

限定なし



宇宙機ソフトウェア開発標準

2025年3月28日 C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	準拠文書	1
2.2	参考文書	1
3	用語、定義および略語	1
3.1	本標準で再定義した用語	1
3.2	本標準で新規に定義した用語	2
3.3	JERG-0-049D で定義された用語	4
4	一般要求事項	10
4.1	テーラリング	11
5	主ライフサイクルプロセス	12
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	12
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	14
5.3.4	コンピュータシステム方式設計	15
5.3.5	ソフトウェア要求分析	16
5.3.6	ソフトウェア設計	18
5.3.7	欠番	20
5.3.8	ソフトウェア製作	20
5.3.9	欠番	22
5.3.10	ソフトウェア統合	22
5.3.11	ソフトウェア統合試験	22
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)	24
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	25
5.3.14	ソフトウェア製品の供給と導入	27
5.3.15	ソフトウェア製品の受け入れ	27
5.4	運用プロセス	27

5.4.1	プロセス開始の準備	27
5.4.2	運用試験	28
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	28
5.4.4	運用結果の管理	28
5.4.5	運用者および利用者への支援	28
5.5	保守プロセス	29
5.5.1	プロセス開始の準備	29
5.5.2	問題把握および修正分析	30
5.5.3	修正の実施	30
5.5.4	ソフトウェアの書換え	30
5.5.5	ロジスティクス支援の実施	30
5.5.6	保守結果の管理	30
5.5.7	移行	31
5.5.8	ソフトウェア廃棄	31
6	支援ライフサイクルプロセス	32
6.1	文書化プロセス	32
6.1.1	プロセス開始の準備	32
6.1.2	作成	32
6.1.3	文書発行	33
6.1.4	文書の維持・改訂・廃棄	33
6.2	構成管理プロセス	33
6.2.1	プロセス開始の準備	33
6.2.2	構成識別	34
6.2.3	構成変更管理(Configuration Control)	35
6.2.4	構成管理記録および状況報告	35
6.2.5	構成評価	35
6.2.6	リリース管理	35
6.2.7	構成監査の実施	36
6.3	品質保証プロセス	36
6.3.1	プロセス開始の準備	36
6.3.2	品質保証活動の実施	37
6.3.3	製品品質の保証	37

6.3.4	プロセスの保証	37
6.3.5	品質システムの保証	37
6.3.6	品質保証記録の管理	38
6.3.7	自動コード生成における管理	38
6.4	検証プロセス	39
6.4.1	プロセス開始の準備	39
6.4.2	検証	40
6.4.3	検証結果の管理	41
6.5	妥当性確認プロセス	41
6.5.1	プロセス開始の準備	42
6.5.2	妥当性確認	42
6.5.3	妥当性確認結果の管理	42
6.6	共同レビュープロセス	43
6.6.1	プロセス開始の準備	43
6.6.2	ソフトウェア開発管理レビュー	43
6.6.3	技術レビュー	44
6.7	アセスメントプロセス	45
6.7.1	プロセス開始の準備	45
6.7.2	アセスメントの実施	45
6.8	問題解決プロセス	45
6.8.1	プロセス開始の準備	45
6.8.2	問題解決	46
6.8.3	傾向分析	47
6.8.4	是正・予防処置	47
	付録	48
付録 1	5.5.3 修正の実施の補足	48
付録 2	本標準の各要求項目の適用基準	49

1 適用範囲

本標準は、宇宙機に搭載されるコンピュータシステム(宇宙機における各サブシステム、コンポーネント、ミッションセンサシステムなど)、ソフトウェアの開発・運用・保守に関する活動を適用対象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。

2 関連文書

2.1 準拠文書

- (1) JERG-0-049D ソフトウェア開発標準

2.2 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

本標準で使用する用語の定義および略語は、基本的に JERG-0-049D に準ずるものとする。本標準で再定義(詳細化)した用語、新規に定義した用語、JERG-0-049D で定義した用語を下記に示す。

3.1 本標準で再定義した用語

用語	説明
インシデント	変則的あるいは予期せぬイベント、イベントの集合、状態又は状況である。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じたすべての時点で起こりうる。
運用	コンピュータシステムを稼動して当該システムの目標を達成するための作業を遂行する行為。システムの起動やシステムの終了、監視、ファイルメンテナンスなどが含まれる。

用語	説明
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 (JERG-0-049D、ISO9000)。 ソフトウェアが要求仕様や設計仕様を満たしていることを証明するアクティビティ全般。検証方法には試験の他、シミュレーションや、ウォークスルー、ソースコードレビューなどの文書レビュー、検査、デモンストレーションなどがある。

3.2 本標準で新規に定義した用語

用語	説明
S00H	衛星運用説明書 (Spacecraft Orbital Operations Handbook:S00H)。S00Hは、衛星の運用において必要となる情報を提供するものである。SOPの作成および追跡管制データベースを設定するための源泉になるとともに、実運用においては、手順書の確認または異常時の対応のような場合、新たな運用計画の立案および運用手順の作成のさいに参照される。
SOP	衛星運用手順書 (Spacecraft Operations Procedure :SOP)。追跡管制システムの機能を十分に活かす目的で、通常は独立した機能単位 (ユニット) ごとに作成し、運用に当たっては、そのユニット単体または複数のユニットを組み合わせて使用する。
運用者	システムを運用する個人または組織。
試験可能性	要求仕様に対して、試験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決定するために試験を実行することが可能であること。
自動コード生成	宇宙機搭載を目的として、自動コード生成元データ (モデルなど) から自動で高級言語相当のソースコードを生成することを指す。ただし、モデル単体試験が可能である範囲に限る。
自動コード生成ハンドブック	自動コード生成するための自動コード生成元データ (モデルなど) の記述方法などを定義した文書。たとえば、公的に提供された文書、またはそれに準じて自組織で定めた文書などとする。
モデリングハンドブック	モデルの記述方法などを定義した文書。たとえば、公的に提供された文書、またはそれに準じて自組織で定めた文書などとする。
モデル	コンピュータシステム/ソフトウェアを抽象化したもので、対象とするコンピュータシステム/ソフトウェアを一定の視点および一定の抽象レベルで特定の記述法を用いて表現したもの。
モデル試験網羅	自動コード生成を実施する場合、モデル単体試験を実施したさいの、モデルに対する網羅度合いを指す。あらかじめ品質評価のために必要な評価基準が設定される必要がある。評価基準の一部は、モデルベースシミュレーションの実施によって充足しても良い。
モデル単体試験	自動コード生成を実施する場合、ソフトウェア設計プロセスなどにおいて、自動コード生成元データ (モデルなど) を用いて、単体試験相当の試験を行う。ここで使用した試験仕様 (テストケース) は、同一の仕様を用いたソフトウェア製作プロセスでの単体試験の等価性評価で利用する。

用語	説明
モデルベース開発	ソフトウェア開発標準の適用対象となる開発（または主ライフサイクル）プロセスの全部または一部をモデルベース技術を用いて実施する開発・検証を行う行為である。作成されたモデルは、成果物の一部になる。
モデルベースシミュレーション	モデルベース開発を実施する場合、ソフトウェア開発プロセスのうち、コンピュータシステム方式設計、ソフトウェア要求分析、およびソフトウェア設計プロセスにおいて、動作可能なモデルなどを用いたシミュレーションにより要求および設計の妥当性を評価すること。
利用者	システムから利益を得る個人または組織。
リリース	文書における正版配付や、システム渡し、試験渡しなど。

3.3 JERG-0-049D で定義された用語

用語	説明
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、客観的指標に基づき、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目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 コンフィギュレーション管理基準)
受け入れ検査および試験	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。
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。
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	<p>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部品目を定義し、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。</p> <p>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。</p>
コンピュータシステム	<p>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。</p> <p>コンピュータシステムに何を含めるかの定義は、コンピュータシステムに含めないものとの境界で定義する形式と、境界の中に含まれる品目で定義する形式がありうる。(なお、境界で定義する形式の場合でも、コンピュータシステムの設計作業として、境界内の品目を決定していく作業を含むことを想定している)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。</p>
サービス	<p>利用者やシステムに対して、機能や働きを提供するもの。</p>
識別子	<p>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。</p> <p>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。</p>
試験計画性	<p>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。JERG-0-049D 付録9を参照のこと。</p>

用語	説明
試験網羅率	<p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p> <ol style="list-style-type: none"> 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	<p>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。</p>
循環的複雑度 (Cyclomatic Complexity)	<p>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。$e-n+2p$</p>
戦略	<p>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。JERG-0-049D 付録8を参照のこと。</p>
相互作用	<p>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。</p>
ソースコード	<p>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。</p>
ソフトウェア	<p>コンピュータシステムの構成目目の一種であり、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)。</p>
ソフトウェアクリティシティクラス	<p>ソフトウェアの機能をシステムの特長(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。JERG-0-049D 付録11を参照のこと。</p>

用語	説明
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。JERG-0-049D 付録 10 を参照のこと。
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
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(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
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（イネープリングシステムまたはサービス）の総称。（JERG-0-049D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。
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、表現に曖昧さがあり漠然とした期待に近いものも含まれる。

用語	説明
要求仕様	<p>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。</p> <p>(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。</p> <p>(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。</p> <p>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。</p>
リスク	<p>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。</p>

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。(図 4.1 および表 4.1 に定めるプロセスを示す)

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関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関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、必要に応じて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。

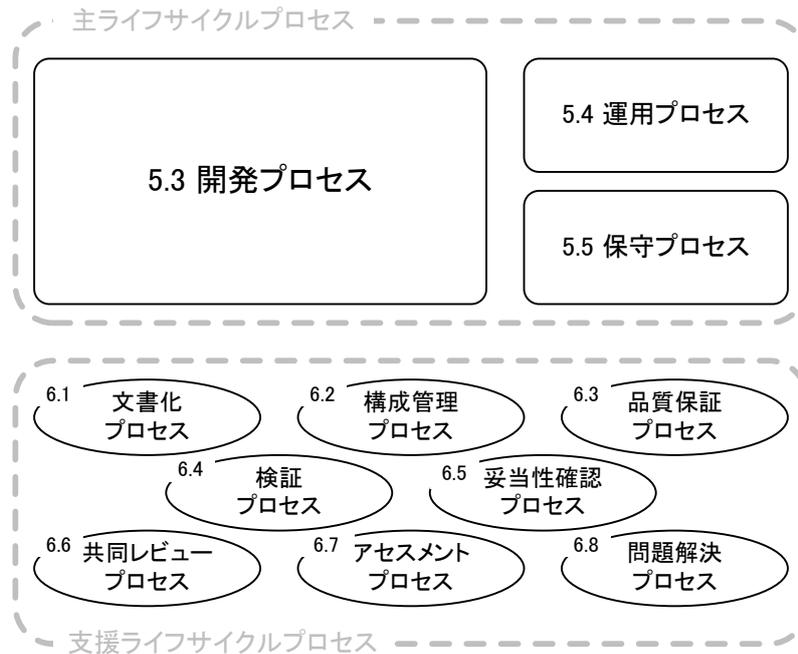


図 4.1 プロセス全体図

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。

なお、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。

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。 本標準における”運用”とは主に、宇宙機運用者、利用者による宇宙機運用の支援作業を示す。
	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 JERG-0-049D 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 2 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(※1)

ソフトウェア製品の受け入れプロセス(※2)

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。

※1： 本標準では、ソフトウェア製品の供給と導入プロセスは規定しない。

※2： 本標準では、ソフトウェア製品の受け入れプロセスは規定しない。

5.3.1 プロセス開始の準備

5.3.1.1 アクティビティ

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。なお、戦略には以下を考慮すること。

(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) ナレッジ資産管理
 - (ア) ナレッジ資産の有効期間における、獲得と維持計画。
 - (イ) ナレッジ資産として収集され、維持される知見、能力などの型の特徴づけ。
 - (ウ) ナレッジ資産を受け入れ、認定し、廃止するための基準
 - (エ) ナレッジ資産の変更を制御するための手順
 - (オ) 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
 - (カ) 保存と検索のメカニズム
- (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(※1)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
 - (e) ソフトウェア開発プロセスとアクティビティの定義(※2)
 - (f) 自動コード生成を実施する場合は、自動コード生成ツールに合わせたアクティビティ定義、範囲(※3)、検証方針(※4)、管理方針、利用ツールおよびその選定に使用した選定基準・選定根拠、自動コード生成ハンドブック
 - (g) ソフトウェア製作に関する次の開発方針と規約
 - (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(エラー処理の明確な実装指針を含む)
 - (ウ) 単体試験方針
 - (エ) 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
 - (オ) 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
 - (h) 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
 - (i) 審査計画
 - (j) 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
 - (k) 文書作成計画(作成部門と作成時期を含む)
 - (l) 適切な作業分割および作業単位ごとの作業計画と達成基準を含む進捗管理の方法(当初開発スケジュールを含む)
 - (m) 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、ソースコード解析ツール、自動コード生成ツールなどのイネープリングシステムまたはサービス)を識別し、利用できるような状態にすること。
 - (n) 使用する COTS およびナレッジ資産(JERG-0-049D 付録 6 参照)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
 - (ウ) 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
 - (o) 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。

※1：宇宙機における各サブシステム、コンポーネント、ミッションセンサシステムなどが該当

※2： ソフトウェアライフサイクルを考慮することが望ましい。また、必要に応じ、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を含めてもよい。

※3： 自動コード生成対象の機能やモジュール

※4： 検証方針として、ソースコードレビュー、自動コード生成元データ（モデルなど）検証方法（モデルベースシミュレーションおよびモデル単体試験）、単体試験、統合試験、モデルとソースコードの等価性評価、およびそれらを組み合わせた総合的な検証戦略を定義する。

5.3.1.2 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) ソフトウェア開発の進捗を把握すること。必要に応じ管理者へ報告すること。
- (3) モデルベース開発の場合には、各プロセスでのモデルベース開発におけるトレーサビリティ評価として、モデルを含む仕様記述およびコードに対して、静的（機能・構造）/動的（振る舞い・性能）なトレーサビリティを評価すること。

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、ソフトウェア開発計画に従い、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（動作モードを含む）と、その状態遷移を識別すること。
 - (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（JERG-0-049D 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識別すること。

- (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連する要求を識別すること。
- (d) 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、整合性を解析して、文書化し、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 またはナレッジ資産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、ナレッジ資産適合性評価結果
- (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、ソフトウェア開発計画に従い、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) モデルベース開発を実施する場合は、モデリングハンドブックに従ってモデルを作成すること。
- (4) 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。
- (5) ソフトウェアに関係するインタフェース要求を抽出すること。
- (6) 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。
- (7) 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)を明らかにすること。
- (8) 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など上位要求とのトレーサビリティを評価すること。

- (9) 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。モデルベース開発を実施する場合は、コンピュータ方式設計仕様レベルのモデルベースシミュレーションにより、設計の妥当性を評価すること。

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
- (3) インタフェース要求(テレメトリ、コマンドに対する要求も含む)
- (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、ソフトウェア開発計画に従い、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) 必要となるソフトウェアの状態(動作モードを含む)と、その状態遷移を識別すること。
- (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(JERG-0-049D 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、ソフトウェアで実装する、異常検知および処理機能に関する仕様を含めること。
- (7) リスク、ソフトウェアの重大性、重要な品質特性に関連する要求を識別すること。
- (8) モデルベース開発を実施する場合は、モデリングハンドブックに従ってモデルを作成すること。
- (9) ソフトウェア要求仕様には、運用中のコンピュータシステムに対してソフトウェアを書き換える場合は、ソフトウェアを書換える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 またはナレッジ資産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) 自動コード生成を実施する場合は、計画立案時に決定した範囲を踏まえて、手動で製作するソースコードと自動コード生成するコードとの間のインタフェースを分析すること。
 - (15) 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運用制約を抽出すること。
 - (16) 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。
 - (17) 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。
 - (18) 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。
 - (19) モデルベース開発を実施する場合は、ソフトウェア要求仕様レベルのモデルベースシミュレーションにより、要求の妥当性を評価すること。
 - (20) ソフトウェア検証計画に対しソフトウェア機能・性能・運用シナリオに関する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。
 - (21) 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計画している場合、あるいは、試験を実施せずレビュー／解析などで確認する項目が含まれている場合、その確認対象の識別と確認方法が適切であるかの評価を、ソフトウェア検証計画に含めること。

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。
- (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
- (3) インタフェース要求
- (4) 運用シナリオ
- (5) COTS またはナレッジ資産適合性評価結果

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、ナレッジ資産適合性評価結果(更新)
- (6) 運用前提および運用制約
- (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

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

実際の開発では、基本設計、詳細設計に分割して実施しても良い。

5.3.6.1 アクティビティ

ソフトウェア設計に対し、ソフトウェア開発計画に従い、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) 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構造、およびモジュール間構造を明確にし、ソフトウェア(基本)設計を行うこと。自動コード生成を実施する場合は、計画立案時に定めた自動コード生成の範囲を確定し、手動で製作するソースコードと自動コード生成するコードとの間のインタフェースの仕様を確定すること。
- (2) 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。
- (3) ソフトウェア(基本)設計には、ソフトウェア要求仕様に定義された機能外要求(処理時間要求、メモリなど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、外部インタフェース境界と相互作用に配慮して、ソフトウェア機能分割、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) ソフトウェア機能分割、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、モジュール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) モデルベース開発を実施する場合は、モデルベースシミュレーションにより、設計の妥当性を評価すること。
- (9) 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。
- (10) モデルベース開発を実施する場合は、モデリングハンドブックに従ってモデルを作成すること。
- (11) 自動コード生成を実施する場合は、自動コード生成ハンドブックに従って自動コード生成元データ(モデルなど)を作成すること。
- (12) 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。
- (13) 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。（JERG-0-49D 付録 9 参照）

- (14) COTS またはナレッジ資産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。
- (15) ソフトウェア設計の前提となっている運用前提および運用制約を明らかにすること。
- (16) 自動コード生成を実施する場合は、モデル単体試験を実施すること。あらかじめ設定したモデル試験網羅の評価基準を満たすように、モデル単体試験を実施すること。
- (17) 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。
 - ソフトウェア試験仕様は、以下の観点を考慮すること。
 - (a) 運用シナリオ
 - (b) インタフェース仕様
 - (c) 想定されるシナリオ上の最大負荷
 - (d) 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
 - (e) 例外や故障などの異常事象
 - (f) COTS/ナレッジ資産のコンピュータシステムに対する適合性
- (18) 運用前提および運用制約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、運用前提および運用制約を更新すること。

5.3.6.2 計測活動

ソフトウェア設計の進捗管理/リスク評価を実施できるように、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) 運用前提および運用制約
- (4) COTS、ナレッジ資産適合性評価結果
- (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、ナレッジ資産適合性評価結果(更新)
- (6) 運用前提および運用制約(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) 進捗管理/リスク評価にかかわる収集データの評価結果

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、ソフトウェア開発計画に従い、以下の事項を満たすアクティビティを実施すること。

(1) 製作準備

- (a) 単体試験では、ソフトウェア設計仕様に対して試験が合格となる基準を設けること。
- (b) 単体試験は、表 5.3.8-1 のソースコードに対する試験網羅率の基準を設定すること。

表 5.3.8-1 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。

(2) 製作・単体試験

- (a) 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すること。
- (b) 自動コード生成を実施する場合は、自動コード生成ハンドブックに従って自動コード生成すること。
- (c) 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。自動コード生成を実施する場合は、自動コード生成ハンドブックに従ったものとする。
- (d) ソースコードレビューを開発計画の検証方針に従って実施すること。
- (e) ソフトウェア検証計画、ソフトウェア試験計画、および(1)(a)で定めた合格基準を基に単体試験仕様を作成すること。自動コード生成を実施する場合は、モデル単体試験で用いた試験仕様(テストケース)を含めた単体試験仕様を作成すること。
- (f) 単体試験仕様に従い単体試験を実施し、実施内容を単体試験の合否判定可能な形式で記録すること。
- (g) 単体試験は、(1)(b)で定めたソースコードに対する試験網羅率を満たすように実施すること。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。また、自動コード生成を実施する場合は、ソフトウェア設計プロセスでモデル単体試験(5.3.6.1(16)項参照)を実施した上で、モデル単体試験で使用した試験仕様(テストケース)に従い自動コード生成したコードの単体試験を実施し、自動コード生成元データ(モデルなど)と自動コード生成したコードとの等価性評価を行うこと。

- (h) 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、ソースコード品質評価を実施すること。(自動コード生成したコードも含む。) なお、自動コード生成したコード以外は、表 5.3.8-2 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 5.3.8-2 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

- (i) 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。
- (j) ソフトウェア製作で識別された運用前提および運用制約を明らかにすること。

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) 運用前提および運用制約
- (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) 運用前提および運用制約(更新)
- (3) 単体試験仕様
- (4) 単体試験記録(自動コード生成を実施する場合は、等価性評価結果を含む)
- (5) トレーサビリティ解析記録
- (6) ソースコード品質評価結果

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、ソフトウェア開発計画に従い、以下の事項を満たすアクティビティを実施すること。

- (1) 統合の準備
 - (a) 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。
 - (b) コンピュータシステム／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。
- (2) 統合の実施
 - (a) 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。
 - (b) 自動コード生成を実施する場合は、手動で製作したソースコードから生成されたソフトウェアと自動コード生成したコードから生成されたソフトウェアとを統合し、設計で明確にしたインタフェース仕様との整合性を確認すること。

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に発見された問題について、ソフトウェア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、定期的には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) 運用前提および運用制約
- (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) ソフトウェア統合において定義・評価したソフトウェアの品質にかかわる収集データの評価結果

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、ソフトウェア開発計画に従い、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) 想定されるシナリオ上の最大負荷
 - (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/ナレッジ資産のコンピュータシステムに対する適合性
 - (キ) 自動コード生成を実施する場合は、自動コード生成元データ(モデルなど)と自動コード生成したコードとの等価性
 - (ク) 自動コード生成を実施する場合は、手動で製作したソースコードから生成されたソフトウェアと自動コード生成したコードから生成したソフトウェアとを統合し、実ターゲット相当環境での正しいソフトウェアの挙動の妥当性を確認すること。
 - (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) 試験が合格となる基準を設けること。
 - (e) ソフトウェア(統合)試験手順作成時に、運用前提および運用制約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、運用前提および運用制約を更新すること。
 - (f) 試験の準備ができているか、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共同レビューを実施し、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、ソフトウェア製作以降に行った各種試験の有効性を評価し、必要な場合はソフトウェア製作以降に行った各種試験を再度実施すること。

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問題(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) 運用前提および運用制約
- (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
- (11) COTS/ナレッジ資産適合性評価結果

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) 運用前提および運用制約(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)試験において設定・収集・評価した品質指標データの評価結果

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ソフトウェア開発計画に従い、以下のアクティビティを実施すること。

- (1) インストールの準備
 - (a) ソフトウェアの開発環境におけるソフトウェアの保管書庫(リポジトリ)から、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。ここでの目標プラットフォームへのインストールとは、ターゲットコンピュータへの組み込み(ROMへの書き込みを含む)のことである。
 - (b) 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手順、インストール結果の確認手順、運用前提および運用制約を準備すること。
 - (c) インストールの準備の状況(インストールの事前検証、手順など)の確認を行

うこと。

- (2) インストールの実施
 - (p) 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。)
 - (a) インストール結果の確認手順に従い、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。

※：たとえばインストール後の元データと目標プラットフォーム上のデータはダンプなどによりファイル比較し、一致していることを確認する照合を行う。

5.3.12.2 インプット

- (1) ソフトウェア
- (2) 運用前提および運用制約

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
- (6) インストール確認結果
- (7) 運用前提および運用制約(更新)

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、ソフトウェア開発計画に従い、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) コンピュータシステム統合の基準と、統合後のインタフェースおよびソフトウェアシステム機能を検証するポイントを特定し、定義すること。
 - (b) コンピュータシステム/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。
 - (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を作成すること。なお、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) 想定されるシナリオ上の最大負荷
 - (ウ) 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、ソフトウェア要求仕様に対する網羅性
 - (エ) 例外や故障などの異常事象
 - (オ) COTS/ナレッジ資産のコンピュータシステムに対する適合性
 - (d) ソフトウェア検証計画およびコンピュータシステム総合試験仕様に従い、コンピュータシステム総合試験手順を作成すること。
 - (e) 試験準備(試験手順確認など)で発見された問題を記録・管理すること。
 - (f) 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作成時に、運用前提および運用制約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、運用前提および運用制約を更新すること。
- (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を活用して実施し、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、すでに実施済みの試験、検証、妥当性確認、共同レビュー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。
 - (f) コンピュータシステム総合試験においては、必要に応じ試験結果の共同レビューを実施し、試験継続の可否の判断を行うこと。
- (3) リリースするソフトウェアの構成管理情報、運用前提および運用制約を準備すること。

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問題(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) 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) コンピュータシステム方式設計仕様
- (4) ソフトウェア要求仕様
- (5) ソフトウェア検証計画
- (6) 運用前提および運用制約
- (7) 統合に対する制約条件の識別結果
- (8) ソフトウェアを組み込んだコンピュータシステム
- (9) 構成管理情報
- (10) COTS/ナレッジ資産適合性評価結果

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) 運用前提および運用制約(更新)
- (5) 統合に対する制約条件の識別結果 (更新)
- (6) ソフトウェア製品
- (7) 構成管理情報(更新)
- (8) 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おいて設定・収集・評価した品質指標データの評価結果

5.3.14 ソフトウェア製品の供給と導入

宇宙機では適用しない。

5.3.15 ソフトウェア製品の受け入れ

宇宙機では適用しない。

5.4 運用プロセス

本プロセスでは、宇宙機打ち上げ前および打ち上げ後に運用者および利用者に対して、宇宙機に組み込まれたソフトウェアの運用支援を行うためのアクティビティを定義する。

- (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) 運用者および利用者への支援

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) 運用開始後から運用終了に至るまでに期待される各基準（容量、稼働率、応答時間、安全性など）
 - (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（通常運用と、緊急オペレーションの手順の準備と確認を含む。）を実現するためのアプローチ

5.4.1.2 運用計画の立案

運用プロセスを実施するために、運用者および利用者への支援の計画（運用計画）を運用戦略に基づいて立案すること。なお、計画には以下を前提として考慮すること。

- (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プリングシステムやサービスを識別し、利用できるような状態にすること。

5.4.1.3 問題報告および変更要求に関する手順の確立

問題解決プロセス(6.8参照)に従い、宇宙機運用下でのソフトウェアの問題報告および変更要求に対する対処の手順を確立すること。

5.4.1.4 ソフトウェア書き換えに関する手順の確立

宇宙機運用下でのソフトウェアの書き換えに関わる手順を確立すること。

5.4.2 運用試験

宇宙機では適用しない。

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

宇宙機では適用しない。

5.4.4 運用結果の管理

宇宙機では適用しない。

5.4.5 運用者および利用者への支援

5.4.5.1 宇宙機運用開始前の支援

- (1) 下記の内容をコンピュータシステム担当またはサブシステム担当へインプットし、SOOH、SOPの源泉とすること。
 - (a) ソフトウェアが受け付けるコマンドリスト、コマンドフォーマット、個々のコマンドのデータパターン(固定値と可変値)、コマンドの送信手順など運用に必要なコマンド情報
 - (b) ソフトウェアが出力するテレメトリリスト、テレメトリフォーマット、テレメトリ出現頻度など、ソフトウェアが出力するテレメトリ情報
 - (c) 送信するコマンドと関連するテレメトリの相互関係
 - (d) ソフトウェア書き換え・照合に必要な軌道上書き換えファイルと地上側のソフトウェアのインタフェース
 - (e) ソフトウェア書き換え・照合に必要なメモリマップや手順
 - (f) ソフトウェアの設計から試験までに識別された運用制約
 - (g) 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、実装、インストールに関する変更を行う場合の運用制約

5.4.5.2 宇宙機運用開始後の支援

- (1) 運用者および利用者の要求事項(インシデントおよび問題を含む)に対し支援を行うこと。これらの要求およびその後の対応状況については、記録し、フォローすること。
- (2) 5.4.1.3 で確立した手順に従って問題解決を行い、必要に応じて運用者および利用者の要求を保守プロセス(5.5 参照)に引き渡す。これらの要求を明確にし、計画しかつ実施する処置の内容を、その要求元に報告する。結論が出るまで、すべての解決策を監視すること。
- (3) 報告された問題に対して本格的な解決処置をとる前に、当面の回避策があれば、その回避策を採用するかどうかの選択肢をその問題の報告元に対して提供すること。
- (4) 恒久的な修正や機能追加は、保守プロセス(5.5 参照)により運用中のソフトウェアに対して軌道上書き換えを行うこと。

5.5 保守プロセス

本プロセスは、訂正、改良に対応して宇宙機搭載ソフトウェアの部分的な修正(パッチ)あるいは機能追加/拡張などに伴うソフトウェアの追加修正を実施することに伴い、ソフトウェアおよび関連文書を修正する場合に行う。本プロセスは、ソフトウェア試験または宇宙機システム試験が終了し、開発した宇宙機搭載ソフトウェアをシステム側へ引き渡した以降、開発した宇宙機搭載ソフトウェアが組み込まれた宇宙機の運用が終了するまでの間、必要に応じて実施される。本プロセスには、宇宙機運用中におけるソフトウェアの書換えアクティビティも含まれる。

本プロセスは、宇宙機の運用が終了、すなわち宇宙機と共に搭載したコンピュータシステムが廃棄されることによって終了する。

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) 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) ミッション要求に基づくシステムの可用性、安全性などを踏まえてソフトウェア保守の実施/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) 保守(たとえば、計画されたフェーズ/問題修正のためのソフトウェア書き換え)の実行に必要なソフトウェア要素の正しいバージョンと情報へのアクセスを獲得すること。また、そのための優先順位と資源を確立すること。
 - (c) 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
 - (d) 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
- (2) 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。
- (3) 入手可能で運用可能で持続可能な保守となるよう活動を比較検討すること。
- (4) 保守をサポートするのに必要なイネーブリングシステムまたはサービスを識別し、利用できるような状態にすること。

5.5.1.2 保守計画の立案

- (1) 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すること。この計画には、以下の記載を含めること。
 - (a) 開発したソフトウェア製品の維持管理方法
 - (b) 体制および関連する文書
 - (c) データ(保守に必要な管理情報を含む)
 - (d) 記録

- (e) 保守環境(開発時の環境など)
 - (f) 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
- (2) 品質保証プロセス(6.3 参照)に従う、品質保証活動を計画すること。

5.5.2 問題把握および修正分析

- (1) 計画に従い、問題把握、必要な修正内容の分析を実施すること。
- (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およびソフトウェアの書き換え手順の合意を得ること。

5.5.3 修正の実施

修正が必要な場合は、合意された修正案に従って修正を実施すること。なお、修正の影響が波及する最上流のプロセスから開発プロセスを再度実施する手順を含めること。(付録 1 参照)

自動コード生成したコード部について、手動で修正する場合は、手動でソースコードを開発する場合と同等のアクティビティを実施すること。

5.5.4 ソフトウェアの書換え

5.5.4.1 ソフトウェアの書換え・照合手順

宇宙機で現用しているソフトウェアから修正されたソフトウェアへまたは以前インストールされていたソフトウェアへ書換える場合、以下を実施すること。なお、ソフトウェアにはパラメータを含むこと。

- (1) ソフトウェアの書換えを行うための手順を必要に応じて修正する。
- (2) ソフトウェア書換え後のソフトウェアバージョンについて決定する。
- (3) 書換え前のソフトウェアバージョンおよびパラメータ設定値について記録する。
- (4) ソフトウェアの書換え・照合手順を作成し、ソフトウェアの書換え・照合を実施する。なお、手順にはソフトウェアの書換えがうまく行かなかったさいのリカバリ手順を含める。
- (5) ソフトウェアの書換えが終了したことを関係者全員に通知する。

5.5.4.2 品質保証活動の実施

5.5.1.2(2)にて計画した品質保証活動を実施すること。

5.5.5 ロジスティクス支援の実施

宇宙機では適用しない。

5.5.6 保守結果の管理

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)、および重要な保守の結果を記録する。なお、宇宙機運用開始後の支援で扱われたインシデントと問題は、5.4.5.2(1)で扱う。

5.5.7 移行

宇宙機では適用しない。

5.5.8 ソフトウェア廃棄

宇宙機では適用しない。

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

文書化プロセスは、ソフトウェアまたはコンピュータシステムを開発、利用する全関係者が必要とする文書の管理計画を定義し、管理計画に従って文書発行・配付などを実施するためのプロセスである。

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) 作成
- (3) 文書発行
- (4) 文書の維持・改訂・廃棄

6.1.1 プロセス開始の準備

本標準で要求されるすべてのプロセスで作成・受領する文書の管理計画を立案すること。また、ソフトウェア製品の開発終了後も必要な関連文書を維持・改訂・廃棄ができるように計画立案すること。なお、文書管理計画には以下の情報を含む。

- (1) 文書の種類
- (2) 文書の目的
- (3) 文書の発行・入手予定時期
- (4) 関係する部門
- (5) 受領文書を含む入力情報、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)に関する手続きおよび責任
- (6) 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。
- (7) 作成・点検・発行・改訂・廃棄時における秘匿要求への対応を定めておくこと。

6.1.2 作成

- (1) 各文書は別途定義された形式／様式・技術的内容に従って作成すること。
- (2) 作成した文書の源泉情報の適切性について確認すること。
- (3) 作成した文書は、そこに含まれる情報の完全性を確認すること。
- (4) 作成した文書について、様式に従っているか確認し校正すること。
- (5) 作成した文書が、文書管理計画に定めた秘匿要求への対応に従っているかを確認すること。

- (6) 作成した文書に廃棄期限を設ける場合は、それが適切かを確認すること。
- (7) 文書は、発行に先立って、権限をもった者が妥当であることを点検し、承認すること。

6.1.3 文書発行

- (1) 文書管理計画に従って、適切な配付先に文書を発行すること。
- (2) 発行した文書について、関連する基準類に従い保管すること。
- (3) 発行時には、必要に応じ、最新版が確実に利用される処置を講ずること。

6.1.4 文書の維持・改訂・廃棄

文書管理計画に従って文書の維持・改訂・廃棄などの作業を行うこと。また、下記について考慮すること。

- (1) 改訂した文書は、そこに含まれる情報の完全性を確認すること。
- (2) 改訂した文書が、文書管理計画に定めた秘匿要求への対応に従っているかを確認すること。
- (3) 文書が廃棄期限に到達した場合は、文書管理計画に従い、廃棄期限延長などの対応を実施すること。
- (4) 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。
- (5) 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、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じて適用される。

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) 構成管理記録および状況報告
- (5) 構成評価
- (6) リリース管理
- (7) 構成監査の実施

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) 必要な利害関係者を含んだ変更管理
- (5) ベースライン承認と変更要求の承認におけるリスクと影響のレベルの検討

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。計画には以下の情報を含むこと。

- (1) 構成管理におけるアクティビティ
- (2) アクティビティを実施するための手続き(適用する基準類、ツールなど)とスケジュール(管理体系を適用した、構成管理のアクティビティを開始する時期、継続する期間などの構成管理としてのマイルストーン)
 - (a) 構成品目へのアクセス、変更、処置の制御
 - (b) 構成管理品目の識別を実施するために必要な基準、およびベースラインの確立・維持のための基準または実施時期(イベント)
- (3) アクティビティを実施する責任を負う組織と、その各部門の役割/相互関係/協力関係
 - (a) 構成管理活動の役割と責任を含むこと。
- (4) 関係する他の組織と、その役割/相互関係/協力関係
 - (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整合
- (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) 構成管理品目の識別
プロジェクトにおいて、ソフトウェア開発を通して生成する成果物の中から、構成管理の対象とな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(※1)
 - (b) その他の識別詳細情報(※2)
- (3) ベースラインの確立
目的(※3)に応じ、構成ベースラインを確立すること。そのベースラインに含まれる構成品目およびバージョンを明確にすること。
- (4) バージョン管理体系
バージョン管理単位およびそれらを管理する体系、手続きを確立すること。ここで確立される体系、手続きにより、構成管理活動の履歴を記録し、追跡を可能とすること。
- (5) 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。

※1：モデルベース開発、自動コード生成を含む場合、利用するモデル、自動コード生成元データ(モデルなど)、ツール、シミュレータなどの個々の要素のバージョンも含むこと。

※2：モデルベースシミュレーションおよびモデル単体試験で利用したモデルやツール、シミュレータ、パラメータなどの構成管理情報、自動コード生成したコードに対する自動コード生成元データ(モデルなど)やツールの構成管理情報が参照できること。

※3：目的としては、以下のようなものがある

- i. 設計審査会の基になっている構成の識別
- ii. ソフトウェア統合試験後の構成の識別
- iii. 打ち上げ時のチェックアウト終了後の構成の識別

6.2.3 構成変更管理(Configuration Control)

6.2.3.1 構成管理品目ごとの管理

構成管理品目に対し、6.2.2(4)で確立された管理体系に基づき、その変更の履歴を記録し、管理すること。

6.2.3.2 構成品目一式の管理

構成品目一式(※)に対し、適宜、構成品目が過不足なく含まれ、適切なバージョンであることを確認すること。

また、その内容を記録し、履歴を管理すること。

※： 構成管理品目の中から、目的に応じて選択された品目の組み合わせ。ベースラインの管理については本項を適用する。

6.2.3.3 変更要求管理

(1) 変更要求の識別および記録（ウェーバを含む）

構成管理品目に対する変更要求を識別すること。また、ほかに変更要求が存在する場合は、他の変更要求との関連性を評価すること。

変更依頼、変更根拠、分析結果、修正状況および変更依頼のステータスを記録すること。

(2) 変更の分析および評価

要求されている変更に対して、技術的影響、コスト、変更におけるメリット、デメリットを分析、評価すること。

変更を実施するさいに実施されるべき検証および妥当性確認活動を分析すること。

(3) 変更要求の承認または不承認

審査により、変更要求の承認、不承認を決定すること。

修正理由および修正承認が追跡できるように審査結果を記録すること。

(4) ソフトウェア品目の修正、検証確認

承認された変更要求に基づき、品目の修正、検証の終了を確認すること。

6.2.4 構成管理記録および状況報告

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 構成評価

構成品目一式のベースラインについて、その構成品目がベースラインの目的に対し、機能的および物理的に過不足なく、適切なバージョンが含まれているかを評価すること。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

6.2.6.1 リリース管理の実施

ソフトウェア製品またはコンピュータシステムのリリースを、決められた手続きに従い管理すること。

また、リリースごとに、6.2.3.2を適用した構成品目一式の管理を実施すること。

構成管理品目の原本およびマスターコピーは、ソフトウェア製品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維持すること。

また、ソースコードおよび文書は、組織の方針(※)、およびソースコードおよび文書に含

まれる情報の種類に応じて決められる方針に従って、取り扱い、保管すること。

※：ここでいう”組織の方針”とは、社規やプロジェクトの規則を含む

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宇宙機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) 品質保証活動の実施
- (3) 製品品質の保証
- (4) プロセスの保証
- (5) 品質システムの保証
- (6) 品質保証記録の管理
- (7) 自動コード生成における管理

6.3.1 プロセス開始の準備

6.3.1.1 品質保証責任者の任命

開発組織から独立した責任と権限を持つ責任者を任命すること。その責任者が品質保証活動の管理、実施状況の確認および改善などの実行に責任を持つこと。

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) 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計画を立案すること。なお、品質保証活動計画には、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) 品質を保証するために実施するアクティビティ
- (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) 品質保証に関わる教育、訓練
- (8) 購買管理・供給業者に対する品質保証に関する要求・アクティビティ、および各供給元(各供給業者を含む)への品質保証活動
- (9) ソフトウェア製品の再利用における管理

- (10) 自動コード生成における管理
- (11) モデルベース開発における管理
- (12) リリースに関する一連の手続き
- (13) 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
- (14) 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)との関連性と選択したアクティビティおよびタスク
- (15) 製品の受け入れ基準含む、プロセス、製品の品質評価基準および評価手法
- (16) 製品に特有かつ必須の検証、妥当性確認、監視、測定、レビュー、検証、アセスメントと試験の活動

6.3.1.4 品質保証活動計画の合意

品質保証活動計画について、取得者と合意すること。

6.3.2 品質保証活動の実施

品質保証活動計画に従い、品質保証活動を実施すること。

6.3.3 製品品質の保証

- (1) ソフトウェアまたはコンピュータシステム、およびそれらの関連文書について、計画されたものが不足なく作成されていること。
- (2) ソフトウェア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。
- (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。

6.3.4 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。

6.3.5 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。

6.3.5.1 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。

6.3.5.2 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。

6.3.5.3 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。

6.3.5.4 ソフトウェア製品の再利用における管理

再利用可能な COTS 品目、ソフトウェア、ソースコード、関連文書に含まれる設計要素、試験手順などのソフトウェア製品(JERG-0-049D 付録 6 参照)に関しては以下を管理項目に含めること。

- (1) ソフトウェア製品を再利用することの利点
- (2) ソフトウェア製品の再利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再利用ソフトウェア製品の適合性
 - (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
- (3) 再利用ソフトウェア製品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

6.3.5.5 取扱い、保管およびラベリング

宇宙機では適用しない。

6.3.5.6 品質の継続的改善活動

組織レベルとプロジェクトレベルで品質の継続的改善を実施していくために、それに必要な情報収集を行い、後続の開発にフィードバックする。

6.3.6 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。
- (3) 品質保証活動に関連した記録は、合意に基づき関係者が閲覧できるようにすること。

6.3.7 自動コード生成における管理

自動コード生成を実施する場合は、以下の管理項目について実施状況を評価すること。

(1) 自動コード生成に関わるツール選定

自動コード生成に必要な、自動コード生成元データ(モデルなど)作成ツール、自動コード生成元データ(モデルなど)からコードを自動コード生成するツール、自動コード生成元データ(モデルなど)とコードのメトリクス収集ツール、自動コード生成元データ(モデルなど)および同生成元データに使用するパラメータの構成管理ツールなどの選定にあたり、以下の評価観点で基準を定義し、定義した基準で評価した結果をま

とめる。同等レベルの品質要求を有する他の製品の実績の評価により、以下の観点に代えることができる。

- (a) 識別した自動コード生成ハンドブックとの適合性
 - (b) モデル単体試験におけるモデル試験網羅の計測環境
 - (c) モデルベースシミュレーションおよびモデル単体試験で使用するツールとの連携における親和性
 - (d) 生成したコードを入力の一つとして使用する他のツール（たとえば、コンパイラ、コード管理システム）との親和性
 - (e) プロジェクトで必要なツールのカスタマイズ可否
 - (f) パラメータを含むツールの構成変更管理
 - (g) ツールの品質保証情報（バージョン、バージョンアップなどの情報など）の入手可否
 - (h) 自動コード生成したコードの性能（サイズ、速度）
- (2) 自動コード生成したコードの修正
- (a) 自動コード生成したコードは、コードレベルでは修正せず、自動コード生成元データ（モデルなど）で修正すること。
 - (b) やむを得ず、自動コード生成したコードを手動で修正する場合は、コーディング規約の準拠、コードレビュー、単体試験など、手動でソースコードを開発する場合と同等の開発プロセスや構成管理プロセスのアクティビティを実施すること。

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。検証プロセスは以下のアクティビティからなる。なお、検証の手段としてはレビュー、試験などの方法がある。

- (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、特性にはコンピュータシステム／ソフトウェア要求、アーキテクチャ、セキュリティ、クリティカルな品質特性などの設計特性、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) コンピュータシステム／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) 検証の区分（検査・分析・解析・実証・試験など）、適用する手法または技術、

- 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ンネーブリングシステムまたはサービスを識別し、利用できるような状態にすること。
 - (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。実施のさいは、検証計画に従って、各タスクに記述される観点(6.4.2.1~6.4.2.6)を考慮して検証を行うこと。

- (1) プロセスの検証
- (2) 要求事項の検証
- (3) 設計の検証
- (4) ソースコードの検証
- (5) 統合の検証
- (6) 文書化の検証

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基準・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 要求事項の検証

次に示す観点を考慮すること。

- (1) コンピュータシステムへの要求がコンピュータシステムに対する要求仕様に適切に反映されている。コンピュータシステムに対する要求仕様に一貫性があり、実現可能かつ検証可能である。
- (2) コンピュータシステムに対する要求仕様が、ハードウェア、ソフトウェア、運用に適切に割り当てられている。
- (3) ソフトウェア要求仕様に一貫性があり、実現可能かつ検証可能である。また、正確にコンピュータシステムに対する要求仕様を反映している。
- (4) ソフトウェア要求仕様がソフトウェア品目に適用される基準などを満たしている。
- (5) 安全性・セキュリティなど特に注意を要する事項に関して、上位の要求や品目に適用される基準など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) 設計は要求仕様を満足し、要求仕様に対してトレーサビリティが取られていること。
- (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。

- (4) ソフトウェア設計がソフトウェア品目に適用される基準類の要求を満たしている。
- (5) モデルベース開発を実施する場合は、モデルベースシミュレーションが実施されている。
- (6) 自動コード生成を実施する場合は、モデル単体試験が実施されている。

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) 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。
- (5) 自動コード生成を実施する場合は、自動コード生成元データ(モデルなど)と自動コード生成したコードの等価性が評価されている。

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。
- (2) ソフトウェアを構成するモジュールやデータが、完全(Completely)にかつ正確(Correctly)に統合されている。
- (3) コンピュータシステムの構成要素が完全(Completely)にかつ正確(Correctly)に統合されている。
- (4) ソフトウェアの統合とコンピュータシステムの統合が、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) 文書化の活動が文書化プロセス(6.1参照)に従って適切に行われている。
- (2) 文書化の活動が適切なタイミングで行われている。
- (3) 文書の構成管理が、決められた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されるさいに、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) 開発するコンピュータシステムへの要求に取り込むべきシステム制約を識別する。

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) 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プリングシステムやサービスを識別し、利用できるような状態にすること。
 - (f) 識別されたインシデントおよび問題に対する問題解決プロセス(6.8 参照)に従った処置。

※：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

妥当性確認計画に従って、以降に記述される観点を考慮して、妥当性確認を行うこと。

- (1) ソフトウェアまたはコンピュータシステムが、想定する使用状況および使用方法を満足しているか。ソフトウェアまたはコンピュータシステムが、その要求仕様を満足しているか。
- (2) 実環境に相当する環境における妥当性確認において、実環境ではなく模擬環境を用いる場合は、実環境と模擬環境の差異を評価しているか。

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、ソフトウェア開発の活動状況および成果物を、適宜、評価するためのプロセスである。

共同レビューでは、ソフトウェア開発の管理(進捗など)および技術の両面をレビューする。

このプロセスは、あらゆるレビューにおいて適用してよい。

共同レビュープロセスは以下のアクティビティからなる。

- (1) プロセス開始の準備
- (2) ソフトウェア開発管理レビュー
- (3) 技術レビュー

6.6.1 プロセス開始の準備

ソフトウェア開発の開始時に、そのプロジェクトのマイルストーンに応じたレビューを、そのレビューで必要となる資源(予算、人、場所、設備およびツールなど)を含め計画すること。

- (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、臨時的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) レビューごとに、以下の事項について、レビュー実施者またはレビュー受審者の双方の当事者と合意すること。
 - (a) レビュー対象
 - (b) レビューの範囲および議題
 - (c) レビューの手順
 - (d) レビューの開始および終了の基準
- (4) レビュー時に識別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) レビュー実施者またはレビュー受審者の双方の当事者は、以下の項目についてレビュー関係者と合意し、その結果を配付すること。
 - (a) レビュー結果
 - (b) 識別された問題の対処に関する責任の所在
 - (c) レビュー終了基準

6.6.2 ソフトウェア開発管理レビュー

6.6.2.1 ソフトウェア開発状況の評価

開発状況をソフトウェア開発計画と照らし合わせ、以下の観点で評価すること。

- (1) 計画に従って開発が進んでいる。
- (2) ソフトウェア開発遂行に必要な資源が適切に配分され、管理されている。
- (3) 計画に対するずれをモニタし、ソフトウェア開発の方向変更または代替計画の必要

性が判断されている。

- (4) プロジェクトの成功を危うくする可能性のあるリスクを識別し、評価し、管理されている。

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、事前に資料など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／指摘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員／下流工程プロセス作業員など)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。

6.6.3.3 開発状況の評価

ソフトウェア製品またはコンピュータシステムを、以下の観点で評価すること。

- (1) ソフトウェア製品またはコンピュータシステムが、完全(Complete)である。
- (2) ソフトウェア製品またはコンピュータシステムが、基準類および仕様に従う。
- (3) ソフトウェア製品またはコンピュータシステムに対する変更が適切に実施されている。
- (4) 次の活動に移る準備(プロセス移行準備)ができています。
- (5) 開発、運用または保守が、プロジェクトの計画、予定、基準類および指針に従って実行されている。

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アクティビティである。

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し、プロセスにおける強み、弱みを識別する。
- (2) アセスメント手法および適用する本標準、基準類などに精通し、ソフトウェアまたはコンピュータシステム開発担当者本人およびその指揮権者を除く第三者をアセッサーとして選定する。
- (3) 要改善事項の改善提案を含むアセスメント結果をアセスメント対象組織に報告する。また、アセスメント対象組織への報告結果を含む最終成果をスポンサーに文書で報告する。

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、開発プロセス(5.3 参照)、運用プロセス(5.4 参照)、保守プロセス(5.5 参照)またはその他のプロセスの、実行の過程で発見された問題を分析し、解決するためのプロセスとする。

なお、問題解決における修正の種類としては、訂正と改良を含む。

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) 問題解決
- (3) 傾向分析
- (4) 是正・予防処置

6.8.1 プロセス開始の準備

6.8.1.1 問題解決計画

品質保証活動計画に基づき、ソフトウェアまたはコンピュータシステムで抽出されたすべてのインシデントおよび問題を取り扱うために、問題解決プロセスを確立すること。

問題の基準に応じて、以下のような記録および対処手順を定めること。

- (1) インシデントおよび問題の管理方法(受領および記録の方法・報告・維持方法(保管期間を含む))
- (2) 問題の原因分析と処置方針

- (3) 問題処置状況の追跡方針
- (4) 是正・予防処置の方針

6.8.2 問題解決

問題解決計画に従ってインシデントを分析し、問題と識別された場合、必要な是正処置を行い問題を解決すること。特に問題が設計仕様やソースコードの変更を必要とする場合には、下記のアクティビティのうち、管理レベルに応じて必要な項目が問題解決計画に含まれていることを確認し、問題解決を実施すること。

6.8.2.1 インシデントおよび問題の受領と記録

- (1) 主ライフサイクルプロセス、支援ライフサイクルプロセスで発見されたすべてのインシデントおよび問題は、各プロセスにおける記録の後、迅速に報告され、問題解決プロセスで受領すること。
- (2) インシデントおよび問題の記録を開始すること。
- (3) インシデントが発生した場合、既知のインシデントや問題と関連付けられるよう識別し、管理すること。
- (4) 是正処置を必要とするものについては、問題として管理すること。そのさいに優先順位をつけることが望ましい。
- (5) インシデントおよび問題の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とともに、問題と識別された場合、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。

また、必要に応じて、報告された問題が再現することを試験環境／保守環境にて確認または机上で解析などによって検証すること。適用する検証手法は検証の制約、限界に応じて選定すること。

- (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) 修正の要否判断(修正による効果、作業規模、致命度、緊急性などから修正を実施するかの判断)を行うこと。
- (2) 分析に基づき、修正方法を検討すること。修正方法は複数検討し、その中から実施する修正を選択できるようにすることが望ましい。
- (3) 実施するソフトウェア修正方法を記録し、問題報告、変更要求およびその分析結果の対応付けがわかるように管理すること。
- (4) 運用方法への留意点などの特記事項があれば文書化すること。

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) 問題解決方法を決定し、関係者と合意すること。必要に応じて、問題解決の計画を立案すること。

- (2) ソフトウェアリリース後の問題の場合で、暫定措置として問題の回避あるいは問題の影響を軽減する必要がある場合、対処方法を決定し、関係者と合意すること。必要に応じて、暫定処置の計画を立案すること。

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、関係者に通知すること。問題は、終結に至るまで追跡管理を実施する。

6.8.2.6 問題の報告

問題の記録を適切に報告すること。

6.8.2.7 記録とモニタ

インシデントおよび問題解決の一連のアクティビティ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。

6.8.2.8 記録の維持

インシデントおよび問題の記録を計画に基づいた期間保管すること。

6.8.3 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。

6.8.4 是正・予防処置

重大問題(類似問題の発生など)は、必要に応じ再発を防止するための対応を実施すること。また、関連／類似する作業プロセスや製品および関連する部門に対し、適切な情報の水平展開を実施すること。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。

付録

付録1 5.5.3 修正の実施の補足

ソフトウェア(パラメータを含む)修正の実施においては、修正が掛かる最上流のプロセスから開発プロセスを再度実施すること。たとえば、コンピュータシステム要求分析に遡る修正の場合には、5.3.3 コンピュータシステム要求分析以降の開発プロセスを再実行して確認した後にソフトウェアの書き換えを行う。

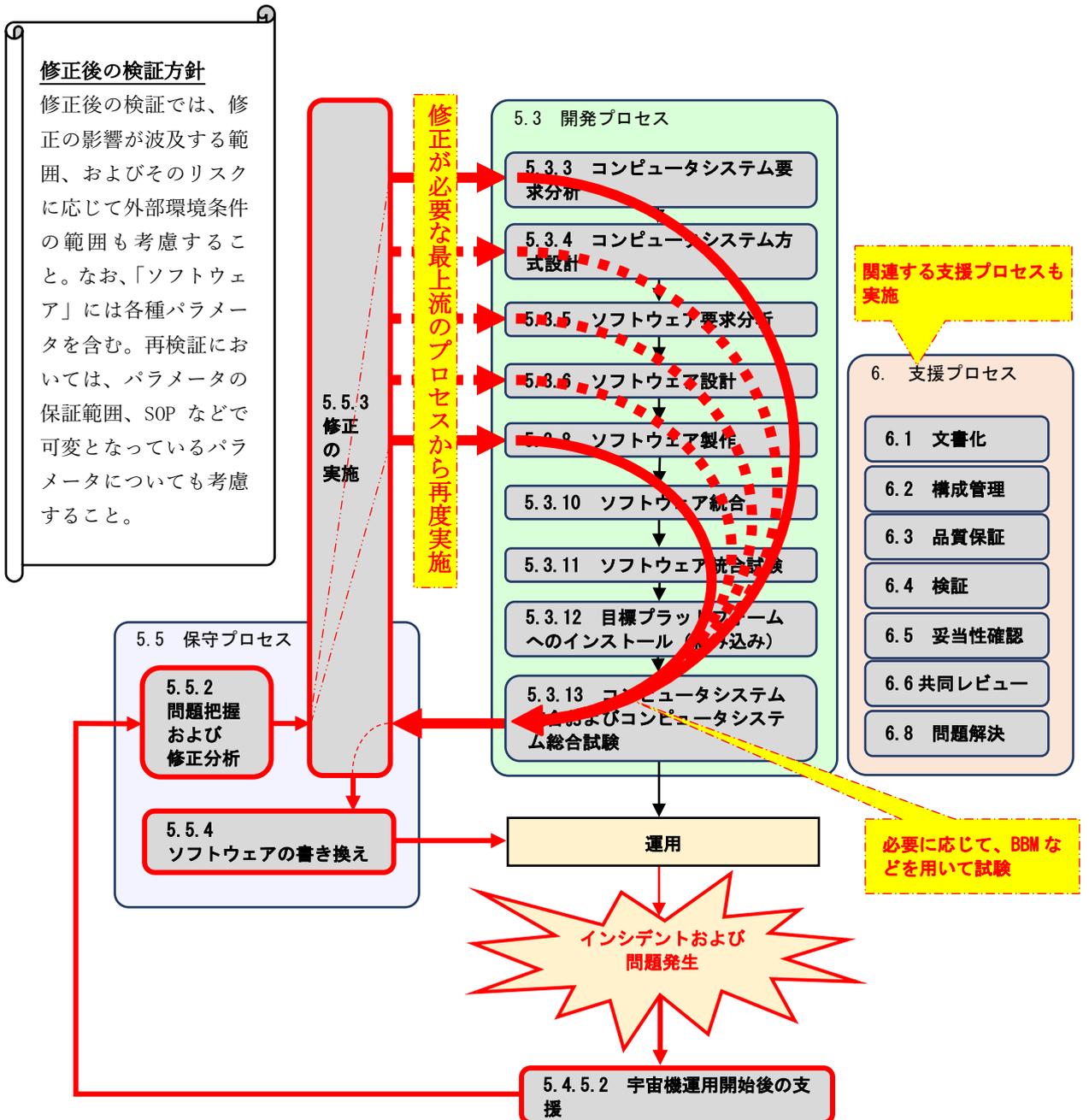


図-付録1 5.5.3 修正の実施の補足

付録2 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2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2 要求マッピングマトリックス

JERG-2-610の要求項目		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		◎	◎	◎	◎
	但し、(1)(a)(ア)、(2)(g)(ア)のみ右記の通り適用	◎	◎	◎	安全性は○ 安全性以外 は◎
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		◎	◎	◎	◎
	但し、トレーサビリティ評価に関し、 5.3.3.1(2)(e)、5.3.3.3(4)は右記の通り	◎	◎	◎	○
5.3.4 コンピュータシステム方式設計		◎	◎	◎	◎
	但し、トレーサビリティ評価に関し、 5.3.4.1(8)、5.3.4.3(4)は右記の通り	◎	◎	◎	○
5.3.5 ソフトウェア要求分析		◎	◎	◎	◎
	但し、トレーサビリティ評価に関し、 5.3.5.1(11)、5.3.5.4(3)は右記の通り	◎	◎	◎	○
	計測活動に関し、 5.3.5.2、5.3.5.4(9)は右記の通り	◎	◎	○	○
	審査に関し、 5.3.5.5は右記の通り	◎	◎	◎	○
5.3.6 ソフトウェア設計		◎	◎	◎	◎
	但し、トレーサビリティ評価に関し、 5.3.6.1(12)、5.3.6.4(3)は右記の通り	◎	◎	◎	○
	計測活動に関し、 5.3.6.2、5.3.6.4(9)は右記の通り	◎	◎	○	○
	審査に関し、 5.3.6.5は右記の通り	◎	◎	◎	○
5.3.8 ソフトウェア製作		◎	◎	◎	◎
	但し、 5.3.8.1(1)(b) ソースコードに対する試験網羅率は 右記の通り	表 5.3.8-1 参照			
	5.3.8.1(2)(h) モジュールの循環的複雑度は右記の 通り	表 5.3.8-2 参照			
	トレーサビリティ評価に関し、 5.3.8.1(2)(i)、5.3.8.4(5)は右記の通り	◎	◎	◎	○
	計測活動に関し、 5.3.8.2、5.3.8.4(6)は右記の通り	◎	◎	○	○
	審査に関し、 5.3.8.5は右記の通り	◎	◎	○	○
5.3.10 ソフトウェア統合		◎	◎	◎	◎
	但し、計測活動に関し、 5.3.10.2、5.3.10.4(4)は右記の通り	◎	◎	○	○
5.3.11 ソフトウェア統合試験		◎	◎	◎	◎
	但し、計測活動に関し、 5.3.11.2、5.3.11.4(7)は右記の通り	◎	◎	○	○

JERG-2-610 の要求項目		SW CC			
		A	B	C	D
	審査に関し、 5.3.11.5 は右記の通り	◎	◎	◎	○
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)	◎	◎	◎	◎
5.3.13	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	◎	◎	◎	◎
	但し、計測活動に関し、 5.3.13.2、5.3.13.4 (8) は右記の通り	◎	◎	○	○
5.4	運用プロセス				
5.4.1	プロセス開始の準備	◎	◎	◎	◎
5.4.5	運用者および利用者への支援	◎	◎	◎	◎
5.5	保守プロセス				
5.5.1	プロセス開始の準備	◎	◎	◎	◎
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	◎	◎	◎	◎
5.5.6	保守結果の管理	◎	◎	◎	◎
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	◎	◎	◎	◎
6.3	品質保証プロセス	◎	◎	◎	◎
	但し、6.3.3 (3) IV&V は右記の通り	◎	◎	○	○
	6.3.5.1 教育・訓練は右記の通り	◎	◎	◎	○
6.4	検証プロセス	◎	◎	◎	◎
	但し、6.4.2.2(5)は右記の通り	◎	◎	◎	安全性は○ 安全性以外は◎
	トレーサビリティ評価に関して、 6.4.2.3 (1)、6.4.2.4(1)は右記の通り	◎	◎	◎	○
	6.4.2.6 文書化の検証 は右記の通り	◎	◎	○	○
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	◎	◎	◎	◎
	但し、6.8.3 傾向分析、6.8.4 是正・予防処置は右記の通り	◎	◎	◎	○