

限定なし



ソフトウェア開発標準

2024 年 3 月 29 日 D 改訂

宇宙航空研究開発機構

免責条項

ここに含まれる情報は、一般的な情報提供のみを目的としています。JAXA は、かかる情報の正確性、有用性又は適時性を含め、明示又は黙示に何ら保証するものではありません。また、JAXA は、かかる情報の利用に関連する損害について、何ら責任を負いません。

Disclaimer

The information contained herein is for general informational purposes only. JAXA makes no warranty, express or implied, including as to the accuracy, usefulness or timeliness of any information herein. JAXA will not be liable for any losses relating to the use of the information.

発行

〒305-8505 茨城県つくば市千現 2-1-1

宇宙航空研究開発機構 安全・信頼性推進部

JAXA (Japan Aerospace Exploration Agency)

目 次

1	適用範囲.....	1
2	関連文書.....	1
2.1	参考文献.....	1
3	用語、定義および略語.....	2
4	一般要求事項.....	9
4.1	テーラリング.....	11
5	主ライフサイクルプロセス.....	11
5.1	欠番.....	12
5.2	欠番.....	12
5.3	開発プロセス.....	12
5.3.1	プロセス開始の準備.....	13
5.3.2	全開発プロセス適用事項.....	14
5.3.3	コンピュータシステム要求分析.....	15
5.3.4	コンピュータシステム方式設計.....	16
5.3.5	ソフトウェア要求分析.....	17
5.3.6	ソフトウェア設計.....	20
5.3.7	欠番.....	22
5.3.8	ソフトウェア製作.....	22
5.3.9	欠番.....	24
5.3.10	ソフトウェア統合.....	25
5.3.11	ソフトウェア統合試験.....	26
5.3.12	目標プラットフォームへのインストール(組み込み).....	28
5.3.13	コンピュータシステム統合およびコンピュータシステム総合試験.....	29
5.3.14	ソフトウェア製品の供給と導入.....	31
5.3.15	ソフトウェア製品の受け入れ.....	32
5.4	運用プロセス.....	33
5.4.1	プロセス開始の準備.....	34
5.4.2	運用試験.....	35
5.4.3	ソフトウェアを含むコンピュータシステムの運用.....	35

5.4.4	運用結果の管理	36
5.4.5	顧客およびユーザサポート	36
5.5	保守プロセス	36
5.5.1	プロセス開始の準備	37
5.5.2	問題把握および修正分析	38
5.5.3	修正の実施	38
5.5.4	ソフトウェアの書き換え	38
5.5.5	ロジスティクス支援の実施	39
5.5.6	保守およびロジスティクス結果の管理	39
5.5.7	移行	40
5.5.8	ソフトウェア廃棄	41
6	支援ライフサイクルプロセス	43
6.1	文書化プロセス	43
6.1.1	プロセス開始の準備	43
6.1.2	作成	44
6.1.3	発行	44
6.1.4	維持・改訂・廃棄	44
6.2	構成管理プロセス	44
6.2.1	プロセス開始の準備	45
6.2.2	構成識別	46
6.2.3	構成変更管理	46
6.2.4	構成変更状況の記録	46
6.2.5	構成変更状況の評価	46
6.2.6	リリース管理と出荷	47
6.2.7	構成監査の実施	47
6.3	品質保証プロセス	47
6.3.1	プロセス開始の準備	47
6.3.2	製品およびサービス品質の保証	49
6.3.3	プロセスの保証	49
6.3.4	品質システムの保証	49

6.3.5	品質保証記録の管理	51
6.4	検証プロセス	51
6.4.1	プロセス開始の準備	51
6.4.2	検証	52
6.4.3	検証結果の管理	54
6.5	妥当性確認プロセス	54
6.5.1	プロセス開始の準備	54
6.5.2	妥当性確認	55
6.5.3	妥当性確認結果の管理	56
6.6	共同レビュープロセス	56
6.6.1	プロセス開始の準備	56
6.6.2	プロジェクト管理レビュー	57
6.6.3	技術レビュー	57
6.7	アセスメントプロセス	58
6.7.1	プロセス開始の準備	58
6.7.2	アセスメントの実施	59
6.8	問題解決プロセス	59
6.8.1	プロセス開始の準備	59
6.8.2	問題の解決	59
6.8.3	予防	61
6.8.4	傾向分析	61
	付録	a
	付録1 開発プロセス全体構成の例(ウォーターフォール型)	a
	付録2 インプット・アウトプットと各プロセス対応表	b
	付録3 「検証」 および「妥当性確認」に関する補足説明	c
	付録4 問題解決プロセスと他プロセスとの関係	d
	付録5 4つの保守タイプに関する補足説明	f
	付録6 ソフトウェア製品、ナレッジ資産、イネーブリングシステムの補足	g
	付録7 開発するコンピュータシステム/ソフトウェアの関連システムに関する補足説明	h
	付録8 戦略の位置付けの明確化	k

付録9 「試験計画性」および「試験可能性」の補足説明	k
付録10 「ソフトウェア結合」および「ソフトウェア統合」の補足説明	k
付録11 ソフトウェアクリティカリティクラス(SW CC)の設定	l
付録12 本標準の各要求項目の適用基準	n

1 適用範囲

本標準は、人工衛星・探査機、ロケット、地上システムなどにかかわるソフトウェアの開発・運用・保守に関する活動を適用範囲とするとともに、必要なシステム開発活動および関連する支援活動も適用対象範囲とする。実際の適用にあたっては、プロジェクトの特徴などに応じた具体化・テーラリングを行うことができる。

なお、本書では基本的にプロセスの実施者区分を規定しない。プロセスの実施者は、本標準を適用した各契約などの条項で規定するものとする。

2 関連文書

2.1 参考文書

- (1) ISO/IEC/IEEE 12207 : 2017 Information technology - Software life cycle process
- (2) JIS X0160:2021 ソフトウェアライフサイクルプロセス
- (3) ISO/IEC 14764 Software Life Cycle Processes — Maintenance
- (4) JIS X0161-2008 ソフトウェア技術—ソフトウェアライフサイクルプロセス—保守
- (5) ISO/IEC 33004:2015 Information technology — Process assessment — Requirements for process reference, process assessment and maturity models
- (6) ISO/IEC 33020:2015 Information technology — Process assessment — Process measurement framework for assessment of process capability
- (7) JIS X 33020:2019 情報技術—プロセスアセスメント—プロセス能力のアセスメントのためのプロセス測定フレームワーク
- (8) 共通フレーム 2013 (Copyright IPA/SEC 2013)
- (9) ISO 9000 : 2015 Quality management systems—Fundamentals and vocabulary
- (10) JIS Q 9000 : 2015 品質マネジメントシステム—基本及び用語
- (11) JMR-004 信頼性プログラム標準
- (12) JERG-0-063 宇宙開発信頼性技術ハンドブック
- (13) JMR-001 システム安全標準

3 用語、定義および略語

以下、英語、アイウエオ順で示す。

用語	説明
CIL	クリティカル品目リストの略で、信頼性評価により識別された重要品目のリスト。JMR-004 4.3.16.2.2を参照のこと。
COTS	Commercial Off-The-Shelf の略で、一般に入手可能な既製品のこと。
アウトプット	アクティビティを実施することによって、インプットから変換される出力情報。
アクティビティ	プロセスを構成する要素であり、関連の強いタスクの集合。
アーキテクチャ	構成要素やその相互関係、デザイン方式やその発展性の方針として具体化されるシステムの基本的な概念や特性。
アセスメント	あらかじめ定められた目的に合わせて、対象プロセスの強みおよび弱みを評価し、改善機会を識別する活動である。
イネーブリングシステム	ライフサイクルのあらゆる段階で、コンピュータシステムを支援するシステム。 注記 1 たとえば、対象システムが製作段階に入ると、生産イネーブリングシステムが必要となる。対象システムが宇宙機搭載ソフトウェアだとすると、イネーブリングシステムとは、ライフサイクルを通していえば、エミュレータ、シミュレータなどを指す。 注記 2 各イネーブリングシステムは、それ自身のライフサイクルをもつ。イネーブリングシステム自体を対象システムとして扱う場合には、この規格は、各イネーブリングシステムに適用可能である。
インシデント	変則的あるいは予期せぬイベントで、プロジェクト・製品・サービス・システムのライフサイクルを通した一連のイベント／状態／状況のこと。
インプット	アクティビティを実施するために必要となる入力情報。
ウェーバ	システムまたは構成品目の製作開始以降に発生したコンフィギュレーション識別文書の要求事項に対する不適合につき、そのまままたは承認された方法による修理の後、受領する場合をいう。(JMR-006 コンフィグレーション管理基準)

用語	説明
受け入れ検査・試験	ソフトウェア製品の受け入れに際して、要求との適合性を評価する行為。 検査とは要求仕様あるいは理論的に正しい値などに基づく判断基準に沿って、目視確認、数量確認、試験結果などの評価記録を通じて確認する行為である。 試験とは検査に必要な評価結果およびデータを取得するために、ソフトウェアを動作させ、機能・性能などの解析・評価・確認を行う行為である。
運用	コンピュータシステムを稼動して当該コンピュータシステムの目標を達成するための作業を遂行する行為。コンピュータシステムの起動やコンピュータシステムの終了、監視およびメンテナンスなどが含まれる。
完全性	本標準で完全性は下記の意味で使用される。 (1) 対象の構成要素が不足なくそろっていること。 (2) 対象の構成要素が適切なバージョンになっていること。
完全化保守	引渡し後のソフトウェア製品の潜在的な障害が、故障として現れる前に、検出し訂正するための修正。 注記 完全化保守は、利用者のための改良、プログラム文書の改善を提供し、ソフトウェアの性能強化、保守性などのソフトウェア属性の改善にむけての記録を提供する。(JIS X 0161 : 2008 (ISO/IEC 14764 : 2006))
機能外要求	性能・安全性・信頼性など、機能以外の要求すべて。
供試ソフトウェア	試験・検査に供試する被試験・検査対象のソフトウェア。
検証	客観的証拠を提示することによって、規定要求事項が満たされていることを確認すること。(JIS Q 9000:2015 (ISO9000))
検証可能性	要求仕様に対して、検証基準を確立することができ、その基準が満たされているかどうかを確認するために試験・解析などの検証を実行することが可能であること。
構成監査	構成管理活動の実施者とは異なる第三者が、構成管理活動の構成管理計画への適合の確認を行うこと。

用語	説明
構成管理	コンピュータシステムあるいはプロジェクトを構成する各種の構成品目を定義し、その品目の変更内容を記録し、また品目の保管、取扱い、引渡しなどを含めた管理を行う行為。ソフトウェアが複数のモジュールから構成される場合には、ソフトウェアのバージョン管理だけでなく、ソフトウェアを構成する各モジュールのバージョン管理も必要である。なお、構成管理品目は、ソフトウェアを構成するモジュールの他、要求仕様や操作マニュアルなども対象にしている。
コンピュータシステム	開発対象であるソフトウェアの実行が可能なプラットフォーム・ハードウェアを含む、ソフトウェア・プラットフォーム・ハードウェアの総体。コンピュータシステムに何を含めるかの定義は任意であるが、開発対象であるソフトウェア製品に対して、その定義は一意であること。 コンピュータシステムとしては、ワンチップマイコン 1 個による構成から、複数の汎用計算機をネットワークで接続した構成まで様々な形態を定義しうる。
サービス	利用者やシステムに対して、機能や働きを提供するもの。
識別子	アウトプットやインプットの各項目に付与され、各項目を特定可能な、比較的短い文字列。項目番号のようなものが、識別子に該当する。識別子は数字だけでなくアルファベットや記号などを組み合わせてもよく、また、必ずしも連番である必要はない。 各項目に識別子を付与することにより、要求などの識別(要求管理)やトレーサビリティ確認などの利便性が向上するといったメリットが期待される。
試験計画性	試験対象項目に対して、適切な開発フェーズ・試験環境を用いて、試験が可能である仕様記述であり、計画できること。(付録9 参照)

用語	説明
試験網羅率	試験及びコードの定量的品質指標の1つであり、本文書では以下の試験網羅率を使用する。 C0： 命令網羅率(ステートメントカバレッジ)とも呼ばれ、コード内の全てのステートメントを少なくとも1回は実行する試験網羅率のことである。 C1： 分岐網羅率(ブランチ(デシジョン)カバレッジ)とも呼ばれ、コード内の全てのブランチを少なくとも1回は実行する試験網羅率のことである。 MC/DC： 改良条件判断網羅率 (Modified Condition/Decision Coverage)とも呼ばれ、以下を満たすテストを実行する試験網羅率のことである。 1) 各入口/出口が呼び出される 2) 各判定式 (デシジョン) が、少なくとも1回は、すべての可能な結果を得る 3) 1つの判定式に含まれる各条件 (コンディション) が、少なくとも1回は、すべての可能な結果を得る 4) 1つの判定式に含まれる各条件が、単独でその判定式の結果に影響する
システム	ロケット、衛星、地上設備等、組織化された機能要素 (ハードウェア及びソフトウェア) の集合。システムを総合的に組み合わせてミッションを達成する。
循環的複雑度 (Cyclomatic Complexity)	プログラムのロジックの複雑さを表す指標であり、プログラム構造をグラフ表現したときに、グラフの頂点数(n)、辺数(e)、接続されたプログラムの要素数(p)から次式で計算される。 $e-n+2p$
是正保守	ソフトウェア製品の引き渡し後に発見された問題を訂正するために行う受身の修正。 注記 この修正によって、要求事項を満たすようにソフトウェア製品を修復する。(JIS X 0161 : 2008 (ISO/IEC 14764 : 2006))
戦略	組織の共通目標 (品質、生産性、環境など) またはプロジェクトのミッションを達成するために、特定の資源および時間の効果的な利用などを目的とした実行計画の立案にあたって、考慮すべき指針。(付録8 参照)
相互作用	複数の機能間で交換されて特定の目的を遂行する一連のやりとり。

用語	説明
ソースコード	ソフトウェア(いわゆるオブジェクトコード)生成の元となるものである。一般的には人間が読み書きするのに適したプログラミング言語で記述される。
ソフトウェア	コンピュータシステムの構成品目の一種であり、主としてソフトウェア要求仕様に定められた機能・性能を実現するために、CPU で実行あるいは処理される命令／データの集合体。CPU で実行あるいは処理される命令／データの集合体であれば、それをソフトウェアに含め、ソフトウェア開発プロセス標準の適用対象とすることを基本とする。ドライバファームウェア・OS・ミドルウェアなどに対し、その特徴に応じたより適切な開発プロセスを採用する場合に限り、それをソフトウェアには含めず、ソフトウェア開発プロセス標準の適用対象から除いてもよい(たとえば、ハードウェアとそのドライバを一体開発する場合などが、該当ケースとして考えられる)。
ソフトウェアクリティカリティクラス	ソフトウェアの機能をシステムの特性(安全や信頼性など) に応じて、割り当てられるクラスであり、A、B、C、D の4段階がある。付録 11 参照。
ソフトウェア結合	ソフトウェアの構成要素を段階的に結合させていくこと。(付録 10 参照)
ソフトウェア検証計画	開発するソフトウェアの検証に関し、範囲・内容・方法・環境(試験装置など)・時期を計画するもの。 妥当性確認計画を含むこともある。
ソフトウェア試験計画	ソフトウェア検証計画において、試験でその機能・性能を検証すると識別された項目に対して、以下を計画するもの。 試験目的 試験対象ソフトウェア品目 試験コンフィギュレーション 試験設備 スケジュール 試験体制

用語	説明
ソフトウェア試験仕様	ソフトウェアが要求仕様を満たすことを証明し得る試験条件と、期待される実行結果が、十分に具体化・詳細化され、曖昧さがなく表現されているもの。 なお、要求仕様が検証可能な形式で表現されている場合、ソフトウェア試験仕様に相当するものとして扱うこともできる。
ソフトウェア使用マニュアル	使用者がソフトウェアを使用する上で必要となる情報をまとめたもの。ソフトウェア単体の操作マニュアルの他、コンピュータシステム運用マニュアル、業務運用マニュアルも含まれる。
ソフトウェア製品	ソフトウェア、ソースコードおよび関連文書を指す。
ソフトウェア要求仕様の安定度(成熟度)	ソフトウェア要求仕様の抽出・分析が十分になされて、以降の仕様の変更可能性が十分小さいことを示す指標。具体的に何を指標としてどのように評価するかは定義は任意である。また、安定度と成熟度を区別する必要はない。一般的に、ソフトウェア要求仕様における要求内容の本質的な変更あるいは詳細化する変更は、ソフトウェア要求仕様をインプットとするプロセスのコストや納期および品質に影響を与える。ソフトウェア要求仕様の安定度・成熟度は、このような観点で評価が可能である指標を採用することが望ましい。
ソフトウェアライフサイクル	要求分析から利用終了までのソフトウェアの生涯期間。
タスク	アクティビティを構成する要素であり、個々の作業に対応するもの。
妥当性確認	客観的証拠を提示することによって、特定の意図された用途または適用に関する要求事項が満たされていることを確認すること。(JIS Q 9000:2015(ISO 9000))
テーラリング	個々のシステム開発プロジェクトなどに対して、本標準で規定するプロセスをプロジェクトの特性に応じて変更し、適切なフレームを設定すること。

用語	説明
適応保守	引渡し後、変化したまたは変化している環境において、ソフトウェア製品を使用できるように保ち続けるために実施するソフトウェア製品の修正。 注記 適応保守は、運用ソフトウェア製品の運用環境変化に順応するために必要な改良を提供する。これらの変更は、環境の変化に歩調を合わせて実施する必要がある。たとえば、オペレーティングシステムの更新が必要になったとき、新オペレーティングシステムに適応するためには、幾つかの変更が必要になる場合がある。(JIS X 0161 : 2008 (ISO/IEC 14764 : 2006))
独立検証・妥当性 確認 (Independent Verification and Validation: IV&V)	ソフトウェア開発を担当する組織と独立した組織が実施する検証および妥当性確認のこと。独立性については予算的、技術的、管理的な観点が考慮される。
トレーサビリティ	複数のインプット、アウトプットなどの対応。たとえば、上位仕様と下位仕様との間、設計仕様とソースコードとの間、コンピュータシステムに対する要求仕様とコンピュータシステム総合試験仕様などがある。
ナレッジ資産	再利用ソフトウェア品目、再利用可能なコードライブラリ、リファレンスアーキテクチャ、設計要素（たとえば、アーキテクチャまたはデザインパターン）、プロセス、基準、ドメイン知識に関連するその他の技術情報（たとえば、トレーニング資料）、教訓、および組織のナレッジを反映した環境（イネーブリングシステムまたはサービス）の総称。（付録6 参照）
プロジェクト	ミッションを達成する手段として設定され、特定の資源と時間のもと時限的組織により実施する有期的活動。
プロセス	互いに関連をもったアクティビティの集合で、インプットをアウトプットに変換するもの。
ミッション	プロジェクトの終了時（定常運用期間を 終了したとき等）において、あるいは終了後の利用・研究等を経て、最終的に獲得すべき状態または成果を含む、特定の計画・活動のこと。

用語	説明
モジュール	ソフトウェア開発における最小単位のことを言う。モジュール単体で動作するものもあれば、他のモジュールと組み合わせて動作するものもある。
モデルベース技術	システム／ソフトウェアを抽象化したもので、対象とするシステム／ソフトウェアを一定の視点および一定の抽象レベルで表したモデルを用いた、開発手法の一つ。
問題	調査と是正措置を必要とする事象や不確実性で、対処しなければ現実化する望ましくない事象／一連の事象／状態／状況のこと。
要求	コンピュータシステムあるいはソフトウェアに対して求められる機能・性能などで、具体化・詳細化が不十分、あるいは、表現に曖昧さがあリ漠然とした期待に近いものも含まれる。
要求仕様	コンピュータシステムあるいはソフトウェアに対して求められる機能・性能などで、具体化・詳細化が十分にされており、実現可能性などを考慮して定量的に表現したもの。原則として、要求仕様は検証可能な形式で表現し、複数の要求仕様の間で相互に矛盾がないものでなければならない。ただし、採用する開発プロセスや要求する機能・性能の性質上、要求仕様の適切な表現形式が検証可能な形式とならない場合には、以下のような方法で要求仕様の検証可能性を補完してもよい。 (1) 要求を満足することをコンピュータシステムの利用者と合意するプロセスを計画し、ソフトウェア検証計画に含める。 (2) 要求仕様の検証に十分と認めうる試験仕様を合わせて示す。 また、要求仕様には法律や規則、プロジェクト方針のような制約事項も含まれる。
予防保守	引渡し後のソフトウェア製品の潜在的な障害が運用障害になる前に発見し、是正を行うための修正。(JIS X 0161 : 2008 (ISO/IEC 14764 : 2006))
リスク	システムの安全性やプロジェクトを取り巻く環境に対して起こりうる危険と、その結果起こり得る好ましくない事象の影響度合い。

4 一般要求事項

本標準は、ソフトウェアライフサイクルにおけるプロセスを、主ライフサイクルプロセスとして大きく3つのプロセス、支援ライフサイクルプロセスとして大きく8つのプロセスに分類し、定めるものである。定めるプロセスを図4.1および表4.1に示す。

主ライフサイクルプロセスとは、対象ソフトウェアの開発に直接かかわるソフトウェアライフサイクルにおけるプロセスであり、開発・運用・保守で行うプロセスの集まりである。

支援ライフサイクルプロセスとは、対象ソフトウェアの開発に間接的にかかわるソフトウェアライフサイクルにおけるプロセスであり、主ライフサイクルプロセスを補い、必要に応じ他のプロセスから呼び出されるプロセスの集まりである。

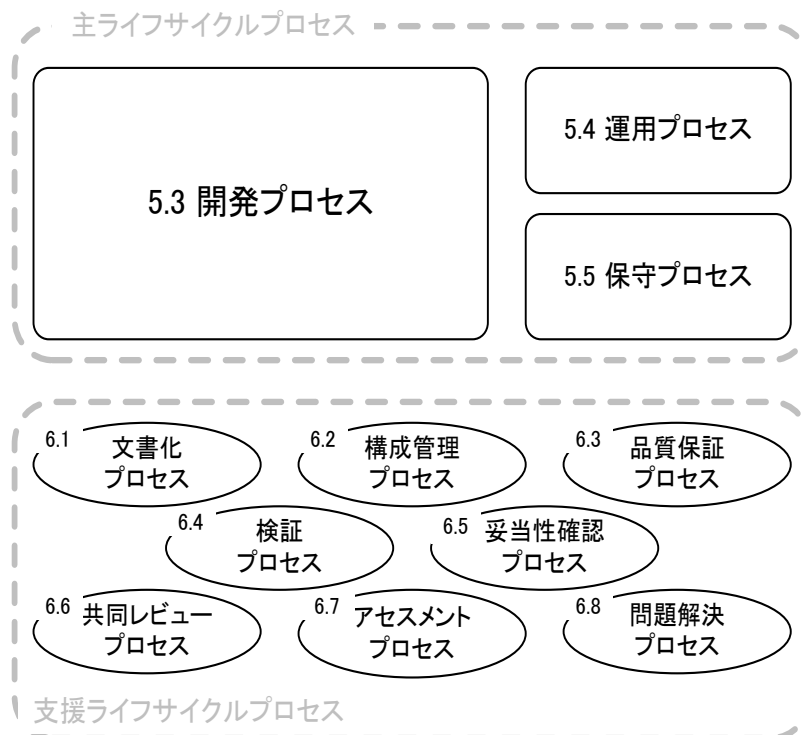


図 4.1 プロセス全体図

各プロセスについては、本書の章番号順に実施することを求めているものではない。また、同一のアクティビティを複数のプロセスに記載していることがある。たとえば、主ライフサイクルプロセスの1つである開発プロセスにおいてソフトウェアの検証に関するアクティビティを記載しているが、これは支援ライフサイクルプロセスの1つである検証プロセスのアクティビティでもある。

プロセスの分類は、視点の違いによる分類と考えるとわかりやすい。本標準では、ソフトウェアライフサイクルに対し、様々な視点からプロセスを定め、全体として(重複は許容し)不足のない内容とする方針を取っている。

表 4.1 プロセス一覧

プロセス		説明
主 ライフ サイクル プロセス	5.1 (欠番)	
	5.2 (欠番)	
	5.3 開発	開発の観点から行われるべきプロセス。 要求分析・設計・製作・試験・目標プラットフォームへのイン ストール(組み込み)・供給・導入・受け入れなど。
	5.4 運用	運用の観点から行われるべきプロセス。 運用に関する計画や規則の立案・運用試験・運用・ユーザサポ ートなど。
	5.5 保守	保守の観点から行われるべきプロセス。 保守に関する計画や規則の立案・問題把握・修正・廃棄など。
支 援 ライフ サイクル プロセス	6.1 文書化	各プロセスの成果の記録に関するプロセス。
	6.2 構成管理	ソフトウェアや文書の管理に関するプロセス。
	6.3 品質保証	プロセスが本標準に適合していることおよびプロセスが計画通 りに行われていることの確認に関するプロセス。
	6.4 検証	客観的証拠を提示することによって、規定された要求事項が満 たされていることを確認するプロセス。
	6.5 妥当性確認	客観的証拠を提示することによって、特定の意図された用途ま たは適用に関する要求事項が満たされていることを確認するプ ロセス。
	6.6 共同レビュー	文書の内容について、立場の異なる複数の当事者が共同で行う レビューに関するプロセス。
	6.7 アセスメント	プロセスの実施状況を確認し、改善項目を識別するプロセス。
	6.8 問題解決	プロセスを実施する中で発生した問題の解決に関するプロセ ス。

4.1 テーラリング

本標準は、プロセスの適切な具体化・テーラリングの上、適用されることを想定している。テーラリングは次の手順によって、実施する。

- (1) 開発するソフトウェアのクリティカリティクラス(以下、SW CC という。)を付録 11 に従い、設定する。
- (2) (1)で設定した SW CC に加えてプロジェクトの特性を加味して、表-付録 12 に示す要求マッピングマトリックスを基本に、本標準の要求をテーラリングする。

5 主ライフサイクルプロセス

ここでは、次の主ライフサイクルプロセスを定義する。

- (1) 欠番
- (2) 欠番
- (3) 開発プロセス
- (4) 運用プロセス
- (5) 保守プロセス

5.1 欠番

5.2 欠番

5.3 開発プロセス

開発プロセスは、

コンピュータシステム要求分析プロセス

コンピュータシステム方式設計プロセス

ソフトウェア要求分析プロセス

ソフトウェア設計プロセス

ソフトウェア製作プロセス

ソフトウェア統合プロセス

ソフトウェア統合試験プロセス

目標プラットフォームへのインストール(組み込み)プロセス

コンピュータシステム統合およびコンピュータシステム総合試験プロセス

ソフトウェア製品の供給と導入プロセス

ソフトウェア製品の受け入れプロセス

の総称であり、それぞれのプロセスではアクティビティ・インプット・アウトプットを定める。

各プロセスおよび各アクティビティの実施順序は本書に記載する順番である必要はないが、各アクティビティの実施順序やプロセス間の関係など開発プロセス全体構成を定義した上、開発プロセス全体の管理方法を定義し、開発プロセスの管理を行うこと。プロセスを実施するにあたり、必要に応じ

て、モデルベース技術の利用を検討すること。

5.3.1 プロセス開始の準備

ソフトウェア開発の開始時に、以下の要求を満たすアクティビティを実施すること。

- (1) 開発戦略を定義すること。戦略には以下を含めること。
 - (a) 開発方針と規約
 - (ア) 適切な安全性、セキュリティ、プライバシー、環境活動に対する方針
 - (イ) プログラミングおよびコーディング規約
 - (ウ) 単体試験方針
 - (b) 再利用ソフトウェアを使用する場合、コンピュータシステムへの適合性を確認する方法および入手経路の安全性
 - (c) ソフトウェア結合、ピアレビュー、ウォークスルー・レビューの実施
 - (d) 手作業で変更管理を行う場合、ソフトウェア製作中の構成管理
 - (e) コンピュータシステムの廃棄に伴う、関連データとソフトウェア移行作業の優先順位
 - (f) ナレッジ資産
 - (ア) ナレッジ資産の有効期間における、獲得と維持計画。
 - (イ) ナレッジ資産を受け入れ、認定し、廃止するための基準
 - (ウ) ナレッジ資産の変更を制御するための手順
 - (エ) 機密または機微なデータおよび情報の保護、制御、およびアクセスのための計画、メカニズム、および手順
- (2) 開発戦略に基づいて、以下の情報を含むソフトウェア開発計画を立案すること。
 - (a) ソフトウェア開発の目的と制約
 - (b) 対象となるコンピュータシステムの範囲
 - (c) 対象となるソフトウェアの識別
 - (d) ソフトウェア開発のプロセスの識別とその関係性の定義(運用プロセス、保守プロセスなどを考慮する)
 - (e) ソフトウェア開発プロセスとアクティビティの定義(※)
 - (f) 各開発プロセスの役割、権限および責任を含むアクティビティの実施管理計画
 - (g) 審査計画

- (h) 開発文書体系の整備と、各開発プロセスにおけるインプット／アウトプットの関係
- (i) 文書作成計画(作成部門と作成時期を含む)
- (j) 適切な作業分割および作業単位ごとの作業計画(スケジュール・マイルストーン)と達成基準を含む進捗管理の方法
- (k) ソフトウェア開発・検証で使用する環境(シミュレータ、実ハードウェア、試験環境など、イネープリングシステムまたはサービス)を獲得あるいはアクセスを取得すること。
- (l) 使用する COTS およびナレッジ資産に関する管理計画(以下を含むものとする)
 - (ア) COTS 品目およびナレッジ資産の識別
 - (イ) 識別した COTS 品目およびナレッジ資産に対する品質保証プロセスの定義
- (m) コンピュータシステムとの適合性評価計画
- (3) ソフトウェア開発計画を文書化し、合意を得ること。

※：ソフトウェアを組み込んだコンピュータシステムとしてのアクティビティなども含む。

5.3.1.1 アウトプット

- (1) ソフトウェア開発計画

5.3.2 全開発プロセス適用事項

5.3.2.1 アクティビティ

開発プロセス全体を通じて、以下の要求を満たすアクティビティを実施すること。

- (1) ソフトウェア開発計画を開発の状況に応じて更新、管理すること。
- (2) ソフトウェア開発進捗を把握すること。必要に応じ管理者へ報告すること。

5.3.2.2 インプット

- (1) ソフトウェア開発計画

5.3.2.3 アウトプット

- (1) ソフトウェア開発計画(更新)
- (2) ソフトウェア開発進捗報告

5.3.3 コンピュータシステム要求分析

5.3.3.1 アクティビティ

コンピュータシステム要求分析に対し、以下の要求を満たすアクティビティを実施すること。

- (1) 要求抽出
 - (a) 開発するコンピュータシステムの機能的境界を明確にすること。
 - (b) 開発するコンピュータシステムへの要求および運用コンセプトを分析し、運用シナリオを作成すること。
 - (c) 開発するコンピュータシステムに必要な状態遷移（動作モードを含む）を識別すること。
 - (d) 開発するコンピュータシステムの関連システムにおいて別途定義されている状態遷移がある場合は、それとの関係を明確化すること（付録7 参照）。
- (2) 要求仕様作成
 - (a) 運用シナリオに基づき、実現性・整合性を確認し、コンピュータシステムに対する要求仕様を定義すること。
 - (b) 要求仕様には、コンピュータシステムで取り扱う、データおよびデータベースに対する仕様を含めること。
 - (c) リスク、コンピュータシステムの重大性、重要な品質特性に関する仕様を含めること。
 - (d) インタフェース要求を分析し、要求仕様にする。また、インタフェース要求について、その相手方と相互に内容の解釈を含めて合意を得ること。
 - (e) コンピュータシステムに対する要求仕様の根拠を明確化し、コンピュータシステムへの要求など上位要求とのトレーサビリティを評価し、維持すること。
 - (f) 要求仕様の各要求項目について、要求根拠、検討手法などを明らかにするとともに、実現可能性を評価すること。
 - (g) COTS または再利用ソフトウェアを利用する場合には、要求仕様との整合性を解析すること。
 - (h) 適切な利害関係者に要求分析内容をフィードバックするとともに、作成した要求仕様は利害関係者とレビューを実施し合意を得ること。
 - (i) 要求仕様に含まれる問題点、不備、不整合を識別し、対策を講じること。
 - (j) 要求仕様の各要求項目について、検証可能性を評価すること。

5.3.3.2 インプット

- (1) コンピュータシステムへの要求
- (2) 運用コンセプト

5.3.3.3 アウトプット

- (1) 運用シナリオ
- (2) コンピュータシステムに対する要求仕様
- (3) インタフェース仕様
- (4) コンピュータシステムに対する要求仕様とコンピュータシステムに対する要求とのトレーサビリティの評価結果
- (5) コンピュータシステムに対する要求仕様根拠および実現可能性評価結果
- (6) COTS、再利用ソフトウェア適合性評価結果
- (7) コンピュータシステムに対する要求仕様検証可能性評価結果

5.3.4 コンピュータシステム方式設計

5.3.4.1 アクティビティ

コンピュータシステム方式設計に対し、以下の要求を満たすアクティビティを実施すること。

- (1) コンピュータシステムに対する要求仕様および運用シナリオに基づき、コンピュータシステム方式を設計し、構成する品目およびその種別(ハードウェア、ファームウェア、ソフトウェア、運用)を明確にすること。
- (2) コンピュータシステムに対する要求仕様をコンピュータシステムの各構成品目に要求として割り当てること。
- (3) 上記の設計結果をまとめて、コンピュータシステム方式設計仕様を作成すること。
- (4) ソフトウェア品目に割り当てられた要求の実現可能性を評価すること。
- (5) コンピュータシステム方式設計仕様に対する設計根拠と前提条件(運用前提など)を明らかにし、適切な評価を実施すること。
- (6) コンピュータシステム方式設計仕様に対し、コンピュータシステムに対する要求仕様とのトレーサビリティを評価すること。

- (7) ソフトウェアに関するインタフェース要求を抽出すること。
- (8) コンピュータシステムに対する要求仕様および運用シナリオに基づき、コンピュータシステム方式設計に対する評価基準を定め、それに基づき、コンピュータシステム方式設計結果を評価すること。また、コンピュータシステム方式設計の選択根拠を記録すること。

5.3.4.2 インプット

- (1) 運用シナリオ
- (2) コンピュータシステムに対する要求仕様

5.3.4.3 アウトプット

- (1) コンピュータシステム方式設計仕様
- (2) ソフトウェアへの要求(分析後の運用シナリオを含む)
- (3) インタフェース要求
- (4) コンピュータシステム方式設計仕様とコンピュータシステムに対する要求仕様とのトレーサビリティの評価結果
- (5) コンピュータシステム方式設計結果の評価結果、およびコンピュータシステム方式設計の選択根拠

5.3.5 ソフトウェア要求分析

5.3.5.1 アクティビティ

ソフトウェア要求分析に対し、以下のアクティビティを実施すること。

- (1) コンピュータシステム方式設計仕様、インタフェース要求、ソフトウェアへの要求(機能外要求を含む)を分析し、ソフトウェア要求仕様を作成すること。
- (2) 求められるソフトウェアの状態遷移(動作モードを含む)を識別すること。
- (3) 開発するソフトウェアの関連システムにおいて別途定義されている状態遷移がある場合は、それとの関係を明確化すること(付録7 参照)。
- (4) ソフトウェア要求仕様の各要求項目には、個別に識別子を付与すること。
- (5) ソフトウェア要求仕様には、ソフトウェアで取り扱う、データおよびデータベースに対する仕様を含めること。

- (6) ソフトウェア要求仕様には、ソフトウェアに実装する、異常検知および、処理機能に関する仕様を含めること。
- (7) ソフトウェア要求仕様には、リスク、ソフトウェアの重大性、重要な品質特性に関する仕様を含めること。
- (8) ソフトウェア要求仕様には、ユーザインタフェース (有する場合)、ユーザへ提供する情報、ユーザトレーニングに対する仕様を含めること。
- (9) ソフトウェア要求仕様には、運用中のシステムに対してソフトウェアを移行する場合は、移行条件を満たす仕様を含めること。
- (10) インタフェース要求を分析し、仕様化すること。また、インタフェース仕様について、その相手方と相互に内容の解釈を含めて合意を得ること。
- (11) ソフトウェア要求仕様に関して、コンピュータシステム方式設計仕様・インタフェース要求とのトレーサビリティおよび整合性を解析して、文書化し、維持すること。
- (12) ソフトウェア要求仕様の各要求項目について、要求根拠を明らかにするとともに、実現可能性を評価すること。
- (13) COTS または再利用ソフトウェアを利用する場合には、ソフトウェア要求仕様との整合性およびコンピュータシステム方式設計仕様との適合性を解析すること。
- (14) ソフトウェア要求仕様の前提となっている運用前提および制約事項を抽出すること。
- (15) 適切な利害関係者に要求分析内容をフィードバックするとともに、作成したソフトウェア要求仕様は利害関係者とレビューを実施し合意を得ること。
- (16) ソフトウェア要求仕様に含まれる問題点、不備、不整合を識別し、対策を講じること。
- (17) ソフトウェア要求仕様およびインタフェース仕様の各要求項目について、検証可能性を評価するとともに、検証方法を含むソフトウェア検証計画 (妥当性確認計画を含む) を立案すること。
- (18) ソフトウェア検証計画に対しソフトウェア機能・性能・インタフェースの視点で検証網羅性を評価するとともに、ソフトウェア要求仕様およびインタフェース仕様に対する試験計画性を評価すること。
- (19) ソフトウェア検証計画において、試験環境と実ハードウェアの挙動の差異の影響を受ける試験を実施する場合、あるいは、試験を実施せずレビュー／解析などでの確認を実施する場合は、その確認対象の識別と確認方法が適切であるかの評価も含めること。

5.3.5.2 計測活動

ソフトウェア要求仕様の確定状況および品質レベルを評価し、ソフトウェア要求分析に対し、計測に関する以下の活動を実施すること。

- (1) ソフトウェア要求仕様の安定度(成熟度)を評価するために収集するデータの定義と、その評価方法の定義を行うこと。なお、計測結果は当該プロジェクトのプロジェクト管理のインプットとなる他、後のプロジェクトでの参照データとなる。
- (2) 上記(1)で定義したデータの収集および評価を計画すること。
- (3) 上記(1)で定義したデータの収集および評価を実施し、その結果を記録すること。

5.3.5.3 インプット

- (1) コンピュータシステム方式設計仕様
- (2) ソフトウェアへの要求(分析後の運用シナリオを含む)
- (3) インタフェース要求

5.3.5.4 アウトプット

- (1) ソフトウェア要求仕様
- (2) インタフェース仕様
- (3) ソフトウェア要求仕様トレーサビリティおよび整合性評価結果
- (4) ソフトウェア要求仕様根拠および実現可能性評価結果
- (5) COTS、再利用ソフトウェア適合性評価結果
- (6) 運用前提および制約事項
- (7) ソフトウェア検証計画(妥当性確認計画を含む)
- (8) ソフトウェア検証計画の検証網羅性および試験計画性評価結果
- (9) ソフトウェア要求安定度(成熟度)評価結果
- (10) ソフトウェア要求分析計測活動結果

5.3.5.5 審査

アウトプットに対して、ソフトウェア要求審査を実施すること。5.3.5.4 に基づき、審査対象を選択し、ソフトウェア開発計画などの計画に定義すること。また、要処置事項について(期限、対処状況、影響度などに応じ)適切にフォローアップを行うこと。審査を実施した場合、審査完了後に審査記録を

作成するとともに、審査における審査員の職制、評価時間・質問数・指摘数などの定量的データを取得し、審査の品質評価を行うこと。

5.3.6 ソフトウェア設計

本標準では、特に基本設計および詳細設計を分割していないが、実際の開発プロセスでは、必要に応じ基本、詳細と分割して、実施すること。

5.3.6.1 アクティビティ

ソフトウェア設計に対し、以下の事項を満たすアクティビティを実施すること。

【基本設計】

- (1) 設計指針（ソフトウェアアーキテクチャなど）および考慮すべき設計特性を選定し、優先度を考慮した設計を行うこと。
- (2) ソフトウェア要求仕様に基づき、機能分割、モジュール分割を行うとともに、各機能を構成するモジュール間の関係、モジュール間およびモジュールの構造を明確にし、ソフトウェア（基本）設計を行うこと。
- (3) ソフトウェア（基本）設計には、ソフトウェア要求に定義された機能外要求（処理時間要求、メモリのリソース要求など）に対する設計、配分を含めること。
- (4) インタフェース仕様について、境界と相互作用に配慮して、機能分割、モジュール分割を踏まえて、詳細化を行うこと。また、インタフェース仕様について、その相手方と相互に内容の解釈を含めて合意を得ること。

【詳細設計】

- (5) 機能分割、モジュール分割に従い、各モジュールの設計を実施し、ソフトウェア（詳細）設計を行うこと。
- (6) インタフェース仕様について、境界と相互作用に配慮して、モジュールの設計に基づいて詳細化を行うこと。また、インタフェース仕様について、その相手方と相互に内容の解釈を含めて合意を得ること。

【基本、詳細設計共通】

- (7) ソフトウェア設計結果に基づき、ソフトウェア（基本、詳細）設計仕様を作成すること。
- (8) 必要な設計技法や組織において維持している過去の知見を識別し、準備または取得すること。

と。

- (9) ソフトウェア設計とソフトウェア要求仕様・インタフェース仕様・必要な関連文書とのトレーサビリティおよび整合性を解析し、文書化し、維持すること。
- (10) ソフトウェア設計の前提となっている運用前提および制約事項を明らかにすること。
- (11) 各ソフトウェア設計項目について、必要に応じて設計根拠を明らかにするとともに、実現可能性、試験可能性（試験ケースを含む）を評価すること。（付録9 参照）
- (12) COTS または再利用ソフトウェアを利用する場合には、ソフトウェア設計との適合性を解析すること。
- (13) ソフトウェア検証計画に従って、ソフトウェア試験計画およびソフトウェア試験仕様を立案すること。
 - (a) ソフトウェア試験仕様は、以下の観点を考慮すること。
 - (ア) 運用シナリオ
 - (イ) インタフェース仕様
 - (ウ) 最大負荷
 - (エ) ソフトウェア要求仕様およびソフトウェア設計仕様に対する網羅性
 - (オ) 例外や故障などの異常事象
 - (カ) COTS／再利用ソフトウェア品目のコンピュータシステムに対する適合性
- (14) 運用前提および制約事項が新たに追加された場合あるいは明らかになった場合には、運用前提および制約事項を更新すること。

5.3.6.2 計測活動

ソフトウェア設計の進捗リスクを評価できるように、ソフトウェア設計に対し、計測に関する以下の活動を実施すること。

- (1) ソフトウェア設計の進捗管理／リスク評価のために収集するデータの定義と、その評価方法の定義を行うこと。
- (2) 上記(1)で定義したデータの収集および評価を計画すること。
- (3) 上記(1)で定義したデータの収集および評価を実施し、その結果を記録すること。

5.3.6.3 インプット

- (1) ソフトウェア要求仕様

- (2) インタフェース仕様
- (3) 運用前提および制約事項
- (4) COTS、再利用ソフトウェア適合性評価結果
- (5) ソフトウェア検証計画

5.3.6.4 **アウトプット**

- (1) ソフトウェア(基本、詳細)設計仕様
- (2) インタフェース仕様(更新)
- (3) ソフトウェア設計トレサビリティおよび整合性評価結果
- (4) ソフトウェア設計根拠および実現可能性評価結果
- (5) COTS、再利用ソフトウェア適合性評価結果(更新)
- (6) 運用前提および制約事項(更新)
- (7) ソフトウェア試験計画
- (8) ソフトウェア試験仕様
- (9) ソフトウェア設計計測活動結果

5.3.6.5 **審査**

アウトプットに対してソフトウェア設計審査を実施すること。5.3.6.4 に基づき、審査対象を選択し、ソフトウェア開発計画などの計画に定義すること。また、要処置事項について(期限、対処状況、影響度などに応じ)適切にフォローアップを行うこと。審査を実施した場合、審査完了後に審査記録を作成するとともに、審査における審査員の職制、評価時間・質問数・指摘数などの定量的データを取得し、審査の品質評価を行うこと。

5.3.7 **欠番**

5.3.8 **ソフトウェア製作**

5.3.8.1 **アクティビティ**

ソフトウェア製作において、以下の事項を満たすアクティビティを実施すること。

- (1) 制約、ソフトウェア設計仕様、およびインタフェース仕様に基づきソースコードを作成し、ソースコードレビューを実施すること。
- (2) エラー処理の明確な実装指針を考慮すること。
- (3) ソースコードはコーディング規約に準拠したものとすること。
- (4) ソースコードチェックツールなどのツールで静的解析を実施し、ソースコードの品質評価を実施すること。なお、表 5.3.8-1 のモジュールの循環的複雑度(Cyclomatic Complexity)の基準を満たすこと。

表 5.3.8-1 モジュールの循環的複雑度(Cyclomatic Complexity)

SW CC	A	B	C	D
モジュールの循環的複雑度 (Cyclomatic Complexity)	15 以下	15 以下	15 以下	20 以下

- (5) ソフトウェア検証計画およびソフトウェア試験計画に従い単体試験仕様を作成すること。
- (6) 単体試験仕様に従い単体試験を実施し、実施結果を単体試験の合否判定可能な形式で記録すること。
- (7) 単体試験は、表 5.3.8-2 のソースコードに対する試験網羅率の基準を満たすよう実施すること。ただし、試験網羅率の基準を達成できない場合には、試験対象外のコードに分析、検査、または設計のレビューを適用すること。

表 5.3.8-2 ソースコードに対する試験網羅率

SW CC	A	B	C	D
C0：命令文カバレッジ	100%	100%	※1	※1
C1：分岐カバレッジ	100%	100%	※1	※1
MC/DC：改良条件判定カバレッジ	100%	※1	※1	※1

※1：プロジェクトの特性に応じて、基準を設定、合意すること。

- (8) ソースコードとソフトウェア設計仕様のトレーサビリティを解析し、その結果を記録すること。なお、ソースコード上で使用する名称（変数名、関数名など）と設計仕様との対応付けを確認すること。
- (9) 運用前提および制約事項が新たに追加された場合あるいは明らかになった場合には、運用前提および制約事項を更新すること。

5.3.8.2 計測活動

品質を評価できるように、ソフトウェア製作に対し、計測に関する以下の活動を実施すること。

- (1) ソースコードの品質評価のために収集データの定義と、その評価方法の定義を行うこと。
- (2) 上記(1)で定義したデータの収集および評価を計画すること。
- (3) 上記(1)で定義したデータの収集および評価を実施し、その結果を記録すること。
- (4) 上記(3)で評価した結果を、定期的にまたは任意のマイルストーンごとに報告すること。

5.3.8.3 インプット

- (1) ソフトウェア設計仕様
- (2) インタフェース仕様
- (3) 運用前提および制約事項
- (4) ソフトウェア検証計画
- (5) ソフトウェア試験計画

5.3.8.4 アウトプット

- (1) ソースコード
- (2) 運用前提および制約事項(更新)
- (3) 単体試験仕様
- (4) 単体試験記録
- (5) トレーサビリティ解析記録
- (6) ソフトウェア製作計測活動結果

5.3.8.5 審査

アウトプットに対して、ソフトウェア製作審査を実施すること。また、要処置事項について(期限、対処状況、影響度などに応じ)適切にフォローアップを行うこと。審査を実施した場合、審査完了後に審査記録を作成するとともに、審査における審査員の職制、評価時間・質問数・指摘数などの定量的データを取得し、審査の品質評価を行うこと。

5.3.9 欠番

5.3.10 ソフトウェア統合

5.3.10.1 アクティビティ

ソフトウェア統合に対し、以下の事項を満たすアクティビティを実施すること。

- (1) 統合の基準およびソフトウェア機能を検証するポイント（正しいインタフェースの動作と完全性）を目的に応じて選別し定義すること。
- (2) システム／ソフトウェア要求、アーキテクチャ、設計に含まれている統合における制約条件を識別すること。
- (3) ソフトウェア設計仕様に基づきソフトウェアを統合し、ソフトウェア統合後のベースラインを確定すること。
- (4) ソフトウェア統合中のデバッグ情報を記録するとともに、必要に応じ、問題解決プロセス(6.8 参照)および構成管理プロセス(6.2 参照)などの関連するプロセスを実施すること。

5.3.10.2 計測活動

品質を評価できるように、ソフトウェア統合に対し、計測に関する以下の活動を実施すること。

- (1) ソフトウェア統合中のデバッグ情報について、ソフトウェア製品の品質評価のために収集データの定義と、その評価方法の定義を行うこと。
- (2) 上記(1)で定義したデータの収集および評価を計画すること。
- (3) 上記(1)で定義したデータの収集および評価を実施し、その結果を記録すること。
- (4) 上記(3)で評価した結果を、定期的にまたは任意のマイルストーンごとに報告すること。

5.3.10.3 インプット

- (1) ソースコード(単体)
- (2) 運用前提および制約事項
- (3) ソフトウェア設計仕様

5.3.10.4 アウトプット

- (1) ソースコード(統合後)
- (2) ソフトウェア(統合後)

- (3) 統合に対する制約条件の識別結果
- (4) ソフトウェア統合計測活動結果

5.3.11 ソフトウェア統合試験

5.3.11.1 アクティビティ

ソフトウェア(統合)試験に対し、以下の事項を満たすアクティビティを実施すること。

- (1) 試験準備
 - (a) ソフトウェア製作およびソフトウェア統合の結果を受け、必要であればソフトウェア試験仕様を更新すること。
 - (b) ソフトウェア試験は、以下の観点を考慮すること。
 - (ア) 運用シナリオ
 - (イ) インタフェース仕様
 - (ウ) 最大負荷
 - (エ) ソフトウェア要求仕様およびソフトウェア設計仕様に対する網羅性
 - (オ) 例外や故障などの異常事象
 - (カ) COTS/再利用ソフトウェア品目のコンピュータシステムに対する適合性
 - (c) ソフトウェア検証計画、ソフトウェア試験計画およびソフトウェア試験仕様に従い、ソフトウェア(統合)試験手順を作成すること。
 - (d) 運用前提および制約事項が新たに追加された場合あるいは明らかになった場合には、運用前提および制約事項を更新すること。
 - (e) 試験仕様と手順を確認すること。
- (2) 試験の実施
 - (a) ソフトウェア(統合)試験手順に従い試験を実施すること。
 - (b) ソフトウェア(統合)試験においては、必要に応じ試験結果のクイックレビューを実施し、試験継続の可否の判断を行うこと。
 - (c) ソフトウェア(統合)試験の試験環境・試験データ・供試ソフトウェアの構成やバージョンなどの構成の情報について記録し、試験条件の再現性を確保すること。
 - (d) 試験結果を記録し、適切に保管するとともに必要に応じて結果を提示すること。
 - (e) ソフトウェア(統合)試験においてソフトウェアまたはソフトウェア(統合)試験仕様に

修正が発生した場合は、以前実施した共同レビュー、検証、妥当性確認、試験などのアクティビティの有効性を評価し、必要な場合は再度実施すること。

5.3.11.2 計測活動

品質を評価できるように、ソフトウェア(統合)試験に対し、計測に関する以下の活動を実施すること。

- (1) 品質指標データの取得
 - (a) ソフトウェア(統合)試験において発見された不具合は、試験ケースに関する情報など、関連する情報とともに記録すること。
- (2) 品質指標データの設定

上記(1)以外の品質指標を設定・収集・評価をする場合には、以下を実施すること。

 - (a) 試験中の品質評価のための指標を設定すること。
 - (b) 識別したデータを収集すること。
 - (c) 識別したデータの分析評価方法を定義すること。
 - (d) 識別したデータの分析評価をすること。
- (3) データの分析評価結果を、定期的にまたは任意のマイルストーンごとに報告すること。

5.3.11.3 インプット

- (1) インタフェース仕様
- (2) ソフトウェア要求仕様
- (3) ソフトウェア設計仕様
- (4) ソフトウェア検証計画
- (5) ソフトウェア試験計画
- (6) ソフトウェア試験仕様
- (7) 運用前提および制約事項
- (8) ソースコード(統合後)
- (9) ソフトウェア(統合後)
- (10) 運用シナリオ

5.3.11.4 アウトプット

- (1) ソフトウェア(統合)試験手順
- (2) ソフトウェア(統合)試験記録(合否判定結果を含む)
- (3) 運用前提および制約事項(更新)
- (4) ソースコード(試験後)
- (5) ソフトウェア(試験後)
- (6) ソフトウェア試験仕様(更新)
- (7) ソフトウェア(統合)試験計測活動結果

5.3.11.5 審査

アウトプットに対して、ソフトウェア試験審査を実施すること。5.3.11.4に基づき、審査対象を選択し、ソフトウェア開発計画などの計画に定義すること。また、要処置事項について(期限、対処状況、影響度などに応じ)適切にフォローアップを行うこと。審査を実施した場合、審査完了後に審査記録を作成するとともに、審査における審査員の職制、評価時間・質問数・指摘数などの定量的データを取得し、審査の品質評価を行うこと。

5.3.12 目標プラットフォームへのインストール(組み込み)

5.3.12.1 アクティビティ

ソフトウェアを目標プラットフォームに組み込むために、以下の事項を満たすアクティビティを実施すること。

- (1) 目標プラットフォームにインストール可能な形態でソフトウェアを準備し、そのソフトウェアの構成管理情報(ファイル名・バージョン情報など)を取得すること。
- (2) リリースするソフトウェアの構成管理情報、目標プラットフォームへのインストール計画、スケジュール、手順、インストール結果の確認手順、運用前提および制約事項を準備すること。

なお、インストール計画の立案においては、以下の観点を考慮すること。

- (a) インストールに必要なイネープリングシステム／サービスの識別とアクセス権取得
- (b) インストールに対する事前検証の設定
- (c) ソフトウェアまたはインストールサービスの提供および支援(取得者に対するサポート提供)

- (3) インストールを実施する前には、インストール準備の状況(運用条件・制約事項への対応完了、事前検証の実行確認、等)を確認すること。その後、インストール手順に従い、ソフトウェアを目標プラットフォームへ組み込むこと。

(ただし、すでに目標プラットフォームに組み込まれている場合は省略可能)

- (4) インストール結果の確認手順・確認結果に従い、ソフトウェアが正しくインストールされていることを確認すること。

5.3.12.2 インプット

- (1) ソフトウェア
- (2) 運用前提および制約事項

5.3.12.3 アウトプット

- (1) ソフトウェア(目標プラットフォームにインストール可能な形態)
- (2) 構成管理情報
- (3) インストール計画、スケジュール、手順
- (4) ソフトウェアを組み込んだコンピュータシステム
- (5) インストール結果の確認手順・確認結果

5.3.13 コンピュータシステム統合およびコンピュータシステム総合試験

5.3.13.1 アクティビティ

コンピュータシステム統合およびコンピュータシステム総合試験に対し、以下の事項を満たすアクティビティを実施すること。

- (1) 試験準備
 - (a) システム／ソフトウェア要求、アーキテクチャ、設計に含まれている統合における制約条件を識別し、コンピュータシステム統合を行うこと。
 - (b) コンピュータシステム統合の基準およびソフトウェア機能を検証するポイント（正しいインタフェースの動作と完全性）を目的に応じて選別し定義すること。
 - (c) ソフトウェア検証計画に従い、コンピュータシステム総合試験仕様・コンピュータシステム総合試験手順を作成すること。

- (d) コンピュータシステム総合試験仕様は、以下の観点を考慮すること。
 - (ア) 運用シナリオ
 - (イ) ソフトウェア要求仕様
 - (ウ) 最大負荷
 - (エ) コンピュータシステムに対する要求仕様およびコンピュータシステム方式設計仕様に対する網羅性
 - (オ) 例外や故障などの異常事象
 - (カ) COTS／再利用ソフトウェア品目のコンピュータシステムに対する適合性
 - (e) 試験準備(試験手順確認など)で見つかったインシデントおよび問題を記録・管理すること。
- (2) 試験の実施
- (a) コンピュータシステム総合試験手順に従い、試験を実施すること。
 - (b) 必要に応じ、運用シナリオベースの試験をシミュレータなどのツールの活用によって実施し、運用シナリオに対する検証網羅性を確認すること。
 - (c) コンピュータシステム総合試験の試験環境・試験データ・供試ソフトウェアの構成やバージョンなどの情報について記録し、試験条件の再現性を確保すること。
 - (d) 試験結果を記録し、適切に保管すること。
 - (e) コンピュータシステム総合試験においてソフトウェアまたはコンピュータシステム総合試験仕様に修正が発生した場合は、以前実施した共同レビュー、検証、妥当性確認、試験などのアクティビティの有効性を評価し、必要な場合は再度実施すること。

5.3.13.2 計測活動

品質を評価できるように、コンピュータシステム統合およびコンピュータシステム総合試験に対し、計測に関する以下の活動を実施すること。

- (1) 品質指標データの取得
 - (a) コンピュータシステム総合試験において発見された不具合は、試験ケースに関する情報など、関連する情報とともに記録すること。
 - (b) 試験準備(試験手順確認など)で発見されたインシデントおよび問題は、試験ケースに関する情報など、関連する情報とともに記録すること。
- (2) 品質指標データの設定

上記(1)以外の品質指標を設定・収集・評価をする場合には、以下を実施すること。

- (a) コンピュータシステム総合試験中の品質評価のための指標を設定すること。
 - (b) 識別したデータを収集すること。
 - (c) 識別したデータの分析評価方法を定義すること。
 - (d) 識別したデータの分析評価をすること。
- (3) データの分析評価結果を定期的にまたは任意のマイルストーンごとに報告すること。

5.3.13.3 インプット

- (1) コンピュータシステムに対する要求仕様
- (2) コンピュータシステム方式設計仕様
- (3) ソフトウェア要求仕様
- (4) ソフトウェア検証計画
- (5) 運用前提および制約事項
- (6) 運用シナリオ
- (7) ソフトウェアを組み込んだコンピュータシステム

5.3.13.4 アウトプット

- (1) コンピュータシステム総合試験仕様
- (2) コンピュータシステム総合試験手順
- (3) コンピュータシステム総合試験記録(合否判定結果を含む)
- (4) 運用前提および制約事項(更新)
- (5) 統合に対する制約条件の識別結果(更新後)
- (6) コンピュータシステム統合および総合試験計測活動結果

5.3.14 ソフトウェア製品の供給と導入

5.3.14.1 アクティビティ

ソフトウェア製品の供給と導入に、以下のアクティビティを行うこと。

- (1) マニュアルの作成
 - ソフトウェアの使用マニュアルを作成すること。

(2) 供給の準備

ソフトウェア製品が供給可能な状態にあることを確認し、確認結果を記録すること。

(3) 導入計画の立案

ソフトウェア製品の導入(既存システムとの置き換え、一時的な並行運転を含む)を計画し、手順を作成すること。導入計画および導入手順は、ソフトウェアの目標プラットフォームへのインストールだけでなく、コンピュータシステムを実動作環境に導入するための作業を考慮すること。

(4) 導入の実施および結果の記録

導入計画および導入手順に基づいてソフトウェアを目標プラットフォームにインストール、コンピュータシステムを実動作環境に導入すること。また、その結果を記録すること。

5.3.14.2 インプット

- (1) ソフトウェア要求仕様
- (2) ソフトウェア設計仕様
- (3) ソースコードまたはソフトウェア
- (4) インストール手順

5.3.14.3 アウトプット

- (1) ソフトウェア使用マニュアル
- (2) ソースコードまたはソフトウェア
- (3) 供給可能な状態にあることの確認結果
- (4) インストール手順
- (5) 導入計画
- (6) 導入手順
- (7) 導入結果の記録

5.3.15 ソフトウェア製品の受け入れ

5.3.15.1 アクティビティ

ソフトウェア製品の受け入れにおいて以下のアクティビティを行うこと。

- (1) 受け入れ検査・試験について、計画を立案し、仕様および手順を作成すること。なお、これを供給者が実施した試験で代える場合には、取得者がその試験計画、仕様、手順の承認を行うこと。
- (2) 上記(1)の計画、仕様、手順に従い、受け入れ検査・試験を実施すること。また、受け入れ検査・試験の記録を残すこと。
- (3) 受け入れするソフトウェア製品が供給可能な状態にあることの確認結果を確認すること。

5.3.15.2 インプット

- (1) ソフトウェア使用マニュアル
- (2) ソースコードまたはソフトウェア
- (3) 構成管理情報
- (4) 供給可能な状態にあることの確認結果の確認記録
- (5) インストール手順

5.3.15.3 アウトプット

- (1) 受け入れ検査・試験の計画
- (2) 受け入れ検査・試験の仕様
- (3) 受け入れ検査・試験の手順
- (4) 受け入れ検査・試験の記録

5.4 運用プロセス

運用プロセスは、意図された環境でソフトウェアを含むコンピュータシステムを運用し、製品に対する顧客およびユーザへのサポートを提供することを目的とする。

このプロセスは、次のアクティビティからなる。

- (1) プロセス開始の準備
- (2) 運用試験
- (3) ソフトウェアを含むコンピュータシステムの運用
- (4) 運用結果の管理
- (5) 顧客およびユーザサポート

5.4.1 プロセス開始の準備

5.4.1.1 運用戦略の定義

- (1) 運用戦略を定義すること。なお、戦略には以下を考慮すること。
 - (a) サービス導入後、定期的に運用されサービス廃止に至るまでに期待される各基準（容量、稼働率、応答時間、安全性 等）
 - (b) 既存のサービスを維持するための修正を考慮したソフトウェアまたはコンピュータシステムのリリース基準およびスケジュール
 - (c) 各運用モード（通常運用／準備段階／運用試験実施／想定される災害・障害発生時）を実装するための手法
 - (d) 性能レベルを評価するための運用にかかわる尺度

5.4.1.2 運用計画の立案

運用プロセスを実施するための計画（運用計画）を運用戦略に基づいて立案し、実行すること。また、運用に関する標準を定めること。なお、計画には以下を考慮すること。

- (1) 運用プロセスに係る、ソフトウェアまたはコンピュータシステムの要求分析、アーキテクチャ、設計、実装、移行における制約条件を識別すること。
- (2) 運用支援に必要なイネーブリングシステムやサービスを識別すること。
- (3) 使用されるイネーブリングシステムまたはサービスを獲得あるいはアクセスを取得すること。

5.4.1.3 運用時の問題管理手続きの確立

問題解決プロセス(6.8 参照)に従い、問題報告の受理、記録、解決、追跡およびそれらの状況の通知に関する処理手順を確立すること。

5.4.1.4 ソフトウェアを含むコンピュータシステムの運用およびユーザサポートに係る作業手順の確立

ソフトウェアを含むコンピュータシステムの運用環境下での試験手順、変更要求を保守プロセス(5.5 参照)に渡すための手順およびソフトウェアまたはコンピュータシステムを実運用にリリースす

るための手順を確立すること。

5.4.2 運用試験

運用試験を実施するための環境を構築すること。ソフトウェアまたはコンピュータシステムがリリースされるごとに運用試験を実施すること。運用試験が計画通りに実行され終了されていることを確認し、レビューを実施すること。該当するソフトウェアまたはコンピュータシステムが規定されている判定基準を満たした時点で実運用にリリースすること。

5.4.3 ソフトウェアを含むコンピュータシステムの運用

以下の要求を満たすタスクを実施すること。

- (1) ソフトウェア使用マニュアルに従った所定の運用環境において運用する。
- (2) 運用とサービス維持のため、必要に応じて資材やリソースを適用する。
- (3) 以下を考慮し、運用を監視する。
 - (a) 運用戦略（たとえば、運用に係る作業手順）に対する準拠
 - (b) ソフトウェアに起こり得る侵害や、データの機密性・完全性などの重大事項に関する記録および報告
 - (c) ソフトウェア、コンピュータシステム、またはサービスの性能が許容可能なパラメータの範囲外である場合の記録
- (4) 運用戦略に沿った、運用障害のリスクを最小限に抑えるための運用手順を作成し、可能な範囲で自動化する。
- (5) 運用戦略に沿って、下記事項を確認するために測定結果を分析する。
 - (a) サービス性能は、許容できるパラメータ以内であるか、合意された作業量に対し合意されたサービスレベルにある。
 - (b) ソフトウェアまたはコンピュータシステム、およびサービスにおける、可用性および応答時間は、許容範囲内である。
 - (c) 潜在的な改善項目が特定され、優先度が付けられている。
- (6) 必要に応じて、災害・障害時運用を実施する。

5.4.4 運用結果の管理

このアクティビティは、次のタスクからなる。

- (1) 運用の結果と、運用中に直面した問題を記録すること。
- (2) 運用上のインシデントと問題を記録し、解決の終結まで追跡すること。
- (3) 運用サービスと構成品目のトレーサビリティを維持すること。
- (4) ベースラインに選定された主要成果物および情報項目を提供すること。

5.4.5 顧客およびユーザサポート

顧客およびユーザサポートに対し、以下のタスクを行うこと。

- (1) インシデント、問題、サービスへの要求を解決するため、顧客およびユーザに対しサポートサービスを提供すること。依頼およびサポートの対応状況は記録し、適切な対応が行われるよう管理する。
- (2) ユーザサポートにおいて問題が識別された場合、5.4.1.3 で確立した手順に従って問題を解決する。
- (3) 識別された問題に対して、暫定的な回避策がある場合、その回避策を顧客およびユーザに提供する。
- (4) 提供するソフトウェア／コンピュータシステムまたはサービスに対する顧客およびユーザの要求に対する満足度を確定する。

5.5 保守プロセス

保守プロセスは、サービスを提供するソフトウェアまたはコンピュータシステムの能力を維持することを目的とし、対象であるソフトウェア(設計文書を含む)および、開発環境の維持または変更が必要となった場合の作業について規定するものである。

このプロセスは、次のアクティビティからなる。

- (1) プロセス開始の準備
- (2) 問題把握および修正分析
- (3) 修正の実施

- (4) ソフトウェアの書き換え
- (5) ロジスティクス支援の実施
- (6) 保守およびロジスティクス結果の管理
- (7) 移行
- (8) ソフトウェア廃棄

5.5.1 プロセス開始の準備

5.5.1.1 保守戦略の定義

- (1) 保守戦略を定義すること。なお、戦略には以下を考慮すること。
 - (a) 運用の可用性要求に準拠したソフトウェア保守変更の実施／検証／配付／実装するための優先順位、標準スケジュールおよび手順の確立
 - (b) 修正の必要性を判断するための技法と手法の確立
 - (c) 保守対象となる製品のバージョンおよび保守に必要な情報へのアクセスを獲得するための優先順位とリソースの確立（たとえば、計画されたフェーズごとのインストール、保守パッチ、またはソフトウェア更新）
 - (d) 問題解決および保守時における、合意されたデータに対する権利、およびソフトウェアまたはコンピュータシステム内のデータへの影響
 - (e) 正しくない、または無許可のソフトウェアコンポーネントがソフトウェアまたはコンピュータシステムにとりこまれないようにするための手法
 - (f) ソフトウェア障害の発生時における、ソフトウェア製品の修正による影響と、ソフトウェア製品の障害箇所をそのままにした場合のリスクとの比較
- (2) ソフトウェア以外についてロジスティクス戦略（保管対象要素の数、種類、保管場所、条件、予測される入替率、保管年限、更新頻度）を定義すること。ロジスティクスは、必要な資材・資源が適切な場所および時間に適切な数量および品質で提供されるよう支援するものである。
- (3) ソフトウェアまたはコンピュータシステムの要求、アーキテクチャ、設計に含まれる保守上の制約条件を定義すること。
- (4) 入手可能で運用可能で持続可能な保守およびロジスティクス活動となるよう活動内容を調整すること。
- (5) 保守をサポートするのに必要なインネーブリングシステムまたはサービスを識別し計画を作

成すること。

- (6) 使用するイネーブリングシステムまたはサービスを獲得あるいはアクセスを取得すること。

5.5.1.2 保守計画の立案

保守プロセスを実施するための計画(保守計画)を保守戦略に基づいて立案し、実行すること。この計画には、以下の記載を含めること。

- (1) 開発したソフトウェア製品の維持管理方法
- (2) 体制および関連する文書
- (3) データ(保守に必要な管理情報を含む)
- (4) 記録
- (5) 保守環境(開発時の環境など)
- (6) その他のプロセス(問題解決プロセス(6.8 参照)、構成管理プロセス(6.2 参照)など)を適切に維持する活動

5.5.2 問題把握および修正分析

- (1) 計画に従い、保守対象の維持または変更についての依頼内容の分析および対応要否案を評価決定すること。
- (2) ソフトウェアの修正、ソフトウェアの書き換えを行う場合は、保守計画に従って修正案の合意を得ること。

5.5.3 修正の実施

修正が必要な場合は、修正の実施計画および手順に従い、修正を実施すること。必要に応じて本標準の該当するプロセスを再度実施すること。

5.5.4 ソフトウェアの書き換え

運用期間中におけるソフトウェアの部分的な修正(パッチ)・ソフトウェアの機能追加/拡張などの作業についても、本標準を適用する。なお、書き換え時に予期せぬシステム障害が発生した場合は、

ソフトウェアまたはコンピュータシステムを運用状態に復元させること。

※: これは、運用期間中のソフトウェア製品の開発～廃棄プロセスとまったく同じ開発プロセスとしなければならないということではない。書き換えを想定するプロジェクトでは、あらかじめ書き換えに関する開発～廃棄プロセスを具体化・テーラリングしておくこと。
基本的には本標準を適用し、適合性を評価すること。

5.5.5 ロジスティクス支援の実施

このアクティビティは、次のタスクからなる。

- (1) プロジェクトのライフサイクルを通して、ソフトウェア製品を支援するためにリソースを獲得する。
- (2) 書き換えたソフトウェア要素およびインテグレーションシステムの品質と可用性、および、それらを配付する仕組みと格納中のそれらの持続的な完全性を監視する。
- (3) ソフトウェア製品を配付するための仕組みを構築する。
- (4) ソフトウェア製品のロジスティクス支援に関して必要な活動を、計画し実施していることを確認する。

5.5.6 保守およびロジスティクス結果の管理

このアクティビティは、次のタスクからなる。

- (1) インシデントと問題（解決策を含む）、および重要な保守とロジスティクスの結果を記録する。
- (2) インシデント、問題、および保守とロジスティクスの活動の傾向を識別し、記録する。
- (3) 保守されているソフトウェア製品またはコンピュータシステムの要素のトレーサビリティを維持する。
- (4) ベースラインに選定された主要成果物と情報項目を提供する。
- (5) ソフトウェア製品またはコンピュータシステムと保守支援に対する顧客およびユーザ満足度を監視し測定する。

5.5.7 移行

このアクティビティはソフトウェア製品を新しい運用環境に移行する場合のアクティビティである。

5.5.7.1 移行戦略の定義

ソフトウェア製品の移行準備として移行戦略を定義すること。なお、戦略には以下を考慮すること。

- (1) 移行の種別と成功基準の確立
- (2) 開発、運用に伴うソフトウェア製品またはコンピュータシステム更新のための移行頻度
- (3) 移行中のセキュリティリスク、切断、中断時間の最小化
- (4) 外部インタフェースを通して受け取るデータを含め、旧システムから新システムに移行するデータの保管／破壊／変換、および妥当性確認
- (5) 緊急時対応策（問題対応、バックアップ、旧システムの最終状態への復旧）
- (6) 段階的または同期したシステム移行を実施する場合、実施中の作業と整合する移行スケジュール
- (7) インタフェースを有する外部組織／オペレータ／システム管理者／ソフトウェアおよびサービスの利用者を含む、関係者に対する変更管理
- (8) 移行システムまたはその要素の妥当性確認戦略
- (9) システム設計書、ユーザ用ドキュメント、試験仕様書の更新に伴うユーザサポートと保守の開始
- (10) 新システムと旧システムが正規に切り替わる際の移行、運用、廃棄の並列実施

5.5.7.2 移行計画の立案と実行

移行戦略に基づいてソフトウェア製品の移行計画を立案し、実施すること。

- (1) 移行に伴い統合するソフトウェアへの要求、アーキテクチャ／設計上の制約条件を含む、移行のための要求分析および要求定義
- (2) 移行頻度と移行スケジュール
- (3) ソフトウェアの移行（または新規導入）によって変更が必要な施設、サイト、通信ネットワーク、ハードウェア環境
- (4) オペレータ、ユーザやシステムサポートにかかわる関係者用の文書および教育
- (5) 移行のためのツール（イネーブリングシステム／サービス含む）
- (6) ソフトウェア製品およびデータの変換

- (7) 移行リハーサル
- (8) 移行の実施
- (9) 移行結果の確認
- (10) 旧環境に対するサポート

5.5.7.3 ユーザへの通知

移行計画および実施内容についてユーザに通知すること。この通知には以下の観点を含めること。

- (1) 旧環境のサポートを停止する理由
- (2) 新しい環境の説明および利用開始日
- (3) 旧環境のサポート停止後にユーザが選択できるサポートの説明

5.5.7.4 移行記録の管理

ソフトウェアの移行実施に伴う以下の記録を管理すること。

- (1) 移行時に発生したインシデントおよび問題
- (2) 移行したソフトウェア要素のトレーサビリティの維持
- (3) ベースライン対象として提供した重要な成果物と情報アイテム

5.5.7.5 旧環境の保管

旧環境に関する文書・ログなどの記録は保管することが望ましい。

5.5.8 ソフトウェア廃棄

ソフトウェアの廃棄には以下の観点を考慮すること。

- (1) 運用および保守を行っている組織がサポートを停止するための廃棄戦略を定義すること。
 なお、戦略には以下を考慮すること。
 - (a) 永久廃棄となるシステム機能および提供サービスの特定
 - (b) ソフトウェアの所有と保持責任、またはソフトウェアのデータと知的所有権の破棄
 - (c) 関係者、社会、環境への影響を避けるため、社会的および物理的に製品を受容可能な状態に変化させること
 - (d) ハードウェアおよびデータの長期的な状態を元にした、衛生、安全、セキュリティと

プライバシーを考慮した廃棄

- (e) 重要な廃棄処理に関する影響をうける関係者への通知
 - (f) 廃棄のためのスケジュール、行動、責任と資源の識別
 - (g) 廃棄に向けた、システム／ソフトウェア要求、アーキテクチャ、設計または実装技術での制約条件の特定
 - (h) 廃棄を支援するのに必要なイネーブリングシステムおよびサービスに対する特定と計画
 - (i) 必要なイネーブリングシステムおよびサービスを獲得あるいはアクセスを取得すること。
 - (j) ソフトウェアまたはデータを保管する場合の設備、保管場所、点検基準と貯蔵期間
 - (k) 廃棄したソフトウェアの再利用を防止する方法
- (2) 廃棄戦略に基づき廃棄計画を立案すること。なお、計画立案には、ユーザを参加させること。
- (3) 廃棄計画および実施内容をユーザに通知すること。通知には、次の項目を含めること。
- (a) ソフトウェアまたはコンピュータシステムの入れ替えまたは増強に関する説明と利用開始日
 - (b) 現行ソフトウェアまたはコンピュータシステムのサポートを停止する理由
 - (c) サポート停止後、ユーザが選択できる他の方法の説明
- (4) 廃棄計画に基づき以下の活動を実施する。
- (a) 新しいソフトウェアあるいはコンピュータシステムへの切り替えを円滑にするために、新旧ソフトウェアまたはコンピュータシステムの並行運用を行い、この期間中にユーザの教育訓練を行うことが望ましい。
 - (b) 対象のソフトウェア、コンピュータシステム、データを撤去する。
 - (c) 新しいソフトウェアあるいはコンピュータシステムを、旧システムの最終状態または決められた状態に設定する。
 - (d) 撤去したソフトウェア／コンピュータシステム／データを、再利用／リサイクル／調整／分解検査／保管／破棄する。
- (5) 予定した廃棄時期がきたとき、関係者全員に通知する。関連するすべての開発文書、ログおよびソースコードは保管することが望ましい。
- (6) 廃棄後のソフトウェアあるいはコンピュータシステムに対して、衛生、安全、セキュリティ

とプライバシー、環境への悪影響を特定し、レビューおよび監査できるように長期に渡ってデータを集めること。

6 支援ライフサイクルプロセス

ここでは、次の支援ライフサイクルプロセスを定義する。

- (1) 文書化プロセス
- (2) 構成管理プロセス
- (3) 品質保証プロセス
- (4) 検証プロセス
- (5) 妥当性確認プロセス
- (6) 共同レビュープロセス
- (7) アセスメントプロセス
- (8) 問題解決プロセス

支援プロセス中のアクティビティおよびタスクは、そのプロセスを実行する組織が責任をもつ。この組織は、そのプロセスを存在させ、機能させる。

6.1 文書化プロセス

このプロセスは、次のアクティビティからなる。

- (1) プロセス開始の準備
- (2) 作成
- (3) 発行
- (4) 維持・改訂・廃棄

6.1.1 プロセス開始の準備

文書化計画として、次のことを含む計画を立案すること。

- (1) ソフトウェアまたはコンピュータシステムのライフサイクルを通じて作成する文書および作成時期を明確にすること。

- (2) 文書の作成(作成・点検・承認)・発行(発行・配付・保管)・改訂(改訂・廃棄)に関する手続きを定めること。
- (3) 文書で適用する様式(記載内容、フォーマットなど)を定めておくこと。

6.1.2 作成

文書の作成は定められた手続きに従って行うこと。下記について考慮すること。

- (1) 文書の源泉情報の適切性を確かめること。
- (2) 作成した文書について、様式・技術的内容について、点検・承認を行うこと。

6.1.3 発行

文書の発行は定められた手続きに従って行うこと。下記について考慮すること。

- (1) 文書の発行・配付は最適なバージョンであること。
- (2) 文書は、配付先を含め関連する要求事項(セキュリティ管理・バックアップなど)に従い管理すること。

6.1.4 維持・改訂・廃棄

文書の維持、改訂、廃棄は定められた手続きに従って行うこと。また、下記について考慮すること。

- (1) 作成された文書は、そこに含まれる情報の完全性を考慮すること。
- (2) 改訂した文書について、様式・技術的内容について、点検・承認を行うこと。
- (3) 不要、無効、または妥当ではない文書は廃棄すること。

6.2 構成管理プロセス

構成管理プロセスは次の管理的および技術的な手続きからなり、ソフトウェアライフサイクルを通して適用される。このプロセスは、次のアクティビティからなる。

- (1) プロセス開始の準備
- (2) 構成識別

- (3) 構成変更管理
- (4) 構成変更状況の記録
- (5) 構成変更状況の評価
- (6) リリース管理と出荷

6.2.1 プロセス開始の準備

6.2.1.1 構成管理戦略の定義

構成管理戦略を定義すること。なお、戦略には以下を考慮すること。

- (1) ソフトウェアの使用許諾ライセンス、データの利用権、および他の知的財産権の管理
- (2) リリースの頻度、優先順位、およびソフトウェアバージョンおよびソフトウェアの内容
- (3) 構成定義情報の継続的な完全性とセキュリティの妥当性に対する構成監査の戦略と責任
- (4) 運用上のソフトウェア／サービスのユーザを含めた変更管理

6.2.1.2 構成管理計画の立案

構成管理戦略に基づき構成管理計画を立案すること。なお、次のことを含む計画を立案すること。

- (1) 構成管理におけるアクティビティ
 - (a) ベースライン変更および、通常時／緊急時の変更要求の承認にかかわるリスクと影響度の評価を含むこと。
- (2) アクティビティを実施するための手続きとスケジュール
 - (a) 構成品目へのアクセス、変更、処置の制御、構成管理の開始、ベースライン構成の維持・更新の基準を含む、確立すべき必要なベースラインを含むこと。
- (3) アクティビティを実施する責任を負う組織
 - (a) 構成管理委員会の運用含む、構成管理活動の役割と責任を含むこと。
- (4) 関係する他の組織
 - (a) ソフトウェアのライフサイクルまたは合意された範囲またはプロジェクトに対する、取得者、供給者、組織を横断した適切な構成管理の統合。
 - (b) 運用上のソフトウェア／サービスのユーザを含めた構成管理・変更管理計画を含むこと。
- (5) 構成品目、構成管理成果物・記録の保管、アーカイブ、と検索手順

6.2.2 構成識別

- (1) 管理する品目を識別すること。
- (2) 管理する品目に対し、以下のような属性を識別すること。
 - (a) バージョンの参照
 - (b) その他の識別詳細情報
- (3) 管理する品目に対し、目的に応じたベースラインを確立すること。
- (4) 確立した構成ベースラインに対する関係者の合意を形成すること。

6.2.3 構成変更管理

構成変更に対して、次のものを識別し、定義すること。

- (1) 変更要求の識別と記録（ウェーバを含む）
- (2) 変更内容の分析と評価
- (3) 修正の可否（承認または不承認）
- (4) 修正された品目の実装と検証およびリリース
- (5) 修正理由および修正承認が追跡できる審査記録
- (6) 欠番

6.2.4 構成変更状況の記録

管理する品目の状況と履歴を示す管理記録を準備し、状況報告を行うこと。

6.2.5 構成変更状況の評価

事前に計画した実施スケジュールに従い、次の確認を行うこと。

- (1) 要求事項に対する品目の機能的な完全性および物理的な完全性
- (2) 承認された構成変更の妥当性

確認結果および必要な処置は記録すること。

6.2.6 リリース管理と出荷

ソフトウェア製品のリリースと出荷を手続きに従って管理すること。特に、安全性またはセキュリティが要求される重大な機能を含んだソースコードと文書は、組織の方針に従って、取扱い、保管し、包装し、出荷すること。

決められたソフトウェア製品のリリースについて、配付を管理すること。

6.2.7 構成監査の実施

構成管理計画に従って構成管理が実施されていることを確認すること。

6.3 品質保証プロセス

品質保証プロセスは、ソフトウェアまたはコンピュータシステムのライフサイクルおよびプロジェクトのライフサイクルにおける各プロセスのアクティビティやアウトプットが、本標準またはそのテーラリング結果に基づき設定した計画に従って行われていることを保証する、品質保証活動のプロセスである。

このプロセスは、次のアクティビティからなる。

- (1) プロセス開始の準備
- (2) 製品およびサービス品質の保証
- (3) プロセスの保証
- (4) 品質システムの保証
- (5) 品質保証記録の管理

6.3.1 プロセス開始の準備

このアクティビティは、次のタスクからなる。

6.3.1.1 組織の独立性の確認、適切な体制の確立

品質保証プロセス実施の管理組織を明確にし、品質保証プロセスの実施状況と妥当性について報告すること。

品質保証プロセスの管理組織には、問題の解決方法を勧告する上での組織的自由と権限を与えること。なお、品質保証の責任者は、組織的に開発組織から独立し、品質保証活動に関するすべての責任・権限を有すること。

6.3.1.2 品質保証戦略の定義

- (1) 品質保証戦略を定義すること。なお、戦略には以下を考慮すること。
 - (a) 優先度を設けて、製品およびサービスの品質に最も重要な影響を与えるプロセスとタスクに品質保証リソースを適用すること
 - (b) 品質保証活動の責任と権限
 - (c) 製品／サービスの受入基準含む、プロセス、製品およびサービスの評価基準および評価手法
 - (d) 各供給元（各供給業者を含む）への品質保証活動
 - (e) 製品またはサービスに特有かつ必須の検証、妥当性確認、監視、測定、レビュー、検査、監査またはアセスメント、試験の活動
 - (f) 問題解決、プロセスおよび製品の改善活動

6.3.1.3 品質保証活動計画の立案

品質保証戦略に基づき品質保証活動の計画を立案する。

品質保証活動の計画に以下の項目を含むこと。

- (1) 適用対象の識別
- (2) 品質保証活動に必要な資源、品質標準、方法論、手続きおよびツール(利用される規程類の識別を含む)
- (3) 検証プロセス(6.4 参照)、妥当性確認プロセス(6.5 参照)、共同レビュープロセス(6.6 参照)、アセスメントプロセス(6.7 参照)および問題解決プロセス(6.8 参照)からとの関連性と選択したアクティビティおよびタスク
- (4) スケジュール
- (5) レビューに関する一連の手続き
- (6) 品質保証活動の記録の識別と、その収集・保管・維持・廃棄などの作業に関する一連の手続

き

- (7) 購買管理・供給業者に対する品質保証に関する要求・活動
- (8) 既存ソフトウェア品目 (COTS 品目・ナレッジ資産) の管理
- (9) 出荷に関する一連の手続き
- (10) 問題の識別と、その解決におけるプロセスおよび製品の改善活動に関する一連の手続き

6.3.2 製品およびサービス品質の保証

- (1) 製品およびサービスが計画されたものが不足なく作成されていることを保証すること。
- (2) ソフトウェア製品またはコンピュータシステムの開発プロセスの各成果物への検証および妥当性確認にてその成果物が合意された要求仕様を満足していることを保証すること。
- (3) 必要に応じて独立検証・妥当性確認 (IV&V) を実施し、製品品質を保証すること。

6.3.3 プロセスの保証

- (1) ソフトウェア開発計画、運用計画または保守計画で定義されたプロセスが本標準に適合していることを保証すること。
- (2) ソフトウェア開発が開発計画、運用計画または保守計画で定義されたプロセスに従って必要なアクティビティ・タスクが行われていることを保証すること。
- (3) プロセスを支援または自動化するためのツールおよび環境が本標準のプロセスに使用可能であることを評価すること。

6.3.4 品質システムの保証

品質システムに以下に示す品質管理活動が含まれることを保証すること。

6.3.4.1 教育・訓練

対象となるソフトウェアまたはコンピュータシステムの開発・保守・運用業務に従事する要員に必要な技術・能力・資格を識別し、計画的な教育・訓練を実施すること。

6.3.4.2 購買管理・供給業者管理

(1) 購買管理

組織における購買管理方針、規則に従い、購買品(COTSを含む)の信頼性・品質が、開発するソフトウェア製品およびサービスの品質保証要求事項に合致していることを確認すること。

(2) 供給業者および購買先の選定

組織において維持している供給業者および購買先に関する能力評価および選定記録に基づき、供給業者および購買先を選定すること。

6.3.4.3 取得者支給品の管理

取得者が支給・貸与する物品の受け入れ時の検査、保管および維持管理のための手順を定め、それに基づいて実施すること。

6.3.4.4 既存ソフトウェア品目(COTS品目・ナレッジ資産)の管理

既存ソフトウェア品目(COTS品目、または再利用ソフトウェア品目)、ソフトウェアアーキテクチャなどの再利用可能なナレッジ資産に関しては以下を管理項目に含めること。当該プロジェクト、将来の再利用に関して下記の情報を管理すること。

- (1) 既存ソフトウェア品目を使用することの目的、理由、利点
- (2) 既存ソフトウェア品目の使用の可否を判断する評価項目とレベル
 - (a) 開発するソフトウェアに対する既存ソフトウェアの適合性
 - (b) 開発ソフトウェア品目に適用される要求事項とのトレーサビリティ
 - (c) 使用するソフトウェア品目の製品実績などの情報から得られるリスク
 - (d) 受け入れおよび保証条件
- (3) 既存ソフトウェア品目の使用を管理する上での留意事項と項目
 - (a) 入手可能または利用可能な関連文書
 - (b) 導入、事前準備、訓練、制約条件
 - (c) バージョンなどの識別と構成管理形態手法
 - (d) 保守および将来的支援サポート
 - (e) 知的財産権などの権利関係
 - (f) 新たに利用可能な、ナレッジ資産の識別

6.3.4.5 取扱い、保管およびラベリング

ソフトウェアまたはコンピュータシステムの取扱いおよび保管、およびラベリングを確実に実施するために、必要事項(次の項目を含む)を文書に定めそれに基づき、リリースすること。

- (1) 媒体には記録したソフトウェアを識別するためのラベリング(名称と識別子など)をすること。
- (2) 媒体からソフトウェアを読み込んだときに、ソフトウェアの識別を確認すること。

6.3.5 品質保証記録の管理

- (1) 品質保証活動に関連した記録を作成すること。
- (2) 記録の維持、保存を行うこと。

6.4 検証プロセス

検証プロセスは、コンピュータシステムが、要求とその特性を満たすことの客観的な証拠を提供することを目的とする。検証プロセスは、次のアクティビティからなる。なお、試験／レビュー／解析などの確認作業を検証プロセスに含めてもよい。

- (1) プロセス開始の準備
- (2) 検証
- (3) 検証結果の管理

6.4.1 プロセス開始の準備

6.4.1.1 検証戦略の定義

- (1) 検証戦略を定義すること。なお、戦略には以下を考慮すること。
 - (a) 検証範囲(対象となるソフトウェア、要素、成果物を含む)、検証すべき特性、および期待される検証結果の識別。なお、特性にはシステム／ソフトウェア要求、アーキテクチャ、セキュリティおよび重要な品質特性等の設計特性、インテグレーション、文書の正確さが含まれる。
 - (b) 検証の制約、限界の識別

- (c) 検証シナリオの優先順位の識別
- (2) システム／ソフトウェア要求、アーキテクチャ、設計に含まれている統合に対する制約条件を識別する。

6.4.1.2 検証計画の立案

- (1) 検証戦略に基づいて、下記を含む検証計画を立案すること。
 - (a) 検証の対象を決定し、その重要度に応じて 6.4.2 で定義するタスクの中から、適切なタスクを含めること。
 - (b) 検証（検査・分析・実証・試験）の適切な手法または技術、および検証に関する標準を選定すること。
 - (c) 検証の手順および一連の活動を定義すること。手順には、検証結果の記録、分析、保存、報告の方法が含まれる。
 - (d) 検証を支援するために必要なイネープリングシステムまたはサービスを識別し、利用可能な状態にすること。
 - (e) 検証計画に従って検証（試験／レビュー／解析など）を行い、識別されたインシデントおよび問題を問題解決プロセス（6.8 参照）に従って処置すること。

6.4.2 検証

このアクティビティは、次のタスクからなる。

6.4.2.1 プロセスの検証

次に示す観点を考慮すること。

- (1) 選択されたプロセスが適切である。
- (2) プロセスの計画が適切である。
- (3) プロセスに関係する適切な標準・環境が準備されている。
- (4) プロセスを実行するための適切な能力をもった要員が適切な人数確保されている。
- (5) プロセスが適切に実行されている。

6.4.2.2 要求の検証

次に示す観点を考慮すること。

- (1) 要求に一貫性があり、実現可能かつ検証可能である。
- (2) ソフトウェア品目に対して配分されている要求が適切である(ハードウェア品目や運用に対して配分されるべき要求を含んでいない)。
- (3) 上位の要求や品目に適用される基準、標準を満たしている。
- (4) 安全性・セキュリティなど特に注意を要する要求に関して、上位の要求や品目に適用される基準、標準が満たされていることを、適切な手法により示すことができる。

6.4.2.3 設計の検証

次に示す観点を考慮すること。

- (1) 設計は要求を満足し、要求に対してトレーサビリティが確保されている。
- (2) 入出力インタフェース・タイミング・計算機リソース(メモリ容量・処理速度など)・論理設計・処理順序と処理内容(特に初期化・終了・例外など)に関し、適切に設計されている。
- (3) 移植容易性・変更容易性・問題解決容易性などの特性について、適切に考慮されている。

6.4.2.4 ソースコードの検証

次に示す観点を考慮すること。

- (1) ソースコードは設計を満足し、設計に対してトレーサビリティが確保されている。
- (2) 入出力インタフェース・タイミング・計算機リソース(メモリ容量・処理速度など)・論理設計・処理順序と処理内容(特に初期化・終了・例外など)に関し、適切に実装されている。
- (3) 移植容易性・変更容易性・問題解決容易性などの特性について、適切に実装されている。
- (4) 安全性・セキュリティなど特に注意を要するソースコードに関して、要求や品目に適用される基準、標準が満たされていることを、適切な手法により示すことができる。
- (5) ソースコードはコーディング規約などの基準を満たしている。

6.4.2.5 統合の検証

次に示す観点を考慮すること。

- (1) ソフトウェアを構成するモジュールやデータの構成が適切で、正しいバージョンである。
各ソフトウェア品目のうちソフトウェア構成要素およびユニットが完全に、かつ、正確にソフトウェア品目として統合されている。

- (2) ソフトウェアを構成するモジュールやデータが、過不足なく統合されており、統合に対してトレーサビリティが確保されている。
- (3) 統合が、計画に従って実施されている。

6.4.2.6 文書化の検証

次に示す観点を考慮すること。

- (1) 文書の内容が正しく、一貫性がある。
- (2) 文書化が計画され、計画に従って行われている。
- (3) 文書に関する構成管理が、適切な手続きに従っている。

6.4.3 検証結果の管理

検証の実施結果を管理すること。このアクティビティは、次のタスクからなる。

- (1) 検証したソフトウェアや要素のトレーサビリティ結果の維持
- (2) ベースライン対象として選定された主要成果物および情報項目（例：検証戦略・検証手順）の提供

6.5 妥当性確認プロセス

妥当性確認プロセスは、コンピュータシステムが意図した環境や用途で使用する際に、ミッションに期待される目標や利害関係者要求を満たすことの客観的な証拠を提供することを目的とする。

このプロセスは、次のアクティビティからなる。

- (1) プロセス開始の準備
- (2) 妥当性確認
- (3) 妥当性確認結果の管理

6.5.1 プロセス開始の準備

6.5.1.1 妥当性確認戦略の定義

- (1) 妥当性確認戦略を定義すること。なお、戦略には以下を考慮すること。
 - (a) 妥当性確認の範囲（対象となるソフトウェア、要素、成果物の特性を含む）、および期待される妥当性確認結果の識別
 - (b) 妥当性確認の制約、限界の識別
 - (c) 妥当性確認シナリオの優先順位の識別
- (2) システム／ソフトウェア要求、アーキテクチャ、設計に含まれている統合に対する制約条件を識別する。

6.5.1.2 妥当性確認計画の立案

- (1) 妥当性確認戦略に基づいて、下記を含む妥当性確認計画を立案すること。
 - (a) 欠番
 - (b) 妥当性確認を計画する。計画には、次の項目を含めること。
 - (ア) 妥当性確認を必要とする品目
 - (イ) 実行する妥当性確認タスク
 - (ウ) 妥当性確認のための資源、責任および予定
 - (エ) 妥当性確認報告を配付する手続き
 - (c) 妥当性確認活動の適切な手法や技術、妥当性確認基準を選定する(※)。
 - (d) 妥当性確認の手順および一連の活動を定義すること。手順には妥当性確認結果の記録、分析、保存、報告の方法が含まれる。
 - (e) 妥当性確認を支援するために必要なイネーブリングシステムやサービスを識別し、利用可能な状態にする。
 - (f) 妥当性確認計画に従って妥当性確認を行い、識別されたインシデントおよび問題を、問題解決プロセス(6.8 参照)に従って処置すること。

※：妥当性確認の手段として試験以外の方法(分析、モデル化、シミュレーションなど)を採用してもよい。

6.5.2 妥当性確認

- (1) 妥当性確認を行う試験要求事項および試験ケースを選び、試験仕様を準備すること。
- (2) 試験ケースがソフトウェアまたはコンピュータシステムが想定する使用状況・使用方法で

あることを保証すること。

- (3) 必要に応じ次の観点を含めて、上記(1)および(2)の試験を行うこと。
 - (a) 欠番
 - (b) 故障試験
 - (c) ユーザによる業務利用試験
- (4) ソフトウェアまたはコンピュータシステムが、意図した使用方法を満足しているかどうかを確認すること。
- (5) 実環境に相当する環境において、試験すること。実環境ではなく模擬環境を用いる場合には、実環境と模擬環境の差異を評価すること。

6.5.3 妥当性確認結果の管理

妥当性確認の実施結果を管理すること。このアクティビティは、次のタスクからなる。

- (1) 妥当性確認の結果および、確認時に直面した問題のレビューとフォローアップ活動の識別
- (2) 妥当性確認時に発生したインシデントおよび問題の記録と解決状況の追跡
- (3) ソフトウェアまたは要素が利害関係者のニーズを満たすことに対する合意獲得
- (4) 妥当性確認したソフトウェアや要素のトレーサビリティ結果の維持
- (5) ベースライン対象として選定された主要成果物および情報項目の提供

6.6 共同レビュープロセス

共同レビュープロセスは、次のアクティビティからなる。

- (1) プロセス開始の準備
- (2) プロジェクト管理レビュー
- (3) 技術レビュー

6.6.1 プロセス開始の準備

- (1) ソフトウェア開発計画のマイルストーンでレビューを実施すること。また、レビュー実施者またはレビュー受審者のいずれかの当事者が必要と考えるときには、臨時のレビューを実

施することが望ましい。

- (2) レビューの実施に必要なすべての資源は、すべての当事者で合意すること。これらの資源には、人、場所、設備、ハードウェア、ソフトウェアおよびツールを含む。
- (3) すべての当事者は、レビューごとに、次の事項について合意することが望ましい。
 - (a) レビュー対象
 - (b) レビューの範囲および観点
 - (c) レビューの方法
 - (d) レビューの開始および終了の基準
- (4) レビュー時に指摘されたインシデントおよび問題を記録し、必要に応じて問題解決プロセス(6.8参照)に従って処置すること。
- (5) レビュー結果を配付すること。
- (6) すべての当事者は、指摘された問題に関し、問題解決の責任の所在と問題解決の基準を合意すること。

6.6.2 プロジェクト管理レビュー

ソフトウェア開発状況を開発計画と照らし合わせて評価し、計画に沿った開発完了に対するリスクを管理すること。また、計画に沿った開発完了が困難と推測された場合、ソフトウェア要求仕様やスケジュールの見直しを含めた計画変更を検討すること。

6.6.3 技術レビュー

技術レビューはソフトウェア品目を技術的な観点から評価するとともに、要求仕様・標準を満たすソフトウェアおよびコンピュータシステムの実現に対するリスクを明確化することを目的に実施すること。

このアクティビティは、次のタスクからなる。

6.6.3.1 審査

- (1) 審査の準備として、下記を行うこと。
 - (a) 審査を実施するにあたり、審査員を選定すること。審査員は、作業関係者だけでなく、

有識者(審査対象に関する知識を持つもの)、プロジェクト関係者(インタフェース設計者・要求者を含む)など、適切な者を選択すること。

(b) 審査対象、目的を明確にして、事前に審査に必要な資料が完成していることを確認すること。

(c) 審査目的、審査対象、審査員を明確にし、文書化すること。

(2) ソフトウェア品目について、開発進展に応じて段階的に評価すること。

(3) ソフトウェアに関する技術的活動を評価すること。

(4) 審査完了後に審査記録を作成するとともに、審査における評価時間、質問数・指摘数などの定量的データを取得し、審査の品質評価を行うこと。

6.6.3.2 ウォークスルー

ウォークスルー(ピアレビューなどを含む)は、設計や製作のエラーを早期に発見・除去し、品質を向上させる狙いから、必要に応じ、不定期に実作業担当者を中心に実施するものである。

(1) ウォークスルーの準備として、レビュー受審者は対象となる文書・ソースコードを提供すること(作成中のものも含む)。

(2) 必要な関係者(上流工程プロセス作業者／下流工程プロセス作業者など)が、対象となる文書・ソースコードに問題がないことを確認すること。

(3) ウォークスルーにより判明した質問事項、問題を記録に残し、対策が完了するまでフォローすること。フォローとは、処置内容を相互に了解することを意味する。

6.7 アセスメントプロセス

アセスメントプロセスは、プロセスの実施状況を確認し、改善項目を識別する活動である。

このプロセスは、次のアクティビティからなる。

(1) プロセス開始の準備

(2) アセスメントの実施

6.7.1 プロセス開始の準備

プロジェクトで定めるアセスメント実施責任者(以下、スポンサーという)の指示に従い、次の事項

を満足するアセスメントを計画し、スポンサーと合意すること。

- (1) 実施されているプロセスが、本標準に準ずる要求事項に適合しているかを評価する。プロセスにおける強み、弱みを識別すること。
- (2) アセスメント手法および適用する本標準、基準類に精通した者をアセッサーとして選定すること。
- (3) 要改善事項の改善提案を含むアセスメント結果をスポンサーに文書で報告すること。

6.7.2 アセスメントの実施

アセスメント計画に基づきアセスメントを実施すること。

6.8 問題解決プロセス

問題解決プロセスは、次のアクティビティからなる。

- (1) プロセス開始の準備
- (2) 問題の解決
- (3) 予防
- (4) 傾向分析

6.8.1 プロセス開始の準備

ソフトウェア製品およびサービスについてインシデントおよび問題(気づき)の発生に備え、あらかじめ、下記を決定しておくこと。

- (1) 管理の対象
- (2) 管理する期間
- (3) 解決(処置)の一連の手続き

インシデントおよび問題が発生した場合、上記に従い、記録を行い管理すること。

6.8.2 問題の解決

このアクティビティは、次のタスクからなる。

6.8.2.1 インシデントおよび問題の識別

インシデントが発生した場合、管理および既知のインシデントや問題と関連付けられるよう識別すること。

是正処置を必要とするものについては、問題として管理できるようにすること。その際に優先順位をつけることが望ましい。

インシデントおよび問題のどちらについても、発生および状況の報告を関係者に通知すること。

6.8.2.2 インシデントおよび問題の調査／分析

下記を行って、インシデントの現象と発生条件を分析すること。問題の根本原因を調査／分析して、明らかにすること。必要に応じて解決を他の組織に依頼すること。

- (1) インシデントの記録、分析、分類を行い、是正処置が必要なインシデントを問題として特定すること。
- (2) 特定した問題の記録、分析、分類を行い、調査して根本原因を明らかにすること。
- (3) インシデントおよび問題の傾向分析を行うこと。
- (4) インシデントおよび問題の状況報告を行うこと。

6.8.2.3 問題解決方法の検討

解決方法を検討すること。

- (1) ソフトウェア製品の修正を必要としない問題の回避あるいは問題の影響を軽減する方法がないか検討すること。
- (2) ソフトウェア製品の変更が必要か検討すること。

なお、解決方法には複数の選択肢(ソフトウェアを修正しない解決策を含む)が用意できることが望ましい。その場合、コスト／スケジュール／リスク／影響範囲などの観点から選択肢を評価すること。

6.8.2.4 問題解決方法の決定

計画を立案し、その計画について関係者と合意すること。

- (1) 暫定措置として問題の回避、あるいは影響を軽減する方法がある場合、計画を立案し、その

計画について関係者と合意すること。

- (2) 恒久措置として問題を解決する方法の計画を立案し、その計画について関係者と合意すること。

6.8.2.5 問題解決実施

合意した計画に従い、問題解決を実施し、利用者に通知すること。問題は、終結に至るまで追跡管理を実施する。

6.8.2.6 記録とモニタ

インシデントおよび問題解決の一連の活動およびその状況について、記録およびモニタを行い、管理すること。なお、モニタは措置の実施後、問題が解決し、また、措置に伴って新たな問題が生じていないと判断できるまでの期間継続すること。

6.8.3 予防

既知のインシデントおよび問題を予防（再発防止）するため、プロセスおよびソフトウェア製品における改善を識別する。

予防が必要なリスクが類推できる場合、予防措置を実施すること。

6.8.4 傾向分析

インシデントおよび問題の傾向分析を実施すること。

付録

付録 1 開発プロセス全体構成の例(ウォーターフォール型)

開発プロセスの全体構成について、下記に例を示す。

凡例

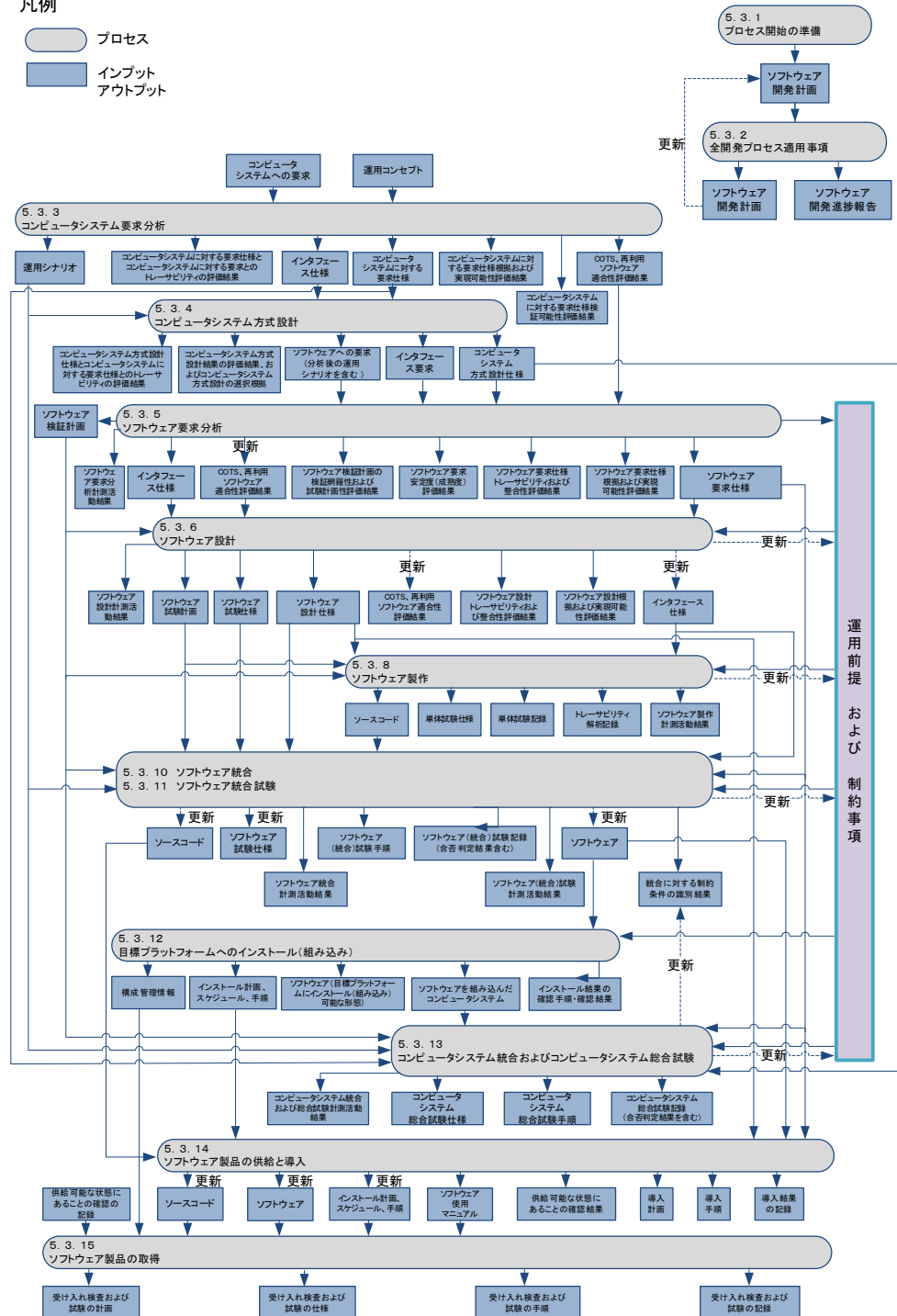
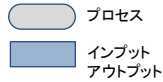


図-付録 1 開発プロセス全体構成の例(ウォーターフォール型)

付録2 インプット・アウトプットと各プロセス対応表

開発プロセスにおける各プロセスのアウトプット・インプットの対応を下記の表に示す。

表-付録2 インプット・アウトプットと各プロセス対応表

インプット・アウトプット	プロセス名称														
	5.3.1	5.3.2	5.3.3	5.3.4	5.3.5	5.3.6	5.3.7	5.3.8	5.3.9	5.3.10	5.3.11	5.3.12	5.3.13	5.3.14	5.3.15
ソフトウェア開発計画	O	M													
ソフトウェア開発進捗報告	O														
コンピュータシステムへの要求			I												
運用コンセプト			I												
運用シナリオ			O	I							I		I		
コンピュータシステムに対する要求仕様			O	I									I		
インタフェース仕様			O		M	M		I			I				
コンピュータシステムに対する要求仕様と			O												
コンピュータシステムに対する要求とのトレーサビリティの評価結果			O												
コンピュータシステムに対する要求仕様根拠および実現可能性評価結果			O												
COTS、再利用ソフトウェア適合性評価結果			O		M	M									
コンピュータシステムに対する要求仕様検証可能性評価結果			O												
コンピュータシステム方式設計仕様				O	I								I		
ソフトウェアへの要求(分析後の運用シナリオを含む)				O	I										
インタフェース要求				O	I										
コンピュータシステム方式設計仕様と				O											
コンピュータシステムに対する要求仕様とのトレーサビリティの評価結果				O											
コンピュータシステム方式設計結果の評価結果、				O											
およびコンピュータシステム方式設計の選択根拠				O											
ソフトウェア要求仕様				O	I						I		I	I	
運用前提および制約事項				O	M	M		M		I	M	I	M		
ソフトウェア検証計画				O	I			I		I	M	I	I		
ソフトウェア検証計画の検証網羅性および試験計画性評価結果				O											
ソフトウェア要求安定度(成熟度)評価結果				O											
ソフトウェア要求仕様トレーサビリティおよび整合性評価結果				O											
ソフトウェア要求仕様根拠および実現可能性評価結果				O											
ソフトウェア要求分析計測活動結果				O											
ソフトウェア設計仕様					O		I			I	I			I	
ソフトウェア設計トレーサビリティおよび整合性評価結果					O										
ソフトウェア設計根拠および実現可能性評価結果					O										
ソフトウェア設計計測活動結果					O										
ソフトウェア試験計画					O		I				I				
ソフトウェア試験仕様					O						M				
単体試験仕様							O								
単体試験記録							O								
トレーサビリティ解析記録							O								
ソフトウェア製作計測活動結果							O								
ソースコード							O			M	M			M	I
ソフトウェア										O	M	M		M	I
統合に対する制約条件の識別結果										O			M		
ソフトウェア統合計測活動結果										O					
ソフトウェア(統合)試験手順											O				
ソフトウェア(統合)試験記録(合否判定結果を含む)											O				
ソフトウェア(統合)試験計測活動結果											O				
構成管理情報												O			I
インストール計画、スケジュール、手順												O		M	I
ソフトウェアを組み込んだコンピュータシステム												O	I		
インストール結果の確認手順・確認結果												O			
コンピュータシステム総合試験仕様													O		
コンピュータシステム総合試験手順													O		
コンピュータシステム総合試験記録(合否判定結果を含む)													O		
コンピュータシステム統合および総合試験計測活動結果													O		
ソフトウェア使用マニュアル														O	I
供給可能な状態にあることの確認結果														O	
供給可能な状態にあることの確認の記録															I
導入計画														O	
導入手順														O	
導入結果の記録														O	
受け入れ検査および試験の計画														O	
受け入れ検査および試験の仕様														O	
受け入れ検査および試験の手順														O	
受け入れ検査および試験の記録														O	

凡例 I: インプット、O: アウトプット、M: インプットかつアウトプット(更新)<主なもの>

付録3 「検証」および「妥当性確認」に関する補足説明

「検証」および「妥当性確認」の差異についての補足説明として、概念を下記の例に示す。

「検証」は、前のプロセスで定められた要求事項が満たされているか確認することを目的とするのに対し、「妥当性確認」は、最初に定められた特定の意図された用途または適用に関する要求事項が満たされているか確認を行い、“要求とのズレ”の累積を小さくすることを目的とする。

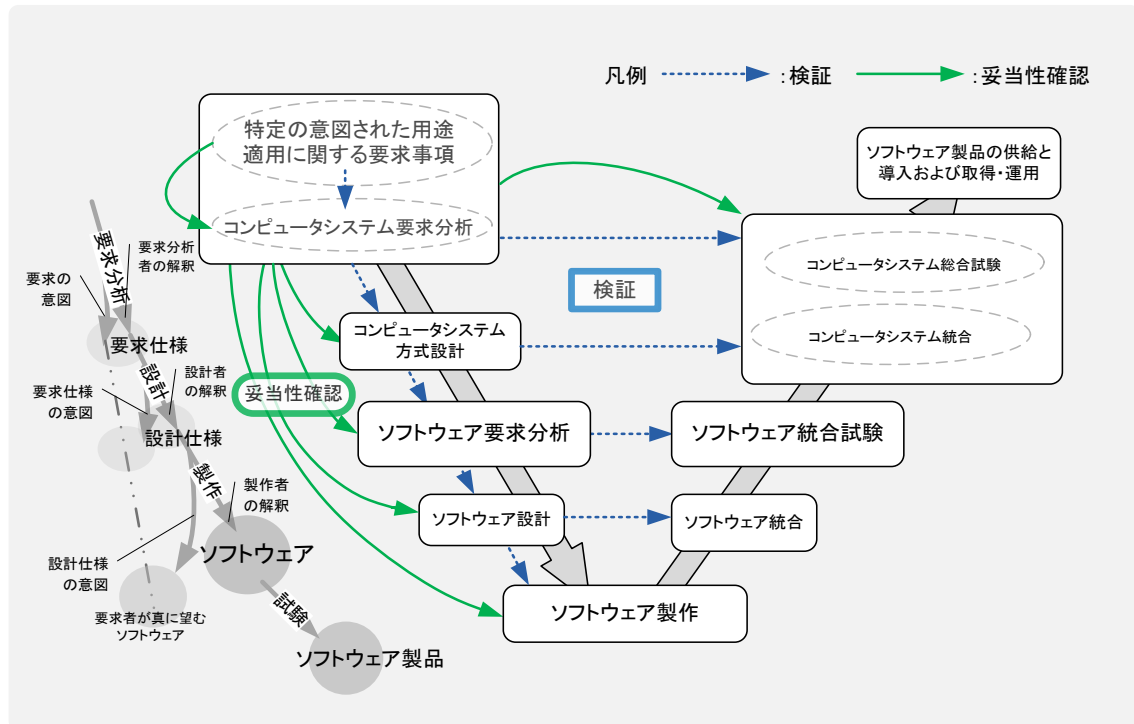


図-付録3 検証と妥当性確認の概念図(例)

付録4 問題解決プロセスと他プロセスとの関係

問題解決プロセスと他プロセスとの関係についての補足説明として、運用プロセスにおいて、インシデントが発生した場合を例に、問題解決プロセスとの関係を下記に示す。

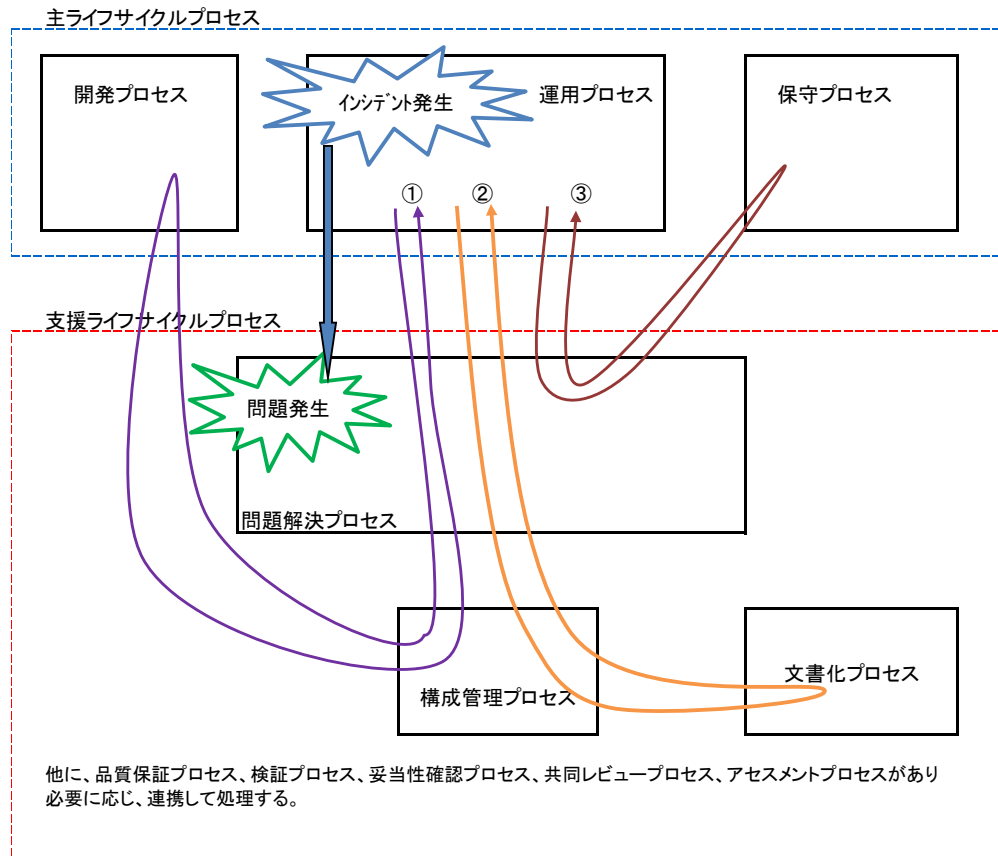


図-付録 4-1 インシデントが発生した場合の問題解決プロセスとの関係 (例)

[① ソフトウェア修正の場合]

運用中にインシデントが発生する。→問題解決プロセスへ送る。→検討の結果、問題を識別し、根本原因と、その解決方法を決定する。→構成管理プロセスから取り出す。→開発プロセスで修正を実施する。→構成管理プロセスへ戻す。→結果を問題解決プロセスで確認する。→運用プロセスへ戻す。

[② 文書改定の場合]

運用中にインシデントが発生する。→問題解決プロセスへ送る。→検討の結果、問題を識別する。→検討の結果、根本原因と、その解決方法を決定する。→構成管理プロセスで管理する。(構成管理文書か判断) →文書化プロセスで修正する。→構成管理プロセスで管理(更新)する。(構成管理文書の場合) →結果を問題解決プロセスで確認する。→運用プロセスへ戻す。

[③ 運用での対応の場合]

運用中にインシデントが発生する。→問題解決プロセスへ送る。→検討の結果、問題を識別し、根本原因と、その解決方法（運用での対応であり、ソフトウェアの修正なし）を決定する。→保守プロセスで運用対応策を実施する。→結果を問題解決プロセスで確認する。→運用プロセスへ戻す。

また、ソフトウェア開発標準における「インシデント」および「問題」の差異について、概念を下記の例に示す。

「インシデント」は、発生した変則的あるいは予期せぬイベント／状態／状況であるのに対し、「問題」は、既知ではない「インシデント」を分析した結果、調査と是正措置を必要として識別された事象／状態／状況である。

なお、「インシデント」によっては、複数の「問題」を含む場合もある。

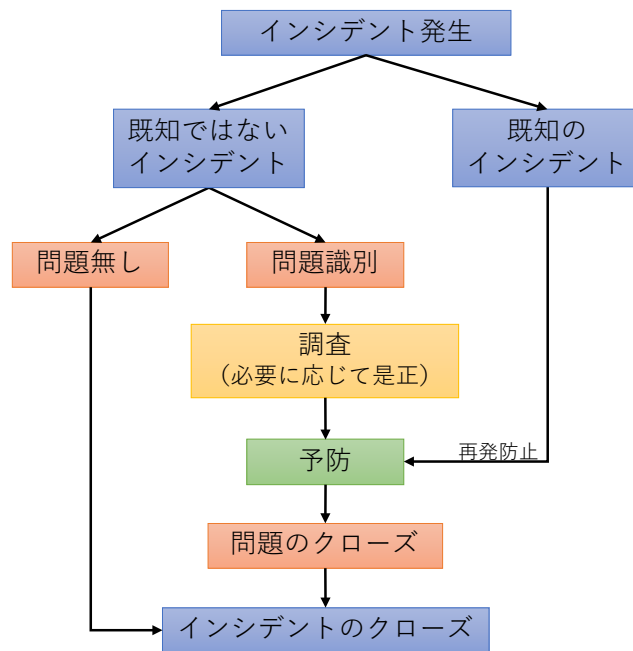


図-付録 4-2 ソフトウェア開発標準におけるインシデントと問題の処理手順概要

付録5 4つの保守タイプに関する補足説明

4つの保守タイプに関する補足説明として、地上システムを想定した場合を例に、4つの保守タイプとの関係を下記に示す。

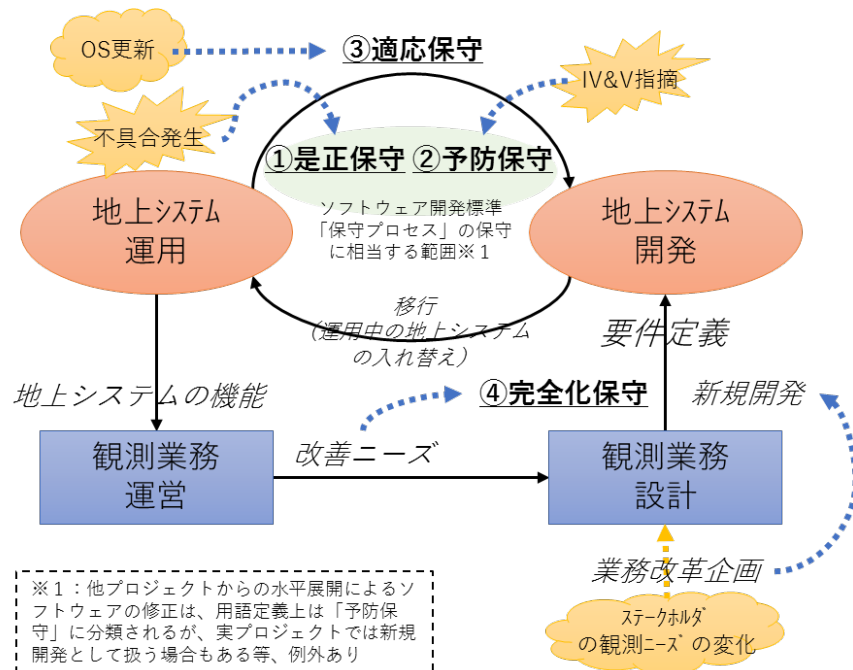


図-付録5 4つの保守タイプ、地上システム、および観測業務の関係(例)

是正

[① 是正保守]

運用中に、地上システムで意図しない事象（不具合）が発生する。→地上システムの開発メーカは、ソフトウェアを修正し、運用環境へ組み込む。

[② 予防保守]

運用へ移行後、地上システムのソフトウェアに対する IV&V を実施した結果、潜在的なバグが検出される。（上記バグによる運用上の不具合はまだ発生していない）→地上システムの開発メーカは、是正するためにソフトウェアを修正し、運用環境へ組み込む。

改良

[③ 適応保守]

地上システムの開発メーカは常に OS の更新状況を監視している。→OS が更新される。→OS の変更への対応を放置すると老朽化、セキュリティリスクが向上するため、地上システムの開発メーカはソフトウェアを修正し、運用環境へ組み込む。

[④ 完全化保守]

追跡管制隊から提示された観測業務の運営に関する改善ニーズに基づき、観測業務の設計が更新する。→上記の観測業務の設計結果を実現するため、地上システムの開発メーカは処理時間向上、機能追加等のソフトウェア修正を実施し、運用環境へ組み込む。

付録6 ソフトウェア製品、ナレッジ資産、イネープリングシステムの補足

ナレッジ資産とソフトウェア製品、ならびにナレッジ資産とイネープリングシステムは、一意に分類できず重複している。そのため、それらの関係を図-付録6で明確にする。

分類	ソフトウェア製品					環境 (イネープリングシステムまたはサービス)	
	ソフトウェア		ソースコード	関連文書			
具体例	カタログ品のソフトウェア(COTS)	自組織または外部による開発ソフトウェア(テレコマDB(注)を含む。)	自組織または外部による開発ソースコード(モデルベース開発におけるモデル、自動コード生成したコード、FOSS(Free and Open Source Software)を含む。)	すべての文書(再利用およびナレッジ資産の観点では、設計要素が対象になる。)	プロセス、基準(criteria)、ドメイン知識に関連するその他の技術情報(たとえば、トレーニング資料)、および教訓	自組織開発や供給者に対して自組織のナレッジを組み込むことを要求して取得した、組織のナレッジを反映したツール(テレコマDBを生成するツールなど)やデータ(ダイナミクスなどの模擬データなど)	組織のナレッジを反映していないツールやデータ(COTSツールをそのまま使用する場合など)
分類	ナレッジ資産						

(注) テレコマ DB は、システムやハードウェア側のナレッジ資産として管理されるものがある。本標準においては、システムやハードウェアの設計に基づくテレコマ DB を除外するが、ソフトウェア設計に基づくテレコマ DB は対象として扱う。

図-付録6 ソフトウェア製品、ナレッジ資産、イネープリングシステムの関係

付録7 開発するコンピュータシステム/ソフトウェアの関連システムに関する補足説明

5.3.3.1(1)(d)項および5.3.5.1(3)項の「関連システムにおいて別途定義されている状態遷移がある場合に、開発するコンピュータシステム/ソフトウェアの状態遷移との関係を明確化すること」について補足説明する。

(1) 対象となる関連システムの状態遷移

(a) 宇宙機システムおよびロケットシステム

開発するソフトウェアから見て、以下の状態遷移を対象とする。

- ・上位のシステムで定義されている状態遷移
- ・下位のハードウェアの状態遷移
- ・データインタフェースをもつシステム/コンポーネントの状態遷移

(b) 地上システム

関連するシステムは運用上必要な経路でデータインタフェースを取っている場合が多い。そのため、データ中継システムを含め、データ送信元からデータ受信先までのすべてのシステムの状態遷移を対象とする(注)。関連システムがソフトウェアとハードウェアで異なる状態遷移を持つような場合は、それらを個別のシステムとして考慮する。

(注) 次のものは除外してもよい。

- ・内部の状態遷移に関係なく中継だけを行っている中継システム
- ・データ送信元からデータ受信先までのシステムの状態遷移に影響されないようにインタフェースが設計されているシステム

(2) 明確にすべき関係について

宇宙機システムを例に説明する。図-付録7の宇宙機システムのシステム構成・コンポーネント構成および状態遷移において、開発するソフトウェアがコンポーネント2のソフトウェアである場合、以下の関係を明確化する。

(a) 上位システムとの関係

- ・システムレベルで定義されている省電力/通常モードと開発するソフトウェアが動作する CPU が持つ省電力/通常モードが一致しているか、一致していない場合にはどのような影響および制約を持つのかを識別し、開発するソフトウェアの状態遷移(セーフ/太陽指向/定常運用モード)との関係を明らかにする。たとえば、CPU が省電力モードになった場合、開発するソフトウェアの機能・性能が維持可能であるかなど矛盾点がないかを確認し、要すれば開発するソフトウェアの見直しを行う。
- ・システムレベルで定義されている姿勢制御モードと開発するソフトウェアの状態遷移(セーフ/太陽指向/定常運用モード)との関係を明らかにする。たとえば、システムレベルの定常運用モード時に太陽指向が含まれる場合に、開発するソフトウェアの状態遷移はどのモードなのか、などを確認する。

(b) ハードウェアとの関係

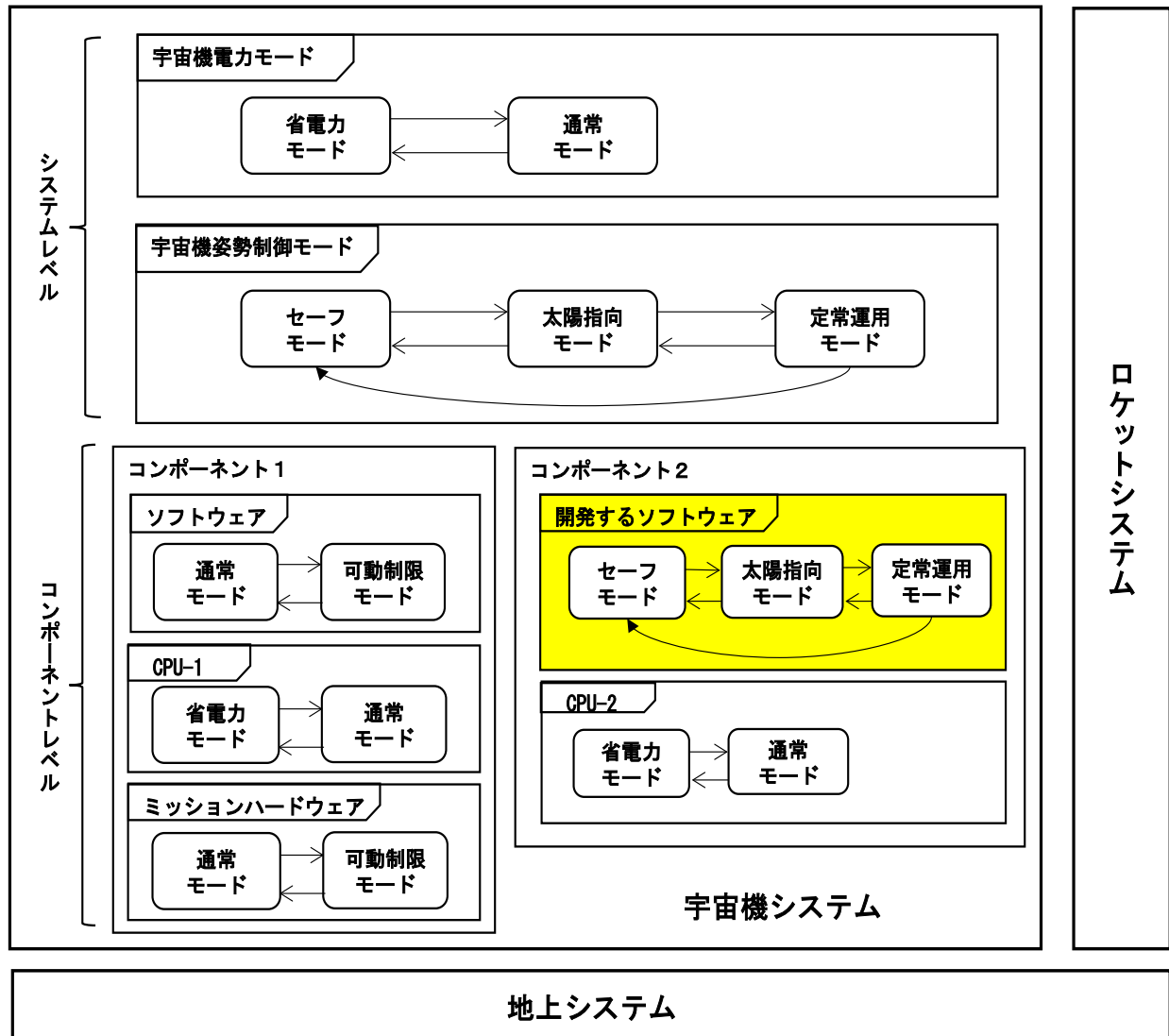
- ・開発するソフトウェアが図のコンポーネント1とインタフェースを取る場合、システムレベルで定義されている省電力／通常モードと、コンポーネント1のソフトウェア・CPU-1・ミッションハードウェアとの関係（省電力モードの場合にミッションハードウェアの電源をオフするなど）を識別し、開発するソフトウェアの状態遷移との関係を明らかにする。
- ・システムレベルで定義されている省電力／通常モード、開発するソフトウェアが動作するCPU-2の状態遷移およびインタフェース先のCPU-1の状態遷移が連動するような場合には、それぞれのCPUの状態遷移での動作によって、開発するソフトウェアの機能・性能に影響がないかなども確認する。

(c) データインタフェースをもつコンポーネントとの関係

- ・開発するソフトウェアが図のコンポーネント1とインタフェースをもつ場合、システムレベルで定義されている姿勢制御モードと、そのコンポーネントのソフトウェア・CPU・ミッションハードウェアとの関係（太陽指向モードの場合はミッションハードウェアは動作するのかなど）を識別し、開発するソフトウェアの状態遷移との関係を明らかにする。

(d) データインタフェースをもつシステムとの関係

- ・ロケットシステムとデータインタフェースをもつ場合には、地上システムの状態遷移との関係を明らかにする。
- ・地上システムとデータインタフェースをもつ場合には、地上システムの状態遷移との関係を明らかにする。

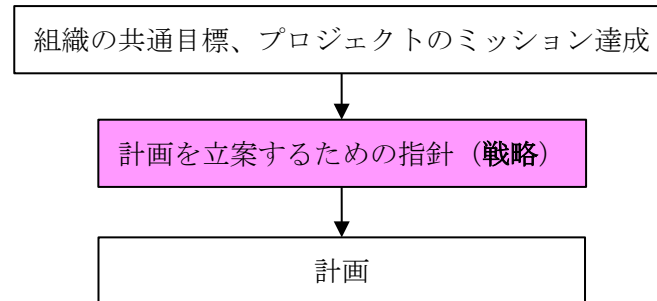


(注：システム構成・コンポーネント構成および状態遷移は説明用であり、実際とは異なる。)

図-付録 7 開発するソフトウェアと関連システムの状態遷移の例(宇宙機)

付録 8 戦略の位置付けの明確化

戦略とは、組織の共通目標（品質、生産性、環境など）に沿って、計画を立案するために定義する、指針のことである。



戦略は、本標準に記載されている事項に限らず、組織としてソフトウェア開発を行うために必要なものを追加で定義することを想定している。

計画には、戦略以外からも導きだされたものを含めて、対象のプロジェクトで実施する内容をすべて記載すること。戦略に対しての計画の網羅性は必ずしも要求していない。

付録 9 「試験計画性」および「試験可能性」の補足説明

本標準では、ソフトウェア要求分析プロセスで「試験計画性」を評価し、ソフトウェア設計プロセスでは「試験可能性」を評価しているが、これらの2つの用語の違いについて、目的の観点から以下に補足する。

◆「試験計画性」

ソフトウェア要求仕様およびインタフェース仕様の仕様記述が、試験可能かどうか確認しながら、その仕様記述を洗練することが目的である。

それぞれの仕様項目が試験可能な仕様記述になっているか、たとえば、否定表現、あいまい表現で要求が書かれると実装・試験ができないため、肯定表現に見直すなどの趣旨を含んでいる。

◆「試験可能性」

試験そのものの成立性を確認し、その結果、設計仕様を洗練し、環境および試験ケースを具体化し、試験精度を向上することが目的である。

付録 10 「ソフトウェア結合」および「ソフトウェア統合」の補足説明

「ソフトウェア結合」は、単体試験およびコードを組合せた試験のためにコードを結合していくことである。

「ソフトウェア統合」は、目標プラットフォームへのインストールを行う前の、ソフトウェア統合試験のためにコードを結合して、ソフトウェアまたはオブジェクト(実行形式)コードを生成することである。

参考文献(6) 共通フレーム 2013 の「ソフトウェア結合」は、本標準の「ソフトウェア結合」および「ソフトウェア統合」の両方に該当する。

付録 11 ソフトウェアクリティカリティクラス (SW CC) の設定

ソフトウェアクリティカリティクラス (SW CC) は、ソフトウェアの機能の特性 (安全や信頼性) などに応じて割り当てられるクラスであり、A、B、C、D の 4 段階がある。割り当てられた SW CC に応じて、適用されるソフトウェア開発標準の要求事項が変わる。

SW CC の割当ては、ソフトウェアの機能の特性 (ステップ 1)、影響防止機能の有無など (ステップ 2) による。

ステップ 1、ステップ 2 の 2 ステップがあるが、どちらのステップで終わらせても良い。

また、SW CC の割当ては、ソフトウェア開発時に割り当てられた後、設計が進むことによって変更することが可能である。ただし、変更した場合には、関係者の合意を得ること。

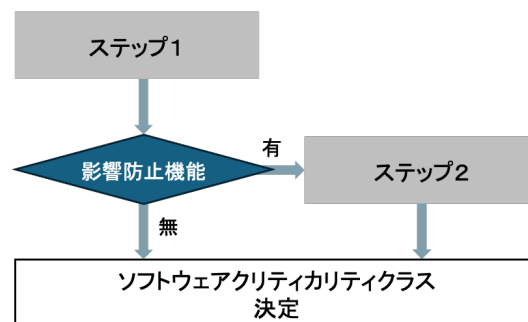


図-付録 11 SW CC を設定する流れ

1 ステップ 1：ソフトウェアの機能の特性による SW CC の設定

対象プロジェクトで、一律に SW CC を定めてしまうと、たとえば実行形式ファイルの単位では、安全性や信頼性に大きな影響を与えないソフトウェアに対しても、過剰な SW CC による開発が必要となってしまう場合がある。そのため、個々のソフトウェアの機能の特性によって、同じプロジェクト内の個々のソフトウェアに、異なる SW CC を設定することを可能としている。

ステップ 1 では、表-付録 11-1 を用いて評価する。個々のソフトウェアを SW CC の A から D への順番で評価していくが、その際、安全の観点、信頼性の観点のいずれか一つでも該当した時点で SW CC を決定する。なお、信頼性上重要な影響を与える可能性のあるソフトウェアは、参考文書 (11) JMR-004 及び参考文書 (12) JERG-0-063 に基づき実施される FMEA 解析の中で CIL (クリティカル品目リスト) として識別される。また、安全の観点では参考文書 (13) JMR-001 の考え方を参考にしている。

個々のソフトウェアが、実行されなかった場合、正しく実行されなかった場合、異常な動作の場合などにおいて、どのような結果をもたらす可能性があるかを FMEA などを用いて安全の観点、信頼性の観点への影響を分析し、ソフトウェアクリティカリティクラスを決定する。

表-付録 11-1 ソフトウェアの機能の特性による SW CC

SW CC	安全の観点	信頼性の観点
A	・死亡・重度の人的被害^{*1} ・環境への永久的な影響 ・公共の財またはサービス、第三者財産の復旧または金銭的な補償が困難で、公共の財もしくはサービス、または、所有者の生活基盤等を著しく損なうと考えられる影響 ・対象プロジェクトの以外のシステムや射場設備など、代替の効きにくい	対象プロジェクトのシステムの機能若しくはミッションの全ての喪失

SW CC	安全の観点	信頼性の観点
	施設・設備等の喪失	
B	・軽度の人的被害*2 ・環境への一時的ではあるが著しい影響 ・公共の財またはサービス、第三者財産の復旧または金銭的な補償は可能であるものの、一時的に公共の財もしくはサービス、または、所有者の生活基盤等を損なうと考えられる影響 ・対象プロジェクトの以外のシステムや射場設備など、代替の効きにくい施設・設備等の重大な損傷	対象プロジェクトのシステムの機能若しくはミッションの重要な機能(エクストラを除くサクセスクライテリアや顧客満足に影響する機能のこと)の一部喪失
C	・軽微な人的被害*3 ・環境への影響が小さく、資金を投入することでリカバリできるもの ・公共の財またはサービス、第三者財産の軽微な損害金銭的な補償のみで対応可能な範囲 ・対象プロジェクトの以外のシステム等の軽度の損害	対象プロジェクトのシステムの機能若しくはミッションの一部喪失又は復旧不能な性能低下
D	上記をもたらさない程度のも	影響はほとんどなし

*1：重度の人的被害：障害の残る被害または消防機関が傷病者の状況を確認する基準で重症および重篤(生命の危険の可能性のあるもの。)

*2：軽度の人的被害：休業災害及び不休災害(いずれも完治可能な被害)(JMR-001に同じ)

休業災害：負傷または疾病のために被災日の翌日から休業せざるを得ないような労働災害。

不休災害：業務遂行中に業務に起因して受けた負傷又は疾病によって、医療機関(事業所内の診療所等を含む)で医師の手当てを受け、被災日の翌日以降1日も休業しなかった労働災害(休業が1日未満のものを含む)。

*3：軽微な人的被害：手当後、直ちに元の作業に戻れる軽傷の被害(JMR-001に同じ)

2 ステップ2：影響防止機能による SW CC の変更

ステップ2では、影響防止機能を取っている場合、ステップ1で割り当てられた SW CC から1クラスだけ下げた SW CC に変更することが可能である。

1クラスだけとしているのは、更に下位のクラスに繰り返し下げた場合、SW CC の低いクラスによる影響防止機能では、安全性、信頼性を確保するのは難しいと考えられるためである。

表-付録 11-2 SW CC を変更することが可能な条件

ステップ1時の SW CC	SW CC を変更することが可能な条件
A	以下のいずれかの影響防止機能が取られる場合、SW CC を B とすることが可能 - ハードウェア(例、異常データの出力停止機能の実装、異常時のインターロック機能など) - SW CC A のソフトウェア(例、ソフトウェア動作を監視して異常があれば当該ソフトウェアを停止させる機能、インタフェースを監視して逸脱したデータを送受信させない機能、など) - 運用手順(例、バッファオーバーフローになる前の定期的な再起動など)
B	以下のいずれかの影響防止機能が取られる場合、SW CC を C とすることが可能 - ハードウェア - SW CC B のソフトウェア - 運用手順
C	以下のいずれかの影響防止機能が取られる場合、SW CC を D とすることが可能 - ハードウェア - SW CC C のソフトウェア - 運用手順
D	-

主機能と影響防止機能において、共通原因による問題(故障)がないこと、無干渉であること(一方の故障が他方の故障に影響しないこと)や、影響防止機能が障害発生までに十分間に合うことなどを示す必要がある。

付録 12 本標準の各要求項目の適用基準

プロジェクトは、プロジェクトの特性に合わせて、4.1 に従い、本標準を適用調整（テーラリング）することが可能である。

表-付録 12 に、本標準の要求を、設定された SW CC に対応してテーラリングする場合のマッピングマトリックスを示す。なお、表中の SW CC 欄の、◎はその下位の要求も含めてすべて必須であることを表し、○はその下位の要求も含めてすべて調整可能であることを表し、空欄は下位の要求に◎または○の記載があることを示している。下記表中の◎、○の区分は、以下のとおりである。

◎適用しない場合正当な理由が必要なもの

○可能な限り適用を推奨するもの（不適用も許容）

表-付録 12 要求マッピングマトリックス

JERG-0-049 の要求内容	SW CC			
	A	B	C	D
4.1 テーラリング	◎	◎	◎	◎
5 主ライフサイクルプロセス				
5.3 開発プロセス				
5.3.1 プロセス開始の準備 ソフトウェア開発の開始時に、以下の要求を満たすアクティビティを実施すること。 (1) 開発戦略を定義すること。戦略には以下を含めること。 (a) 開発方針と規約 (ア) 適切な安全性、セキュリティ、プライバシー、環境活動に対する方針	◎ ◎ ◎ ◎	◎ ◎ ◎ セキュリティおよびプライバシーは○、安全性および環境活動は◎	◎ ◎ ◎ セキュリティおよびプライバシーは○、安全性および環境活動は◎	◎ ◎ ◎ ○
(イ) プログラミングおよびコーディング規約	◎	◎	◎	◎
(ウ) 単体試験方針	◎	◎	◎	◎
(b) 再利用ソフトウェアを使用する場合、コンピュータシステムへの適合性を確認する方法および入手経路の安全性	◎	◎	◎	◎
(c) ソフトウェア結合、ピアレビュー、ウォークスルー・レビューの実施	◎	◎	◎	◎
(d) 手作業で変更管理を行う場合、ソフトウェア製作中の構成管理	◎	◎	◎	◎
(e) コンピュータシステムの廃棄に伴う、関連データとソフトウェア移行作業の優先順位	◎	◎	◎	◎
(f) ナレッジ資産	◎	◎	◎	◎
(2) 開発戦略に基づいて、以下の情報を含むソフトウェア開発計画を立案すること。	◎	◎	◎	◎
(3) ソフトウェア開発計画を文書化し、合意を得ること。	◎	◎	◎	◎
※：ソフトウェアを組み込んだコンピュータシステムとしてのアクティビティなども含む。	◎	◎	◎	◎
5.3.1.1 アウトプット	◎	◎	◎	◎
5.3.2 全開発プロセス適用事項	◎	◎	◎	◎
5.3.3 コンピュータシステム要求分析				
5.3.3.1 アクティビティ コンピュータシステム要求分析に対し、以下の要求を満たすアクティビティを実施すること。	◎	◎	◎	◎
(1) 要求抽出	◎	◎	◎	◎
(2) 要求仕様作成	◎	◎	◎	◎
(a) 運用シナリオに基づき、実現性・整合性を確認し、コンピュータシステムに対する要求仕様を定義すること。	◎	◎	◎	◎
(b) 要求仕様には、コンピュータシステムで取り扱う、データおよびデータベースに対する仕様を含めること。	◎	◎	◎	◎
(c) リスク、コンピュータシステムの重大性、重要な品質特性に関する仕様を含めること。	◎	◎	◎	◎
(d) インタフェース要求を分析し、要求仕様にする。また、インタフェース要求について、その相手方と相互に内容の解釈を含めて合意を得ること。	◎	◎	◎	◎
(e) コンピュータシステムに対する要求仕様の根拠を明確化し、コンピュータシステムへの要求など上位要求とのトレーサビリティを評価し、維持すること。	◎	◎	◎	○
(f) 要求仕様の各要求項目について、要求根拠、検討手法などを明らかにするとともに、実現可能性を評価すること。	◎	◎	◎	◎
(g) COTS または再利用ソフトウェアを利用する場合には、要求仕様との整合性を解析すること。	◎	◎	◎	◎
(h) 適切な利害関係者に要求分析内容をフィードバックするとともに、作成した要求仕様は利害関係者とレビューを実施し合意を得ること。	◎	◎	◎	◎
(i) 要求仕様に含まれる問題点、不備、不整合を識別し、対策を講じること。	◎	◎	◎	◎
(j) 要求仕様の各要求項目について、検証可能性を評価すること。	◎	◎	◎	◎

JERG-0-049 の要求内容	SW CC			
	A	B	C	D
5.3.3.2 インプット				
5.3.3.3 アウトプット				
(1) 運用シナリオ	◎	◎	◎	◎
(2) コンピュータシステムに対する要求仕様	◎	◎	◎	◎
(3) インタフェース仕様	◎	◎	◎	◎
(4) コンピュータシステムに対する要求仕様とコンピュータシステムに対する要求とのトレーサビリティの評価結果	◎	◎	◎	○
(5) コンピュータシステムに対する要求仕様根拠および実現可能性評価結果	◎	◎	◎	◎
(6) COTS、再利用ソフトウェア適合性評価結果	◎	◎	◎	◎
(7) コンピュータシステムに対する要求仕様検証可能性評価結果	◎	◎	◎	◎
5.3.4 コンピュータシステム方式設計				
5.3.4.1 アクティビティ				
コンピュータシステム方式設計に対し、以下の要求を満たすアクティビティを実施すること。	◎	◎	◎	◎
(1) コンピュータシステムに対する要求仕様および運用シナリオに基づき、コンピュータシステム方式を設計し、構成する品目およびその種別(ハードウェア、ファームウェア、ソフトウェア、運用)を明確にすること。	◎	◎	◎	◎
(2) コンピュータシステムに対する要求仕様をコンピュータシステムの各構成項目に要求として割り当てること。	◎	◎	◎	◎
(3) 上記の設計結果をまとめて、コンピュータシステム方式設計仕様を作成すること。	◎	◎	◎	◎
(4) ソフトウェア品目に割り当てられた要求の実現可能性を評価すること。	◎	◎	◎	◎
(5) コンピュータシステム方式設計仕様に対する設計根拠と前提条件(運用前提など)を明らかにし、適切な評価を実施すること。	◎	◎	◎	◎
(6) コンピュータシステム方式設計仕様に対し、コンピュータシステムに対する要求仕様とのトレーサビリティを評価すること。	◎	◎	◎	○
(7) ソフトウェアに關係するインタフェース要求を抽出すること。	◎	◎	◎	◎
(8) コンピュータシステムに対する要求仕様および運用シナリオに基づき、コンピュータシステム方式設計に対する評価基準を定め、それに基づき、コンピュータシステム方式設計結果を評価すること。また、コンピュータシステム方式設計の選択根拠を記録すること。	◎	◎	◎	◎
5.3.4.2 インプット				
5.3.4.3 アウトプット				
(1) コンピュータシステム方式設計仕様	◎	◎	◎	◎
(2) ソフトウェアへの要求(分析後の運用シナリオを含む)	◎	◎	◎	◎
(3) インタフェース要求	◎	◎	◎	◎
(4) コンピュータシステム方式設計仕様とコンピュータシステムに対する要求仕様とのトレーサビリティの評価結果	◎	◎	◎	○
(5) コンピュータシステム方式設計結果の評価結果、およびコンピュータシステム方式設計の選択根拠	◎	◎	◎	◎
5.3.5 ソフトウェア要求分析				
5.3.5.1 アクティビティ				
ソフトウェア要求分析に対し、以下のアクティビティを実施すること。	◎	◎	◎	◎
(1) コンピュータシステム方式設計仕様、インタフェース要求、ソフトウェアへの要求(機能外要求を含む)を分析し、ソフトウェア要求仕様を作成すること。	◎	◎	◎	◎
(2) 求められるソフトウェアの状態遷移(動作モードを含む)を識別すること。	◎	◎	◎	◎
(3) 開発するソフトウェアの関連システムにおいて別途定義されている状態遷移がある場合は、それとの関係を明確化すること(付録 7 参照)。	◎	◎	◎	◎
(4) ソフトウェア要求仕様の各要求項目には、個別に識別子を付与すること。	◎	◎	◎	◎
(5) ソフトウェア要求仕様には、ソフトウェアで取り扱う、データおよびデータベースに対する仕様を含めること。	◎	◎	◎	◎
(6) ソフトウェア要求仕様には、ソフトウェアに実装する、異常検知および、処理機能に関する仕様を含めること。	◎	◎	◎	◎
(7) ソフトウェア要求仕様には、リスク、ソフトウェアの重大性、重要な品質特性に関する仕様を含めること。	◎	◎	◎	◎
(8) ソフトウェア要求仕様には、ユーザインタフェース(有する場合)、ユーザへ提供する情報、ユーザトレーニングに対する仕様を含めること。	◎	◎	◎	◎
(9) ソフトウェア要求仕様には、運用中のシステムに対してソフトウェアを移行する場合は、移行条件を満たす仕様を含めること。	◎	◎	◎	◎
(10) インタフェース要求を分析し、仕様化すること。また、インタフェース仕様について、その相手方と相互に内容の解釈を含めて合意を得ること。	◎	◎	◎	◎
(11) ソフトウェア要求仕様に関して、コンピュータシステム方式設計仕様・インタフェース要求とのトレーサビリティおよび整合性を解析して、文書化し、維持すること。	◎	◎	◎	○
(12) ソフトウェア要求仕様の各要求項目について、要求根拠を明らかにするとともに、実現可能性を評価すること。	◎	◎	◎	◎
(13) COTS または再利用ソフトウェアを利用する場合には、ソフトウェア要求仕様との整合性およびコンピュータシステム方式設計仕様との適合性を解析すること。	◎	◎	◎	◎
(14) ソフトウェア要求仕様の前提となっている運用前提および制約事項を抽出すること。	◎	◎	◎	◎

JERG-0-049 の要求内容	SW CC			
	A	B	C	D
(15) 適切な利害関係者に要求分析内容をフィードバックするとともに、作成したソフトウェア要求仕様は利害関係者とレビューを実施し合意を得ること。	◎	◎	◎	◎
(16) ソフトウェア要求仕様に含まれる問題点、不備、不整合を識別し、対策を講じること。	◎	◎	◎	◎
(17) ソフトウェア要求仕様およびインタフェース仕様の各要求項目について、検証可能性を評価するとともに、検証方法を含むソフトウェア検証計画(妥当性確認計画を含む)を立案すること。	◎	◎	◎	◎
(18) ソフトウェア検証計画に対しソフトウェア機能・性能・インタフェースの視点で検証網羅性を評価するとともに、ソフトウェア要求仕様およびインタフェース仕様に対する試験計画性を評価すること。	◎	◎	◎	◎
(19) ソフトウェア検証計画において、試験環境と実ハードウェアの挙動の差異の影響を受ける試験を実施する場合、あるいは、試験を実施せずレビュー／解析などでの確認を実施する場合は、その確認対象の識別と確認方法が適切であるかの評価も含めること。	◎	◎	◎	◎
5.3.5.2 計測活動	◎	◎	○	○
5.3.5.3 インプット				
5.3.5.4 アウトプット				
(1) ソフトウェア要求仕様	◎	◎	◎	◎
(2) インタフェース仕様	◎	◎	◎	◎
(3) ソフトウェア要求仕様トレーサビリティおよび整合性評価結果	◎	◎	◎	○
(4) ソフトウェア要求仕様根拠および実現可能性評価結果	◎	◎	◎	◎
(5) COTS、再利用ソフトウェア適合性評価結果	◎	◎	◎	◎
(6) 運用前提および制約事項	◎	◎	◎	◎
(7) ソフトウェア検証計画(妥当性確認計画を含む)	◎	◎	◎	◎
(8) ソフトウェア検証計画の検証網羅性および試験計画性評価結果	◎	◎	◎	◎
(9) ソフトウェア要求安定度(成熟度)評価結果	◎	◎	○	○
(10) ソフトウェア要求分析計測活動結果	◎	◎	○	○
5.3.5.5 審査	◎	◎	◎	○
5.3.6 ソフトウェア設計				
5.3.6.1 アクティビティ				
ソフトウェア設計に対し、以下の事項を満たすアクティビティを実施すること。	◎	◎	◎	◎
【基本設計】	◎	◎	◎	◎
(1) 設計指針(ソフトウェアアーキテクチャなど)および考慮すべき設計特性を選定し、優先度を考慮した設計を行うこと。	◎	◎	◎	◎
(2) ソフトウェア要求仕様に基づき、機能分割、モジュール分割を行うとともに、各機能を構成するモジュール間の関係、モジュール間およびモジュールの構造を明確にし、ソフトウェア(基本)設計を行うこと。	◎	◎	◎	◎
(3) ソフトウェア(基本)設計には、ソフトウェア要求に定義された機能外要求(処理時間要求、メモリのリソース要求など)に対する設計、配分を含めること。	◎	◎	◎	◎
(4) インタフェース仕様について、境界と相互作用に配慮して、機能分割、モジュール分割を踏まえて、詳細化を行うこと。また、インタフェース仕様について、その相手方と相互に内容の解釈を含めて合意を得ること。	◎	◎	◎	◎
【詳細設計】	◎	◎	◎	◎
(5) 機能分割、モジュール分割に従い、各モジュールの設計を実施し、ソフトウェア(詳細)設計を行うこと。	◎	◎	◎	◎
(6) インタフェース仕様について、境界と相互作用に配慮して、モジュールの設計に基づいて詳細化を行うこと。また、インタフェース仕様について、その相手方と相互に内容の解釈を含めて合意を得ること。	◎	◎	◎	◎
【基本、詳細設計共通】	◎	◎	◎	◎
(7) ソフトウェア設計結果に基づき、ソフトウェア(基本、詳細)設計仕様を作成すること。	◎	◎	◎	◎
(8) 必要な設計技法や組織において維持している過去の知見を識別し、準備または取得すること。	◎	◎	◎	◎
(9) ソフトウェア設計とソフトウェア要求仕様・インタフェース仕様・必要な関連文書とのトレーサビリティおよび整合性を解析し、文書化し、維持すること。	◎	◎	◎	○
(10) ソフトウェア設計の前提となっている運用前提および制約事項を明らかにすること。	◎	◎	◎	◎
(11) 各ソフトウェア設計項目について、必要に応じて設計根拠を明らかにするとともに、実現可能性、試験可能性(試験ケースを含む)を評価すること。(付録 9 参照)	◎	◎	◎	◎
(12) COTS または再利用ソフトウェアを利用する場合には、ソフトウェア設計との適合性を解析すること。	◎	◎	◎	◎
(13) ソフトウェア検証計画に従って、ソフトウェア試験計画およびソフトウェア試験仕様を立案すること。	◎	◎	◎	◎
(14) 運用前提および制約事項が新たに追加された場合あるいは明らかになった場合には、運用前提および制約事項を更新すること。	◎	◎	◎	◎
5.3.6.2 計測活動	◎	◎	○	○
5.3.6.3 インプット				
5.3.6.4 アウトプット				

JERG-0-049 の要求内容	SW CC			
	A	B	C	D
(1) ソフトウェア(基本、詳細)設計仕様	◎	◎	◎	◎
(2) インタフェース仕様(更新)	◎	◎	◎	◎
(3) ソフトウェア設計トレーサビリティおよび整合性評価結果	◎	◎	◎	○
(4) ソフトウェア設計根拠および実現可能性評価結果	◎	◎	◎	◎
(5) COTS、再利用ソフトウェア適合性評価結果(更新)	◎	◎	◎	◎
(6) 運用前提および制約事項(更新)	◎	◎	◎	◎
(7) ソフトウェア試験計画	◎	◎	◎	◎
(8) ソフトウェア試験仕様	◎	◎	◎	◎
(9) ソフトウェア設計計測活動結果	◎	◎	○	○
5.3.6.5 審査	◎	◎	◎	○
5.3.8 ソフトウェア製作				
5.3.8.1 アクティビティ				
ソフトウェア製作において、以下の事項を満たすアクティビティを実施すること。	◎	◎	◎	◎
(1) 制約、ソフトウェア設計仕様、およびインタフェース仕様に基づきソースコードを作成し、ソースコードレビューを実施すること。	◎	◎	◎	◎
(2) エラー処理の明確な実装指針を考慮すること。	◎	◎	◎	◎
(3) ソースコードはコーディング規約に準拠したものとすること。	◎	◎	◎	◎
(4) ソースコードチェックツールなどのツールで静的解析を実施し、ソースコードの品質評価を実施すること。	◎	◎	◎	◎
なお、表 5.3.8-1 のモジュールの循環的複雑度(Cyclomatic Complexity)の基準を満たすこと。	表 5.3.8-1 参照			
(5) ソフトウェア検証計画およびソフトウェア試験計画に従い単体試験仕様を作成すること。	◎	◎	◎	◎
(6) 単体試験仕様に従い単体試験を実施し、実施結果を単体試験の合否判定可能な形式で記録すること。	◎	◎	◎	◎
(7) 単体試験は、表 5.3.8-2 のソースコードに対する試験網羅率の基準を満たすよう実施すること。ただし、試験網羅率の基準を達成できない場合には、試験対象外のコードに分析、検査、または設計のレビューを適用すること。	表 5.3.8-2 参照			
(8) ソースコードとソフトウェア設計仕様のトレーサビリティを解析し、その結果を記録すること。なお、ソースコード上で使用する名称(変数名、関数名など)と設計仕様との対応付けを確認すること。	◎	◎	◎	○
(9) 運用前提および制約事項が新たに追加された場合あるいは明らかになった場合には、運用前提および制約事項を更新すること。	◎	◎	◎	◎
5.3.8.2 計測活動	◎	◎	○	○
5.3.8.3 インプット				
5.3.8.4 アウトプット				
(1) ソースコード	◎	◎	◎	◎
(2) 運用前提および制約事項(更新)	◎	◎	◎	◎
(3) 単体試験仕様	◎	◎	◎	◎
(4) 単体試験記録	◎	◎	◎	◎
(5) トレーサビリティ解析記録	◎	◎	◎	○
(6) ソフトウェア製作計測活動結果	◎	◎	○	○
5.3.8.5 審査	◎	◎	○	○
5.3.10 ソフトウェア統合				
5.3.10.1 アクティビティ	◎	◎	◎	◎
5.3.10.2 計測活動	◎	◎	○	○
5.3.10.3 インプット				
5.3.10.4 アウトプット				
(1) ソースコード(統合後)	◎	◎	◎	◎
(2) ソフトウェア(統合後)	◎	◎	◎	◎
(3) 統合に対する制約条件の識別結果	◎	◎	◎	◎
(4) ソフトウェア統合計測活動結果	◎	◎	○	○
5.3.11 ソフトウェア統合試験				
5.3.11.1 アクティビティ	◎	◎	◎	◎
5.3.11.2 計測活動	◎	◎	○	○
5.3.11.3 インプット				
5.3.11.4 アウトプット				
(1) ソフトウェア(統合)試験手順	◎	◎	◎	◎
(2) ソフトウェア(統合)試験記録(合否判定結果を含む)	◎	◎	◎	◎
(3) 運用前提および制約事項(更新)	◎	◎	◎	◎
(4) ソースコード(試験後)	◎	◎	◎	◎
(5) ソフトウェア(試験後)	◎	◎	◎	◎
(6) ソフトウェア試験仕様(更新)	◎	◎	◎	◎
(7) ソフトウェア(統合)試験計測活動結果	◎	◎	○	○
5.3.11.5 審査	◎	◎	◎	○
5.3.12 目標プラットフォームへのインストール(組み込み)				
5.3.12.1 アクティビティ	◎	◎	◎	◎

JERG-0-049 の要求内容	SW CC			
	A	B	C	D
5.3.12.2 インプット				
5.3.12.3 アウトプット	◎	◎	◎	◎
5.3.13 コンピュータシステム統合およびコンピュータシステム総合試験				
5.3.13.1 アクティビティ	◎	◎	◎	◎
5.3.13.2 計測活動	◎	◎	○	○
5.3.13.3 インプット				
5.3.13.4 アウトプット				
(1) コンピュータシステム総合試験仕様	◎	◎	◎	◎
(2) コンピュータシステム総合試験手順	◎	◎	◎	◎
(3) コンピュータシステム総合試験記録(合否判定結果を含む)	◎	◎	◎	◎
(4) 運用前提および制約事項(更新)	◎	◎	◎	◎
(5) 統合に対する制約条件の識別結果(更新後)	◎	◎	◎	◎
(6) コンピュータシステム統合および総合試験計測活動結果	◎	◎	○	○
5.3.14 ソフトウェア製品の供給と導入				
5.3.14.1 アクティビティ	◎	◎	◎	◎
5.3.14.2 インプット				
5.3.14.3 アウトプット	◎	◎	◎	◎
5.3.15 ソフトウェア製品の受け入れ				
5.3.15.1 アクティビティ	◎	◎	◎	◎
5.3.15.2 インプット				
5.3.15.3 アウトプット	◎	◎	◎	◎
5.4 運用プロセス				
5.4.1 プロセス開始の準備				
5.4.1.1 運用戦略の定義				
(1) 運用戦略を定義すること。なお、戦略には以下を考慮すること。	◎	◎	◎	◎
(a) サービス導入後、定期的に運用されサービス廃止に至るまでに期待される各基準(容量、稼働率、応答時間、安全性 等)	◎	◎	◎	安全性以外◎ 安全性○
(b) 既存のサービスを維持するための修正を考慮したソフトウェアまたはコンピュータシステムのリリース基準およびスケジュール	◎	◎	◎	◎
(c) 各運用モード(通常運用/準備段階/運用試験実施/想定される災害・障害発生時)を実装するための手法	◎	◎	◎	◎
(d) 性能レベルを評価するための運用にかかわる尺度	◎	◎	◎	◎
5.4.1.2 運用計画の立案	◎	◎	◎	◎
5.4.1.3 運用時の問題管理手続きの確立	◎	◎	◎	◎
5.4.1.4 ソフトウェアを含むコンピュータシステムの運用およびユーザサポートに係る作業手順の確立	◎	◎	◎	◎
5.4.2 運用試験	◎	◎	◎	◎
5.4.3 ソフトウェアを含むコンピュータシステムの運用	◎	◎	◎	◎
5.4.4 運用結果の管理	◎	◎	◎	◎
5.4.5 顧客およびユーザサポート	◎	◎	◎	◎
5.5 保守プロセス				
5.5.1 プロセス開始の準備				
5.5.1.1 保守戦略の定義	◎	◎	◎	◎
5.5.1.2 保守計画の立案	◎	◎	◎	◎
5.5.2 問題把握および修正分析	◎	◎	◎	◎
5.5.3 修正の実施	◎	◎	◎	◎
5.5.4 ソフトウェアの書き換え	◎	◎	◎	◎
5.5.5 ロジスティクス支援の実施	◎	◎	◎	◎
5.5.6 保守およびロジスティクス結果の管理	◎	◎	◎	◎
5.5.7 移行				
5.5.7.1 移行戦略の定義	◎	◎	◎	◎
5.5.7.2 移行計画の立案と実行	◎	◎	◎	◎
5.5.7.3 ユーザへの通知	◎	◎	◎	◎
5.5.7.4 移行記録の管理	◎	◎	◎	◎
5.5.7.5 旧環境の保管	○	○	○	○
5.5.8 ソフトウェア廃棄				
ソフトウェアの廃棄には以下の観点を考慮すること。	◎	◎	◎	◎
(1) 運用および保守を行っている組織がサポートを停止するための廃棄戦略を定義すること。なお、戦略には以下を考慮すること。	◎	◎	◎	◎
(a) 永久廃棄となるシステム機能および提供サービスの特定	◎	◎	◎	◎
(b) ソフトウェアの所有と保持責任、またはソフトウェアのデータと知的所有権の破棄	◎	◎	◎	◎
(c) 関係者、社会、環境への影響を避けるため、社会的および物理的に製品を受容可能な状態に変化させること	◎	◎	◎	◎

JERG-0-049 の要求内容	SW CC			
	A	B	C	D
(d)ハードウェアおよびデータの長期的な状態を元にした、衛生、安全、セキュリティとプライバシーを考慮した廃棄	◎	セキュリティ およびプライ バシーは○、 安全性は◎	セキュリティ およびプライ バシーは○、 安全性は◎	○
(e)重要な廃棄処理に関する影響をうける関係者への通知	◎	◎	◎	◎
(f) 廃棄のためのスケジュール、行動、責任と資源の識別	◎	◎	◎	◎
(g) 廃棄に向けた、システム/ソフトウェア要求、アーキテクチャ、設計または実装技術での制約条件の特定	◎	◎	◎	◎
(h) 廃棄を支援するのに必要なイネープリングシステムおよびサービスに対する特定と計画	◎	◎	◎	◎
(i) 必要なイネープリングシステムおよびサービスを獲得あるいはアクセスを取得すること。	◎	◎	◎	◎
(j) ソフトウェアまたはデータを保管する場合の設備、保管場所、点検基準と貯蔵期間	◎	◎	◎	◎
(k) 廃棄したソフトウェアの再利用を防止する方法	◎	◎	◎	◎
(2) 廃棄戦略に基づき廃棄計画を立案すること。なお、計画立案には、ユーザを参加させること。	◎	◎	◎	◎
(3) 廃棄計画および実施内容をユーザに通知すること。通知には、次の項目を含めること。	◎	◎	◎	◎
(4) 廃棄計画に基づき以下の活動を実施する。	◎	◎	◎	◎
(5) 予定した廃棄時期がきたとき、関係者全員に通知する。関連するすべての開発文書、ログおよびソースコードは保管することが望ましい。	◎	◎	◎	◎
(6) 廃棄後のソフトウェアあるいはコンピュータシステムに対して、衛生、安全、セキュリティとプライバシー、環境への悪影響を特定し、レビューおよび監査できるように長期に渡ってデータを集めること。	◎	セキュリティ およびプライ バシーは○、 安全性は◎	セキュリティ およびプライ バシーは○、 安全性は◎	○
6 支援ライフサイクルプロセス				
6.1 文書化プロセス	◎	◎	○	○
6.2 構成管理プロセス				
6.2.1 プロセス開始の準備				
6.2.1.1 構成管理戦略の定義	◎	◎	◎	◎
6.2.1.2 構成管理計画の立案	◎	◎	◎	◎
6.2.2 構成識別	◎	◎	◎	◎
6.2.3 構成変更管理	◎	◎	◎	◎
6.2.4 構成変更状況の記録	◎	◎	◎	◎
6.2.5 構成変更状況の評価	◎	◎	◎	◎
6.2.6 リリース管理と出荷 ソフトウェア製品のリリースと出荷を手続きに従って管理すること。特に、安全性またはセキュリティが要求される重大な機能を含んだソースコードと文書は、組織の方針に従って、取扱い、保管し、包装し、出荷すること。	◎	セキュリティ は○、 それ以外は ◎	セキュリティ は○、 それ以外は ◎	セキュリティと 安全性は○、 それ以外は◎
決められたソフトウェア製品のリリースについて、配付を管理すること。	◎	◎	◎	◎
6.2.7 構成監査の実施	◎	◎	◎	◎
6.3 品質保証プロセス				
6.3.1 プロセス開始の準備				
6.3.1.1 組織の独立性の確認、適切な体制の確立	◎	セキュリティ は○、それ以 外は◎	セキュリティ は○、それ以 外は◎	セキュリティは ○、それ以外 は◎
6.3.1.2 品質保証戦略の定義	◎	◎	◎	◎
6.3.1.3 品質保証活動計画の立案	◎	◎	◎	◎
6.3.2 製品およびサービス品質の保証				
(1) 製品およびサービスが計画されたものが不足なく作成されていることを保証すること。	◎	◎	◎	◎
(2) ソフトウェア製品またはコンピュータシステムの開発プロセスの各成果物への検証および妥当性確認にてその成果物が合意された要求仕様を満足していることを保証すること。	◎	◎	◎	◎
(3) 必要に応じて独立検証・妥当性確認(IV&V)を実施し、製品品質を保証すること。	◎	◎	○	○
6.3.3 プロセスの保証	◎	◎	◎	◎
6.3.4 品質システムの保証				
6.3.4.1 教育・訓練				
対象となるソフトウェアまたはコンピュータシステムの開発・保守・運用業務に従事する要員に必要な技術・能力・資格を識別し、計画的な教育・訓練を実施すること。	◎	◎	◎	○
6.3.4.2 購買管理・供給業者管理	◎	◎	◎	◎
6.3.4.3 取得者支給品の管理	◎	◎	◎	◎
6.3.4.4 既存ソフトウェア品目(COTS 品目・ナレッジ資産)の管理	◎	◎	◎	◎
6.3.4.5 取扱い、保管およびラベリング	◎	◎	◎	◎
6.3.5 品質保証記録の管理	◎	◎	◎	◎

JERG-0-049 の要求内容	SW CC			
	A	B	C	D
6.4 検証プロセス				
6.4.1 プロセス開始の準備	◎	◎	◎	◎
6.4.2 検証 このアクティビティは、次のタスクからなる。				
6.4.2.1 プロセスの検証	◎	◎	◎	◎
6.4.2.2 要求の検証 次に示す観点を考慮すること。 (1) 要求に一貫性があり、実現可能かつ検証可能である。 (2) ソフトウェア品目に対して配分されている要求が適切である(ハードウェア品目や運用に対して配分されるべき要求を含んでいない)。 (3) 上位の要求や品目に適用される基準、標準を満たしている。 (4) 安全性・セキュリティなど特に注意を要する要求に関して、上位の要求や品目に適用される基準、標準が満たされていることを、適切な手法により示すことができる。	◎ ◎ ◎ ◎ ◎	◎ ◎ ◎ ◎ セキュリティは○、それ以外は◎	◎ ◎ ◎ ◎ セキュリティは○、それ以外は◎	◎ ◎ ◎ ◎ セキュリティおよび安全性は○、それ以外は◎
6.4.2.3 設計の検証 次に示す観点を考慮すること。 (1) 設計は要求を満たし、要求に対してトレーサビリティが確保されている。 (2) 入出力インタフェース・タイミング・計算機リソース(メモリ容量・処理速度など)・論理設計・処理順序と処理内容(特に初期化・終了・例外など)に関し、適切に設計されている。 (3) 移植容易性・変更容易性・問題解決容易性などの特性について、適切に考慮されている。	◎ ◎ ◎ ◎	◎ ◎ ◎ ◎	◎ ◎ ◎ ◎	◎ ○ ◎ ◎
6.4.2.4 ソースコードの検証 次に示す観点を考慮すること。 (1) ソースコードは設計を満たし、設計に対してトレーサビリティが確保されている。 (2) 入出力インタフェース・タイミング・計算機リソース(メモリ容量・処理速度など)・論理設計・処理順序と処理内容(特に初期化・終了・例外など)に関し、適切に実装されている。 (3) 移植容易性・変更容易性・問題解決容易性などの特性について、適切に実装されている。 (4) 安全性・セキュリティなど特に注意を要するソースコードに関して、要求や品目に適用される基準、標準が満たされていることを、適切な手法により示すことができる。 (5) ソースコードはコーディング規約などの基準を満たしている。	◎ ◎ ◎ ◎ ◎ ◎	◎ ◎ ◎ ◎ セキュリティは○、それ以外は◎ ◎	◎ ◎ ◎ ◎ セキュリティは○、それ以外は◎ ◎	◎ ○ ◎ ◎ セキュリティおよび安全性は○、それ以外は◎ ◎
6.4.2.5 統合の検証	◎	◎	◎	◎
6.4.2.6 文書化の検証	◎	◎	○	○
6.4.3 検証結果の管理	◎	◎	◎	◎
6.5 妥当性確認プロセス	◎	◎	◎	◎
6.6 共同レビュープロセス	◎	◎	◎	○
6.7 アセスメントプロセス	◎	◎	○	○
6.8 問題解決プロセス				
6.8.1 プロセス開始の準備	◎	◎	◎	◎
6.8.2 問題の解決	◎	◎	◎	◎
6.8.3 予防	◎	◎	◎	○
6.8.4 傾向分析	◎	◎	◎	○