

ソフトウェア開発標準
(宇宙機用)

平成 23年 5月 31日 A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	参考文書	1
3	用語、定義および略語	1
4	一般要求事項	6
5	主ライフサイクルプロセス	8
5.1	欠番	8
5.2	欠番	8
5.3	開発プロセス	8
5.3.1	プロセス開始の準備	9
5.3.2	全開発プロセス適用事項	10
5.3.3	コンピュータシステム要求分析	10
5.3.4	コンピュータシステム方式設計	11
5.3.5	ソフトウェア要求分析	12
5.3.6	ソフトウェア設計	14
5.3.7	欠番	16
5.3.8	ソフトウェア製作	16
5.3.9	欠番	18
5.3.10	ソフトウェア統合	18
5.3.11	ソフトウェア統合試験	19
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)	21
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	22
5.3.14	ソフトウェア製品の供給と導入	24
5.3.15	ソフトウェア製品の取得	26
5.4	運用プロセス	26
5.4.1	プロセス開始の準備	27
5.4.2	運用試験	27
5.4.3	ソフトウェアまたはコンピュータシステムの運用	27
5.4.4	ユーザサポート	28

5.5	保守プロセス	28
5.5.1	プロセス開始の準備	28
5.5.2	問題把握および修正分析	29
5.5.3	修正の実施	29
5.5.4	ソフトウェアの書き換え	29
5.5.5	移行	29
5.5.6	ソフトウェア廃棄	30
6	支援ライフサイクルプロセス	31
6.1	文書化プロセス	31
6.1.1	プロセス開始の準備	31
6.1.2	作成	32
6.1.3	発行	32
6.1.4	改訂	32
6.2	構成管理プロセス	32
6.2.1	プロセス開始の準備	33
6.2.2	構成識別	33
6.2.3	構成変更管理	33
6.2.4	構成変更状況の記録	33
6.2.5	構成変更状況の評価	34
6.2.6	リリース管理と出荷	34
6.3	品質保証プロセス	34
6.3.1	プロセス開始の準備	34
6.3.2	製品品質の保証	35
6.3.3	プロセスの保証	35
6.3.4	品質システムの保証	36
6.4	検証プロセス	37
6.4.1	プロセス開始の準備	37
6.4.2	検証	38
6.5	妥当性確認プロセス	39
6.5.1	プロセス開始の準備	40

6.5.2	妥当性確認	40
6.6	共同レビュープロセス	41
6.6.1	プロセス開始の準備	41
6.6.2	プロジェクト管理レビュー	41
6.6.3	技術レビュー	42
6.7	アセスメントプロセス	43
6.7.1	プロセス開始の準備	43
6.7.2	アセスメントの実施	43
6.8	問題解決プロセス	43
6.8.1	プロセス開始の準備	43
6.8.2	問題の解決	44
6.8.3	予防	45
6.8.4	傾向分析	45
	付録	46

1 適用範囲

本標準は、宇宙機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、本標準に準じた第3階層標準また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 12207 : 1995 Information technology - Software life cycle process
- (2) JIS X0160-1996 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 15504-2 : 2003 Software engineering - Process assessment — Part 2 : Performing an assessment
- (6) 共通フレーム 2007 (Copyright IPA/SEC 2009)

3 用語、定義および略語

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、相関の強いタスクの集合。
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。

用語	説明
受け入れ検査および試験	ソフトウェア製品の取得にさいして、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。
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 (ISO9000)。
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。

用語	説明
コンピュータシステム	<p>開発対象であるソフトウェアを実行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的设计作業として、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)</p> <p>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>
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。</p>
ソフトウェア	<p>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が、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>

用語	説明
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・詳細化がされ、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(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。
トレーサビリティ	上位文書との対応。

用語	説明
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。
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。
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> <ol style="list-style-type: none"> (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。 <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	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。

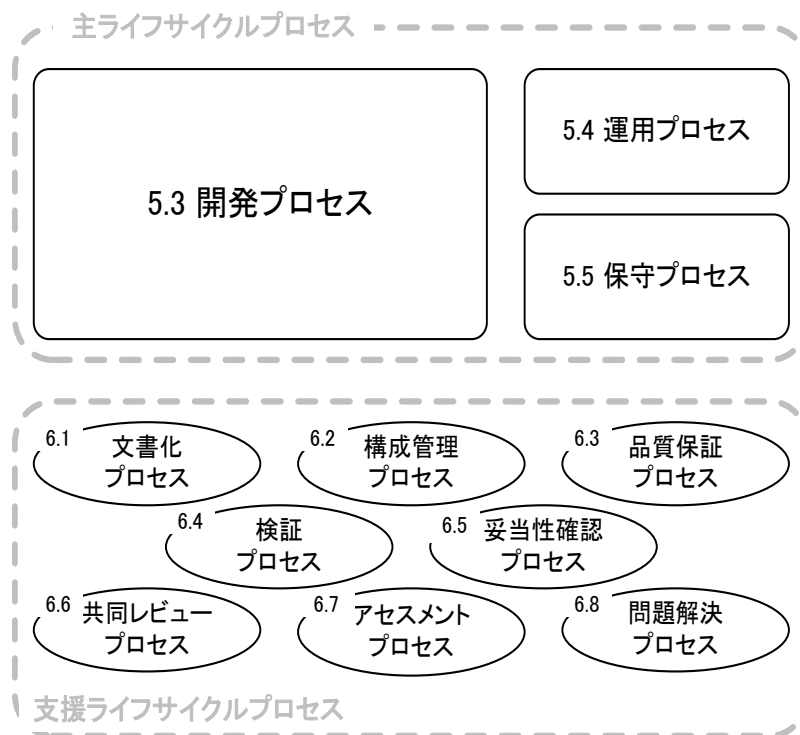


図4.1 プロセス全体図

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。

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・取得など。
	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・改修・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。

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取得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。

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活動を実施すること。

- (1) 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) 対象となるコンピュータシステムの範囲
 - (b) 対象となるソフトウェアの識別
 - (c) 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)
 - (d) ソフトウェア開発プロセスおよびアクティビティの定義(※)
※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。
 - (e) 各開発プロセスにおけるアクティビティとその実施管理計画
 - (f) 審査計画
 - (g) 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
 - (h) 文書作成計画(作成部門と作成時期を含む)
 - (i) 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)・進捗管理の方法
 - (j) 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)
 - (k) COTS および再利用ソフトウェアに関する管理計画(以下を含むものとする)
 - (ア) COTS 品目、再利用ソフトウェア品目の識別
 - (イ) COTS 品目、再利用ソフトウェア品目に対する品質保証プロセスの定義
 - (l) コンピュータシステムとの適合性評価計画
- (2) ソフトウェア開発計画を文書化すること。

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

開発プロセス全体を通じて、以下の要求を満たす活動を実施すること。

- (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。

5.3.2.1 インプット

- (1) ソフトウェア開発計画

5.3.2.2 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に対し、以下の要求を満たす活動を実施すること。

- (1) 要求抽出

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。

- (2) 要求仕様作成

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。

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とのトレーサビリティを評価すること。

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) 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

5.3.4 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活動を実施すること。

- (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(アーキテクチャ)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) 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。
- (4) 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。
- (5) 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。
- (6) 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。

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

5.3.5 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、以下の活動を実施すること。

- (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) 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。
- (3) 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。
- (4) 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、処理機能に関する仕様を含めること。
- (5) 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。
- (6) 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、文書化すること。
- (7) 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。
- (8) 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。
- (9) 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。

- (10) 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。
- (11) 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・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。
- (12) 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による確認を計画している場合、あるいは、試験を実施せず検証のみ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。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

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・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。

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の段階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活動を実施すること。

【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間の関係を明確にし、ソフトウェア(基本)設計を行うこと。
- (2) 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、メモリなどのリソース要求)に対する設計、配分を含めること。
- (3) インタフェース仕様について、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】

- (4) 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。
- (5) インタフェース仕様について、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】

- (6) 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、文書化すること。
- (7) 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、実現可能性を評価すること。
- (8) 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。
- (9) 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。
- (10) 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。
- (11) 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

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・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。

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活動を実施すること。

- (1) コーディングの基準となるルールをコーディング規約として定義すること。

- (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。
- (4) 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。
- (5) ソフトウェア検証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、実施内容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、ソースコードに対する試験網羅率の基準を設け、その基準を満たすよう実施すること。少なくともソースコードの分岐を十分に網羅するものであること。
- (8) コードチェックツールなど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。
- (9) 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。
- (10) 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のデータ定義と、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

5.3.8.4 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・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。

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活動を実施すること。

- (1) ソフトウェア統合し、ソフトウェア統合後のベースラインを確定すること。
- (2) 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にかかわる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)

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)

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活動を実施すること。

- (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
 - (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などを実施し、試験継続の可否の判断などを行うこと。
- (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)

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・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。

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活動を実施すること。

- (1) 目標プラットフォームにインストール(組み込み)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(組み込み)手順、インストール(組み込み)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。

- (3) インストール(組み込み)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) 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(組み込まれ)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(組み込み)可能な形態)
- (2) 構成管理情報
- (3) インストール(組み込み)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(組み込み)結果の確認手順

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活動を実施すること。

- (1) 試験準備
 - (a) コンピュータシステム統合を行うこと。
 - (b) 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。
 - (c) 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。

(d) 試験準備(試験手順確認など)で見つかった問題を記録・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中に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、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。

5.3.13.2 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)

5.3.14 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の供給と導入に、以下の活動を行うこと。

- (1) 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(組み込み)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(組み込み)、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(組み込み)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(組み込み)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取得

5.3.15.1 アクティビティ

ソフトウェア製品を取得する以下の活動を行うこと。

- (1) 受け入れ検査および試験の計画を立案し、受け入れ検査および試験の仕様、受け入れ検査および試験の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、取得者がその試験内容の承認を行うこと。
- (2) 受け入れ検査および試験の計画、受け入れ検査および試験の仕様、受け入れ検査および試験の手順に従い、受け入れ検査および試験を実施すること。また、受け入れ検査および試験の記録を残すこと。
- (3) 取得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の記録
- (5) インストール(組み込み)手順

5.3.15.3 アウトプット

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

5.4 運用プロセス

以下のソフトウェアまたはコンピュータシステムの運用およびユーザに対する運用支援を行うこと。

- (1) プロセス開始の準備

- (2) 運用試験
- (3) ソフトウェアまたはコンピュータシステムの運用
- (4) ユーザのサポート

5.4.1 プロセス開始の準備

プロセスを実施するための計画を作成し実行すること。また、運用に関する標準を定めること。

5.4.1.1 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。

5.4.1.2 システム運用およびユーザサポートに係る作業手順の確立

ソフトウェアまたはコンピュータシステムの運用環境下での試験手順、修正依頼を保守プロセス(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

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。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 ソフトウェアまたはコンピュータシステムの運用

ソフトウェア使用マニュアルに従って所定の環境において運用すること。

5.4.4 ユーザサポート

ユーザサポートに対し、以下の活動を行うこと。

- (1) ユーザに対しサポートサービスを提供すること。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こと。
- (2) ユーザサポートにおいて問題が識別された場合、5.4.1.1 で確立した手順に従って問題を解決すること。
- (3) 識別された問題に対して、暫定的な回避策がある場合、その回避策をユーザに提供すること。

5.5 保守プロセス

保守対象であるソフトウェア製品(設計文書を含む)並びに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。以下の活動を行うこと。

- (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) 移行
- (6) ソフトウェア廃棄

5.5.1 プロセス開始の準備

保守プロセスを実施するための計画(保守計画)を作成し、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、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、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

宇宙機の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/拡張などについても、本標準を適用する。

※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 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。

5.5.5.1 欠番

5.5.5.2 移行計画の作成と実行

ソフトウェア製品の移行計画を作成し、実施すること。なお、計画には以下の観点を考慮すること。

- (1) 移行のための要求分析および要求定義
- (2) 移行のためのツール
- (3) ソフトウェア製品およびデータの変換
- (4) 移行リハーサル

- (5) 移行の実施
- (6) 移行結果の確認
- (7) 旧環境に対するサポート

5.5.5.3 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

5.5.5.4 旧環境の保管

旧環境に関する文書・ログなどは保管することが望ましい。

5.5.6 ソフトウェア廃棄

ソフトウェアの破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計画を立案すること。計画立案には、ユーザを参加させること。
- (2) 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
- (3) 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。
- (4) 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。
- (5) 廃棄するソフトウェアまたはコンピュータシステムが使用していたデータまたは廃棄するソフトウェアまたはコンピュータシステム関連のデータをアクセス可能な状態にすること。

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

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) 改訂

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・技術的内容などについて、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 改訂

文書の改訂は定められた手続きに従って行うこと。また、改訂した文書について、様式・技術的内容などについて、点検・承認を行うこと。

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。

- (1) 品目の識別と定義およびベースライン化
- (2) 品目に対する修正依頼についての記録と報告
- (3) 品目の修正とリリース管理
- (4) 品目の履歴検索
- (5) 品目の完全性と正確性の保証
- (6) 品目の保管・取扱い・出荷の管理

6.2.1 プロセス開始の準備

構成管理計画として、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
- (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
- (3) アクティビティを実施する責任を負う組織
- (4) 関係する他の組織

6.2.2 構成識別

管理する品目とそのバージョンに対して、次のものを識別し、定義すること。

- (1) ベースライン
- (2) バージョンの参照
- (3) その他の識別詳細情報

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) 変更依頼の識別と記録
- (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) 安全性またはセキュリティが要求される重大な機能にかかわる品目における、アクセス管理および監査。

6.2.4 構成変更状況の記録

管理すべき品目の状況と履歴を示す管理記録および状況報告を準備すること。

6.2.5 構成変更状況の評価

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を確認すること。

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。

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活動やアウトプットが、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) 製品品質の保証
- (3) プロセスの保証
- (4) 品質システムの保証

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

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 品質保証活動の計画

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) 品質保証に関する組織・体制・責任・教育・訓練
- (5) スケジュール
- (6) レビューに関する一連の手続き
- (7) 品質保証活動の記録の識別と、その収集・保管・維持・廃棄などに関する一連の手続き
- (8) 購買管理・供給業者に対する品質保証に関する要求・活動
- (9) 既存ソフトウェア品目(COTS 品目・再利用ソフトウェア品目)の管理
- (10) 出荷に関する一連の手続き

6.3.2 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出荷時において、そのソフトウェアまたはコンピュータシステムが合意された要求仕様を満足し、取得者が受け入れ可能な形態となっていること。

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。

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・再利用ソフトウェア品目)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)に関しては以下を管理項目に含めること。

- (1) 既存ソフトウェア品目を使用することの利点
- (2) 既存ソフトウェア品目の使用を可能に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から得られるリスク
 - (d) 受け入れおよび保証条件
 - (e) 入手可能または利用可能な関連文書
 - (f) 導入、事前準備、訓練、制約条件

- (g) バージョンなどの識別と構成管理形態手法
- (h) 保守および将来的支援サポート
- (i) 知的財産権 など

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。

6.4 検証プロセス

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

6.4.1 プロセス開始の準備

- (1) 検証の要否・検証組織にどの程度の独立性をもたせるかを決定すること。
- (2) 検証を要すると決定した場合、下記を含む検証プロセスを確立すること。
 - (a) 検証組織を決定し、検証組織に適切な独立性と権限を保証すること。
 - (b) 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、適切なタスクを選択すること。
 - (c) 選択したタスクに基づき、検証を計画すること。
 - (d) 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) プロセスには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などを満たしている。
- (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などを満たしていることを、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・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・トラブルシュート容易性などについて、適切に考慮されている。
- (4) 安全性・セキュリティなど特に注意を要する設計に関して、要求や品目に適用される基準などを満たしていることを、適切な手法により示すことができる。

6.4.2.4 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・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・トラブルシュート容易性などについて、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設計に関して、要求や品目に適用される基準などを満たしていることを、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。

6.5 妥当性確認プロセス

妥当性確認プロセスは、意図した使用目的に合致しているかどうかを確認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) 妥当性確認

6.5.1 プロセス開始の準備

- (1) 妥当性確認の要否・妥当性確認組織にどの程度の独立性をもたせるかを決定すること。
- (2) 妥当性確認を要すると決定した場合、下記を含む妥当性確認プロセスを確立すること。
 - (a) 妥当性確認組織を決定し、妥当性確認組織に適切な独立性と権限を保証すること。
 - (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) 妥当性確認計画に従って妥当性確認を行い、識別された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。

6.5.2 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) 負荷試験、境界および異常値入力試験
 - (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。

※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。

6.6 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、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) レビューの当事者は、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、被験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。

6.7 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、基準類など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

6.8.1 プロセス開始の準備

ソフトウェア製品について問題(気づき)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

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 問題の識別

問題が発生した場合、管理できるよう識別すること。そのさいに優先順位をつけることが望ましい。また、関係者に通知すること。

6.8.2.2 調査/分析

問題の現象、発生条件および原因を調査/分析し、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。

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ければ検討すること。
- (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/影響範囲などを含め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問題の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。

6.8.2.6 記録とモニタ

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

予防が必要なリスクが類推できる場合に、予防措置を実施すること。

6.8.4 傾向分析

問題の傾向分析を行うことが望ましい。

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)

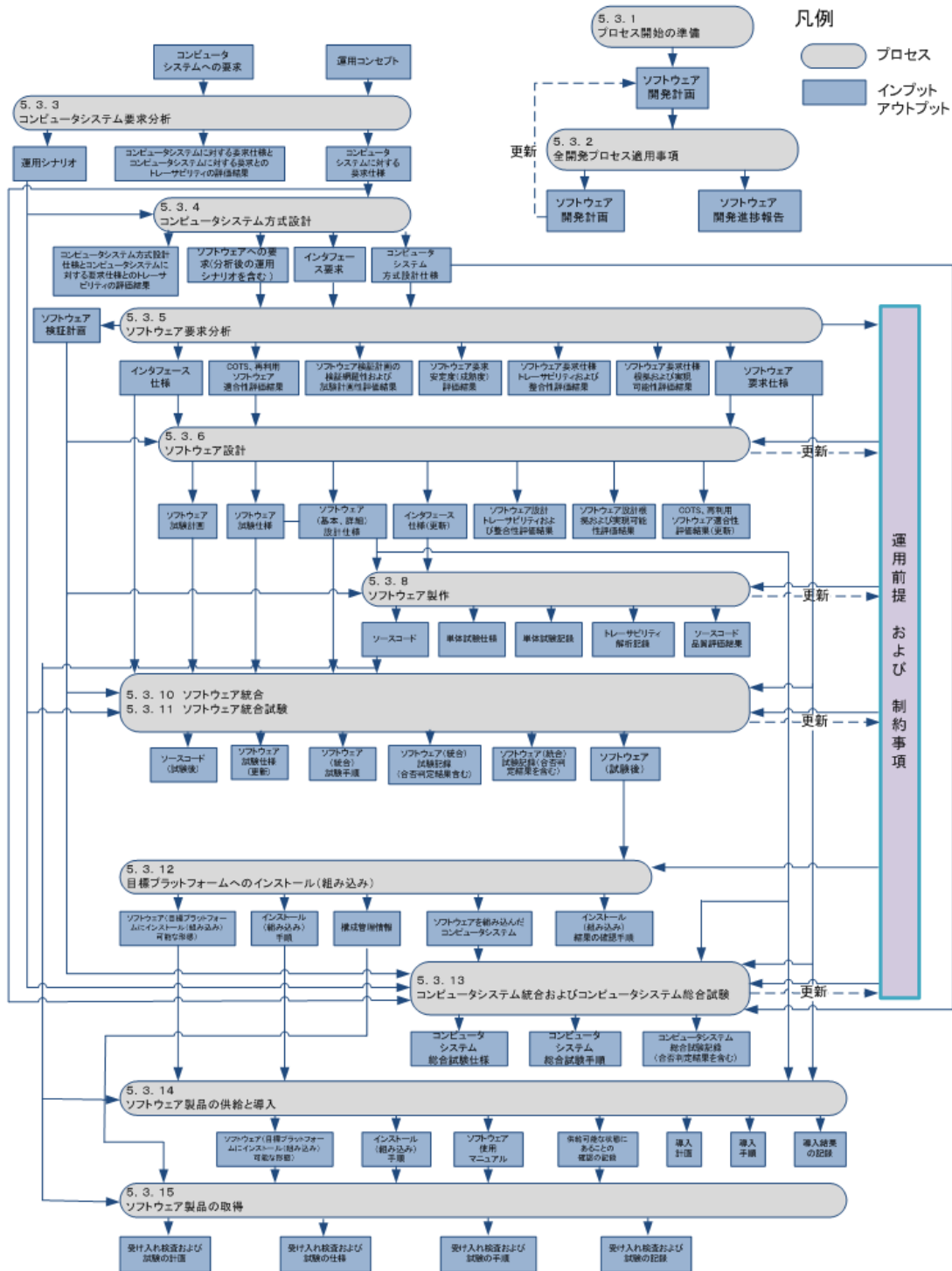


図-付録 1 開発プロセス全体構成の例(ウォーターフォール型)

付録2 インプット・アウトプットと各プロセス対応表

表-付録2 インプット・アウトプットと各プロセス対応表

	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
インプット・アウトプット															
ソフトウェア開発計画	O														
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		
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			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															
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											
ソフトウェア要求仕様				O	I						I		I	I	
インタフェース仕様				O	M		I			I	I				
運用前提および制約事項				O	M		M			M	I	M			
CO/TS、再利用ソフトウェア適合性評価結果				O	M										
ソフトウェア検証計画				O	I		I			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	O										
ソフトウェア要求安定度(成熟度)評価結果				O	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	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	O									
ソフトウェア設計根拠および実現可能性評価結果					O	O									
ソフトウェア試験計画					O	O				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							
ソースコード品質評価結果								O							
ソースコード								O		M	M			M	I
ソフトウェア										O	M	I			
ソフトウェア(目標プラットフォームにインストール(組み込み)可能な形態)												O		M	I
構成管理情報												O			I
インストール(組み込み)手順												O			I
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				
ソフトウェアを組み込んだコンピュータシステム												O	I		
インストール(組み込み)結果の確認手順												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	
ソフトウェア使用マニュアル															O
供給可能な状態にあることの確認の記録															O
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)

付録3 「検証」及び「妥当性確認」に関する補足説明

「検証」及び「妥当性確認」の差異について、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と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又は適用に関する要求事項と確認を行い、“要求とのズレ”の蓄積を小さくすることを目的とする。

表-付録3 検証と妥当性確認の説明

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。
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又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。

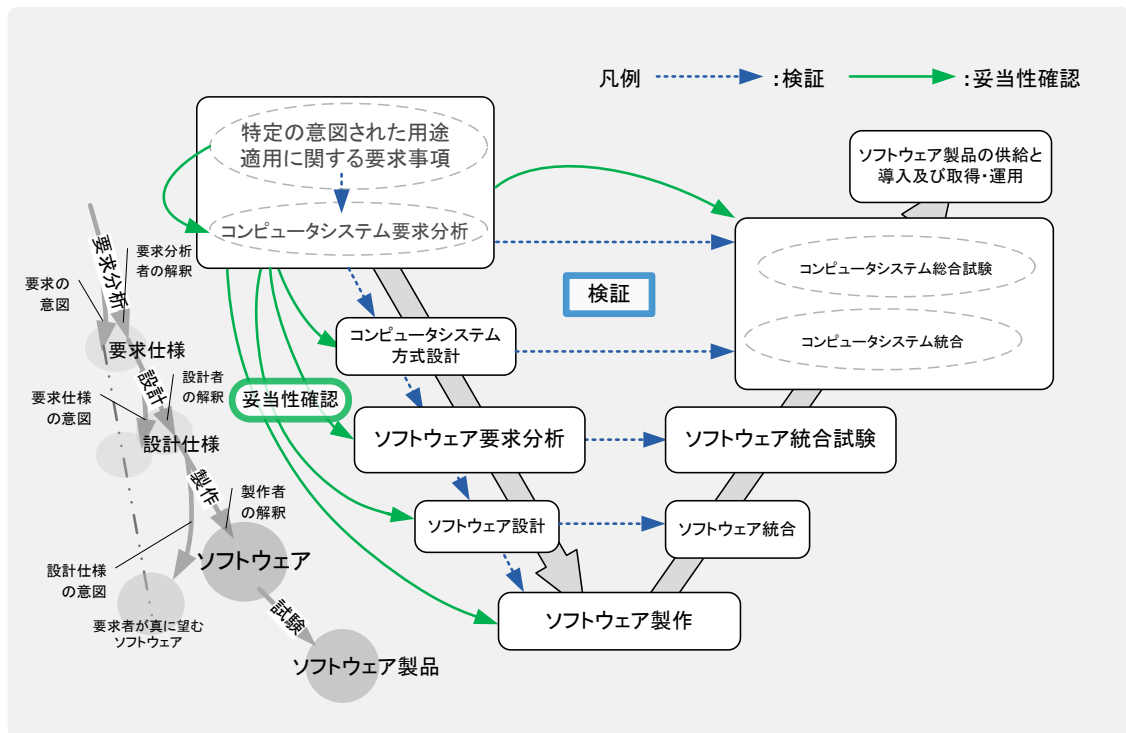


図-付録3 検証と妥当性確認の概念図(例)

付録4 問題解決プロセスと他プロセスとの関係

運用プロセスにおいて、問題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。

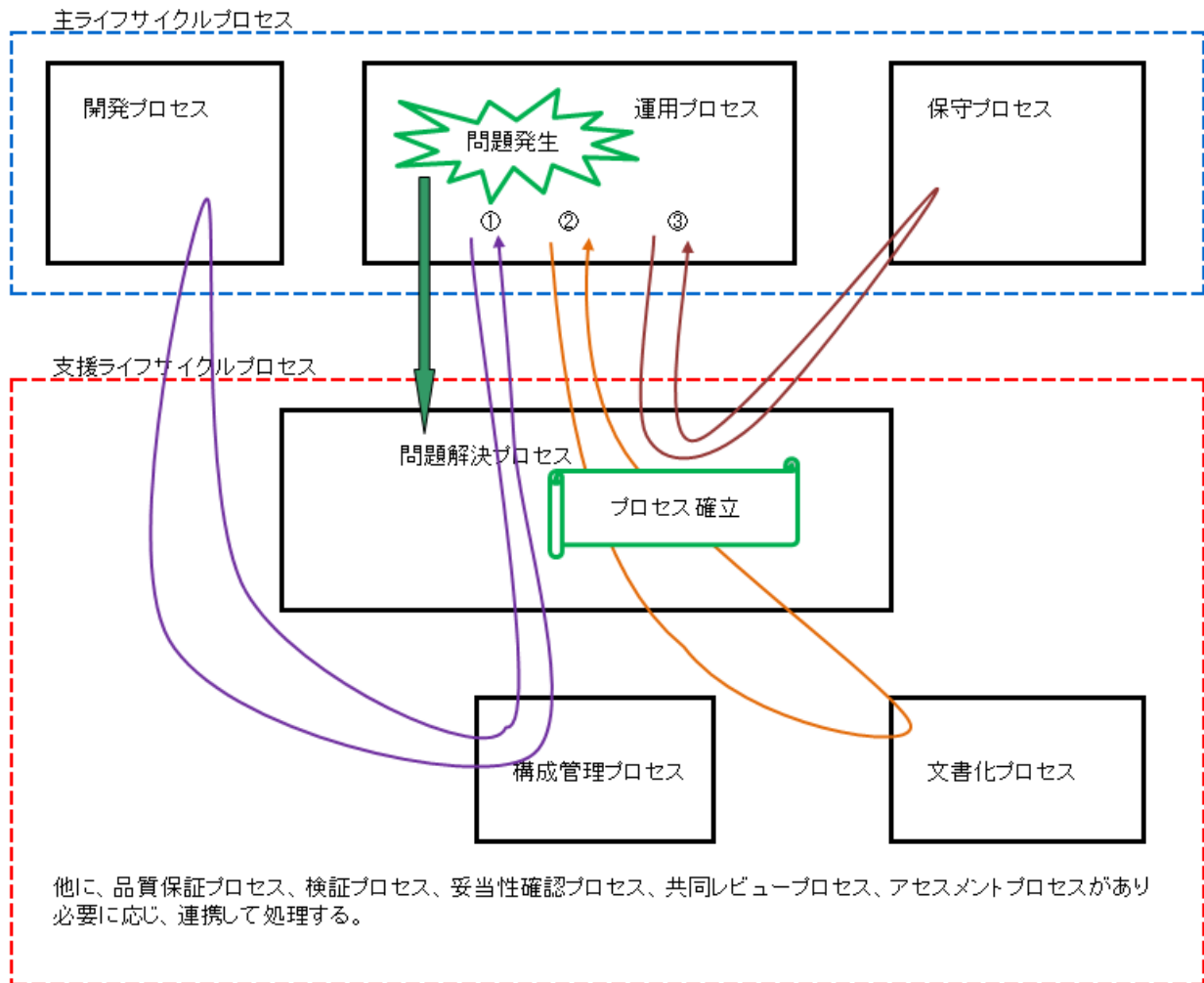


図-付録4 問題が発生した場合の問題解決プロセスとの関係(例)

[① ソフトウェア改修の場合]

運用中に問題が発生する。→問題解決プロセスへ送り検討の結果、障害と判断、その対策(ソフトウェアの修正あり)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]

運用中に問題が発生する。→問題解決プロセスへ送り検討の結果、文書の問題と判明する。→文書化プロセスで修正する。→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。

[③ 運用での対応の場合]

運用中に問題が発生する。→問題解決プロセスへ送り検討の結果、障害と判断、運用での対応(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。