織の事業ニーズに基づく体系的な改善の基礎として使用され実行できるプロセスを確立することである。

『妥当な』ソフトウェアプロセスの基準とは何か。妥当なソフトウェアプロセスとは、組織的能力の構築に効果的であり、定義されたプロセスのほとんどの要件を満たすプロセスである。具体的には、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、改善できるものである。

ある組織がさいころを振って見積るソフトウェアプロセスを確立したとしたら、これは合理的なプロセスになり得るだろうか。このプロセスは、確かに文書化し首尾一貫して従うことができる。多くの見積もり技法と同じくらい現実的であると主張する人がいるかもしれない。しかし、大半のソフトウェア専門家は『さいころを振る』を合理的な見積りプロセスとは判定しないだろう。なぜならそれは確率法則に支配され、改良の余地がないからである。

『ジョージに聞きに行け』というプロセスを記述することは、『さいころを振る』とどれほど違うだろうか。『ジョージに聞きに行け』は、非常に良い見積り手法になり得る。ジョージが身近にいる限り、一貫性があり、反復可能でもある。ただし、他の個人をトレーニングすることができないため、我々の基準を満たしていないだろう。これは、ジョージがいなかったら反復できない人間中心のプロセスである。これでは継続する組織能力を構築できない。

見積りにデルファイ法(その分野の専門家が検討事項をレビューし、勧告について合意に到達する方法)の一種を使用することは、合理的なソフトウェアプロセスと通常は判定されるだろう。デルファイ法は人間中心のプロセスであるが、デルファイ法に基づく規模見積り方法は、合理的で効果的なプロセスという基準を満たす。組織の能力は、デルファイ法のような構造化された技法を基盤とすることも可能である。

根本的な意味で、専門家の判定はこのような区別をするために必要である。『遵守』と『良さ』の判別は難しい。ゴールはキープラクティスを要約したものであり、キープラクティスは合理的なソフトウェアプロセスを記述している。ただし、合理的なプロセスを遵守することは、そのプロセスが目的の達成に効率的であるということの意味しない。組織とプロジェクトの成功には、様々な要

因が影響するだろう。例えば、できあがった製品が全く売れなかったとしたら、プロジェクト自体は成功であっても、商業的には失敗である。

『良さ』は、プロジェクトと組織の事業環境や特定の状況に照らしてはじめて解釈できる。『良さ』は、継続的プロセス改善サイクルの一部として、その組織によってのみ判断できる。完璧は永遠に達成されず、継続的プロセス改善に終わりはない。

要件管理

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』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する共通理解を確立することである。

『要件管理』に必然的に含まれる活動には、ソフトウェアプロジェクトの要件に関し顧客との合意を確立すること、およびその合意を維持することが挙げられる。この合意を、『ソフトウェアに割り当てられたシステム要件』という。『顧客』は、システムエンジニアリンググループ、マーケティンググループ、あるいは他の社内組織であったり、外部の顧客であったりする。合意は、技術的な要件と非技術的な要件（例えば、納期）の両方を扱う。合意は、ソフトウェアのライフサイクル全般にわたって、ソフトウェアプロジェクト活動の見積り、計画、実施、および進捗確認の基盤となる。

ソフトウェア、ハードウェア、およびその他のシステムコンポーネント（例えば、人間）へのシステム要件の割り当ては、ソフトウェアエンジニアリンググループ外のグループ（例えば、システムエンジニアリンググループ）が実施することがある。すなわち、ソフトウェアエンジニアリンググループがこの割り当てに直接的な制御権を持たないこともある。プロジェクトの制約条件のなかで、ソフトウェアエンジニアリンググループは、自身の担当する「ソフトウェアに割り当てられたシステム要件」の文書化と制御を確実なものにするため、適切な処置を講ずる。

この制御を実現するため、ソフトウェアエンジニアリンググループは、「ソフトウェアに割り当てられたシステム要件」の初期レビューおよび更新時レビューを実施する。レビューで判明したすべての課題は、ソフトウェア

に割り当てられたシステム要件」をソフトウェアプロジェクトに組み込む前の段階で解決しておかねばならない。「ソフトウェアに割り当てられたシステム要件」に変更があった場合は、影響を受けるソフトウェア計画、作業成果物、および活動についても、更新された要件と首尾一貫するよう常に補正する。

ゴール

- 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。

実施のコミットメント

- コミットメント1 プロジェクトは、「ソフトウェアに割り当てられたシステム要件」の管理に関して、明文化された組織方針に従う。

これらのプラクティスでは、「ソフトウェアに割り当てられたシステム要件」を、『割り当てられた要件』という。

割り当てられた要件とは、システム要件の部分集合であり、システムのソフトウェアコンポーネントにおいて実装される部分を指す。割り当てられた要件は、ソフトウェア開発計画への主要なインプットとなる。ソフトウェア要件分析の結果、割り当てられた要件が詳細化され改良される。その結果、ソフトウェア要件として文書化される。

(コミットメント1) この方針で明記される典型的な事項：

1. 割り当てられた要件が文書化される。
2. 割り当てられた要件は、
 - ソフトウェアマネージャ
 - 他の影響を受けるグループによってレビューされる。

影響を受けるグループの例：

- システムテスト
- ソフトウェアエンジニアリング (ソフトウェア設計などのすべてのサブグループを含む)
- システムエンジニアリング
- ソフトウェア品質保証
- ソフトウェア構成管理
- 文書化支援

3. ソフトウェア計画、作業成果物、および活動は、割り当てられた要件に対する変更と首尾一貫するように変更される。

実施能力

能力1

各プロジェクトに対し、システム要件を分析し、それらをハードウェア、ソフトウェア、および他のシステムコンポーネントに割り当てる責任が確立される。

システム要件の分析と割り当ては、ソフトウェアエンジニアリンググループの責任ではないが、当該グループの作業の前提要件である。

(能力1) この責任の部分で扱う項目：

1. プロジェクトの全期間にわたって、システム要件ならびにその割り当てを管理し文書化する。
2. システム要件ならびにそれらの割り当てに変更を反映させる。

能力2 割り当てられた要件が文書化される。

割り当てられた要件は、以下のものを含む：

1. ソフトウェアプロジェクトの活動に影響を与え、またその活動を決定する非技術的要件（合意事項、条件、および/または契約条項）。

合意事項、条件、および契約条項の例：

- 納入する成果物
- 納入日
- マイルストーン

2. ソフトウェアに対する技術的要件。

技術的要件の例：

- エンドユーザ、オペレータ、支援、あるいは統合の各機能
- 性能要件
- 設計上の制約
- プログラミング言語
- インターフェイス要件

3. ソフトウェア成果物が、割り当てられた要件を満足しているかどうかの妥当性を確認するのに使用する検収基準。

能力3 割り当てられた要件を管理するために、適切な資源と資金が提供される。

1. 割り当てられた要件の管理担当者として、アプリケーションドメインやソフトウェアエンジニアリングに経験や専門知識を有する個人を割り当てる。
2. 要件管理活動を支援するツールが利用可能になっている。

支援ツールの例：

- 表計算プログラム
- 構成管理用ツール
- 追跡用ツール
- テスト管理用ツール

能力4 ソフトウェアエンジニアリンググループや他のソフトウェア関連グループのメンバは、要件管理活動を実施するためのトレーニングを受ける。

トレーニングの例：

- プロジェクトで使用する、手法、標準、および手順
- 当該アプリケーションドメイン

実施される活動

活動1 ソフトウェアエンジニアリンググループは、割り当てられた要件がソフトウェアプロジェクトに組み込まれる前の段階でレビューする。

1. 割り当てられた要件の不完全なところや抜けが特定される。

(活動1)

2. 割り当てられた要件がレビューされ、以下の項目が判断される：
 - ソフトウェアでの実装が可能かつ適切である
 - 記述が明確で適当である
 - 互いに首尾一貫している
 - テスト可能である
3. 割り当てられた要件について、潜在的な問題を有すると特定された場合、システム要件の分析および割り当てに責任を持つグループによるレビューが実施され、必要な変更が行われる。
4. 割り当てられた要件から派生するコミットメントについては、影響を受けるグループとの協議が行われる。

影響を受けるグループの例：

- ソフトウェアエンジニアリング（ソフトウェア設計などのすべてのサブグループを含む）
- ソフトウェア見積り
- システムエンジニアリング
- システムテスト
- ソフトウェア品質保証
- ソフトウェア構成管理
- 契約管理
- 文書化支援

コミットメントの協議を扱うプラクティスについては、『ソフトウェアプロジェクト計画』キープロセスエリアの活動6を参照。

活動2

ソフトウェアエンジニアリンググループは、ソフトウェア計画、作業成果物、および活動の基盤として、割り当てられた要件を使用する。

(活動2)

割り当てられた要件は：

1. 管理され制御されている。

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キーププロセスエリアに記述されている。

2. ソフトウェア開発計画の基盤となる。
3. ソフトウェア要件を開発するための基盤となる。

活動3

割り当てられた要件への変更をレビューし、ソフトウェアプロジェクトに組み込む。

1. 既存のコミットメントへの影響がアセスメントされ、適宜、変更が協議される。
 - 組織外の個人あるいはグループに対するコミットメントの変更については、上級管理層によりレビューされる。

組織外へなされるコミットメントを扱うプラクティスについては、『ソフトウェアプロジェクト計画』キーププロセスエリアの活動4、および『ソフトウェアプロジェクト進捗管理』キーププロセスエリアの活動3を参照。

- 組織内のコミットメントに対する変更は、影響を受けるグループとの間で協議される。

(活動3)

コミットメントの変更に関する協議を扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キープロセスエリアの活動5、6、7、および8を参照。

2. 割り当てられた要件の変更から、結果として生じたソフトウェア計画、作業成果物、および活動に必要な変更に対して、次の条件が満たされている：
- 特定されている
 - 評価されている
 - リスクがアセスメントされている
 - 文書化されている
 - 計画されている
 - 影響を受けるグループや個人に変更情報が伝達されている
 - 完了まで進捗が確認されている

計測と分析

計測1

計測を行い、その結果を使用して、割り当てられた要件管理の活動状況を判断する。

計測値の例：

- 割り当てられた要件のそれぞれの状態
- 割り当てられた要件に対する変更活動
- 割り当てられた要件に対する変更の累積数で、その内訳は、変更提案、変更未処理、変更承認、およびシステムベースラインに組み込まれた変更のそれぞれの総数

履行検証

検証1

割り当てられた要件の管理活動は、上級管理層によって定期的にレビューされる。

上級管理層による定期的なレビューの主目的は、ソフトウェアプロセス活動に対する認識と見通しを上級管理層に提供することである。活動のレビューは、適切な抽象レベルでタイムリーに行う。レビューの間隔は、組織のニーズに合わせる。例外報告への体制が十分整っていれば、上級管理層のレビュー間隔は長くても構わない。

上級管理層の監督レビューに関して、その典型的な内容を扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証1を参照。

検証2

割り当てられた要件の管理活動は、プロジェクトマネージャによって定期的に、かつイベント発生を契機としてレビューされる。

プロジェクト管理の監督レビューに関して、その典型的な内容を扱うプラクティスについては『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証2を参照。

検証3

ソフトウェア品質保証グループは、割り当てられた要件の管理活動と作業成果物をレビューかつ/または監査し、その結果を報告する。

『ソフトウェア品質保証』キープロセスエリアを参照。

(検証3)

レビューや監査では、最低限、以下のことを検証する：

1. 割り当てられた要件がレビューされ、ソフトウェアエンジニアリンググループがそれらをコミットする前に、問題が解決されている。
2. 割り当てられた要件が変更された時、ソフトウェア計画、作業成果物、および活動が適切に更新されている。
3. 割り当てられた要件の変更から生じたコミットメントの変更について、影響を受けるグループと協議が行われている。

ソフトウェアプロジェクト計画

レベル2のキープロセスエリア： 反復できる段階

『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。

『ソフトウェアプロジェクト計画』に必然的に含まれる活動には、実施すべき作業を見積ること、必要なコミットメントを確立すること、および作業を実施するための計画を定義することが挙げられる。

ソフトウェア計画は、実施すべき作業内容記述書、およびソフトウェアプロジェクトを定義しその境界を定めるその他の制約と目標（これらは『要件管理』キープロセスエリアのプラクティスによって確立される）から始まる。ソフトウェア計画プロセスには、ソフトウェア作業成果物の規模や必要な資源の見積り、スケジュールの作成、ソフトウェアリスクの特定とアセスメント、およびコミットメントの協議といったステップが含まれる。ソフトウェアプロジェクトの計画（ソフトウェア開発計画）を確立するには、これらのステップを繰り返し実施する必要があるかもしれない。

この計画は、ソフトウェアプロジェクト活動を実施し、管理するための基盤を提供するとともに、ソフトウェアプロジェクトの顧客に対しては、ソフトウェアプロジェクトの資源、制約、および能力に応じたコミットメントを明示する。

ゴール

- 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。

実施のコミットメント

- コミットメント1 コミットメントの協議と、プロジェクトのソフトウェア開発計画の策定に対し、責任を持つプロジェクトソフトウェアマネージャが任命される。
- コミットメント2 プロジェクトは、ソフトウェアプロジェクトの計画に関して、明文化された組織方針に従う。

この方針で明記される典型的な事項：

1. 「ソフトウェアに割り当てられたシステム要件」をソフトウェアプロジェクト計画の基盤として使用する。

『要件管理』キープロセスエリアの活動2を参照。

2. ソフトウェアプロジェクトのコミットメントは、次の関係者間で調整する：
 - プロジェクトマネージャ
 - プロジェクトソフトウェアマネージャ

(コミットメント2) その他のソフトウェアマネージャ

3. 他のエンジニアリンググループのソフトウェア活動への関与は、これらのグループとともに協議し文書化する。

他のエンジニアリンググループの例：

- システムエンジニアリング
- ハードウェアエンジニアリング
- システムテスト

4. 影響を受けるグループはソフトウェアプロジェクトの以下の項目をレビューする：

- ソフトウェアの規模見積り
- 工数とコストの見積り
- スケジュール
- その他コミットメント

影響を受けるグループの例：

- ソフトウェアエンジニアリング（ソフトウェア設計などのすべてのサブグループを含む）
- ソフトウェア見積り
- システムエンジニアリング
- システムテスト
- ソフトウェア品質保証
- ソフトウェア構成管理
- 契約管理
- 文書化支援

5. 上級管理層は、組織外の個人とグループになされたソフトウェアプロジェクトのすべてのコミットメントをレビューする。

6. プロジェクトのソフトウェア開発計画は、管理され制御されている。

(コミットメント2)

『ソフトウェア開発計画』という用語は、これらのプラクティスの中で使用され、ソフトウェアプロジェクトを管理するための全体計画を指している。『開発』という用語は、ソフトウェアの保守や支援のプロジェクトを意図的に除外するものではなく、個別のプロジェクトの状況に応じ、適切に解釈されるべきである。

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キーププロセスエリアに記述されている。

実施能力

能力1

ソフトウェアプロジェクトに関して、文書化され承認された作業内容記述書が存在する。

1. 作業内容記述書で扱う項目：

- 作業の範囲
- 技術上の目標とねらい
- 顧客とエンドユーザの特定

これらのプラクティスで意味するエンドユーザとは、顧客の指定するエンドユーザ、またはエンドユーザの代表者をいう。

- 課せられた標準
- 割り当てられた責任

(能力1)

- コストとスケジュールの制約と目標
- ソフトウェアプロジェクトと他の組織との依存関係

他の組織の例：

- 顧客
- 外注先
- ジョイントベンチャーパートナー

- 資源の制約と目標
- 開発や保守に関する他の制約と目標

2. 作業内容記述書は以下の関係者によってレビューされる：

- プロジェクトマネージャ
- プロジェクトソフトウェアマネージャ
- 他のソフトウェアマネージャ
- 他の影響を受けるグループ

3. 作業内容記述書は、管理され制御されている。

能力2

ソフトウェア開発計画策定の責任が割り当てられている。

1. プロジェクトのソフトウェアマネージャは、直接に、あるいは権限を委譲して、プロジェクトのソフトウェア計画作業を調整する。
2. ソフトウェア作業成果物と活動に対する責任は分割され、ソフトウェアマネージャ達に対して、追跡と説明が可能となるように割り当てられている。

(能力2)

ソフトウェア作業成果物の例：

- 外部の顧客やエンドユーザなどに適宜納入される作業成果物
- 他のエンジニアリンググループで使用される作業成果物
- ソフトウェアエンジニアリンググループ内部で使用される主要な作業成果物

能力3

ソフトウェアプロジェクトを計画するために、適切な資源と資金が提供される。

1. 計画対象のソフトウェアプロジェクトのアプリケーションドメインに専門知識を持つ経験者がいるのであれば、彼らをソフトウェア開発計画の策定に利用できる。
2. ソフトウェアプロジェクトの計画活動を支援するツールが利用可能である。

支援ツールの例：

- 表計算プログラム
- 見積りモデル
- プロジェクトの計画作成やスケジュール作成のためのプログラム

能力4

ソフトウェアプロジェクトの計画作業に関わるソフトウェアマネージャ、ソフトウェアエンジニア、およびその他の担当者は、各自の責任領域に適用されるソフトウェアの見積りや計画作業の手順に関するトレーニングを受ける。

実施される活動

活動1

ソフトウェアエンジニアリンググループは、プロジェクト提案チームに参加する。

(活動1)

1. ソフトウェアエンジニアリンググループは以下の活動にかかわる：
 - 提案を準備し提出する
 - 明確化の議論をし提出する
 - ソフトウェアプロジェクトに影響を与えるコミットメントの変更に関する協議を行う
2. ソフトウェアエンジニアリンググループはプロジェクトが提案するコミットメントをレビューする。

プロジェクトのコミットメントの例：

- プロジェクトの技術上の目標とねらい
- システムおよびソフトウェアの技術的解決策
- ソフトウェアに関する予算、スケジュール、および資源
- ソフトウェアに関する標準と手順

活動2

ソフトウェアプロジェクト計画の策定は、プロジェクト全体計画の早期段階から、かつ並行して開始する。

活動3

ソフトウェアエンジニアリンググループは、他の影響を受けるグループとともに、プロジェクトの全期間にわたってプロジェクト全体計画に参加する。

1. ソフトウェアエンジニアリンググループは、プロジェクトレベルの計画をレビューする。

活動4

上級管理層は、組織外の個人とグループに対してなされたソフトウェアプロジェクトのコミットメントを、文書化された手順に従ってレビューする。

活動5

あらかじめ定義された管理可能な大きさのステージを持つソフトウェアライフサイクルを特定または定義する。

(活動5)

ソフトウェアライフサイクルの例：

- ウォータフォール
- オーバーラップしたウォータフォール
- スパイラル
- シリアルビルド
- 単一プロトタイプ/オーバーラップしたウォータフォール

活動6

プロジェクトのソフトウェア開発計画は、文書化された手順に従って策定する。

この手順で明記される典型的な事項：

1. ソフトウェア開発計画は、以下を基盤としかつ準拠する：
 - 顧客の標準など、適宜
 - プロジェクトの標準
 - 承認された作業内容記述書
 - 割り当てられた要件
2. ソフトウェア関連グループおよびその他のエンジニアリンググループがソフトウェアエンジニアリンググループの活動に参加することについての計画は、それらのグループ間で協議し、支援工数を予算化し、そして合意事項を文書化する。

ソフトウェア関連グループの例：

- ソフトウェア品質保証
- ソフトウェア構成管理
- 文書化支援

(活動6)

その他のエンジニアリンググループの例：

- システムエンジニアリング
- ハードウェアエンジニアリング
- システムテスト

3. ソフトウェアエンジニアリンググループがその他のソフトウェア関連グループおよびその他のエンジニアリンググループの活動に参加することについての計画は、それらのグループ間で調整し、支援工数を予算化し、そして合意事項を文書化する。
4. 以下の関係者はソフトウェア開発計画をレビューする：
 - プロジェクトマネージャ
 - プロジェクトソフトウェアマネージャ
 - 他のソフトウェアマネージャ
 - 他の影響を受けるグループ
5. ソフトウェア開発計画は、管理され制御されている。

活動7

ソフトウェアプロジェクト計画を文書化する。

これらのキープラクティスでは、この計画、あるいは計画を集めたものを、ソフトウェア開発計画という。

ソフトウェアプロジェクト計画のプロジェクトでの使用に関するプラクティスについては、『ソフトウェアプロジェクト進捗管理』キープロセスエリアの活動1を参照。

ソフトウェア開発計画が扱う事項：

1. ソフトウェアプロジェクトの目的、範囲、目標、およびねらい。

2. ソフトウェアライフサイクルの選択。
- (活動7) 3. ソフトウェアの開発や保守のために選択した手順、手法、および標準の特定。

ソフトウェア標準および手順の例：

- ソフトウェア開発計画策定
- ソフトウェア構成管理
- ソフトウェア品質保証
- ソフトウェア設計
- 問題の進捗確認と解決
- ソフトウェア計測

4. 開発すべきソフトウェア作業成果物の特定。
5. ソフトウェア作業成果物およびソフトウェア作業成果物に対する変更の規模見積り。
6. ソフトウェアプロジェクトの工数およびコスト見積り。
7. 重要なコンピュータ資源の使用についての見積り。
8. ソフトウェアプロジェクトのスケジュール。これにはマイルストーンやレビューの特定も含む。
9. プロジェクトのソフトウェアリスクの特定とアセスメント。
10. プロジェクトのソフトウェアエンジニアリング設備と支援ツールに関する計画。

- 活動8 **ソフトウェアプロジェクトの制御を確立し維持するために必要なソフトウェア作業成果物を特定する。**

『ソフトウェア構成管理』キープロセスエリアの活動4を参照。

活動9

ソフトウェア作業成果物の規模（またはソフトウェア作業成果物の変更規模）の見積りは、文書化された手順に従って算出する。

この手順で明記される典型的な事項：

1. すべての主要なソフトウェア作業成果物および活動について、規模の見積りを行う。

ソフトウェア規模の計測値の例：

- ファンクションポイント
- フィーチャポイント
- コード行数
- 要件の数
- ページ数

規模見積りの対象となる作業成果物および活動タイプの例：

- 実稼動ソフトウェアと支援ソフトウェア
- 納入する作業成果物と納入しない作業成果物
- ソフトウェアと非ソフトウェア作業成果物（例えば、文書）
- 作業成果物の開発活動、検証活動、および妥当性確認活動

2. 見積りのねらいを満たす程度の粒度に、ソフトウェア作業成果物を分解する。
3. 利用可能な場合は過去のデータを使用する。

4. 規模見積りにおける仮定を文書化する。
5. 規模見積り結果は文書化し、レビューし、そして合意を得る。

(活動9)

規模見積りをレビューし合意するグループや個人の例：

- プロジェクトマネージャ
- プロジェクトソフトウェアマネージャ
- その他のソフトウェアマネージャ

活動10

ソフトウェアプロジェクトの工数とコストの見積りは、文書化された手順に従って算出する。

この手順で明記される典型的な事項：

1. ソフトウェアプロジェクトの工数やコストの見積りは、ソフトウェア作業成果物の規模（あるいはその変更規模）の見積りと関係づける。
2. 利用可能な場合は、生産性データ（過去のデータや現在のデータ）を見積りに使用する。これらのデータの出所および論理的根拠を文書化する。
 - 可能であれば、生産性およびコストのデータは、その組織のプロジェクトから得られるものを使用する。
 - 生産性およびコストのデータは、ソフトウェア作業成果物の作成に含まれる工数ならびに重要コストを考慮する。

ソフトウェア作業成果物の作成に含まれる重要コストの例：

- 直接人件費
- 間接費
- 旅費や交通費
- コンピュータ使用経費

3. 工数、配員、およびコストの見積りは過去の経験を基盤にする。
- 可能な時は類似のプロジェクトを使用する。
 - 活動を時間軸上で分割する。
- (活動10)
- 工数、配員、およびコストの見積り値を、ソフトウェアライフサイクル全体に配分する。
4. 見積りや見積り作成で使用した仮定を文書化し、レビューし、そして合意を得る。

活動11

プロジェクトの重要なコンピュータ資源の見積りは、文書化された手順に従って算出する。

重要なコンピュータ資源は、ホスト環境中、統合およびテスト環境中、ターゲット環境中、またはこれらを組み合わせた環境の中に存在することがある。

この手順で明記される典型的な事項：

1. プロジェクトの重要なコンピュータ資源を特定する。

重要なコンピュータ資源の例：

- コンピュータメモリの容量
- コンピュータプロセッサの使用
- 通信チャンネルの容量

2. 重要なコンピュータ資源の見積りには、以下の見積りを関係づける：
- ソフトウェア作業成果物の規模
 - 実稼動時の処理負荷
 - 通信トラフィック
3. 重要なコンピュータ資源の見積りを文書化し、レビューし、そして合意を得る。

活動12 **プロジェクトのソフトウェアスケジュールは、文書化された手順に従って策定する。**

(活動12) この手順で明記される典型的な事項：

1. ソフトウェアスケジュールには、以下を関係づける：
 - ソフトウェア作業成果物の規模（あるいはその変更規模）の見積り
 - ソフトウェアの工数とコスト
2. ソフトウェアスケジュールは、過去の経験を基盤とする。
 - 可能な場合は、類似プロジェクトを使用する。
3. ソフトウェアスケジュールは、課せられたマイルストーン期日、重要な依存関係のある期日、およびその他の制約に適応させる。
4. 正確な進捗計測を支援するため、ソフトウェアスケジュール上の活動は適切な持続の長さを持ち、マイルストーンは適切な時間間隔を持つようにする。
5. スケジュール策定の根拠となる仮定を文書化する。
6. ソフトウェアスケジュールを文書化し、レビューし、そして合意を得る。

活動13 **プロジェクトのコスト、資源、スケジュール、および技術面にかかわるソフトウェアのリスクを特定し、アセスメントし、そして文書化する。**

1. リスクは、プロジェクトに対する潜在的な影響を基に分析し、優先順位をつける。
2. リスクについてその対応策を特定する。

リスク対応策の例：

- スケジュール上のバッファ
- 配員の代替計画
- コンピュータ機器追加の代替計画

活動14

プロジェクトのソフトウェアエンジニアリング設備および支援ツールに関する計画を作成する。

1. これらの設備や支援ツールの容量の要件の見積りは、ソフトウェア作業成果物の規模見積りやその他の特徴を基盤とする。

ソフトウェア開発設備や支援ツールの例：

- ソフトウェア開発用のコンピュータおよび周辺機器
- ソフトウェアテスト用のコンピュータおよび周辺機器
- ターゲットコンピュータ環境のソフトウェア
- その他の支援ソフトウェア

2. これらの設備や支援ツールの調達や開発について、責任を分担しコミットメントを協議する。
3. 計画は、すべての影響を受けるグループによってレビューされる。

活動15

ソフトウェア計画策定に関するデータを記録する。

1. 記録される情報には、見積りの結果、およびその見積りを再構築し妥当性をアセスメントするのに必要な関連情報といったものがある。
2. ソフトウェア計画策定データは、管理され制御されている。

計測と分析

計測1 計測を行い、その結果を使用して、ソフトウェア計画策定の活動状況を判断する。

(計測1)

計測値の例：

- ソフトウェアプロジェクト計画活動のマイルストーンの完了で、計画と比較したもの
- ソフトウェアプロジェクト計画活動の完了した作業、工数、および予算使用額で、計画と比較したもの

履行検証

検証1 ソフトウェアプロジェクト計画の活動は、上級管理層によって定期的にレビューされる。

上級管理層による定期的なレビューの主目的は、ソフトウェアプロセス活動に対する認識と見通しを上級管理層に提供することである。活動のレビューは、適切な抽象レベルでタイムリーに行う。レビューの間隔は、組織のニーズに合わせる。例外報告への体制が十分整っていれば、上級管理層のレビュー間隔は長くても構わない。

1. 技術、コスト、配員、およびスケジュールの各面での実績がレビューされる。
2. 低いレベルでは解決できない対立や課題が取り上げられる。
3. ソフトウェアプロジェクトのリスクが取り上げられる。
4. 処置項目が割り当てられ、レビューされ、決着まで進捗が確認される。

5. 各会議の要約報告書が作成され、影響を受けるグループと個人に配布される。

検証2 **ソフトウェアプロジェクト計画の活動は、プロジェクトマネージャによって定期的に、かつイベント発生を契機としてレビューされる。**

(検証2)

1. 影響を受けるグループが代表されている。
2. ソフトウェアプロジェクト計画活動の状況と現在の結果が、ソフトウェアプロジェクトの作業内容記述書および割り当てられた要件に対してレビューされる。
3. グループ間の依存関係が取り上げられる。
4. 低いレベルでは解決できない対立や課題が取り上げられる。
5. ソフトウェアプロジェクトのリスクがレビューされる。
6. 処置項目が割り当てられ、レビューされ、決着まで進捗が確認される。
7. 各会議の要約報告書が作成され、影響を受けるグループと個人に配布される。

検証3 **ソフトウェア品質保証グループは、ソフトウェアプロジェクト計画の活動と作業成果物をレビューかつ/または監査し、その結果を報告する。**

『ソフトウェア品質保証』キープロセスエリアを参照。

レビューや監査では、最低限、以下のことを検証する：

1. ソフトウェア見積りと計画策定に関する活動。
2. プロジェクトのコミットメントのレビューおよび確立に関する活動。

3. ソフトウェア開発計画の準備に関する活動。
4. ソフトウェア開発計画の準備に使用された標準。
5. ソフトウェア開発計画の内容。

ソフトウェアプロジェクト進捗管理

レベル2のキーププロセスエリア：反復できる段階

『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。

『ソフトウェアプロジェクト進捗管理』に必然的に含まれる活動には、ソフトウェアの成果と結果について、文書化された見積り、コミットメント、および計画に照らして進捗の確認を行いレビューすること、および計画を実際の成果や結果を踏まえて補正することが挙げられる。

ソフトウェアプロジェクトの文書化された計画（即ち、『ソフトウェアプロジェクト計画』キーププロセスエリアで記述するようなソフトウェア開発計画）が、ソフトウェア活動の進捗確認、状況伝達、および計画改訂の基盤として使用される。ソフトウェア活動は管理層によりモニターされる。進捗は、本来、選択されたソフトウェア作業成果物が完成したときおよび選択されたマイルストーンにおいて、実際のソフトウェアの規模、工数、コスト、およびスケジュールを計画と比較することにより判断される。ソフトウェアプロジェクトが計画通りに進んでいないと判断されたときは、是正処置がとられる。是正処置としては、実際の成果を反映させてソフトウェア開発計画を改訂すること、残っている作業を再計画すること、あるいは行動を改善するための処置をとることなどがある。

ゴール

- 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。

実施のコミットメント

- コミットメント1 プロジェクトのソフトウェア活動と結果に対し、責任を持つプロジェクトソフトウェアマネージャが任命される。
- コミットメント2 プロジェクトは、ソフトウェアプロジェクトの管理に関して、明文化された組織方針に従う。

この方針で明記される典型的な事項：

1. ソフトウェアプロジェクトの進捗を確認する基盤として、文書化されたソフトウェア開発計画を使用し保守する。
2. ソフトウェアプロジェクトの状況と課題についての情報が、常にプロジェクトマネージャに届く。
3. ソフトウェア計画が達成されそうにないときは、行動の補正または計画の補正により是正処置をとる。
4. ソフトウェアコミットメントの変更は、影響を受けるグループの参加と合意のもとに行う。

(コミットメント2)

影響を受けるグループの例：

- ソフトウェアエンジニアリング（ソフトウェア設計などのすべてのサブグループを含む）
- ソフトウェア見積り
- システムエンジニアリング
- システムテスト
- ソフトウェア品質保証
- ソフトウェア構成管理
- 契約管理
- 文書化支援

5. 上級管理層は、組織外の個人とグループになされたすべてのコミットメントの変更と、ソフトウェアプロジェクトの新たなコミットメントをレビューする。

実施能力

能力1

ソフトウェアプロジェクトのソフトウェア開発計画は、文書化され承認されている。

ソフトウェア開発計画を扱うキープラクティスについては、『ソフトウェアプロジェクト計画』キーププロセスエリアの活動6と活動7を参照。

能力2

プロジェクトソフトウェアマネージャは、ソフトウェア作業成果物および活動に対する責任を明示的に割り当てる。

責任の割り当てで扱う項目：

1. 開発されるソフトウェア作業成果物、あるいは提供されるサービス

- (能力2)
2. ソフトウェア活動の工数とコスト
 3. ソフトウェア活動のスケジュール
 4. ソフトウェア活動の予算

能力3 **ソフトウェアプロジェクトの進捗を確認するために、適切な資源と資金が提供される。**

1. ソフトウェアマネージャとソフトウェアタスクリーダーに、ソフトウェアプロジェクトの進捗を確認するための明確な責任が割り当てられる。
2. ソフトウェアの進捗確認を支援するツールが利用可能になっている。

支援ツールの例：

- 表計算プログラム
- プロジェクトの計画作成やスケジュール作成のためのプログラム

能力4 **ソフトウェアマネージャは、ソフトウェアプロジェクトの技術面ならびに人事面の管理に関するトレーニングを受ける。**

トレーニングの例：

- 技術的なプロジェクトの管理
- ソフトウェアの規模、工数、コスト、およびスケジュールの進捗管理
- 人の管理

能力5 **第一線ソフトウェアマネージャは、ソフトウェアプロジェクトの技術面のオリエンテーションを受ける。**

(能力5)

オリエンテーションの例：

- プロジェクトのソフトウェアエンジニアリングに関する標準と手順
- プロジェクトのアプリケーションドメイン

実施する活動

活動1

ソフトウェア活動の進捗を確認し状況を伝達するために、文書化されたソフトウェア開発計画を使用する。

ソフトウェア開発計画の内容を扱うキープラクティスについては、『ソフトウェアプロジェクト計画』キーププロセスエリアの活動7を参照。

このソフトウェア開発計画は：

1. 成果を反映するように、作業の進捗に合わせて更新される。特にマイルストーン完了時には更新される。
2. 以下の関係者には直ちに利用可能である：
 - ソフトウェアエンジニアリンググループ（ソフトウェア設計のようなすべてのサブグループを含む）
 - ソフトウェアマネージャ
 - プロジェクトマネージャ
 - 上級管理層
 - その他の影響を受けるグループ

活動2

プロジェクトのソフトウェア開発計画は、文書化された手順に従って改訂する。

(活動2)

ソフトウェア開発計画作成のための活動を扱うプラクティスについては、『ソフトウェアプロジェクト計画』キープロセスエリアの活動6を参照。

この手順で明記される典型的な事項：

1. 適宜、特に計画に著しい変更があった場合は、ソフトウェア開発計画を改訂し、計画の改良点と変更点を取り入れる。

計画に対するすべての変更においては、「ソフトウェアに割り当てられたシステム要件」、設計の制約、資源、コスト、およびスケジュール間の相互依存関係を反映させる必要がある。

2. ソフトウェア開発計画を更新し、すべての新しいソフトウェアプロジェクトコミットメントおよびコミットメントに対する変更を取り入れる。
3. ソフトウェア開発計画は、改訂毎にレビューされる。
4. ソフトウェア開発計画は、管理され制御されている。

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キープロセスエリアに記述されている。

活動3 上級管理層は、組織外のグループと個人に対するソフトウェアプロジェクトのコミットメントおよびコミットメントに対する変更を、文書化された手順に従ってレビューする。

活動4 承認されたコミットメントの変更がソフトウェアプロジェクトに影響をおよぼす場合、その変更をソフトウェアエンジニアリンググループおよび他のソフトウェア関連グループメンバに伝達する。

他のソフトウェア関連グループの例：

- ソフトウェア品質保証
- ソフトウェア構成管理
- 文書化支援

活動5 ソフトウェア作業成果物の規模（あるいはその変更規模）についての進捗を確認し、必要に応じて是正処置をとる。

規模見積りの算出を扱うプラクティスについては、『ソフトウェアプロジェクト計画』キープロセスエリアの活動9を参照。

1. すべての主要なソフトウェア作業成果物の規模（あるいはその変更規模）についての進捗を確認する。
2. コード（生成されたコード、十分にテストされたコード、および納入されるコード）の実際の規模を、ソフトウェア開発計画に文書化されている見積り規模と比較する。
3. 納入文書の実際の数量を、ソフトウェア開発計画に文書化されている見積りと比較する。
4. ソフトウェア作業成果物の全体的な予想規模（実績値と組み合わせた見積り）を、定期的に、改良し、モニターし、そして補正する。

- (活動5) 5. ソフトウェアのコミットメントに影響するソフトウェア作業成果物の見積り規模の変更は、影響を受けるグループと協議され文書化される。

活動6 **プロジェクトのソフトウェア工数とコストについての進捗を確認し、必要に応じて是正処置をとる。**

コスト見積りの算出を扱うキープラクティスについては、『ソフトウェアプロジェクト計画』キーププロセスエリアの活動10を参照。

1. 工数とコストについて、これまでの実際の消費量および完了した作業に対する実際の消費量を、ソフトウェア開発計画に文書化されている見積りと比較し、潜在的な超過および未達の可能性を特定する。
2. ソフトウェアコストの進捗を確認し、ソフトウェア開発計画に文書化されている見積りと比較する。
3. 工数と配員を、ソフトウェア開発計画に文書化されている見積りと比較する。
4. ソフトウェアのコミットメントに影響する配員やその他のソフトウェアコストの変更は、影響を受けるグループと協議され文書化される。

活動7 **プロジェクトの重要なコンピュータ資源の進捗を確認し、必要に応じて是正処置をとる。**

コンピュータ資源見積りの算出を扱うプラクティスについては、『ソフトウェアプロジェクト計画』キーププロセスエリアの活動11を参照。

1. プロジェクトにおける重要コンピュータ資源の実際の使用量と予想される使用量の進捗を確認し、ソフトウェア開発計画に文書化されている各主要なソフトウェアコンポーネントの見積りと比較する。

- (活動7) 2. ソフトウェアのコミットメントに影響する重要なコンピュータ資源の見積りの変更は、影響を受けるグループと協議され文書化される。

活動8 プロジェクトのソフトウェアスケジュールについての進捗を確認し、必要に応じて是正処置をとる。

スケジュールの導出を扱うキープラクティスについては、『ソフトウェアプロジェクト計画』キーププロセスエリアの活動12を参照。

1. ソフトウェア活動、マイルストーン、およびその他のコミットメントについて、実際の完了状況をソフトウェア開発計画と比較する。
2. ソフトウェア活動、マイルストーン、およびその他のコミットメントについて、その完了が遅れた場合と早まった場合の影響を、今後の活動やマイルストーンに対する影響の点から評価する。
3. ソフトウェアのコミットメントに影響するソフトウェアスケジュールの改訂は、影響を受けるグループと協議され文書化される。

活動9 ソフトウェアエンジニアリング技術活動の進捗を確認し、必要に応じて是正処置をとる。

1. ソフトウェアエンジニアリンググループのメンバは、技術面の状況を第一線マネージャに定期的に報告する。
2. 一連のビルドに関するソフトウェアリリースの内容を、ソフトウェア開発計画に文書化されている計画と比較する。
3. ソフトウェア作業成果物で特定された問題を報告し文書化する。
4. 問題報告は、決着まで進捗が確認される。

2. プロジェクトソフトウェアマネージャ、第一線ソフトウェアマネージャ、およびその他のソフトウェアマネージャなど、適宜

活動13

ソフトウェアプロジェクトの成果と結果を取り上げるための公式レビューは、文書化された手順に従って、選定されたプロジェクトのマイルストーンで開催する。

これらのレビューは：

1. 選定されたステージの開始と終了時といった、ソフトウェアプロジェクトのスケジュール上の重要な時点で行うように計画する。
2. 顧客、エンドユーザ、および組織内の影響を受けるグループなどが適宜参加して行われる。

これらのプラクティスで意味するエンドユーザとは、顧客の指名するエンドユーザ、またはエンドユーザの代表者をいう。

3. 責任あるソフトウェアマネージャがレビューし承認した資料を使用する。
4. ソフトウェア活動のコミットメント、計画、および状況を取り上げる。
5. 結果として、重要な課題、処置項目、および決定事項が特定され文書化される。
6. ソフトウェアプロジェクトのリスクを取り上げる。
7. 結果として、必要に応じてソフトウェア開発計画を改良する。

計測と分析

計測1 **計測を行い、その結果を使用して、ソフトウェア進捗管理の活動状況を判断する。**

(計測1)

計測値の例：

- 進捗管理活動の実施に費やされた工数とその他の資源
- ソフトウェア開発計画書の変更活動で、その内訳は、ソフトウェア作業成果物の規模見積り、ソフトウェアコスト見積り、重要なコンピュータ資源の見積り、およびスケジュールに対するそれぞれの変更。

履行検証

検証1 **ソフトウェアプロジェクト進捗管理の活動は、上級管理層によって定期的にレビューされる。**

上級管理層による定期的なレビューの主目的は、ソフトウェアプロセス活動に対する認識と見通しを上級管理層に提供することである。活動のレビューは、適切な抽象レベルでタイムリーに行う。レビューの間隔は、組織のニーズに合わせる。例外報告への体制が十分整っていれば、上級管理層のレビュー間隔は長くても構わない。

1. 技術、コスト、配員、およびスケジュールの各面での実績がレビューされる。
2. 低いレベルでは解決できない対立や課題が取り上げられる。
3. ソフトウェアプロジェクトのリスクが取り上げられる。

4. 処置項目が割り当てられ、レビューされ、決着まで進捗が確認される。
5. 各会議の要約報告書が作成され、影響を受けるグループと個人に配布される。

検証2

ソフトウェアプロジェクト進捗管理の活動は、プロジェクトマネージャによって定期的に、かつイベント発生を契機としてレビューされる。

1. 影響を受けるグループが代表されている。
2. 技術、コスト、配員、およびスケジュールの各面での実績が、ソフトウェア開発計画に対してレビューされる。
3. 重要なコンピュータ資源の使用状況がレビューされる；これらの重要なコンピュータ資源に関する現在の見積りと実際の使用状況が、もともとの見積りに対して報告される。
4. グループ間の依存関係が取り上げられる。
5. 低いレベルでは解決できない対立や課題が取り上げられる。
6. ソフトウェアプロジェクトのリスクが取り上げられる。
7. 処置項目が割り当てられ、レビューされ、決着まで進捗が確認される。
8. 各会議の要約報告書が作成され、影響を受けるグループに配布される。

検証3

ソフトウェア品質保証グループは、ソフトウェアプロジェクト進捗管理の活動と作業成果物をレビューかつ/または監査し、その結果を報告する。

『ソフトウェア品質保証』キープロセスエリアを参照。

レビューや監査では、最低限、以下のことを検証する：

1. コミットメントのレビューや改訂に関する活動。
2. ソフトウェア開発計画の改訂に関する活動。

(検証3)

3. 改訂されたソフトウェア開発計画の内容。
4. ソフトウェアプロジェクトにおける、コスト、スケジュール、リスク、技術と設計上の制約、機能充足性、ならびに性能に関する進捗確認に関する活動。
5. 計画された技術および管理のレビュー開催に関する活動。

ソフトウェア外注管理

レベル2のキープロセスエリア： 反復できる段階

『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』に必然的に含まれる活動には、ソフトウェア外注先を選定すること、外注先とコミットメントを確立すること、および外注先の行動と結果の進捗を確認しレビューすることが挙げられる。これらのプラクティスでは、ソフトウェア、ハードウェア、およびその他のシステム構成要素を包含するようなソフトウェア構成要素についての外注管理だけでなく、ソフトウェア（単独の）外注管理も扱う。

外注先は、その作業の実施能力を基準にして選定される。外部発注元（以下、本KPAでは「発注元」という）の作業の一部を外注する場合、その決定にはさまざまな要因が関係する。技術的な考慮だけでなく、戦略的な業務提携を基盤として外注先が選定されることもある。このキープロセスエリアのプラクティスでは、作業の定義された部分を他の組織へ外注することに関連した、従来からの調達プロセスについて取り上げる。

外注に当たっては、技術要件と非技術要件（例えば、納入日）を取り扱う文書化された合意が確立される。この合意は、外注管理の基盤として使用される。外注先で行われる作業とその作業に対する計画が文書化される。

外注先が従うべき標準は、発注元の標準と適合しているものとする。

外注された作業に関するソフトウェア計画および進捗管理の活動は、外注先が実施する。発注元は、これらの計画および進捗管理の活動が適切に実施されていること、そして外注先から納入されたソフトウェア成果物が検収基準を満足することを確実なものにする。発注元は、それらの成果物とプロセスのインタフェースの管理に関して、外注先と協力して作業をする。

ゴール

- 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。

実施のコミットメント

- コミットメント1 プロジェクトは、ソフトウェア外注管理に関して、明文化された組織方針に従う。

(コミットメント1) この方針で明記される典型的な事項：

1. ソフトウェア外注先の選定とソフトウェア外注管理に、文書化された標準や手順が使用される。
2. 契約条項を外注管理の基盤とする。
3. 外注に関する変更は、発注元と外注先双方の参加と合意のもとに行われる。

コミットメント2 ソフトウェア外注の確立と管理のため、責任者となる外注管理者が任命される。

1. 外注管理者は、ソフトウェアエンジニアリングの知識がある経験者であるか、あるいは知識と経験のある補佐役を持つ。
2. 外注管理者は、外注作業の技術範囲と外注条件について、影響を受ける当事者ととも調整する責任を負う。

プロジェクトのシステムエンジニアリンググループとソフトウェアエンジニアリンググループが外注作業の技術範囲を定義する。

調達、財務、および法務などの適切な機能担当グループは、外注契約条項を確立しモニターする。

3. 外注管理者に課される責任：
 - ソフトウェア外注先の選定
 - ソフトウェア外注管理
 - 外注成果物に対する外注契約終了後の支援に関する手配

実施能力

能力1 **ソフトウェア外注先の選定と外注管理のために、適切な資源と資金が提供される。**

1. ソフトウェアマネージャやその他の個人に、外注管理のための明確な責任が割り当てられる。
2. 外注管理を支援するツールが利用可能になっている。

支援ツールの例：

- 見積りモデル
- 表計算プログラム
- プロジェクト管理とスケジュール作成に関するプログラム

能力2 **ソフトウェア外注の確立と管理を行うソフトウェアマネージャやその他の個人は、これらの活動を実施するためのトレーニングを受ける。**

トレーニングの例：

- ソフトウェア外注の準備と計画
- 入札者のソフトウェアプロセス能力の評価
- 入札者のソフトウェア見積りや計画の評価
- 外注先の選定
- 外注管理

能力3 **ソフトウェア外注管理を行うソフトウェアマネージャやその他の個人は、外注契約の技術面に関するオリエンテーションを受ける。**

(能力3)

オリエンテーションの例：

- 当該アプリケーションドメイン
- 適用されるソフトウェア技術
- 用いられるソフトウェアツール
- 用いられる方法論
- 用いられる標準
- 用いられる手順

実施される活動

活動1

外注する作業は、文書化された手順に従って定義し計画する。

この手順で明記される典型的な事項：

1. 外注するソフトウェア成果物や活動は、プロジェクトの技術面と非技術面の特徴についてバランスをとったアセスメント結果を基盤として選定される。
 - 外注する機能やサブシステムは、外注先候補のスキルや能力に合わせて設定される。
 - 外注するソフトウェア成果物や活動の仕様は、システムやソフトウェア要件の体系的な分析ならびに適切な分割を基盤として決定される。
2. 外注する作業の仕様ならびに適用される標準や手順は、プロジェクトの持つ下記の項目から導出される：
 - 作業内容記述書
 - 「ソフトウェアに割り当てられたシステム要件」
 - ソフトウェア要件
 - ソフトウェア開発計画
 - ソフトウェア標準と手順
3. 外注の作業内容記述書は：

(活動1)

- 作成される
- レビューされる
- 合意される

外注の作業内容記述書をレビューし合意する担当者の例：

- プロジェクトマネージャ
- プロジェクトソフトウェアマネージャ
- 責任のあるソフトウェアマネージャ
- ソフトウェア構成管理マネージャ
- ソフトウェア品質保証マネージャ
- 外注管理者

- 必要に応じて改訂される
- 管理され制御されている

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キープロセスエリアに記述されている。

作業内容記述書の典型的な内容を扱うプラクティスについては、『ソフトウェアプロジェクト計画』キープロセスエリアの能力1を参照。

4. 外注先の選定計画は、外注の作業内容記述書と並行して作成され、適宜レビューされる。

活動2

ソフトウェア外注先は、外注契約入札者の作業実施能力に関する評価を基盤とし、文書化された手順に従って選定する。

この手順では以下の評価項目を扱う：

1. 計画した外注に対して提出された提案
2. もしあれば、過去の類似作業の実績記録
3. 外注入札組織と発注元の所在地の地理関係

外注先の効果的管理には、直接顔を付き合わせた頻繁な会合が必要となることがある。

4. ソフトウェアエンジニアリングとソフトウェア管理の能力

外注先の能力評価方法の例としては、SEIのソフトウェア能力審査法がある。

5. 作業を実施可能な要員
6. 類似のアプリケーションにおける過去の経験で、外注先のソフトウェア管理チームのソフトウェア専門知識を含むもの
7. 利用可能な資源

資源の例：

- 設備
- ハードウェア
- ソフトウェア
- トレーニング

活動3

発注元とソフトウェア外注先との間の契約合意を、外注管理の基盤とする。

契約合意書には以下の内容を記録する：

1. 契約条項
2. 作業内容記述書

作業内容記述書の典型的な内容を扱うプラクティスについては、『ソフトウェアプロジェクト計画』キープロセスエリアの能力1を参照。

3. 開発すべき製品の要件
4. 外注先と発注元間の依存関係一覧
5. 発注元に納入する外注した作業成果物

成果物の例：

- ソースコード
- ソフトウェア開発計画
- シミュレーション環境
- 設計書
- 検収テスト計画

6. 成果物の改訂版を提出する際の条件
7. 発注元が外注した成果物を受け入れる前に、それら进行评估するために用いられる検収手順と検収基準
- (活動3) 8. 外注先の実績をモニターし评估するため、発注元が使用する手順と評価基準

活動4

文書化された外注先のソフトウェア開発計画について、発注元がレビューし承認する。

1. このソフトウェア開発計画は、発注元のソフトウェア開発計画からの適切な項目を（直接にまたは間接に）扱う。

場合によっては、外注先のソフトウェア開発計画は発注元のソフトウェア開発計画の一部に組み込まれており、独立した計画を別途作成する必要がない場合もある。

プロジェクトのソフトウェア開発計画の内容を扱うプラクティスについては、『ソフトウェアプロジェクト計画』キープロセスエリアの活動7を参照。

活動5 文書化され承認された外注先のソフトウェア開発計画を用いて、ソフトウェア活動の進捗を確認し、状況を伝達する。

活動6 ソフトウェア外注先の作業内容記述書、外注契約条項、およびその他のコミットメントに関する変更は、文書化された手順に従って解決する。

1. 典型的には、この手順では、発注元と外注先の双方の影響を受けるすべてのグループが含まれていることを明示する。

活動7 発注元の管理層は、外注先の管理層とともに定期的な状況/調整レビューを開催する。

(活動7) 1. 成果物に関する顧客やエンドユーザのニーズおよび要望について、適宜、外注先に可視性が提供される。

エンドユーザとは、ここでは顧客が指定したエンドユーザ、またはエンドユーザの代表をいう。

2. 技術、コスト、配員、およびスケジュールに関する外注先の実績は、外注先のソフトウェア開発計画に照らしてレビューされる。
3. プロジェクトにとって重要と指定されたコンピュータリソースについてレビューする。つまり、現在の見積りに対する外注先の寄与状況は、外注先のソフトウェア開発計画に書かれている各ソフトウェア構成要素別の見積りに対して、進捗が確認され比較される。
4. 外注先のソフトウェアエンジニアリンググループと他の外注先グループ間における重要な依存関係やコミットメントが取り上げられる。
5. 発注元と外注先との間の重要な依存関係やコミットメントが取り上げられる。

2. レビューでは、ソフトウェア活動に対しての外注先のコミットメント、計画、および活動状況を取り上げる。
3. 重要な課題、処置項目、および決定が特定され文書化される。
4. ソフトウェアリスクを取り上げる。
5. 適宜、外注先のソフトウェア開発計画を改良する。

活動10 **発注元のソフトウェア品質保証グループは、文書化された手順に従って外注先のソフトウェア品質保証の活動をモニターする。**

(活動10) この手順で明記される典型的な事項：

1. ソフトウェア品質保証に関する外注先の計画、資源、手順、および標準を定期的にレビューし、それらが外注先の実績をモニターする上で適切であることを確実なものにする。
2. 外注先との定期的なレビューを開催し、承認された手順や標準に従っていることを確実なものにする。
 - 発注元のソフトウェア品質保証グループは、外注先のソフトウェアエンジニアリング活動および成果物について、抜き取り検査を実施する。
 - 発注元のソフトウェア品質保証グループは、外注先のソフトウェア品質保証記録を適宜監査する。
3. 外注先のソフトウェア品質保証活動記録を定期的に監査し、ソフトウェア品質の計画、標準、および手順にいかに従っているかをアセスメントする。

活動11 **発注元のソフトウェア構成管理グループは、文書化された手順に従って外注先のソフトウェア構成管理活動をモニターする。**

この手順で明記される典型的な事項：

1. ソフトウェア構成管理に関する外注先の計画、資源、手順、および標準をレビューし、適切であることを確実なものにする。
2. 発注元と外注先は、相互のソフトウェア構成管理活動を調整し、外注先の成果物については、発注元のプロジェクト環境への統合や組み込みがたやすく可能なことを確実なものにする。
3. 外注先のソフトウェアベースラインライブラリについて定期的に監査し、ソフトウェア構成管理の標準と手順にいかに従っているか、またそれらが、ソフトウェアベースラインを管理する上でどれほど効果的であるかについてアセスメントする。

活動12

発注元は、外注先のソフトウェア成果物納入の一環として、文書化された手順に従って検収テストを実施する。

この手順で明記される典型的な事項：

1. 検収手順と検収基準は、発注元と外注先の双方により、テストに先立ち成果物個別に定義され、レビューされ、そして承認される。
2. 検収テストの結果は文書化される。
3. 検収テスト不合格のソフトウェア成果物については、処置計画が確立される。

活動13

ソフトウェア外注先の実績を定期的に評価し、評価結果は外注先とともにレビューする。

外注先の実績に関する評価結果は、外注先にとっては顧客（発注元の）ニーズを満足したか否かに関するフィードバックを受け取るよい機会となる。プロジェクトの中で起こる定期的な調整や技術レビューとは別に、実績に応じた報償のレビューといったメカニズムがこの種のフィードバックを提供する。また、これらの評価資料は、将来における外注先選定活動のためのインプットとしても役立つ。

計測と分析

計測1 計測を行い、その結果を使用して、ソフトウェア外注管理の活動状況を判断する。

(計測1)

計測値の例：

- 外注管理活動のためのコストで、計画と比較したもの
- 外注成果物の実際の納入日で、計画と比較したもの
- 発注元から外注先への実際の納入日で、計画と比較したもの

履行検証

検証1 ソフトウェア外注管理の活動は、上級管理層によって定期的にレビューされる。

3. 構成管理についての発注元と外注先の調整活動。
4. 外注先との計画されたレビューの開催。
5. 外注契約における主要プロジェクトマイルストーンや主要ステージの完了を確立するレビューの開催。
6. 外注先のソフトウェア成果物に対する検収プロセス。

ソフトウェア品質保証

レベル2のキープロセスエリア： 反復できる段階

『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』に必然的に含まれる活動には、適用される手順や標準に準拠しているかを検証するためにソフトウェア成果物や活動をレビューしかつ監査すること、およびこれらのレビューや監査結果をソフトウェアプロジェクトマネージャならびに他の適切なマネージャへ提出することが挙げられる。

ソフトウェア品質保証グループは、ソフトウェアプロジェクトの初期段階から参加することにより、ソフトウェアプロジェクトに付加価値をもたらす、プロジェクトや組織の方針面での制約条件を満足する計画、標準、ならびに手順を確立する。ソフトウェア品質保証グループは、計画、標準、および手順の確立に関与し、それらがプロジェクトニーズに適するものとなることを助け、またソフトウェアライフサイクル全般にわたって行われるレビューや監査に使用可能なことを確実なものにする。ソフトウェア品質保証グループは、ライフサイクル全般にわたるプロジェクト活動をレビューし、ソフトウェア成果物の監査を行う。そして、ソフトウェアプロジェクトが自身で確立した計画、標準、および手順を忠実に守っているかどうかについての可視性を管理層へ提供する。

遵守に関係する課題は、ソフトウェアプロジェクト内で最初に取り上げられ、可能であればその場で解決される。ソフトウェアプロジェクト内部で

解決できない課題については、ソフトウェア品質保証グループが解決に向け、適切な管理層レベルまで引き上げる。

このキープロセスエリアでは、ソフトウェア品質保証機能を実行するグループのプラクティスを扱う。ソフトウェア品質保証グループによりレビューや監査される特定の活動および作業成果物を明記するプラクティスは、通常、他のキープロセスエリアの『履行検証』コモンフィーチャの中にある。

ゴール

- | | |
|------|---|
| 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。 |

実施のコミットメント

- | | |
|----------|---|
| コミットメント1 | プロジェクトは、ソフトウェア品質保証（SQA）の実装に関して、明文化された組織方針に従う。 |
|----------|---|

この方針で明記される典型的な事項：

1. すべてのソフトウェアプロジェクトにおいてSQA機能が存在している。

(コミットメント1) 2. SQAグループは、下記報告先とは独立に上級管理層への報告経路を持つ：

- プロジェクトマネージャ
- プロジェクトのソフトウェアエンジニアリンググループ
- その他のソフトウェア関連グループ

その他のソフトウェア関連グループの例：

- ソフトウェア構成管理
- 文書化支援

戦略的事業目標や事業環境といった状況の中で、組織は、SQAなど独立性が必要な活動を支援する組織構造を決定する必要がある。

独立性とは以下の事項を満たすものをいう：

- SQAの役割を実施する担当者に、ソフトウェアプロジェクトに対する上級管理層の『目と耳』としての組織的な自由を与える。
- ソフトウェアプロジェクトの管理層が行う実績査定の対象から、そのプロジェクトに対してSQAの役割を実施する担当者を除外する。
- 上級管理層に、ソフトウェアプロジェクトのプロセスと成果物について客観的な報告が行われているという信頼感を与える。

3. 上級管理層は、定期的にSQA活動とその結果をレビューする。

実施能力

能力1

プロジェクトには、SQAの調整と履行に責任を持つグループ（SQAグループ）が存在する。

(能力1)

グループは、一連のタスクまたは活動に責任を持つ部門、マネージャ、および個人の集まりである。グループは、専任でない個人の場合、異なる部門から割り当てられた専任でない人達の場合、および専任の人達の場合など、さまざまな形態がある。グループ編成にあたっては、割り当てるタスクや活動、プロジェクト規模、組織構造、および組織の文化などを考慮する必要がある。品質保証グループのようにプロジェクトの活動に焦点を当てるグループもあれば、ソフトウェアエンジニアリングプロセスグループのように組織全体の活動に焦点を当てるグループもある。

能力2**SQA活動を実施するために、適切な資源と資金が提供される。**

1. プロジェクトのSQA活動に明確な責任を負うマネージャが割り当てられる。
2. SQAの役割に精通し、監督権限を持つ上級マネージャが、ソフトウェアの非遵守問題の受け取りと対処の担当として任命される。
 - 上級マネージャを頂点として、SQA報告系統に連なるマネージャは、全員がSQAの役割、責任、および権限に精通しているものとする。
3. SQA活動を支援するツールが利用可能になっている。

支援ツールの例：

- ワークステーション
- データベースプログラム
- 表計算プログラム
- 監査ツール

能力3**SQAグループのメンバは、SQA活動を実施するためのトレーニングを受ける。**

(能力3)

トレーニングの例：

- ソフトウェアエンジニアリングのスキルとプラクティス
- ソフトウェアエンジニアリンググループやその他のソフトウェア関連グループの役割と責任
- ソフトウェアプロジェクトのための標準、手順、および手法
- ソフトウェアプロジェクトのアプリケーションドメイン
- SQAの目的、手順、および手法
- ソフトウェア活動におけるSQAグループの関与
- SQA手法やツールの効果的な使い方
- コミュニケーション

能力4

ソフトウェアプロジェクトに携わるメンバは、SQAグループの役割、責任、権限、およびその意義について、オリエンテーションを受ける。

実施される活動

活動1

ソフトウェアプロジェクトのSQA計画は、文書化された手順に従って作成する。

この手順で明記される典型的な事項：

1. SQA計画は、プロジェクト全体計画の早期段階から、かつ並行して策定される。
2. SQA計画は、影響を受けるグループや個人によりレビューされる。

(活動1)

影響を受けるグループや個人の例：

- プロジェクトソフトウェアマネージャ
- その他のソフトウェアマネージャ
- プロジェクトマネージャ
- 顧客のSQA代表者
- SQAグループから、非遵守問題の報告を受ける上級マネージャ
- ソフトウェアエンジニアリンググループ（ソフトウェア設計などの全サブグループや、ソフトウェアタスクリダを含む）

3. SQA計画は、管理され制御されている。

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キープロセスエリアに記述されている。

活動2**SQAグループの活動は、SQA計画に従って実施する。**

この計画が扱う事項：

1. SQAグループの責任と権限
2. SQAグループの資源要件（要員、ツール、および設備を含む）
3. プロジェクトのSQAグループ活動スケジュールと資金

(活動2)

4. プロジェクトが策定するソフトウェア開発計画、標準、および手順へのSQAグループの関与
5. SQAグループによって実施されるべき評価項目

評価対象となるべき成果物や活動の例：

- 実稼働ソフトウェアおよび支援ソフトウェア
- 納入する成果物と納入しない成果物
- ソフトウェアと非ソフトウェア成果物（例えば、文書）
- 成果物の開発活動と成果物の検証活動（例えば、テストケースの実行）
- その他、成果物の生成過程で遂行される諸活動

6. SQAグループが開催すべき監査とレビュー
7. SQAグループのレビューと監査の基盤として用いるべきプロジェクト標準や手順
8. 非遵守問題の文書化と決着までの進捗の確認に関する手順

これらの手順は、本計画の中に策定されることもあるし、手順について定めた資料が参考資料として掲げられることもある。

9. SQAグループが作成を求められる文書
10. ソフトウェアエンジニアリンググループやその他のソフトウェア関連グループへの、SQA活動に関するフィードバックの提供方法と頻度

活動3 SQAグループは、プロジェクトのソフトウェア開発計画、標準、および手順の作成とレビューに参加する。

- (活動3)
1. SQAグループは、計画、標準、および手順について、次の観点からコンサルテーションを行いレビューする：
 - 組織方針の遵守
 - 外部から課せられる標準や要件（例えば、作業内容記述書で要求されている標準）の遵守
 - プロジェクトでの使用において、適切である標準
 - ソフトウェア開発計画で取り上げられるべき事項
 - プロジェクトにより指定された他の分野
 2. SQAグループは、プロジェクトのソフトウェア開発計画、標準、および手順が存在しており、ソフトウェアプロジェクトのレビューと監査に利用できることを検証する。

活動4 SQAグループは、ソフトウェアエンジニアリング活動をレビューし、その遵守状況を検証する。

1. ソフトウェア開発計画および指定されたソフトウェア標準や手順に照らして、活動の評価をおこなう。

SQAグループが実施する特定のレビューや監査活動で扱うプラクティスについては、他のキープロセスエリアにおける『履行検証』コモンフィチャーを参照。

2. 逸脱事項を特定し、文書化し、そして決着まで進捗を確認する。
3. 修正内容を検証する。

活動5 SQAグループは、指定されたソフトウェア作業成果物を監査し、その遵守状況を検証する。

1. 納入するソフトウェア成果物については、顧客への納入前に評価する。

2. 指定されたソフトウェア標準、手順、および契約要件に照して、ソフトウェア作業成果物を評価する。

(活動5)

3. 逸脱事項を特定し、文書化し、そして決着まで進捗を確認する。
4. 修正内容を検証する。

活動6

SQAグループは、ソフトウェアエンジニアリンググループに対し、定期的にその活動結果を報告する。

活動7

ソフトウェア活動やソフトウェア作業成果物について特定された逸脱事項は、文書化された手順に従って文書化し取り扱う。

この手順で明記される典型的な事項：

1. ソフトウェア開発計画および指定されたプロジェクト標準や手順からの逸脱事項を文書化し、可能な範囲で、適切なソフトウェアタスクリーダー、ソフトウェアマネージャ、またはプロジェクトマネージャと協力して解決する。
2. ソフトウェア開発計画および指定されたプロジェクト標準や手順からの逸脱事項のうち、ソフトウェアタスクリーダー、ソフトウェアマネージャ、またはプロジェクトマネージャと協力して解決することができない事項について文書化し、非遵守問題担当の上級マネージャに報告する。
3. 上級マネージャに提出された非遵守問題については、解決されるまで定期的にレビューする。
4. 非遵守問題を記録した文書は、管理され制御されている。

活動8

SQAグループは、顧客のSQA担当者などとともに、適宜、SQAグループの活動や所見の定期的レビューを開催する。

計測と分析

計測1

計測を行い、その結果を使用して、SQA活動のコストやスケジュール状況を判断する。

計測値の例：

- SQA活動に関するマイルストーン完了状況で、計画と比較したもの
- SQA活動における完了した作業、費やした工数、および費やしたコストで、計画と比較したもの
- 成果物監査や活動レビューの数で、計画と比較したもの

履行検証

検証1

SQA活動は、上級管理層によって定期的にレビューされる。

上級管理層による定期的なレビューの主目的は、ソフトウェアプロセス活動に対する認識と見通しを上級管理層に提供することである。活動のレビューは、適切な抽象レベルでタイムリーに行う。レビューの間隔は、組織のニーズに合わせる。例外報告への体制が十分整っていれば、上級管理層のレビュー間隔は長くても構わない。

上級管理層の監督レビューに関して、その典型的な内容を扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証1を参照。

検証2 **SQA活動は、プロジェクトマネージャによって定期的に、かつイベント発生を契機としてレビューされる。**

プロジェクトマネージャの監督レビューに関して、その典型的な内容を扱うプラクティスについては『ソフトウェアプロジェクト進捗管理』キーププロセスエリアの検証2を参照。

検証3 **プロジェクトにおけるSQAグループの活動ならびにSQAグループのソフトウェア作業成果物は、SQAグループとは独立した専門家が定期的にレビューする。**

ソフトウェア構成管理

レベル2のキープロセスエリア： 反復できる段階

『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』に必然的に含まれる活動には、所定の時点におけるソフトウェア構成（選択されたソフトウェア作業成果物とそれらの記述）を特定すること、その構成に関する変更を系統的に制御すること、およびソフトウェアのライフサイクル全般にわたって構成の一貫性と追跡可能性を維持することが挙げられる。ソフトウェア構成管理下におく作業成果物としては、顧客に納入されるソフトウェア成果物（例えば、ソフトウェア要件に関する文書やコード）、およびこれらのソフトウェア成果物と一緒に特定されるアイテムまたはこれらのソフトウェア成果物の作成に必要なアイテム（例えば、コンパイラ）がある。

ソフトウェアベースラインライブラリは、ソフトウェアベースラインの開発に伴いそれを含む形で確立される。ベースラインの変更やソフトウェアベースラインライブラリから作られるソフトウェア成果物のリリースについては、ソフトウェア構成管理の機能である変更制御や構成監査をとおして系統的な制御を行う。

このキープロセスエリアでは、ソフトウェア構成管理機能を実施するためのプラクティスを扱う。特定の構成アイテム/ユニットを明らかにするプラクティスは、それぞれの構成アイテム/ユニットの開発と保守について述べているキープロセスエリアの中に含まれる。

ゴール

- 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。

実施のコミットメント

- コミットメント1 プロジェクトは、ソフトウェア構成管理（SCM）の実装に関して、明文化された組織方針に従う。

この方針で明記される典型的な事項：

1. 各プロジェクトにSCMの責任を明確に割り当てる。
2. プロジェクトのライフサイクル全般にわたってSCMを実装する。
3. 外部に納入されるソフトウェア成果物、指定された内部ソフトウェア作業成果物、およびプロジェクト内で使用する指定されたサポートツール（例えば、コンパイラ）について、SCMを実装する。

4. プロジェクトは、構成アイテム/ユニットおよび関連SCM記録を保管するためのリポジトリを確立するか、または利用できるようにする。

(コミットメント1)

これらのプラクティスにおいて、このリポジトリの内容は『ソフトウェアベースラインライブラリ』として参照される。

これらのプラクティスでは、このリポジトリへアクセスするためのツールと手順は『構成管理ライブラリシステム』として参照される。

構成管理下に置かれ、ひとつのまとまりとして扱われる作業成果物は、構成アイテムとして参照される。

典型的には、構成アイテムは構成コンポーネントに分割され、構成コンポーネントはユニットに分割される。ハードウェア/ソフトウェアシステムでは、すべてのソフトウェアをまとめてひとつの構成アイテムと見なす場合もあるし、あるいはソフトウェアをいくつかの構成アイテムに分割して捉える場合もある。これらのキープラクティスでは、『構成アイテム/ユニット』という用語は構成管理の対象となる要素を指して用いられる。

5. ソフトウェアベースラインとSCM活動は、定期的に監査される。

実施能力

能力1

プロジェクトのソフトウェアベースラインの管理権限を持つ委員会（ソフトウェア構成制御委員会：Software Configuration Control Board=SCCB）が存在するか、あるいは確立される。

SCCBは：

1. ソフトウェアベースラインの確立と構成アイテム/ユニットの特定を認可する。

2. ソフトウェアベースラインの変更で影響を受ける可能性のあるプロジェクトマネージャおよびすべてのグループの利害を代表する。

(能力1)

影響を受けるグループの例：

- ハードウェア品質保証
- ハードウェア構成管理
- ハードウェアエンジニアリング
- 製造技術
- ソフトウェアエンジニアリング (ソフトウェア設計などのすべてのサブグループを含む)
- システムエンジニアリング
- システムテスト
- ソフトウェア品質保証
- ソフトウェア構成管理
- 契約管理
- 文書化支援

3. ソフトウェアベースラインの変更をレビューし認可する。
4. ソフトウェアベースラインライブラリからの成果物の生成を認可する。

能力2

プロジェクトには、SCMの調整と実装に責任を持つグループ (SCMグループ) が存在する。

グループは、一連のタスクまたは活動に責任を持つ部門、マネージャ、および個人の集まりである。グループは、専任でない個人の場合、異なる部門から割り当てられた専任でない人達の場合、および専任の人達の場合など、さまざまな形態がある。グループ編成にあたっては、割り当てるタスクや活動、プロジェクト規模、組織構造、および組織の文化などを考慮する必要がある。品質保証グループのようにプロジェクトの活動に焦点を当てるグループもあれば、ソフトウェアエンジニアリングプロセスグループのように組織全体の活動に焦点を当てるグループもある。

(能力2)

SCMグループは、次の項目を調整し実装する：

1. プロジェクトのソフトウェアベースラインライブラリの生成と管理
2. SCMの計画、標準、および手順、に関する開発、保守、および配布
3. SCMの下におかれる作業成果物セットの特定

作業成果物とは、ソフトウェアプロセスを定義、維持、または使用した結果として生ずる任意の成果物をいう。

4. ソフトウェアベースラインライブラリへのアクセスの管理
5. ソフトウェアベースラインの更新
6. ソフトウェアベースラインライブラリからの成果物生成
7. SCM活動の記録
8. SCM報告の作成と配布

能力3 **SCM活動を実施するために、適切な資源と資金が提供される。**

1. SCMに明確な責任を負うマネージャが割り当てられる。
2. SCM活動を支援するツールが利用可能である。

支援ツールの例：

- ワークステーション
- データベースプログラム
- 構成管理ツール

能力4 **SCMグループのメンバは、SCM活動を実施するためのねらい、手順、および手法についてトレーニングを受ける。**

トレーニングの例：

- SCM標準、手順、および手法
- SCMツール

能力5 **ソフトウェアエンジニアリンググループおよびその他のソフトウェアに関連するグループのメンバは、SCM活動を実施するためのトレーニングを受ける。**

その他のソフトウェア関連グループの例：

- ソフトウェア品質保証
- 文書化支援

トレーニングの例：

- ソフトウェアエンジニアリンググループおよびその他のソフトウェア関連グループの内部で実施するSCM活動に関する標準、手順、および手法
- SCMグループの役割、責任、および権限

実施される活動

活動1

SCM計画は、文書化された手順に従ってソフトウェアプロジェクト個別に作成する。

この手順で明記される典型的な事項：

1. SCM計画は、プロジェクトの早期段階でかつプロジェクトの全体計画と並行して策定する。
2. SCM計画は、影響を受けるグループがレビューする。
3. SCM計画は、管理され制御されている。

(活動1)

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キープロセスエリアに記述されている。

活動2

文書化され承認されたSCM計画を、SCM活動実施の基盤として使用する。

この計画が扱う事項：

1. SCMで実施する活動、活動のスケジュール、割り当てられる責任、および必要な資源（要員、ツール、およびコンピュータ設備を含む）
2. SCM要件およびソフトウェアエンジニアリンググループと他のソフトウェア関連グループが実施する活動

活動3

ソフトウェアベースラインのリポジトリとして、構成管理ライブラリシステムを構築する。

このライブラリシステムは：

1. SCMの複数の制御レベルを支援する。

(活動3)

複数の制御レベルが必要となる状況の例：

- ライフサイクルの時間経過に伴い、必要な制御レベルが異なる場合（成果物が完成に近づくほど厳格な制御が必要となる場合）
- ソフトウェア単独システムと、ハードウェアとソフトウェアの双方を含むシステムの場合で、必要な制御レベルが異なる場合

2. 構成アイテム/ユニットの格納と取り出しの機能を提供する。
3. 影響を受けるグループ間やライブラリ内の異なる制御レベル間で、構成アイテム/ユニットの共有と転送機能を提供する。
4. 構成アイテム/ユニットに対し、成果物標準の使用を助ける。
5. 構成アイテム/ユニットの保管されたバージョンに対し、格納と復旧機能を提供する。

6. ソフトウェアベースラインライブラリから、正しい成果物を確実に生成することを助ける。
7. SCM記録の格納、更新、および取り出しの機能を提供する。
8. SCMレポートの作成を支援する。
9. ライブラリの構造と内容の保守機能を提供する。

ライブラリの保守機能の例：

- ライブラリファイルのバックアップ/復元
- ライブラリエラーからの復旧

活動4

構成管理下におくべきソフトウェア作業成果物を特定する。

(活動4)

1. 文書化された基準に基づいて、構成アイテム/ユニットを選択する。

構成アイテム/ユニットとして特定されるソフトウェア作業成果物の例：

- プロセス関連文書（例えば、計画、標準、または手順）
- ソフトウェア要件
- ソフトウェア設計
- ソフトウェアコードユニット
- ソフトウェアテスト手順
- ソフトウェアテスト用に作られるソフトウェアシステムのビルド
- 顧客やエンドユーザに納入されるソフトウェアシステムのビルド
- コンパイラ
- その他の支援ツール

2. 構成アイテム/ユニットに一意的識別子を割り当てる。
3. 各構成アイテム/ユニットの特徴が明記される。
4. 各構成アイテム/ユニットを含むソフトウェアベースラインが明記される。
5. 各構成アイテム/ユニットが構成管理下におかれる時期が明記される。
6. 各構成アイテム/ユニットの責任者（構成管理の観点からの所有者）が明記される。

活動5

すべての構成アイテム/ユニットの変更要求や問題報告は、文書化された手順に従って発行し、記録し、レビューし、承認し、そして進捗を確認する。

活動6

ベースラインの変更は、文書化された手順に従って制御する。

この手順で明記される典型的な事項：

1. レビューや回帰テストを実施し、変更による意図しない影響がベースラインに起こっていないことを確実なものにする。
2. SCCBで承認された構成アイテム/ユニットだけをソフトウェアベースラインライブラリに入れる。
3. ソフトウェアベースラインライブラリの正しさと一貫性を維持する方法で、構成アイテム/ユニットのチェックイン/チェックアウトを行う。

報告書の例：

- SCCB議事録
- 変更要求の概要と状況
- 障害報告の概要と状況（修正を含む）
- ソフトウェアベースラインへの変更の概要
- 構成アイテム/ユニット改訂履歴
- ソフトウェアベースライン状況
- ソフトウェアベースライン監査結果

活動10 **ソフトウェアベースライン監査は、文書化された手順に従って実施する。**

この手順で明記される典型的な事項：

1. 監査のため、十分な準備をする。
2. ソフトウェアベースラインの一貫性について評価する。
3. 構成管理ライブラリシステムの構造と設備をレビューする。
4. ソフトウェアベースラインライブラリの内容の完全性と正しさを検証する。

- (活動10)**
5. 適用されるSCM標準や手順が遵守されていることを検証する。
 6. 監査結果を、プロジェクトソフトウェアマネージャに報告する。
 7. 監査の結果生じた処置項目について、決着するまで進捗を確認する。

計測と分析

計測1

計測を行い、その結果を使用して、SCMの活動状況を判断する。

計測値の例：

- 単位時間に処理された変更要求の数
- SCM活動に関するマイルストーン完了状況で、計画と比較したもの
- SCM活動で、完了した作業、費やした工数、およびコスト

履行検証

検証1

SCMの活動は、上級管理層によって定期的にレビューされる。

上級管理層による定期的なレビューの主目的は、ソフトウェアプロセス活動に対する認識と見通しを上級管理層に提供することである。活動のレビューは、適切な抽象レベルでタイムリーに行う。レビューの間隔は、組織のニーズに合わせる。例外報告への体制が十分整っていれば、上級管理層のレビュー間隔は長くても構わない。

(検証1)

上級管理層の監督レビューに関して、その典型的な内容を扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キーププロセスエリアの検証1を参照。

検証2 **SCMの活動は、プロジェクトマネージャによって定期的に、かつイベント発生を契機としてレビューされる。**

プロジェクトマネージャの監督レビューに関して、その典型的な内容を扱うプラクティスについては『ソフトウェアプロジェクト進捗管理』キーププロセスエリアの検証2を参照。

検証3 **SCMグループは、定期的にソフトウェアベースラインを監査し、ソフトウェアベースラインを定義した文書との準拠性を検証する。**

検証4 **ソフトウェア品質保証グループは、SCMの活動と作業成果物をレビューかつ/または監査し、その結果を報告する。**

『ソフトウェア品質保証』キーププロセスエリアを参照。

レビューや監査では、最低限、以下のことを検証する：

1. SCM標準や手順が次の関係者によって遵守されている：
 - SCMグループ
 - SCCB
 - ソフトウェアエンジニアリンググループ
 - その他のソフトウェア関連グループ
2. 定期的なソフトウェアベースライン監査が行われている

レベル2 索引

(英和索引と日本語索引は巻末にある。)

acceptance testing
 subcontractor's products 52
allocated requirements
 definition of 2
audit
 software baseline 76
 software work products 61
baseline configuration management
 definition of 66
commitment review procedure 7
commitments
 changes to 6
 review by senior management 29, 32
 software project 10
configuration items
 audit of 77
 identification of 73
 procedure for changing 74
 procedure for controlling changes 74
 recording status of 75
configuration management library
system
 definition of 67
 establishment of 71
contract
 software subcontract management 47
Corrective actions
 deviation from software development
 plan 28
critical computer resources
 estimates for 21
 tracking of 34

Customer
 examples of 1
data
 recording of 36
developmental configuration management
 definition of 65
documented procedure
 acceptance testing of
 subcontractor's products 52
 changes to software subcontractor's
 scope of work 49
 changing configuration items 74
 controlling changes to
 configuration items 74
 deriving software schedule 22
 development of software development
 plan 16
 deviations in software engineering
 activities 61
 estimating critical computer
 resources 21
 estimating resources and costs 20
 estimating size of software work
 products 18
 milestone reviews 37
 monitoring subcontractor's software
 configuration management 52
 monitoring subcontractor's software
 quality assurance 51
 preparation of software quality
 assurance plan 58
 products from software baseline
 library 75

- recording status of configuration items 75
- review of commitments 32
- reviews of software subcontractor 51
- revising software development plan 31
- selection of software subcontractor 46
- software baseline audits 76
- software configuration management plan 71
- subcontracted work 45
- Estimates
 - critical computer resources 21
 - resources and costs 20
 - size of software work products 18
- manage and control
 - definition of 73
 - noncompliance items 62
 - software configuration management plan 71
 - software development plan 11, 17, 32
 - software planning data 23
 - software quality assurance plan 59
 - software replanning data 36
 - statement of work for software project 13
 - subcontract statement of work 46
 - system requirements allocated to software 6
- Measurements
 - requirements management 7
 - software configuration management 77
- software project planning 24
- software project tracking and oversight 37
- software quality assurance 62
- software subcontract management 53
- noncompliance items
 - management and control of 62
- orientation
 - software project tracking and oversight 30
 - software quality assurance 58
 - software subcontract management 44
- Plan 46
 - software engineering 23
 - software quality assurance 58
- plan for software project
 - items in 17
- policy
 - management of system requirements allocated to software 2
 - managing the software project 28
 - software configuration management 67
 - software project planning 10
 - software quality assurance 56
 - software subcontract management 43
- project manager
 - review
 - requirements management 8
 - software configuration management 77
 - software project planning 24
 - software project tracking and oversight 38

- software quality assurance 63
- software subcontract management 53
- project software manager
 - assignment of work tasks 29
 - coordination of software project planning 13
 - responsibilities of 28
 - software project planning responsibility 10
- reports
 - software configuration management activities 75
- Requirements Management 1-8
 - measurements for 7
 - resources and funding for 4
 - review
 - project manager 8
 - senior management 8
 - software quality assurance 8
 - tools for 4
- resources and costs
 - estimates for 20
- resources and funding
 - requirements management 4
 - software configuration management 69
 - software project planning 14
 - software project tracking and oversight 30
 - software quality assurance 57
 - software subcontract management 43
- review
 - at milestones 37
- commitments 29
- requirements management
 - project manager 8
 - senior management 8
 - software quality assurance 8
- software configuration management
 - project manager 77
 - senior management 77
 - software quality assurance 78
- software project commitments 11, 15
- software project planning
 - project manager 24
 - senior management 24
 - software quality assurance 25
- software project tracking and oversight
 - project manager 38
 - senior management 38
 - software quality assurance 39
- software quality assurance
 - project manager 63
 - senior management 63
- software subcontract management
 - project manager 53
 - senior management 53
 - software quality assurance 54
- technical aspects of software subcontract 50
- tracking progress 36
- risks
 - identification of 22
 - tracking of 35
- senior management
 - review

- commitment changes 6
- commitments 29, 32
- requirements management 8
- software configuration management 77
- software project commitments 11
- software project planning 24
- software project tracking and oversight 38
- software quality assurance 57, 63
- software subcontract management 53
- software activities and software work products
 - deviations in 61
- Software baseline
 - audit of 76
- software baseline library
 - check-in/out procedure 74
 - creating products from 75
 - creation of 75
 - definition of 67
- software baselines
 - repository for 71
- Software Capability Evaluation 47
- software commitments
 - changes to 28, 33
- software configuration control board (SCCB)
 - activities of 68
- Software Configuration Management 65, 70-78
 - audit of 68
 - measurements for 77
 - resources and funding for 69
 - reports of activities 75
 - review
 - project manager 77
 - senior management 77
 - software quality assurance 78
 - tools for 69
- software configuration management group
 - activities of 69
 - audit of configuration items 77
 - monitor of subcontractor's software configuration management 52
- software configuration management plan
 - management and control of 71
 - procedure for 71
 - use of 71
- software costs
 - tracking of 34
- software development 13
- software development plan
 - basis for tracking project 28
 - definition of 12
 - description of 9
 - development of 16
 - deviations from 28
 - documentation of 29
 - for software project tracking and oversight 31
 - for subcontractor 48
 - management and control of 11, 17, 32
 - refinement of 37
 - revision of 31
 - software quality assurance group and

- 60
- software engineering group
 - participation in software project
 - planning 15
 - participation on project proposal
 - team 14
 - review of project tracking 36
 - review of system requirements allocated to software 5
- software engineering technical activities
 - tracking of 35
- software life cycle
 - identification of 15
- software manager
 - responsibility for software work products 13
- Software planning data 23
 - management and control of 23
- software project 12
 - commitment for 10
 - recording of data for 36
- software project commitments
 - review of 11, 15
- Software Project Planning 9-25
 - coordination of 13
 - measurements for 24
 - overall project planing and 15
 - policy for 10
 - resources and funding for 14
 - responsibility for 10
 - review
 - project manager 24
 - senior management 24
 - software quality assurance 25
 - tools for 14
 - training for 14
- Software Project Tracking and Oversight 27-39
 - assignment of work tasks 29
 - measurements for 37
 - policy for 28
 - resources and funding for 30
 - review
 - project manager 38
 - senior management 38
 - software quality assurance 39
 - software development plan for 31
 - tools for 30
 - training for 30
- Software Quality Assurance 55-63
 - measurements for 62
 - orientation in 58
 - policy for 56
 - resources and funding for 57
 - review
 - independent experts 63
 - project manager 63
 - senior management 63
 - tools for 57
 - training for 58
- software quality assurance group
 - review
 - software configuration management 78
- software quality assurance group
 - activities of 60
 - audit

- software products 61
- coordination with customer 62
- monitor of subcontractor's quality assurance 51
- reports of activities 61
- review
 - requirements management 8
 - software engineering activities 60
 - software project planning 25
 - software project tracking and oversight 39
 - software subcontract management 54
- software development plan and 60
- software quality assurance plan 58
 - management and control of 59
- software replanning data
 - management and control of 36
- software schedule
 - procedure for 22
 - tracking of 35
- software subcontract
 - policy for managing 43
 - responsibility for managing 42
- Software Subcontract Management 41-54
 - basis for 47
 - contract for 47
 - measurements for 53
 - orientation in 44
 - review
 - project manager 53
 - senior management 53
 - software quality assurance 54
 - training for 44
- software subcontractor
 - acceptance testing of products 52
 - performance evaluation 53
 - procedure for selection of 46
 - status reviews with 49
 - technical reviews with 50
- software subcontractor
 - procedure for reviews with 51
- software subcontractor's scope of work
 - changes to 49
- software work products
 - audit of 61
 - estimating size of 18
 - responsibility for 13, 29
 - tracking size of 33
- statement of work
 - subcontract 45
- statement of work for software project
 - management and control of 13
 - review of 13
- subcontract manager
 - responsibilities of 42
- subcontract statement of work 45
 - management and control of 46
- subcontracted work
 - procedure for 45
 - selection of 45
- system requirements
 - responsibility for 3
- system requirements allocated to software
 - changes to 6
 - documentation of 3

examples of 4
policy for managing 2
review of 5
system requirements allocated to the software
definition of 1
Tools
requirements management 4
software configuration management 69
software project planning 14
software project tracking and oversight 30
software quality assurance 57
software subcontract management 44
tracking
critical computer resources 34
risks 35
size of software work products 33
software costs 34
Software engineering technical activities 35
training
software configuration management 70
software project planning 14
software project tracking and oversight 30
software quality assurance 58
software subcontract management 44

組織プロセス重視

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』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』に必然的に含まれる活動には、組織のソフトウェアプロセスとプロジェクトのソフトウェアプロセスに関する理解を深めこれを維持すること、ソフトウェアプロセスのアセスメント、開発、維持、および改善に関しての調整を行うことが挙げられる。

組織は、長期的なコミットメントと資源を提供して、現在と未来のソフトウェアプロジェクトにわたり、ソフトウェアエンジニアリングプロセスグループのようなグループによるソフトウェアプロセスの開発と維持を調整する。このグループは、組織のソフトウェアプロセス活動に責任をもつ。このグループは、特に「組織の標準ソフトウェアプロセス」および（『組織プロセス定義』キープロセスエリアで記述されているような）関連のプロセス資産の開発と維持に責任を負い、そしてソフトウェアプロジェクトとともにプロセス活動を調整する。

ゴール

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。

実施のコミットメント

- コミットメント1 組織は、組織横断的なソフトウェアプロセスの開発と改善活動の調整に関して明文化された組織方針に従う。

この組織方針で明記される典型的な事項：

1. 組織レベルのソフトウェアプロセス活動と、それらの活動をプロジェクトと調整することに責任を持つグループが確立される。
2. プロジェクトで使用されるソフトウェアプロセスは、プロセスの強みと弱みを判断するために定期的にアセスメントされる。
3. プロジェクトで使用される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」から適切にテラリングされる。

「組織の標準ソフトウェアプロセス」のテラリングを扱うプラクティスについては、『ソフトウェア統合管理』キープロセスエリアの活動1を参照。

4. 各プロジェクトのソフトウェアプロセス、ツール、および手法について、それらの改善などの有用な情報は他のプロジェクトでも利用可能である。

- コミットメント2 上級管理層は、組織のソフトウェアプロセスの開発と改善の活動を主催する。

(コミットメント2) 上級管理層は：

1. これらのソフトウェアプロセス活動へのコミットメントを、組織の要員とマネージャに対して実際に行動で示す。
2. 資金、配員、およびその他の資源について、長期的な計画とコミットメントを確立する。
3. プロセスの開発および改善活動の管理と実装について、戦略を確立する。

コミットメント3 上級管理層は、ソフトウェアプロセスの開発と改善に対し、組織の活動を監督する。

上級管理層は：

1. 「組織の標準ソフトウェアプロセス」が事業目標や戦略を支援することを確実なものにする。
2. ソフトウェアプロセスの開発と改善活動の優先順位設定について、助言する。
3. ソフトウェアプロセスの開発と改善に対する計画の確立作業に参加する。
 - 上級管理層は、ソフトウェアプロセス要件と課題について、高いレベルの要員やマネージャと共に調整する。
 - 上級管理層は、マネージャと要員による支援と参加を確保するため、組織のマネージャと共に調整する。

実施能力

能力1 組織のソフトウェアプロセス活動に責任を持つグループが存在する。

(能力1)

グループは、一連のタスクまたは活動に責任を持つ部門、マネージャ、および個人の集まりである。グループは、専任でない個人の場合、異なる部門から割り当てられた専任でない人達の場合、および専任の人達の場合など、さまざまな形態がある。グループ編成にあたっては、割り当てるタスクや活動、プロジェクト規模、組織構造、および組織の文化などを考慮する必要がある。品質保証グループのようにプロジェクトの活動に焦点を当てるグループもあれば、ソフトウェアエンジニアリングプロセスグループのように組織全体の活動に焦点を当てるグループもある。

1. 可能であれば、このグループの中核にはソフトウェアの技術専門家を専任で配員し、できれば専任でない他のスタッフも補佐につける。

このグループのもっとも良くある例は、ソフトウェアエンジニアリングプロセスグループ(SEPG)である。

2. このグループは、ソフトウェアエンジニアリングの機能とソフトウェアに関連する機能を代表するように配員される。

ソフトウェアエンジニアリング機能やソフトウェア関連機能の例：

- ソフトウェア要件分析
- ソフトウェア設計
- コーディング
- ソフトウェアテスト
- ソフトウェア構成管理
- ソフトウェア品質保証

能力2

組織のソフトウェアプロセス活動のために、適切な資源と資金が提供される。

(能力2)

1. 専門分野の経験と知識を持った個人がこのグループの支援のためコミットしている。

専門分野の例：

- ソフトウェア再利用
- コンピュータ支援ソフトウェアエンジニアリング(CASE)技術
- 計測
- トレーニングコースの開発

2. 組織のソフトウェアプロセス活動を支援するツールが利用可能になっている。

支援ツールの例：

- 統計分析ツール
- DTPツール
- データベース管理システム
- プロセスモデリングツール

能力3

組織のソフトウェアプロセス活動に責任を持つグループのメンバーは、活動実施のための必修トレーニングを受ける。

トレーニングの例：

- ソフトウェアエンジニアリングのプラクティス
- プロセス制御技法
- 組織の変更管理
- ソフトウェアプロセスの計画、管理、およびモニター
- 技術の変化

(能力3)

『トレーニングプログラム』キープロセスエリアを参照。

能力4

ソフトウェアエンジニアリンググループや他のソフトウェアに関連するグループのメンバは、組織のソフトウェアプロセス活動とこれらの活動における自分たちの役割についてオリエンテーションを受ける。

『トレーニングプログラム』キープロセスエリアを参照。

実施される活動

活動1

ソフトウェアプロセスを定期的にアセスメントし、アセスメントの所見を取り上げるために処置計画を策定する。

典型的には、アセスメントは1.5～3年おきに開催される。

アセスメントでは、組織で使用されているすべてのソフトウェアプロセスを調べるが、プロセスエリアとプロジェクトをサンプリングしてアセスメントしても良い。

組織のソフトウェアプロセス能力アセスメント手法のひとつの例が、「SEIソフトウェアプロセスアセスメント」手法である。

処置計画では、以下の内容を明らかにする：

- アセスメントのどの所見を取り上げるか
- 所見を取り上げて変更を実装するためのガイドライン
- 変更を実施する責任グループまたは責任者

活動2 **組織は、ソフトウェアプロセスの開発と改善活動に関する計画を策定し保守する。**

この計画は：

1. 主要なインプットとして、ソフトウェアプロセスアセスメントや他の組織改善構想からの処置計画を使用する。
2. 実施する活動を定義し、活動スケジュールを定める。
3. 活動に責任を持つグループと個人を明記する。
4. 要員とツールを含めて必要な資源を明らかにする。
5. 初期リリース時と大規模改訂のたびに、ピアレビューを受ける。

『ピアレビュー』キープロセスエリアを参照。

6. 組織のソフトウェアマネージャと上級マネージャによって、レビューされ合意される。

活動3 **ソフトウェアプロセスの開発と改善に関して、組織やプロジェクトの活動を組織レベルで調整する。**

この調整では、以下の項目の開発や改善を扱う：

1. 「組織の標準ソフトウェアプロセス」

「組織の標準ソフトウェアプロセス」を扱うプラクティスについては、『組織プロセス定義』キープロセスエリアの活動1と2を参照。

(活動3)**2. 「プロジェクトの定義されたソフトウェアプロセス」**

「プロジェクトの定義されたソフトウェアプロセス」を扱うプラクティスについては、『ソフトウェア統合管理』キープロセスエリアの活動1と2を参照。

活動4**「組織のソフトウェアプロセスデータベース」の使用について、組織レベルで調整する。**

「組織のソフトウェアプロセスデータベース」は、組織やプロジェクトにおけるソフトウェアプロセスと、結果としてのソフトウェア成果物に関する情報を収集するために使用される。

組織のソフトウェアプロセスデータベースを扱うプラクティスは、『組織プロセス定義』キープロセスエリアの活動5を参照。

活動5**組織で限定的に使用される新しいプロセス、手法、およびツールを、モニターし、評価し、そして適切であれば組織の他の部分に移転させる。****活動6****組織やプロジェクトのソフトウェアプロセスに関するトレーニングを、組織横断的に調整する。**

1. 組織やプロジェクトのソフトウェアプロセスに関係する主題毎のトレーニング計画を作成する。
2. 適切であれば、組織のソフトウェアプロセス活動の責任グループ(例えば、SEPG)またはトレーニンググループがトレーニングを準備し開催する。

(活動6)

『トレーニングプログラム』キープロセスエリアを参照。

活動7

組織やプロジェクトのソフトウェアプロセスの開発と改善に関する活動について、ソフトウェアプロセスの実装に携わるグループに情報を伝える。

これらの人々に情報を提供し彼らを巻き込む手段の例：

- プロセスに関する電子掲示板
- プロセス諮問委員会
- 作業グループ
- 情報交換会
- 調査
- プロセス改善チーム
- 非公式な討議

計測と分析

計測1

計測を行い、その結果を使用して、組織プロセスの開発と改善の活動状況を判断する。

計測値の例：

- プロセスのアセスメント、開発、および改善といった組織活動のうち、完了した作業、費やした工数、および費やした資金で、これらの活動に対する計画と比較したもの
- 各ソフトウェアプロセスアセスメントの結果で、以前のアセスメントの結果や勧告と比較したもの

履行検証

検証1

ソフトウェアプロセスの開発と改善の活動は、上級管理層によって定期的にレビューされる。

上級管理層による定期的なレビューの主目的は、ソフトウェアプロセス活動に対する認識と見通しを上級管理層に提供することである。活動のレビューは、適切な抽象レベルでタイムリーに行う。レビューの間隔は、組織のニーズに合わせる。例外報告への体制が十分整っていれば、上級管理層のレビュー間隔は長くても構わない。

1. ソフトウェアプロセスを開発し改善する活動の進捗状況を、計画に照らしてレビューする。
2. 低いレベルでは解決できない対立や課題が取り上げられる。
3. 処置項目が割り当てられ、レビューされ、そして決着まで進捗が確認される。
4. 各レビューの要約報告書が作成され、影響を受けるグループと個人に配布される。

組織プロセス定義

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。

『組織プロセス定義』に必然的に含まれる活動には、「組織の標準ソフトウェアプロセス」およびこれに関連するソフトウェア資産を開発すること、そしてそれらを維持することが挙げられる。プロセス資産とは、ソフトウェアライフサイクルの記述、プロセステーラリングのガイドラインと基準、「組織のソフトウェアプロセスデータベース」、および「ソフトウェアプロセス関連文書のライブラリ」のことをいう。

これらの資産の収集には、その組織における『組織プロセス定義』の実装の仕方によりさまざまな方法がある。例えば、ソフトウェアライフサイクルの記述は「組織の標準ソフトウェアプロセス」を構成するために必要な部分であることがあり、また「ソフトウェアプロセス関連文書のライブラリ」の一部は「組織のソフトウェアプロセスデータベース」に蓄積されることがある。

「組織のソフトウェアプロセス資産」は、「プロジェクトの定義されたソフトウェアプロセス」を開発し、履行し、そして維持するのに用いることができる。（「プロジェクトの定義されたソフトウェアプロセス」の開発および維持に関連するプラクティスは、『ソフトウェア統合管理』キープロセスエリアに記述されている。）

ゴール

- 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。

実施のコミットメント

- コミットメント1 組織は、標準ソフトウェアプロセスおよび関連するプロセス資産の開発と維持に関して、明文化された方針に従う。

「組織のソフトウェアプロセス資産」は、以下のものを含む。

- 「組織の標準ソフトウェアプロセス」
- 「組織の標準ソフトウェアプロセス」をプロジェクトがテラリングするためのガイドラインと基準
- 使用が承認されたソフトウェアライフサイクルの記述
- 「組織のソフトウェアプロセスデータベース」
- あらかじめ開発され、再利用可能になっているソフトウェアプロセス関連文書のライブラリ

この方針で明記される典型的な事項：

1. 「組織の標準ソフトウェアプロセス」が定義されている。

(コミットメント1)

標準ソフトウェアプロセスの第一の目的は、プロジェクト横断的にプロセス資産と経験の共有を最大化し、組織レベルでプロジェクトから収集するプロセス計測値の標準的な集合を定義し収集する能力を提供することにある。

「組織の標準ソフトウェアプロセス」は、複数のソフトウェアプロセスからなることがある。ソフトウェアプロジェクトは、異なったアプリケーション、ライフサイクル、方法論、およびツールを多様な形で組み合わせることがあり、これらのニーズに対処するために、複数のソフトウェアプロセスが必要なことがある。

2. 「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」をテーラリングしたバージョンである。

「組織の標準ソフトウェアプロセス」のテーラリングを扱うプラクティスについては、『ソフトウェア統合管理』キープロセスエリアの活動1を参照。

3. 組織のソフトウェアプロセス資産が維持される。
4. プロジェクトから収集された情報は、「組織の標準ソフトウェアプロセス」を改善するために整理され利用される。

収集される情報の例：

- プロセスと成果物の計測値
- 教訓
- その他のプロセス関連文書

実施能力

能力1

「組織の標準ソフトウェアプロセス」ならびに関連するプロセス資産を開発し維持するために、適切な資源と資金が提供される。

1. 「組織の標準ソフトウェアプロセス」ならびに関連するプロセス資産の開発と維持は、組織のソフトウェアプロセス活動に責任のあるグループ（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）によって実施され調整される。

組織のソフトウェアプロセス活動に責任のあるグループを扱うプラクティスについては、『組織プロセス重視』キープロセスエリアを参照。

2. プロセスの開発と維持を支援するためのツールが、利用可能になっている。

支援ツールの例：

- DTPツール
- データベース管理システム
- プロセスモデリングツール

能力2

「組織の標準ソフトウェアプロセス」ならびに関連するプロセス資産を開発し維持する担当者は、これらの活動を実施するための必修トレーニングを受ける。

トレーニングの例：

- ソフトウェアエンジニアリングのプラクティスと手法
- プロセス分析と文書化手法
- プロセスモデリング

(能力2)

『トレーニングプログラム』キープロセスエリアを参照。

実施される活動

活動1

文書化された手順に従って、「組織の標準ソフトウェアプロセス」を開発し維持する。

この手順で明記される典型的な事項：

1. 「組織の標準ソフトウェアプロセス」は、組織に課せられたソフトウェア方針、プロセス標準、および成果物標準などを適宜満足する。
2. 「組織の標準ソフトウェアプロセス」は、その組織のプロジェクトに対し顧客から一般的に課せられるソフトウェアプロセスおよび成果物標準などを適宜満足する。
3. 一般に実績のあるソフトウェアエンジニアリングのツールおよび手法などが、適宜「組織の標準ソフトウェアプロセス」に組み込まれている。
4. ソフトウェア機能グループ間の内部プロセスインタフェースが記述されている。

ソフトウェアエンジニアリング機能の例：

- ソフトウェア要求分析
- ソフトウェア設計
- コーディング
- ソフトウェアテスト
- ソフトウェア構成管理
- ソフトウェア品質保証

(活動1)

5. ソフトウェアプロセスと他の影響を受けるグループの各種プロセス間の外部プロセスインタフェースが記述されている。

他の影響を受けるグループの例：

- システムエンジニアリング
- システムテスト
- 契約管理
- 文書化支援

6. 「組織の標準ソフトウェアプロセス」に対して提案された変更は、組み込まれる前に、組織のソフトウェアプロセス活動に責任のあるグループ（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）が文書化し、レビューし、そして承認する。

変更の情報源の例：

- ソフトウェアプロセスアセスメントの所見および勧告
- 「組織の標準ソフトウェアプロセス」をプロジェクトがテラリングした結果
- 組織やプロジェクトのソフトウェアプロセス活動をモニターすることから得られた教訓
- 組織の要員およびマネージャから提案された変更
- 分析され解釈されたプロセスおよび成果物の計測データ

7. 進行中のプロジェクトのソフトウェアプロセスに変更を取り入れるための計画が適宜定義されている。
8. 「組織の標準ソフトウェアプロセス」が最初に開発されたとき、および重要な変更もしくは追加がなされるたびに、その記述内容についてピアレビューを受ける。

(活動1)

『ピアレビュー』のキープロセスエリアを参照。

9. 「組織の標準ソフトウェアプロセス」の記述を構成管理下に置く。

『ソフトウェア構成管理』のキープロセスエリアを参照。

活動2

「組織の標準ソフトウェアプロセス」は、確立された組織標準に従って文書化する。

この標準で明記される典型的な事項：

1. プロセスは、プロセスを理解し記述するのに必要な粒度のプロセス構成要素に分解されている。

それぞれのプロセス要素は、整った形で定義され、境界が明確にされ、そして密接に関連付けられた一連の活動を取り扱う。

プロセス要素の例：

- ソフトウェア見積り
- ソフトウェア設計
- コーディング
- ピアレビュー

プロセス要素の記述は、記入すべきテンプレートであったり、完成されるべき部品であったり、改良されるべき抽象概念であったり、あるいは変更されるかまたはそのまま利用される完全な記述であったりする。

2. それぞれのプロセス要素が記述され、以下の項目を取り上げる。

- 要求される手順、プラクティス、手法、および技術

(活動2)

- 適用されるプロセスおよび成果物標準
 - プロセス実装の責任
 - 必要とされるツールおよび資源
 - インプット
 - 作成されるソフトウェア作業成果物
 - ピアレビューを受けるべきソフトウェア作業成果物
 - 開始および完了基準
 - 収集の必要がある成果物およびプロセスデータ
3. プロセス要素間の関係が記述され、以下の項目を取り上げる。
- 順序
 - インタフェース
 - 相互依存性

プロセス要素間の関係は、ソフトウェアプロセスアーキテクチャといわれることもある。

活動3

プロジェクトにおいて使用することが承認されたソフトウェアライフサイクルの記述を文書化し維持する。

ソフトウェアライフサイクルの例：

- ウォーターフォール
- オーバーラッピングウォーターフォール
- スパイラル
- シリアルビルド
- 単一プロトタイプ / オーバーラッピングウォーターフォール

1. ソフトウェアライフサイクルは、「組織の標準ソフトウェアプロセス」に適合している。

2. ソフトウェアライフサイクルの記述に対して提案された変更は、組み込まれる前に、組織のソフトウェアプロセス活動に責任のあるグループ（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）が文書化し、レビューし、そして承認する。
- (活動3)
3. ソフトウェアライフサイクルが最初に文書化されたとき、および重要な変更もしくは追加がなされるたびに、その記述内容についてピアレビューを受ける。

『ピアレビュー』のキープロセスエリアを参照。

4. ソフトウェアライフサイクルの記述は、管理され制御されている。

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キープロセスエリアに記述されている。

活動4

プロジェクトが「組織の標準ソフトウェアプロセス」をテーラリングするためのガイドラインおよび基準を開発し維持する。

1. テーラリングのガイドラインと基準は、以下の項目を扱う。
- プロジェクトに対し、ソフトウェアライフサイクルを選択することおよびテーラリングすること
 - ソフトウェアライフサイクルやプロジェクトの特徴に合わせて、「組織の標準ソフトウェアプロセス」をテーラリングすること

(活動4)

テーラリングの例：

- 新しい製品シリーズやホスト環境にプロセスを適合させる。
- 特定のプロジェクトまたはプロジェクト群にあわせて、プロセスをカスタマイズする。
- 結果的に「プロジェクトの定義されたソフトウェアプロセス」が実行できるように、プロセスを詳細化し細部を補う。

□ 「プロジェクトの定義されたソフトウェアプロセス」を文書化するための標準

2. テーラリングのガイドラインおよび基準に対して提案された変更は、組み込まれる前に、組織のソフトウェアプロセス活動に責任のあるグループ（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）が文書化し、レビューし、そして承認する。
3. テーラリングのガイドラインおよび基準は、管理され制御されている。

活動5

「組織のソフトウェアプロセスデータベース」を確立し維持する。

1. ソフトウェアプロセスおよび結果としてのソフトウェア作業成果物に関するデータを、収集し利用可能とするためにデータベースが確立されている。

プロセスおよび作業成果物データの例：

- ソフトウェア規模、工数、およびコストの見積り
- ソフトウェア規模、工数、およびコストに関する実際のデータ
- 生産性データ
- 品質の計測値
- ピアレビューのカバレッジおよび効率
- テストのカバレッジおよび効率
- ソフトウェア信頼性の尺度
- ソフトウェア要件の中で検出された欠陥の数および重大度
- ソフトウェアコードの中で検出された欠陥の数および重大度

(活動5)

2. データベースにインプットされたデータは、データベースの内容の一貫性を確実なものにするため、レビューされる。

加えて、データベースは、実際の計測データ、関連情報、およびその他のデータを含むかあるいは参照のための一覧を持つ。その他のデータは、計測データを理解し解釈し、その妥当性と適用可能性をアセスメントするために必要なものである。

3. データベースは、管理され制御されている。
4. データの完全性、一貫性、および正確性を確実なものにするため、データベースの内容へのユーザアクセスが制御されている。

アクセスは、データの入力、変更、閲覧、分析、および抽出を行う必要のある人に制限されている。

機密データは保護され、これらのデータへのアクセスは適切に制御されている。

活動6 「ソフトウェアプロセス関連文書のライブラリ」を確立し維持する。

ソフトウェアプロセス関連文書の例：

- 「プロジェクトの定義されたソフトウェアプロセス」の記述
- プロジェクトの標準
- プロジェクトの手順
- プロジェクトのソフトウェア開発計画
- プロジェクトの計測計画
- プロジェクトのプロセストレーニング資料

(活動6)

1. 登録候補の文書アイテムはレビューされ、将来役立ちそうな適切なアイテムがライブラリへ登録される。
2. アクセスが容易なように、文書アイテムは分類されている。
3. その時点でライブラリに登録されている文書アイテムのレビジョンがレビューされ、ライブラリの内容が適宜更新されている。
4. ライブラリの内容は、ソフトウェアプロジェクトおよびソフトウェア関連グループが利用可能な形にされている。

ソフトウェア関連グループの例：

- ソフトウェア品質保証
- ソフトウェア構成管理
- ソフトウェアテスト
- 文書化支援

5. 各文書アイテムの利用は定期的にレビューされ、その結果がライブラリの内容物を保守するために使われている。
6. ライブラリの内容物は、管理され制御されている。

計測と分析

計測1 計測を行い、その結果を使用して、組織のプロセス定義の活動状況を判断する。

計測値の例：

- プロセス開発ならびに維持に関するスケジュール上のマイルストーンの状況
- プロセス定義活動のコスト

履行検証

検証1 ソフトウェア品質保証グループは、「組織の標準ソフトウェアプロセス」および関連するプロセス資産の開発や維持のための組織活動と作業成果物をレビューかつ/または監査し、その結果を報告する。

『ソフトウェア品質保証』のキープロセスエリアを参照。

レビューや監査では、最低限、以下のことを検証する：

1. 「組織の標準ソフトウェアプロセス」ならびに関連するプロセス資産の開発、文書化、および維持において、適切な標準に従っている。
2. 「組織の標準ソフトウェアプロセス」ならびに関連するプロセス資産が、適切に制御され使用されている。

トレーニングプログラム

レベル3のキープロセスエリア： 定義された段階

『トレーニングプログラム』キープロセスエリアの目的は、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。

『トレーニングプログラム』に必然的に含まれる活動には、まず組織、プロジェクト、および個人のトレーニングニーズを特定すること、次にそのニーズに対応するトレーニングを開発または調達することが挙げられる。

各ソフトウェアプロジェクトは、現在および将来必要となるスキルを評価し、そのスキルをどのようにして獲得するかを決定する。スキルの中には、非公式な手段（例えば、OJTや非公式なメンタリング(訳注：先輩による指導)など）によって効果的かつ効率的に供与されるものもある。一方、効果的かつ効率的に供与するには、公式のトレーニング手段（例えば、教室形式トレーニングや指導付き自習など）が必要となるスキルもある。適切な手段を選んで用いる。

このキープロセスエリアでは、トレーニングの機能を実施するグループに関するプラクティスを扱っている。特定のトレーニング題材（必要とされる知識やスキル）を明らかにするプラクティスは、個々のキープロセスエリアの実施能力コモンフィーチャに含まれる。

ゴール

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。

実施のコミットメント

- コミットメント1 組織は、トレーニングのニーズを満たすことに関して、明文化された方針に従う。

この方針で明記される典型的な事項：

1. ソフトウェアの管理面や技術面の各役割に必要なスキルと知識が特定される。
2. スキルと知識を供与するトレーニング手段が特定され承認される。

承認されるトレーニング手段の例：

- 教室形式トレーニング
- コンピュータ支援による教育
- 指導付きの自習
- 公式の実習やメンタリングプログラム
- ビデオ教材

3. 組織のスキル基盤の構築、プロジェクト特有のニーズの充足、および個人スキルの開発のため、トレーニングが提供される。
4. トレーニングは、組織内で開発されるかまたは適切な場合に外部から調達される。

(コミットメント1)

組織外トレーニングの例：

- 顧客が提供するトレーニング
- 商業的トレーニングコース
- 学校教育プログラム
- 専門家向けのコンファレンス
- セミナー

実施能力

能力1

組織のトレーニングニーズの充足に責任を持つグループが存在する。

トレーニンググループのメンバは、組織内から専任または専任でないインストラクタを選び出しても良いし、組織外部のメンバを含めてもよい。

グループは、一連のタスクまたは活動に責任を持つ部門、マネージャ、および個人の集まりである。グループは、専任でない個人の場合、異なる部門から割り当てられた専任でない人達の場合、および専任の人達の場合など、さまざまな形態がある。グループ編成にあたっては、割り当てるタスクや活動、プロジェクト規模、組織構造、および組織の文化などを考慮する必要がある。品質保証グループのようにプロジェクトの活動に焦点を当てるグループもあれば、ソフトウェアエンジニアリングプロセスグループのように組織全体の活動に焦点を当てるグループもある。

能力2

トレーニングプログラムを履行するために、適切な資源と資金が提供される。

(能力2)

トレーニングプログラムの構成要素の例：

- 組織のトレーニング計画
- トレーニング教材
- トレーニングの開発や調達
- トレーニングの開催
- トレーニング設備
- トレーニングの評価
- トレーニング記録の維持

1. 組織のトレーニングプログラムの実施に責任を持つマネージャが任命される。
2. トレーニングプログラム活動の支援ツールが利用可能になっている。

支援ツールの例：

- ワークステーション
- 教育用の設計ツール
- データベースプログラム
- プレゼンテーション教材開発のパッケージ

3. 適切な施設がトレーニング実施のために利用可能になっている。

教室でのトレーニングに用いる施設は、妨害されないように、受講者の作業環境から分離されているのがよい。

必要な場合は、トレーニングは、実際の行動条件と極めて似た状況で開催され、実際の作業状態をシミュレートする活動を含むようにする。

能力3

トレーニンググループのメンバは、各自のトレーニング活動を実施するために必須のスキルと知識を備えている。

(能力3)

知識とスキルの提供方法の例：

- 教育技法のトレーニング
- 題材の内容についてのリフレッシュトレーニング

能力4

ソフトウェアマネージャは、トレーニングプログラムについてオリエンテーションを受ける。

実施される活動

活動1

各ソフトウェアプロジェクトは、そのトレーニングニーズを明記するトレーニング計画を策定し保守する。

この計画が扱う事項：

1. 必要なスキルのセットと、それらのスキルが必要となる時期
2. トレーニングが必要なスキルと、他の方法で習得するスキル

スキルには、非公式なトレーニング手段（例えば、非公式のトレーニングやプレゼンテーション、本や雑誌を読むこと、チョークトーク(黒板を用いた講義)、弁当持参の昼食時セミナー、OJT、および非公式なメンタリングなど）によって習得する方が効果的かつ効率的なものもあれば、より公式なトレーニング手段（例えば、教室形式トレーニング、コンピュータ支援による教育、指導付きの自習、ビデオ教材、および公式な実習とメンタリングプログラム）が必要なスキルもある。

3. どんなトレーニングを、だれが、いつ、必要としているか。

(活動1)

トレーニングニーズの具体例については、他のすべてのキープロセスエリアの「実施能力」コモンフィーチャを参照。

適宜、個人に対するトレーニングをその人の作業責任に結びつける。そのことにより、OJT活動またはトレーニング以外の経験が、トレーニング修了後の妥当な期間でトレーニング内容を補強するようになる。

4. どのようにしてトレーニングを提供するか。

トレーニングには、ソフトウェアプロジェクト、組織のトレーニンググループ、および外部組織によって提供されるものがある。

ソフトウェアプロジェクトで適切に行うトレーニングの例：

- プロジェクトに特有なアプリケーションや要件に関するトレーニング
- プロジェクトのソフトウェアアーキテクチャに関するトレーニング
- その他、プロジェクトレベルで実施した方が効果的または効率的なトレーニング

活動2

文書化された手順に従って、組織のトレーニング計画を策定し改訂する。

この手順で明記される典型的な事項：

1. 組織のトレーニング計画には、ソフトウェアプロジェクトのトレーニング計画で特定されたトレーニングニーズを用いる。
2. 提供される特定のトレーニングは、組織が必要とするスキルやそれらのスキルが必要とされる時期に基づいて明らかにされる。

(活動2)

3. 組織のトレーニング計画は、変更を組み込むために、適宜改訂される。
4. 組織のトレーニング計画は、初期のリリース時や大規模改訂のたびに、影響を受ける個人によってレビューされる。

影響を受ける個人の例：

- 上級管理層
- ソフトウェアマネージャ
- ソフトウェア関連グループのマネージャ

5. 組織のトレーニング計画は、管理され制御されている。

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キーププロセスエリアに記述されている。

6. 組織のトレーニング計画は、影響を受けるグループや個人が容易に利用可能である。

(活動2)

影響を受けるグループや個人の例：

- 上級管理層
- トレーニンググループ
- ソフトウェア関連グループのマネージャ
- ソフトウェアエンジニアリング（ソフトウェア設計などのすべてのサブグループを含む）
- ソフトウェア見積り
- システムエンジニアリング
- システムテスト
- ソフトウェア品質保証
- ソフトウェア構成管理
- 契約管理
- 文書化支援

活動3

組織のトレーニング計画に従って、組織のトレーニングを実施する。

この計画が扱う事項：

1. 組織内で必要とされる特定のトレーニング、およびそれが必要とされる時期。
2. 組織外から調達されるトレーニングとトレーニンググループが提供するトレーニング。
3. トレーニングの準備と開催、または調達に必要な資金と資源（要員、ツール、および設備を含む）。
4. トレーニンググループで開発されるトレーニングコース用教材の標準。
5. トレーニンググループが開発するトレーニングコースの開発と改訂のスケジュール。

6. 必要時期や受講者数の予想を踏まえて設定するトレーニングの実施スケジュール。

(活動3)

7. 以下の手順：
 - トレーニング受講者の選定
 - トレーニング受講の登録と参加方法
 - 提供されたトレーニング記録の保守
 - トレーニングの評価とトレーニングからのその他フィードバックの収集、レビュー、および活用

活動4

組織レベルで準備するトレーニングコースは、組織標準に従って開発し保守する。

これらの標準が要求する事項：

1. 各トレーニングコースの説明が作成されている。

説明に取り上げるトピックの例：

- 想定受講者
- 参加のための準備
- トレーニング目的
- トレーニング期間
- レッスン計画
- 受講者の修了判定基準
- トレーニングの有効性を定期的に評価する手順
- 特別な考慮事項で、トレーニングコースの試行や実地テスト、リフレッシュ用トレーニングのニーズ、およびフォローアップトレーニング機会、など

2. トレーニングコース教材はレビューされる。

(活動4)

トレーニング教材をレビューする個人の例：

- 教育専門家
- 題材の内容に関する専門家
- レビューされるトレーニングコースの試行に参加した受講生の代表

3. トレーニングコースの教材は、管理され制御されている。

活動5

任命された役割を遂行するのに必要な知識とスキルが、すでに個人の身に付いているかどうかを判断するために、必修トレーニングに対する免除手順を確立し使用する。

活動6

トレーニング記録を維持する。

1. 各トレーニングコースまたは他の承認されたトレーニング活動の修了者全員の記録が保持される。
2. 指定された必修トレーニングの修了者全員の記録が保持される。
3. トレーニング修了者の記録が、要員やマネージャの任命検討に利用できる。

計測と分析

計測1

計測を行い、その結果を使用して、トレーニングプログラムの活動状況を判断する。

(計測1)

計測値の例：

- 各トレーニングコースの実際の出席者数で、予想受講者数と比較したもの
- トレーニングコース提供の進捗で、組織やプロジェクトのトレーニング計画と比較したもの
- 承認された受講免除者の延べ人数

計測2

計測を行い、その結果を使用して、トレーニングプログラムの品質を判断する。

計測値の例：

- トレーニング修了後のテストの結果
- 受講者によるコースのレビュー
- ソフトウェアマネージャからのフィードバック

履行検証

検証1

トレーニングプログラムの活動は、上級管理層によって定期的にレビューされる。

上級管理層による定期的なレビューの主目的は、ソフトウェアプロセス活動に対する認識と見通しを上級管理層に提供することである。活動のレビューは、適切な抽象レベルでタイムリーに行う。レビューの間隔は、組織のニーズに合わせる。例外報告への体制が十分整っていれば、上級管理層のレビュー間隔は長くても構わない。

(検証1)

上級管理層の監督レビューに関して、その典型的な内容を扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キーププロセスエリアの検証1を参照。

検証2

トレーニングプログラムは、組織ニーズとの首尾一貫性や関連性について、定期的に独立して評価される。

検証3

トレーニングプログラムの活動と作業成果物は、レビューかつ/または監査され、その結果が報告される。

レビューや監査では、最低限、以下のことを検証する：

1. 組織のトレーニング計画を策定し改訂するプロセスが守られている。
2. トレーニングコースを開発し改訂するプロセスが守られている。
3. トレーニング記録が適切に保守されている。
4. 特定のトレーニングが必要であると指名された人は、そのトレーニングを修了している。
5. 組織のトレーニング計画が守られている。

ソフトウェア統合管理

レベル3のキープロセスエリア： 定義された段階

『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ー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。

『ソフトウェア統合管理』に必然的に含まれる活動には、「プロジェクトの定義されたソフトウェアプロセス」を開発すること、およびこの定義されたソフトウェアプロセスを用いてソフトウェアプロジェクトを管理することが挙げられる。「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」をもとにプロジェクト固有の特徴に合わせてテーラリングされる。

ソフトウェア開発計画は、「プロジェクトの定義されたソフトウェアプロセス」に基づいて、どのように「プロジェクトの定義されたソフトウェアプロセス」の活動が履行され管理されるかということを記述したものである。ソフトウェアプロジェクトの規模、工数、コスト、スケジュール、配員、およびその他の資源の管理は、「プロジェクトの定義されたソフトウェアプロセス」のタスクに結び付けられる。

「プロジェクトの定義されたソフトウェアプロセス」は、すべて「組織の標準ソフトウェアプロセス」からテーラリングされる。したがって、プロセスデータや教訓は、複数のソフトウェアプロジェクトで共有することができる。

ソフトウェアプロジェクトの見積り、計画、および進捗の確認に関する基本的なプラクティスについては、『ソフトウェアプロジェクト計画』および『ソフトウェアプロジェクト進捗管理』キープロセスエリアに記述されている。この2つのキープロセスエリアでは、問題が発生したときにそれを認識し、その問題を取り上げるために計画や行動を補正することに重点が置かれている。『ソフトウェア統合管理』キープロセスエリアのプラクティスは、この2つのキープロセスエリアのプラクティスを基にして、追加されたものである。『ソフトウェア統合管理』キープロセスエリアでは、問題の想定、およびこれらの問題による影響の予防あるいは最小化の活動に重点を移している。

ゴール

- 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」をテーラリングしたものである。
- 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」に従って、プロジェクトが計画され管理されている。

実施のコミットメント

- コミットメント1 プロジェクトは、「組織の標準ソフトウェアプロセス」および関連するプロセス資産を用いてソフトウェアプロジェクトを計画し管理することに関して、そのことを要求する明文化された組織方針に従う。

「組織の標準ソフトウェアプロセス」および関連するプロセス資産を扱うプラクティスについては、『組織プロセス定義』キープロセスエリアを参照。

(コミットメント1) この方針で明記される典型的な事項：

1. それぞれのプロジェクトは、「組織の標準ソフトウェアプロセス」をテーラリングし、「プロジェクトの定義されたソフトウェアプロセス」を文書化する。
2. 「組織の標準ソフトウェアプロセス」からプロジェクトが逸脱した事項は、文書化され承認されている。
3. それぞれのプロジェクトは、「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、ソフトウェア活動を実施する。
4. それぞれのプロジェクトは、適切なプロジェクト計測データを収集し、「組織のソフトウェアプロセスデータベース」に蓄積する。

「組織のソフトウェアプロセスデータベース」を扱うプラクティスについては、『組織プロセス定義』キープロセスエリアの活動5を参照。

実施能力

能力1

「プロジェクトの定義されたソフトウェアプロセス」を用いたソフトウェアプロジェクト管理のために、適切な資源と資金が提供される。

ソフトウェアプロジェクト計画と進捗管理に関する資源と資金を扱うプラクティスについては、『ソフトウェアプロジェクト計画』キープロセスエリアの能力3、および『ソフトウェアプロジェクト進捗管理』キープロセスエリアの能力3を参照。

能力2

「プロジェクトの定義されたソフトウェアプロセス」の開発責任者は、どのように「組織の標準ソフトウェアプロセス」をテーラリングし、関連するプロセス資産を利用するかについて、必修トレーニングを受ける。

(能力2)

トレーニングの例：

- ソフトウェアプロセスデータベースの使用
- 「組織の標準ソフトウェアプロセス」の使用
- 「組織の標準ソフトウェアプロセス」をテーラリングしてソフトウェアプロジェクトのニーズを満足するためのガイドラインや基準の使用

『トレーニングプログラム』キーププロセスエリアを参照。

能力3

ソフトウェアマネージャは、ソフトウェアプロジェクトの技術、運営、および人事の各側面に関し、「プロジェクトの定義されたソフトウェアプロセス」に基づいた管理についての必修トレーニングを受ける。

ソフトウェアプロジェクトの計画と進捗管理に対するトレーニングを扱うプラクティスについては、『ソフトウェアプロジェクト計画』キーププロセスエリアの能力4、および『ソフトウェアプロジェクト進捗管理』キーププロセスエリアの能力4を参照。

トレーニングの例：

- 「プロジェクトの定義されたソフトウェアプロセス」に基づいたソフトウェアの見積り、計画、および進捗確認の手法と手順
- ソフトウェアリスクの特定、管理、および伝達の手法と手順

『トレーニングプログラム』キーププロセスエリアを参照。

実施される活動

活動1

文書化された手順に従って「組織の標準ソフトウェアプロセス」をテーラリングすることにより、「プロジェクトの定義されたソフトウェアプロセス」を開発する。

組織の標準ソフトウェアプロセスの内容を扱うプラクティスについては、『組織プロセス定義』キープロセスエリアの活動2を参照。

この手順で明記される典型的な事項：

1. ソフトウェアのライフサイクルは、
 - 組織承認が得られているライフサイクルから選択され、プロジェクトの契約や実施に関する制約条件を満足させる。

承認されたソフトウェアライフサイクルの扱うプラクティスについては、『組織プロセス定義』キープロセスエリアの活動3を参照。

- 必要ならば、組織のテーラリングのガイドラインや基準により許可された方法で修正されている。
- 組織の標準に従って文書化されている。

組織のテーラリングのガイドラインや基準を扱うプラクティスについては、『組織プロセス定義』キープロセスエリアの活動4を参照。

2. 「プロジェクトの定義されたソフトウェアプロセス」の記述は文書化されている。

(活動1)

プロセス定義で期待される内容を扱うプラクティスについては、『組織プロセス定義』キープロセスエリアの活動2を参照。

テーラリングでは、組織のプロセス資産を適宜使用する。

3. プロジェクトに対する「組織の標準ソフトウェアプロセス」のテーラリングは、組織のソフトウェアプロセス活動の調整に責任のあるグループ（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）によってレビューされ、上級管理層により承認される。

ソフトウェアプロセス関連文書のライブラリを扱うプラクティスについては、『組織プロセス定義』キープロセスエリアの活動6を参照。

- 「組織の標準ソフトウェアプロセス」からの逸脱に関する免除事項は文書化され、そして上級管理層によりレビューされ承認される。
4. 契約上のソフトウェアプロセス要件からの逸脱に関する免除事項は文書化され、そして上級管理層やソフトウェアプロジェクトの顧客などにより適宜レビューされ承認される。
 5. 「プロジェクトの定義されたソフトウェアプロセス」の記述は、管理され制御されている。

(活動1)

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キープロセスエリアに記述されている。

活動2

それぞれの「プロジェクトの定義されたソフトウェアプロセス」は、文書化された手順に従って改訂する。

この手順で明記される典型的な事項：

1. 下記項目に由来する変更は、文書化され系統的にレビューされる：
 - 組織におけるプロジェクトのソフトウェア活動をモニターすることから得られた教訓
 - ソフトウェアプロジェクトで提案された変更
 - プロセスと作業成果物の計測データ
2. 「プロジェクトの定義されたソフトウェアプロセス」の変更については、変更が組み込まれる前に、レビューされ承認される。

変更をレビューする個人の例：

- 組織のソフトウェアプロセス活動に責任をもつグループのメンバー（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）
- ソフトウェアマネージャ
- プロジェクトのソフトウェアマネージャ

(活動2)

変更を承認する担当者の例：

- プロジェクトのソフトウェアマネージャ
- プロジェクトマネージャ

活動3

プロジェクトのソフトウェア開発計画は、「プロジェクトの定義されたソフトウェアプロセス」の使用を記述するものであり、文書化された手順に従って策定し改訂する。

ソフトウェア開発計画を扱うプラクティスについては、『ソフトウェアプロジェクト計画』キープロセスエリアの活動6、7、および『ソフトウェアプロジェクト進捗管理』キープロセスエリアの活動1、2を参照。

活動4

「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、ソフトウェアプロジェクトを管理する。

ソフトウェアプロジェクトの管理を扱う基本的なプラクティスについては、『ソフトウェアプロジェクト計画』および『ソフトウェアプロジェクト進捗管理』キープロセスエリアを参照。

この「プロジェクトの定義されたソフトウェアプロセス」で明記される典型的な事項：

1. ソフトウェアプロジェクトの管理に必要な計測データの収集、分析、および報告に関する規定が作成されている。
2. ソフトウェアの見積り、計画、および進捗の確認に関する活動は、「プロジェクトの定義されたソフトウェアプロセス」の主要なタスクや作業成果物と結び付けられている。
3. 開始と完了の基準が確立され、文書化され、そして主要タスクの開始認可や完了決定に用いられている。

(活動4)

4. ソフトウェアプロジェクトの再計画時期を示す文書化された基準が定義されている。
5. 技術や管理に関する教訓は文書化され、組織のソフトウェアプロセス関連文書ライブラリに蓄積される。

組織のソフトウェアプロセス関連文書ライブラリを扱うプラクティスについては、『組織プロセス定義』キープロセスエリアの活動6を参照。

6. 組織内における他のプロジェクトの活動をモニターすることで得られた技術や管理に関する教訓は、系統的にレビューされ、ソフトウェアプロジェクトの見積り、計画、進捗の確認、および再計画に利用される。
7. 配員計画では、特別なスキルやアプリケーションドメインの知識を持つ個人に対するソフトウェアプロジェクトのニーズを取り上げる。
8. ソフトウェアプロジェクト特有のニーズにあうように、トレーニングのニーズが特定され文書化される。

プロジェクトのトレーニングにおけるニーズの特定を扱うプラクティスについては、『トレーニングプログラム』キープロセスエリアの活動1を参照。

9. 他のグループとのやり取りで用いるソフトウェア計画とソフトウェアプロセスは、各グループ間の差異や潜在的な問題を考慮に入れて補正される。

(活動4)

差異事項や問題点の例：

- プロセス成熟度の違い
- プロセスの不整合
- 各種事業要因

活動5

「組織のソフトウェアプロセスデータベース」は、ソフトウェアの計画と見積りに使用する。

「組織のソフトウェアプロセスデータベース」を扱うプラクティスについては、『組織プロセス定義』キープロセスエリアの活動5を参照。

1. データベースは、ソフトウェアプロジェクトの見積り、計画、進捗の確認、および再計画のためのデータソースとして利用される。可能ならば、類似のソフトウェアプロジェクトのデータが利用される。

「組織のソフトウェアプロセスデータベース」に含まれるデータの例：

- ソフトウェア作業成果物の規模
- ソフトウェア工数
- ソフトウェアコスト
- スケジュール
- 配員
- 技術活動

2. ソフトウェアの規模、工数、コスト、スケジュール、および重要なコンピュータ資源の見積りを得るために使われるパラメータ値は、その妥当性を評価するために他のソフトウェアプロジェクトのパラメータ値と比較される。
 - アプリケーションドメインや設計アプローチの面で、他のプロジェクトとの類似点や相違点が評価され記録される。

(活動5)

- パラメータ値間の類似点や相違点の論理的根拠は記録される。
 - プロジェクト見積りの信頼性判定に用いた根拠は記録される。
3. ソフトウェアプロジェクトは、「組織のソフトウェアプロセスデータベース」に蓄積するためのソフトウェア計画データ、再計画データ、および実際の計測データを適切に提供する。

ソフトウェアプロジェクトで記録されるデータの例：

- タスク記述
- 仮定
- 見積り
- 改訂された見積り
- 実際の計測データ
- 見積りを再構築し、それらの妥当性をアセスメントし、そして新しい作業についての見積りを導出するために必要な関連情報

活動6

ソフトウェア作業成果物の規模（またはソフトウェア作業成果物の変更規模）は、文書化された手順に従って管理する。

ソフトウェア作業成果物の規模の計画、および進捗の確認を扱う基本的なプラクティスについては、『ソフトウェアプロジェクト計画』キープロセスエリアの活動9、および『ソフトウェアプロジェクト進捗管理』キープロセスエリアの活動5を参照。

この手順で明記される典型的な事項：

1. 信頼性の高い見積りを確立するために、ソフトウェアエンジニアリンググループとは独立したグループが、ソフトウェア作業成果物の規模見積りに関する手順をレビューし、「組織のソフトウェアプロセスデータベース」における過去のデータの利用についてガイダンスを提供する。

(活動6)

独立したグループの例としては、ソフトウェア見積りグループがある。

ソフトウェアの規模見積りの信頼性を評価する手法の例としては、完成したシステムと機能毎に比較する方法がある。

- 規模見積りをする個人は、見積りに使われる手順やデータが適切であることを確実なものにする。
 - 規模見積りの妥当性に疑問があるときは、同僚と専門家から構成されるチームによって見積りをレビューする。
2. ソフトウェアリスクとして特定されたそれぞれのソフトウェア要素については、規模見積りにリスク対応係数が付与される。
 - リスク対応策についての論理的根拠が文書化される。
 - リスク対応策の削減あるいは除去に関するリスクは、アセスメントされ文書化される。
 3. 市販あるいは再利用可能なソフトウェアコンポーネントは特定される。
 - 再利用の計測は、要件、設計、コード、テスト計画、およびテスト手順などの再利用を対象とする。
 - 再利用可能なコンポーネントの修正や組み込みの工数は、規模見積り要素に取り入れられる。
 4. ソフトウェア作業成果物の規模に大きな影響を及ぼす要因は特定され、念入りにモニターされる。
 5. それぞれの管理されたソフトウェア要素に対して、規模しきい値が確立される。規模しきい値を超過すると予想されるときには、対処が必要となる。

活動7

プロジェクトのソフトウェア工数とコストは、文書化された手順に従って管理する。

(活動7)

ソフトウェア工数とコストの計画および進捗の確認を扱う基本的なプラクティスについては、『ソフトウェアプロジェクト計画』キープロセスエリアの活動10、および『ソフトウェアプロジェクト進捗管理』キープロセスエリアの活動6を参照。

この手順で明記される典型的な事項：

1. ソフトウェア工数、コスト、および配員プロファイルモデル（もし使用されるならば）は、プロジェクトに適合され、適宜利用できる過去のデータを使用する。
2. 参考にされる生産性やコストデータは、プロジェクト変数を組み込んで補正される。

プロジェクト変数の例：

- プロジェクトのグループや組織（例えば、外注先）の所在地
- システムの規模と複雑さ
- 要件の安定性
- 開発におけるホスト環境
- 対象とするシステム環境
- 開発者のアプリケーションに関する習熟度と経験
- 資源の利用可能性
- その他の特別な制約

3. 工数とコストを効果的に管理するために、全体のソフトウェア工数とコストは、個々に管理されるタスクあるいはステージに必要な応じて割り当てられる。
4. ソフトウェア工数とコストの状態がレビューされ見積りが改訂されるときには、これまでの実際の消費量および完了した作業に対する実際の消費量を、ソフトウェア開発計画での見積りと比較し、残る作業に対する工数とコストの見積りを改良するために利用される。

(活動7)

- ソフトウェア工数とコストの見積りに用いられたモデルのパラメータ値は、ソフトウェア要件に大きな変更が生じるたびに更新される。
 - 適宜、プロジェクト生産性や他の新しいソフトウェアコストに関する実際のデータが用いられる。
5. 個々に管理されるそれぞれソフトウェアタスクあるいはステージに対して、工数とコストのしきい値が確立される。これらのしきい値を超過すると予測されるときには、対処が必要となる。

活動8

プロジェクトの重要なコンピュータ資源は、文書化された手順に従って管理する。

重要なコンピュータ資源の計画および進捗の確認を扱う基本的なプラクティスについては、『ソフトウェアプロジェクト計画』キープロセスエリアの活動11、および『ソフトウェアプロジェクト進捗管理』キープロセスエリアの活動7を参照。

この手順で明記される典型的な事項：

1. プロジェクトの重要なコンピュータ資源に対する見積りは、過去の経験、シミュレーション、プロトタイプ作成、あるいは分析などに基づいて適宜導出される。
 - 見積りの情報源と論理的根拠は文書化される。
 - アプリケーションドメインや設計アプローチの面に関するプロジェクトと過去のデータソース間の類似点および相違点は、アセスメントされ、記録される。
 - 見積りの信頼性判定に用いた根拠は記録される。
2. 計画されたコンピュータ資源、「ソフトウェアに割り当てられたシステム要件」、ソフトウェア要件、かつ/またはソフトウェア設計は、プロジェクトの重要なコンピュータ資源要件を達成するために補正される。
3. 利用可能なコンピュータ資源は、ソフトウェアコンポーネントに割り当てられる。

(活動8)

4. 初期見積り時には、重要なコンピュータ資源の最大容量に対して、具体的な予備容量含めておく。
5. それぞれの重要なコンピュータ資源に対してしきい値が確立される。しきい値を超過すると予測されるときには、対処が必要となる。

活動9

プロジェクトのソフトウェアスケジュールの重要な依存関係とクリティカルパスは、文書化された手順に従って管理する。

重要な依存関係の協議および確認を扱うプラクティスについては、『ソフトウェアプロジェクト計画』キープロセスエリアの活動12、『ソフトウェアプロジェクト進捗管理』キープロセスエリアの活動8、および『グループ間調整』キープロセスエリアの活動4を参照。

この手順で明記される典型的な事項：

1. マイルストーン、タスク、コミットメント、重要な依存関係、配員、コスト、およびレビューは、「プロジェクトの定義されたソフトウェアプロセス」と首尾一貫したスケジュールに割り当てられる。
- ソフトウェアスケジュールでは、完了が客観的に判断できる特定のタスクやマイルストーンを明らかにする。(0,1あるいはYes / Noでの判断)

異なったグループや個人のニーズに合わせるために、それぞれが適切に結び付けられた異なったレベルの詳細スケジュールが策定される。

2. 重要な依存関係は、定義され、協議され、そしてソフトウェアスケジュールに反映される。

(活動9)

重要な依存関係には、ソフトウェアエンジニアリンググループ内部（サブグループ間）の依存関係と、ソフトウェアエンジニアリンググループと他の影響を受けるグループの依存関係が含まれる。

3. スケジュールのクリティカルパスは、定義されそしてソフトウェアスケジュールに反映される。
4. ソフトウェアプロジェクトの重要な依存関係とスケジュールのクリティカルパスの進捗を、定期的に確認される。
5. それぞれのクリティカルパスに対し、文書化された特定のしきい値基準が確立される。しきい値を超過すると予測されるときには、処置が必要となる。

処置の例：

- 機能、品質、コスト、スケジュール、配員、およびその他の資源のトレードオフについて、分析とシミュレーションを行う。
- リスク対応策を割り当て、可能であればスケジュールを緩める。
- すべてのクリティカルパスに対して、考えた処置の効果を評価する。
- 影響を受けるグループが視認できる様な決定を下す。

活動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従って特定し、アセスメントし、文書化し、そして管理する。

リスクの特定と進捗確認を扱う基本的なプラクティスについては、『ソフトウェアプロジェクト計画』キープロセスエリアの活動13、および『ソフトウェアプロジェクト進捗管理』キープロセスエリアの活動10を参照。

(活動10)

管理されるべきソフトウェアリスクの例として、ソフトウェアプロジェクトが目標の達成に失敗するという重大な可能性が挙げられる。そのような目標分野の例：

- スケジュール
- コスト
- 機能充足性
- スループットまたはリアルタイム性能
- 信頼性または可用性
- 重要なコンピュータ資源の使用

リスク管理活動の例：

- リスクの高いプロジェクト目標を早期に特定すること
- リスクを導入または増大させるイベントを特定すること
- リスクの高いモジュールのプロトタイプを作成または早期実装すること
- プロジェクトの主要リスク指標を念入りにモニターすること

この手順で明記される典型的な事項：

1. ソフトウェアリスク管理計画は文書化され、ソフトウェアリスクの特定と管理のために活用される。

(活動10)

ソフトウェアリスク管理計画の項目の例：

- 必要な資源（要員とツールを含む）
- リスク管理手法（例えば、特定、分析、優先付け、計画、モニターすること、および解決）
- 特定されたリスクのリスト（アセスメント、優先付け、状況、および計画を含む）
- リスク管理のスケジュール
- 責任と権限
- リスク状況および活動の伝達手法と頻度
- 計測

2. リスク対応計画は、「プロジェクトの定義されたソフトウェアプロセス」に基づいて、プロジェクトのソフトウェアライフサイクル全般にわたって実施される。

リスク対応計画の活動で扱われるエリアの例：

- オプションの特定
- オプションの影響アセスメント
- オプションの技術的な実現可能性
- 管理予備費の割り当て
- オプション実行の判断基準

3. それぞれのソフトウェアリスクに対する代替案が、可能な場合には代替案の選択基準と共に定義される。
4. ソフトウェアリスク管理計画の初期リリース時および主要となる改訂時には、ピアレビューを受ける。

『ピアレビュー』キープロセスエリアを参照。

5. ソフトウェアリスク管理計画は、管理され制御されている。

(活動10)

6. ソフトウェアリスクは、選択されたプロジェクトのマイルストーン、指定されたリスクのチェックポイント、およびソフトウェアプロジェクトに影響がある重大な変更計画の中で、確認され、再アセスメントされ、そして再計画される。
 - 上記の各再アセスメント時に、リスク優先順位とソフトウェアリスク管理計画はレビューされ改訂される。
 - リスクをモニターすることで得られた情報は、リスクアセスメントとソフトウェアリスク管理計画の改良に活用される。
7. ソフトウェアリスク、ソフトウェアリスク管理計画、およびリスク軽減の結果伝達には、ソフトウェアエンジニアリンググループと他の影響を受けるグループおよび個人が含まれる。

影響を受けるグループおよび個人の例：

- 顧客
- 外注先
- エンドユーザ
- ソフトウェア見積り
- システムエンジニアリング
- システムテスト
- ソフトウェア品質保証
- ソフトウェア構成管理
- 契約管理
- 文書化支援

活動11

ソフトウェアプロジェクトのレビューを定期的 to 実施し、ソフトウェアプロジェクトの実際の行動や結果を事業、顧客、およびエンドユーザなどの現在および予測されるニーズに適宜調和させるのに必要な処置を決定する。

(活動11)

処置の例：

- スケジュールの加速
- 市場機会の変化、または顧客とエンドユーザのニーズの変更に応じたシステム要件の変更
- プロジェクトの終結

これらのプラクティスで意味するエンドユーザとは、顧客の任命するエンドユーザ、またはエンドユーザの代表者をいう。

計測と分析

計測1

計測を行い、その結果を使用して、ソフトウェア統合管理活動の有効性を判断する。

計測値の例：

- ソフトウェアプロジェクトの管理に費やされた延べ工数で、計画と比較されたもの
- 工数再計画の頻度、原因、および規模
- 特定された各々のソフトウェアリスクに関し、実際に発生した有害な影響で、見積った損失と比較したもの
- ソフトウェアプロジェクトにおいて想定しなかった、主な有害な影響の数と規模で、これまでに確認されたもの

履行検証

検証1

ソフトウェアプロジェクト管理の活動は、上級管理層によって定期的にレビューされる。

(検証1)

上級管理層の監督レビューに関して、その典型的な内容を扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証1を参照。

検証2

ソフトウェアプロジェクト管理の活動は、プロジェクトマネージャによって定期的に、かつイベント発生を契機としてレビューされる。

プロジェクトマネージャの監督レビューに関して、その典型的な内容を扱うプラクティスについては『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証2を参照。

検証3

ソフトウェア品質保証グループは、ソフトウェアプロジェクト管理の活動と作業成果物をレビューかつ/または監査し、その結果を報告する。

『ソフトウェア品質保証』キープロセスエリアを参照。

レビューや監査では、最低限、以下のことを検証する：

1. 「プロジェクトの定義したソフトウェアプロセス」の開発と改訂に関するプロセス。
2. プロジェクトのソフトウェア開発計画およびソフトウェアリスク管理計画の策定に関するプロセス。
3. 「プロジェクトの定義したソフトウェアプロセス」に従ったプロジェクト管理に関するプロセス。
4. 適切なデータの収集と「組織のソフトウェアプロセスデータベース」への提供に関するプロセス。

(検証3)

5. ソフトウェアプロジェクトの計画、見積り、および進捗確認の活動を支援するための「組織のソフトウェアプロセスデータベース」利用に関するプロセス。

ソフトウェアプロダクトエンジニアリング

レベル3のキープロセスエリア： 定義された段階

『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』に必然的に含まれる活動には、「プロジェクトの定義されたソフトウェアプロセス」（『ソフトウェア統合管理』キープロセスエリアで記述されている）と適切な手法やツールを用いて、ソフトウェアを構築し保守するためのエンジニアリングタスクを実施することが挙げられる。

ソフトウェアエンジニアリングタスクには、「ソフトウェアに割り当てられたシステム要件」の分析、ソフトウェア要件の開発、ソフトウェアアーキテクチャの開発、ソフトウェアの設計、ソフトウェアのコードへの実装、ソフトウェアコンポーネントの統合、およびソフトウェアのテストが含まれる。なお、システム要件については『要件管理』キープロセスエリアで記述されている。ソフトウェアのテストは、明記された要件（「ソフトウェアに割り当てられたシステム要件」とソフトウェア要件）をソフトウェアが満足していることを検証するために行う。

ソフトウェアエンジニアリングタスクの実施に必要な文書（例えば、ソフトウェア要件文書、ソフトウェア設計文書、テスト計画、およびテスト手順書）については、各タスクが先行タスクの結果を処理し、生成された結果が後続タスク（ソフトウェアの運用保守タスクを含む）に対して適切で

あることが確実なものとなるように、それらの文書を作成しレビューする。変更が承認されると、それを反映するように、影響を受けるソフトウェア作業成果物、計画、コミットメント、プロセス、および活動が改訂される。

ゴール

- 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。

実施のコミットメント

- コミットメント1 プロジェクトは、ソフトウェアエンジニアリング活動の実施に関して、明文化された組織方針に従う。

この方針で明記される典型的な事項：

1. ソフトウェアエンジニアリングタスクは、「プロジェクトの定義されたソフトウェアプロセス」に従って実施される。

「プロジェクトの定義されたソフトウェアプロセス」を扱うプラクティスについては、『ソフトウェア統合管理』キープロセスエリアの活動1、2を参照。

2. ソフトウェア成果物の構築と保守には、適切な手法とツールが使用される。
3. ソフトウェア計画、タスク、および成果物は、「ソフトウェアに割り当てられたシステム要件」への追跡が可能である。

(コミットメント1)

これらのプラクティスでは、「ソフトウェアに割り当てられたシステム要件」を「割り当てられた要件」という。

「ソフトウェアに割り当てられたシステム要件」を扱うプラクティスについては、『要件管理』キープロセスエリアを参照。

実施能力

能力1

ソフトウェアエンジニアリングタスクを実施するために、適切な資源と資金が提供される。

1. 以下のような異なるソフトウェアエンジニアリングタスクを実施するために、スキルを持つ人材が利用可能である。
 - ソフトウェア要件分析
 - ソフトウェア設計
 - コーディング
 - テスト
 - ソフトウェア保守
2. ソフトウェアエンジニアリングタスクを支援するツールが利用可能になっている。

一般的な支援ツールの例：

- ワークステーション
- データベース管理システム
- オンラインヘルプ
- グラフィックツール
- 対話型文書化ツール
- ワードプロセッサ

(能力1)

ソフトウェア要件分析用の支援ツールの例：

- 要件の進捗確認ツール
- 仕様記述ツール
- プロトタイピングツール
- モデリングツール
- シミュレーションツール

ソフトウェア設計用の支援ツールの例：

- 仕様記述ツール
- プロトタイピングツール
- シミュレーションツール
- プログラム設計言語

コーディング用の支援ツールの例：

- エディタ
- コンパイラ
- クロスリファレンスジェネレータ
- 整形印刷ツール

ソフトウェアテスト用の支援ツールの例：

- テスト管理ツール
- テストジェネレータ
- テストドライバ
- テストプロファイラ
- シンボリックデバッガ
- テストカバレッジアナライザ

能力2

ソフトウェアエンジニアリングの技術要員メンバは、各自の技術的任務を実施するための必修トレーニングを受ける。

ソフトウェアエンジニアリングの技術要員メンバは、当該アプリケーションドメインのトレーニングを受けるべきである。

ソフトウェア要件分析のトレーニングの例：

- ソフトウェア要件分析の原理
- 保守されるべき全ての既存のソフトウェアに対する既存のソフトウェア要件
- ソフトウェア要件を確立するために、エンドユーザやアプリケーションドメインの専門家にインタビューするスキル（要件獲得）
- ソフトウェア要件分析のためにプロジェクトが選択したツール、手法、慣例、および標準の使用

ソフトウェア設計のトレーニングの例：

- 設計コンセプト
- 保守されるべき全ての既存のソフトウェアに対する既存の設計
- ソフトウェア設計のためにプロジェクトが選択したツール、手法、慣例、および標準の使用

コーディングのトレーニングの例：

- 選択されたプログラミング言語
- 保守されるべき任意のソフトウェアに対する既存ソースコードのレビュー
- プログラミングのためにプロジェクトが選択したツール、手法、慣例、および標準の使用
- ユニットテスト技法

(能力2)

ソフトウェアテストとその他検証技法のトレーニングの例：

- 検証方法（テストだけでなく、分析、実証、およびインスペクションについても）
- テスト計画
- ソフトウェアのテストと検証のためにプロジェクトが選択したツール、方法、慣例、および標準の使用
- テスト開始、終了基準
- テストカバレッジの計測

『トレーニングプログラム』キープロセスエリアを参照。

能力3

ソフトウェアエンジニアリングの技術要員メンバは、関連するソフトウェアエンジニアリング機能についてオリエンテーションを受ける。

関連するソフトウェアエンジニアリング機能の例：

- ソフトウェア要件分析
- ソフトウェア設計
- コーディング
- テスト
- ソフトウェア構成管理
- ソフトウェア品質保証

『トレーニングプログラム』キープロセスエリアを参照。

能力4

プロジェクトマネージャとすべてのソフトウェアマネージャは、ソフトウェアプロジェクトの技術面についてオリエンテーションを受ける。

(能力4)

オリエンテーションの例：

- ソフトウェアエンジニアリングの手法とツール
- 当該アプリケーションドメイン
- 納入するかしないかに依らず、ソフトウェアとそれらに関連する作業成果物
- 所定の手法やツールを用いてプロジェクトをいかに管理するかのガイドライン

『トレーニングプログラム』キープロセスエリアを参照。

実施される活動

活動1

適切なソフトウェアエンジニアリングの手法とツールを、「プロジェクトの定義されたソフトウェアプロセス」に統合する。

「プロジェクトの定義されたソフトウェアプロセス」を扱うプラクティスについては、『ソフトウェア統合管理』キープロセスエリアの活動1、2を参照。

1. 「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、ソフトウェアエンジニアリングタスクを統合する。
2. そのソフトウェアプロジェクトに適した手法とツールを選択する。

(活動1)

組織標準への適用可能性、「プロジェクトの定義されたソフトウェアプロセス」、既存のスキル基盤、トレーニング実施の可能性、契約要件、パワー、使いやすさ、および支援サービスに基づいて、手法やツールの候補を選択する。

- 特定のツールや手法を選択した論理的根拠を文書化する。
3. そのソフトウェアプロジェクトに適した構成管理モデルを選択し使用する。

構成管理モデルの例：

- チェックアウト/チェックインモデル
- 構成モデル
- トランザクションモデル
- 変更集合モデル

4. ソフトウェア成果物を開発し保守するために使用するツールを、構成管理下に置く。

『ソフトウェア構成管理』キープロセスエリアを参照。

活動2

「プロジェクトの定義されたソフトウェアプロセス」に従って割り当てられた要件を系統的に分析することにより、ソフトウェア要件を開発し、保守し、文書化し、そして検証する。

1. ソフトウェア要件の開発に携わる人は、割り当てられた要件をレビューし、ソフトウェア要件分析に影響を与える課題を特定し解決することを確実なものにする。

(活動2)

ソフトウェア要件は、ソフトウェアの機能と性能、ハードウェアとソフトウェアへのインターフェース、およびその他のシステムコンポーネント(例えば、人間)を扱う。

2. 効果的な要件分析手法を利用して、ソフトウェア要件を特定し導出する。

要件分析手法の例：

- 機能型分解
- オブジェクト指向型分解
- トレードオフ調査
- シミュレーション
- モデリング
- プロトタイピング
- シナリオ作成

3. 要件分析結果と代替案選択の論理的根拠を文書化する。
4. ソフトウェア要件を分析して、ソフトウェアでの実装が可能かつ適切であること、明確に表現されていること、相互に首尾一貫していること、テスト可能であること、および(集合として捉えた場合に)完全であることを確実なものにする。
 - ソフトウェア要件の問題が特定され、システム要件に責任を持つグループによってレビューされる。そして割り当てられた要件やソフトウェア要件に対して、適切な変更を加える。

『要件管理』キープロセスエリアを参照。

5. ソフトウェア要件を文書化する。

6. ソフトウェアのシステムテストと検収テストに責任を持つグループは、各ソフトウェア要件を分析し、テスト可能であるか検証する。

(活動2)

7. 各ソフトウェア要件が満足されているか検証し、妥当性を確認するための手法を特定し文書化する。

検証および妥当性確認手法の例：

- デモンストレーション
- システムテスト
- 検収テスト
- 分析
- インスペクション

8. ソフトウェア要件文書は、完了と見なされる前にピアレビューを受ける。

『ピアレビュー』キープロセスエリアを参照。

9. ソフトウェア要件文書は、レビューされ承認される。

ソフトウェア要件文書をレビューし承認する人の例：

- プロジェクトマネージャ
- システムエンジニアリングマネージャ
- プロジェクトソフトウェアマネージャ
- ソフトウェアテストマネージャ

10. ソフトウェア要件文書を顧客やエンドユーザなどが適宜レビューする。

これらのプラクティスで意味するエンドユーザとは、顧客が指名するエンドユーザ、またはエンドユーザの代表をいう。

(活動2)

11. ソフトウェア要件文書を構成管理下に置く。

『ソフトウェア構成管理』キープロセスエリアを参照。

12. 割り当てられた要件に変更があるときは、常にソフトウェア要件を適切に変更する。

『要件管理』キープロセスエリアを参照。

活動3

ソフトウェア要件に適応し、コーディングに対する枠組みを形成するため、「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、ソフトウェア設計を開発し、保守し、文書化し、そして検証する。

ソフトウェア設計は、ソフトウェアアーキテクチャと詳細なソフトウェア設計で構成される。

1. 設計基準を策定しレビューする。

設計基準の例：

- 検証可能性
- 設計標準への忠実さ
- 構築の容易さ
- 単純さ
- 計画の容易さ

2. ソフトウェア設計に携わる人は、ソフトウェア要件をレビューし、ソフトウェア設計に影響を及ぼす課題が特定され解決されることを確実なものにする。

(活動3)

3. 適切な場合、アプリケーション標準を使用する。

アプリケーション標準の例：

- オペレーティングシステムインターフェース標準
- コンピュータヒューマンインターフェース標準
- ネットワーキングインターフェース標準

4. 効果的な手法をソフトウェア設計に使用する。

ソフトウェア設計手法の例：

- プロトタイピング
- 構造化モデル
- 設計の再利用
- オブジェクト指向設計
- 重点システム分析

5. ソフトウェアライフサイクルと使用する技術の制約が許す範囲で、ソフトウェアアーキテクチャを早期に開発する。

ソフトウェアアーキテクチャは、トップレベルのソフトウェアの枠組みを確立するもので、整った形で定義された内部および外部インターフェースを備えている。

6. ソフトウェアアーキテクチャをレビューし、ソフトウェア詳細設計に影響を及ぼすアーキテクチャの課題を特定し解決することを確実なものにする。
7. ソフトウェアアーキテクチャに基づいて、ソフトウェア詳細設計を開発する。
8. ソフトウェア設計（ソフトウェアアーキテクチャと詳細設計）を文書化する。

(活動3)

- ソフトウェア設計書は、ソフトウェアコンポーネント、ソフトウェアコンポーネント間の内部インターフェース、および他のソフトウェアシステムやハードウェアやその他のシステムコンポーネント（例えば、人間）とのソフトウェアインタフェースを扱う。
9. ソフトウェア設計文書は、完了と見なされる前にピアレビューを受ける。

『ピアレビュー』キープロセスエリアを参照。

10. ソフトウェア設計文書を、構成管理下に置く。

『ソフトウェア構成管理』キープロセスエリアを参照。

11. ソフトウェア要件に変更があるときは、常にソフトウェア設計文書を適切に変更する。

活動4

ソフトウェア要件とソフトウェア設計を実装するため、「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、ソフトウェアコードを開発し、保守し、文書化し、そして検証する。

1. コーディングに携わる人は、ソフトウェア要件とソフトウェア設計をレビューし、コーディングに影響を及ぼす課題が特定され解決されることを確実なものにする。
2. 効果的なプログラミング手法をソフトウェアのコーディングに使用する。

プログラミング手法の例：

- 構造化プログラミング
- コードの再利用

(活動4)

3. コードユニットの開発される順序は、重要性、難易度、統合とテストの課題、および顧客やエンドユーザなどのニーズといった要因を適宜考慮するような計画に基づいている。
4. 各コードユニットは、完了と見なされる前にピアレビューを受け、ユニットテストが行われる。

『ピアレビュー』キープロセスエリアを参照。

5. コードを構成管理下に置く。

『ソフトウェア構成管理』キープロセスエリアを参照。

6. ソフトウェア要件またはソフトウェア設計に変更があるときは、常にコードを適切に変更する。

活動5

「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、ソフトウェアテストを実施する。

1. テスト基準を策定し、適宜、顧客やエンドユーザとレビューする。
2. 効果的な手法をソフトウェアのテストに使用する。
3. テストの十分性は以下に基づいて判断する：

実施したテストのレベル

(活動5)

テストのレベルの例：

- ユニットテスト
- 統合テスト
- システムテスト
- 検収テスト

選択したテスト戦略

テスト戦略の例：

- 機能的（ブラックボックス）
- 構造的（ホワイトボックス）
- 統計的

達成すべきテストカバレッジ

テストカバレッジアプローチの例：

- ステートメントカバレッジ
- パスカバレッジ
- 分岐カバレッジ
- 利用プロファイル

4. ソフトウェアテストの各レベルに対して、テスト開始基準を確立し使用する。

(活動5)

テスト開始基準の例：

- ソフトウェアユニットが、コードのピアレビューとユニットテストを完了し合格してから、統合テストに入る。
- ソフトウェアが、統合テストを完了し合格してから、システムテストに入る。
- テスト開始判定レビューを開催してから、ソフトウェアの検収テストに入る。

5. テストされるソフトウェアやその環境が変化したときは、常に回帰テストを各テストレベルで適宜実施する。
6. テスト計画、テスト手順、およびテストケースは、使用可能と見なされる前にピアレビューを受ける。

『ピアレビュー』キープロセスエリアを参照。

7. テスト計画、テスト手順、およびテストケースは、管理され制御されている。

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キープロセスエリアに記述されている。

8. 割り当てられた要件、ソフトウェア要件、ソフトウェア設計、またはテストされるコードに変更があるときは、常にテスト計画、テスト手順、およびテストケースを適切に変更する。

活動6 「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、ソフトウェアの統合テストを計画し実施する。

(活動6)

1. ソフトウェア開発計画に基づき、統合テスト計画を文書化する。
2. 統合テストのテストケースとテスト手順は、ソフトウェア要件、ソフトウェア設計、システムテスト、および検収テストに責任のある人がレビューする。
3. ソフトウェアの統合テストは、指定されたバージョンのソフトウェア要件文書とソフトウェア設計文書に対して実施する。

活動7 ソフトウェアのシステムテストと検収テストを計画し実施して、ソフトウェアが要件を満足していることを実証する。

システムテストは、ソフトウェアがソフトウェア要件を満足していることを確実なものにするために実施する。

検収テストは、ソフトウェアが割り当てられた要件を満足していることを、顧客やエンドユーザに対して実証するために実施する。

1. 適切なテスト準備を行うため、ソフトウェアテストの資源を十分早い時期に割り当てる。

テスト準備に必要な活動の例：

- テスト文書の作成
- テスト資源のスケジュール作成
- テストドライバの開発
- シミュレータの開発

2. システムテストと検収テストはテスト計画の中に文書化する。テスト計画は、顧客やエンドユーザなどと適宜レビューし承認を得る。テスト計画では以下の内容を取り扱う：

(活動7)

- 全体的なテストと検証に対するアプローチ
- 開発組織、外注先、顧客、およびエンドユーザへの責任割り当てなど、適宜
- テスト施設、テスト機器、およびテスト支援要件
- 検収基準

3. テストケースとテスト手順は、ソフトウェア開発者から独立したテストグループが計画し準備する。
4. テストケースは、テストの開始前に文書化し、顧客やエンドユーザなどと適宜レビューし承認を得る。
5. ソフトウェアのテストは、ベースライン化されたソフトウェアと、割り当てられた要件やソフトウェア要件のベースライン化された文書に対して実施する。

6. テストで発見された問題を文書化し、決着まで進捗を確認する。

問題の文書化と進捗の確認を扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キープロセスエリアの活動9と、『ソフトウェア構成管理』キープロセスエリアの活動5を参照。

7. テスト結果を文書化し、ソフトウェアがその要件を満足しているか否かを判定する基盤として用いる。
8. テスト結果は、管理され制御されている。

活動8

「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、ソフトウェアの運用と保守に使用する文書を作成し保守する。

1. 適切な手法とツールを文書作成に使用する。

(活動8)

手法とツールの例：

- ワードプロセッサ
- ケーススタディ
- 文書の再利用

2. 文書化の計画、文書作成、および文書保守には、文書化の専門家が積極的に参加する。
3. 文書の暫定版をソフトウェアライフサイクルの早い段階で作成して、顧客、エンドユーザ、およびソフトウェア保守担当者などと適宜レビューし、フィードバックを提供できるようにする。

文書の例：

- トレーニング文書
- オンライン文書
- ユーザーズマニュアル
- オペレーターズマニュアル
- 保守マニュアル

4. 文書の最終版は、ソフトウェア検収テスト用にベースライン化されたソフトウェアに対して検証される。
5. 文書は、ピアレビューを受ける。

『ピアレビュー』キープロセスエリアを参照。

6. 文書は、管理され制御されている。
7. 文書の最終版は、顧客、エンドユーザ、およびソフトウェア保守担当者などと適宜レビューし承認を得る。

活動9

「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、ピアレビューやテストで特定された欠陥のデータを収集し分析する。

収集し分析するデータの種類の例：

- 欠陥の記述
- 欠陥の分類
- 欠陥の重大度
- 欠陥を含むユニット
- 欠陥で影響を受けるユニット
- 欠陥を混入した活動
- 欠陥を特定したピアレビューやテストケース
- 欠陥の特定に使われたシナリオの記述
- 欠陥を特定した際の期待された結果と実際の結果

活動10

ソフトウェア作業成果物全体をとおして首尾一貫性を保つ。ソフトウェア作業成果物には、ソフトウェア計画、プロセス記述、割り当てられた要件、ソフトウェア要件、ソフトウェア設計、コード、テスト計画、およびテスト手順が含まれる。

1. ソフトウェア作業成果物を文書化し、その文書が容易に利用できるようにする。
2. ソフトウェア要件、設計、コード、およびテストケースについて、それらのインプットとなった情報源と、後続のソフトウェアエンジニアリング活動の成果物との関係を追跡する。
3. ソフトウェア要件、設計、コード、およびテストケースと、割り当てられた要件との関係を追跡する文書は、管理され制御されている。
4. ソフトウェアへの理解が深まるに伴い、ソフトウェア作業成果物、計画、プロセス記述、および活動に対する変更は、適宜、提案され、分析され、そして組み込まれる。

(活動10)

- プロジェクトは、変更による影響を判断してから、変更を実施する。

- 割り当てられた要件の変更が必要なときは、その変更に対する承認を得て、要件の変更を組み込んでから、ソフトウェア作業成果物や活動を変更する。
- すべてのソフトウェア成果物、計画、プロセス記述、および活動に対する変更を調整する。
- 影響を受けるグループに対して、変更の協議と伝達を行う。

影響を受けるグループの例：

- ソフトウェアエンジニアリング
- ソフトウェア見積り
- システムテスト
- ソフトウェア品質保証
- ソフトウェア構成管理
- 契約管理
- 文書化支援

- 変更の完了まで進捗を確認する。

計測と分析

計測1

計測を行い、その結果を使用して、ソフトウェア成果物の機能充足性と品質を判断する。

計測値の例：

- ソフトウェア成果物の中で特定された欠陥の数、タイプ、および重大度で、累積的にかつステージ毎に確認したもの
- 割り当てられた要件で、分類別（セキュリティ、システム構成、性能、および信頼性など）にまとめたものや、ソフトウェア要件やシステムテストケースまで追跡したもの

計測2 **計測を行い、その結果を使用して、ソフトウェアプロダクトエンジニアリングの活動状況を判断する。**

計測値の例：

- プロジェクト全般にわたる、それぞれの割り当てられた要件の状況
- 重大度もしくは未解決期間の長さによって分類した問題報告
- 割り当てられた要件に対して行われた変更活動
- 提案された各変更の分析に費やされた工数やその累計
- ソフトウェアベースラインへ組み込まれた変更件数で、分類別（例えば、インターフェース、セキュリティ、システム構成、性能、および使用性）のもの
- 組み込まれた変更の実装やテストの規模とコストで、初期の見積り値と実績値を含むもの

履行検証

検証1 **ソフトウェアプロダクトエンジニアリングの活動は、上級管理層によって定期的にレビューされる。**

上級管理層の監督レビューに関して、その典型的な内容を扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証1を参照。

検証2 **ソフトウェアプロダクトエンジニアリングの活動は、プロジェクトマネージャによって定期的に、かつイベント発生を契機としてレビューされる。**

(検証2)

プロジェクトマネージャの監督レビューに関して、その典型的な内容を扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証2を参照。

検証3

ソフトウェア品質保証グループは、ソフトウェアプロダクトエンジニアリングの活動と作業成果物をレビューかつ/または監査し、その結果を報告する。

『ソフトウェア品質保証』キープロセスエリアを参照。

レビューや監査では、最低限、以下のことを検証する：

1. 次の点を確実なものにするため、ソフトウェア要件がレビューされている：
 - 完全である
 - 正しい
 - 首尾一貫している
 - 実現可能である
 - テスト可能である
2. 各ソフトウェアエンジニアリングタスクに対する開始基準と完了基準が満足されている。
3. ソフトウェア成果物は、明記された標準と要件に準拠している。
4. 必要なテストが実施されている。
5. 文書化された計画と手順に従って、ソフトウェアのシステムテストと検収テストが実施されている。
6. ソフトウェアテスト計画で文書化されているとおりに、テストが検収基準を満足している。

7. テストが満足な結果で完了し、記録されている。

(検証3)

8. 検出された問題や欠陥が、文書化され、進捗が確認され、そして取り上げられている。

9. ソフトウェア要件、設計、コード、およびテストケースにわたって、割り当てられた要件の追跡が行われている。

10. ソフトウェアの運用と保守に使われる文書は、ソフトウェアベースラインならびに所定の割り当てられた要件に照らして検証され、その後ソフトウェア成果物が顧客またはエンドユーザに引き渡されている。

グループ間調整

レベル3のキープロセスエリア： 定義された段階

『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』に必然的に含まれる活動には、ソフトウェアエンジニアリンググループが他のプロジェクトのエンジニアリンググループと連携して、システムレベルの要件、ねらい、および課題を取り上げることが挙げられる。プロジェクトの中のエンジニアリンググループの代表者は、顧客やエンドユーザなどと適宜作業を行い、システムレベルの要件、ねらい、および計画を確立するために連携する。これらの要件、ねらい、および計画は、すべてのエンジニアリング活動の基盤となる。

グループ間における技術活動のインターフェースとやり取りは、システム全体の品質と一貫性を確実なものにするため、計画され管理される。すべてのエンジニアリンググループがすべてのグループの状況や計画を認識し、そしてシステムとグループ間の課題が適切に注目されることを確実なものとするため、プロジェクトのエンジニアリンググループの代表者間で定期的に技術レビューと意見交換会が開催される。

これらのエンジニアリングタスクに関連するプラクティスでソフトウェア特有のものは、『要件管理』および『ソフトウェアプロダクトエンジニアリング』キープロセスエリアに記述されている。

ゴール

- 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。

実施のコミットメント

- コミットメント1 プロジェクトは、各エンジニアリングの代表者チームの確立に関して、明文化された組織方針に従う。

この方針で明記される典型的な事項：

1. プロジェクトにおけるシステム要件およびプロジェクトレベルのねらいは、定義され、すべての影響を受けるグループによりレビューされる。

影響を受けるグループの例：

- ソフトウェアエンジニアリング
- ソフトウェア見積もり
- システムテスト
- ソフトウェア品質保証
- ソフトウェア構成管理
- 契約管理
- 文書化支援

2. エンジニアリンググループは、各自の計画と活動を調整する。

- (コミットメント1) 3. プロジェクトのエンジニアリンググループ間、顧客もしくはエンドユーザ（適宜）とプロジェクトとの間、および組織全体に渡って、マネージャは、交流、調整、支援、およびチームワークを促進する環境の確立と維持について責任を負う。

これらのプラクティスで意味するエンドユーザとは、顧客の指定するエンドユーザ、またはエンドユーザの代表者をいう。

実施能力

能力1 ソフトウェアエンジニアリング活動を他のエンジニアリンググループと調整するために、適切な資源と資金が提供される。

能力2 異なったエンジニアリンググループで用いられる各支援ツールは整合しており効果的な伝達や調整が可能である。

整合すべき支援ツールの例：

- ワードプロセッサ
- データベースシステム
- グラフィックツール
- 表計算プログラム
- 問題の進捗確認パッケージ
- ライブラリ管理ツール

能力3 組織のマネージャー全員が、チームワークについての必修トレーニングを受ける。

(能力3)

トレーニングの例：

- チーム作り
- チーム管理
- チームワークの確立、推進、および促進
- グループ力学

『トレーニングプログラム』キープロセスエリアを参照。

能力4

各エンジニアリンググループのタスクリーダー全員が、他のエンジニアリンググループで使用されるプロセス、手法、および標準についてオリエンテーションを受ける。

『トレーニングプログラム』キープロセスエリアを参照。

能力5

エンジニアリンググループのメンバは、チームとして作業することについてオリエンテーションを受ける。

『トレーニングプログラム』キープロセスエリアを参照。

実施される活動

活動1

ソフトウェアエンジニアリンググループとその他のエンジニアリンググループは、顧客やエンドユーザなどと適宜連携し、システム要件を確立する。

(活動1) これらのグループの具体的な活動：

1. 顧客やエンドユーザの要件の中で重要な特徴を適宜定義する。
2. 重要な依存関係を協議する。
3. 顧客もしくはエンドユーザに適宜納入される成果物毎に、検収基準を文書化する。

活動2

プロジェクトのソフトウェアエンジニアリンググループの代表者は、他のエンジニアリンググループの代表者ととともに、技術活動をモニターしかつ調整し、そして技術的な課題を解決する。

1. 各グループの代表者は、以下により技術活動をモニターし調整する：
 - 仕様を調整すること、およびシステム要件とシステム設計の技術レビューならびに承認を提供すること

システム要件とシステム設計は、典型的にはシステムエンジニアリンググループの責任である。しかし、他のエンジニアリンググループの代表者もそれらの活動に深く関与することが期待される。

システム要件とシステム設計の内容：

- システム要件全般
- システム構成（ハードウェア、ソフトウェア、およびその他のシステムコンポーネント）
- システムコンポーネントへの要件割り当てと追跡
- システムコンポーネント間のインターフェースの定義

- プロジェクトのライフサイクル全般にわたったシステム要件およびプロジェクトレベルのねらいの変更を管理し制御するため、プロジェクトレベルでの技術レビューと分析を提供すること

(活動2)

- ハードウェア、ソフトウェア、およびその他のシステムコンポーネントに関する設計開発活動の進捗を確認しレビューすること
- 複数のエンジニアリンググループを巻き込む技術リスクをアセスメントし、勧告を作成し、そして進捗を確認すること

リスク管理を扱うプラクティスについては、『ソフトウェア統合管理』キープロセスエリアの活動10を参照。

2. 各グループの代表者は、以下により技術課題を取り扱う：

- プロジェクトレベルの対立を解決すること、およびシステム要件や設計に関する課題を明確にすること
- 問題解決の合同勧告を作成すること
- プロジェクトのエンジニアリンググループ全体におよぶプロセス課題に取り組むこと

活動3

グループ間のコミットメントの伝達、および実施される作業の調整と進捗の確認のため、文書化された計画を使用する。

この計画は：

1. 以下のベースラインになる：

- プロジェクトのスケジュール
- プロジェクトの契約面と技術面
- エンジニアリンググループへの責任の割り当て

2. 異なったエンジニアリンググループ間の活動調整に用いられる。

3. すべてのエンジニアリンググループのメンバが容易に利用可能である。

4. すべてのグループ間コミットメント、およびこれらのコミットメントの変更を取り入れるために更新される。

5. 作業が進むにつれ、プロジェクトレベルでの進捗と計画変更を反映するために、特に、プロジェクトの主要なマイルストーンの完了時および大幅な計画変更時に更新される。

(活動3)

6. すべてのエンジニアリンググループとプロジェクトマネージャによりレビューされ合意される。

活動4

エンジニアリンググループ間の重要な依存関係は、文書化された手順に従って特定し、協議し、そして進捗を確認する。

重要な依存関係の管理を扱うプラクティスについては、『ソフトウェア統合管理』キープロセスエリアの活動9を参照。

この手順で明記される典型的な事項：

1. それぞれの重要な依存関係について、以下を含めて明示的に定義する：
 - 提供されるべきアイテム
 - 誰が提供するか
 - いつ提供されるか
 - 検収基準
2. 重要な依存関係については、プロジェクトおよび組織内部のソフトウェアエンジニアリンググループと他のエンジニアリンググループとの間で協議される。
3. 重要な依存関係アイテムの必要期日および利用可能期日については、プロジェクトスケジュールおよびソフトウェアスケジュールと結びつけられる。
4. 重要な依存関係アイテムの受け入れ側グループと提供側の責任グループの両者によって、重要な依存関係毎に合意事項が文書化され、レビューされ、そして承認される。
5. 重要な依存関係については、定期的に進捗が確認され、適宜是正処置が取られる。

- 状況、ならびに実際の完了あるいは完了予想は、グループ間コミットメント調整に用いる計画と比較される。
- 完了の遅れならびに早期終了は、将来の活動とマイルストーンに影響が無いか評価される。
- 起こった問題と潜在的問題は、適切なマネージャーに報告される。

活動5 **他のエンジニアリンググループへのインプットとなる作業成果物は、その作業成果物が受け入れ側のエンジニアリンググループのニーズを満足することを確実なものにするため、受け入れ側グループの代表者がレビューする。**

活動6 **個々のプロジェクトエンジニアリンググループの代表者で解決できないグループ間の課題は、文書化された手順に従って取り扱う。**

グループ間課題の例：

- スケジュールの不整合
- 不十分な資金
- 技術的なリスク
- システムレベルの設計や要件の欠陥
- システムレベルの問題

活動7 **プロジェクトのエンジニアリンググループの代表者は、定期的に技術レビューと意見交換会を開催する。**

これらの会議で参加者は：

1. 顧客やエンドユーザなどのニーズおよび要望について、適宜、可視性を提供する。
2. プロジェクトの技術活動をモニターする。
3. 技術要件に関する各グループの解釈と実装が、システム要件に準拠していることを確実なものにする。

4. コミットメントをレビューして、それらが満足されているかを判断する。

レビューを扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キープロセスエリアを参照。

5. 技術的なリスクと他の技術課題をレビューする。

(活動7)

リスク管理を扱うプラクティスについては、『ソフトウェア統合管理』キープロセスエリアの活動10を参照。

計測と分析

計測1

計測を行い、その結果を使用して、グループ間調整の活動状況を判断する。

計測値の例：

- ソフトウェアエンジニアリンググループが他のエンジニアリンググループを支援するために使用した実際の工数とその他の資源
- 他のエンジニアリンググループがソフトウェアエンジニアリンググループを支援するために使用した実際の工数とその他の資源
- ソフトウェアエンジニアリンググループが他のエンジニアリンググループの活動を支援するために実施した特定のタスクやマイルストーンの完了実績
- 他のエンジニアリンググループがソフトウェアエンジニアリンググループの活動を支援するために実施した特定のタスクやマイルストーンの完了実績

履行検証

検証1 **グループ間調整の活動は、上級管理層によって定期的にレビューされる。**

上級管理層の監督レビューに関して、その典型的な内容を扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証1を参照。

検証2 **グループ間調整の活動は、プロジェクトマネージャによって定期的に、かつイベント発生を契機としてレビューされる。**

プロジェクトマネージャの監督レビューに関して、その典型的な内容を扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証2を参照。

検証3 **ソフトウェア品質保証グループは、グループ間調整の活動と作業成果物をレビューかつ/または監査し、その結果を報告する。**

『ソフトウェア品質保証』キープロセスエリアを参照。

このキープロセスエリアでのソフトウェア品質保証の責任は、すべてのプロジェクトエンジニアリンググループを取り扱う品質保証機能に包括されることがある。

レビューや監査では、最低限、以下のことを検証する：

1. プロジェクトエンジニアリンググループ間の重要な依存関係の特定、協議、および進捗確認のための手順。

2. グループ間の課題の処理。

ピアレビュー

レベル3のキープロセスエリア： 定義された段階

『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効果がある。

『ピアレビュー』に必然的に含まれる活動には、欠陥や変更が必要な領域を特定するため、ソフトウェア作業成果物を作成者の同僚が手法に則って検査することが挙げられる。ピアレビューを受ける特定のソフトウェア成果物は、『ソフトウェア統合管理』キープロセスエリアに記述されているように、「プロジェクトの定義されたソフトウェアプロセス」の中で明らかにされ、ソフトウェアプロジェクト計画活動の中でスケジュールが設定される。

このキープロセスエリアでは、ピアレビューの実施に関するプラクティスを取り上げる。ピアレビューを受ける特定のソフトウェア作業成果物を明らかにするプラクティスは、各ソフトウェア作業成果物の開発と保守を記述するキープロセスエリアに含まれる。

ゴール

- 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。

実施のコミットメント

コミットメント1 プロジェクトは、ピアレビューの実施に関して、明文化された組織方針に従う。

この方針で明記される典型的な事項：

1. 組織が、ピアレビューを受けるソフトウェア作業成果物の標準セットを特定する。
2. プロジェクトが、ピアレビューを受けるソフトウェア作業成果物を特定する。

ピアレビューを受けるソフトウェア作業成果物の特定を扱うプラクティスについては、『ソフトウェア統合管理』キープロセスエリアの活動1、および『組織プロセス定義』キープロセスエリアの活動2を参照。

ピアレビューを受けるソフトウェア作業成果物の例：

- 実稼動ソフトウェアと支援ソフトウェア
- 納入されるソフトウェア作業成果物と納入されないソフトウェア作業成果物
- ソフトウェア（ソースコードなど）およびソフトウェア以外の作業成果物（文書など）
- プロセス記述

3. ピアレビューは、トレーニングを受けたピアレビューリーダーによって主導される。
4. ピアレビューは、作成者ではなくレビュー対象のソフトウェア作業成果物に焦点を当てる。
5. ピアレビューの結果は、マネージメントによる個人の実績評価に用いない。

実施能力

能力1

レビューされる各ソフトウェア作業成果物をピアレビューするために、適切な資源と資金が提供される。

資源と資金の提供の対象となる活動：

1. ピアレビュー資料を準備し配布する。
2. ピアレビューを主導する。
3. 資料をレビューする。
4. ピアレビューに参加する。ピアレビューで特定された欠陥に基づいて要求されるフォローアップレビューに参加する。
5. ピアレビューで特定した欠陥に基づくソフトウェア作業成果物の手戻りをモニターする。
6. ピアレビューの結果として得られたデータを収集し報告する。

能力2

ピアレビューのリーダーは、ピアレビューの主導方法についての必修なトレーニングを受ける。

(能力2)

トレーニングの例：

- ピアレビューのねらい、原理、および手法
- ピアレビューの計画と開催準備
- ピアレビューの開始および完了を判定する基準の評価
- ピアレビューの開催と促進
- ピアレビュー結果の報告
- ピアレビューで特定された処置に取り組むための手戻り作業についての進捗の確認と結果の確認
- ピアレビューに必要なデータの収集と報告

『トレーニングプログラム』キープロセスエリアを参照。

能力3

ピアレビューに参加するレビューは、ピアレビューのねらい、原理、および手法についての必修トレーニングを受ける。

トレーニングの例：

- ピアレビューのタイプ（例えば、ソフトウェア要件、ソフトウェア設計、コード、およびソフトウェアテスト手順のレビュー）
- ピアレビューのねらい、原理、および手法
- レビューの役割
- ピアレビューの準備および参加に必要な工数の見積り

『トレーニングプログラム』キープロセスエリアを参照。

実施される活動

活動1

ピアレビューを計画し、計画を文書化する。

計画の内容：

1. ピアレビューを受けるソフトウェア作業成果物を特定する。
 - 選択されるソフトウェア作業成果物には、「組織の標準ソフトウェアプロセス」で特定されたセットが含まれる。

「組織の標準ソフトウェアプロセス」を扱うプラクティスについては、『組織プロセス定義』キーププロセスエリアの活動2を参照。

2. ピアレビューのスケジュールを明記する。

近い将来に実施予定のピアレビューに対しては、トレーニングを受けたピアレビューリーダーと他のレビューが特定されている。

活動2

文書化された手順に従って、ピアレビューを実施する。

この手順で明記される典型的な事項：

1. ピアレビューは、トレーニングを受けたピアレビューリーダーが計画し主導する。
2. ピアレビューのための十分な準備が整えられるように、レビュー資料を前もってレビューに配布する。

(活動2)

レビュー資料には、ピアレビューを受けるソフトウェア作業成果物の開発に関係があるインプットが含まれる。

関係があるインプットの例：

- ソフトウェア作業成果物のねらい
- 適用される標準
- 設計モジュールに関連する要件
- コードモジュールに関連する詳細設計

3. ピアレビューにおける役割を各レビューアに割り振る。
4. ピアレビューの開始および完了の判定基準を明記し徹底する。
 - この基準の達成に関する課題は、適切なマネージャに報告される。
5. チェックリストを使用し、ソフトウェア作業成果物のレビューに関する基準を首尾一貫した方法で特定する。
 - 作業成果物とピアレビューの特定のタイプに応じ、チェックリストをテーラリングする。

チェックリストのテーラリングの対象となる項目の例：

- 標準と手順の遵守性
- 完全性
- 正確性
- コンストラクション規則
- 保守性

- チェックリスト作成者の同僚と潜在的ユーザが、チェックリストをレビューする。
6. ピアレビューで特定された処置について、それが解決されるまで進捗を確認する。

- (活動2) 7. ピアレビューの完了は、ピアレビューで特定された項目に取り組むための手戻りも含め、関連タスクの完了基準に使用する。

活動3 **ピアレビューの開催と結果に関するデータを記録する。**

データの例：

- レビューされるソフトウェア作業成果物の特定
- ソフトウェア作業成果物の規模
- レビューチームの人数とメンバー構成
- レビューあたりの準備時間
- レビュー会議の所要時間
- 検出され修正された欠陥のタイプと件数
- 手戻り工数

計測と分析

- 計測1** **計測を行い、その結果を使用して、ピアレビューの活動状況を判断する。**

計測値の例：

- 実施されたピアレビューの数で、計画と比較したもの
- ピアレビューに用いられた総工数で、計画と比較したもの
- レビューされた作業成果物の数で、計画と比較したもの

履行検証

検証1

ソフトウェア品質保証グループは、ピアレビューの活動と作業成果物をレビューかつ/または監査し、その結果を報告する。

『ソフトウェア品質保証』キープロセスエリアを参照。

レビューや監査では、最低限、以下のことを検証する：

1. 計画されたピアレビューが開催されている。
2. ピアレビューリーダーは、その役割について十分なトレーニングを受けている。
3. レビュアは、各自の役割について適切なトレーニングを受けているか、経験を積んでいる。
4. ピアレビューの準備、開催、およびフォローアップ活動の実施について、プロセスが守られている。
5. ピアレビューのデータに関する報告は、完全、正確、かつタイムリーである。

レベル3 索引

(英和索引と日本語索引は巻末にある。)

- assessment
 - software process 6
- checklist for evaluating software work products 95
- Contingency planning 53
- critical computer resources
 - basis for estimates 49
 - management of 48
 - provision for reserve capacity 49
- Critical dependencies 87
- customer
 - review
 - software test plan 72
 - software testing criteria 70
 - waivers from contractual software requirements 41
- defects
 - use of data on 74
- defined software process
 - coordination at organization level 7
 - development of software design 67
 - development of software requirements 64
 - use in development of software code 69
- documented procedure
 - developing project's defined software process 40
 - development of organization's standard software process 15
 - identification of critical dependencies 87
 - intergroup issues 88
 - library of software process-related documentation 20
 - management of critical computer resources 48
 - management of software costs 47
 - management of software schedule 50
 - organization's training plan 30
 - peer reviews 95
 - project's software development plan 43
 - review of organization's standard software process and related process assets 21
 - revision of project's defined software process 42
 - risk management 51
 - software process database 19
 - software size management 46
 - use of organization's software process database 44
- engineering groups
 - dependencies between 87
- Integrated Software Management 37-56
 - resources and funding for 39
 - review
 - project manager 56
 - senior management 55
 - training for 39-40
- intergroup commitments
 - plan for 86
- Intergroup Coordination 81-90
 - measurements for 89
 - orientation in 84
 - policy for 82

- resources and funding for 83
- review
 - project manager 90
 - senior management 89
- tools for 83
- intergroup issues
 - procedure for 88
- intergroup team
 - conduct of technical reviews 88
 - responsibilities of 85
- library of software process-related documentation
 - items stored in 43
 - procedure for 20
 - use in tailoring organization's standard software process 41
- manage and control
 - documentation tracing allocated requirements 75
 - organization's standard software process 16
 - organization's training plan 31
 - project's defined software process 41
 - software life cycle 18
 - software process database 19
 - software risk management plan 53
 - training materials 33
- Measurements
 - integrated software management 55
 - intergroup coordination 89
 - organization process definition 22
 - organization process focus 9
 - peer reviews 96
 - software engineering activities 77
 - software product engineering 76
 - training program 34-35
- milestones
 - identification of 50
- new technology
 - transfer within organization 8
- Organization Process Definition 11-23
 - measurements for 22
 - resources and funding for 14
 - review
 - software quality assurance 22
 - tools for 14
 - training for 14
- Organization Process Focus 1-10
 - coordination of training for 8
 - involvement of key groups 8
 - management sponsorship of 2
 - measurements for 9
 - orientation in 5
 - plan for 6
 - policy for 2
 - resources and funding for 3
 - review
 - senior management 9
 - senior management oversight of 3
 - tools for 4
 - training for 5
- organization's software process database
 - examples of data in 45
- organization's standard software process
 - coordination at organization level 7
 - deviations from 38
 - guidelines for tailoring 18

- management and control of 16
- peer review of 16
- procedure for developing 15
- review of 21
- tailoring of 40
- tailoring of for project 38
- waivers for deviations from 41
- organization's standard software process and related process assets
 - use in project planning and management 38
- orientation
 - intergroup coordination 84
 - organization process focus 5
 - software product engineering for managers 63
 - software product engineering for technical staff 61
 - training program 28
- Peer Review
 - checklist for evaluating work products 96
 - organization's standard software process 16
 - plan for organization process focus 7
 - software code 69
 - software design document 68
 - software requirements document 66
 - software risk management plan 53
 - software test plan 71
- Peer Reviews 91-97
 - measurements for 96
 - plans for 94
 - policy for 92
- procedure for 95
- resources and funding for 92
- review
 - software quality assurance 97
- training to lead 93
- training to participate in 94
- plan
 - communication of intergroup commitments 86
 - organization process focus 6
 - Peer reviews 94
- plan for organization process focus
 - peer review of 7
- policy
 - intergroup coordination 82
 - organization process focus 2
 - peer reviews 92
 - project planning and management 38
 - software product engineering 58
 - standard software process and related process assets 12
 - training 26
- process assets
 - examples of 11
 - policy for 12
 - review of 21
- process element
 - definition of 17
- project manager
 - review
 - integrated software management 56
 - intergroup coordination 90
 - software product engineering 77
- project's defined software process

- documentation of 38
- guidelines for development 40
- integration of software engineering activities 63
- management and control of 41
- revision of 42
- software engineering tasks and 58
- use for software testing 70
- use in developing system documentation 73
- use in managing software project 43
- use of defect data 74
- project's software development plan development and revision of 43
- resources and funding
 - integrated software management 39
 - intergroup coordination 83
 - organization process definition 14
 - organization process focus 3
 - peer reviews 92
 - software product engineering 59
 - training program 27
- review
 - integrated software management
 - project manager 56
 - senior management 55
 - integrated software management
 - software quality assurance 56
 - intergroup coordination
 - project manager 90
 - senior management 89
 - organization process definition
 - software quality assurance 22
 - organization process focus
 - senior management 9
- peer reviews
 - software quality assurance 97
- software product engineering
 - project manager 77
 - senior management 77
 - software quality assurance 78
- training materials 33
- training program
 - senior management 35
- work products 88
- reviews
 - technical aspects of intergroup coordination 88
- risks
 - management of 51
- Senior management
 - approval of tailoring of organization's standard software process 41
 - oversight of organization process focus activities 3
 - review
 - deviations from organization's standard software process 38
 - integrated software management 55
 - intergroup coordination 89
 - organization process focus 9
 - software product engineering 77
 - training program 35
 - waivers from contractual software requirements 41
 - role in software process focus 3
 - sponsorship of organization process focus 2

- software architecture
 - development of 68
- software code
 - development of 69
 - peer review of 69
 - software configuration management of 70
- software configuration management
 - models for 64
 - software code 70
 - software design document 69
 - software documentation 74
 - software requirements document 66
 - tools for developing and maintaining software 64
- software costs
 - management of 47
- software design
 - definition of 67
 - requirements for 67
- software design document
 - peer review of 68
 - software configuration management of 69
- software development plan
 - relation to project's defined software process 37
- software documentation
 - software configuration management of 74
- software engineering group
 - role in establishing system requirements 84
- software engineering process group
 - control of library of software process-related documentation 20
 - responsibility for organization's standard software process 14
 - responsibility of 1
 - review of tailoring of organization's standard software process 41
- software engineering process group (SEPG)
 - establishment of 4
- software life cycle
 - documentation of 18
 - management and control of 18
 - selection of 40
- Software managers
 - orientation for training program 28
- software process
 - assessment of 6
- software process architecture
 - definition of 18
- Software Process Assessment (SPA) 6
- software process database
 - coordination at organization level 7
 - management and control of 19
 - procedure for establishing 19
 - use for planning and estimating 44
- software process focus
 - coordination of 7
- Software Product Engineering 57-79
 - measurements for 76
 - orientation for managers 63
 - orientation for technical staff 61
 - policy for 58
 - resources and funding for 59

- review
 - project manager 77
 - senior management 77
 - software quality assurance 78
- tools for 59
- training in 61
- software project
 - management of 43
 - review of 54
- software project planning and management
 - policy for 38
- software quality assurance group
 - review
 - integrated software management 56
 - organization process definition 22
 - peer reviews 97
 - software product engineering 78
- software requirements
 - changes to 67
 - development of 64
- software requirements document
 - peer review of 66
 - software configuration management of 66
- software risk
 - plan for managing 52
- software risk management plan
 - management and control of 53
 - peer review of 53
- software schedule
 - management of 50
- software size
 - management of 46
- software test plan
 - peer review of 71
 - requirements for 72
- Software testing 70
- software testing criteria
 - review of 70
- software work products
 - consistency of 75
 - evaluation of 95
- standard software process 3
 - policy for 12
- standards
 - for training courses 32
- System and acceptance testing 72
- system documentation
 - development of 73
- system requirements
 - coordination to establish 84
- system requirements allocated to software
 - management and control of documentation for 75
- testing
 - of software 70
 - of system for acceptance 72
- Tools
 - building and maintaining software 58
 - intergroup coordination 83
 - organization process definition 14
 - organization process focus 4
 - selection of for project 63
 - software product engineering 59
 - training 28
- tools for developing and maintaining

- software
 - software configuration management of 64
- training 40
 - coordination across organization 8
 - development of courses for 32
 - facilities for 28
 - integrated software management 39
 - intergroup coordination 83
 - leading peer reviews 93
 - organization process definition 14
 - organization process focus 5
 - participation in peer reviews 94
 - plan for project 29, 30
 - policy for 26
 - records of 34
 - review of materials for 33
 - software product engineering 61
 - tools for 28
 - waiver for 33
- training materials
 - management and control of 33
- training plan
 - items included in 31
 - management and control of 31
- Training Program 25-36
 - independent evaluation of 35
 - items included in 27
 - measurements for 34
 - orientation in 28
 - resources and funding for 27
 - review
 - senior management 35
 - review or audit of 35
- training staff
 - skills and knowledge of 28
- training vehicles
 - examples of 26
- Work products
 - review of 88

定量的プロセス管理

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』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。

『定量的プロセス管理』に必然的に含まれる活動には、『ソフトウェア統合管理』キープロセスエリアで述べられている「プロジェクトの定義されたソフトウェアプロセス」の実績に対する目標を確立すること、プロセス実績の計測を行なうこと、それらの計測値を分析すること、および許容限界内でプロセス実績を維持するための補正を行うことが挙げられる。プロセス実績が許容限界内で安定化した時点で、「プロジェクトの定義されたソフトウェアプロセス」、関連する計測値、およびその計測値の許容限界は、ベースラインとして確立され、定量的にプロセス実績を制御するために用いられる。

組織は、ソフトウェアプロジェクトからプロセス実績データを収集し、「組織の標準ソフトウェアプロセス」のプロセス能力（新しいプロジェクトが達成すると期待できるプロセス実績）の特徴を示すためにこれらのデータを用いる。「組織の標準ソフトウェアプロセス」は、『組織プロセス定義』キープロセスエリアで記述されている。プロセス能力は、ソフトウェアプロセスに従った際に期待される結果（組織が着手する次のソフトウェアプロジェクトで期待されるもっともあり得そうな成果）の範囲を記述する。次に、ソフトウェアプロジェクトは、これらのプロセス能力データを用いてプロセス実績に対する目標を確立し改訂し、そして「プロジェクトの定義されたソフトウェアプロセス」の実績を分析する。

ゴール

- 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。

実施のコミットメント

- コミットメント1 プロジェクトは、「プロジェクトの定義されたソフトウェアプロセス」についての実績の計測と定量的な制御に関して、明文化された組織方針に従う。

この方針で明記される典型的な事項：

1. 各プロジェクトは、「プロジェクトの定義されたソフトウェアプロセス」を定量的制御の下に置くために、文書化された計画を履行する。

(コミットメント1)

『定量的制御』という用語は、全ての適切な定量的または統計的技法を意味する。定量的制御の目的は、ソフトウェアプロセスを分析し、ソフトウェアプロセス実績における変動の特殊原因を特定し、そして整った形で定義された限度内にソフトウェアプロセスの実績を収めることである。

変動の特殊原因とは、ある一時的な状況（例えば、特定の局所的条件、ある一台の機械、ある一個人、あるいはある小さなグループが予期しない方法をとっている）であり、プロセス実績に予期しない一時的な変化を引き起こす。

2. 個人の実績に関わる機密データは保護され、これらのデータへのアクセスは適切に制御される。

計測データの個人評価への利用は、報告される計測データの正しさと有用性に負の影響を与える。

コミットメント2 組織は、「組織の標準ソフトウェアプロセス」のプロセス能力分析に関して、明文化された方針に従う。

この方針で明記される典型的な事項：

1. プロジェクトのプロセス実績の計測値は、「組織の標準ソフトウェアプロセス」に対するプロセス能力ベースラインを確立し維持するために分析される。

プロセス能力ベースラインは以下のものを含む：

- 「組織の標準ソフトウェアプロセス」の記述
- 計測の標準定義
- 計測値の期待される範囲

- (コミットメント2) 2. 「組織の標準ソフトウェアプロセス」に対するプロセス能力ベースラインは、ソフトウェアプロジェクトがプロセス実績目標を確立するために使用される。

実施能力

能力1

組織の定量的プロセス管理活動の調整に責任を持つグループが存在する。

グループは、一連のタスクまたは活動に責任を持つ部門、マネージャ、および個人の集まりである。グループは、専任でない個人の場合、異なる部門から割り当てられた専任でない人達の場合、および専任の人達の場合など、さまざまな形態がある。グループ編成にあたっては、割り当てるタスクや活動、プロジェクト規模、組織構造、および組織の文化などを考慮する必要がある。品質保証グループのようにプロジェクトの活動に焦点を当てるグループもあれば、ソフトウェアエンジニアリングプロセスグループのように組織全体の活動に焦点を当てるグループもある。

1. このグループは、組織のソフトウェアプロセス活動に責任のあるグループ（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）の一部であるか、その活動がグループ間で密に調整されているかのどちらかである。

能力2

定量的プロセス管理活動のために適切な資源と資金が提供される。

1. ソフトウェアエンジニアリンググループと他のソフトウェア関連グループのマネージャおよびタスクリーダーは、プロジェクトの定量的プロセス管理活動を実施する。

(能力2)

ソフトウェア関連グループの例：

- ソフトウェア品質保証
- ソフトウェア構成管理
- 文書化支援

2. 組織全体にわたる計測プログラムが存在する。

組織の計測プログラムには、以下のものを含む：

- 組織全体にわたる計測値の定義
- 組織の計測データの収集
- 組織の計測データの分析
- 組織に対する定量的計測目標

3. 定量的プロセス管理を支援するツールが利用可能になっている。

支援ツールの例：

- ソフトウェアソースコード解析ツール
- 自動テストカバレッジ解析ツール
- データベースシステム
- 定量的分析パッケージ
- 問題の進捗確認パッケージ

能力3

選択されたプロセスおよび成果物の計測のためのデータの収集、記録、および分析に対する支援が存在する。

これらのプラクティスで参照する成果物データは、ソフトウェアプロセスの分析で使用される成果物の計測値である。

能力4 **定量的プロセス管理を履行あるいは支援する個人は、これらの活動を実施するための必修トレーニングを受ける。**

トレーニングの例：

- ソフトウェアプロセスのモデリングと分析
- プロセス管理データの選択、収集、および妥当性確認
- 基本的な定量的手法と分析技法の適用（例えば、見積りモデル、パレート図、および管理図）

『トレーニングプログラム』キープロセスエリアを参照。

能力5 **ソフトウェアエンジニアリンググループと他のソフトウェア関連グループのメンバは、定量的プロセス管理の目標と意義に関するオリエンテーションを受ける。**

『トレーニングプログラム』キープロセスエリアを参照。

実施される活動

活動1 **定量的プロセス管理に対するソフトウェアプロジェクト計画は、文書化された手順に従って策定する。**

この手順で明記される典型的な事項：

1. 定量的プロセス管理計画は、以下のものに基づく：

- 成果物品質、生産性、および成果物開発サイクルタイムに対する組織の戦略的目標
- 組織の計測プログラム

(活動1)

- 「組織の標準ソフトウェアプロセス」
- ソフトウェア成果物の品質、生産性、および成果物開発サイクルタイムに対するプロジェクト目標
- 他の「プロジェクトの定義されたソフトウェアプロセス」で計測された実績
- 「プロジェクトの定義されたソフトウェアプロセス」の記述

2. 計画はピアレビューを受ける。

『ピアレビュー』キープロセスエリアを参照。

3. 計画は、組織のソフトウェアプロセス活動に責任のあるグループによりレビューされる（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）。

4. 計画は、管理され制御されている。

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キープロセスエリアに記述されている。

活動2

ソフトウェアプロジェクトの定量的プロセス管理活動は、プロジェクトの定量的プロセス管理計画に従って実施する。

この計画が扱う事項：

1. 定量的プロセス管理活動の目標とねらい。

(活動2)

2. 計測され分析されるソフトウェアタスクまたは他のソフトウェア活動。
3. 「プロジェクトの定義されたソフトウェアプロセス」を計測する仕組み。

計測する仕組みは、組織の計測プログラム、「組織の標準ソフトウェアプロセス」の記述、および「プロジェクトの定義されたソフトウェアプロセス」の記述に基づく。

4. 実施すべき定量的プロセス管理活動とこれらの活動のスケジュール。

現状の組織およびプロジェクトのニーズに加えて、将来の作業に有用だと考えられる計測があげられる。

5. 定量的プロセス管理活動に責任のあるグループおよび個人。
6. 定量的プロセス管理活動を実施するに必要な資源。これには要員およびツールを含む。
7. 定量的プロセス管理活動を実施する際に従うべき手順。

活動3

実施するデータ収集と定量的分析の戦略は、「プロジェクトの定義されたソフトウェアプロセス」に基づいて決定する。

「プロジェクトの定義されたソフトウェアプロセス」の以下の属性が考慮される：

1. タスク、活動、およびそれらの相互関係。

- (活動3)
2. ソフトウェア作業成果物とそれらの相互関係および「プロジェクトの定義されたソフトウェアプロセス」との関係。
 3. プロセス制御箇所およびデータ収集箇所。

活動4 「プロジェクトの定義されたソフトウェアプロセス」を定量的に制御するために用いられる計測データは、文書化された手順に従って収集する。

この手順で明記される典型的な事項：

1. 収集される計測データは、組織とソフトウェアプロジェクトでの計測の目標とねらいを支援する。
2. 収集する特定の計測データ、その精確な定義、各計測の意図する用途と分析、および収集するプロセス制御箇所が定義される。

計測データの例：

- ソフトウェア規模、コスト、およびスケジュールの見積り / 計画データに対する実データ
- 生産性データ
- ソフトウェア品質計画で定義された品質計測項目
- ピアレビューのカバレッジと効率
- トレーニングの有効性
- テストのカバレッジと効率
- ソフトウェア信頼性尺度
- ソフトウェア要件で発見された欠陥の数と重大度
- ソフトウェアコードで発見された欠陥の数と重大度
- 処置項目の数と決着率

3. 計測値はソフトウェアライフサイクル全体から選ばれる。（例えば、開発ステージと開発終了後のステージとの両方）

(活動4)

4. 計測値はキーとなるソフトウェアプロセス活動と主要なソフトウェア作業成果物の特性を扱う。
5. 「組織の標準ソフトウェアプロセス」に関連する計測データは、ソフトウェアプロジェクト横断的に均一に収集される。
6. 制御される計測値は、可能であればソフトウェア活動の自然な結果に基づくものとする。
7. 計測値は事前に定義された分析活動を支援するために選定される。

ときとして計測は研究指向であり、そのように明確に指定されるべきである。

8. 計測データの妥当性は独立してアセスメントされる。
9. 収集された計測データは、「組織のソフトウェアプロセスデータベース」に適宜格納する。

「組織のソフトウェアプロセスデータベース」を扱うプラクティスについては、『組織のプロセス定義』キープロセスエリアの活動5を参照。

活動5

「プロジェクトの定義されたソフトウェアプロセス」は、文書化された手順に従って分析し、定量的な制御下に置く。

この手順で明記される典型的な事項：

1. 特定のデータ分析活動が事前に定義される。

(活動5)

データ分析活動の記述は、以下のものを扱う：

- 必要なインプットデータ
- 使用されるツール
- 実施されるデータ操作
- 導き出される情報
- 分析を実施する際および分析の結果としてどのような処置をとるかを決定する際に用いられる判断基準

分析技法の例：

- パレート図
- 管理図
- 傾向図
- 散布図

2. 「プロジェクトの定義されたソフトウェアプロセス」を通じたプロセス活動の計測データは、特定され、収集され、そして分析される。
3. 選択された計測値は、それが示すプロセスの特徴を適切に表す。
4. 平均と分散の期待値が各計測値に対して明記される。
5. 各計測値の許容限界が定義され、プロジェクトのプロセス実績ベースラインが確立される。

許容限界を確立する例としては、そのプロセスの平均実績値からのこれまでの偏差を計算するものがある。

6. 各計測の値は、平均および分散の期待値と比較される。

(活動5)

実際のプロセス実績と定義された許容限界の比較の例：

- ソースコード1000行あたりのピアレビュー消費時間を履歴データの分析により決定された上限および下限と比較する。
- ソフトウェア要件（例えば、『せねばならない』の件数）のソースコード行数への展開比率を履歴データの分析により決定された上限および下限と比較する。

7. 実際のプロセス実績が定義された許容限界に収まるように、適宜補正される。
8. 「プロジェクトの定義されたソフトウェアプロセス」が定量的に制御される場合、以下のベースラインが確立される：
 - 計測の定義
 - 実際の計測データ
 - 計測値の許容限界
9. ソフトウェアプロジェクトに対するプロセス実績ベースラインは、管理され制御されている。

活動6

ソフトウェアプロジェクトの定量的プロセス管理活動の結果について文書化した報告書を作成し、配布する。

1. データ分析の結果は、それが他の人に対して報告される前に、データの影響を受ける人達によってレビューされる。
2. ソフトウェアマネージャ、ソフトウェアタスクリーダー、および上級管理層は、それぞれのニーズに応じた定例報告書を受けとる。
3. ソフトウェア品質保証グループは、ニーズに応じた定例報告書を受けとる。

4. プロジェクトマネージャ、上級マネージャ、ソフトウェアマネージャ、およびソフトウェアタスクリーダーは、要求により特別報告書を受けとる。

活動7

「組織の標準ソフトウェアプロセス」に対するプロセス能力ベースラインは、文書化された手順に従って確立され維持される。

この手順で明記される典型的な事項：

1. プロジェクトのソフトウェアプロセスデータは、そのプロセス実績ベースラインとして要約され、「組織のソフトウェアプロセスデータベース」に記録される。

「組織のソフトウェアプロセスデータベース」を扱うプラクティスについては、『組織プロセス定義』キープロセスエリアの活動5を参照。

2. 各「プロジェクトの定義されたソフトウェアプロセス」に対するプロセス実績ベースラインは、「組織の標準ソフトウェアプロセス」に対するプロセス能力ベースラインに、適宜、取り入れられる。
3. 「組織の標準ソフトウェアプロセス」に対するプロセス能力ベースラインは、文書化される。
4. 「組織の標準ソフトウェアプロセス」についてのプロセス能力の傾向を調べて、可能性のある問題または改善の機会を予測する。

能力の傾向の使用例：

- ソフトウェア欠陥の発生を予測し、予測と実際を比較する。
- ピアレビューやテストから得たデータに基づき、成果物に残っている欠陥の分布と特徴を予測する。

(活動7)

欠陥の元となりそうな領域の例：

- 見積りと計画のアイテム
- 要件分析のような、ソフトウェアライフサイクルの初期段階で実施される活動
- 主要な文書アイテム
- 過去に欠陥を作り込みがちだったアイテムと活動
- 変更の実装と欠陥修正のための活動
- 労働集約的な活動

改善の機会がありそうな領域の例：

- 他のプロジェクトおよび組織が自動化に成功している活動
- トレーニングやツールのような、非納入物および支援のためのアイテムと活動
- ピアレビューやテストのような、品質指向の活動
- 労働集約的な活動

5. 「組織の標準ソフトウェアプロセス」に対するプロセス能力ベースラインが、管理され制御されている。
6. 過去のプロジェクトと大きく異なるソフトウェアプロジェクトに着手する場合、「組織の標準ソフトウェアプロセス」のテーラリングの一部として、そのプロジェクトに対して新しいプロセス実績ベースラインが確立される。

「組織の標準ソフトウェアプロセス」のプロジェクト用テーラリングを扱うプラクティスについては、『統合ソフトウェア管理』キープロセスエリアの活動1を参照。

(活動7)

大きな差異の例：

- 新しいアプリケーションドメイン
- 根本的に異なる技術の使用
- アプリケーション規模の大幅な変化

7. 「組織の標準ソフトウェアプロセス」への変更は、そのプロセス能力ベースラインへの影響をアセスメントするために、進捗が確認され分析される。

計測と分析

計測1

計測を行い、その結果を使用して、定量的プロセス管理の活動状況を判断する。

計測値の例：

- 定量的プロセス管理活動の延べコストで、計画と比較したもの
- 定量的プロセス管理活動に対するスケジュール上のマイルストーンの達成で、承認された計画と比較したもの（例えば、プロジェクトで使用されるべきプロセス計測値の確立、プロセスデータの収集方法の決定、およびプロセスデータの収集）

履行検証

検証1

定量的プロセス管理の活動は、上級管理層によって定期的にレビューされる。

(検証1)

上級管理層の監督レビューの典型的な内容を扱うプラクティスについては、『組織プロセス重視』キープロセスエリアの検証1および『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証1を参照。

検証2

定量的プロセス管理に対するソフトウェアプロジェクトの活動は、プロジェクトマネージャによって定期的に、かつイベント発生を契機としてレビューされる。

プロジェクトマネージャの監督レビューに関して、その典型的な内容を扱うプラクティスについては『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証2を参照。

検証3

ソフトウェア品質保証グループは、定量的プロセス管理の活動と作業成果物をレビューかつ/または監査し、その結果を報告する。

『ソフトウェア品質保証』キープロセスエリアを参照。

レビューや監査では、最低限、以下のことを検証する：

1. 定量的プロセス管理活動の計画に従っている。
2. 定量的プロセス管理の手順に従っている。
3. 定量的プロセス管理データの収集と分析が、以下の検証を含めて、要求どおり実施されている：
 - 必要なデータが存在している。
 - 必要なデータが収集される。
 - 収集されたデータは必要なものである。

(検証3)

- 収集されたデータは、組織の計測プログラムの目標とねらいを支援するものである。
- データ収集コストは、そのデータの有用性から見て正当である。
- データは、ソフトウェアライフサイクルの正しい箇所で収集される。
- データは正確で正しい。
- データはタイムリーである。
- データの機密性は適切に保護される。

ソフトウェア品質管理

レベル4のキープロセスエリア： 管理された段階

『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』に必然的に含まれる活動には、ソフトウェア成果物の品質目標を定義すること、これらの目標を達成するための計画を確立すること、および顧客とエンドユーザの高品質な成果物に対するニーズと要望を満足させるため、ソフトウェア計画、ソフトウェア作業成果物、活動、および品質目標をモニターし補正を行うことが挙げられる。

『ソフトウェア品質管理』のプラクティスは、『ソフトウェア統合管理』と『ソフトウェアプロダクトエンジニアリング』、および『定量的プロセス管理』キープロセスエリアのプラクティスの上に構築される。『ソフトウェア統合管理』と『ソフトウェアプロダクトエンジニアリング』は「プロジェクトの定義されたソフトウェアプロセス」を確立し実装する。『定量的プロセス管理』は望まれる結果を達成するために「プロジェクトの定義されたソフトウェアプロセス」の能力の定量的な理解を確立する。

定量的な目標は、組織、顧客、およびエンドユーザのニーズに基づき、ソフトウェア成果物に対して確立される。組織はこれらの定量的な目標を達成すべく戦略と計画を確立し、そしてプロジェクトは品質目標を達成すべく「プロジェクトの定義されたソフトウェアプロセス」を具体的に補正する。

ゴール

- 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。

実施のコミットメント

- コミットメント1 プロジェクトは、ソフトウェア品質管理に関して、明文化された組織方針に従う。

この方針で明記される典型的な事項：

1. プロジェクトのソフトウェア品質管理活動は、ソフトウェア成果物の品質を改善するための組織のコミットメントを支援する。

ソフトウェア成果物の品質を向上させるプロセスの改善は、組織の最優先事項である。

ソフトウェア成果物の新しいリリースはそれぞれ、前のリリースまたは主な競争相手よりも優れていることが計測可能であるべきである。

2. プロジェクトは、「プロジェクトの定義されたソフトウェアプロセス」に基づいて、ソフトウェア品質管理のために用いる計測値を定義し収集する。
3. プロジェクトは、ソフトウェア成果物のための品質目標を定義し、それに向けての進捗をモニターする。

- (コミットメント1) 4. ソフトウェア品質管理の責任が定義され、ソフトウェアエンジニアリンググループと他のソフトウェア関連グループに割り当てられる。

ソフトウェア関連グループの例：

- ソフトウェア品質保証
- ソフトウェア構成管理
- 文書化支援

- そのグループがソフトウェア成果物に対する品質目標の達成に成功したと判断できるように、基準が確立される。

実施能力

能力1

ソフトウェア成果物の品質を管理するために適切な資源と資金が提供される。

1. ソフトウェア品質目標の設定やその目標に対する進捗のレビューを手助けしてもらうために、安全性や信頼性のような分野の専門技術者を利用することができる。
2. ソフトウェア品質の予測、計測、確認、および分析を支援するツールが利用可能になっている。

支援ツールの例：

- データ収集ツール
- データベースシステム
- 表計算プログラム
- ソフトウェアライフサイクルシミュレータ
- 定量的分析ツール
- コード監査ツール

能力2 **ソフトウェア品質管理を履行し支援する個人は、その活動を実施するための必修トレーニングを受ける。**

トレーニングの例：

- 成果物に対する品質コミットメントの計画と品質目標の計画
- 成果物とプロセス品質の計測
- 定義されたソフトウェアプロセスを用いた成果物品質の制御

『トレーニングプログラム』キープロセスエリアを参照。

能力3 **ソフトウェアエンジニアリンググループと他のソフトウェア関連グループのメンバは、ソフトウェア品質管理についての必修トレーニングを受ける。**

トレーニングの例：

- 成果物品質を定量的に管理することの目標およびその利点の理解
- 計測データの収集
- ソフトウェアプロセスと成果物に対する品質計測の理解
- ソフトウェア成果物の品質の計画と制御

『トレーニングプログラム』キープロセスエリアを参照。

実施される活動

活動1

プロジェクトのソフトウェア品質計画は、文書化された手順に従って策定し保守する。

この手順で明記される典型的な事項：

1. 組織、顧客、およびエンドユーザなどのソフトウェア品質ニーズが適宜、理解される。

これらのプラクティスで意味するエンドユーザとは、顧客が指名するエンドユーザ、またはエンドユーザの代表をいう。

顧客とエンドユーザのソフトウェア品質ニーズを計測する方法の例：

- 調査
- フォーカスグループ
- ユーザによる成果物評価

2. 組織、顧客、およびエンドユーザのソフトウェア品質のニーズと優先順位は、その「ソフトウェアに割り当てられたシステム要件」とソフトウェア品質目標に対して、追跡可能である。

これらのニーズと優先順位を追跡する手法の例として、『品質機能展開』(QFD)がある。

成果物に対するソフトウェア品質目標のニーズと優先順位を追跡する例：納入後の欠陥数の目標値を確立する。その後、成果物が成熟するにつれて予測活動を実施し、これらの目標値に合致する見込みをアセスメントする。

(活動1)

この「ソフトウェアに割り当てられたシステム要件」は、これらのプラクティスでは、『割り当てられた要件』として参照される。

「ソフトウェアに割り当てられたシステム要件」を扱うプラクティスについては、『要件管理』キープロセスエリアを参照。

3. ソフトウェアの品質目標を満足するため、「プロジェクトの定義されたソフトウェアプロセス」の能力がアセスメントされ文書化される。

頑健な設計のための『品質機能展開』や『田口メソッド』のような技法は、成果物の品質目標をプロセス能力に関連づけるために用いることができる。

4. ソフトウェア品質計画は組織の品質計画を適宜満足する。
5. ソフトウェア品質計画は、組織における以前のもしくは現行のプロジェクト計画に適宜基づいている。
6. ソフトウェア品質計画は、プロジェクトの開始時、プロジェクトの主要なマイルストーン時、および割り当てられた要件が著しく変更される時点毎に更新される。
7. ソフトウェア品質計画はピアレビューを受ける。

『ピアレビュー』キープロセスエリアを参照。

8. ソフトウェア品質計画は、影響を受けるグループと個人によりレビューされる。

(活動1)

影響を受けるグループと個人の例：

- 顧客
- エンドユーザ
- ソフトウェアエンジニアリング（ソフトウェア設計などのすべてのサブグループを含む）
- ソフトウェア見積り
- システムエンジニアリング
- システムテスト
- ソフトウェア品質保証
- ソフトウェア構成管理
- 契約管理
- 文書化支援

9. 上級管理層がソフトウェア品質計画をレビューする。
10. ソフトウェア品質計画は、管理され制御されている。

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キープロセスエリアに記述されている。

11. ソフトウェア品質計画は、影響を受けるすべてのグループと個人が利用可能である。

活動2

プロジェクトのソフトウェア品質計画が、プロジェクトのソフトウェア品質管理活動の基礎である。

(活動2)

この計画が扱う事項：

1. プロセスにおけるソフトウェア品質が計測される箇所。
2. ソフトウェア成果物に対する品質目標でレバレッジの高いもの。

ソフトウェア成果物に対する品質目標でレバレッジの高いものとは、最小のコストで最大の顧客満足をもたらすものであったり、顧客またはエンドユーザの『必須要件』である。

3. ソフトウェアプロジェクトが過去の品質実績から改善するための処置。
4. ソフトウェア成果物品質を計測するための活動。

ソフトウェア成果物品質を計測するためのソフトウェア活動の例：

- ピアレビュー
- プロトタイプ開発
- 成果物のシミュレーション
- テスト

5. ソフトウェア作業成果物に対する品質目標で、適宜。

ソフトウェア成果物の品質目標の中で、プロジェクトのソフトウェア品質計画に文書化するものとして適切な例：

- 満足されるべく計画された特徴。
- 満足されない場合、その成果物が顧客またはエンドユーザにとって、好ましくないもの、ないしは不必要なものになってしまうような決定的な特徴。

6. ソフトウェア成果物品質が品質目標を満足しないと予想された時にとられる処置。

活動3

ソフトウェアライフサイクルを通じて、ソフトウェア成果物に対するプロジェクトの定量的品質目標を、定義し、モニターし、そして改訂する。

1. ソフトウェア成果物がいかに良く動作するか、またはそれがいかに良く開発され保守されるかを記述する成果物品質の特徴が特定される。

ソフトウェア成果物品質の特徴の例：

- 機能性
- 信頼性
- 保守性
- 使用性

2. ソフトウェア成果物品質の特徴を定量化するために使用される計測値が特定される。

ソフトウェア成果物品質に対する計測値を特定する活動の例：

- 過去の実績データと顧客要件のレビュー
- プロトタイプ開発
- 形式的表現による中間ソフトウェア成果物の記述
- 形式的なソフトウェアエンジニアリング手法の使用
- テストの開催

3. ソフトウェア成果物品質の各特徴について、計測可能な数値が、その要求値と要望値に基づき、成果物の品質目標として選択される。

(活動3)

ソフトウェア成果物の信頼性に対する品質目標の候補の例：

- 要件に明記された平均故障時間
- 達成すべき平均故障時間（分析と実験により判断された）
- 達成されるように計画された平均故障時間

4. ソフトウェア成果物の品質目標は、プロジェクトのソフトウェア品質計画の中に文書化される。

ソフトウェア成果物の品質目標の中で、プロジェクトのソフトウェア品質計画に文書化するものとして適切な例：

- 満足されるべく計画された特徴。
- 満足されない場合、その成果物が顧客またはエンドユーザにとって、好ましくないもの、ないしは不必要なものになってしまうような決定的な特徴。

5. ソフトウェアライフサイクルの各ステージの品質目標は、定義され文書化される。

ソフトウェアライフサイクルのステージの例：

- ソフトウェア要件
- ソフトウェア設計
- コーディング
- ソフトウェアテスト

(活動3)

ソフトウェアライフサイクルのステージに関連する品質目標の例：

- ソフトウェアライフサイクルの各ステージに関連した成果物欠陥は、予め決定された割合により、以前の成果物リリースよりも引き下げる
- テストサイクルの終了までに、予測により予め決定された割合の欠陥を発見する

6. ソフトウェア成果物とソフトウェアライフサイクルステージに対する品質目標は、成果物の理解と、組織、顧客、およびエンドユーザのニーズの理解が深まるにつれて改訂される。

活動4

プロジェクトのソフトウェア成果物の品質は、イベント発生を契機として計測し、分析し、そして成果物の定量的品質目標との比較を行う。

計測データの利用を扱うプラクティスについては、『定量的プロセス管理』キープロセスエリアを参照。

1. ソフトウェアタスクは、プロジェクトのソフトウェア品質目標を取り上げるため、計画され実施される。ソフトウェアタスクの開始時には、そのタスクを実施するチームは以下を行う：
- ソフトウェア成果物の品質目標をレビューする。
 - ソフトウェアタスクに適用される品質目標を決定する。
 - ソフトウェア品質目標を達成するための計画を特定する。
 - ソフトウェア品質目標を満足するためにプロセスになされた変更をレビューする。

変更の例としては、ピアレビューで見逃されたことが判明した欠陥を取り上げるピアレビューチェックリストの改訂がある。

2. ソフトウェアライフサイクルの各ステージにおけるソフトウェア作業成果物の品質を計測する。

(活動4)

作業成果物の品質を計測する手法の例：

- ピアレビュー
- シミュレーション
- テスト

3. 品質目標が満足されたか否かを判断するために品質計測値を分析し、ソフトウェア品質目標と比較する。
4. 成果物の品質尺度がソフトウェア品質目標に収まるように、ソフトウェア品質計画と首尾一貫した適切な処置を取る。
5. ソフトウェア品質目標間で矛盾が存在すると判断された場合（すなわち、ある目標が他の目標の妥協なしには達成できない場合）、その矛盾を解決するための処置を取る。
 - ソフトウェア品質目標を達成するためのコストが分析される。
 - ソフトウェア品質目標の代替案が、短期の優先順位と同様に長期の事業戦略に照らして検討される。
 - 顧客やエンドユーザなどが品質のトレードオフ決定に適宜参加する。
 - ソフトウェア作業成果物や計画などが、トレードオフ結果を反映するように適宜改訂される。

活動5

成果物に対するソフトウェアプロジェクトの定量的品質目標は、そのプロジェクトにソフトウェア成果物を納入する外注先に適切に割り当てる。

『ソフトウェア外注管理』キープロセスエリアの活動1を参照。

計測と分析

計測1

計測を行い、その結果を使用して、ソフトウェア品質管理の活動状況を判断する。

計測値の例：

- 低品質の場合のコスト（いかなる精度であれ、収集されうる既知の品質計測値に基づく）
- 品質目標を達成するためのコスト

履行検証

検証1

ソフトウェア品質管理の活動は、上級管理層によって定期的にレビューされる。

上級管理層の監督レビューに関して、その典型的な内容を扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証1を参照。

検証2

ソフトウェア品質管理の活動は、プロジェクトマネージャによって定期的に、かつイベント発生を契機としてレビューされる。

プロジェクトマネージャの監督レビューに関して、その典型的な内容を扱うプラクティスについては、『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証2を参照。

検証3

ソフトウェア品質保証グループは、ソフトウェア品質管理の活動と作業成果物をレビューかつ/または監査し、その結果を報告する。

『ソフトウェア品質保証』キープロセスエリアを参照。

レビューや監査では、最低限、以下のことを検証する：

1. プロジェクトのソフトウェア品質計画の準備。
2. ソフトウェア品質目標の確立と進捗確認のプロセス。

レベル4 索引

(英和索引と日本語索引は巻末にある。)

- defects
 - actions to prevent 21
- documented procedure
 - control of project's defined software process 10
 - process capability baseline 13
 - quantitative process management plan 6
 - software quality plan 23
 - quantitative process control data 9
- manage and control
 - process capability baseline 14
 - process performance baseline 12
 - quantitative process management plan 7
 - software quality plan 25
- measurement
 - program for organization 4
- measurement data
 - collecting and analyzing 5
 - collection of 9
 - examples of 9
 - storage of 10
 - techniques for analyzing 11
- Measurements
 - quantitative process management 15
 - software quality management 31
- organization's software process database 13
 - storage of quantitative process management data 10
- organization's standard software process
 - analyzing capability of 3
 - baseline for 3, 13
 - changes to 15
 - process capability trends for 13
- orientation
 - quantitative process management 6
- Peer Review
 - quantitative process management plan 7
 - software quality plan 25
- policy
 - quantitative process management 2-3
 - software quality management 20
- process capability
 - definition of 1
 - trends in 13
- process capability baseline 3
 - for organization's standard software process 13
 - management and control of 14
- process performance
 - definition of 1
- process performance baseline 13
 - management and control of 12
- project manager
 - review
 - quantitative process management 16

- software quality management 31
- project's defined software process
 - measurement and control of 2
 - relation to quality goals 19, 24
 - quantitative control of 10
 - use for quantitative process management 8
 - use for software quality management 21
- Quality Function Deployment (QFD) 24
- quality goals
 - allocation to subcontractors 30
 - definition and revision of 27
- quality needs
 - traceability of 24
- Quantitative Process Management 1-17
 - measurements for 15
 - orientation in 6
 - plan for 6
 - policy for 2-3
 - reports for 12
 - resources and funding for 4
 - review
 - project manager 16
 - senior management 16
 - software quality assurance 17
 - team for 4
 - tools for 5
 - training for 5
- quantitative process management plan
 - items in 7
- Report
 - quantitative process management 12
- resources and funding
 - quantitative process management 4
 - software quality management 21
- review
 - quantitative process management
 - project manager 16
 - senior management 16
 - software quality assurance 17
 - software quality management
 - project manager 31
 - senior management 31
 - software quality assurance 32
- senior management
 - review
 - quantitative process management 16
 - software quality management 31
 - software quality plan 25
- software engineering group
 - training for software quality management 22
- software engineering process group
 - review of quantitative process management plan 7
- software quality assurance group
 - review
 - quantitative process management 17
 - software quality management 32

- software quality goals
 - actions to resolve conflicts 30
 - documentation of 28
- Software Quality Management 19-32
 - measurements for 31
 - policy for 20
 - resources and funding for 21
 - review
 - project manager 31
 - senior management 31
 - software quality assurance 32
 - tools for 21
 - training for software engineering group 22
 - training to implement 22
- software quality plan
 - documentation of quality goals 28
 - items in 26
 - management and control of 25
 - peer review of 25
 - requirements for 23
 - review by senior management 25
- software subcontractor
 - quality goals for 30
- quantitative control
 - definition of 3
- Taguchi's method for robust design 24
- team
 - quantitative process management 4
- Tools
 - quantitative process measurement 5
 - software quality management 21
- training
 - implementing software quality management 22
 - quantitative process management 5

欠陥予防

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』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。

『欠陥予防』に必然的に含まれる活動には、過去に遭遇した欠陥を分析すること、および将来同じタイプの欠陥が発生することを予防するために特定の処置を行うことが挙げられる。欠陥は、現在のプロジェクトの先行ステージまたは先行タスクと同様に他のプロジェクトにおいて特定されている場合もある。欠陥予防活動もまた、教訓をプロジェクト間で広めるメカニズムのひとつである。

遭遇した欠陥のタイプを確認し、再発しそうな欠陥を特定するために、傾向を分析する。「プロジェクトの定義されたソフトウェアプロセス」とそれがどのように履行されたか(『ソフトウェア統合管理』と、『ソフトウェアプロダクトエンジニアリング』のキープロセスエリアに記述されているように)の理解に基づいて、欠陥の根本原因を決定し、将来の活動に対する欠陥の影響を見定める。

欠陥の再発を予防するため、プロジェクトおよび組織の双方は、特定の処置を行う。組織的処置は、『プロセス変更管理』のキープロセスエリアに記述されているように扱われるものもある。

ゴール

- 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。

実施のコミットメント

- コミットメント1 組織は、欠陥予防活動に関して、明文化された方針に従う。

この方針で明記される典型的な事項：

1. 長期計画とコミットメントが、欠陥予防のための予算、配員、およびその資源のために確立される。
2. 必要な資源が欠陥予防活動に割り当てられる。
3. ソフトウェアプロセスと成果物を改善するため、欠陥予防活動が組織横断的に実施される。
4. 活動の有効性を確実なものとするために、欠陥予防活動の結果がレビューされる。
5. 欠陥予防活動の結果として特定された管理面と技術面の処置が取り上げられる。

コミットメント2 プロジェクトは、欠陥予防活動に関して、明文化された組織方針に従う。

(コミットメント2) この方針で明記される典型的な事項：

1. 欠陥予防活動が、各プロジェクトのソフトウェア開発計画に含まれる。
2. 必要な資源が、欠陥予防活動に割り当てられる。
3. 欠陥予防活動の結果として特定されたプロジェクトの管理面と技術面の処置が取り上げられる。

実施能力

能力1

欠陥予防活動を調整する組織レベルのチームが存在する。

1. このチームは、組織のソフトウェアプロセス活動に責任のあるグループ（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）の一部であるか、その活動がグループ間で密に調整されているかのどちらかである。

『組織プロセス重視』キープロセスエリアを参照。

能力2

ソフトウェアプロジェクトのために欠陥予防活動を調整するチームが存在する。

1. このチームは、「プロジェクトの定義されたソフトウェアプロセス」の開発と維持に責任を持つチームと、密に結び付いている。

欠陥予防活動を調整するチームのメンバは、通常このチームに専任でない形で割り当てられており、他のソフトウェアエンジニアリング活動が主要な責任である。

(能力2)

「プロジェクトの定義されたソフトウェアプロセス」の開発と維持を扱うプラクティスについては、『ソフトウェア統合管理』キープロセスエリアの活動1と2を参照。

能力3

プロジェクト及び組織レベルでの欠陥予防活動のために、十分な資源と資金が提供される。

1. 欠陥予防活動は、各人に責任が適宜割り当てられるように、計画される。

欠陥予防活動の例：

- タスク立ち上げ会議
- 原因分析会議
- 提案された処置内容のレビューと計画
- 処置の履行

2. 欠陥予防活動への管理層の参加が計画される。
3. 各ソフトウェアプロジェクトは、組織のための欠陥予防活動を調整するチームに適宜、代表を出している。
4. 欠陥予防活動を支援するツールが利用可能になっている。

支援ツールの例：

- 統計的分析ツール
- データベースシステム

能力4

ソフトウェアエンジニアリンググループと他のソフトウェア関連グループのメンバは、欠陥予防活動を実施するための必修トレーニングを受ける。

(能力4)

ソフトウェア関連グループの例：

- ソフトウェア品質保証
- ソフトウェア構成管理
- 文書化支援

トレーニングの例：

- 欠陥予防の手法
- タスク立ち上げ会議の開催
- 原因分析会議の開催
- 統計的手法(例えば、原因/結果ダイアグラムやパレート分析)

『トレーニングプログラム』キーププロセスエリアを参照。

実施される活動

活動1

ソフトウェアプロジェクトは、欠陥予防活動のための計画を策定し保守する。

この計画は：

1. 今後実施される欠陥予防活動（例えばタスク立ち上げ会議や原因分析会議）を特定する。
2. 欠陥予防活動のスケジュールを明記する。
3. 割り当てられる責任と必要な資源(要員とツールを含む)を扱う。
4. ピアレビューを受ける。

(活動1)

『ピアレビュー』キープロセスエリアを参照。

活動2

ソフトウェアタスクの開始時に、そのタスクを実行するチームのメンバーはソフトウェアタスクの活動の準備と、関連する欠陥予防活動の準備について打ち合わせる。

プロセスの履行の詳細および最近のプロセスの変更について、チームのメンバーに熟知させるために、立ち上げ会議が開催される。

これらの立ち上げ会議が扱う事項：

1. 最近の変更に重点をおいた、そのタスクに適用されるソフトウェアプロセス、標準、手順、手法、およびツール。

以前の原因分析会議からの勧告を評価するため、実験的に変更が実装されることがある。

2. タスクに必要で利用可能なインプット。

3. もし利用可能であれば、例を用いて作成されたアウトプット。
4. アウトプットを評価するために使われる手法。
5. ソフトウェアプロセスへの忠実さを検証するために使われる手法。
6. 現ステージで一般的に起きたり混入されるエラーと、それらのエラーに対して推奨される予防処置のリスト。
7. チームの割り当て。
8. タスクスケジュール。
9. タスクおよびソフトウェアプロジェクトに対するソフトウェア成果物の品質目標。

(活動2)

『ソフトウェア品質管理』キープロセスエリアを参照。

活動3**文書化された手順に従って原因分析会議を開催する。**

この手順で明記される典型的な事項：

1. ソフトウェアタスクを実施する各々のチームは原因分析会議を開催する。
 - 原因分析会議は、タスクの完了後速やかに開催される。
 - 多数の欠陥が発見された結果、追加の原因分析会議が必要であると見なされる場合に限って、ソフトウェアタスクの期間中に開催される。
 - 定期的な原因分析会議は、ソフトウェア成果物が顧客などにリリースされた後、開催される。
 - 長い期間のソフトウェアタスクに対して、適宜、定期的なプロセス内の欠陥予防会議が開催される。

長い間継続しているタスクの例としては、一様の工数を使うような顧客支援タスクがある。

2. 会議は、原因分析会議の開催についてトレーニングされた人によって主導される。
3. 欠陥は、根本原因を判断するために特定され分析される。

根本原因を判断する手法の一つの例としては、原因 / 結果ダイアグラムがある。

4. 欠陥は、根本原因のカテゴリ毎に分類される。

(活動3)

欠陥の根本原因のカテゴリ例：

- 不十分なトレーニング
- 伝達の漏れ
- 問題のすべての詳細についての考慮不足
- 人間が行う手順における間違い（例えばタイピング）

5. 特定された欠陥や同様の欠陥が将来発生することを予防するための処置を提案し、文書化する。

処置提案の例として、以下に対する修正がある：

- プロセス
- トレーニング
- ツール
- 手法
- 伝達
- ソフトウェア作業成果物

6. 欠陥の共通原因は特定され文書化される。

共通原因の例：

- あるシステムの機能呼び出しのなかで頻繁に発生するエラー
- ソフトウェア単位の関連するグループの中で頻繁に発生するエラー

7. 会議の結果は組織や他のプロジェクトで使用するために記録される。

活動4

欠陥予防活動を調整するために割り当てられた各チームは、原因分析会議から出される処置提案の履行について、レビューし調整するために、定期的に会議を行う。

(活動4)

参加するチームは、組織レベルのこともプロジェクトレベルのこともある。

各チームは：

1. 原因分析会議からのアウトプットをレビューし、取りあげられるべき処置提案を選択する。
2. その組織において欠陥予防活動を調整する他のチームから割り当てられた処置提案をレビューし、取りあげられるべき処置提案を選択する。
3. その組織において他のチームが取った処置が、自分たちの活動やプロセスに適用できるかどうかをアセスメントするためのレビューを行う。
4. 処置提案の予備分析を実施し、それらの優先順位を設定する。

優先順位は通常厳格なものではなく、以下の事項の理解に基づく：

- 欠陥の原因
- その欠陥を取り上げなかった場合の影響
- その欠陥予防のためのプロセス改善を履行するコスト
- ソフトウェア品質に対して期待される影響

処置提案に優先順位を設定するために使われる技法の例として、パレート分析がある。

5. 組織内の他のレベルのチームに対して、適宜、処置提案を再割り当てる。
6. 決定の理論的根拠を文書化し、処置提案の提出者に決定とその理論的根拠を提供する。
7. 処置提案に起因する処置項目の履行責任を割り当てる。

(活動4)

- 処置項目の履行には、チームの権限の範囲内にある活動に対する即座の変更を行ったり、他の変更を準備したりすることが含まれる。
 - 通常はチームのメンバーが処置項目を履行するが、場合によっては、他の誰かに処置項目の履行を手配することがある。
8. 欠陥予防の実験結果をレビューし、成功した実験の結果を残りのプロジェクトや組織に対して、適宜、組み込むための処置を取る。

欠陥予防実験の例：

- 一時的に修正されたプロセスの使用
- 新しいツールの使用

9. 処置提案および処置項目の状況を確認する。

10. 適宜、「組織の標準ソフトウェアプロセス」および「プロジェクトの定義されたソフトウェアプロセス」に対して、ソフトウェアプロセス改善提案を文書化する。

処置提案の提出者は、ソフトウェアプロセス改善提案の提出者として指名される。

ソフトウェアプロセス改善提案を扱うプラクティスについては、『プロセス変更管理』キープロセスエリアの活動5を参照。

11. 決着する前に、完了した処置項目をレビューし検証する。
12. 欠陥予防についての著しい努力や成功が認識されることを確実なものにする。

活動5

欠陥予防データを文書化し、欠陥予防活動を調整しているチーム間で横断的に進捗を確認する。

1. 原因分析会議において特定された処置提案は、文書化される。

処置提案の記述にあるデータの例：

- 処置提案の考案者
- 欠陥の記述
- 欠陥原因の記述
- 欠陥原因の分類
- 欠陥が作り込まれたステージ
- 欠陥が特定されたステージ
- 処置提案の記述
- 処置提案の分類

2. 処置提案に伴う処置項目は、文書化される。

処置項目の記述にあるデータの例：

- 処置項目の履行責任者
- 処置項目の影響範囲の記述
- 処置項目の状況を常に通知されるべき個人
- 状況を次回レビューする日
- 主な決定の理論的根拠
- 履行活動の記述
- 欠陥を特定し是正する時間とコスト
- 欠陥を解決しない場合の見積りコスト

3. 欠陥予防データは、管理され制御されている。**(活動5)**

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キープロセスエリアに記述されている。

活動6

欠陥予防処置に伴う「組織の標準ソフトウェアプロセス」の改訂は、文書化された手順に従って組み込む。

「組織の標準ソフトウェアプロセス」を扱うプラクティスについては、『組織プロセス定義』キープロセスエリアの活動1を参照。

活動7 欠陥予防処置に伴う「プロジェクトの定義されたソフトウェアプロセス」の改訂は、文書化された手順に従って組み込む。

「プロジェクトの定義されたソフトウェアプロセス」を扱うプラクティスについては、『ソフトウェア統合管理』キープロセスエリアの活動2を参照。

活動8 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に関連したグループのメンバは、組織とプロジェクトの欠陥予防活動の状況と結果について、フィードバックを定期的にする。

(活動8) フィードバックでは以下を提供する：

1. 主な欠陥の分類の要約
2. 主な欠陥の分類における欠陥の度数分布
3. 主な欠陥の分類を取り上げるためにとられた著しい革新と処置
4. 処置提案と処置項目の要約状況

このフィードバックを行う方法の例：

- 電子掲示板
- ニュースレター
- 情報伝達会議

計測と分析

計測1 計測を行い、その結果を使用して、欠陥予防の活動状況を判断する。

(計測1)

計測値の例：

- 欠陥予防活動（例えば、原因分析会議の開催や処置項目の履行）の累積的なコスト
- 欠陥を特定しそれらを是正するためにかかる時間とコストで、欠陥を是正しない場合の見積りコストと比較したもの
- 提案され、オープンになり、そして完了した処置項目の数を計測した一覧表
- 各ステージで作られた欠陥についての累積数および同様な成果物のリリースでの欠陥数
- 欠陥の数

履行検証

検証1 欠陥予防のための組織活動は、上級管理層によって定期的にレビューされる。

上級管理層による定期的なレビューの主目的は、ソフトウェアプロセス活動に対する認識と見通しを上級管理層に提供することである。活動のレビューは、適切な抽象レベルでタイムリーに行う。レビューの間隔は、組織のニーズに合わせる。例外報告への体制が十分整っていれば、上級管理層のレビュー間隔は長くても構わない。

これらレビューでは以下の項目を扱う：

1. 主な欠陥の分類と、これらの分類における欠陥の度数分布の要約。
2. 主な処置の分類と、これらの分類における処置の度数分布の要約。
3. 主な欠陥の分類を取り上げるためにとられた重要な処置。
4. 提案され、オープンになり、そして完了した処置項目の要約状況。
5. 欠陥予防活動の有効性と、欠陥予防活動による節約量の要約。
6. 完了した欠陥予防活動の実際のコスト、計画された欠陥予防活動の予想コスト。

(検証1)

検証2

ソフトウェアプロジェクトの欠陥予防のための活動は、プロジェクトマネージャによって定期的に、かつイベント発生を契機としてレビューされる。

プロジェクトマネージャの監督レビューに関して、その典型的な内容を扱うプラクティスについては『ソフトウェアプロジェクト進捗管理』キープロセスエリアの検証2を参照。

検証3

ソフトウェア品質保証グループは、欠陥予防のための活動と作業成果物をレビューかつ/または監査し、その結果を報告する。

『ソフトウェア品質保証』キープロセスエリアを参照。

レビューや監査では、最低限、以下のことを検証する：

1. ソフトウェアエンジニアリングマネージャおよび技術要員は、欠陥予防での自らの役割に対するトレーニングを受けている。
2. タスク開始会議や原因分析会議が、適切に開催される。
3. 処置提案のレビューおよび処置項目の履行のためのプロセスに従っている。

技術変更管理

レベル5のキープロセスエリア：最適化する段階

『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、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』に必然的に含まれる活動には、新しい技術を特定し、選択し、そして評価すること、および効果的な技術を組織に取り入れることが挙げられる。そのねらいは、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。

組織は、ソフトウェアプロジェクトと作業するグループ（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、技術支援グループ）を確立し、新しい技術を導入、評価し、また既存技術の変更を管理する。特に、「組織の標準ソフトウェアプロセス」（『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）の能力を改善するであろう技術の変更を重視する。

常にソフトウェア関連の技術革新に注意をはらい、系統的にそれら进行评估、実験することにより、組織は適切な技術を選択し、ソフトウェア品質とソフトウェア活動の生産性を改善する。正規のプラクティスに取り入れる前に、試行を実施し、新しくて証明のない技術をアセスメントする。組織の管理層の適切な主催により、選択された技術を「組織の標準ソフトウェアプロセス」および現在のプロジェクトに適宜組み込む。

これらの技術変更の結果から生じる「組織の標準ソフトウェアプロセス」（『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）と「プロジェクトの定義されたソフトウェアプロセス」（『ソフトウェア統合管理』キープロセスエリアに記述されている）への変更は、『プロセス変更管理』キープロセスエリアに記述されているように扱われる。

ゴール

- 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。

実施のコミットメント

コミットメント1 組織は、技術能力の改善に関して、明文化された組織方針に従う。

この方針で明記される典型的な事項：

1. 技術変更管理の目標を確立し、文書化する。

(コミットメント1) 2. 文書化された計画では、技術変更管理の目標を取り上げる。

コミットメント2 上級管理層は、組織の技術変更管理活動を主催する。

上級管理層は：

1. 成果物開発における成果物品質、生産性、およびサイクルタイムについて、それらの組織目標に対応する戦略の定義作業を助ける。
2. 顧客やエンドユーザなどのニーズと要望に対応する戦略の定義作業を助ける。

これらのプラクティスで意味するエンドユーザとは、顧客が指名するエンドユーザ、またはエンドユーザの代表をいう。

3. 組織の戦略を完遂するため、彼らの目標とアプローチを定義するのに、組織のマネージャと調整する。
4. 組織全体に可視的な技術変更管理の活動に対して、コミットメントする。
5. 予算、配員、および他の資源に関する長期計画とコミットメントを確立する。

コミットメント3 上級管理層は、組織の技術変更管理活動を監督する。

上級管理層は：

1. 技術変更管理の方針確立を助け、これらの方針をレビューして承認する。

2. 技術変更管理活動のための資源を割り当てる。
- (コミットメント3)
3. 組織の戦略と目標を、技術変更管理の戦略へ関連づけさせることを助ける。
 4. 技術変更管理計画の確立に参加する。
 - 上級管理層は、組織のすべての適切なレベルにおいて、技術変更管理の要件と課題を調整する。
 - 上級管理層は、マネージャと要員による支援と参加を確保するため、組織のマネージャと調整する。

実施能力

能力1

組織の技術変更管理活動に責任を持つグループが存在する。

グループは、一連のタスクまたは活動に責任を持つ部門、マネージャ、および個人の集まりである。グループは、専任でない個人の場合、異なる部門から割り当てられた専任でない人達の場合、および専任の人達の場合など、さまざまな形態がある。グループ編成にあたっては、割り当てるタスクや活動、プロジェクト規模、組織構造、および組織の文化などを考慮する必要がある。品質保証グループのようにプロジェクトの活動に焦点を当てるグループもあれば、ソフトウェアエンジニアリングプロセスグループのように組織全体の活動に焦点を当てるグループもある。

1. このグループは、組織のソフトウェアプロセス活動に責任のあるグループ（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）の一部であるか、その活動がグループ間で密に調整されているかのどちらかである。
2. このグループは、以下の活動を調整し助ける：

(能力1)

- 新しい技術を適用するため、可能性のある分野を調査する。
- 新しい技術を選択し計画する。
- 新しい技術を獲得し、導入し、そしてカスタマイズする。
- 組織内の関連する研究開発活動と連絡調整する。
- 問題および機能拡張に関して、技術の供給元と連絡をとる。

能力2

組織の技術変更管理活動に責任を持つグループを確立するとともに要員を配置するために、適切な資源と資金が提供される。

1. このグループは、技術変更管理の構想を評価し、計画し、そして支援するために、専門分野で知識と経験を持つ要員の助力を利用できる。

専門分野の例：

- ワークステーション
- コンピュータのハードウェア
- ソフトウェアの再利用
- コンピュータ支援ソフトウェアエンジニアリング (CASE) 技術
- ソフトウェア計測
- 形式的手法
- プログラミング言語

2. 技術変更管理を支援するツールが利用可能になっている。

支援ツールの例：

- ワークステーション
- データベースプログラム
- オンラインの技術データベースの購読

能力3 **技術変更の評価で必要となるデータを、収集し分析するための支援が存在する。**

(能力3) この支援には、以下の能力が含まれる：

1. 選択されたプロセスおよび成果物のデータを自動的に記録する。
2. データ分析を支援する。
3. 選択されたデータを表示する。

データ分析の結果は、例えばグラフィカルな表示のように、情報内容を伝えるのに適した形式で表示される。

能力4 **技術変更の評価と選択の分析を支援するために、ソフトウェアプロセスとソフトウェア作業成果物に関する適切なデータが利用可能である。**

プロセスと成果物のデータの例：

- プロジェクト、プロセスのステージ、使用されるツールと手法、プログラムの分類、およびプログラム変更の程度など、それぞれの資源消費と生産性
- プロジェクト、各プロジェクトのプロセスのステージ、プログラムの分類、プログラムの規模、およびプログラム変更の程度など、それぞれのスケジュールされた時間
- 欠陥データとレビュー効率を含むピアレビューのデータ
- 欠陥が混入したステージ、除去されたステージ、種類、原因、重大度、および修正に要した時間と工数を示す欠陥データ
- 作成したコード量と作成した文書量を含む変更活動
- 欠陥修正活動に関するデータで、欠陥の特定、欠陥の修正が実装された成果物のバージョン、および各欠陥の修正の実装時に混入した欠陥の特定など
- プロジェクト、成果物の種類、特定の成果物、および特定の成果物の部分（例えば、プログラムモジュール）毎の欠陥密度

能力5

組織の技術変更管理活動に責任を持つグループのメンバは、これらの活動を実施するための必修トレーニングを受ける。

トレーニングの例：

- 「組織の標準ソフトウェアプロセス」
- 技術移転と変更管理
- ソフトウェアプロセス改善
- 組織が使用するツールと手法
- 組織が利用可能な分析や支援の設備
- 統計的品質管理の原理

『トレーニングプログラム』キープロセスエリアを参照。

実施される活動

活動1

組織は、技術変更管理計画を策定し保守する。

この計画が扱う事項：

1. 要員やツールを含め、割り当てられた責任と必要な資源を扱う。
2. 「組織の標準ソフトウェアプロセス」の自動化と改善を行い、組織の市場地位を拡大するための、長期的な技術戦略を定義する。
3. 組織の技術変更管理活動を実施する際に従うべき手順を特定する。

(活動1)

4. 組織ならびにプロジェクト特有のニーズに対応する新しい技術の導入方法を記述する。
 - 技術変更が可能なプロセスエリアが特定される。
 - 技術変更の機会を特定するアプローチが明らかにされる。
 - 特定の計画された技術または候補と考えられる技術が特定される。
 - 適宜、計画した技術の導入から置き換えまでの間の寿命が見積られる。
 - 作成/購入のトレードオフに関する調査が文書化される。
 - 証明のない候補技術をアセスメントするアプローチが定義される。
 - 調達手順と導入手順が定義される。

- 初期トレーニング、継続トレーニング、およびコンサルテーション支援が定義される。

5. ピアレビューを受ける。

『ピアレビュー』キーププロセスエリアを参照。

6. 影響を受けるマネージャによってレビューされる。

活動2

組織の技術変更管理活動に責任を持つグループは、ソフトウェアプロジェクトと作業し技術変更領域を特定する。

このグループは：

1. 技術変更に関する提案を募集する。
2. 組織とプロジェクトのニーズに適すると思われる利用可能な新しい技術を特定する。
 - 定期的な調査が実施され、商用として入手可能な技術が明らかにされる。それらは特定されかつ想定されるニーズを満たす。
 - 系統的な作業を行い、関連する先端技術成果や新しい技術のトレンドに注意を払う。
 - 系統的な作業を行い、組織外で使用されている技術をレビューし、これらの技術と組織内で使用されている技術とを比較する。
 - 新しい技術の利用に成功してきた分野が特定され、利用経験に関するデータと文書が収集され、そしてレビューされる。
3. 新しい技術を評価して、組織とプロジェクトの現在と将来のニーズに適用できるかどうかを判断する。

(活動2)

活動3

ソフトウェアマネージャと技術要員に、常に新技術の情報を通知する。

1. 適宜、新しい技術に関する情報が広められる。
2. 適宜、組織の一部で既に利用されている先進技術の情報が広められる。
3. 適宜、組織に移転されつつある技術の状況に関する情報が広められる。

活動4

組織の技術変更管理に責任を持つグループは、「組織の標準ソフトウェアプロセス」を系統的に分析し、新技術を必要としたり、それにより利益を得る可能性がある領域を特定する。

このグループは：

1. 「組織の標準ソフトウェアプロセス」を分析し、新しい技術がもっとも有益な領域を決定する。
2. 有益な技術変更を特定し、それらの変更の経済性を判断する。
3. 特定した技術と「組織の標準ソフトウェアプロセス」との関係を定義する。

(活動4)

4. 技術変更から期待される成果を、適宜、定量的かつ/または定性的に定義する。
5. 潜在的な技術変更の試行についてそれぞれ、その必要性を判断する。
6. 新しい技術候補の優先順位を判断する。
7. 分析活動の結果を文書化する。

活動5

文書化された手順に従って、組織およびソフトウェアプロジェクトのための技術を選択し獲得する。

この手順で明記される典型的な事項：

1. 新しい技術の調達要求が文書化される。
 - 予想コストがあらかじめ定義されたレベルを超える技術については、管理層の承認が必要になる。
2. 潜在的な技術変更に対して、費用対効果があらかじめ分析される。
3. あらかじめ定義され承認された選択基準を使用して、最大の潜在的な効果が特定される。
4. 選択した技術変更の要件と計画が定義され、文書化される。
 - 実用的であれば、期待される寿命と置き換え/アップグレード計画が見積られる。
 - 適宜、トレードオフ調査が実施され、レビューされ、そして文書化され、技術を内部で開発するか外部から調達するかが判断される。
 - 適宜、試行に基づいた新しい技術の導入が計画され、有効性と経済的な利益が判断される。
 - 影響を受けるグループのマネージャと技術変更管理活動に責任を持つグループにより、要件と計画がレビューされる。

活動6

新技術を正規のプラクティスとして導入する前に、適宜、技術を改善するための試行作業を行う。

1. 使用実績がない技術または先進技術については、その実現可能性と経済性を判断するため、試行作業が行われる。
2. 試行作業の計画が文書化される。
 - 計画は、試行作業の目標、評価基準、および活動を扱う。

3. 試行作業の実施計画は、影響を受けるグループのマネージャによりレビューされ、承認される。

影響を受けるグループの例：

- ソフトウェアエンジニアリング(すべてのサブグループを含む)
- ソフトウェア見積り
- システムエンジニアリング
- システムテスト
- ソフトウェア品質保証
- ソフトウェア構成管理
- 契約管理
- 文書化支援

4. 技術変更管理活動に責任を持つグループは、試行作業を履行するプロジェクトに対してコンサルテーションと援助を提供する。

5. 試行作業は、開発あるいは保守環境に相当する環境で実施される。

6. 試行作業の結果は、収集、分析、そして文書化される。

(活動6)

- 作業をとおして得られた教訓と遭遇した問題が文書化される。
- 組織でさらに広く使用する場合の利益と影響が見積られる。見積りの不確定性がアセスメントされる。
- 作業を終結するか、その技術を大規模に実装するか、あるいは再計画を行い試行を継続するかについての決定がなされる。

活動7

文書化された手順に従って、「組織の標準ソフトウェアプロセス」に適切な新技術を組み込む。

「組織の標準ソフトウェアプロセス」の変更を扱うキープラクティスについては、『組織プロセス定義』キーププロセスエリアの活動1と『プロセス変更管理』キーププロセスエリアの活動5を参照。

活動8 文書化された手順に従って、「プロジェクトの定義されたソフトウェアプロセス」に適切な新技術を組み込む。

「プロジェクトの定義されたソフトウェアプロセス」の改訂を扱うキープラクティスについては、『ソフトウェア統合管理』キーププロセスエリアの活動2を参照。

計測と分析

計測1 計測を行い、その結果を使用して、組織の技術変更管理の活動状況を判断する。

(計測1)

計測値の例：

- 変更の数、種類、および規模など、技術変更活動全般
- 技術変更の実装効果で、目標と比較したもの

履行検証

検証1 組織の技術変更管理の活動は、上級管理層によって定期的にレビューされる。

上級管理層による定期的なレビューの主目的は、ソフトウェアプロセス活動に対する認識と見通しを上級管理層に提供することである。活動のレビューは、適切な抽象レベルでタイムリーに行う。レビューの間隔は、組織のニーズに合致させる。例外報告が可能な体制が十分整っていれば、上級管理層のレビュー間隔は長くても構わない。

これらのレビューでは：

1. 技術変更管理についての活動を要約する。
2. 必要な戦略上の変更を特定する。
3. 結果として課題を解決する。
4. 結果として、適宜、技術変更管理に対する計画の改訂を承認する。

検証2

ソフトウェア品質保証グループは、技術変更管理の活動と作業成果物をレビューかつ/または監査し、その結果を報告する。

(検証2)

『ソフトウェア品質保証』キープロセスエリアを参照。

レビューや監査では、最低限、以下のことを検証する：

1. 技術変更管理の計画
2. 新しい技術の選択、調達、および導入に関するプロセス

プロセス変更管理

レベル5のキープロセスエリア：最適化する段階

『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』に必然的に含まれる継続的な活動には、プロセス改善目標の定義をすること、上級管理層の主催により、先を見越してかつ系統的に、「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」への改善を特定し、評価し、そして実装することが挙げられる。

組織に属する人たち全員のプロセス改善活動への参加を可能にし奨励するため、トレーニングプログラムと奨励プログラムが確立される。組織への潜在的な見返りに対し、改善の機会が特定され評価される。プロセス変更をアセスメントするために、正規のプラクティスに組み込まれる前に試行作業が実施される。

ソフトウェアプロセスの各改善が正規のプラクティスに対して承認されたとき、「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」は適宜改訂される。「組織の標準ソフトウェアプロセス」を改訂するプラクティスは『組織プロセス定義』キープロセスエリアにあり、「プロジェクトの定義されたソフトウェアプロセス」を改訂するプラクティスは『ソフトウェア統合管理』キープロセスエリアにある。

ゴール

- ゴール1 継続的なプロセス改善が計画されている。
- 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。

実施のコミットメント

- コミットメント1 組織は、ソフトウェアプロセス改善の実装に関して明文化された方針に従う。

この方針で明記される典型的な事項：

1. 組織は、定量的かつ計測可能なソフトウェアプロセス改善の目標を持ち、これらの目標に照らして実績の進捗を確認する。
2. 組織のプロセス改善は、プロダクト品質の改善、生産性の向上、および成果物開発のサイクルタイム短縮に向けられる。
3. すべての組織の要員とマネージャは、ソフトウェアプロセス改善への参加を期待される。

熟練し動機づけられた人は、主要なプロセス改善資源と見なされる。

- コミットメント2 上級管理層は、ソフトウェアプロセス改善の組織活動を主催する。

(コミットメント2) 上級管理層は：

1. 組織のプロセス改善の長期目標と計画を確立する。
2. プロセス改善活動の資源を割り当てる。
3. ソフトウェアマネージャと調整し、妥当で、かつ積極的なプロセス改善目標を持ち、これらの目標を満足する効果的なプロセス改善計画を持つことを確実なものにする。
4. 目標に対しプロセス改善実績をモニターする。
5. 成果物が危機に直面しても、プロセス改善の重視に対して首尾一貫した優先順位を維持する。
6. プロセス改善の課題を迅速に解決することを確実なものにする。
7. プロセス改善活動への従業員の参加に報酬を与える。

実施能力

能力1

ソフトウェアプロセス改善活動のために、適切な資源と資金が提供される。

1. 資源は以下のために割り当てられる：
 - プロセス改善活動を主導し、ガイドし、そして支援する。
 - プロセス改善の記録を維持する。
 - プロセスの変更内容を作成し、制御し、そして広める。
 - 従業員の高い参加レベルを維持するのに必要なコミュニケーション、動機付け、および表彰の活動を開催するために、管理運営上および人的資源の機能を確立し運用する。

- (能力1)
2. ソフトウェアプロセスの定義と分析に専門知識と経験を持つ人が、組織のプロセス改善活動に利用可能である。
 3. プロセス改善を支援するツールが利用可能になっている。

支援ツールの例：

- 統計的分析ツール
- データベースシステム
- プロセス自動化ツール
- プロセスモデリングツール

- 能力2
- ソフトウェアマネージャはソフトウェアプロセス改善についての必修トレーニングを受ける。**

トレーニングの例：

- 技術面と組織面の変更の管理
- チーム作り
- 継続的なプロセス改善に適用されるチームワークスキル

『トレーニングプログラム』キープロセスエリアを参照。

- 能力3
- ソフトウェアエンジニアリンググループと他のソフトウェア関連グループのマネージャと技術要員は、ソフトウェアプロセス改善についての必修トレーニングを受ける。**

(能力3)

ソフトウェア関連グループの例：

- ソフトウェア品質保証
- ソフトウェア構成管理
- 文書化支援

トレーニングの例：

- 品質とプロセス改善の原理
- プロセス改善を提案する手順

『トレーニングプログラム』キープロセスエリアを参照。

能力4

上級管理層は、ソフトウェアプロセス改善についての必修トレーニングを受ける。

トレーニングの例：

- ベンチマ - キングと比較評価
- プロセス改善の原理
- プロセス改善の目標設定と進捗確認
- 継続的なプロセス改善環境下での動機づけとチーム作り

実施される活動

活動1

ソフトウェアプロセス改善プログラムを確立して、組織のメンバに組織のプロセスを改善する権限を与える。

活動2

組織のソフトウェアプロセス活動に関して責任のあるグループ（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）は、ソフトウェアプロセス改善活動を調整する。

組織のソフトウェアプロセス改善活動に関して責任のあるグループを取り扱っているプラクティスについては、『組織プロセス重視』キープロセスエリアを参照。

このグループは：

1. ソフトウェアプロセス実績に対して組織の目標と計測計画を定義する。
2. 承認を得るために、プロセス実績に対する組織目標を上級管理層とレビューする。
3. プロセス改善に対する組織のトレーニングニーズを定義する作業に参加し、トレーニングコース教材の開発と説明を支援する。

『トレーニングプログラム』キープロセスエリアを参照。

4. プロセス改善提案を扱うための手順を定義し、維持する。
5. ソフトウェアプロセス改善提案をレビューし、そしてこれらの提案に対する処置を調整する。
6. プロセス改善活動の状況、成果、および参加状況の進捗を確認し、その結果を上級管理層に定期的に報告する。
7. 「組織の標準ソフトウェアプロセス」への変更を調整し進捗を確認する。

8. プロセス改善記録を定義し、確立し、そして維持する。

活動3

組織は、文書化された手順に従って、ソフトウェアプロセス改善の計画を策定し保守する。

組織のソフトウェアプロセス改善計画を扱うプラクティスについては、『組織プロセス重視』キープロセスエリアの活動2を参照。

この手順で明記される典型的な事項：

1. ソフトウェアプロセス改善計画は、以下に基づいている：
 - 組織の事業計画および戦略的運営計画
 - 顧客満足度指標
2. ソフトウェアプロセス改善計画は、ピアレビューを受ける。

『ピアレビュー』キープロセスエリアを参照。

3. ソフトウェアプロセス改善計画は、影響を受けるマネージャによってレビューされる。
4. ソフトウェアプロセス改善計画は、管理され制御されている。

『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

『管理され制御されている』状態よりも強い制御状態を望む場合は、作業成果物を厳格な構成管理下に置くことができる。厳格な構成管理については、『ソフトウェア構成管理』キープロセスエリアに記述されている。

活動4

ソフトウェアプロセス改善活動は、ソフトウェアプロセス改善計画に従って実施する。

この計画が扱う事項：

1. 必要な資源で、要員とツールを含む。
2. 改善の優先順位が最も高いプロセスエリア。
3. ソフトウェアプロセスの実績と改善についての、短期および長期の計測可能な目標。
4. 特定のプロセスエリアの改善を取り上げるチームとその任務。

チームの例：

- 作業グループ
- プロセスアクションチーム
- 技術委員会

5. 以下の活動のための手順：
 - 上級マネージャがソフトウェアプロセス改善活動を監督する
 - ソフトウェアマネージャがソフトウェアプロセス改善活動を計画し調整する
 - 個人とチームが、適切なソフトウェアプロセス改善策を特定し、評価し、そして導入する
 - チームが、割り当てられたプロセスエリアのソフトウェアプロセス改善策を構築する
6. 継続的なプロセス改善を維持するのに必要な運営管理と支援の計画。
 - 適切な運営管理の手順が含まれ、ソフトウェアプロセス改善活動への参加が奨励され促進される。

(活動4)

- 運営管理の職員が、ソフトウェアプロセス改善活動の監督とレビューに含まれる。
- 継続的なプロセス改善における従業員の役割および貢献は、認められ表彰される。

活動5

ソフトウェアプロセス改善提案は、文書化された手順に従って取扱われる。

この手順で明記される典型的な事項：

1. ソフトウェアプロセス改善提案が提出される。

ソフトウェアプロセス改善提案は、いつでも提出することが可能であり、またどのソフトウェアプロセスエリアでも取り上げることができる。

ソフトウェアプロセス改善提案の出所の例：

- ソフトウェアプロセスアセスメントの所見や勧告
- 組織のソフトウェアプロセス改善目標
- 顧客のクレームや顧客満足度のデータの分析
- ソフトウェア品質や生産性の目標に照らしたプロジェクトの実績データの分析
- プロセスベンチマークの結果
- プロセスやタスクの自動化の可能性
- 欠陥原因のデータ分析
- ソフトウェアプロセス活動の計測された有効性
- うまく採用されたソフトウェアプロセス改善提案の例
- 以前に提出されたソフトウェアプロセス改善提案へのフィードバックなどで、適宜

2. 各ソフトウェアプロセス改善提案は評価される。ソフトウェアプロセス改善提案を実現するかどうかの決定がなされ、決定の論理的根拠は文書化される。

(活動5)

3. 各ソフトウェアプロセス改善提案の期待利益が判断される。

期待利益のエリアの例：

- 生産性
- 品質
- サイクルタイム
- 顧客あるいはエンドユーザの満足度についてのその他の指標
- その他内部要因

4. 実装のため選ばれたソフトウェアプロセス改善提案の優先順位が判断される。

優先順位の高いソフトウェアプロセス改善提案への重視を維持する。

5. 提案の結果としてのソフトウェアプロセス改善処置の履行が、割り当てられ計画される。

6. かなりの努力を必要とするソフトウェアプロセス改善処置は、履行の責任を持ったチームに割り当てる。

かなりの努力が必要な例として、新しい技術の試行とその他の大きな変化を必要とする改善がある。

特定のソフトウェアプロセスエリアに焦点をあてたチームが確立される。試行に適した処置は調整される。

チームの例：

- ワーキンググループ
- プロセスアクションチーム
- 技術委員会

(活動5)

7. 各ソフトウェアプロセス改善提案の状況は、進捗が確認される。
8. ソフトウェアプロセス改善提案への対応が通常よりも長い場合は、特定され処置が施される。
9. 成果物品質、生産性に主要な影響を与えると思われるか、顧客とエンドユーザの満足度を著しく変えるであろうソフトウェアプロセスの変更は、実装される前に、適切な管理層によってレビューされ承認される。
10. ソフトウェアプロセス改善処置は、完了とする前にレビューされ、検証され、そして承認される。
11. ソフトウェアプロセス改善提案の提出者が受け取るもの：
 - 提案の敏速な受理通知
 - 提案の処理通知

活動6

組織のメンバは、割り当てられたプロセスエリアでソフトウェアプロセス改善策を作成するため、積極的にチームに参加する。

1. 各プロセス改善チームは、資金を与えられ、活動が計画され、スケジュールされる。
2. 各プロセス改善活動毎に目標が確立される。可能な場合、これらの目標は定量的に定義される。
3. 計画は、影響を受けるグループ、および影響を受けるプロセス記述を定義し維持するグループのマネージャによって承認される。

(活動6)

影響を受けるグループの例：

- ソフトウェアエンジニアリング（ソフトウェア設計などすべてのサブグループおよびソフトウェアのタスクリーダー）
- ソフトウェア見積り
- システムエンジニアリング
- システムテスト
- ソフトウェア品質保証
- ソフトウェア構成管理
- 契約管理
- 文書化支援

活動7

ソフトウェアプロセス改善策を適宜、試行的に導入して、正規のプラクティスに組み込む前に利点や有効性を判断する。

1. 実装を最適化するために、試行作業の間、提案されたプロセス改善に対して補正が行われ、文書化される。
2. 教訓および遭遇した問題点が文書化される。
3. プロセス改善策を組織内でより広く使用する場合の利益、リスク、および影響が見積られる。見積りの不確実性がアセスメントされる。
4. 作業を終結するか、その改善策をより大規模に実装するか、あるいは再計画を行い試行を継続するかについての決定がなされる。

活動8

改善策を正規のプラクティスとして組み込むことを決定した場合、文書化された手順に従って履行する。

この手順で明記される典型的な事項：

- (活動8)
1. ソフトウェアプロセスに対する大きな変更を支援するのに必要な資源が確立され資金が与えられる。
 2. ソフトウェアプロセスの実績変化を計測し、進捗を確認するため、データ収集の戦略が文書化され、レビューされ、そして合意される。
 - この戦略は、変更によって影響を受けるソフトウェアプロセスに履行責任を持つ個人により、合意される。
 - 支援ツールには、要望されるデータを自動的に記録するための計測の仕組みが適宜備わっている。
 3. トレーニングコースは、現時点のソフトウェアプロセスを反映するために更新され、プロセスの変更を全体に導入する前に提供される。

『トレーニングプログラム』キープロセスエリア参照。

4. 広い範囲でのプロセス変更の導入前に、期待されたニーズに対して適切なコンサルテーション支援が確立されており、必要に応じて継続される。
5. 適切なプロセス変更は、「組織の標準ソフトウェアプロセス」に組み込まれる。

「組織の標準ソフトウェアプロセス」を扱うプラクティスについては、『組織プロセス定義』キープロセスエリアの活動1参照。

6. 適切なプロセス変更は、「プロジェクトの定義されたソフトウェアプロセス」に組み込まれる。

「プロジェクトの定義されたソフトウェアプロセス」を扱うプラクティスについては、『ソフトウェア統合管理』キープロセスエリアの活動2参照。

活動9 ソフトウェアプロセス改善活動の記録を維持する。**(活動9)**

1. ソフトウェアプロセス改善提案の開始、状況、ならびに実装についての情報は維持される。
2. ソフトウェアプロセス改善の記録には、容易にアクセスできる。
3. 履歴データが維持され、ソフトウェアプロセス改善策のレポートが作成される。

記録およびレポートの例：

- プロジェクトの生産性、品質、およびスケジュールの実績
- プログラムの欠陥履歴
- 組織のソフトウェア品質と生産性の傾向
- ソフトウェアプロセス開発と改善のコスト、スケジュール、および生産性

「組織のソフトウェアプロセスデータベース」を扱うプラクティスについては、『組織プロセス定義』キープロセスエリアの活動5を参照。そのデータベースは、プロセス改善の記録を維持することが可能なメカニズムのひとつである。

活動10

ソフトウェアのマネージャや技術要員は、イベント発生を契機として、ソフトウェアプロセス改善活動の状況と結果のフィードバックを受ける。

フィードバックは以下の項目を提供する：

1. 主要なソフトウェアプロセス改善活動の要約。
2. ソフトウェアプロセス改善に対応するために取られた重要な革新や処置。
3. 提出され、処理中であり、完了したものについてのソフトウェアプロセス改善提案の要約状況。

(活動10)

このフィードバックを提供する手段の例：

- 電子掲示板
- ニュースレター
- 情報伝達会議

計測と分析

計測1

計測を行い、その結果を使用して、ソフトウェアプロセス改善の活動状況を判断する。

計測値の例：

- 各プロセスエリアに対し、提出されたソフトウェアプロセス改善提案の数、および実装された数
- 各プロジェクト、各グループ、および各部門が提出したソフトウェアプロセス改善提案の数
- 各プロジェクト、各グループ、および各部門が受けた賞や報奨のタイプと数
- ソフトウェアプロセス改善提案の処理に要した対応時間
- ある報告周期の中で採用されたソフトウェアプロセス改善提案の割合
- 変更の数、タイプ、および規模を含む全体的な変更活動
- それぞれのプロセス改善の履行の影響で、その定義された目標と比較したもの
- 組織およびプロジェクトのプロセスについての全体的な実績(有効性、品質、および生産性を含む)で、定義された目標と比較したもの
- 各プロジェクトにおける生産性とソフトウェア品質の全体的な傾向
- 顧客満足の指標に関連するプロセス計測

履行検証

検証1

ソフトウェアプロセス改善の活動は、上級管理層によって定期的にレビューされる。

上級管理層による定期的なレビューの主目的は、ソフトウェアプロセス活動に対する認識と見通しを上級管理層に提供することである。活動のレビューは、適切な抽象レベルでタイムリーに行う。レビューの間隔は、組織のニーズに合わせる。例外報告への体制が十分整っていれば、上級管理層のレビュー間隔は長くても構わない。

これらのレビュー開催の目的は：

1. プロセス改善活動への参加を要約する。
2. プロセス実績をアセスメントする。
3. 必要な目標変更を特定する。
4. 課題を解決する。
5. 適宜、ソフトウェアプロセス改善の計画に対する改訂を承認する。

検証2

ソフトウェア品質保証グループは、ソフトウェアプロセス改善の活動と作業成果物をレビューかつ/または監査し、その結果を報告する。

『ソフトウェア品質保証』キープロセスエリアを参照。

(検証2)

レビューや監査では、最低限、以下のことを検証する：

1. 組織のソフトウェアプロセス改善計画の準備。
2. ソフトウェアプロセス改善提案の実装を開始し、提出し、レビューし、承認し、そして計画するプロセス。
3. プロセス計測がソフトウェアプロセスの記述に準拠し、実績を反映する度合い。
4. 「組織の標準ソフトウェアプロセス」および「プロジェクトの定義するソフトウェアプロセス」への変更を文書化し、レビューし、承認し、制御し、そして広めるためのプロセス。
5. ソフトウェアプロセス改善の活動が、首尾一貫して、計測され進捗確認される度合い。

6. ソフトウェアプロセス改善の実績が、計画および目標を達成する度合い。

レベル5 索引

(英和索引と日本語索引は巻末にある。)

-
- action item
 - data in 12
 - action proposal
 - data in 11
 - Causal analysis meetings
 - requirements for 7
 - data
 - defect prevention 11
 - defect
 - examples of root causes 8
 - Defect Prevention 1-15
 - causal analysis meeting for 7
 - documentation of data on 11
 - examples of activities for 4
 - feedback on 13
 - kick-off meeting for task 6
 - management and control of data for 12
 - measurements for 13
 - plan for project 5
 - policy for organization 2
 - policy for project 2
 - resources and funding for 4
 - review
 - project manager 15
 - senior management 14
 - software quality assurance 15
 - team for organization 3
 - team for project 3
 - tools for 4
 - training for 5
 - defect prevention plan
 - peer review of 6
 - defect prevention team
 - activities of 9
 - establishment of 3
 - documented procedure
 - causal analysis meetings for defect prevention 7
 - implementation of software process improvement 43
 - incorporation of new technology into organization's standard software process 28
 - incorporation of new technology into project's defined software process 29
 - revisions to organization's standard software process based on defect prevention 12
 - revisions to project's defined software process based on defect prevention 12
 - selection of new technologies 26
 - software process improvement plan 37
 - software process improvement proposals 39
 - manage and control
 - defect prevention data 12
 - software process improvement plan 38
 - Measurements
 - defect prevention 13
 - software process improvement 46
 - technology change management 29
 - new technology
 - information on 25
 - selection of 26
 - organization's standard software process
 - analysis for new technology 26
 - incorporation of new technology 28
 - incorporation of process changes 44

- revision based on defect prevention
12
- Pareto analysis
 - use in defect prevention 10
- Peer Review
 - defect prevention plan 6
 - software process improvement plan
38
 - technology change management plan
24
- Pilot efforts
 - technology changes 27
- plan
 - defect prevention 5
 - technology change management 23
- policy
 - defect prevention for organization
2
 - defect prevention for project 2
 - software process improvement 32
 - technology change management 18
- Process Change Management 31-47
- project manager
 - review
 - defect prevention 15
- project's defined software process
 - coordination with defect
prevention team 3
 - incorporation of new technology 29
 - incorporation of process changes 44
 - revision based on defect prevention
12
- records
 - software process improvement 44
- resources and funding
 - defect prevention 4
 - software process improvement 33
- technology change management 20
- review
 - defect prevention
 - project manager 15
 - senior management 14
 - software quality assurance 15
 - software process improvement
 - senior management 46
 - software quality assurance 47
 - technology change management
 - senior management 30
 - software quality assurance 30
- senior management
 - oversight of software process
improvement 32
 - review
 - defect prevention 14
 - software process improvement 46
 - technology change management 30
 - role in technology change
management 19-20
 - training for software process
improvement 35
- software development plan
 - incorporation of defect prevention
activities 2
- software engineering environment
 - support for technology change 21
- software engineering group
 - training for software process
improvement 35
- software engineering process group
 - coordination of software process
improvement 36
 - relation with technology change
management 20
- software improvement plan

- management and control of 38
- Software manager
 - responsibility for processes 34
 - training for software process improvement 34
- software process
 - responsibility for 34
- software process improvement
 - feedback on 45
 - implementation of 43
 - measurements for 46
 - oversight by senior management 32
 - pilot installation 42
 - plans for 37
 - policy for 32
 - program for 36
 - records of 44
 - resources and funding for 33
 - review
 - senior management 46
 - software quality assurance 47
 - teams for process areas 42
 - tools for 34
 - training for 34-35
 - update of training for 43
- software process improvement plan
 - items in 38
 - peer review of 38
 - requirements for 37
- software process improvement proposal 36
 - procedure for handling 39
- software process improvement proposals
 - documentation of 11
- software quality assurance group
 - review
 - defect prevention 15
 - software process improvement 47
 - technology change management 30
- software task
 - kick-off meeting for defect prevention 6
- technology change
 - data to support 22
 - identification of areas for 25
 - pilot effort for 27
- Technology Change Management 17-30
 - group responsible for 20
 - measurements for 29
 - plan for 23
 - policy for 18
 - resources and funding for 20
 - review
 - senior management 30
 - software quality assurance 30
 - senior management role in 19
 - training for 23
- technology change management plan 23
 - peer review of 24
- technology support group 20
- Tools
 - defect prevention 4
 - software process improvement 34
- training
 - defect prevention 5
 - software process improvement 34-35
 - technology change management 23
 - update based on software process improvement 43

付録A: 参照文献

- Fagan86 M.E. Fagan, "Advances in Software Inspections," *IEEE Transactions on Software Engineering*, Vol. 12, No. 7, July, 1986, pp. 744-751.
- Feiler92 P.H. Feiler and W.S. Humphrey, *Software Process Development and Enactment: Concepts and Definitions*, CMU/SEI-92-TR-4, ADA258465, March 1992.
- 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.
- 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.
- IEEE-STD-610 ANSI/IEEE Std 610.12-1990, "IEEE Standard Glossary of Software Engineering Terminology," February 1991.
- 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.

参照文献

- 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 Charles V. Weber, *Capability Maturity Model for Software, Version 1.1*, Software Engineering Institute, CMU/SEI-93-TR-24, February 1993.
- 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.
- 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B: 用語集

CMM - 能力成熟度モデル(*capability maturity model*)の頭文字。

SCE - ソフトウェア能力審査(*software capability evaluation*)の頭文字。

SCM - ソフトウェア構成管理(*software configuration management*)の頭文字。

SPA - ソフトウェアプロセスアセスメント(*software process assessment*)の頭文字。

SQA - ソフトウェア品質保証(*software quality assurance*)の頭文字。

アセスメント[*assessment*] - (『ソフトウェアプロセスアセスメント』を参照。)

アプリケーションドメイン[*application domain*] - 関連するシステムの境界づけられた集合(すなわち、特定のタイプの問題を取り上げるシステム)。あるアプリケーションドメインでの開発と保守は、通常、特別の技術かつ/または資源を必要とする。例としては、人事給与システム、命令制御システム、コンパイラ、およびエキスパートシステムがある。

依存関係アイテム[*dependency item*] - 成果物、処置および情報などで、計画されたタスクをある個人やグループが実施するために、別の個人やグループにより、その個人やグループに対して提供されなければならないもの。

逸脱[*deviation*] - 適切な規範、計画、標準、手順、またはレビューされている変数からの注目されるまたは著しい乖離。

イベント発生を契機とするレビュー/活動[*event-driven review/activity*] - プロジェクト内のあるイベントの発生(例えば、公式レビューやライフサイクルステージの完了)に基づいて実施されるレビューまたは活動。(比較のため、『定期的なレビュー/活動』を参照。)

エンジニアリンググループ[*engineering group*] - エンジニアリング機能を代表する個人(マネージャと技術要員の両方)の集まり。エンジニアリング機能の例としては、システムエンジニアリング、ハードウェアエンジニアリング、システムテスト、ソフトウェアエンジニアリング、ソフトウェア構成管理、そしてソフトウェア品質保証がある。

エンドユーザ*[end user]* - システムがその環境に配備された時に、そのシステムが意図された用途に利用する個人またはグループ。

エンドユーザの代表者*[end user representatives]* - エンドユーザ全体を代表するものとして選ばれたエンドユーザのサンプル。

オリエンテーション*[orientation]* - ある題材領域における活動の遂行に責任のある個人を監督し、または接触を持つ者に対して提供される、その題材の概要や入門の説明。（比較のために、『トレーニング』を参照。）

外注管理者*[subcontract manager]* - ひとつまたは複数の外部発注契約を運営管理することに直接の責任を持つ外部発注元組織内のマネージャ。

開発構成管理*[developmental configuration management]* - 技術的および管理的な統制を適用することであって、ソフトウェアおよび関連する技術的文書を指定し制御する。この技術的文書は、開発期間中に進化するソフトウェア作業成果物の構成を定義するものである。開発構成管理は、開発者の直接制御下にある。開発構成管理下にあるアイテムはベースラインではないが、開発のある時点でベースライン化したり、ベースライン構成管理下に置くこともある。

外部発注先*[subcontractor]* - ひとつあるいは複数の成果物を設計、開発、かつ/または製造するための契約を組織（すなわち外部発注元）と結ぶ個人または法人や団体など。

活動*[activity]* - ある目標の達成に向けて、物心両面において取られるステップまたは実施される機能。活動には、マネージャおよび技術要員が、プロジェクトのタスクと組織のタスクを実施するために行うすべての作業が含まれる。（比較のために『タスク』を参照。）

監査*[audit]* - 作業成果物またはその集合の独立した検査で、仕様、標準、契約合意、または他の基準への遵守をアセスメントするためのもの。[IEEE-STD-610]

管理され制御されている*[managed and controlled]* - ソフトウェア作業成果物を特定し定義するプロセスのことであり、対象となるソフトウェア作業成果物は、ベースラインの一部ではなく、したがって構成管理下には置けられないが、プロジェクトが秩序ある方法で進むために制御されるべきもの。『管理され制御されている』とは、ある時点（過去または現在）において使用されている作業成果物のバージョンを把握し（バージョン制御）、制御された方法で変更を取り入れること（変更制御）を指す。

管理されたレベル*[managed level]* - （『成熟度レベル』を参照。）

キープラクティス*[key practices]* - キープロセスエリアの効果的な実装と制度化にもっとも寄与する基盤と活動。

キープロセスエリア*[key process area]* - 関連するひとまとまりの活動であって、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合が達成される。各キープロセスエリアは、ひとつの成熟度レベルに属するように定義されている。これらは主要な組み立てブロックとしてSEIが特定した領域であり、組織のソフトウェアプロセス能力を判断することと、より高い成熟度レベルへ進むために必要な改善項目を理解することに役立つ。CMMにおけるレベル2のキープロセスエリアは、『要件管理』、『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進捗管理』、『ソフトウェア外注管理』、『ソフトウェア品質保証』、および『ソフトウェア構成管理』である。CMMにおけるレベル3のキープロセスエリアは、『組織プロセス重視』、『組織プロセス定義』、『トレーニングプログラム』、『ソフトウェア統合管理』、『ソフトウェアプロダクトエンジニアリング』、『グループ間調整』、『ピアレビュー』である。CMMにおけるレベル4のキープロセスエリアは『定量的プロセス管理』、『ソフトウェア品質管理』である。CMMにおけるレベル5のキープロセスエリアは『欠陥予防』、『技術変更管理』、『プロセス変更管理』である。

技術*[technology]* - ある特定の結果を成し遂げるために、科学および/または工学を適用すること。

技術要件*[technical requirements]* - ソフトウェアが行なければならないこととその運用上の制約を述べた要件。技術要件の例には、機能、性能、インタフェース、そして品質などの要件がある。

機能[*function*] - 関連した活動の集合であって、明確に割り当てられたか、またはその役割にふさわしい個人やツールにより行われるもので、目的や目標の集合を達成するためのもの。

(欠陥の) 共通原因[*common cause (of a defect)*] - プロセスやシステムの一部にもともと存在する欠陥の原因。共通原因は、そのプロセスのすべての結果とそのプロセスで作業するすべての者に影響を与える。(比較のために『特殊原因』を参照。)

クリティカルパス[*critical path*] - プロジェクトにおける依存関係にある一連のタスクで、プロジェクト全体をスケジュール通りに保つために、計画された通りに完了されなければならないもの。

グループ[*group*] - グループは、一連のタスクまたは活動に責任を持つ部門、マネージャ、および個人の集まりである。グループは、専任でない個人の場合、異なる部門から割り当てられた専任でない人達の場合、および専任の人達の場合など、さまざまな形態がある。

計測値[*measurement*] - 寸法、容量、数量、ないしは何かの量(例えば、300ソースコード行、または7設計文書ページ数)。

契約条項[*contract terms and conditions*] - 契約で定められた法的、財務的、および管理的側面。

欠陥[*defect*] - システムまたはシステムコンポーネントの欠点で、システムまたはコンポーネントがその要求された機能を実行することに失敗する原因となるもの。欠陥は、実行時に遭遇した場合、システムの障害の原因となり得る。

欠陥の根本原因[*defect root cause*] - 欠陥が入り込むのを許した内在的な理由(例えば、不完全なプロセス)。

欠陥密度[*defect density*] - 成果物内で特定された欠陥の数を成果物コンポーネントの規模で除算したもの(その成果物に対する標準的な計測用語で示されたもの)。

欠陥予防[*defect prevention*] - 欠陥や潜在的欠陥の特定と、それらが成果物に入り込むのを予防することに伴う活動。

原因分析[*causal analysis*] - 根本原因を判断するための欠陥の分析。

原因分析会議*[causal analysis meeting]* - 特定のタスクの完了後に開催される会議で、タスクの実施中に発見された欠陥を分析するためのもの。

検収基準*[acceptance criteria]* - ユーザ、顧客、または権限を与えられた他の実体により受け入れられるために、システムまたはコンポーネントが満足すべき基準。[IEEE-STD-610]

検収テスト*[acceptance testing]* - システムが検収基準を満足するか否かを判断し、かつ顧客がそのシステムの受け入れの可否を判断できるようにするために実施される公式なテスト。[IEEE-STD-610]

検証*[verification]* - ある開発フェーズの成果物が、そのフェーズの開始時に課された条件を満足するか否かを決定するために、ソフトウェアを評価するプロセス。[IEEE-STD-610]

効果的なプロセス*[effective process]* - 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、そして改善可能なものとして特徴づけられ得るプロセス。（『整った形で定義されたプロセス』も参照。）

公式レビュー*[formal review]* - 公式な会議であって、コメントと承認を得るために、エンドユーザ、顧客、または他の利害関係のある団体に成果物が提示される。これは、管理的、技術的活動およびプロジェクト進捗のレビューにもなり得る。

構成*[configuration]* - 構成管理において、ハードウェアまたはソフトウェアの機能的および物理的特徴であって、技術文書で記述されているもの、または成果物で達成されるもの。[IEEE-STD-610]

構成アイテム*[configuration item]* - ハードウェア、ソフトウェア、あるいは両方を集めたもので、構成管理のために指定され、構成管理プロセスにおいてひとつの実体として扱われるもの。[IEEE-STD-610]

構成管理*[configuration management]* - 技術的および管理的な統制と監視を適用する機能であって、構成アイテムの機能的及び物理的特徴を特定し文書化し、それらの特徴の変更を制御し、変更処理と実施状況を記録し報告し、そして明記された要件に遵守しているか検証をすること。[IEEE-STD-610]

構成管理ライブラリシステム*[configuration management library system]* - ソフトウェアベースラインライブラリの内容にアクセスするためのツールと手順。

構成制御*[configuration control]* - 構成管理の要素であって、構成特定の公式な確立後における構成アイテムの変更の評価、調整、承認または非承認、および実施からなる。[IEEE-STD-610]

構成特定*[configuration identification]* - 構成管理の要素であって、システムの構成アイテムの選択と、それらの機能的及び物理的特徴を技術文書に記録することからなる。[IEEE-STD-610]

構成ユニット*[configuration unit]* - 構成アイテムまたはコンポーネントの最下位の实体で、構成管理ライブラリシステムに保存したり、取り出したりすることができるもの。

ゴール*[goals]* - キープロセスエリアのキープラクティスの要約で、組織やプロジェクトが効果的にキープロセスエリアを実装したか否かを判断するために用いられ得る。ゴールは、各キープロセスエリアの範囲、境界および意図を示す。

顧客*[customer]* - 成果物の検収と開発組織への支払い許可に対して責任がある個人または組織。

コミットメント*[commitment]* - すべての当事者によって自主的に引き受けられ、可視的であり、守られると期待される約束。

コモンスリーチャ*[common features]* - CMMのキープロセスエリアの区分分類。コモンスリーチャは、キープロセスエリアの実装と制度化が効果的か、反復できるか、そして継続的かどうかを示す属性である。CMMのコモンスリーチャは以下の通り：

- **実施のコミットメント***[commitment to perform]* - 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取らなければならない処置。実施のコミットメントには、組織的な方針と上級管理層の支持の確立を必然的に含むのが典型的である。
- **実施能力***[ability to perform]* - ソフトウェアプロセスを十分実装するために、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。実施能力は、資源、組織的構造、およびトレーニングを必然的に含むのが典型的である。

- **実施される活動***[activities performed]* - キープロセスエリアを実装するために必要な役割と手順の記述。実施される活動は、計画と手順の確立、作業の実施、進捗の確認、および必要に応じた是正処置を必然的に含むのが典型的である。
- **計測と分析***[measurement and analysis]* - プロセスを計測し、計測値を分析する必要性の記述。計測と分析は、『実施される活動』の状況と有効性を判断するために採取され得る計測値の例を含んでいるのが典型的である。
- **履行検証***[verifying implementation]* - 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ステップ。検証は、管理層とソフトウェア品質保証によるレビューと監査を含むのが典型的である。

最適化するレベル*[optimizing level]* - (『成熟度レベル』を参照。)

作業内容記述書*[statement of work]* - プロジェクトを完了するために必要なすべての作業の記述であって、顧客から提供されるもの。

システム*[system]* - 特定の機能または機能の集合をするために編成されたコンポーネントの集まり。[IEEE-STD-610]

システムエンジニアリンググループ*[system engineering group]* - システムエンジニアリンググループは、以下の活動に責任を負う個人(マネージャと技術要員の両方)の集まりである。システム要件を明記する；システム要件をハードウェア、ソフトウェア、およびその他のコンポーネントに割り当てる；ハードウェア、ソフトウェア、およびその他のコンポーネント間のインタフェースを明記する；そして、それらの仕様に準拠することを確実なものにするためにこれらのコンポーネントの設計と開発をモニターする。

システム要件*[system requirement]* - システムまたはシステムコンポーネントが満たし、または持つべき条件または能力であって、ユーザが問題を解決するために必要とする条件または能力を満足するためのもの。[IEEE-STD-610]

実稼働ソフトウェア*[operational software]* - 顧客に納入され、意図された環境に配備された時に、あるシステムの中で使用され、運用されることを意図されたソフトウェア。

実施される活動*[activities performed]* - (『コモンフィーチャ』を参照。)

実施能力*[ability to perform]* - (『コモンフィーチャ』を参照。)

実施のコミットメント[*commitment to perform*] - (『コモンフィーチャ』を参照。)

尺度[*measure*] - 計測の単位(ソースコードの行数、設計文書のページ数など)。

重要なコンピュータ資源[*critical computer resource*] - コンピュータ資源のパラメータであって、潜在的な需要が利用可能な量を越える可能性があるために、プロジェクトにとってリスクの源になると考えられるもの。例としては、対象コンピュータのメモリやホストコンピュータのディスク容量がある。

首尾一貫性[*consistency*] - ドキュメントやシステムの部分またはコンポーネント間の均一性、標準化、および矛盾のない度合。[IEEE-STD-610]

手法[*method*] - ほぼ完全な規則と基準の集合であって、タスクを実施して要望される結果に到達する方法を、精確かつ反復できるように確立したもの。

上級マネージャ[*senior manager*] - 組織における十分に高いレベルの管理層の役割であって、主要な焦点は、組織の長期的な活力であり、短期的なプロジェクトや契約上の関心事やプレッシャではない。一般に、エンジニアリングの上級マネージャは、複数のプロジェクトに対して責任を負う。

初期レベル[*initial level*] - (『成熟度レベル』を参照。)

所見[*findings*] - アセスメント、審査、監査、またはレビューの結論であって、調査領域内で最も重要な課題、問題点、または機会を特定するもの。

処置項目[*action item*] - (1) 決着に向けて個人またはグループに割り当てられたリストの中の一単位。(2) 受け入れられた処置提案。

処置提案[*action proposal*] - プロセスまたはプロセス関連項目の変更に対する文書化された提案で、欠陥予防活動の結果として特定された欠陥の将来的な発生を予防するもの。(『ソフトウェアプロセス改善提案』も参照。)

審査[*evaluation*] - (『ソフトウェア能力審査』を参照。)

ステージ[stage] - ソフトウェア開発作業を分割したもの。管理可能な大きさであり、プロジェクトで実施される関連タスクを意味がありかつ計測可能な集合に表わしたもの。ステージは通常、ソフトウェアライフサイクルを小分けにしたものであり、多くの場合、次のステージ着手前に公式レビューを受けることで終結する。

成果物[product] - (『ソフトウェア成果物』および『ソフトウェア作業成果物』を参照。)

成熟度質問票[maturity questionnaire] - ソフトウェアプロセスに関する質問の集合であって、CMMの各キープロセスエリアにおけるキープラクティスを抽出している。成熟度質問票は、組織やプロジェクトがソフトウェアプロセスを確実に実行するための能力を査定する足がかりとして使用される。

成熟度レベル[maturity level] - 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。SEIの『能力成熟度モデル』における5つの成熟度レベルは以下の通りである：

- **初期**[initial] - ソフトウェアプロセスは場当たりの、場合によっては混沌と特徴付けられる。わずかなプロセスしか定義されておらず、成功は個人の努力に依存する。
- **反復できる**[repeatable] - 基本的なプロジェクト管理プロセスが、コスト、スケジュール、および機能充足性の進捗を確認するために確立されている。以前の成功を同様のアプリケーションのプロジェクトで繰り返すために必要なプロセス上の機能が適切に存在する。
- **定義された**[defined] - 管理とエンジニアリングの両方の活動に対するソフトウェアプロセスが、文書化され、標準化され、そして「組織の標準ソフトウェアプロセス」に統合されている。ソフトウェアの開発と保守のために、「組織の標準ソフトウェアプロセス」の承認され、テーラリングされたバージョンをすべてのプロジェクトが使用する。
- **管理された**[managed] - ソフトウェアのプロセスと成果物の品質の詳細な計測値が収集されている。ソフトウェアのプロセスと成果物が両方とも定量的に理解され、制御される。
- **最適化する**[optimizing] - プロセスからの定量的なフィードバックや、革新的なアイデアや技術の試行からの定量的フィードバックにより、継続的なプロセス改善が可能になっている。

制度化[*institutionalization*] - 基盤と企業文化を構築すること。これらは、たとえ元々それらを定義した者が去っても、事業を行う上で継続するように手法、プラクティス、および手順を支援する。

組織[*organization*] - 企業や他の実体（例えば、政府機関やサービス拠点）内の単位で、その中で多くのプロジェクトが全体として管理される。組織内のすべてのプロジェクトは、共通のトップレベルマネージャと共通の方針を共有する。

組織の計測プログラム[*organization's measurement program*] - 組織の計測ニーズを明示することに関連する要素の集合。これには、組織全体に渡る計測値の定義、組織的計測データの収集に関する手法とプラクティス、組織的計測データの分析に対する手法とプラクティス、および組織の計測ゴールが含まれる。

組織のソフトウェアプロセス資産[*organization's software process assets*] - 組織で保守され、ソフトウェアプロセスの開発、テーラリング、維持、および実装においてプロジェクトによって使用される実体の集まり。これらのソフトウェアプロセス資産には、典型的には以下のものが含まれる：

- 「組織の標準ソフトウェアプロセス」
- 使用を承認されたソフトウェアライフサイクルの記述
- 「組織の標準ソフトウェアプロセス」をテーラリングするためのガイドラインと基準
- 「組織のソフトウェアプロセスデータベース」
- ソフトウェアプロセス関連文書のライブラリ

組織がプロセスの定義と保守の活動を実施する上で有用と考えるすべての実体をプロセス資産に含めることができる。

組織のソフトウェアプロセスデータベース [*organization's software process database*] - ソフトウェアプロセスおよび結果としてのソフトウェア作業成果物に関するデータ、特に「組織の標準ソフトウェアプロセス」に関係するものを収集し、利用可能にするために確立されるデータベース。このデータベースは、実際の計測データと関連情報の両方について格納するか参照のための一覧を持つ。関連情報は、計測値データを理解し、その妥当性と適用可能性とをアセスメントするために必要とされる。プロセスと作業成果物のデータの例としては、ソフトウェアの規模、工数、およびコストの見積り；ソフトウェアの規模、工数、およびコストの実際のデータ；生産性データ；ピアレビューのカバレッジおよび効率、およびソフトウェアコード中で発見された欠陥の数と重要度がある。

組織の標準ソフトウェアプロセス [*organization's standard software process*] - 組織内のソフトウェアプロジェクトに渡る共通ソフトウェアプロセスの確立のする基本プロセスの作業面から見た定義。これには、各ソフトウェアプロジェクトがそれ自身の定義されたソフトウェアプロセスに取り入れることを期待されている基本的なソフトウェアプロセス要素を記述する。また、これらソフトウェアプロセス要素間の関係（例えば、順序やインタフェース）も記述する。

ソフトウェアアーキテクチャ [*software architecture*] - ソフトウェアまたはモジュールの編成上の構造。[IEEE-STD-610]

ソフトウェアエンジニアリンググループ [*software engineering group*] - プロジェクトでソフトウェアの開発と保守(要件分析、設計、コーディング、テスト)に責任を持つ個人の集まり(マネージャと技術要員の両方)。ソフトウェア品質保証グループ、ソフトウェア構成管理グループ、およびソフトウェアエンジニアリングプロセスグループなど、ソフトウェア関連作業を実施するグループは、ソフトウェアエンジニアリンググループには含まれない。

ソフトウェアエンジニアリングプロセスグループ [*software engineering process group*] - 組織で使用されるソフトウェアプロセスの定義、保守、および改善を促進する専門家のグループ。キープラクティスでは、一般化して『組織のソフトウェアプロセス活動に責任を持つグループ』と表現される。

ソフトウェアエンジニアリング要員[*software engineering staff*] - ソフトウェアタスクリーダーも含め、ソフトウェア技術に携わる人(例えば、アナリスト、プログラマ、エンジニア)であり、プロジェクトでソフトウェア開発や保守を実施する。ただしこれにマネージャは含まれない。

ソフトウェア開発計画[*software development plan*] - ソフトウェアプロジェクトのために実施される活動を記述した計画の集合。これは、ソフトウェアプロジェクトのためにソフトウェアエンジニアリンググループが実施する活動の管理を統御する。DOD-STD-2167AやIEEE-STD-1058などの同様の用語を使用するいかなる特定の計画策定標準の適用範囲に制限されるものではない。

ソフトウェア関連グループ[*software-related group*] - ソフトウェアエンジニアリングにおける機能を代表する個人の集まり(マネージャと技術要員の両方)であって、ソフトウェアの開発や保守をサポートするが、直接の責任は持たない。ソフトウェアエンジニアリング機能の例には、ソフトウェア品質保証とソフトウェア構成管理がある。

ソフトウェア計画[*software plans*] - 公式、非公式を問わず、ソフトウェア開発かつ/または保守の活動がどのように実施されるかを表現するために使用される計画の集まり。計画の例としては以下のものを含む：ソフトウェア開発計画、ソフトウェア品質保証計画、ソフトウェア構成管理計画、ソフトウェアテスト計画、リスク管理計画、およびプロセス改善計画。

ソフトウェア構成制御委員会[*software configuration control board*] - 構成アイテムに対する変更提案の評価と承認または不承認、および承認された変更の実装を確実なものにすることに責任を持つグループ。

ソフトウェア作業成果物[*software work product*] - ソフトウェアプロセスを定義し、維持し、または使用する際に作成されるあらゆるもののことであり、顧客やエンドユーザに納入されるものもあればそうでないものもある。ソフトウェア作業成果物には、プロセス記述、計画、手順、コンピュータプログラム、および関連する文書が含まれる。(比較のため『ソフトウェア成果物』を参照。)

ソフトウェア成果物[*software product*] - 顧客やエンドユーザーに納入することが指定されているコンピュータプログラム、手順書、および関連する文書とデータの一式、またはその中の幾つかのことである。[IEEE-STD-610] (比較のために、『ソフトウェア作業成果物』を参照。)

ソフトウェア統合*[software integration]* - 選択されたソフトウェアコンポーネントを組み立てるプロセスであって、最終バージョンのソフトウェアシステムが提供する能力の集合、または特定の部分集合を提供するためのプロセス。

ソフトウェア統合管理*[integrated software management]* - 「組織の標準ソフトウェアプロセス」と関連プロセス資産に基づいて、ソフトウェアのエンジニアリングと管理の活動を、一体として定義されたソフトウェアプロセスに統一し統合すること。

ソフトウェアに割り当てられたシステム要件*[system requirements allocated to software]* - システム要件の部分集合であり、システムのソフトウェアコンポーネントにおいて実装される部分を指す。割り当てられた要件はソフトウェア開発計画への主要なインプットである。ソフトウェア要件分析の結果、割り当てられた要件が詳細化され改良される。その結果、ソフトウェア要件として文書化される。

ソフトウェア能力審査*[software capability evaluation]* - 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。

ソフトウェアビルド*[software build]* - 実稼動バージョンのソフトウェアシステムまたはコンポーネントであって、最終バージョンのソフトウェアシステムまたはコンポーネントが提供する能力の特定の部分集合を取り入れたもの。
[IEEE-STD-610]

ソフトウェア品質管理*[software quality management]* - 顧客とエンドユーザのニーズと要望を満足するために、ソフトウェア成果物の品質目標を定義し、目標を達成するための計画を確立し、そして、ソフトウェア計画、ソフトウェア作業成果物、活動、および品質目標をモニターし補正するプロセス。

ソフトウェア品質保証*[software quality assurance]* - (1) 計画されていて系統的なパターンを持つ活動であって、ソフトウェア作業成果物が確立された技術要件に準拠しているという十分な確信が提供されるために必要なものすべて。(2) ソフトウェア作業成果物を開発かつ/または保守するプロセスを評価するために設計された活動の集合。

ソフトウェア品質目標*[software quality goal]* - ソフトウェア作業成果物に対して定義される定量的品質の目標。

ソフトウェアプロジェクト*[software project]* - 協調作業を必要とする仕事であって、システムのソフトウェアコンポーネントおよび関連文書を分析し、仕様化し、設計し、開発し、テストし、かつ/または保守することに重点を置くもの。ソフトウェアプロジェクトは、ハードウェア/ソフトウェアシステムを構築するプロジェクトの部分のこともある。

ソフトウェアプロセス*[software process]* - 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。

ソフトウェアプロセスアセスメント*[software process assessment]* - 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。

ソフトウェアプロセス改善計画*[software process improvement plan]* - ソフトウェアプロセスアセスメントの勧告から導出された計画であって、ソフトウェアプロセスを改善するために取られる具体的な処置を明らかにし、これらの処置を実施するための計画を概説する。時には処置計画ともいう。

ソフトウェアプロセス改善提案*[software process improvement proposal]* - プロセスまたはプロセス関連項目の変更に関する文書化された提案であって、ソフトウェアプロセス能力と実績を改善するもの（『処置提案』も参照。）

ソフトウェアプロセス関連文書*[software process-related documentation]* - 例となる文書および文書の断片であって、将来のプロジェクトが、「組織の標準ソフトウェアプロセス」をテーラリングする場合に役に立つと期待されるもの。例えば、「プロジェクトの定義されたソフトウェアプロセス」、標準、手順、ソフトウェア開発計画、計測計画、およびプロセストレーニング教材などがある。

ソフトウェアプロセス記述[*software process description*] - 主要なソフトウェアプロセスコンポーネントの作業面から見た定義であって、「プロジェクトの定義されたソフトウェアプロセス」または「組織の標準ソフトウェアプロセス」の中で明らかにされているもの。ソフトウェアプロセスの要件、設計、振舞い、または他の特徴を、完全で精確に検証可能な方法で文書化する。(『プロセス記述』も参照。)

ソフトウェアプロセス資産[*software process assets*] - (『組織のソフトウェアプロセス資産』を参照。)

ソフトウェアプロセス実績[*software process performance*] - (『プロセス実績』を参照。)

ソフトウェアプロセス成熟度[*software process maturity*] - 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首尾一貫性の両方を示している。

ソフトウェアプロセス能力[*software process capability*] - (『プロセス能力』を参照。)

ソフトウェアプロセス要素[*software process element*] - ソフトウェアプロセス記述の構成要素である。各プロセス要素は、整った形で定義され、境界が設定され、密接に関係したひと組のタスク(例えば、ソフトウェア見積もり要素、ソフトウェア設計要素、コーディング要素、ピアレビュー要素)を扱っている。プロセス要素の記述は、利用時に記入されるテンプレート、完成される断片、精緻化される抽象概念、あるいは修正をしてまたは修正せずに使用される完全な記述のいずれであってもよい。

ソフトウェアベースライン監査[*software baseline audit*] - ソフトウェアベースラインライブラリの構造、内容、および設備の検査であって、ベースラインがそのベースラインについて記述している文書に準拠することを検証する。

ソフトウェアベースラインライブラリ[*software baseline library*] - 構成アイテムと関連する記録を格納するリポジトリの内容。

ソフトウェアマネージャ*[software manager]* - ソフトウェアの開発かつ / または保守に対する直接の責任を持つプロジェクトまたは組織レベルのマネージャ。

ソフトウェア要件*[software requirement]* - ユーザが必要とするソフトウェアが、問題を解決するため、または目標を達成するために満たさなければならない条件または能力。 [IEEE-STD-610]

ソフトウェアライフサイクル*[software life cycle]* - ソフトウェア成果物が着想されたときから始まり、ソフトウェアがもうこれ以上使われなくなったときに終わるまでの期間のことである。典型的には、コンセプトフェーズ、要件フェーズ、設計フェーズ、実装フェーズ、テストフェーズ、設置と検収フェーズ、運用保守フェーズ、および場合によっては廃棄フェーズがソフトウェアライフサイクルに含まれる。 [IEEE-STD-610]

第一線ソフトウェアマネージャ*[first-line software manager]* - ソフトウェア技術者や他の関連スタッフで構成されるひとつの組織上の単位（例えば、部やプロジェクトチーム）における配員や活動に対して直接の管理責任（技術的統制の提供、人事と給与の機能の運営管理を含む）を持つマネージャ。

対象コンピュータ*[target computer]* - 納入されるソフトウェアがその上で稼働することが意図されているコンピュータ。（比較のため『ホストコンピュータ』を参照。）

タスク*[task]* - (1) 作業の基本単位として扱われる一連の指示。 [IEEE-STD-610] (2) ソフトウェアプロセスにおける整った形で定義された作業単位であり、プロジェクトの状況に対する可視的なチェックポイントを管理層に与える。タスクには、開始基準（前条件）と完了基準（後条件）がある。（比較のために、『活動』を参照。）

タスク立ち上げ会議*[task kick-off meeting]* - プロジェクトのタスクの開始時に開催される会議であって、タスクの活動の効果的な実施をそれに携わる個人に準備させるためのもの。

タスクリーダー*[task leader]* - 特定のタスクに関する技術チームのリーダーであって、技術的な責任を負い、そのタスクに携わる要員に対して技術的な指示を与える。

妥当性確認[*validation*] - 明記された要件を満足するか否かを判断するために、開発プロセスの途中または最後にソフトウェアを評価するプロセス。
[IEEE-STD-610]

チーム[*team*] - 多くの場合、様々なしかし関連したグループから引き抜かれ、整った形で定義された職務を組織またはプロジェクトのために遂行するように割り当てられた人の集まり。チームメンバが、そのチームの専任ではなく、他に主要な責務を持っていることもある。

追跡可能性[*traceability*] - 開発プロセスのふたつまたはそれ以上の成果物間の関係が確立されている度合。特に先行-後続関係、または主従関係を持つ成果物間の関係が確立されている度合。 [IEEE-STD-610]

定義されたソフトウェアプロセス[*defined software process*] - (『プロジェクトの定義されたソフトウェアプロセス』を参照。)

定義されたレベル[*defined level*] - (『成熟度レベル』を参照。)

定期的なレビュー/活動[*periodic review/activity*] - 特定の一定期間毎に行うレビューまたは活動。(比較のために、『イベント発生を契機とするレビュー/活動』を参照。)

定量的制御[*quantitative control*] - 定量的または統計に基づく技法であって、ソフトウェアプロセスの分析、ソフトウェアプロセスの実績変動の特殊原因の特定、およびソフトウェアプロセスの実績を整った形で定義された限界内に収めることに適したもの

テーラリングする[*tailor*] - プロセス、標準、または手順を修正し、プロセスまたは成果物の要件に対して、より調和させること。

手順[*procedure*] - 与えられたタスクを実施するために取られる一連の行為の記述。 [IEEE-STD-610]

テスト可能性[*testability*] - (1) システムまたはコンポーネントが、テスト基準の確立と、これらの基準を満足するか否かを判断するためのテストの実施を促進する度合。(2) 要件が、テスト基準の確立と、これらの基準を満足するか否かを判断するためのテストの実施を可能にするように明確に述べられている度合。 [IEEE-STD-610]

統合[*integration*] - (『ソフトウェア統合』を参照。)

(欠陥の) **特殊原因**[*special cause (of a defect)*] - 欠陥の原因で、ある一時的な状況に特有なものであり、プロセスに内在するものではないもの。特殊原因はプロセス実績においてランダムな変動(ノイズ)を与える。(比較のために、『共通原因』を参照。)

整った形で定義されたプロセス[*well-defined process*] - 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)検証機構、アウトプット、および完了基準を含むプロセス。(『効果的なプロセス』を参照。)

トレーニングする[*train*] - 専門的な教育と演習で熟練させること。(『オリエンテーション』も参照。)

トレーニンググループ[*training group*] - 組織のトレーニング活動を調整し手配する責任を負う個人(マネージャと要員の両方)の集まりである。典型的には、このグループが大半のトレーニングコースを準備し開催し、また他のトレーニング手段の利用に関する調整も行う。

トレーニングの免除[*training waiver*] - 特定の役割に必要なと指定されているトレーニングを、ある個人から免除する明文化された承認。免除は、その役割を実施するために必要なスキルをすでにその個人が持っている、客観的に判断される場合に認められる。

トレーニングプログラム[*training program*] - 組織のトレーニングのニーズを取り上げることに焦点を当てた関連する要素の集合。これには、組織のトレーニング計画、トレーニング教材、トレーニングの開発、トレーニングの開催、トレーニング設備、トレーニングの評価、およびトレーニング記録の維持を含む。

入札者[*bidder*] - 個人、法人や団体などで、ひとつまたはそれ以上の成果物を設計し、開発し、かつ/または製造するために、提案を提出しており、契約を獲得する候補者となっているもの。

能力成熟度モデル[*capability maturity model*] - ソフトウェア組織が、そのソフトウェアプロセスを定義し、実装し、計測し、制御し、そして改善するにたがって、進化していくステージの記述。このモデルは、プロセス改善戦略を選定するための指針であり、現状のプロセス能力の判断と、ソフトウェア品質とプロセス改善にもっとも重要な課題の特定を容易にする。

発注元 (外部発注元) [*prime contractor*] - ひとつあるいは複数の成果物を設計、開発、かつ/または製造するための外注契約を運営管理する個人、法人や団体など。

パレート分析[*Pareto analysis*] - 原因を最も重要なものから最も重要でないものへと順序付けすることによる欠陥の分析。パレート分析は、19世紀の経済学者Vilfredo Paretoにちなんで名付けられたもので、結果の大部分は比較的少数の原因から生ずるといふ原理に基づいている。すなわち、80%の結果は20%の原因に起因する。

反復できるレベル[*repeatable level*] - (『成熟度レベル』を参照。)

ピアレビュー[*peer review*] - ソフトウェア作業成果物のレビューであって、欠陥と改善項目を特定するために、その成果物の作成者の同僚により、定義された手順に従って行われる。

ピアレビューリーダー[*peer review leader*] - ピアレビューを計画し、組織化し、指導することについて、特にトレーニングを受けた適格な個人。

非技術要件[*nontechnical requirements*] - 合意、条件、かつ/または契約条項であって、ソフトウェアプロジェクトの管理活動に影響しそれを決定するもの。

必修トレーニング[*required training*] - 組織が特定の役割の遂行に必要であると指定するトレーニング。

標準[*standard*] - ソフトウェア開発の秩序ある均一な取り組みを規定するために用いられ、強制される、必須の要件。

標準ソフトウェアプロセス[*standard software process*] - (『組織の標準のソフトウェアプロセス』を参照。)

品質[*quality*] - (1) システム、コンポーネント、またはプロセスが明記された要件に合致する度合。(2) システム、コンポーネント、またはプロセスが、顧客またはユーザのニーズまたは期待に合致する度合。[IEEE-STD-610]

品質保証[*quality assurance*] - (『ソフトウェア品質保証』を参照。)

プロジェクト[*project*] - 協調作業を必要とする仕事であって、特定の成果物を開発しかつ/または保守することに焦点を当てる。成果物にはハードウェア、ソフトウェア、およびその他のコンポーネントが含まれる。プロジェクトは、独自の資金、費用会計、納入スケジュールを持つのが典型的である。

プロジェクトソフトウェアマネージャ[*project software manager*] - プロジェクトのすべてのソフトウェア活動に対する全責任を負う役割。プロジェクトソフトウェアマネージャは、プロジェクトマネージャがソフトウェアのコミットメントに関して任せた個人であり、プロジェクトのソフトウェアに関わるすべての資源を制御する。

プロジェクトの定義されたソフトウェアプロセス[*project's defined software process*] - プロジェクトによって使用されるソフトウェアプロセスの作業面から見た定義。「プロジェクトの定義されたソフトウェアプロセス」は、整った形で特徴付けられ理解されるソフトウェアプロセスであり、ソフトウェアの標準、手順、ツール、および手法の点から記述されている。これは、「組織の標準ソフトウェアプロセス」を、そのプロジェクト固有の特徴に合うようにテーラリングすることによって作成される。(『組織の標準ソフトウェアプロセス』、『効果的なプロセス』、『整った形で定義されたプロセス』も参照。)

プロジェクトマネージャ[*project manager*] - プロジェクト全体に対して事業上の全責任を持つ役割。ソフトウェア、あるいはハードウェア/ソフトウェアシステムを構築するプロジェクトを統制し、制御し、運営管理し、そして規制する個人。プロジェクトマネージャは、顧客に対して最終的な責任を負う個人である。

プロセス[*process*] - 与えられた目的のために実施される一連のステップ。例えば、ソフトウェア開発プロセス。[IEEE-STD-610]

プロセス開発[*process development*] - プロセスを定義し記述する行為。計画、アーキテクチャ構想、設計、実装、および妥当性確認を含むこともある。

プロセス記述[*process description*] - プロセスの主要コンポーネントの作業面から見た定義。完全で、精確で、検証可能な方法で、プロセスの要件、設計、振舞い、または他の特徴を明記した文書。これらの規定が満足されているか否かを判断するための手順を含むこともある。プロセス記述は、タスク、プロジェクト、または組織レベルで見出すことができる。

プロセス計測[*process measurement*] - プロセスを特徴付け理解する目的で、プロセスとその結果である成果物の計測値を採取するために使用される定義、手法、および活動の集合。

プロセス実績[*process performance*] - プロセスに従うことによって達成される実際の結果の尺度。（比較のために、『プロセス能力』を参照。）

プロセス実績ベースライン[*process performance baseline*] - あるプロセスに従うことによって達成される実際の結果の文書化された特徴付け。期待されたプロセス実績に対して実際のプロセス実績を比較するためのベンチマークとして使用される。プロセス実績ベースラインは、典型的には、プロジェクトレベルで確立されるが、初期のプロセス実績ベースラインは、通常、プロセス能力ベースラインから導き出される。（比較のために、『プロセス能力ベースライン』を参照。）

プロセスデータベース[*process database*] - 『組織のソフトウェアプロセスデータベース』を参照。

プロセステーラリング[*process tailoring*] - プロセス要素の詳細または他の不完全なプロセスの仕様を詳細化し、適合させ、かつ/または完成させることによりプロセス記述を生成する活動。プロジェクト特有の事業ニーズは、普通はプロセステーラリングの間に取り上げられる。

プロセス能力[*process capability*] - あるプロセスに従うことによって達成することができる期待される結果の範囲。（比較のために、『プロセス実績』を参照。）

プロセス能力ベースライン[*process capability baseline*] - 典型的な状況下で特定のプロセスに従うことにより、通常達成されるであろうと期待される結果の範囲についての文書化された特徴づけ。プロセス能力ベースラインは、典型的には、組織レベルで確立されている。（比較のために、『プロセス実績ベースライン』を参照。）

プロフィール*[profile]* - 計画または予想と、実際の比較。通常はグラフ形式であり、典型的にはある期間に渡る実際との比較。

文書化された手順*[documented procedure]* - (『手順』を参照。)

ベースライン*[baseline]* - 公式にレビューされ合意された仕様または成果物であって、それ以降、将来の開発の基盤として利用され、公式の変更制御手順を通じてのみ変更され得るもの。[IEEE-STD-610]

ベースライン管理*[baseline management]* - 構成管理において、技術的、管理的統制を適用することであって、文書とそれらの文書に対する変更を指定する。これらの文書は、ある構成アイテムのライフサイクルにおける特定の時期に、ベースラインを公式に特定し確立する。[IEEE-STD-610]

ベースライン構成管理*[baseline configuration management]* - 公式にレビューされ合意され、将来の開発の基盤として利用されるベースラインを確立すること。あるソフトウェア作業成果物、例えばソフトウェア設計やコードは、事前に定められた時点で確立されたベースラインを持ち、厳格な変更制御プロセスがこれらのアイテムに適用されるべきである。これらのベースラインは顧客とのやりとりの際に制御と安定をもたらす。(『ベースライン管理』も参照。)

ベンチマーク*[benchmark]* - 計測または比較が行えるように設定される標準。[IEEE-STD-610]

方針*[policy]* - 決断に影響を及ぼし、決定を下すために、組織やプロジェクトによって採用される指針となる原則。典型的には上級管理層により確立される。

方法論*[methodology]* - 手法、手順、および標準の集合であって、成果物の開発に向けたエンジニアリングアプローチの統合された総体を定義したもの。

保守*[maintenance]* - 納入後のソフトウェアシステムまたはコンポーネントを修正するプロセスで、故障を訂正し、性能や他の属性を改善し、または環境変更に適応するためのもの。[IEEE-STD-610]

ホストコンピュータ*[host computer]* - ソフトウェアの開発に使用されるコンピュータ。(比較のために、『対象コンピュータ』を参照。)

マイルストーン[*milestone*] - スケジュールされたイベントであって、ある個人に説明責任があり、進捗を計測するために使用される。

マネージャ[*manager*] - マネージャの責任範囲内でタスクまたは活動を実施する個人に対して、技術的および管理的な統制と制御を提供することを含む役割。マネージャの伝統的な機能には、責任範囲内において作業を計画し、資源を割り当て、組織化し、統制し、そして制御することが含まれる。

免除[*waiver*] - (『トレーニングの免除』を参照。)

役割[*role*] - 一人または複数の個人によって引き受けられる定義された責任の単位。

ユーザ[*user*] - (『エンドユーザ』を参照。)

ユニット[*unit*] - (1) コンピュータソフトウェアコンポーネントの設計で明記された単独にテスト可能な要素。(2) コンピュータプログラムの論理的に分離可能な部分。(3) それ以上他のコンポーネントに細分化されないソフトウェアコンポーネント。[IEEE-STD-610]

要員[*staff*] - ソフトウェア開発またはソフトウェア構成管理といった割り当てられた職務を遂行することに責任がある個人。タスクリーダーは含むがマネージャは含まない。

ライフサイクル[*life cycle*] - (『ソフトウェアライフサイクル』を参照。)

履行検証[*verifying implementation*] - (『コモンフィーチャ』を参照。)

リスク[*risk*] - 損失を被る可能性。

リスク管理[*risk management*] - 問題分析の取り組み方であって、リスク確率を用いて、ある状況におけるリスクを量り、伴うリスクをより正確に理解する。リスク管理は、リスクの特定、分析、優先順位付け、および制御を含む。

リスク管理計画[*risk management plan*] - プロジェクトで実施されるべきリスク管理活動を記述した計画の集合。

リスク対応係数[*contingency factor*] - 規模、コスト、またはスケジュールの調整（増加）部分であって、不完全な仕様、そのアプリケーションドメインの見積りにおける経験不足などのために、これらのパラメータにありがちな過小見積りに対応する部分。

割り当てられた要件[*allocated requirements*] - (『ソフトウェアに割り当てられたシステム要件』を参照。)

付録 C : キープラクティス抄録

この付録では、キープラクティスの抄録版を提供する。この抄録は、それぞれのキープロセスエリアに対して、SEIが規定した主要な活動のハイレベルな概要を提供するものである。それぞれのキープロセスエリアの“早見”としても活用できる。しかし、ここではキープラクティスに対する活動の詳細は提供しておらず、全てのキープラクティスをカバーしていかない。これは、情報提供の目的を意図したものであり、キープラクティスの遵守を判断するためあるいはプロセス改善の計画を立てるために利用することを意図したものではない。

この抄録に含まれるのは、キープロセスエリアの短い説明、ゴール、及びそのキープロセスエリアの「実施される活動」コモンフィーチャから抜き出したキープラクティスの記述である。これらの項目は、詳細なキープラクティスの一覧からそのままの形で抜粋したものである。

他のコモンフィーチャ（実施のコミットメント、実施能力、計測と分析、及び履行検証）の中に記述されている他の多くのキープラクティスは、この付録には含まれていない。これらのキープラクティスは、キープラクティスが適切にかつ効果的に実装され、堅実に確立され、維持されそして時を経ても損なわれず、そして新しい作業に効果的に適用できることを確実なものにするために存在しなければならない。キープロセスエリアを適切に確立するためには、キープラクティスの全てを用いるべきである。

「実施のコミットメント」に含まれる活動の典型は、組織的な方針と上級管理層の支持を確立することである。「実施能力」に含まれる典型は、資源、組織構造、およびトレーニングである。「計測と分析」に含まれる典型は、「実施される活動」の状態や有効性を判断するために取りうる計測値の例である。「履行検証」に含まれる典型は、管理層およびソフトウェア品質保証によるレビューと監査である。

レベル 2：要件管理

『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する共通理解を確立することである。

『要件管理』に必然的に含まれる活動には、ソフトウェアプロジェクトの要件に関し顧客との合意を確立すること、およびその合意を維持することが挙げられる。この合意を、『ソフトウェアに割り当てられたシステム要件』という。

『顧客』は、システムエンジニアリンググループ、マーケティンググループ、あるいは他の社内組織であったり、外部の顧客であったりする。合意は、技術的な要件と非技術的な要件（例えば、納期）の両方を扱う。合意は、ソフトウェアのライフサイクル全般にわたって、ソフトウェアプロジェクト活動の見積り、計画、実施、および進捗確認の基盤となる。

ソフトウェア、ハードウェア、およびその他のシステムコンポーネント（例えば、人間）へのシステム要件の割り当ては、ソフトウェアエンジニアリンググループ外のグループ（例えば、システムエンジニアリンググループ）が実施することがある。すなわち、ソフトウェアエンジニアリンググループがこの割り当てに直接的な制御権を持たないこともある。プロジェクトの制約条件のなかで、ソフトウェアエンジニアリンググループは、自身の担当する「ソフトウェアに割り当てられたシステム要件」の文書化と制御を確実なものにするため、適切な処置を講ずる。

この制御を実現するため、ソフトウェアエンジニアリンググループは、「ソフトウェアに割り当てられたシステム要件」の初期レビューおよび更新時レビューを実施する。レビューで判明したすべての課題は、ソフトウェアに割り当てられたシステム要件」をソフトウェアプロジェクトに組み込む前の段階で解決しておかねばならない。「ソフトウェアに割り当てられたシステム要件」に変更があった場合は、影響を受けるソフトウェア計画、作業成果物、および活動についても、更新された要件と首尾一貫するよう常に補正する。

要件管理のゴール：

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。
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。

要件管理におけるトップレベルの「実施される活動」：

1. ソフトウェアエンジニアリンググループは、割り当てられた要件がソフトウェアプロジェクトに組み込まれる前の段階でレビューする。
2. ソフトウェアエンジニアリンググループは、ソフトウェア計画、作業成果物、および活動の基盤として、割り当てられた要件を使用する。
3. 割り当てられた要件への変更をレビューし、ソフトウェアプロジェクトに組み込む。

レベル 2： 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。

『ソフトウェアプロジェクト計画』に必然的に含まれる活動には、実施すべき作業を見積ること、必要なコミットメントを確立すること、および作業を実施するための計画を定義することが挙げられる。

ソフトウェア計画は、実施すべき作業内容記述書、およびソフトウェアプロジェクトを定義しその境界を定めるその他の制約と目標（これらは『要件管理』キープロセスエリアのプラクティスによって確立される）から始まる。ソフトウェア計画プロセスには、ソフトウェア作業成果物の規模や必要な資源の見積り、スケジュールの作成、ソフトウェアリスクの特定とアセスメント、およびコミットメントの協議といったステップが含まれる。ソフトウェアプロジェクトの計画（ソフトウェア開発計画）を確立するには、これらのステップを繰り返し実施する必要があるかもしれない。

この計画は、ソフトウェアプロジェクト活動を実施し、管理するための基盤を提供するとともに、ソフトウェアプロジェクトの顧客に対しては、ソフトウェアプロジェクトの資源、制約、および能力に応じたコミットメントを明示する。

ソフトウェアプロジェクト計画のゴール：

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。
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。
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計画におけるトップレベルの「実施される活動」：

1. ソフトウェアエンジニアリンググループは、プロジェクト提案チームに参加する。
2. ソフトウェアプロジェクト計画の策定は、プロジェクト全体計画の早期段階から、かつ並行して開始する。
3. ソフトウェアエンジニアリンググループは、他の影響を受けるグループとともに、プロジェクトの全期間にわたってプロジェクト全体計画に参加する。
4. 上級管理層は、組織外の個人とグループに対してなされたソフトウェアプロジェクトのコミットメントを、文書化された手順に従ってレビューする。
5. あらかじめ定義された管理可能な大きさのステージを持つソフトウェアライフサイクルを特定または定義する。
6. プロジェクトのソフトウェア開発計画は、文書化された手順に従って策定する。
7. ソフトウェアプロジェクト計画を文書化する。
8. ソフトウェアプロジェクトの制御を確立し維持するために必要なソフトウェア作業成果物を特定する。
9. ソフトウェア作業成果物の規模（またはソフトウェア作業成果物の変更規模）の見積りは、文書化された手順に従って算出する。
10. ソフトウェアプロジェクトの工数とコストの見積りは、文書化された手順に従って算出する。
11. プロジェクトの重要なコンピュータ資源の見積りは、文書化された手順に従って算出する。
12. プロジェクトのソフトウェアスケジュールは、文書化された手順に従って策定する。
13. プロジェクトのコスト、資源、スケジュール、および技術面にかかわるソフトウェアのリスクを特定し、アセスメントし、そして文書化する。
14. プロジェクトのソフトウェアエンジニアリング設備および支援ツールに関する計画を作成する。
15. ソフトウェア計画策定に関するデータを記録する。

レベル 2： ソフトウェアプロジェクト進捗管理

『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。

『ソフトウェアプロジェクト進捗管理』に必然的に含まれる活動には、ソフトウェアの成果と結果について、文書化された見積り、コミットメント、および計画に照らして進捗の確認を行いレビューすること、および計画を実際の成果や結果を踏まえて補正することが挙げられる。

ソフトウェアプロジェクトの文書化された計画（即ち、『ソフトウェアプロジェクト計画』キープロセスエリアで記述するようなソフトウェア開発計画）が、ソフトウェア活動の進捗確認、状況伝達、および計画改訂の基盤として使用される。ソフトウェア活動は管理層によりモニターされる。進捗は、本来、選択されたソフトウェア作業成果物が完成したときおよび選択されたマイルストーンにおいて、実際のソフトウェアの規模、工数、コスト、およびスケジュールを計画と比較することにより判断される。ソフトウェアプロジェクトが計画通りに進んでいないと判断されたときは、是正処置がとられる。是正処置としては、実際の成果を反映させてソフトウェア開発計画を改訂すること、残っている作業を再計画すること、あるいは行動を改善するための処置をとることなどがある。

ソフトウェアプロジェクト進捗管理のゴール：

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。
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。
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。

ソフトウェアプロジェクト進捗管理におけるトップレベルの「実施される活動」：

1. ソフトウェア活動の進捗を確認し状況を伝達するために、文書化されたソフトウェア開発計画を使用する。
2. プロジェクトのソフトウェア開発計画は、文書化された手順に従って改訂する。
3. 上級管理層は、組織外のグループと個人に対するソフトウェアプロジェクトのコミットメントおよびコミットメントに対する変更を、文書化された手順に従ってレビューする。
4. 承認されたコミットメントの変更がソフトウェアプロジェクトに影響をおよぼす場合、その変更をソフトウェアエンジニアリンググループおよび他のソフトウェア関連グループメンバに伝達する。
5. ソフトウェア作業成果物の規模（あるいはその変更規模）についての進捗を確認し、必要に応じて是正処置をとる。
6. プロジェクトのソフトウェア工数とコストについての進捗を確認し、必要に応じて是正処置をとる。
7. プロジェクトの重要なコンピュータ資源の進捗を確認し、必要に応じて是正処置をとる。
8. プロジェクトのソフトウェアスケジュールについての進捗を確認し、必要に応じて是正処置をとる。
9. ソフトウェアエンジニアリング技術活動の進捗を確認し、必要に応じて是正処置をとる。
10. プロジェクトのコスト、資源、スケジュール、および技術面に関するソフトウェアリスクを確認する。
11. ソフトウェアプロジェクトの実計測データと再計画データを記録する。
12. ソフトウェアエンジニアリンググループは定期的に内部レビューを行い、技術面での進展、計画、実績、および課題をソフトウェア開発計画に照らして進捗を確認する。
13. ソフトウェアプロジェクトの成果と結果を取り上げるための公式レビューは、文書化された手順に従って、選定されたプロジェクトのマイルストーンで開催する。

レベル 2： ソフトウェア外注管理

『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』に必然的に含まれる活動には、ソフトウェア外注先を選定すること、外注先とコミットメントを確立すること、および外注先の行動と結果の進捗を確認しレビューすることが挙げられる。これらのプラクティスでは、ソフトウェア、ハードウェア、およびその他のシステム構成要素を包含するようなソフトウェア構成要素についての外注管理だけでなく、ソフトウェア（単独の）外注管理も扱う。

外注先は、その作業の実施能力を基準にして選定される。外部発注元（以下、本KPAでは「発注元」という）の作業の一部を外注する場合、その決定にはさまざまな要因が関係する。技術的な考慮だけでなく、戦略的な業務提携を基盤として外注先が選定されることもある。このキープロセスエリアのプラクティスでは、作業の定義された部分を他の組織へ外注することに関連した、従来からの調達プロセスについて取り上げる。

外注に当たっては、技術要件と非技術要件（例えば、納入日）を取り扱う文書化された合意が確立される。この合意は、外注管理の基盤として使用される。外注先で行われる作業とその作業に対する計画が文書化される。外注先が従うべき標準は、発注元の標準と適合しているものとする。

外注された作業に関してのソフトウェア計画および進捗管理の活動は、外注先が実施する。発注元は、これらの計画および進捗管理の活動が適切に実施されていること、そして外注先から納入されたソフトウェア成果物が検収基準を満足することを確実なものにする。発注元は、それらの成果物とプロセスのインタフェースの管理に関して、外注先と協力して作業をする。

ソフトウェア外注管理のゴール：

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。
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。
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。
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。

ソフトウェア外注管理におけるトップレベルの「実施される活動」：

1. 外注する作業は、文書化された手順に従って定義し計画する。
2. ソフトウェア外注先は、外注契約入札者の作業実施能力に関する評価を基盤とし、文書化された手順に従って選定する。
3. 発注元とソフトウェア外注先との間の契約合意を、外注管理の基盤とする。
4. 文書化された外注先のソフトウェア開発計画について、発注元がレビューし承認する。
5. 文書化され承認された外注先のソフトウェア開発計画を用いて、ソフトウェア活動の進捗を確認し、状況を伝達する。
6. ソフトウェア外注先の作業内容記述書、外注契約条項、およびその他のコミットメントに関する変更は、文書化された手順に従って解決する。
7. 発注元の管理層は、外注先の管理層とともに定期的な状況/調整レビューを開催する。

キープラクティス抄録

8. ソフトウェア外注先とともに、定期的に技術レビューと情報交換を行う。
9. 外注先のソフトウェアエンジニアリングの成果と結果を取り上げるための正式レビューは、選択されたマイルストーンで、文書化された手順に従って開催する。
10. 発注元のソフトウェア品質保証グループは、文書化された手順に従って外注先のソフトウェア品質保証の活動をモニターする。
11. 発注元のソフトウェア構成管理グループは、文書化された手順に従って外注先のソフトウェア構成管理活動をモニターする。
12. 発注元は、外注先のソフトウェア成果物納入の一環として、文書化された手順に従って検収テストを実施する。
13. ソフトウェア外注先の実績を定期的に評価し、評価結果は外注先とともにレビューする。

レベル 2： ソフトウェア品質保証

『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』に必然的に含まれる活動には、適用される手順や標準に準拠しているかを検証するためにソフトウェア成果物や活動をレビューしかつ監査すること、およびこれらのレビューや監査結果をソフトウェアプロジェクトマネージャならびに他の適切なマネージャへ提出することが挙げられる。

ソフトウェア品質保証グループは、ソフトウェアプロジェクトの初期段階から参加することにより、ソフトウェアプロジェクトに付加価値をもたらす、プロジェクトや組織の方針面での制約条件を満足する計画、標準、ならびに手順を確立する。ソフトウェア品質保証グループは、計画、標準、および手順の確立に関与し、それらがプロジェクトニーズに適するものとなることを助け、またソフトウェアライフサイクル全般にわたって行われるレビューや監査に使用可能なことを確実なものにする。ソフトウェア品質保証グループは、ライフサイクル全般にわたるプロジェクト活動をレビューし、ソフトウェア成果物の監査を行う。そして、ソフトウェアプロジェクトが自身で確立した計画、標準、および手順を忠実に守っているかどうかについての可視性を管理層へ提供する。

遵守に関係する課題は、ソフトウェアプロジェクト内で最初に取り上げられ、可能であればその場で解決される。ソフトウェアプロジェクト内部で解決できない課題については、ソフトウェア品質保証グループが解決に向け、適切な管理層レベルまで引き上げる。

このキープロセスエリアでは、ソフトウェア品質保証機能を実行するグループのプラクティスを扱う。ソフトウェア品質保証グループによりレビューや監査される特定の活動および作業成果物を明記するプラクティスは、通常、他のキープロセスエリアの『履行検証』コモンフィーチャの中にある。

ソフトウェア品質保証のゴール：

1. ソフトウェア品質保証活動が計画されている。
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。
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。
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。

ソフトウェア品質保証におけるトップレベルの「実施される活動」：

1. ソフトウェアプロジェクトのSQA計画は、文書化された手順に従って作成する。
2. SQAグループの活動は、SQA計画に従って実施する。
3. SQAグループは、プロジェクトのソフトウェア開発計画、標準、および手順の作成とレビューに参加する。
4. SQAグループは、ソフトウェアエンジニアリング活動をレビューし、その遵守状況を検証する。
5. SQAグループは、指定されたソフトウェア作業成果物を監査し、その遵守状況を検証する。
6. SQAグループは、ソフトウェアエンジニアリンググループに対し、定期的にその活動結果を報告する。
7. ソフトウェア活動やソフトウェア作業成果物について特定された逸脱事項は、文書化された手順に従って文書化し取り扱う。
8. SQAグループは、顧客のSQA担当者などとともに、適宜、SQAグループの活動や所見の定期的レビューを開催する。

レベル 2： ソフトウェア構成管理

『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』に必然的に含まれる活動には、所定の時点におけるソフトウェア構成（選択されたソフトウェア作業成果物とそれらの記述）を特定すること、その構成に関する変更を系統的に制御すること、およびソフトウェアのライフサイクル全般にわたって構成の一貫性と追跡可能性を維持することが挙げられる。ソフトウェア構成管理下におく作業成果物としては、顧客に納入されるソフトウェア成果物（例えば、ソフトウェア要件に関する文書やコード）、およびこれらのソフトウェア成果物と一緒に特定されるアイテムまたはこれらのソフトウェア成果物の作成に必要なアイテム（例えば、コンパイラ）がある。

ソフトウェアベースラインライブラリは、ソフトウェアベースラインの開発に伴いそれを含む形で確立される。ベースラインの変更やソフトウェアベースラインライブラリから作られるソフトウェア成果物のリリースについては、ソフトウェア構成管理の機能である変更制御や構成監査をとおして系統的な制御を行う。

このキープロセスエリアでは、ソフトウェア構成管理機能を実施するためのプラクティスを扱う。特定の構成アイテム/ユニットを明らかにするプラクティスは、それぞれの構成アイテム/ユニットの開発と保守について述べているキープロセスエリアの中に含まれる。

ソフトウェア構成管理のゴール：

1. ソフトウェア構成管理活動が計画されている。
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。
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。
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。

ソフトウェア構成管理におけるトップレベルの「実施される活動」：

1. SCM計画は、文書化された手順に従ってソフトウェアプロジェクト個別に作成する。
2. 文書化され承認されたSCM計画を、SCM活動実施の基盤として使用する。
3. ソフトウェアベースラインのリポジトリとして、構成管理ライブラリシステムを構築する。
4. 構成管理下におくべきソフトウェア作業成果物を特定する。
5. すべての構成アイテム/ユニットの変更要求や問題報告は、文書化された手順に従って発行し、記録し、レビューし、承認し、そして進捗を確認する。
6. ベースラインの変更は、文書化された手順に従って制御する。
7. 文書化された手順に従って、ソフトウェアベースラインライブラリから成果物を生成し、それらのリリースを制御する。
8. 構成アイテム/ユニットの状況は、文書化された手順に従って記録する。
9. SCM活動およびソフトウェアベースラインの内容に関する標準報告書を作成し、影響を受けるグループと個人に利用可能とする。
10. ソフトウェアベースライン監査は、文書化された手順に従って実施する。

レベル 3： 組織プロセス重視

『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』に必然的に含まれる活動には、組織のソフトウェアプロセスとプロジェクトのソフトウェアプロセスに関する理解を深めこれを維持すること、ソフトウェアプロセスのアセスメント、開発、維持、および改善に関する調整を行うことが挙げられる。

組織は、長期的なコミットメントと資源を提供して、現在と未来のソフトウェアプロジェクトにわたり、ソフトウェアエンジニアリングプロセスグループのようなグループによるソフトウェアプロセスの開発と維持を調整する。このグループは、組織のソフトウェアプロセス活動に責任をもつ。このグループは、特に「組織の標準ソフトウェアプロセス」および（『組織プロセス定義』キープロセスエリアで記述されているような）関連のプロセス資産の開発と維持に責任を負い、そしてソフトウェアプロジェクトとともにプロセス活動を調整する。

組織プロセス重視のゴール：

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。
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。
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。

組織プロセス重視におけるトップレベルの「実施される活動」：

1. ソフトウェアプロセスを定期的にあセスメントし、アセスメントの所見を取り上げるために処置計画を策定する。
2. 組織は、ソフトウェアプロセスの開発と改善活動に関する計画を策定し保守する。
3. ソフトウェアプロセスの開発と改善に関して、組織やプロジェクトの活動を組織レベルで調整する。
4. 「組織のソフトウェアプロセスデータベース」の使用について、組織レベルで調整する。
5. 組織で限定的に使用される新しいプロセス、手法、およびツールを、モニターし、評価し、そして適切であれば組織の他の部分に移転させる。
6. 組織やプロジェクトのソフトウェアプロセスに関するトレーニングを、組織横断的に調整する。
7. 組織やプロジェクトのソフトウェアプロセスの開発と改善に関する活動について、ソフトウェアプロセスの実装に携わるグループに情報を伝える。

レベル 3： 組織プロセス定義

『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。

『組織プロセス定義』に必然的に含まれる活動には、「組織の標準ソフトウェアプロセス」およびこれに関連するソフトウェア資産を開発すること、そしてそれらを維持することが挙げられる。プロセス資産とは、ソフトウェアライフサイクルの記述、プロセステーラリングのガイドラインと基準、「組織のソフトウェアプロセスデータベース」、および「ソフトウェアプロセス関連文書のライブラリ」のことをいう。

これらの資産の収集には、その組織における『組織プロセス定義』の実装の仕方によりさまざまな方法がある。例えば、ソフトウェアライフサイクルの記述は「組織の標準ソフトウェアプロセス」を構成するために必要な部分であることがあり、また「ソフトウェアプロセス関連文書のライブラリ」の一部は「組織のソフトウェアプロセスデータベース」に蓄積されることがある。

「組織のソフトウェアプロセス資産」は、「プロジェクトの定義されたソフトウェアプロセス」を開発し、履行し、そして維持するのに用いることができる。（「プロジェクトの定義されたソフトウェアプロセス」の開発および維持に関連するプラクティスは、『ソフトウェア統合管理』キープロセスエリアに記述されている。）

組織プロセス定義のゴール：

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。
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。

組織プロセス定義におけるトップレベルの「実施される活動」：

1. 文書化された手順に従って、「組織の標準ソフトウェアプロセス」を開発し維持する。
2. 「組織の標準ソフトウェアプロセス」は、確立された組織標準に従って文書化する。
3. プロジェクトにおいて使用することが承認されたソフトウェアライフサイクルの記述を文書化し維持する。
4. プロジェクトが「組織の標準ソフトウェアプロセス」をテーラリングするためのガイドラインおよび基準を開発し維持する。
5. 「組織のソフトウェアプロセスデータベース」を確立し維持する。
6. 「ソフトウェアプロセス関連文書のライブラリ」を確立し維持する。

レベル 3： トレーニングプログラム

『トレーニングプログラム』 キープロセスエリアの目的は、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。

『トレーニングプログラム』に必然的に含まれる活動には、まず組織、プロジェクト、および個人のトレーニングニーズを特定すること、次にそのニーズに対応するトレーニングを開発または調達することが挙げられる。

各ソフトウェアプロジェクトは、現在および将来必要となるスキルを評価し、そのスキルをどのようにして獲得するかを決定する。スキルの中には、非公式な手段（例えば、OJTや非公式なメンタリング(訳注：先輩による指導)など）によって効果的かつ効率的に供与されるものもある。一方、効果的かつ効率的に供与するには、公式のトレーニング手段（例えば、教室形式トレーニングや指導付き自習など）が必要となるスキルもある。適切な手段を選んで用いる。

このキープロセスエリアでは、トレーニングの機能を実施するグループに関するプラクティスを扱っている。特定のトレーニング題材（必要とされる知識やスキル）を明らかにするプラクティスは、個々のキープロセスエリアの実施能力コモンフィーチャに含まれる。

トレーニングプログラムのゴール：

1. トレーニング活動が計画されている。
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。
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。

トレーニングプログラムにおけるトップレベルの「実施される活動」：

1. 各ソフトウェアプロジェクトは、そのトレーニングニーズを明記するトレーニング計画を策定し保守する。
2. 文書化された手順に従って、組織のトレーニング計画を策定し改訂する。
3. 組織のトレーニング計画に従って、組織のトレーニングを実施する。
4. 組織レベルで準備するトレーニングコースは、組織標準に従って開発し保守する。
5. 任命された役割を遂行するのに必要な知識とスキルが、すでに個人の身に付いているかどうかを判断するために、必修トレーニングに対する免除手順を確立し使用する。
6. トレーニング記録を維持する。

レベル 3： ソフトウェア統合管理

『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。

『ソフトウェア統合管理』に必然的に含まれる活動には、「プロジェクトの定義されたソフトウェアプロセス」を開発すること、およびこの定義されたソフトウェアプロセスを用いてソフトウェアプロジェクトを管理することが挙げられる。「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」をもとにプロジェクト固有の特徴に合わせてテラリングされる。

ソフトウェア開発計画は、「プロジェクトの定義されたソフトウェアプロセス」に基づいて、どのように「プロジェクトの定義されたソフトウェアプロセス」の活動が履行され管理されるかということを記述したものである。ソフトウェアプロジェクトの規模、工数、コスト、スケジュール、配員、およびその他の資源の管理は、「プロジェクトの定義されたソフトウェアプロセス」のタスクに結び付けられる。

「プロジェクトの定義されたソフトウェアプロセス」は、すべて「組織の標準ソフトウェアプロセス」からテラリングされる。したがって、プロセスデータや教訓は、複数のソフトウェアプロジェクトで共有することができる。

ソフトウェアプロジェクトの見積り、計画、および進捗の確認に関する基本的なプラクティスについては、『ソフトウェアプロジェクト計画』および『ソフトウェアプロジェクト進捗管理』キープロセスエリアに記述されている。この2つのキープロセスエリアでは、問題が発生したときにそれを認識し、その問題を取り上げるために計画や行動を補正することに重点が置かれている。『ソフトウェア統合管理』キープロセスエリアのプラクティスは、この2つのキープロセスエリアのプラクティスを基にして、追加されたものである。『ソフトウェア統合管理』キープロセスエリアでは、問題の想定、およびこれらの問題による影響の予防あるいは最小化の活動に重点を移している。

ソフトウェア統合管理のゴール：

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」をテーラリングしたものである。
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。

ソフトウェア統合管理におけるトップレベルの「実施される活動」：

1. 文書化された手順に従って「組織の標準ソフトウェアプロセス」をテーラリングすることにより、「プロジェクトの定義されたソフトウェアプロセス」を開発する。
2. それぞれの「プロジェクトの定義されたソフトウェアプロセス」は、文書化された手順に従って改訂する。
3. プロジェクトのソフトウェア開発計画は、「プロジェクトの定義されたソフトウェアプロセス」の使用を記述するものであり、文書化された手順に従って策定し改訂する。
4. 「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、ソフトウェアプロジェクトを管理する。
5. 「組織のソフトウェアプロセスデータベース」は、ソフトウェアの計画と見積りに使用する。
6. ソフトウェア作業成果物の規模（またはソフトウェア作業成果物の変更規模）は、文書化された手順に従って管理する。
7. プロジェクトのソフトウェア工数とコストは、文書化された手順に従って管理する。
8. プロジェクトの重要なコンピュータ資源は、文書化された手順に従って管理する。
9. プロジェクトのソフトウェアスケジュールの重要な依存関係とクリティカルパスは、文書化された手順に従って管理する。
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従って特定し、アセスメントし、文書化し、そして管理する。
11. ソフトウェアプロジェクトのレビューを定期的実施し、ソフトウェアプロジェクトの実際の行動や結果を事業、顧客、およびエンドユーザなどの現在および予測されるニーズに適宜調和させるのに必要な処置を決定する。

レベル 3： ソフトウェアプロダクトエンジニアリング

『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』に必然的に含まれる活動には、「プロジェクトの定義されたソフトウェアプロセス」（『ソフトウェア統合管理』キープロセスエリアで記述されている）と適切な手法やツールを用いて、ソフトウェアを構築し保守するためのエンジニアリングタスクを実施することが挙げられる。

ソフトウェアエンジニアリングタスクには、「ソフトウェアに割り当てられたシステム要件」の分析、ソフトウェア要件の開発、ソフトウェアアーキテクチャの開発、ソフトウェアの設計、ソフトウェアのコードへの実装、ソフトウェアコンポーネントの統合、およびソフトウェアのテストが含まれる。なお、システム要件については『要件管理』キープロセスエリアで記述されている。ソフトウェアのテストは、明記された要件（「ソフトウェアに割り当てられたシステム要件」とソフトウェア要件）をソフトウェアが満足していることを検証するために行う。

ソフトウェアエンジニアリングタスクの実施に必要な文書（例えば、ソフトウェア要件文書、ソフトウェア設計文書、テスト計画、およびテスト手順書）については、各タスクが先行タスクの結果を処理し、生成された結果が後続タスク（ソフトウェアの運用保守タスクを含む）に対して適切であることが確実なものとなるように、それらの文書を作成しレビューする。変更が承認されると、それを反映するように、影響を受けるソフトウェア作業成果物、計画、コミットメント、プロセス、および活動が改訂される。

ソフトウェアプロダクトエンジニアリングのゴール：

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。
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。

ソフトウェアプロダクトエンジニアリングにおけるトップレベルの「実施される活動」：

1. 適切なソフトウェアエンジニアリングの手法とツールを、「プロジェクトの定義されたソフトウェアプロセス」に統合する。
2. 「プロジェクトの定義されたソフトウェアプロセス」に従って割り当てられた要件を系統的に分析することにより、ソフトウェア要件を開発し、保守し、文書化し、そして検証する。
3. ソフトウェア要件に適応し、コーディングに対する枠組みを形成するため、「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、ソフトウェア設計を開発し、保守し、文書化し、そして検証する。
4. ソフトウェア要件とソフトウェア設計を実装するため、「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、ソフトウェアコードを開発し、保守し、文書化し、そして検証する。
5. 「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、ソフトウェアテストを実施する。
6. 「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、ソフトウェアの統合テストを計画し実施する。
7. ソフトウェアのシステムテストと検収テストを計画し実施して、ソフトウェアが要件を満足していることを実証する。
8. 「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、ソフトウェアの運用と保守に使用する文書を作成し保守する。
9. 「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、ピアレビューやテストで特定された欠陥のデータを収集し分析する。
10. ソフトウェア作業成果物全体をとおして首尾一貫性を保つ。ソフトウェア作業成果物には、ソフトウェア計画、プロセス記述、割り当てられた要件、ソフトウェア要件、ソフトウェア設計、コード、テスト計画、およびテスト手順が含まれる。

レベル 3： グループ間調整

『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』に必然的に含まれる活動には、ソフトウェアエンジニアリンググループが他のプロジェクトのエンジニアリンググループと連携して、システムレベルの要件、ねらい、および課題を取り上げることが挙げられる。プロジェクトの中のエンジニアリンググループの代表者は、顧客やエンドユーザなどと適宜作業を行い、システムレベルの要件、ねらい、および計画を確立するために連携する。これらの要件、ねらい、および計画は、すべてのエンジニアリング活動の基盤となる。

グループ間における技術活動のインターフェースとやり取りは、システム全体の品質と一貫性を確実なものにするため、計画され管理される。すべてのエンジニアリンググループがすべてのグループの状況や計画を認識し、そしてシステムとグループ間の課題が適切に注目されることを確実なものとするため、プロジェクトのエンジニアリンググループの代表者間で定期的に技術レビューと意見交換会が開催される。

これらのエンジニアリングタスクに関連するプラクティスでソフトウェア特有のものは、『要件管理』および『ソフトウェアプロダクトエンジニアリング』キープロセスエリアに記述されている。

グループ間調整のゴール：

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。
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。
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。

グループ間調整におけるトップレベルの「実施される活動」：

1. ソフトウェアエンジニアリンググループとその他のエンジニアリンググループは、顧客やエンドユーザなどと適宜連携し、システム要件を確立する。
2. プロジェクトのソフトウェアエンジニアリンググループの代表者は、他のエンジニアリンググループの代表者とともに、技術活動をモニターしかつ調整し、そして技術的な課題を解決する。
3. グループ間のコミットメントの伝達、および実施される作業の調整と進捗の確認のため、文書化された計画を使用する。
4. エンジニアリンググループ間の重要な依存関係は、文書化された手順に従って特定し、協議し、そして進捗を確認する。
5. 他のエンジニアリンググループへのインプットとなる作業成果物は、その作業成果物が受け入れ側のエンジニアリンググループのニーズを満足することを確実なものにするため、受け入れ側グループの代表者がレビューする。
6. 個々のプロジェクトエンジニアリンググループの代表者で解決できないグループ間の課題は、文書化された手順に従って取り扱う。
7. プロジェクトのエンジニアリンググループの代表者は、定期的に技術レビューと意見交換会を開催する。

レベル 3：ピアレビュー

『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効果がある。

『ピアレビュー』に必然的に含まれる活動には、欠陥や変更が必要な領域を特定するため、ソフトウェア作業成果物を作成者の同僚が手法に則って検査することが挙げられる。ピアレビューを受ける特定のソフトウェア成果物は、『ソフトウェア統合管理』キープロセスエリアに記述されているように、「プロジェクトの定義されたソフトウェアプロセス」の中で明らかにされ、ソフトウェアプロジェクト計画活動の中でスケジュールが設定される。

このキープロセスエリアでは、ピアレビューの実施に関するプラクティスを取り上げる。ピアレビューを受ける特定のソフトウェア作業成果物を明らかにするプラクティスは、各ソフトウェア作業成果物の開発と保守を記述するキープロセスエリアに含まれる。

ピアレビューのゴール：

1. ピアレビュー活動が計画されている。
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。

ピアレビューにおけるトップレベルの「実施される活動」：

1. ピアレビューを計画し、計画を文書化する。
2. 文書化された手順に従って、ピアレビューを実施する。
3. ピアレビューの開催と結果に関するデータを記録する。

レベル 4： 定量的プロセス管理

『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。

『定量的プロセス管理』に必然的に含まれる活動には、『ソフトウェア統合管理』キープロセスエリアで述べられている「プロジェクトの定義されたソフトウェアプロセス」の実績に対する目標を確立すること、プロセス実績の計測を行なうこと、それらの計測値を分析すること、および許容限界内でプロセス実績を維持するための補正を行うことが挙げられる。プロセス実績が許容限界内で安定化した時点で、「プロジェクトの定義されたソフトウェアプロセス」、関連する計測値、およびその計測値の許容限界は、ベースラインとして確立され、定量的にプロセス実績を制御するために用いられる。

組織は、ソフトウェアプロジェクトからプロセス実績データを収集し、「組織の標準ソフトウェアプロセス」のプロセス能力（新しいプロジェクトが達成すると期待できるプロセス実績）の特性を示すためにこれらのデータを用いる。「組織の標準ソフトウェアプロセス」は、『組織プロセス定義』キープロセスエリアで記述されている。プロセス能力は、ソフトウェアプロセスに従った際に期待される結果（組織が着手する次のソフトウェアプロジェクトで期待されるもっともあり得そうな成果）の範囲を記述する。次に、ソフトウェアプロジェクトは、これらのプロセス能力データを用いてプロセス実績に対する目標を確立し改訂し、そして「プロジェクトの定義されたソフトウェアプロセス」の実績を分析する。

定量的プロセス管理のゴール：

1. 定量的プロセス管理の活動が計画されている。
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。
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。

定量的プロセス管理におけるトップレベルの「実施される活動」：

1. 定量的プロセス管理に対するソフトウェアプロジェクト計画は、文書化された手順に従って策定する。
2. ソフトウェアプロジェクトの定量的プロセス管理活動は、プロジェクトの定量的プロセス管理計画に従って実施する。
3. 実施するデータ収集と定量的分析の戦略は、「プロジェクトの定義されたソフトウェアプロセス」に基づいて決定する。
4. 「プロジェクトの定義されたソフトウェアプロセス」を定量的に制御するために用いられる計測データは、文書化された手順に従って収集する。
5. 「プロジェクトの定義されたソフトウェアプロセス」は、文書化された手順に従って分析し、定量的な制御下に置く。
6. ソフトウェアプロジェクトの定量的プロセス管理活動の結果について文書化した報告書を作成し、配布する。
7. 「組織の標準ソフトウェアプロセス」に対するプロセス能力ベースラインは、文書化された手順に従って確立され維持される。

レベル 4： ソフトウェア品質管理

『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』に必然的に含まれる活動には、ソフトウェア成果物の品質目標を定義すること、これらの目標を達成するための計画を確立すること、および顧客とエンドユーザの高品質な成果物に対するニーズと要望を満足させるため、ソフトウェア計画、ソフトウェア作業成果物、活動、および品質目標をモニターし補正を行うことが挙げられる。

『ソフトウェア品質管理』のプラクティスは、『ソフトウェア統合管理』と『ソフトウェアプロダクトエンジニアリング』、および『定量的プロセス管理』キープロセスエリアのプラクティスの上に構築される。『ソフトウェア統合管理』と『ソフトウェアプロダクトエンジニアリング』は「プロジェクトの定義されたソフトウェアプロセス」を確立し実装する。『定量的プロセス管理』は望まれる結果を達成するために「プロジェクトの定義されたソフトウェアプロセス」の能力の定量的な理解を確立する。

定量的な目標は、組織、顧客、およびエンドユーザのニーズに基づき、ソフトウェア成果物に対して確立される。組織はこれらの定量的な目標を達成すべく戦略と計画を確立し、そしてプロジェクトは品質目標を達成すべく「プロジェクトの定義されたソフトウェアプロセス」を具体的に補正する。

ソフトウェア品質管理のゴール：

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。
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。
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。

ソフトウェア品質管理におけるトップレベルの「実施される活動」：

1. プロジェクトのソフトウェア品質計画は、文書化された手順に従って策定し保守する。
2. プロジェクトのソフトウェア品質計画が、プロジェクトのソフトウェア品質管理活動の基礎である。
3. ソフトウェアライフサイクルを通じて、ソフトウェア成果物に対するプロジェクトの定量的品質目標を、定義し、モニターし、そして改訂する。
4. プロジェクトのソフトウェア成果物の品質は、イベント発生を契機として計測し、分析し、そして成果物の定量的品質目標との比較を行う。
5. 成果物に対するソフトウェアプロジェクトの定量的品質目標は、そのプロジェクトにソフトウェア成果物を納入する外注先に適切に割り当てる。

レベル 5： 欠陥予防

『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。

『欠陥予防』に必然的に含まれる活動には、過去に遭遇した欠陥を分析すること、および将来同じタイプの欠陥が発生することを予防するために特定の処置を行うことが挙げられる。欠陥は、現在のプロジェクトの先行ステージまたは先行タスクと同様に他のプロジェクトにおいて特定されている場合もある。欠陥予防活動もまた、教訓をプロジェクト間で広めるメカニズムのひとつである。

遭遇した欠陥のタイプを確認し、再発しそうな欠陥を特定するために、傾向を分析する。「プロジェクトの定義されたソフトウェアプロセス」とそれがどのように履行されたか(『ソフトウェア統合管理』と、『ソフトウェアプロダクトエンジニアリング』のキープロセスエリアに記述されているように)の理解に基づいて、欠陥の根本原因を決定し、将来の活動に対する欠陥の影響を見定める。

欠陥の再発を予防するため、プロジェクトおよび組織の双方は、特定の処置を行う。組織的処置は、『プロセス変更管理』のキープロセスエリアに記述されているように扱われるものもある。

欠陥予防のゴール：

1. 欠陥予防活動が計画されている。
2. 欠陥の共通原因が探し出され、特定されている。
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。

欠陥予防におけるトップレベルの「実施される活動」：

1. ソフトウェアプロジェクトは、欠陥予防活動のための計画を策定し保守する。
2. ソフトウェアタスクの開始時に、そのタスクを実行するチームのメンバはソフトウェアタスクの活動の準備と、関連する欠陥予防活動の準備について打ち合わせる。
3. 文書化された手順に従って原因分析会議を開催する。
4. 欠陥予防活動を調整するために割り当てられた各チームは、原因分析会議から出される処置提案の履行について、レビューし調整するために、定期的に会議を行う。
5. 欠陥予防データを文書化し、欠陥予防活動を調整しているチーム間で横断的に進捗を確認する。
6. 欠陥予防処置に伴う「組織の標準ソフトウェアプロセス」の改訂は、文書化された手順に従って組み込む。
7. 欠陥予防処置に伴う「プロジェクトの定義されたソフトウェアプロセス」の改訂は、文書化された手順に従って組み込む。
8. 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に関連したグループのメンバは、組織とプロジェクトの欠陥予防活動の状況と結果について、フィードバックを定期的に受ける。

レベル 5： 技術変更管理

『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、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』に必然的に含まれる活動には、新しい技術を特定し、選択し、そして評価すること、および効果的な技術を組織に取り入れることが挙げられる。そのねらいは、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。

組織は、ソフトウェアプロジェクトと作業するグループ（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、技術支援グループ）を確立し、新しい技術を導入、評価し、また既存技術の変更を管理する。特に、「組織の標準ソフトウェアプロセス」（『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）の能力を改善するであろう技術の変更を重視する。

常にソフトウェア関連の技術革新に注意をはらい、系統的にそれら进行评估、実験することにより、組織は適切な技術を選択し、ソフトウェア品質とソフトウェア活動の生産性を改善する。正規のプラクティスに取り入れる前に、試行を実施し、新しくて証明のない技術をアセスメントする。組織の管理層の適切な主催により、選択された技術を「組織の標準ソフトウェアプロセス」および現在のプロジェクトに適宜組み込む。

これらの技術変更の結果から生じる「組織の標準ソフトウェアプロセス」（『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）と「プロジェクトの定義されたソフトウェアプロセス」（『ソフトウェア統合管理』キープロセスエリアに記述されている）への変更は、『プロセス変更管理』キープロセスエリアに記述されているように扱われる。

技術変更管理のゴール：

1. 技術変更の導入が計画されている。
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。
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。

技術変更管理におけるトップレベルの「実施される活動」：

1. 組織は、技術変更管理計画を策定し保守する。
2. 組織の技術変更管理活動に責任を持つグループは、ソフトウェアプロジェクトと作業し技術変更領域を特定する。
3. ソフトウェアマネージャと技術要員に、常に新技術の情報を通知する。
4. 組織の技術変更管理に責任を持つグループは、「組織の標準ソフトウェアプロセス」を系統的に分析し、新技術を必要としたり、それにより利益を得る可能性がある領域を特定する。
5. 文書化された手順に従って、組織およびソフトウェアプロジェクトのための技術を選択し獲得する。
6. 新技術を正規のプラクティスとして導入する前に、適宜、技術を改善するための試行作業を行う。
7. 文書化された手順に従って、「組織の標準ソフトウェアプロセス」に適切な新技術を組み込む。
8. 文書化された手順に従って、「プロジェクトの定義されたソフトウェアプロセス」に適切な新技術を組み込む。

レベル 5： プロセス変更管理

『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』に必然的に含まれる継続的な活動には、プロセス改善目標の定義をすること、上級管理層の主催により、先を見越してかつ系統的に、「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」への改善を特定し、評価し、そして実装することが挙げられる。

組織に属する人たち全員のプロセス改善活動への参加を可能にし奨励するため、トレーニングプログラムと奨励プログラムが確立される。組織への潜在的な見返りに対し、改善の機会が特定され評価される。プロセス変更をアセスメントするために、正規のプラクティスに組み込まれる前に試行作業が実施される。

ソフトウェアプロセスの各改善が正規のプラクティスに対して承認されたとき、「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」は適宜改訂される。「組織の標準ソフトウェアプロセス」を改訂するプラクティスは『組織プロセス定義』キープロセスエリアにあり、「プロジェクトの定義されたソフトウェアプロセス」を改訂するプラクティスは『ソフトウェア統合管理』キープロセスエリアにある。

プロセス変更管理のゴール：

1. 継続的なプロセス改善が計画されている。
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。
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。

プロセス変更管理におけるトップレベルの「実施される活動」：

1. ソフトウェアプロセス改善プログラムを確立して、組織のメンバに組織のプロセスを改善する権限を与える。
2. 組織のソフトウェアプロセス活動に関して責任のあるグループ（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）は、ソフトウェアプロセス改善活動を調整する。
3. 組織は、文書化された手順に従って、ソフトウェアプロセス改善の計画を策定し保守する。
4. ソフトウェアプロセス改善活動は、ソフトウェアプロセス改善計画に従って実施する。
5. ソフトウェアプロセス改善提案は、文書化された手順に従って取扱われる。
6. 組織のメンバは、割り当てられたプロセスエリアでソフトウェアプロセス改善策を作成するため、積極的にチームに参加する。
7. ソフトウェアプロセス改善策を適宜、試行的に導入して、正規のプラクティスに組み込む前に利点や有効性を判断する。
8. 改善策を正規のプラクティスとして組み込むことを決定した場合、文書化された手順に従って履行する。
9. ソフトウェアプロセス改善活動の記録を維持する。
10. ソフトウェアのマネージャや技術要員は、イベント発生を契機として、ソフトウェアプロセス改善活動の状況と結果のフィードバックを受ける。

キープラクティス抄録

付録D：変更履歴

日付	版	変更内容
1990年3月1日	0.0	キープラクティス表の素案；CMMユーザのワーキンググループに配布された。1990年3月に開催されたCMMワークショップでのレビュー用。
1990年5月1日	0.1	1990年3月のCMMワークショップにおける第0.0版への勧告を取り込むための、キープラクティス表付録の素案。SEI内部でのピアレビュー用。
1990年6月6日	0.2	草案。第0.1版に対するSEIでのピアレビューで出されたコメントを取り込んだ、キープラクティス表の改訂版。CMMユーザワーキンググループにレビューとコメントを求めるために配布された。(キープラクティス表のベースラインとなった版)
1991年2月26日	0.3	草案。キープラクティス表の改訂版(レベル2のキーププロセスエリアのみ)。 CMMユーザワーキンググループと「質問票助言委員会 [Questionnaire Advisory Board]」による第0.2版へのコメントを反映させている。 SEI内部でのピアレビュー用。SEI内部でのレベル2のキーププロセスエリアとキープラクティス表のピアレビューを支援するために、概要と定義の(最初の版の)素案を付加。
1991年3月18日	0.4	草案。SEI内部での第0.3版に対するピアレビューでのコメントを反映したレベル2のキーププロセスエリアとキープラクティス表の改訂版。 CMMユーザワーキンググループと質問票助言委員会によるレビューとコメントを求めるために配布された。両組織によるレベル2のキーププロセスエリアとキープラクティス表のレビューを支援するために、概要と定義の(最初の版の)素案を付加。

変更履歴

1991年4月10日	0.41	草案。キープラクティス表の改訂版(レベル3のキープロセスエリアのみ。ただし、ソフトウェアプロダクトエンジニアリングのキープロセスエリアは含んでいない)。CMMユーザーキンググループと質問票助言委員会からの第0.2版へのコメントを反映した版でSEI内部でのピアレビュー用。
1991年4月22日	0.5	草案。キープラクティス表の改訂版(レベル3のキープロセスエリアのみ。ただし、ソフトウェアプロダクトエンジニアリングのキープロセスエリアは含んでいない)。SEI内部での第0.41版のピアレビューからのコメントを反映した版。CMMユーザーキンググループと質問票助言委員会によるレビューとコメントを求めるために配布された改訂版。両組織によるレベル3のキープロセスエリアとキープラクティス表のレビューを支援するために、概要と定義の素案の改訂版を付加。
1991年5月17日	0.51	草案。CMMユーザーキンググループからの第0.4版のレベル2のキープロセスエリア[訳注:原文ではキープラクティスエリアと書いてある]に対するコメントを反映させた。また、単一の成熟度レベル内での各キープロセスエリアのプラクティスを作成。SEI内部でのピアレビュー用。
1991年5月23日	0.52	草案。キープラクティス表の改訂版。CMMユーザーキンググループと質問票助言委員会によるコメントを反映した版。ソフトウェア要件管理のキープロセスエリア、ソフトウェア要件分析・ソフトウェア設計・ソフトウェアテストのキープロセスエリアを集約したもの、独立したキープロセスエリアとして追加されたコーディングプラクティスおよびソフトウェアプロダクトエンジニアリングの最初の版が加わった。
1991年6月5日	0.53	予備的ベースライン。SEI内部での第0.51版のピアレビューでのコメントを反映させてレベル2のキープロセスエリアを改訂した版。この版は、SEIの情報管理部門に提出された。

1991年6月7日	0.54	草案。キープラクティス表を改訂(ソフトウェアプロジェクト管理のキープロセスエリアの最初の版を追加したのみ)した版。SEI内部でのピアレビュー用。
1991年6月14日	0.6	草案。SEIでのピアレビューでのコメントを反映したキープラクティス表の改訂版。ソフトウェア要件管理(第0.52版)、ソフトウェアプロジェクト管理(第0.54版)、およびソフトウェアプロダクトエンジニアリング(第0.52版)のキープロセスエリア、レベル2におけるその他のキープロセスエリアの第0.53版に対するコメントを反映。CMMユーザーキンググループと質問票助言委員会によるレビューとコメントを求めるために配布された。素案。CMMユーザーキンググループと質問票助言委員会におけるキープラクティス表(ソフトウェア要件管理、ソフトウェアプロジェクト管理、およびソフトウェアプロダクトエンジニアリングのキープロセスエリア)のレビューを支援するために、概要と定義の改訂版を付与。この版における重要な変更点は以下の通りである。(1)コモンキーフィーチャの変更にとまなうキープロセスエリアプラクティスの再構成、(2)キープロセスエリアの組み合わせ方の変更、(3)各キープロセスエリアが単一の成熟度レベルに対するプラクティス群を含むことができるようにするためのプラクティス記述の再整理。
1991年6月21日	0.61	草案。キープラクティス表(レベル4と5のキープロセスエリアのみ)の改訂版。CMMユーザーキンググループと質問票助言委員会による第0.2版へのコメントを反映した。SEI内部でのピアレビュー用。
1991年6月28日	0.7	草案。キープラクティス表の付録(レベル4と5のキープロセスエリアのみ)を改訂した版。SEIにおける第0.61版のピアレビューでのコメントを反映。CMMユーザーキンググループと質問票助言委員会によるレビューとコメントを求めるために配布された。

変更履歴

1991年7月10日	0.71	草案。CMMユーザーキンググループによる第0.5版および第0.6版に対するコメントを反映してレベル3のキープロセスエリアとソフトウェア要件管理のキープロセスエリアを改訂した。また、単一の成熟度レベルにおける各キープロセスエリアのプラクティス群を作成した。さらに、3つのキープロセスエリアの名称を変更した(ソフトウェア要件管理を要件管理に、ソフトウェアプロジェクト管理をソフトウェア統合管理に、技術チーム調整をグループ間調整に変更)。SEI内部でのピアレビュー用。
1991年7月14日	0.72	草案。SEI内部での第0.71版に対するピアレビューでのコメントを反映し、レベル3のキープロセスエリアとソフトウェア要件管理のキープロセスエリアを改訂した版。SEI内部でのピアレビュー用。
1991年7月15日	0.73	草案。SEI内部でのピアレビューでのコメントを反映してレベル3のキープロセスエリアとソフトウェア要件管理のキープロセスエリアを改訂した版。
1991年7月15日	0.74	予備的ベースライン。レベル3のキープロセスエリアとソフトウェア要件管理のキープロセスエリアの改訂にともない、文書を整形し直して編集上の誤りを訂正した版。この版はSEIの情報管理部門に提出された。
1991年7月22日	0.75	草案。CMMユーザーキンググループによる第0.7版に対するコメントを反映し、レベル4と5のキープロセスエリアを改訂した版。SEI内部でのピアレビュー用。
1991年7月30日	0.76	予備的ベースライン。SEI内部でのピアレビューによるコメントを反映し、レベル4と5のキープロセスエリアを改訂した版。この版はSEIの情報管理部門に提出された。

変更履歴

1991年7月30日	0.77	草案。1991年7月30日に開催された質問票助言委員会会議でのコメントを反映し、要件管理のキーププロセスエリアについて、予備的ベースラインを改訂した版。SEI内部でのピアレビュー用。
1991年8月5日	0.78	草案。1991年7月31日に開催された質問票助言委員会会議でのコメントを反映し、要件管理のキーププロセスエリアについて、第0.77版を改訂した版。SEI内部でのピアレビュー用。
1991年8月7日	0.79	改訂された予備的ベースライン。質問票助言委員会会議のコメントとSEI内部での第0.78版に対するピアレビューでのコメントを反映した版。
1991年8月7日	0.80	草案。CMMユーザーキンググループと質問票助言委員会による第0.4、0.5、0.6版に対するコメントを反映して概要と定義を改訂するとともに、文書を再構成した版。SEI内部でのピアレビュー用。
1991年8月15日	1.0	一般向け公開のためのベースライン版。
1992年9月1日	1.01	草案。1992年1月17日までに受領した変更要求書と、1992年4月6-7日に開催されたCMMワークショップでの勧告に基づいて第1.0版を改訂した版。トレーニングプログラムというキーププロセスエリアがスキル構築に置き換えられた。技術革新が技術変更管理に名称変更された。
1992年12月8日	1.02	草案。CMM助言委員会からの勧告と、SEI内部でのピアレビューでのコメントを反映するために、第1.01版を改訂した版。スキル構築に関する小規模な格上げを行った上で、トレーニングプログラムというキーププロセスエリアが復元された。プロセス計測および分析というキーププロセスエリアの名称が、定量的プロセス管理に変更された。SEI内部でのピアレビュー用。

変更履歴

1992年12月14日	1.03	草案。SEI内部でのピアレビューでのコメントを反映して第1.02版を改訂した版。品質管理というキープロセスエリアの名称が、ソフトウェア品質管理に変更された。SEI内部でのピアレビュー用。
1992年12月18日	1.04	草案。SEI内部でのピアレビューでのコメントを反映して第1.03版を改訂した版。第1.01版のレビューアとCMM助言委員会にコメントとフィードバックを求めるために配布された。
1993年1月15日	1.05	草案。CMM助言委員会の勧告と、SEI内部でのピアレビューでのコメントを反映して第1.04版を改訂した版。SEI内部でのピアレビュー用。
1993年2月10日	1.1	一般向け公開のためのベースライン版。

付録E:索引

A

- Ability to Perform
 - description of 0-28
- acceptance testing
 - subcontractor's products L2-55
- action item
 - data in L5-11
- action proposal
 - data in L5-11
- Activities Performed
 - description of 0-28
- activity
 - definition of 0-59
- allocated requirements. *See also*
 - system requirements allocated to software
 - definition of L2-2
- assessment
 - software process L3-6
- audit
 - defect prevention L5-15
 - integrated software management L3-57
 - intergroup coordination L3-92
 - organization process definition L3-23
 - peer reviews L3-100
 - process change management L5-46
 - quantitative process management L4-16
 - requirements management L2-9
 - software baseline L2-81
 - software configuration management L2-83
 - software product engineering L3-81
- 実施能力
 - 記述： 0-28
- 検収テスト
 - 外注した作業成果物： L2-55
- 処置項目
 - データ： L5-11
- 処置提案
 - データ： L5-11
- 実施される活動
 - 記述： 0-28
- 活動
 - 定義： 0-59
- 割り当てられた要件（参照先：ソフトウェアに割り当てられたシステム要件）
 - 定義： L2-2
- アセスメント
 - ソフトウェアプロセス： L3-6
- 監査
 - 欠陥予防： L5-15
 - ソフトウェア統合管理： L3-57
 - グループ間調整： L3-92
 - 組織プロセス定義： L3-23
 - ピアレビュー： L3-100
 - プロセス変更管理： L5-46
 - 定量的プロセス管理： L4-16
 - 要件管理： L2-9
 - ソフトウェアベースライン： L2-81
 - ソフトウェア構成管理： L2-83
- ソフトウェアプロダクトエンジニアリング：
 - L3-81

索引

software project planning L2-27
software project tracking and oversight
L2-41
software quality management L4-32
software subcontract management 57
software work products L2-66
technology change management L5-29

ソフトウェアプロジェクト計画 : L2-27
ソフトウェアプロジェクト進捗管理 : L2-
41
ソフトウェア品質管理 : L4-32
ソフトウェア外注管理 : 57
ソフトウェア作業成果物 : L2-66
技術変更管理 : L5-29

B

baseline
 procedure for controlling changes to 80
baseline configuration management
 description of 0-46
budget
 versus funding 0-37

ベースライン
 変更を制御する手順 : 80
ベースライン構成管理
 記述 : 0-46
予算
 と資金 : 0-37

C

Capability Maturity Model
 background of 0-7
 description of 0-7
 evolution of 0-8
 flexibility in 0-75
 need for 0-1
 organizational structure used in 0-64
 relation to appraisal methods 0-4
 relation to documents 0-62
 relation to life cycles 0-61
 relation to technology 0-62
 sources of 0-3
 structure of 0-9, 0-12
 use for small and/or commercial
 organizations 0-2
 uses of 0-1
 requirements for L5-7
causal analysis meetings
 procedure for 79

能力成熟度モデル
 背景 : 0-7
 記述 : 0-7
 発展 : 0-8
 柔軟性 : 0-75
 ニーズ : 0-1
 使用されている組織構造 : 0-64
 査定手法との関係 : 0-4
 文書との関係 : 0-62
 ライフサイクルとの関係 : 0-61
 技術との関係 : 0-62
 起源 : 0-3
 構造 : 0-9, 0-12
 小規模かつ/または民間組織での使用 :
 0-2
 使用 : 0-1
 要件 : L5-7
原因分析会議
 手順 : 79

- change request
 procedure for evaluating L2-79
- checklist for evaluating software work products L3-98
- Commitment to Perform
 description of 0-28
- commitments
 changes to 7
 review by senior management L2-31, L2-35
 software project L2-12
- Common Features
 definition of 0-11, 0-27
- configuration items/units
 definition of L2-73
 identification of L2-78
 procedure for changing L2-79
 recording status of L2-80
- configuration management. *See also* software configuration management and "manage and control"
 baseline 0-46
 developmental 0-46
 versus manage and control 0-46
- configuration management library system
 definition of L2-73
 establishment of L2-77
- contingency planning L3-54
- contract
 software subcontract management L2-50
- conventions
 description of 0-36
 deviation from software development plan L2-30
- corrective actions
 deviations from software development plan L2-30
- 変更要求
 評価の手順： L2-79
ソフトウェア成果物を評価するチェックリスト： L3-98
実施のコミットメント
 記述： 0-28
コミットメント
 変更： 7
 上級管理層によるレビュー： L2-31, L2-35
 ソフトウェアプロジェクト： L2-12
コモンフィーチャ
 定義： 0-11, 0-27
構成アイテム/ユニット
 定義： L2-73
 特定： L2-78
 変更する手順： L2-79
 状況記録： L2-80
構成管理（参照先：ソフトウェア構成管理と“管理と制御”）
 ベースライン： 0-46
 開発： 0-46
 と管理制御： 0-46
構成管理ライブラリシステム
 定義： L2-73
 確立： L2-77
リスク対応計画： L3-54
契約
 ソフトウェア外注管理： L2-50
慣例
 記述： 0-36
 ソフトウェア開発計画からの逸脱： L2-30
是正処置
 ソフトウェア開発計画からの逸脱： L2-30

索引

critical computer resources
 basis for estimates L3-50
 estimates for L2-23
 management of L3-50
 provision for reserve capacity L3-51
 tracking of L2-36
critical dependencies L3-89
customer
 examples of L2-1
 internal versus external 0-44
 milestone review with L2-39
 review
 software quality assurance L2-67
 software requirements document
 L3-68
 software test plan L3-75
 software testing criteria L3-72
 waivers from contractual software
 requirements L3-42
 review and approve
 system documentation L3-77

D

data
 defect prevention L5-11
 evolution of practices for collecting
 and analyzing 0-63
defect
 examples of root causes L5-8
 prevention at Level 5 0-17
 use of data on L3-78
Defect Prevention L5-1 through L5-15
 causal analysis meeting for L5-7
 description of 0-24
 documentation of data on L5-11
 examples of activities for L5-4

重要なコンピュータ資源
見積りの基盤： L3-50
見積り： L2-23
管理： L3-50
予備の容量の提供： L3-51
進捗の確認： L2-36
重要な依存関係： L3-89
顧客
 例： L2-1
 内部の場合と外部の場合： 0-44
 マイルストーンレビュー： L2-39
 レビュー
 ソフトウェア品質保証： L2-67
 ソフトウェア要件文書： L3-68

 ソフトウェアテスト計画： L3-75
 ソフトウェアテスト基準： L3-72
 契約上のソフトウェア要件からの免
 除： L3-42
 レビューと承認
 システム文書： L3-77

データ
 欠陥予防： L5-11
 収集と分析のプラクティスの発展： 0-63

欠陥
 根本原因の例： L5-8
 レベル5で予防： 0-17
 データの使用： L3-78
欠陥予防： L5-1～L5-15
 原因分析会議： L5-7
 記述： 0-24
 データの文書化： L5-11
 活動の例： L5-4

- feedback on L5-12
- kick-off meeting for task L5-6
- management and control of data for L5-11
- measurements for L5-13
- plan for project L5-5
- policy for organization L5-2
- policy for project L5-2
- resources and funding for L5-4
- review
 - project manager L5-15
 - senior management L5-14
- review and audit
 - software quality assurance L5-15
- team for organization L5-3
- team for project L5-3
- tools for L5-4
- training for L5-4
- defect prevention plan
 - peer review of L5-5
- defect prevention team
 - activities of L5-8
 - establishment of L5-3
- Defined Level
 - description of 0-15
- defined software process
 - development of software design L3-69
 - development of software requirements L3-66
 - use in development of software code L3-71
- developmental configuration management
 - description of 0-46
- documented procedure
 - acceptance testing of subcontractor's products L2-55
- フィードバック： L5-12
- タスクの立ち上げ会議： L5-6
- データの管理と制御： L5-11
- 計測： L5-13
- プロジェクトの計画： L5-5
- 組織方針： L5-2
- プロジェクト方針： L5-2
- 資源と資金： L5-4
- レビュー
 - プロジェクトマネージャ： L5-15
 - 上級管理層： L5-14
- レビューと監査
 - ソフトウェア品質保証： L5-15
- 組織のためのチーム： L5-3
- プロジェクトのためのチーム： L5-3
- ツール： L5-4
- トレーニング： L5-4
- 欠陥予防計画
 - ピアレビュー： L5-5
- 欠陥予防チーム
 - 活動： L5-8
 - 確立： L5-3
- 定義されたレベル
 - 記述： 0-15
- 定義されたソフトウェアプロセス
 - ソフトウェア設計の開発： L3-69
 - ソフトウェア要件の開発： L3-66
- ソフトウェアコードの開発における使用： L3-71
- 開発構成管理
 - 記述： 0-46
- 文書化された手順
 - 外注先成果物の検収テスト： L2-55

索引

- causal analysis meetings for defect prevention L5-7
- changes to software subcontractor's statement of work L2-51
- changing baselines L2-80
- changing configuration items/units L2-79
- control of project's defined software process L4-10
- deriving software schedule L2-23
- description of 0-42
- developing project's defined software process L3-41
- developing software configuration management plan L2-76
- development of organization's standard software process L3-15
- development of software development plan L2-18
- deviations in software engineering activities L2-67
- estimating critical computer resources L2-23
- estimating resources and costs L2-22
- estimating size of software work products L2-21
- identification of critical dependencies L3-89
- implementation of software process improvement L5-42
- incorporation of new technology into organization's standard software process L5-28
- incorporation of new technology into project's defined software process L5-28
- 欠陥予防の原因分析会議： L5-7
- ソフトウェア外注先の作業内容記述書の変更： L2-51
- ベースラインの変更： L2-80
- 構成アイテム/ユニットの変更： L2-79
- プロジェクトの定義されたソフトウェアプロセスの制御： L4-10
- ソフトウェアスケジュールの作成： L2-23
- 記述： 0-42
- プロジェクトの定義されたソフトウェアプロセスの開発： L3-41
- ソフトウェア構成管理計画の策定： L2-76
- 組織の標準ソフトウェアプロセスの開発： L3-15
- ソフトウェア開発計画の策定： L2-18
- ソフトウェアエンジニアリング活動の逸脱： L2-67
- 重要なコンピュータ資源の見積り： L2-23
- 資源と工数の見積り： L2-22
- ソフトウェア作業成果物の規模の見積り： L2-21
- 重要な依存関係の特定： L3-89
- ソフトウェアプロセス改善の実装： L5-42
- 組織の標準ソフトウェアプロセスへの新しい技術の導入： L5-28
- プロジェクトの定義されたソフトウェアプロセスへの新しい技術の導入： L5-28

- intergroup issues L3-90
management of critical computer resources L3-50
management of software costs L3-48
management of software schedule L3-51
milestone reviews L2-39
monitoring subcontractor's software configuration management L2-54
monitoring subcontractor's software quality assurance L2-53
organization's training plan L3-30
peer reviews L3-97
preparation of software quality assurance plan L2-63
process capability baseline L4-13
products from software baseline library L2-80
project's software development plan L3-44
quantitative process control data L4-9
quantitative process management plan L4-6
recording status of configuration items L2-80
review of commitments L2-35
reviews of software subcontractor L2-53
revising software development plan L2-33
revision of project's defined software process L3-43
revisions to organization's standard software process based on defect prevention L5-12
revisions to project's defined software process based on defect prevention L5-12
- グループ間の課題： L3-90
重要なコンピュータ資源の管理： L3-50
ソフトウェアコストの管理： L3-48
ソフトウェアスケジュールの管理： L3-51
マイルストーンレビュー： L2-39
外注先のソフトウェア構成管理のモニター： L2-54
外注先のソフトウェア品質保証のモニター： L2-53
組織のトレーニング計画： L3-30
ピアレビュー： L3-97
ソフトウェア品質保証計画の準備： L2-63
プロセス能力ベースライン： L4-13
ソフトウェアベースラインライブラリからの成果物： L2-80
プロジェクトのソフトウェア開発計画： L3-44
定量的プロセス制御データ： L4-9
定量的プロセス管理計画： L4-6
構成アイテムの状況記録： L2-80
コミットメントのレビュー： L2-35
ソフトウェア外注先のレビュー： L2-53
ソフトウェア開発計画の改訂： L2-33
プロジェクトの定義されたソフトウェアプロセスの改訂： L3-43
欠陥予防に基づく組織の標準ソフトウェアプロセスの改訂： L5-12
欠陥予防に基づくプロジェクトの定義されたソフトウェアプロセスの改訂： L5-12

索引

- risk management L3-52
 - selection of new technologies L5-26
 - selection of software subcontractor L2-49
 - software baseline audits L2-81
 - software process improvement plan L5-37
 - software process improvement proposals L5-39
 - software quality plan L4-23
 - software size management L3-47
 - subcontracted work L2-47
 - documents
 - relation to Capability Maturity Model 0-62
- リスク管理： L3-52
 - 新しい技術の選択： L5-26
 - ソフトウェア外注先の選定： L2-49
 - ソフトウェアベースライン監査： L2-81
 - ソフトウェアプロセス改善計画： L5-37
 - ソフトウェアプロセス改善提案： L5-39
 - ソフトウェア品質計画： L4-23
 - ソフトウェア規模管理： L3-47
 - 外注された作業： L2-47
 - 文書
 - 能力成熟度モデルとの関係： 0-62

E

- effort and costs
 - estimates for L2-22
 - engineering groups
 - dependencies between L3-89
 - estimates
 - critical computer resources L2-23
 - effort and costs L2-22
 - size of software work products L2-21
- 工数とコスト
 - 見積り： L2-22
 - エンジニアリンググループ
 - 依存関係： L3-89
 - 見積り
 - 重要なコンピュータ資源： L2-23
 - 工数とコスト： L2-22
 - ソフトウェア作業成果物の規模： L2-21

F

- Fagan-style inspections
 - use in Peer Reviews 0-23
 - first-line software manager
 - definition of 0-67
 - review
 - project tracking L2-38
 - funding
 - versus budget 0-37
- Faganスタイルインスペクション
 - ピアレビューでの使用： 0-23
 - 第一線ソフトウェアマネージャ
 - 定義： 0-67
 - レビュー
 - プロジェクトの進捗確認： L2-38
 - 資金
 - と予算： 0-37

G

goals

- definition of 0-11
- use in interpreting key practices 0-75
- use of 0-27

group

- definition of 0-70
- organization process focus L3-3
- quantitative process management L4-4
- software configuration management L2-74
- software quality assurance L2-61
- technology change management L5-20
- training L3-27

ゴール

- 定義： 0-11
- キープラクティスの解釈での使用： 0-75
- 使用： 0-27

グループ

- 定義： 0-70
- 組織プロセス重視： L3-3
- 定量的プロセス管理： L4-4
- ソフトウェア構成管理： L2-74
- ソフトウェア品質保証： L2-61
- 技術変更管理： L5-20
- トレーニング： L3-27

I

independence

- need for 0-73

Initial Level

- description of 0-13

Integrated Software Management L3-37 through L3-58

- description of 0-22
- policy for L3-38
- resources and funding for L3-39
- review

- project manager L3-57
- senior management L3-56

- training for L3-39-40

interdisciplinary engineering teams

- policy for L3-84

intergroup commitments

- plan for L3-88

Intergroup Coordination L3-83 through L3-92

独立性

- ニーズ： 0-73

初期レベル

- 記述： 0-13

ソフトウェア統合管理： L3-37～L3-58

- 記述： 0-22
- 方針： L3-38
- 資源と資金： L3-39

レビュー

- プロジェクトマネージャ： L3-57
- 上級管理層： L3-56

- トレーニング： L3-39-40

分野横断的なエンジニアリングチーム

- 方針： L3-84

グループ間のコミットメント

- 計画： L3-88

グループ間調整： L3-83～L3-92

索引

description of 0-22
measurements for L3-91
orientation in L3-86
resources and funding for L3-85
review
 project manager L3-92
 senior management L3-91
review and audit
 software quality assurance L3-92
tools for L3-85
intergroup issues
 procedure for L3-90

J

judgment
 use in interpreting key practices 0-75

K

key practices
 conventions used in 0-35
 definition of 0-12
 description of 0-26, 0-31
 example page 0-33
 implementation of 0-35
 intention of 0-35
 interpretation of 0-27, 0-75
 structure of 0-31
key process areas
 categories of 0-26
 definition of 0-10
 description of 0-18
 description of Level 2 0-20 through 0-21
 description of Level 3 0-21 through 0-23
 description of Level 4 0-23

記述： 0-22
計測： L3-91
オリエンテーション： L3-86
資源と資金： L3-85
レビュー
 プロジェクトマネージャ： L3-92
 上級管理層： L3-91
レビューと監査
 ソフトウェア品質保証： L3-92
ツール： L3-85
グループ間の課題
 手順： L3-90

判定
 キープラクティスの解釈での使用： 0-75

キープラクティス
 使用されている慣例： 0-35
 定義： 0-12
 記述： 0-26, 0-31
 例のページ： 0-33
 実装： 0-35
 意図： 0-35
 解釈： 0-27, 0-75
 構造： 0-31
キープロセスエリア
 区分： 0-26
 定義： 0-10
 記述： 0-18
 レベル 2 の記述： 0-20 ~ 0-21
 レベル 3 の記述： 0-21 ~ 0-23
 レベル 4 の記述： 0-23

description of Level 5 0-23 through 0-24
 identification by maturity level 0-19
 spanning maturity levels
 0-24

レベル 5 の記述： 0-23 ~ 0-24
 成熟度レベルの特定： 0-19
 成熟度レベルの重なり： 0-24

L

leadership
 description of 0-36
 library of software process-related
 documentation
 definition of 0-57
 life cycle
 relation to Capability Maturity Model
 0-61

リーダーシップ
 記述： 0-36
 ソフトウェアプロセス関連文書のライブラリ
 定義： 0-57
 ライフサイクル
 能力成熟度モデルとの関係： 0-61

M

manage and control. *See also* software
 configuration management and
 configuration management
 defect prevention data L5-11
 noncompliance items L2-67
 process capability baseline L4-14
 process performance baseline L4-12
 quantitative process management plan
 L4-7
 software configuration management plan
 L2-77
 software development plan L2-13, L2-19,
 L2-34
 software planning data L2-25
 software process improvement plan L5-
 37
 software quality assurance plan L2-64
 software quality plan L4-25

管理と制御（参照先：ソフトウェア構成管理
 と構成管理）
 欠陥予防データ： L5-11
 非遵守問題： L2-67
 プロセス能力ベースライン： L4-14
 プロセス実績ベースライン： L4-12
 定量的プロセス管理計画： L4-7
 ソフトウェア構成管理計画： L2-77
 ソフトウェア開発計画： L2-13, L2-19,
 L2-34
 ソフトウェア計画データ： L2-25
 ソフトウェアプロセス改善計画： L5-37
 ソフトウェア品質保証計画： L2-64
 ソフトウェア品質計画： L4-25

索引

- software replanning data L2-38
- statement of work for software project L2-15
- subcontract statement of work L2-48
- system requirements allocated to software L2-7
- versus configuration management 0-46
- Managed Level
 - description of 0-16
- manager
 - definition of 0-64
 - software quality assurance L2-62
 - subcontract L2-45
- managing
 - versus tracking 0-45
- maturity level
 - definition of 0-10
 - relation to process capability 0-12
 - spanning multiple 0-24
- maturity questionnaire
 - relation to Capability Maturity Model 0-4
- Measurement and Analysis
 - description of 0-28
- measurement data
 - collection of L4-9
 - examples of L4-9
 - storage of L4-10
 - techniques for analyzing L4-11
 - program for organization L4-5
- measurements. *See also* Quantitative Process Management
 - defect prevention L5-13
 - institutionalization at Level 4 0-16
 - integrated software management L3-56
- ソフトウェア再計画データ： L2-38
- ソフトウェアプロジェクトの作業内容記述書： L2-15
- 外注の作業内容記述書： L2-48
- ソフトウェアに割り当てられたシステム要件： L2-7
- と構成管理： 0-46
- 管理されたレベル
 - 記述： 0-16
- マネージャ
 - 定義： 0-64
 - ソフトウェア品質保証： L2-62
 - 外注： L2-45
- 管理
 - と進捗の確認： 0-45
- 成熟度レベル
 - 定義： 0-10
 - プロセス能力の関係： 0-12
 - 複数の重なり： 0-24
- 成熟度質問票
 - 能力成熟度モデルとの関係： 0-4
- 計測と分析
 - 定義： 0-28
- 計測データ
 - 収集： L4-9
 - 例： L4-9
 - 保存： L4-10
 - 分析技法： L4-11
 - 組織のプログラム： L4-5
- 計測（参照先：定量的プロセス管理）
 - 欠陥予防： L5-13
 - レベル4での制度化： 0-16
 - ソフトウェア統合管理： L3-56

- intergroup coordination L3-91
 organization process definition L3-22
 organization process focus L3-9
 peer reviews L3-99
 process change management L5-45
 quantitative process management L4-15
 requirements management L2-8
 software configuration management L2-82
 software engineering activities L3-80
 software product engineering L3-79
 software project planning L2-25
 software project tracking and oversight L2-39
 software quality assurance L2-68
 software quality management L4-31
 software subcontract management L2-55
 technology change management L5-28
 training program L3-34-35
 milestones
 identification of L3-51
- N**
- new technology
 information on L5-25
 selection of L5-26
 transfer within organization L3-8
 noncompliance items
 management and control of L2-67
- O**
- Optimizing Level
 description of 0-17
- グループ間調整： L3-91
 組織プロセス定義： L3-22
 組織プロセス重視： L3-9
 ピアレビュー： L3-99
 プロセス変更管理： L5-45
 定量的プロセス管理： L4-15
 要件管理： L2-8
 ソフトウェア構成管理： L2-82
 ソフトウェアエンジニアリング活動：
 L3-80
 ソフトウェアプロダクトエンジニアリン
 グ： L3-79
 ソフトウェアプロジェクト計画： L2-25
 ソフトウェアプロジェクト進捗管理：
 L2-39
 ソフトウェア品質保証： L2-68
 ソフトウェア品質管理： L4-31
 ソフトウェア外注管理： L2-55
 技術変更管理： L5-28
 トレーニングプログラム： L3-34-35
 マイルストーン
 定義： L3-51
- 新しい技術
 情報： L5-25
 選択： L5-26
 組織内への移転： L3-8
 非遵守問題
 管理と制御： L2-67
- 最適化するレベル
 記述： 0-17

索引

Organization Process Definition L3-11 through L3-23	組織プロセス定義： L3-11 ~ L3-23
description of 0-21	記述： 0-21
measurements for L3-22	計測： L3-22
policy for L3-12	方針： L3-12
resources and funding for L3-14	資源と資金： L3-14
review and audit	レビューと監査
software quality assurance L3-23	ソフトウェア品質保証： L3-23
tools for L3-14	ツール： L3-14
training for L3-14	トレーニング： L3-14
Organization Process Focus L3-1 through L3-10	組織プロセス重視： L3-1 ~ L3-10
coordination of training for L3-8	トレーニングの調整： L3-8
description of 0-21	記述： 0-21
group for L3-3	グループ： L3-3
involvement of key groups L3-9	キーグループの関与： L3-9
management sponsorship of L3-2	管理層の主催者としての態度： L3-2
measurements for L3-9	計測： L3-9
orientation in L3-6	オリエンテーション： L3-6
plan for L3-7	計画： L3-7
policy for L3-2	方針： L3-2
resources and funding for L3-4	資源と資金： L3-4
review	レビュー
senior management L3-10	上級管理層： L3-10
senior management oversight of L3-3	上級管理層の監督レビュー： L3-3
tools for L3-5	ツール： L3-5
training for L3-5	トレーニング： L3-5
organization's software process database L4-13	組織のソフトウェアプロセスデータベース： L4-13
definition of 0-56	定義： 0-56
examples of data in L3-47	データの例： L3-47
storage of quantitative process management data L4-10	定量的プロセス管理データの保存： L4-10
organization's standard software process analysis for new technology L5-25	組織の標準ソフトウェアプロセス 新しい技術の分析： L5-25

- analyzing capability of L4-3
baseline for L4-3, L4-13
changes to L4-15
configuration management of L3-17
coordination at organization level L3-7
definition of 0-15, 0-54
deviations from L3-39
guidelines for tailoring L3-19, 0-56
- incorporation of new technology L5-28
incorporation of process changes L5-43
peer review of L3-16
policy for L3-12
procedure for developing L3-15
process capability trends for L4-13
relation to software process definition
0-50
revision based on defect prevention
L5-12
tailoring of L3-41
use in project planning and management
L3-38
waivers for deviations from L3-42
definition of 0-70
- orientation. *See also* training and required
training
description of 0-39
intergroup coordination L3-86
organization process focus L3-6
quantitative process management L4-6
software product engineering for
managers L3-64
software product engineering for
technical staff L3-64
- 能力分析： L4-3
ベースライン： L4-3, L4-13
変更： L4-15
ソフトウェア構成管理： L3-17
組織レベルでの調整： L3-7
定義： 0-15, 0-54
逸脱： L3-39
テーラリングのガイドライン： L3-19, 0-
56
新しい技術の導入： L5-28
プロセス変更の導入： L5-43
ピアレビュー： L3-16
方針： L3-12
開発の手順： L3-15
プロセス能力の傾向： L4-13
ソフトウェアプロセス定義との関係： 0-
50
欠陥予防に基づく改訂： L5-12
- テーラリング： L3-41
プロジェクトの計画と管理での使用：
L3-38
逸脱の免除： L3-42
定義： 0-70
- オリエンテーション（参照先：トレーニング
と必修トレーニング）
記述： 0-39
グループ間調整： L3-86
組織プロセス重視： L3-6
定量的プロセス管理： L4-6
マネージャにとってのソフトウェアプロ
ダクトエンジニアリング： L3-64
技術要員にとってのソフトウェアプロダ
クトエンジニアリング： L3-64

索引

training program L3-29

トレーニングプログラム： L3-29

P

Pareto analysis

use in defect prevention L5-9

Peer Reviews L3-93 through L3-100

description of 0-22

plans for L3-97

policy for L3-94

procedure for L3-97

resources and funding for L3-95

review and audit

software quality assurance L3-100

measurements for L3-99

training to lead L3-95

training to participate in L3-96

defect prevention plan L5-6

quantitative process management plan
L4-7

peer review

checklist for evaluating work products
L3-98

organization's standard software
process L3-16

plan for organization process focus
L3-7

software code L3-72

software design document L3-71

software life cycle L3-19

software process improvement plan L5-
37

software requirements document L3-68

software risk management plan L3-54

software quality plan L4-24

software test plan L3-74

system documentation L3-77

パレート分析

欠陥予防での使用： L5-9

ピアレビュー： L3-93～L3-100

記述： 0-22

計画： L3-97

方針： L3-94

手順： L3-97

資源と資金： L3-95

レビューと監査

ソフトウェア品質保証： L3-100

計測： L3-99

主導するためのトレーニング： L3-95

参加するためのトレーニング： L3-96

欠陥予防計画： L5-6

定量的プロセス管理計画： L4-7

ピアレビュー

作業成果物評価のチェックリスト： L3-98

組織の標準ソフトウェアプロセス： L3-16

組織プロセス重視の計画： L3-7

ソフトウェアコード： L3-72

ソフトウェア設計文書： L3-71

ソフトウェアライフサイクル： L3-19

ソフトウェアプロセス改善計画： L5-37

ソフトウェア要件文書： L3-68

ソフトウェアリスク管理計画： L3-54

ソフトウェア品質計画： L4-24

ソフトウェアテスト計画： L3-74

システム文書： L3-77

- technology change management plan L5-24
 versus review 0-45
 periodic and event-driven review
 description of 0-49
 periodic review
 description of 0-48
 pilot efforts
 technology changes L5-27
 plan
 communication of intergroup commitments L3-88
 defect prevention L5-5
 description of 0-40
 formal 0-41
 informal 0-41
 organization process focus L3-7
 peer reviews L3-97
 process change management L5-37
 quantitative process management L4-6
 software configuration management L2-76
 software development L2-14
 software engineering L2-25
 software quality assurance L2-63
 software quality L4-23
 technology change management L5-23
 training L3-29
 plan for organization process focus
 peer review of L3-7
 planning. *See also* Software Project Planning
 policy
 defect prevention for organization L5-2
 defect prevention for project L5-2
 技術変更管理計画： L5-24
 とレビュー： 0-45
 定期的およびイベント発生を契機とするレビュー
 記述： 0-49
 定期的レビュー
 記述： 0-48
 試行作業
 技術変更： L5-27
 計画
 グループ間調整の伝達： L3-88
 欠陥予防： L5-5
 記述： 0-40
 公式： 0-41
 非公式： 0-41
 組織プロセス重視： L3-7
 ピアレビュー： L3-97
 プロセス変更管理： L5-37
 定量的プロセス管理： L4-6
 ソフトウェア構成管理： L2-76
 ソフトウェア開発： L2-14
 ソフトウェアエンジニアリング： L2-25
 ソフトウェア品質保証： L2-63
 ソフトウェア品質： L4-23
 技術変更管理： L5-23
 トレーニング： L3-29
 組織プロセス重視の計画
 ピアレビュー： L3-7
 計画（参照先：ソフトウェアプロジェクト計画）
 方針
 組織の欠陥予防： L5-2
 プロジェクトの欠陥予防： L5-2

索引

- description of 0-36
- integrated software management L3-38
- interdisciplinary engineering teams L3-84
- management of system requirements allocated to software L2-2
- organization process definition L3-12
- organization process focus L3-2
- peer reviews L3-94
- quantitative process management L4-23
- software configuration management L2-72
- software process improvement L5-32
- software product engineering L3-60

- software project planning L2-12

- software project tracking and oversight L2-30
- software quality assurance L2-60
- software quality management L4-20
- software subcontract management L2-44
- technology change management L5-18
- training L3-26
- prerequisite items
 - description of 0-39
- problem report
 - procedure for L2-79
- procedure. *See also* documented procedure
- process capability
 - definition of L4-1
 - trends in L4-13
- Process Change Management L5-31 through L5-47. *See also* software process improvement

- 記述： 0-36
- ソフトウェア統合管理： L3-38
- 分野横断的なエンジニアリングチーム： L3-84
- ソフトウェアに割り当てられたシステム要件の管理： L2-2
- 組織プロセス定義： L3-12
- 組織プロセス重視： L3-2
- ピアレビュー： L3-94
- 定量的プロセス管理： L4-23
- ソフトウェア構成管理： L2-72

- ソフトウェアプロセス改善： L5-32
- ソフトウェアプロダクトエンジニアリング： L3-60
- ソフトウェアプロジェクト計画： L2-12

- ソフトウェアプロジェクト進捗管理： L2-30
- ソフトウェア品質保証： L2-60
- ソフトウェア品質管理： L4-20
- ソフトウェア外注管理： L2-44
- 技術変更管理： L5-18
- トレーニング： L3-26
- 前提要件項目
 - 記述： 0-39
- 問題レポート
 - 手順： L2-79
- 手順（参照先：文書化された手順）
- プロセス能力
 - 定義： L4-1
 - 傾向： L4-13
- プロセス変更管理： L5-31～L5-47（参照先：ソフトウェアプロセス改善）

- description of 0-24
 measurements for L5-42
 resources and funding for L5-33
 review
 senior management
 review and audit
 software quality assurance L5-46
 tools for L5-34
 training for L5-34 through L5-35
- process assets
 policy for L3-12
 use in project planning and management L3-38
 use in tailoring organization's standard software process L3-42
- process capability baseline L4-3
 for organization's standard software process L4-13
 management and control of L4-14
 relation to maturity levels 0-12
- process element
 definition of L3-17
- process improvement. *See also* software process improvement
- process improvement
 focus at Level 5 0-17
- process performance
 definition of L4-1
- process performance baseline L4-13
 management and control of L4-12
 definition of L4-1
- project manager
 definition of 0-66
 negotiation of commitments L2-12
 review
 defect prevention L5-15
 integrated software management L3-57
 intergroup commitments L3-89
- 記述： 0-24
 計測： L5-42
 資源と資金： L5-33
 レビュー
 上級管理層
 レビューと監査
 ソフトウェア品質保証： L5-46
 ツール： L5-34
 トレーニング： L5-34～L5-35
- プロセス資産
 方針： L3-12
 プロジェクトの計画と管理での使用： L3-38
 組織の標準ソフトウェアプロセスのテーラリングでの使用： L3-42
- プロセス能力ベースライン： L4-3
 組織の標準ソフトウェアプロセス： L4-13
- 管理と制御： L4-14
 成熟度レベルとの関係： 0-12
- プロセス構成要素
 定義： L3-17
- プロセス改善（参照先：ソフトウェアプロセス改善）
 レベル5での焦点： 0-17
- プロセス実績
 定義： L4-1
- プロセス実績ベースライン： L4-13
 管理と制御： L4-12
 定義： L4-1
- プロジェクトマネージャ
 定義： 0-66
 コミットメントの協議： L2-12
 レビュー
 欠陥予防： L5-15
 ソフトウェア統合管理： L3-57
- グループ間のコミットメント： L3-89

索引

- intergroup coordination L3-92
- quantitative process management L4-16
- requirements management L2-9
- risks L2-38
- software configuration management L2-83
- software development plan L2-19
- software product engineering L3-80

- software project planning L2-26
- software project tracking and oversight L2-41
- software quality assurance L2-69
- software quality management L4-31
- software subcontract management L2-56
- statement of work L2-15
- project software manager
 - assignment of work tasks L2-31
 - coordination of software project planning L2-15
 - definition of 0-67
 - negotiation of commitments L2-12
 - responsibilities of L2-30
 - review
 - project tracking L2-38
 - software development plan L2-19
 - statement of work L2-15
 - software project planning responsibility L2-12
- project tracking and oversight.
See also Software Project Tracking and Oversight
- project's defined software process

- グループ間調整： L3-92
- 定量的プロセス管理： L4-16

- 要件管理： L2-9
- リスク： L2-38
- ソフトウェア構成管理： L2-83

- ソフトウェア開発計画： L2-19
- ソフトウェアプロダクトエンジニアリング： L3-80
- ソフトウェアプロジェクト計画： L2-26
- ソフトウェアプロジェクト進捗管理： L2-41
- ソフトウェア品質保証： L2-69
- ソフトウェア品質管理： L4-31
- ソフトウェア外注管理： L2-56
- 作業内容記述書： L2-15
- プロジェクトソフトウェアマネージャ
 - 作業タスクの割り当て： L2-31
 - ソフトウェアプロジェクト計画の調整： L2-15
 - 定義： 0-67
 - コミットメントの協議： L2-12
 - 責任： L2-30
 - レビュー
 - プロジェクトの進捗確認： L2-38
 - ソフトウェア開発計画： L2-19
 - 作業内容記述書： L2-15
 - ソフトウェアプロジェクト計画の責任： L2-12
- プロジェクト進捗管理（参照先：ソフトウェアプロジェクト進捗管理）

- プロジェクトの定義されたソフトウェアプロセス

- coordination at organization level L3-8
coordination with defect prevention team L5-3
definition of 0-16, 0-58
documentation of L3-39
guidelines for development L3-41
incorporation of new technology L5-28
incorporation of process changes L5-43
integration of software engineering activities L3-65
management and control of L3-42
measurement and control of L4-2
quantitative control of L4-10
relation to quality goals L4-24
revision based on defect prevention L5-12
revision of L3-43
software engineering tasks and L3-60

use for quantitative process management L4-8
use for software quality management L4-20
use for software testing L3-72
use in developing system documentation L3-76
use in managing software project L3-44

use of defect data L3-78
project's software development plan
development and revision of L3-44
definition of 0-70
- 組織レベルでの調整： L3-8
欠陥予防チームとの調整： L5-3

定義： 0-16, 0-58
文書： L3-39
開発のためのガイドライン： L3-41
新しい技術の導入： L5-28
プロセス変更の導入： L5-43
ソフトウェアエンジニアリング活動の統合： L3-65
管理と制御： L3-42
計測と制御： L4-2
定量的制御： L4-10
品質目標との関係： L4-24
欠陥予防に基づく改訂： L5-12

改訂： L3-43
ソフトウェアエンジニアリングタスク： L3-60
定量的プロセス管理での使用： L4-8

ソフトウェア品質管理での使用： L4-20

ソフトウェアテストでの使用： L3-72
システム文書作成での使用： L3-76

ソフトウェアプロジェクトの管理での使用： L3-44
欠陥データでの使用： L3-78
プロジェクトのソフトウェア開発計画
開発と改訂： L3-44
定義： 0-70

索引

Q

quality function deployment (QFD) L4-23
quality goals

- allocation to subcontractors L4-30
- definition and revision of L4-27
- establishment at Level 4 0-16

quality needs

- traceability of L4-23

Quantitative Process Management
L4-1 through L4-17. *See also*

measurement

- description of 0-23
 - group for L4-4
 - measurements for L4-15
 - orientation in L4-6
 - plan for L4-6
 - policy for L4-2-3
 - reports for L4-12
 - resources and funding for L4-4
 - review
 - project manager L4-16
 - senior management L4-15
 - review and audit
 - software quality assurance L4-16
 - training for L4-6
 - tools for L4-5
- quantitative control
- definition of L4-3
- quantitative process management plan
- items in L4-7

品質機能展開(QFD) : L4-23

品質目標

- 外注先への割り当て : L4-30
- 定義と改訂 : L4-27
- レベル4での確立 : 0-16

品質ニーズ

- 追跡可能性 : L4-23

定量的プロセス管理 : L4-1 ~ L4-17 (参照先 :
計測)

記述 : 0-23

グループ : L4-4

計測 : L4-15

オリエンテーション : L4-6

計画 : L4-6

方針 : L4-2-3

報告 : L4-12

資源と資金 : L4-4

レビュー

プロジェクト管理者 : L4-16

上級管理層 : L4-15

レビューと監査

ソフトウェア品質保証 : L4-16

トレーニング : L4-6

ツール : L4-5

定量的制御

定義 : L4-3

定量的プロセス管理計画

項目 : L4-7

R

records

- software process improvement L5-43

Repeatable Level

- description of 0-14

記録

ソフトウェアプロセス改善 : L5-43

反復できるレベル

記述 : 0-14

- report
- quantitative process management L4-12
 - software configuration management activities L2-81
 - software quality assurance activities L2-67
- required training. *See also* training and orientation
- defect prevention L5-4
 - description of 0-38
 - implementing software quality management L4-22
 - leading peer reviews L3-95
 - participation in peer reviews L3-96
 - process change management L5-34-35
 - quantitative process management L4-6
 - technology change management L5-23
- Requirements Management L2-1 through L2-10
- description of 0-20
 - measurements for L2-8
 - resources and funding for L2-5
 - review
 - project manager L2-9
 - senior management L2-9
 - review and audit
 - software quality assurance L2-9
 - tools for L2-5
- resources and funding
- defect prevention L5-4
 - description of 0-37
 - integrated software management L3-39
 - intergroup coordination L3-85
 - organization process definition L3-14
- レポート
- 定量的プロセス管理： L4-12
 - ソフトウェア構成管理活動： L2-81
 - ソフトウェア品質保証活動： L2-67
- 必修トレーニング（参照先：トレーニングとオリエンテーション）
- 欠陥予防： L5-4
 - 記述： 0-38
 - ソフトウェア品質管理の実装： L4-22
 - ピアレビューの主導： L3-95
 - ピアレビューへの参加： L3-96
 - プロセス変更管理： L5-34-35
 - 定量的プロセス管理： L4-6
 - 技術変更管理： L5-23
- 要件管理： L2-1～L2-10
- 記述： 0-20
 - 計測： L2-8
 - 資源と資金： L2-5
- レビュー
- プロジェクトマネージャ： L2-9
 - 上級管理層： L2-9
- レビューと監査
- ソフトウェア品質保証： L2-9
- ツール： L2-5
- 資源と資金
- 欠陥予防： L5-4
 - 記述： 0-37
 - ソフトウェア統合管理： L3-39
 - グループ間調整： L3-85
 - 組織プロセス定義： L3-14

索引

- organization process focus L3-4
- peer reviews L3-95
- process change management L5-33
- quantitative process management L4-4
- requirements management L2-5
- software configuration management L2-75
- software product engineering L3-61

- software project planning L2-16
- software project tracking and oversight L2-32
- software quality assurance L2-62
- software quality management L4-21
- software subcontract management L2-46
- technology change management L5-21
- training program L3-27
- review
 - at milestones L2-39
 - commitments L2-31
 - defect prevention
 - project manager L5-15
 - senior management L5-14
 - software quality assurance L5-15
 - integrated software management
 - project manager L3-57
 - senior management L3-56
 - software quality assurance L3-57
 - intergroup coordination
 - project manager L3-92
 - senior management L3-91
 - software quality assurance L3-92
 - organization process definition
 - software quality assurance L3-23
- 組織プロセス重視： L3-4
- ピアレビュー： L3-95
- プロセス変更管理： L5-33
- 定量的プロセス管理： L4-4
- 要件管理： L2-5
- ソフトウェア構成管理： L2-75

- ソフトウェアプロダクトエンジニアリング： L3-61
- ソフトウェアプロジェクト計画： L2-16
- ソフトウェアプロジェクト進捗管理： L2-32
- ソフトウェア品質保証： L2-62
- ソフトウェア品質管理： L4-21
- ソフトウェア外注管理： L2-46
- 技術変更管理： L5-21
- トレーニングプログラム： L3-27
- レビュー
 - マイルストーンでの： L2-39
 - コミットメント： L2-31
 - 欠陥予防
 - プロジェクトマネージャ： L5-15
 - 上級管理層： L5-14
 - ソフトウェア品質保証： L5-15
 - ソフトウェア統合管理
 - プロジェクトマネージャ： L3-57
 - 上級管理層： L3-56
 - ソフトウェア品質保証： L3-57
 - グループ間調整
 - プロジェクトマネージャ： L3-92
 - 上級管理層： L3-91
 - ソフトウェア品質保証： L3-92
 - 組織プロセス定義
 - ソフトウェア品質保証： L3-23

- organization process focus
 - senior management L3-10
- peer reviews
 - software quality assurance L3-100
- periodic and event-driven 0-49
- periodic 0-48
- process change management
 - senior management L5-46
 - software quality assurance L5-46
- quantitative process management
 - project manager L4-16
 - senior management L4-15
 - software quality assurance L4-16
- requirements management
 - project manager L2-9
 - senior management L2-9
 - software quality assurance L2-9
- software configuration management
 - project manager L2-83
 - senior management L2-82
 - software quality assurance L2-83
 - software product engineering
 - project manager L3-80
 - senior management L3-80
 - software quality assurance L3-81
- software project commitments L2-13, L2-17
- software project planning
 - project manager L2-26
 - senior management L2-26
 - software quality assurance L2-27
- software project tracking and oversight
 - project manager L2-41
 - senior management L2-40
 - software quality assurance L2-41
- 組織プロセス重視
 - 上級管理層： L3-10
- ピアレビュー
 - ソフトウェア品質保証： L3-100
- 定期的およびイベント発生を契機とする： 0-49
- 定期的： 0-48
- プロセス変更管理
 - 上級管理層： L5-46
 - ソフトウェア品質保証： L5-46
- 定量的プロセス管理
 - プロジェクトマネージャ： L4-16
 - 上級管理層： L4-15
 - ソフトウェア品質保証： L4-16
- 要件管理
 - プロジェクトマネージャ： L2-9
 - 上級管理層： L2-9
 - ソフトウェア品質保証： L2-9
- ソフトウェア構成管理
 - プロジェクトマネージャ： L2-83
 - 上級管理層： L2-82
 - ソフトウェア品質保証： L2-83
- ソフトウェアプロダクトエンジニアリング
 - プロジェクトマネージャ： L3-80
 - 上級管理層： L3-80
 - ソフトウェア品質保証： L3-81
- ソフトウェアプロジェクトのコミットメント： L2-13, L2-17
- ソフトウェアプロジェクト計画
 - プロジェクトマネージャ： L2-26
 - 上級管理層： L2-26
 - ソフトウェア品質保証： L2-27
- ソフトウェアプロジェクト進捗管理
 - プロジェクトマネージャ： L2-41
 - 上級管理層： L2-40
 - ソフトウェア品質保証： L2-41

索引

- software quality assurance
 - project manager L2-69
 - senior management L2-68
 - software quality management
 - project manager L4-31
 - senior management L4-31
 - software quality assurance L4-32
 - software subcontract management
 - project manager L2-56
 - senior management L2-56
 - software quality assurance L2-57
 - technical aspects of intergroup coordination L3-90
 - technical aspects of software subcontract L2-52
 - technology change management
 - senior management L5-29
 - software quality assurance L5-29
 - tracking progress L2-38
 - training materials L3-33
 - training program
 - senior management L3-35
 - versus peer review 0-45
 - work products L3-90
- risks
- identification of L2-24
 - management of L3-52
 - managing at Level 4 0-17
 - tracking of L2-37
- role
- definition of 0-64
- S**
- senior management
 - approval of tailoring organization's standard software process L3-42
 - approval of waivers for organization's standard software process L3-42
 - ソフトウェア品質保証
 - プロジェクトマネージャ： L2-69
 - 上級管理層： L2-68
 - ソフトウェア品質管理
 - プロジェクトマネージャ： L4-31
 - 上級管理層： L4-31
 - ソフトウェア品質保証： L4-32
 - ソフトウェア外注管理
 - プロジェクトマネージャ： L2-56
 - 上級管理層： L2-56
 - ソフトウェア品質保証： L2-57
 - グループ間調整の技術面： L3-90
 - ソフトウェア外注の技術面： L2-52
 - 技術変更管理
 - 上級管理層： L5-29
 - ソフトウェア品質保証： L5-29
 - 進捗の確認： L2-38
 - トレーニング教材： L3-33
 - トレーニングプログラム
 - 上級管理層： L3-35
 - とピアレビュー： 0-45
 - 作業成果物： L3-90
- リスク
- 特定： L2-24
 - 管理： L3-52
 - レベル4での管理： 0-17
 - 進捗： L2-37
- 役割
- 定義： 0-64
- 上級管理層
- 組織の標準ソフトウェアプロセスのテラリングの承認： L3-42
 - 組織の標準ソフトウェアプロセスに対する免除の承認： L3-42

- oversight of organization process focus activities L3-3
- oversight of software process improvement L5-32
- sponsorship of organization process focus L3-2
- review
- commitment changes L2-7
 - commitments L2-31, L2-35
 - defect prevention L5-14
 - integrated software management L3-56
 - intergroup coordination L3-91
 - organization process focus L3-10
 - process change management L5-46
 - quantitative process management L4-15
 - requirements management L2-9
 - software configuration management L2-82
 - software product engineering L3-80
- software project commitments L2-13, L2-17
- software project planning L2-26
- software project tracking and oversight L2-40
- software quality assurance L2-61, L2-68
- software quality management L4-31
- software quality plan L4-25
- software subcontract management L2-56
- 組織プロセス重視活動の監督： L3-3
- ソフトウェアプロセス改善の監督： L5-32
- 組織プロセス重視への主催者としての態度： L3-2
- レビュー
- コミットメントの変更： L2-7
 - コミットメント： L2-31, L2-35
 - 欠陥予防： L5-14
 - ソフトウェア統合管理： L3-56
 - グループ間調整： L3-91
 - 組織プロセス重視： L3-10
 - プロセス変更管理： L5-46
 - 定量的プロセス管理： L4-15
 - 要件管理： L2-9
 - ソフトウェア構成管理： L2-82
- ソフトウェアプロダクトエンジニアリング： L3-80
- ソフトウェアプロジェクトのコミットメント： L2-13, L2-17
- ソフトウェアプロジェクト計画： L2-26
- ソフトウェアプロジェクト進捗管理： L2-40
- ソフトウェア品質保証： L2-61, L2-68
- ソフトウェア品質管理： L4-31
- ソフトウェア品質計画： L4-25
- ソフトウェア外注管理： L2-56

索引

- technology change management L5-29
- training program L3-35
- waivers from contractual software requirements L3-42
- role in oversight 0-48
- role in software process focus L3-3
- role in technology change management L5-19
- training for process change management L5-35
- senior manager
 - definition of 0-65
- software baseline
 - audit of L2-81
- software activities and software work products
 - deviations in L2-67
- software architecture
 - development of L3-70
- software baseline library
 - check-in/out procedure L2-80

 - creating products from L2-80
 - creation of L2-80
 - definition of L2-73
- software baselines
 - audit of L2-83
 - repository for L2-77
- software code
 - configuration management of L3-72
 - development of L3-71
 - peer review of L3-72
- software commitments
 - changes to L2-30, L2-35
- software configuration control board (SCCB)
 - 技術変更管理： L5-29
 - トレーニングプログラム： L3-35
 - 契約上のソフトウェア要件からの免除： L3-42
 - 監督の役割： 0-48
 - ソフトウェアプロセス重視の役割： L3-3
 - 技術変更管理での役割： L5-19

 - プロセス変更管理のトレーニング： L5-35
- 上級マネージャ
 - 定義： 0-65
- ソフトウェアベースライン
 - 監査： L2-81
- ソフトウェア活動とソフトウェア作業成果物
 - 逸脱： L2-67
- ソフトウェアアーキテクチャ
 - 開発： L3-70
- ソフトウェアベースラインライブラリ
 - チェックイン/チェックアウト手順： L2-80
 - 成果物の生成： L2-80
 - 生成： L2-80
 - 定義： L2-73
- ソフトウェアベースライン
 - 監査： L2-83
 - リポジトリ： L2-77
- ソフトウェアコード
 - 構成管理： L3-72
 - 開発： L3-71
 - ピアレビュー： L3-72
- ソフトウェアコミットメント
 - 変更： L2-30, L2-35
- ソフトウェア構成制御委員会 (SCCB)

- activities of L2-73
- software configuration management. *See also* ソフトウェア構成管理 (参照先: 管理と制御)
- manage and control
- models for L3-66
- organization's standard software process L3-17
- software code L3-72
- software design document L3-71
- software requirements document L3-69
- tools for developing and maintaining software L3-66
- Software Configuration Management L2-71 through L2-83
- audit of L2-73
- description of 0-21
- measurements for L2-82
- reports of activities L2-81
- resources and funding for L2-75
- review
- project manager L2-83
- senior management L2-82
- review and audit
- software quality assurance L2-83
- tools for L2-75
- software configuration management group L2-74
- audit of software baselines L2-83
- definition of 0-72
- monitor of subcontractor's configuration management L2-54
- software configuration management plan management and control of L2-77
- procedure for L2-76
- use of L2-77
- software design
- definition of L3-69
- 活動: L2-73
- ソフトウェア構成管理 (参照先: 管理と制御)
- モデル: L3-66
- 組織の標準ソフトウェアプロセス: L3-17
- ソフトウェアコード: L3-72
- ソフトウェア設計文書: L3-71
- ソフトウェア要件文書: L3-69
- ソフトウェアを開発し保守するツール: L3-66
- ソフトウェア構成管理: L2-71~L2-83
- 監査: L2-73
- 記述: 0-21
- 計測: L2-82
- 活動報告: L2-81
- 資源と資金: L2-75
- レビュー
- プロジェクトマネージャ: L2-83
- 上級管理層: L2-82
- レビューと監査
- ソフトウェア品質保証: L2-83
- ツール: L2-75
- ソフトウェア構成管理グループ: L2-74
- ソフトウェアベースラインの監査: L2-83
- 定義: 0-72
- 外注先のソフトウェア構成管理のモニター: L2-54
- ソフトウェア構成管理計画
- 管理と制御: L2-77
- 手順: L2-76
- 使用: L2-77
- ソフトウェア設計
- 定義: L3-69

索引

- requirements for L3-69
- software design document L3-71
- software development L2-15
- software development plan
 - basis for tracking project L2-30
 - definition of L2-14, 0-61
 - description of L2-11
 - deviations from L2-30
 - documentation of L2-31
 - for software project tracking and oversight L2-33
 - for subcontractor L2-51
 - incorporation of defect prevention activities L5-3
 - management and control of L2-13, L2-19, L2-34
 - procedure for L2-18
 - refinement of L2-39
 - relation to project's defined software process L3-37
 - revision of L2-33
 - software quality assurance group and L2-65
- software documentation
 - management and control of L3-77
- software effort and costs
 - management of L3-48
 - tracking of L2-36
- software engineering group
 - definition of 0-71
 - participation in software project planning L2-17
 - participation on project proposal team L2-16
 - review of project tracking L2-38
- 要件： L3-69
- ソフトウェア設計文書： L3-71
- ソフトウェア開発： L2-15
- ソフトウェア開発計画
 - プロジェクトを進捗確認する基盤： L2-30
 - 定義： L2-14, 0-61
 - 記述： L2-11
 - 逸脱： L2-30
 - 文書： L2-31
 - ソフトウェアプロジェクト進捗管理： L2-33
 - 外注先への： L2-51
 - 欠陥予防活動の導入： L5-3
- 管理と制御： L2-13, L2-19, L2-34
- 手順： L2-18
- 改良： L2-39
- プロジェクトの定義されたソフトウェアプロセスとの関係： L3-37
- 改訂： L2-33
- ソフトウェア品質保証グループ： L2-65
- ソフトウェア文書化
 - 管理と制御： L3-77
- ソフトウェア工数とコスト
 - 管理： L3-48
 - 進捗： L2-36
- ソフトウェアエンジニアリンググループ
 - 定義： 0-71
 - ソフトウェアプロジェクト計画への参加： L2-17
 - プロジェクト提案チームへの参加： L2-16
 - プロジェクト進捗確認のレビュー： L2-38

review of system requirements allocated to software L2-5	ソフトウェアに割り当てられたシステム要件のレビュー： L2-5
role in establishing system requirements L3-86	システム要件を確立する役割： L3-86
training for process change management L5-34	プロセス変更管理のトレーニング： L5-34
training for software quality management L4-22	ソフトウェア品質管理のトレーニング： L4-22
software engineering process group	ソフトウェアエンジニアリングプロセスグループ
coordination of software process improvement L5-36	ソフトウェアプロセス改善の調整： L5-36
definition of 0-71	定義： 0-71
establishment of L3-4, 0-15	確立： L3-4, 0-15
relation with technology change management L5-20	技術変更管理との関係： L5-20
responsibility for organization's standard software process L3-14	組織の標準ソフトウェアプロセスに対する責任： L3-14
responsibility of L3-1	責任： L3-1
review of changes to tailoring guidelines L3-20	テーラリングガイドラインの変更のレビュー： L3-20
review of quantitative process management plan L4-7	定量的プロセス管理のレビュー： L4-7
review of tailoring organization's standard software process L3-42	組織の標準ソフトウェアプロセスのテーラリングのレビュー： L3-42
software engineering technical activities L2-37	ソフトウェアエンジニアリングの技術的活動： L2-37
software life cycle	ソフトウェアライフサイクル
definition of 0-55	定義： 0-55
documentation of L3-18	文書： L3-18
identification of L2-17	特定： L2-17
management and control of L3-19	管理と制御： L3-19
peer review of L3-19	ピアレビュー： L3-19
selection of L3-41	選択： L3-41
software management	ソフトウェア管理
use at Level 2 0-15	レベル2での使用： 0-15
software manager	ソフトウェアマネージャ
negotiation of commitments L2-13	コミットメントの協議： L2-13

索引

- orientation for training program L3-29
- project tracking responsibility L2-32
- responsibility for software work products L2-15
- review
 - allocated requirements L2-3
 - project tracking L2-38
 - software development plan L2-19
 - statement of work L2-15
- training for process change management L5-34
- software planning data L2-25
- software process
- software process architecture
 - definition of L3-18, 0-54
- software process assessment (SPA) L3-6
- software process assets
 - development of at Level 3 0-21
 - examples of 0-53
- software process capability
 - definition of 0-10
- software process database
 - coordination at organization level L3-8
 - establishing and maintaining L3-20
 - management and control of L3-21
 - use for planning and estimating L3-46
 - use of at Level 4 0-16
- software process definition. *See also* Organization Process Focus
 - relation to Capability Maturity Model 0-50
- software process element
- トレーニングプログラムのオリエンテーション： L3-29
- プロジェクト進捗確認の責任： L2-32
- ソフトウェア作業成果物の責任： L2-15
- レビュー
 - 割り当てられた要件： L2-3
 - プロジェクト進捗確認： L2-38
 - ソフトウェア開発計画： L2-19
 - 作業内容記述書： L2-15
- プロセス変更管理のトレーニング： L5-34
- ソフトウェア計画データ： L2-25
- ソフトウェアプロセス
- ソフトウェアプロセスアーキテクチャ
 - 定義： L3-18, 0-54
- ソフトウェアプロセスアセスメント (SPA) : L3-6
- ソフトウェアプロセス資産
 - レベル3での開発： 0-21
 - 例： 0-53
- ソフトウェアプロセス能力
 - 定義： 0-10
- ソフトウェアプロセスデータベース
 - 組織レベルでの調整： L3-8
 - 確立と維持： L3-20
 - 管理と制御： L3-21
 - 計画と見積りでの使用： L3-46
 - レベル4での使用： 0-16
- ソフトウェアプロセス定義 (参照先：組織プロセス重視)
 - 能力成熟度モデルとの関連： 0-50
- ソフトウェアプロセス要素

- definition of 0-55
software process focus
 coordination of L3-7
software process improvement. *See also*
Process Change Management and Organization
Process Focus
 feedback on L5-44
 implementation of L5-42
 oversight by senior management L5-32
 pilot installation L5-42
 plans for L5-37
 policy for L5-32
 program for L5-35
 records of L5-43
 teams for process areas L5-41
 update of training for L5-43
software process improvement plan
 items in L5-38
 management and control of L5-37
 peer review of L5-37
 requirements for L5-37
software process improvement proposal
 documentation of L5-10
 procedure for handling L5-39
software process performance
 assessment of L3-6
 definition of 0-23
software product
 definition of 0-60
 software quality assurance audit of
 L2-66
Software Product Engineering L3-59 through
L3-82
 description of 0-22
 measurements for L3-79
- 定義： 0-55
ソフトウェアプロセス重視
調整： L3-7
ソフトウェアプロセス改善（参照先：プロセス
変更管理と組織プロセス重視）
- フィードバック： L5-44
実装： L5-42
上級管理層による監督： L5-32
試行的な導入： L5-42
計画： L5-37
方針： L5-32
プログラム： L5-35
記録： L5-43
プロセスエリア毎のチーム： L5-41
トレーニングの更新： L5-43
ソフトウェアプロセス改善計画
項目： L5-38
管理と制御： L5-37
ピアレビュー： L5-37
要件： L5-37
ソフトウェアプロセス改善提案
文書： L5-10
操作手順： L5-39
ソフトウェアプロセス実績
アセスメント： L3-6
定義： 0-23
ソフトウェア成果物
定義： 0-60
ソフトウェア品質保証監査： L2-66
ソフトウェアプロダクトエンジニアリ
ング： L3-59～L3-82
記述： 0-22
計測： L3-79

索引

- orientation for managers L3-64
- orientation for technical staff L3-64
- policy for L3-60
- resources and funding for L3-61
- review
 - project manager L3-80
 - senior management L3-80
- review and audit
 - software quality assurance L3-81
- tools for L3-61
- training for L3-63
- software project
- software project commitments
 - review of L2-13, L2-17
- software project data
 - recording of L2-38
- software project L2-14
 - commitment for L2-12
 - management of L3-44
 - review of L3-55
- Software Project Planning L2-11 through L2-27
 - coordination of L2-15
 - description of 0-20
 - measurements for L2-25
 - overall project planing and L2-17
 - policy for L2-12
 - resources and funding for L2-16
 - responsibility for L2-12
 - review
 - project manager L2-26
 - senior management L2-26
 - review and audit
 - software quality assurance L2-27
 - tools for L2-16
- マネージャへのオリエンテーション：
L3-64
- 技術要員へのオリエンテーション： L3-64
- 方針： L3-60
- 資源と資金： L3-61
- レビュー
 - プロジェクトマネージャ： L3-80
 - 上級管理層： L3-80
- レビューと監査
 - ソフトウェア品質保証： L3-81
- ツール： L3-61
- トレーニング： L3-63
- ソフトウェアプロジェクト
- ソフトウェアプロジェクトコミットメント
 - レビュー： L2-13, L2-17
- ソフトウェアプロジェクトデータ
 - 記録： L2-38
- ソフトウェアプロジェクト： L2-14
 - コミットメント： L2-12
 - 管理： L3-44
 - レビュー： L3-55
- ソフトウェアプロジェクト計画： L2-11～L2-27
 - 調整： L2-15
 - 記述： 0-20
 - 計測： L2-25
 - 全体プロジェクト計画： L2-17
 - 方針： L2-12
 - 資源と資金： L2-16
 - 責任： L2-12
 - レビュー
 - プロジェクトマネージャ： L2-26
 - 上級管理層： L2-26
 - レビューと監査
 - ソフトウェア品質保証： L2-27
 - ツール： L2-16

- training for L2-16
- Software Project Tracking and Oversight L2-29 through L2-42
- assignment of work tasks L2-31
 - description of 0-20
 - measurements for L2-39
 - policy for L2-30
 - resources and funding for L2-32
 - review
 - project manager L2-41
 - senior management L2-40
 - review and audit
 - software quality assurance L2-41
 - software development plan for L2-33
 - tools for L2-32
 - training for L2-32
- Software Quality Assurance L2-59 through L2-69
- description of 0-21
 - manager for L2-62
 - measurements for L2-68
 - orientation in L2-63
 - policy for L2-60
 - resources and funding for L2-62
 - review
 - independent experts L2-69
 - project manager L2-69
 - senior management L2-68
 - tools for L2-62
 - training for L2-62
- software quality assurance group L2-61
- activities of L2-65
 - audit of software products L2-66
 - coordination with customer L2-67
 - definition of 0-72
 - monitor of subcontractor's quality assurance L2-53
 - reports of activities L2-67
- トレーニング： L2-16
- ソフトウェアプロジェクト進捗管理： L2-29 ~L2-42
- 作業タスクの割り当て： L2-31
 - 記述： 0-20
 - 計測： L2-39
 - 方針： L2-30
 - 資源と資金： L2-32
 - レビュー
 - プロジェクトマネージャ： L2-41
 - 上級管理層： L2-40
 - レビューと監査
 - ソフトウェア品質保証： L2-41
 - ソフトウェア開発計画： L2-33
 - ツール： L2-32
 - トレーニング： L2-32
- ソフトウェア品質保証： L2-59 ~L2-69
- 記述： 0-21
 - マネージャ： L2-62
 - 計測： L2-68
 - オリエンテーション： L2-63
 - 方針： L2-60
 - 資源と資金： L2-62
 - レビュー
 - 独立した専門家： L2-69
 - プロジェクト： L2-69
 - 上級管理層： L2-68
 - ツール： L2-62
 - トレーニング： L2-62
- ソフトウェア品質保証グループ： L2-61
- 活動： L2-65
 - ソフトウェア成果物の監査： L2-66
 - 顧客との調整： L2-67
 - 定義： 0-72
 - 外注先のソフトウェア構成管理のモニター： L2-53
 - 活動報告： L2-67

索引

- review
- defect prevention L5-15
 - integrated software management L3-57
 - intergroup coordination L3-92
 - organization process definition L3-23
 - peer reviews L3-100
 - process change management L5-46
 - quantitative process management L4-16
 - requirements management L2-9
 - software configuration management L2-83
 - software engineering activities L2-66
 - software product engineering L3-81

 - software project planning L2-27

 - software project tracking and oversight L2-41
 - software quality management L4-32
 - software subcontract management L2-57
 - technology change management L5-29
 - software development plan and L2-65
 - software quality assurance plan L2-63
 - management and control of L2-64
 - software quality assurance review
 - description of 0-49
 - software quality goals
 - actions to resolve conflicts L4-30
 - documentation of L4-28
- レビュー
- 欠陥予防： L5-15
 - ソフトウェア統合管理： L3-57

 - グループ間調整： L3-92
 - 組織プロセス定義： L3-23

 - ピアレビュー： L3-100
 - プロセス変更管理： L5-46
 - 定量的プロセス管理： L4-16

 - 要件管理： L2-9
 - ソフトウェア構成管理： L2-83

 - ソフトウェアエンジニアリング活動： L2-66
 - ソフトウェアプロダクトエンジニアリング： L3-81
 - ソフトウェアプロジェクト計画： L2-27
 - ソフトウェアプロジェクト進捗管理： L2-41
 - ソフトウェア品質管理： L4-32
 - ソフトウェア外注管理： L2-57

 - 技術変更管理： L5-29
 - ソフトウェア開発計画： L2-65
 - ソフトウェア品質保証計画： L2-63

 - 管理と制御： L2-64
 - ソフトウェア品質保証レビュー
 - 記述： 0-49
 - ソフトウェア品質目標
 - 対立を解消するための処置： L4-30
 - 文書化： L4-28

Software Quality Management L4-19 through L4-32	ソフトウェア品質管理： L4-19 ~ L4-32
description of 0-23	記述： 0-23
measurements for L4-31	計測： L4-31
policy for L4-20	方針： L4-20
resources and funding for L4-21	資源と資金： L4-21
review	レビュー
project manager L4-31	プロジェクトマネージャ： L4-31
senior management L4-31	上級管理層： L4-31
review and audit	レビューと監査
software quality assurance L4-32	ソフトウェア品質保証： L4-32
tools for L4-21	ツール： L4-21
training for software engineering group L4-22	ソフトウェアエンジニアリンググループへのトレーニング： L4-22
training to implement L4-22	実装のためのトレーニング： L4-22
software quality plan	ソフトウェア品質計画
documentation of quality goals L4-28	品質目標の文書化： L4-28
items in L4-25	項目： L4-25
management and control of L4-25	管理と制御： L4-25
peer review of L4-24	ピアレビュー： L4-24
requirements for L4-23	要件： L4-23
review by senior management L4-25	上級管理層によるレビュー： L4-25
software replanning data	ソフトウェア再計画データ
management and control of L2-38	管理と制御： L2-38
software requirements	ソフトウェア要件
changes to L3-69	変更： L3-69
development of L3-66	開発： L3-66
software requirements document	ソフトウェア要件文書
configuration management of L3-69	構成管理： L3-69
peer review of L3-68	ピアレビュー： L3-68
software risk	ソフトウェアリスク
software risk management plan	ソフトウェアリスク管理計画
management and control of L3-54	管理と制御： L3-54
peer review of L3-54	ピアレビュー： L3-54
plan for managing L3-53	管理のための計画： L3-53
software schedule	ソフトウェアスケジュール

索引

- management of L3-51
- procedure for L2-23
- tracking of L2-37
- software size
 - management of L3-47
- software subcontract
 - policy for managing L2-44
 - responsibility for managing L2-45
- Software Subcontract Management L2-43 through L2-57
 - basis for L2-50
 - contract for L2-50
 - description of 0-20
 - measurements for L2-55
 - orientation in L2-46
 - responsibility for L2-46
 - review
 - project manager L2-56
 - senior management L2-56
 - review and audit
 - software quality assurance L2-57
 - training for L2-46
- software subcontractor
 - performance evaluation L2-55
 - procedure for selection of L2-49
 - procedure for reviews with L2-53
 - quality goals for L4-30
 - status reviews with L2-51
 - technical reviews with L2-52
- software subcontractor's statement of work
 - changes to L2-51
 - acceptance testing of products L2-55
- software task
 - kick-off meeting for defect prevention L5-6
- software task leader
- 管理： L3-51
- 手順： L2-23
- 進捗： L2-37
- ソフトウェア規模
 - 管理： L3-47
- ソフトウェア外注
 - 管理方針： L2-44
 - 管理に対する責任： L2-45
- ソフトウェア外注管理： L2-43～L2-57
 - 基盤： L2-50
 - 契約： L2-50
 - 記述： 0-20
 - 計測： L2-55
 - オリエンテーション： L2-46
 - 責任： L2-46
 - レビュー
 - プロジェクトマネージャ： L2-56
 - 上級管理層： L2-56
 - レビューと監査
 - ソフトウェア品質保証： L2-57
 - トレーニング： L2-46
- ソフトウェア外注先
 - 実績評価： L2-55
 - 選定手順： L2-49
 - レビュー手順： L2-53
 - 品質目標： L4-30
 - 状況レビュー： L2-51
 - 技術レビュー： L2-52
- ソフトウェア外注の作業内容記述書
 - 変更： L2-51
 - 成果物の検収テスト： L2-55
- ソフトウェアタスク
 - 欠陥予防の立ち上げ会議： L5-6
- ソフトウェアタスクリーダー

- definition of 0-68
- software testing L3-72
- software test plan
 - peer review of L3-74
 - requirements for L3-75
- software testing criteria
 - review of L3-72
- software work product
 - audit of L2-66
 - consistency of L3-78
 - definition of 0-60
 - estimating size of L2-21
 - evaluation of L3-98
 - responsibility for L2-15, L2-31
 - tracking size of L2-35
- software-related group
 - definition of 0-71
- sponsorship
 - description of 0-36
 - organization process focus L3-2
- staff
 - definition of 0-68
- stage
 - definition of 0-58
- standards
 - for training courses L3-33
 - software quality assurance involvement in developing L2-65
- statement of work
 - subcontract L2-47
- statement of work for software project
 - management and control of L2-15
 - review of L2-15
- subcontract. *See also* software subcontract
- subcontract manager
 - responsibilities of L2-45
- subcontract statement of work L2-47
 - management and control of L2-48
- 定義： 0-68
- ソフトウェアテスト： L3-72
- ソフトウェアテスト計画
 - ピアレビュー： L3-74
 - 要件： L3-75
- ソフトウェアテスト基準
 - レビュー： L3-72
- ソフトウェア作業成果物
 - 監査： L2-66
 - 首尾一貫性： L3-78
 - 定義： 0-60
 - 規模の見積り： L2-21
 - 審査： L3-98
 - 責任： L2-15, L2-31
 - 規模の進捗確認： L2-35
- ソフトウェア関連グループ
 - 定義： 0-71
- 主催者としての態度
 - 記述： 0-36
 - 組織プロセス重視： L3-2
- 要員
 - 定義： 0-68
- ステージ
 - 定義： 0-58
- 標準
 - トレーニングコース： L3-33
 - 開発に関わるソフトウェア品質保証： L2-65
- 作業内容記述書
 - 外注： L2-47
- ソフトウェアプロジェクトの作業内容記述書
 - 管理と制御： L2-15
 - レビュー： L2-15
- 外注（参照先：ソフトウェア外注）
- 外注管理者
 - 責任： L2-45
- 外注の作業内容記述書： L2-47
 - 管理と制御： L2-48

索引

subcontracted work	外注される作業
procedure for L2-47	手順： L2-47
selection of L2-47	選択： L2-47
subpractices	サブプラクティス
description of 0-31	記述： 0-31
supplementary information	補足情報
description of 0-32	記述： 0-32
system and acceptance testing L3-75	システムテストと検収テスト： L3-75
system documentation	システム文書
development of L3-76	開発： L3-76
peer review of L3-77	ピアレビュー： L3-77
system engineering group	ソフトウェアエンジニアリンググループ
definition of 0-72	定義： 0-72
system requirements	システム要件
coordination to establish L3-86	確立のための調整： L3-86
responsibility for L2-3	責任： L2-3
system requirements allocated to software.	ソフトウェアに割り当てられたシステム要件
See also allocated requirements	(参照先：割り当てられた要件)
changes to L2-7	変更： L2-7
definition of L2-1	定義： L2-1
description of 0-43	記述： 0-43
documentation of L2-4	文書： L2-4
examples of L2-4	例： L2-4
management and control of documentation	文書の管理と制御： L3-78
for L3-78	
policy for managing L2-2	管理の方針： L2-2
review of L2-5	レビュー： L2-5
system test group	システムテストグループ
definition of 0-72	定義： 0-72

T

Taguchi's method for robust design L4-24	頑健な設計のための田口メソッド： L4-24
task	タスク
definition of 0-59	定義： 0-59
team	チーム

- defect prevention L5-3
- interdisciplinary engineering L3-84
- technology
- technology change
- Technology Change Management L5-17 through L5-30
 - description of 0-24
 - group for L5-20
 - measurements for L5-28
 - plan for L5-23
 - policy for L5-18
 - resources and funding for L5-21
 - review
 - senior management L5-29
 - review and audit
 - software quality assurance L5-29
 - senior management role in L5-19
 - tools for L5-21
 - training for L5-23
- technology change management plan L5-23
 - peer review of L5-24
 - data to support L5-22
 - identification of areas for L5-24
 - pilot effort for L5-27
 - relation to Capability Maturity Model 0-62
- test plans
 - management and control of L3-74
- testing
 - of software L3-72
 - of system for acceptance L3-75
- tools
 - building and maintaining software L3-60
 - defect prevention L5-4
- 欠陥予防： L5-3
- 分野横断的なエンジニアリング： L3-84
- 技術
- 技術変更
- 技術変更管理： L5-17～L5-30
 - 記述： 0-24
 - グループ： L5-20
 - 計測： L5-28
 - 計画： L5-23
 - 方針： L5-18
 - 資源と資金： L5-21
 - レビュー
 - 上級管理層： L5-29
 - レビューと監査
 - ソフトウェア品質保証： L5-29
 - 上級管理層の役割： L5-19
 - ツール： L5-21
 - トレーニング： L5-23
- 技術変更管理計画： L5-23
 - ピアレビュー： L5-24
 - 支援するためのデータ： L5-22
 - エリアの特定： L5-24
 - 試行作業： L5-27
 - 能力成熟度モデルとの関連： 0-62
- テスト計画
 - 管理と制御： L3-74
- テスト
 - ソフトウェア： L3-72
 - システムの検収： L3-75
- ツール
 - ソフトウェアの構築と保守： L3-60
- 欠陥予防： L5-4

索引

- intergroup coordination L3-85
- organization process definition L3-14
- organization process focus L3-5
- process change management L5-34
- quantitative process measurement L4-5
- requirements management L2-5
- selection of for project L3-65
- software configuration management L2-75
- software product engineering L3-61

- software project planning L2-16
- software project tracking and oversight L2-32
- software quality assurance L2-62
- software quality management L4-21
- software subcontract management L2-46
- technology change management L5-21
- training L3-28
- tools for developing and maintaining software
 - configuration management of L3-66
- tracking
 - critical computer resources L2-36
 - risks L2-37
 - size of software work products L2-35
 - Software engineering technical activities L2-37
 - software effort and costs L2-36
 - versus managing 0-45
- training. *See also* orientation and required training

- グループ間調整： L3-85
- 組織プロセス定義： L3-14
- 組織プロセス重視： L3-5
- プロセス変更管理： L5-34
- 定量的プロセス計測： L4-5
- 要件管理： L2-5
- プロジェクトの選択： L3-65
- ソフトウェア構成管理： L2-75

- ソフトウェアプロダクトエンジニアリング： L3-61
- ソフトウェアプロジェクト計画： L2-16
- ソフトウェアプロジェクト進捗管理： L2-32
- ソフトウェア品質保証： L2-62
- ソフトウェア品質管理： L4-21
- ソフトウェア外注管理： L2-46
- 技術変更管理： L5-21
- トレーニング： L3-28
- ソフトウェアを開発し保守するツール
 - 構成管理： L3-66
- 進捗
 - 重要なコンピュータ資源： L2-36
 - リスク： L2-37
 - ソフトウェア作業成果物の規模： L2-35
 - ソフトウェアエンジニアリングの技術的活動： L2-37
 - 工数とコスト： L2-36
 - と管理： 0-45
- トレーニング（参照先：オリエンテーションと必修トレーニング）

- coordination across organization L3-8
- description of 0-38
- development of courses for L3-33
- facilities for L3-28
- integrated software management L3-39
- intergroup coordination L3-85
- organization process definition L3-14
- organization process focus L3-5
- plan for project L3-29, L3-30
- policy for L3-26
- records of L3-34
- review of materials for L3-33
- software configuration management L2-76
- software product engineering L3-63

- software project planning L2-16
- software project tracking and oversight L2-32
- software quality assurance L2-62
- software subcontract management L2-46
- tools for L3-28
- update based on software process improvement L5-43
- waiver for L3-34
- training group
 - definition of 0-73
 - skills and knowledge of L3-28
- training materials
 - management and control of L3-34
- training plan
 - items included in L3-32
 - management and control of L3-31

- 組織横断的な調整： L3-8
- 記述： 0-38
- コースの開発： L3-33
- 施設： L3-28
- ソフトウェア統合管理： L3-39
- グループ間調整： L3-85
- 組織プロセス定義： L3-14
- 組織プロセス重視： L3-5
- プロジェクトの計画： L3-29, L3-30
- 方針： L3-26
- 記録： L3-34
- 教材のレビュー： L3-33
- ソフトウェア構成管理： L2-76

- ソフトウェアプロダクトエンジニアリング： L3-63
- ソフトウェアプロジェクト計画： L2-16
- ソフトウェアプロジェクト進捗管理： L2-32
- ソフトウェア品質保証： L2-62
- ソフトウェア外注管理： L2-46
- ツール： L3-28
- ソフトウェアプロセス改善に基づく更新： L5-43
- 免除： L3-34
- トレーニンググループ
 - 定義： 0-73
 - スキルと知識： L3-28
- トレーニング教材
 - 管理と制御： L3-34
- トレーニング計画
 - 含まれる項目： L3-32
 - 管理と制御： L3-31

索引

Training Program L3-25 through L3-36
description of 0-22
group for L3-27
independent evaluation of L3-36
measurements for L3-34
orientation in L3-29
resources and funding for L3-27
review
 senior management L3-35
review and audit of L3-36
training program elements
 items included in L3-28
training vehicles
 examples of L3-26

V

Verifying Implementation
description of 0-28

W

work products
review of L3-90

トレーニングプログラム : L3-25 ~ L3-36
記述 : 0-22
グループ : L3-27
独立の評価 : L3-36
計測 : L3-34
オリエンテーション : L3-29
資源と資金 : L3-27
レビュー
 上級管理層 : L3-35
 レビューと監査 : L3-36
トレーニングプログラム要素
 項目 : L3-28
トレーニング手段
 例 : L3-26

履行検証
記述 : 0-28

作業成果物
レビュー : L3-90

あ

アセスメント

ソフトウェアプロセス： L3-6

新しい技術

情報： L5-25

選択： L5-26

組織内への移転： L3-8

え

エンジニアリンググループ

依存関係： L3-89

お

オリエンテーション（参照先：トレーニングと必修トレーニング）

グループ間調整： L3-86

記述： 0-39

組織プロセス重視： L3-6

ソフトウェアプロダクトエンジニアリングのマネージャ： L3-64

ソフトウェアプロダクトエンジニアリングの技術要員： L3-64

定量的プロセス管理： L4-6

トレーニングプログラム： L3-29

か

外注（参照先：ソフトウェア外注）

外注される作業

手順： L2-47

選択： L2-47

外注の作業内容記述書： L2-47

管理と制御： L2-48

外注管理者

責任： L2-45

開発構成管理

記述： 0-46

活動

定義： 0-59

頑健な設計のための田口メソッド： L4-24

定義： 0-59

監査

技術変更管理： L5-29

グループ間のコミットメント： L3-92

欠陥予防： L5-15

組織プロセス定義： L3-23

ソフトウェアプロジェクト計画： L2-27

ソフトウェアプロジェクト進捗管理： L2-41

ソフトウェアプロダクトエンジニアリング： L3-81

ソフトウェア外注管理： 57

ソフトウェア構成管理： L2-83

ソフトウェア作業成果物： L2-66

ソフトウェア統合管理： L3-57

ソフトウェア品質管理： L4-32

ソフトウェアベースライン： L2-81

定量的プロセス管理： L4-16

ピアレビュー： L3-100

プロセス変更管理： L5-46

要件管理： L2-9

管理

と進捗の確認： 0-45

管理されたレベル

記述： 0-16

管理者

外注： L2-45

ソフトウェア品質保証： L2-62

定義： 0-64

管理と制御（参照先：ソフトウェア構成管理と構成管理）

外注の作業内容記述書： L2-48

欠陥予防データ： L5-11

ソフトウェア開発計画： L2-13, L2-19, L2-34

ソフトウェア計画データ： L2-25

ソフトウェア構成管理計画： L2-77

索引

ソフトウェア再計画データ： L2-38
ソフトウェアに割り当てられたシステム要件： L2-7
ソフトウェア品質計画： L4-25
ソフトウェア品質保証： L2-64
ソフトウェアプロジェクトの作業内容記述書： L2-15
ソフトウェアプロセス改善計画： L5-37
定量的プロセス管理計画： L4-7
と構成管理： 0-46
非遵守問題： L2-67
プロセス実施ベースライン： L4-12
プロセス能力ベースライン： L4-14

慣例

記述： 0-36
ソフトウェア開発計画からの逸脱： L2-30

き

キープラクティス

意図： 0-35
解釈： 0-27, 0-75
慣例の使用： 0-35
記述： 0-26, 0-31
構造： 0-31
定義： 0-12
履行： 0-35
例ページ： 0-33

キープロセスエリア

記述： 0-18
区分： 0-26
成熟度レベルの重なり： 0-24
成熟度レベルの特定： 0-19
定義： 0-10
レベル 2 の記述： 0-20 ~ 0-21
レベル 3 の記述： 0-21 ~ 0-23
レベル 4 の記述： 0-23
レベル 5 の記述： 0-23 ~ 0-24

技術

技術変更

技術変更管理： L5-17 ~ L5-30

記述： 0-24
グループ： L5-20
計画： L5-23
計測： L5-28
資源と資金： L5-21
上級管理層の役割： L5-19
ツール： L5-21
トレーニング： L5-23
方針： L5-18
レビュー
 上級管理層： L5-29
レビューと監査
 ソフトウェア品質保証： L5-29

技術変更管理計画： L5-23

エリアの特定： L5-24
支援するためのデータ： L5-22
試行作業： L5-27
能力成熟度モデルとの関連： 0-62
ピアレビュー： L5-24

記録

ソフトウェアプロセス改善： L5-43

く

グループ

技術変更管理： L5-20
組織プロセス重視： L3-3
ソフトウェア構成管理： L2-74
ソフトウェア品質保証： L2-61
定義： 0-70
定量的プロセス管理： L4-4
トレーニング： L3-27
グループ間コミットメント
 計画： L3-88
グループ間調整： L3-83 ~ L3-92
オリエンテーション： L3-86
記述： 0-22
計測： L3-91
資源と資金： L3-85

ツール： L3-85
 レビュー
 プロジェクトマネージャ： L3-92
 上級管理層： L3-91
 レビューと監査
 ソフトウェア品質保証： L3-92
 グループ間の課題
 手順： L3-90

け

計画

記述： 0-40
 技術変更管理： L5-23
 グループ間調整の伝達： L3-88
 欠陥予防： L5-5
 公式： 0-41
 組織プロセス重視の計画： L3-7
 ソフトウェアエンジニアリング： L2-25
 ソフトウェア開発： L2-14
 ソフトウェア構成管理： L2-76
 ソフトウェア品質： L4-23
 ソフトウェア品質保証： L2-63
 定量的プロセス管理計画： L4-6
 トレーニング： L3-29
 ピアレビュー： L3-97
 非公式： 0-41
 プロセス変更管理： L5-37
 計画（参照先：ソフトウェアプロジェクト計画）
 計測（参照先：定量的プロセス管理）
 技術変更管理： L5-28
 グループ間調整： L3-91
 欠陥予防： L5-13
 組織プロセス重視： L3-9
 組織プロセス定義： L3-22
 ソフトウェアエンジニアリング活動：
 L3-80
 ソフトウェア外注管理： L2-55
 ソフトウェア構成管理： L2-82

ソフトウェア統合管理： L3-56
 ソフトウェア品質管理： L4-31
 ソフトウェア品質保証： L2-68
 ソフトウェアプロジェクト計画： L2-25
 ソフトウェアプロジェクト進捗管理：
 L2-39
 ソフトウェアプロダクトエンジニアリン
 グ： L3-79
 定量的プロセス管理： L4-15
 トレーニングプログラム： L3-34-35
 ピアレビュー： L3-99
 プロセス変更管理： L5-45
 要件管理： L2-8
 レベル4の制度化： 0-16
 計測データ
 収集： L4-9
 組織のプログラム： L4-5
 分析技法： L4-11
 保存： L4-10
 例： L4-9
 計測と分析
 定義： 0-28
 契約
 ソフトウェア外注管理： L2-50
 欠陥
 データの使用： L3-78
 根本原因の例： L5-8
 レベル5で予防： 0-17
 欠陥予防： L5-1～L5-15
 活動の例： L5-4
 記述： 0-24
 計測： L5-13
 原因分析会議： L5-7
 資源と資金： L5-4
 組織のためのチーム： L5-3
 組織方針： L5-2
 タスクの立ち上げ会議： L5-6
 ツール： L5-4
 データの管理と制御： L5-11

索引

データの文書化： L5-11
トレーニング： L5-4
フィードバック： L5-12
プロジェクト計画： L5-5
プロジェクトのためのチーム： L5-3
プロジェクト方針： L5-2
レビュー
 上級管理層： L5-14
 プロジェクトマネージャ： L5-15
レビューと監査
 ソフトウェア品質保証： L5-15
欠陥予防計画
 ピアレビュー： L5-5
欠陥予防チーム
 活動： L5-8
 確立： L5-3
原因分析会議
 手順： 79
検収テスト
 外注した作業成果物： L2-55

こ

工数とコスト
 見積り： L2-22
構成アイテム/ユニット
 記録状況： L2-80
 定義： L2-73
 特定： L2-78
 変更の手順： L2-79
構成管理（参照先：ソフトウェア構成管理と
“管理と制御”）
 開発： 0-46
 と管理制御： 0-46
 ベースライン： 0-46
構成管理ライブラリシステム
 確立： L2-77
 定義： L2-73
顧客
 内部の場合と外部の場合： 0-44

マイルストーンレビュー： L2-39
例： L2-1
レビュー
 契約上のソフトウェア要件からの免
 除： L3-42
 ソフトウェアテスト基準： L3-72
 ソフトウェアテスト計画： L3-75
 ソフトウェア品質保証： L2-67
 ソフトウェア要件文書： L3-68
レビューと承認
 システム文書： L3-77
コミットメント
 上級管理層によるレビュー： L2-31, L2-
 35
 ソフトウェアプロジェクト： L2-12
 変更： 7
コモンフィーチャ
 定義： 0-11, 0-27

さ

最適化するレベル
 記述： 0-17
作業成果物
 レビュー： L3-90
作業内容記述書
 外注： L2-47
サブプラクティス
 記述： 0-31

し

資金
 と予算： 0-37
資源と資金
 記述： 0-37
 技術変更管理： L5-21
 グループ間調整： L3-85
 欠陥予防： L5-4
 組織プロセス重視： L3-4
 組織プロセス定義： L3-14

- ソフトウェア外注管理： L2-46
ソフトウェア構成管理管理： L2-75
ソフトウェア統合管理： L3-39
ソフトウェア品質管理： L4-21
ソフトウェア品質保証： L2-62
ソフトウェアプロジェクト計画： L2-16
ソフトウェアプロジェクト進捗管理：
L2-32
ソフトウェアプロダクトエンジニアリン
グ： L3-61
定量的プロセス管理： L4-4
トレーニングプログラム： L3-27
ピアレビュー： L3-95
プロセス変更管理： L5-33
要件管理： L2-5
- 試行作業
技術変更： L5-27
- システムテストグループ
定義： 0-72
- システムテストと検収テスト： L3-75
- システム文書
開発： L3-76
ピアレビュー： L3-77
- システム要件
確立のための調整： L3-86
責任： L2-3
- 実施される活動
記述： 0-28
- 実施能力
記述： 0-28
- 実施のコミットメント
記述： 0-28
- 重要な依存関係： L3-89
- 重要なコンピュータ資源
管理： L3-50
進捗： L2-36
見積り： L2-23
見積りの基盤： L3-50
予備の容量の提供： L3-51
- 主催者としての態度
記述： 0-36
組織プロセス重視： L3-2
- 主導
記述： 0-36
- 上級管理層
監督の役割： 0-48
技術変更管理での役割： L5-19
組織プロセス重視活動の監督： L3-3
組織の標準ソフトウェアプロセスに対す
る免除の承認： L3-42
組織プロセス重視の主催者としての態
度： L3-2
組織の標準ソフトウェアプロセスのテー
ラリング： L3-42
ソフトウェアプロセス重視の役割： L3-3
ソフトウェアプロセス改善の監督： L5-32
プロセス変更管理のトレーニング： L5-35
レビュー
技術変更管理： L5-29
グループ間調整： L3-91
契約上のソフトウェア要件からの免
除： L3-42
欠陥予防： L5-14
コミットメント： L2-31, L2-35
コミットメントの変更： L2-7
組織プロセス重視： L3-10
ソフトウェア外注管理： L2-56
ソフトウェア構成管理： L2-82
ソフトウェア統合管理： L3-56
ソフトウェア品質管理： L4-31
ソフトウェア品質計画： L4-25
ソフトウェア品質保証： L2-61, L2-
68
ソフトウェアプロジェクト計画：
L2-26
ソフトウェアプロジェクト進捗管
理： L2-40
ソフトウェアプロジェクトのコミッ
トメント： L2-13, L2-17
ソフトウェアプロダクトエンジニア

索引

リング： L3-80
定量的プロセス管理： L4-15
トレーニングプログラム： L3-35
プロセス変更管理： L5-46
要件管理： L2-9

上級マネージャ
定義： 0-65

初期レベル
記述： 0-13

処置項目
データ： L5-11

処置提案
データ： L5-11

進捗
工数とコスト： L2-36
重要なコンピュータ資源： L2-36
ソフトウェアエンジニアリングの技術的
活動： L2-37
ソフトウェア作業成果物の規模： L2-35
と管理： 0-45
リスク： L2-37

す

ステージ
定義： 0-58

せ

成熟度質問票
能力成熟度モデルとの関係： 0-4

成熟度レベル
定義： 0-10
複数の重なり： 0-24
プロセス能力の関係： 0-12

是正処置
ソフトウェア開発計画からの逸脱： L2-30

前提要件
記述： 0-39

そ

組織のソフトウェアプロセスデータベース：
L4-13
定義： 0-56
定量的プロセス管理データの保存： L4-10
データの例： L3-47

組織の標準ソフトウェアプロセス
新しい技術の導入： L5-28
新しい技術の分析： L5-25
逸脱： L3-39
逸脱の免除： L3-42
開発の手順： L3-15
欠陥予防に基づく改訂： L5-12
組織レベルでの調整： L3-7
ソフトウェア構成管理： L3-17
ソフトウェアプロセス定義との関係： 0-
50
定義： 0-15, 0-54, 0-70
テーラリング： L3-41
テーラリングのガイドライン： L3-19,
0-56
能力分析： L4-3
ピアレビュー： L3-16
プロジェクトの計画と管理での使用：
L3-38
プロセス能力の傾向： L4-13
プロセス変更の導入： L5-43
ベースライン： L4-3, L4-13
変更： L4-15
方針： L3-12

組織プロセス重視： L3-1～L3-10
オリエンテーション： L3-6
管理支持： L3-2
キーグループの関与： L3-9
記述： 0-21
グループ： L3-3
計画： L3-7
計測： L3-9

- 資源と資金： L3-4
 上級管理層の監督レビュー： L3-3
 ツール： L3-5
 トレーニング： L3-5
 トレーニングの調整： L3-8
 方針： L3-2
 レビュー
 上級管理層： L3-10
 組織プロセス重視の計画
 ピアレビュー： L3-7
 組織プロセス定義： L3-11～L3-23
 記述： 0-21
 計測： L3-22
 資源と資金： L3-14
 ツール： L3-14
 トレーニング： L3-14
 方針： L3-12
 レビューと監査
 ソフトウェア品質保証： L3-23
 ソフトウェア文書化
 管理と制御： L3-77
 ソフトウェアアーキテクチャ
 開発： L3-70
 ソフトウェアエンジニアリングの技術的活動： L2-37
 ソフトウェアエンジニアリンググループ
 システム要件を確立する役割： L3-86
 ソフトウェアに割り当てられたシステム要件のレビュー： L2-5
 ソフトウェア品質管理のトレーニング： L4-22
 ソフトウェアプロジェクト計画への参加： L2-17
 定義： 0-71
 プロジェクト進捗確認のレビュー： L2-38
 プロジェクト提案チームへの参加： L2-16
 プロセス変更管理のトレーニング： L5-34
 ソフトウェアエンジニアリングプロセスグループ
 確立： L3-4, 0-15
 技術変更管理との関係： L5-20
 責任： L3-1
 組織の標準ソフトウェアプロセスに対する： L3-14
 組織の標準ソフトウェアプロセスのテラリングのレビュー： L3-42
 ソフトウェアプロセス改善の調整： L5-36
 定義： 0-71
 定量的プロセス管理計画のレビュー： L4-7
 テラリングガイドラインの変更のレビュー： L3-20
 ソフトウェア外注
 管理に対する責任： L2-45
 管理方針： L2-44
 ソフトウェア外注先
 技術レビュー： L2-52
 実績評価： L2-55
 状況レビュー： L2-51
 選定手順： L2-49
 品質目標： L4-30
 レビュー手順： L2-53
 ソフトウェア外注管理： L2-43～L2-57
 記述： 0-20
 基盤： L2-50
 計測： L2-55
 契約： L2-50
 責任： L2-46
 トレーニング： L2-46
 方向： L2-46
 レビュー
 上級管理層： L2-56
 プロジェクトマネージャ： L2-56
 レビューと監査
 ソフトウェア品質保証： L2-57
 ソフトウェア外注の作業内容記述書
 成果物の検収テスト： L2-55
 変更： L2-51
 ソフトウェア開発： L2-15

索引

- ソフトウェア開発計画
 - 逸脱： L2-30
 - 外注先への： L2-51
 - 改訂： L2-33
 - 改良： L2-39
 - 管理と制御： L2-13, : L2-19, L2-34
 - 記述： L2-11
 - 欠陥予防活動の導入： L5-3
 - ソフトウェア品質保証グループ： L2-65
 - ソフトウェアプロジェクト進捗管理： L2-33
 - 定義： L2-14, 0-61
 - 手順： L2-18
 - プロジェクトを進捗確認の基盤： L2-30
 - プロジェクトの定義されたソフトウェアプロセスとの関係： L3-37
 - 文書： L2-31
- ソフトウェア活動やソフトウェア作業成果物
 - 逸脱： L2-67
- ソフトウェア管理
 - レベル2で使用される： 0-15
- ソフトウェア規模
 - 管理： L3-47
- ソフトウェア計画データ： L2-25
- ソフトウェア工数とコスト
 - 管理： L3-48
 - 進捗： L2-36
- ソフトウェア構成管理： L2-71 ~ L2-83
 - 活動報告： L2-81
 - 監査： L2-73
 - 記述： 0-21
 - 計測： L2-82
 - 資源と資金： L2-75
 - ツール： L2-75
 - レビュー
 - 上級管理層： L2-82
 - プロジェクトマネージャ： L2-83
 - レビューと監査
 - ソフトウェア品質保証： L2-83
- ソフトウェア構成管理（参照先：管理と制御）
 - 組織の標準ソフトウェアプロセス： L3-17
 - ソフトウェアコード： L3-72
 - ソフトウェア設計文書： L3-71
 - ソフトウェア要件文書： L3-69
 - モデル： L3-66
 - ソフトウェアを開発し保守するツール： L3-66
- ソフトウェア構成管理グループ： L2-74
 - 外注先のソフトウェア構成管理のモニター： L2-54
 - ソフトウェアベースラインの監査： L2-83
 - 定義： 0-72
- ソフトウェア構成管理計画
 - 管理と制御： L2-77
 - 手順： L2-76
 - 使い方： L2-77
- ソフトウェア構成制御委員会 (SCCB)
 - 活動： L2-73
- ソフトウェアコード
 - 開発： L3-71
 - 構成管理： L3-72
 - ピアレビュー： L3-72
- ソフトウェアコミットメント
 - 変更： L2-30, L2-35
- ソフトウェア再計画データ
 - 管理と制御： L2-38
- ソフトウェアスケジュール
 - 管理： L3-51
 - 進捗： L2-37
 - 手順： L2-23
- ソフトウェア作業成果物
 - 首尾一貫性： L3-78
 - 監査： L2-66
 - 規模の進捗確認： L2-35
 - 規模の見積り： L2-21
 - 審査： L3-98
 - 責任： L2-15, L2-31
 - 定義： 0-60
- ソフトウェア成果物を評価するチェックリスト： L3-98

- ソフトウェア設計
 定義： L3-69
 要件： L3-69
- ソフトウェア設計文書： L3-71
- ソフトウェアタスク
 欠陥予防の立ち上げ会議： L5-6
- ソフトウェアタスクリーダー
 定義： 0-68
- ソフトウェアテスト： L3-72
- ソフトウェアテスト基準
 レビュー： L3-72
- ソフトウェアテスト計画
 ピアレビュー： L3-74
 要件： L3-75
- ソフトウェア統合管理： L3-37～L3-58
 記述： 0-22
 資源と資金： L3-39
 トレーニング： L3-39-40
 方針： L3-38
 レビュー
 上級管理層： L3-56
 プロジェクトマネージャ： L3-57
- ソフトウェア関連グループ
 定義： 0-71
- ソフトウェアに割り当てられたシステム要件
 (参照先：割り当てられた要件)
 管理の方針： L2-2
 記述： 0-43
 定義： L2-1
 文書： L2-4
 文書の管理と制御： L3-78
 変更： L2-7
 例： L2-4
 レビュー： L2-5
- ソフトウェア品質管理： L4-19～L4-32
 記述： 0-23
 計測： L4-31
 資源と資金： L4-21
 実装のためのトレーニング： L4-22
 ソフトウェアエンジニアリンググループ
 へのトレーニング： L4-22
 ツール： L4-21
 方針： L4-20
 レビュー
 上級管理層： L4-31
 プロジェクトマネージャ： L4-31
 レビューと監査
 ソフトウェア品質保証： L4-32
- ソフトウェア品質計画
 項目： L4-25
 管理と制御： L4-25
 上級管理層によるレビュー： L4-25
 ピアレビュー： L4-24
 品質目標の文書： L4-28
 要件： L4-23
- ソフトウェア品質保証： L2-59～L2-69
 オリエンテーション： L2-63
 記述： 0-21
 計測： L2-68
 資源と資金： L2-62
 ツール： L2-62
 トレーニング： L2-62
 方針： L2-60
 マネージャ： L2-62
 レビュー
 上級管理層： L2-68
 独立した専門家： L2-69
 プロジェクト： L2-69
- ソフトウェア品質保証グループ： L2-61
 外注先のソフトウェア構成管理のモニター： L2-53
 活動： L2-65
 活動報告： L2-67
 顧客との調整： L2-67
 ソフトウェア開発計画： L2-65
 定義： 0-72
 ソフトウェア成果物の監査： L2-66
 レビュー
 技術変更管理： L5-29
 グループ間調整： L3-92

索引

- 欠陥予防： L5-15
- 組織プロセス定義： L3-23
- ソフトウェアエンジニアリング活動： L2-66
- ソフトウェア外注管理： L2-57
- ソフトウェア構成管理： L2-83
- ソフトウェア統合管理： L3-57
- ソフトウェア品質管理： L4-32
- ソフトウェアプロジェクト計画： L2-27
- ソフトウェアプロジェクト進捗管理： L2-41
- ソフトウェアプロダクトエンジニアリング： L3-81
- 定量的プロセス管理： L4-16
- ピアレビュー： L3-100
- プロセス変更管理： L5-46
- 要件管理： L2-9
- ソフトウェア品質保証計画： L2-63
 - 管理と制御： L2-64
- ソフトウェア品質目標
 - 対立を解消するための処置： L4-30
 - 文書化： L4-28
- ソフトウェア品質保証レビュー
 - 記述： 0-49
- ソフトウェアプロジェクト
- ソフトウェアプロジェクト： L2-14
 - 管理： L3-44
 - コミットメント
 - レビュー： L3-55
- ソフトウェアプロジェクト計画： L2-11 ~ L2-27
 - 記述： 0-20
 - 計測： L2-25
 - 資源と資金： L2-16
 - 責任： L2-12
 - 全体プロジェクト計画： L2-17
 - 調整： L2-15
 - ツール： L2-16
 - トレーニング： L2-16
- 方針： L2-12
- レビュー
 - 上級管理層： L2-26
 - プロジェクトマネージャ： L2-26
- レビューと監査
 - ソフトウェア品質保証： L2-27
- ソフトウェアプロジェクトコミットメント
 - レビュー： L2-13, L2-17
- ソフトウェアプロジェクト進捗管理： L2-29 ~ L2-42
 - 記述： 0-20
 - 計測： L2-39
 - 作業タスクの割り当て： L2-31
 - 資源と資金： L2-32
 - ソフトウェア開発計画： L2-33
 - ツール： L2-32
 - トレーニング： L2-32
 - 方針： L2-30
- レビュー
 - 上級管理層： L2-40
 - プロジェクト： L2-41
- レビューと監査
 - ソフトウェア品質保証： L2-41
- ソフトウェアプロジェクトデータ
 - 記録： L2-38
- ソフトウェアプロジェクトの作業内容記述書
 - 管理と制御： L2-15
 - レビュー： L2-15
- ソフトウェアプロセス
- ソフトウェアプロセスアーキテクチャ
 - 定義： L3-18, 0-54
- ソフトウェアプロセスアセスメント (SPA) : L3-6
- ソフトウェアプロセス改善 (参照先：プロセス変更管理と組織プロセス重視)
 - 記録： L5-43
 - 計画： L5-37
 - 試行的な導入： L5-42
 - 上級管理層から監督： L5-32
 - トレーニングの更新： L5-43

- フィードバック： L5-44
- プログラム： L5-35
- プロセスエリア毎のチーム： L5-41
- 方針： L5-32
- 実装： L5-42
- ソフトウェアプロセス改善計画
 - 項目： L5-38
 - 管理と制御： L5-37
 - ピアレビュー： L5-37
 - 要件： L5-37
- ソフトウェアプロセス改善提案
 - 操作手順： L5-39
 - 文書： L5-10
- ソフトウェアプロセス要素
 - 定義： 0-55
- ソフトウェアプロセス資産
 - 例： 0-53
 - レベル3での開発： 0-21
- ソフトウェアプロセス実績
 - アセスメント： L3-6
 - 定義： 0-23
- ソフトウェアプロセス重視
 - 調整： L3-7
- ソフトウェアプロセスデータベース
 - 確立と維持： L3-20
 - 管理と制御： L3-21
 - 計画と見積りでの使用： L3-46
 - 組織レベルでの調整： L3-8
 - レベル4での使用： 0-16
- ソフトウェアプロセス能力
 - 定義： 0-10
- ソフトウェアプロセス定義（参照先：組織プロセス重視）
 - 能力成熟度モデルとの関連： 0-50
- ソフトウェア成果物
 - ソフトウェア品質保証監査： L2-66
 - 定義： 0-60
- ソフトウェアプロダクトエンジニアリング： L3-59～L3-82
 - 記述： 0-22
- 技術要員のオリエンテーション： L3-64
- 計測： L3-79
- 資源と資金： L3-61
- ツール： L3-61
- トレーニング： L3-63
- 方針： L3-60
- マネージャへのオリエンテーション： L3-64
- レビュー
 - 上級管理層： L3-80
 - プロジェクトマネージャ： L3-80
- レビューと監査
 - ソフトウェア品質保証： L3-81
- ソフトウェアベースライン
 - 監査： L2-81
- ソフトウェアベースライン
 - 監査： L2-83
 - リポジトリ： L2-77
- ソフトウェアベースラインライブラリ
 - 生成： L2-80
 - 成果物の生成： L2-80
 - チェックイン/チェックアウト手順： L2-80
 - 定義： L2-73
- ソフトウェアマネージャ
 - コミットメントの協議： L2-13
 - ソフトウェア作業成果物の責任： L2-15
 - トレーニングプログラムのオリエンテーション： L3-29
 - プロジェクト進捗確認の責任： L2-32
 - プロセス変更管理のトレーニング： L5-34
- レビュー
 - 作業内容記述書： L2-15
 - ソフトウェア開発計画： L2-19
 - プロジェクト進捗確認： L2-38
 - 割り当てられた要件： L2-3
- ソフトウェア要件
 - 開発： L3-66
 - 変更： L3-69
- ソフトウェア要件文書

索引

構成管理： L3-69
ピアレビュー： L3-68
ソフトウェアライフサイクル
管理と制御： L3-19
特定： L2-17
選択： L3-41
定義： 0-55
ピアレビュー： L3-19
文書： L3-18
ソフトウェアリスク
ソフトウェアリスク管理計画
管理と制御： L3-54
管理のための計画： L3-53
ピアレビュー： L3-54
ソフトウェアを開発し保守するツール
構成管理： L3-66

た

第一線ソフトウェアマネージャ
定義： 0-67
レビュー
プロジェクト進捗： L2-38
タスク
定義： 0-59

ち

チーム
欠陥予防： L5-3
分野横断的なエンジニアリング： L3-84

つ

ツール
技術変更管理： L5-21
グループ間調整： L3-85
欠陥予防： L5-4
組織プロセス重視： L3-5
組織プロセス定義： L3-14
ソフトウェア外注管理： L2-46

ソフトウェア構成管理： L2-75
ソフトウェア品質管理： L4-21
ソフトウェア品質保証： L2-62
ソフトウェアプロジェクト計画： L2-16
ソフトウェアプロジェクト進捗管理：
L2-32
ソフトウェアプロダクトエンジニアリン
グ： L3-61
ソフトウェアの構築と保守： L3-60
定量的プロセス計測： L4-5
トレーニング： L3-28
プロジェクトの選択： L3-65
プロセス変更管理： L5-34
要件管理： L2-5

て

データ

欠陥予防： L5-11
収集と分析のプラクティスの発展： 0-63
定義されたソフトウェアプロセス
ソフトウェアコードの開発における使
用： L3-71
ソフトウェア設計の開発： L3-69
ソフトウェア要件の開発： L3-66
定義されたレベル
記述： 0-15
定期的およびイベント発生を契機とするレビ
ュー
記述： 0-49
定期的レビュー
記述： 0-48
定量的制御
定義： L4-3
定量的プロセス管理： L4-1 ~ L4-17(参照先：
計測)
記述： 0-23
グループ： L4-4
計画： L4-6
計測： L4-15

資源と資金： L4-4
 ツール： L4-5
 トレーニング： L4-6
 方向： L4-6
 報告： L4-12
 方針： L4-2-3
 レビュー
 上級管理層： L4-15
 プロジェクトマネージャ： L4-16
 レビューと監査
 ソフトウェア品質保証： L4-16
 定量的プロセス管理計画
 項目： L4-7
 手順（参照先：文書化された手順プロセス能力）
 傾向： L4-13
 定義： L4-1
 テスト
 システムの検収： L3-75
 ソフトウェア： L3-72
 テスト計画
 管理と制御： L3-74

と

独立性
 ニーズ： 0-73
 トレーニング（参照先：オリエンテーションと必修トレーニング）
 免除： L3-34
 記述： 0-38
 記録： L3-34
 グループ間調整： L3-85
 コースの開発： L3-33
 施設： L3-28
 教材のレビュー： L3-33
 組織横断的な調整： L3-8
 組織プロセス重視： L3-5
 組織プロセス定義： L3-14
 ソフトウェア外注管理： L2-46

ソフトウェア構成管理： L2-76
 ソフトウェア統合管理： L3-39
 ソフトウェア品質保証： L2-62
 ソフトウェアプロジェクト進捗管理： L2-32
 ソフトウェアプロジェクト計画： L2-16
 ソフトウェアプロセス改善に基づく更新： L5-43
 ソフトウェアプロダクトエンジニアリング： L3-63
 ツール： L3-28
 プロジェクトの計画： L3-29, L3-30
 方針： L3-26
 トレーニンググループ
 スキルと知識： L3-28
 定義： 0-73
 トレーニング計画
 項目： L3-32
 管理と制御： L3-31
 トレーニング手段
 例： L3-26
 トレーニング教材
 管理と制御： L3-34
 トレーニングプログラム： L3-25 ~ L3-36
 オリエンテーション： L3-29
 記述： 0-22
 グループ： L3-27
 計測： L3-34
 資源と資金： L3-27
 独立の評価： L3-36
 レビュー
 上級管理層： L3-35
 レビューと監査： L3-36
 トレーニングプログラム要素
 項目： L3-28

の

能力成熟度モデル
 起源： 0-3

索引

記述： 0-7
技術との関係： 0-62
背景： 0-7
構造： 0-9, 0-12
査定手法との関係： 0-4
使用： 0-1
柔軟性： 0-75
小規模かつ/または民間組織での使用：
0-2
使用されている組織構造： 0-64
発展： 0-8
ニーズ： 0-1
文書との関係： 0-62
要件： L5-7
ライフサイクルとの関係： 0-61

は

パレート分析

欠陥予防での使用： L5-9

判定

キープラクティスの解釈での使用： 0-75

反復できるレベル

記述： 0-14

ひ

ピアレビュー

技術変更管理計画： L5-24
作業成果物評価のチェックリスト： L3-98
システム文書： L3-77
組織の標準ソフトウェアプロセス： L3-16
組織プロセス重視の計画： L3-7
ソフトウェアコード： L3-72
ソフトウェア設計文書： L3-71
ソフトウェアテスト計画： L3-74
ソフトウェアライフサイクル： L3-19
ソフトウェア品質計画： L4-24
ソフトウェアプロセス改善計画： L5-37
ソフトウェア要件文書： L3-68

ソフトウェアリスク管理計画： L3-54
とレビュー： 0-45
ピアレビュー： L3-93~L3-100
記述： 0-22
計画： L3-97
計測： L3-99
欠陥予防計画： L5-6
参加するためのトレーニング： L3-96
資源と資金： L3-95
主導するためのトレーニング： L3-95
定量的プロセス管理計画： L4-7
手順： L3-97
方針： L3-94
レビューと監査
ソフトウェア品質保証： L3-100

非遵守問題

管理と制御： L2-67

必修トレーニング(参照先：トレーニングと オリエンテーション)

記述： 0-38

技術変更管理： L5-23

欠陥予防： L5-4

ソフトウェア品質管理の実装： L4-22

定量的プロセス管理： L4-6

ピアレビューの主導： L3-95

ピアレビューへの参加： L3-96

プロセス変更管理： L5-34-35

標準

開発に関わるソフトウェア品質保証：
L2-65

トレーニングコース： L3-33

品質機能展開(QFD)： L4-23

品質ニーズ

追跡可能性： L4-23

品質目標

外注先への割り当て： L4-30

定義と改定： L4-27

レベル4での確立： 0-16

ふ

Faganスタイルインスペクション

ピアレビューでの使用： 0-23

プロジェクト進捗管理（参照先：ソフトウェアプロジェクト進捗管理）

プロジェクトソフトウェアマネージャ

コミットメントの協議： L2-12

作業タスクの割り当て： L2-31

責任： L2-30

ソフトウェアプロジェクト計画の調整：
L2-15

ソフトウェアプロジェクト計画の責任：
L2-12

定義： 0-67

レビュー

作業内容記述書： L2-15

ソフトウェア開発計画： L2-19

プロジェクトの進捗確認： L2-38

プロジェクトのソフトウェア開発計画

開発と改訂： L3-44

定義： 0-70

プロジェクトの定義されたソフトウェアプロセス

新しい技術の導入： L5-28

開発のためのガイドライン： L3-41

管理と制御： L3-42

計測と制御： L4-2

欠陥データでの使用： L3-78

欠陥予防チームとの調整： L5-3

欠陥予防に基づく改訂： L5-12

システム文書作成での使用： L3-76

組織レベルでの調整： L3-8

ソフトウェアエンジニアリング活動の統合：
L3-65

ソフトウェアエンジニアリングタスク：
L3-60

ソフトウェアテストでの使用： L3-72

ソフトウェアプロジェクトの管理での使用：
L3-44

定義： 0-16, 0-58

定量的制御： L4-10

定量的制御： L4-10

定量的プロセス管理での使用： L4-10

品質目標との関係： L4-24

プロセス変更の導入： L5-43

文書： L3-39

改訂： L3-43

プロジェクトマネージャ

コミットメントの協議： L2-12

定義： 0-66

レビュー

グループ間のコミットメント： L3-89

グループ間調整： L3-92

欠陥予防： L5-15

作業内容記述書： L2-15

ソフトウェア外注管理： L2-56

ソフトウェア開発計画： L2-19

ソフトウェア構成管理： L2-83

ソフトウェア統合管理： L3-57

ソフトウェア品質管理： L4-31

ソフトウェア品質保証： L2-69

ソフトウェアプロジェクト計画：
L2-26

ソフトウェアプロジェクト進捗管
理： L2-41

ソフトウェアプロダクトエンジニア
リング： L3-80

定量的プロセス管理計画： L4-16

要件管理： L2-9

リスク： L2-38

プロセス改善（参照先：ソフトウェアプロセス改善）

レベル5での焦点： 0-17

プロセス要素

定義： L3-17

プロセス資産

組織の標準ソフトウェアプロセスのテー
ラリングでの使用： L3-42

プロジェクトの計画と管理での使用：

索引

- L3-38
 - 方針： L3-12
- プロセス実施
 - 定義： L4-1
- プロセス実施ベースライン： L4-13
 - 管理と制御： L4-12
 - 定義： L4-1
- ソフトウェアプロセス関連文書のライブラリ
 - 定義： 0-57
- プロセス能力ベースライン： L4-3
 - 管理と制御： L4-14
 - 成熟度レベルとの関係： 0-12
 - 組織の標準ソフトウェアプロセス： L4-13
- プロセス変更管理： L5-31～L5-47（参照先：ソフトウェアプロセス改善）
 - 記述： 0-24
 - 計測： L5-42
 - 資源と資金： L5-33
 - ツール： L5-34
 - トレーニング： L5-34～L5-35
 - レビュー
 - 上級管理層
 - レビューと監査
 - ソフトウェア品質保証： L5-46
- 文書
 - 能力成熟度モデルとの関係： 0-62
- 文書化された手順
 - 新しい技術の選択： L5-26
 - 外注された作業： L2-47
 - 外注先成果物の検収テスト： L2-55
 - 外注先のソフトウェア構成管理のモニター： L2-54
 - 外注先のソフトウェア品質保証のモニター： L2-53
 - 記述： 0-42
 - グループ間の課題： L3-90
 - 欠陥予防に基づく組織の標準ソフトウェアプロセスの改訂： L5-12
 - 欠陥予防に基づくプロジェクトの定義されたソフトウェアプロセスの改訂： L5-12
- 欠陥予防の原因分析会議： L5-7
- 構成アイテム/ユニットの変更： L2-79
- 構成アイテムの状況記録： L2-80
- コミットメントのレビュー： L2-35
- 資源と工数の見積り： L2-22
- 重要な依存関係の特定： L3-89
- 重要なコンピュータ資源の管理： L3-50
- 重要なコンピュータ資源の見積り： L2-23
- 組織のトレーニング計画： L3-30
- 組織の標準ソフトウェアプロセスへの新しい技術の導入： L5-28
- 組織の標準ソフトウェアプロセスの開発：
 - L3-15
- ソフトウェアエンジニアリング活動の逸脱： L2-67
- ソフトウェア外注先の作業内容記述書の変更： L2-51
- ソフトウェア外注先の選定： L2-49
- ソフトウェア外注先のレビュー： L2-53
- ソフトウェア開発計画の改訂： L2-33
- ソフトウェア開発計画の策定： L2-18
- ソフトウェア規模管理： L3-47
- ソフトウェア構成管理計画の策定： L2-76
- ソフトウェア作業成果物の規模の見積り： L2-21
- ソフトウェアスケジュールの管理： L3-51
- ソフトウェアスケジュールの作成： L2-23
- ソフトウェアコストの管理： L3-48
- ソフトウェア品質計画： L4-23
- ソフトウェア品質保証計画の準備： L2-63
- ソフトウェアプロセス改善計画： L5-37
- ソフトウェアプロセス改善提案： L5-39
- ソフトウェアプロセス改善の実装： L5-42
- ソフトウェアベースライン監査： L2-81
- ソフトウェアベースラインライブラリからの成果物： L2-80
- 定量的プロセス管理計画： L4-6
- ピアレビュー： L3-97
- 定量的プロセス制御データ： L4-9

プロジェクトのソフトウェア開発計画：
L3-44
プロジェクトの定義されたソフトウェア
プロセスへの新しい技術の導入： L5-28
プロジェクトの定義されたソフトウェア
プロセスの改訂： L3-43
プロジェクトの定義されたソフトウェア
プロセスの開発： L3-41
プロジェクトの定義されたソフトウェア
プロセスの制御： L4-10
プロセス能力ベースライン： L4-13
ベースラインの変更： L2-80
マイルストーンレビュー： L2-39
リスク管理： L3-52
分野横断的なエンジニアリングチーム
方針： L3-84

へ

ベースライン
変更を制御する手順： 80
ベースライン構成管理
記述： 0-46
変更要求
評価の手順： L2-79

ほ

方針
記述： 0-36
技術変更管理： L5-18
組織の欠陥予防： L5-2
組織プロセス重視： L3-2
組織プロセス定義： L3-12
ソフトウェア外注管理： L2-44
ソフトウェア構成管理活動： L2-72
ソフトウェア統合管理： L3-38
ソフトウェアに割り当てられたシステム
要件の管理： L2-2
ソフトウェア品質管理： L4-20

ソフトウェア品質保証： L2-60
ソフトウェアプロジェクト計画： L2-12
ソフトウェアプロジェクト進捗管理：
L2-30
ソフトウェアプロセス改善： L5-32
ソフトウェアプロダクトエンジニアリン
グ： L3-60
定量的プロセス管理： L4-2-3
トレーニング： L3-26
ピアレビュー： L3-94
プロジェクトの欠陥予防： L5-2
分野横断的なエンジニアリングチーム：
L3-84
補足情報
記述： 0-32

ま

マイルストーン
定義： L3-51

み

見積り
工数とコスト： L2-22
重要なコンピュータ資源： L2-23
ソフトウェア作業成果物の規模： L2-21

も

目標
キープラクティスの解釈での使用： 0-75
使用： 0-27
定義： 0-11
問題レポート
手順： L2-79

や

役割
定義： 0-64

索引

よ

要員

定義： 0-68

予算

と資金： 0-37

要件管理： L2-1～L2-10

記述： 0-20

計測： L2-8

資源と資金： L2-5

ツール： L2-5

レビュー

上級管理層： L2-9

プロジェクトマネージャ： L2-9

レビューと監査

ソフトウェア品質保証： L2-9

ソフトウェア品質保証： L3-92

プロジェクトマネージャ： L3-92

グループ間調整の技術面： L3-90

欠陥予防

上級管理層： L5-14

ソフトウェア品質保証： L5-15

プロジェクトマネージャ： L5-15

コミットメント： L2-31

進捗の確認： L2-38

成果物： L3-90

組織プロセス重視

上級管理層： L3-10

組織プロセス定義

ソフトウェア品質保証： L3-23

ソフトウェア外注管理

上級管理層： L2-56

ソフトウェア品質保証： L2-57

プロジェクトマネージャ： L2-56

ソフトウェア外注の技術面： L2-52

ソフトウェア構成管理

上級管理層： L2-82

ソフトウェア品質保証： L2-83

プロジェクトマネージャ： L2-83

ソフトウェア統合管理

上級管理層： L3-56

ソフトウェア品質保証： L3-57

プロジェクトマネージャ： L3-57

ソフトウェア品質管理

上級管理層： L4-31

ソフトウェア品質保証： L4-32

プロジェクトマネージャ： L4-31

ソフトウェア品質保証

上級管理層： L2-68

プロジェクトマネージャ： L2-69

ソフトウェアプロジェクト計画

上級管理層： L2-26

ソフトウェア品質保証： L2-27

プロジェクトマネージャ： L2-26

ソフトウェアプロジェクト進捗管理

上級管理層： L2-40

ら

ライフサイクル

能力成熟度モデルの関係： 0-61

り

履行検証

記述： 0-28

リスク

管理： L3-52

特定： L2-24

進捗： L2-37

レベル4での管理： 0-17

リスク対応計画： L3-54

れ

レビュー

技術変更管理

上級管理層： L5-29

ソフトウェア品質保証： L5-29

グループ間調整

上級管理層： L3-91

ソフトウェア品質保証： L2-41
プロジェクトマネージャ： L2-41
ソフトウェアプロジェクトのコミットメント： L2-13, L2-17
ソフトウェアプロダクトエンジニアリング
上級管理層： L3-80
ソフトウェア品質保証： L3-81
プロジェクトマネージャ： L3-80
定期的： 0-48
定期的およびイベント発生を契機とする： 0-49
定量的プロセス管理
上級管理層： L4-15
ソフトウェア品質保証： L4-16
プロセスマネージャ： L4-16
トレーニング教材： L3-33
トレーニングプログラム
上級管理層： L3-35
ピアレビュー
ソフトウェア品質保証： L3-100
とピアレビュー： 0-45
プロセス変更管理
上級管理層： L5-46
ソフトウェア品質保証： L5-46
マイルストーン： L2-39
要件管理
上級管理層： L2-9
ソフトウェア品質保証： L2-9
プロジェクトマネージャ： L2-9
レポート
ソフトウェア構成管理活動： L2-81
ソフトウェア品質保証活動： L2-67
定量的プロセス管理： L4-12

わ

割り当てられた要件（参照先：ソフトウェアに割り当てられたシステム要件）
定義： L2-2

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: 標題:

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:

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 --