

限定なし

J E R G - 3 - 0 0 3 A



## 地上ソフトウェア開発標準

平成27年4月22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	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.....	2
3.2	本標準で新規に定義した用語.....	2
3.3	JERG-0-049 で定義された用語.....	3
4	一般要求事項.....	6
5	主ライフサイクルプロセス.....	9
5.1	欠番.....	9
5.2	欠番.....	9
5.3	開発プロセス.....	9
5.3.1	プロセス開始の準備.....	10
5.3.2	全開発プロセス適用事項.....	10
5.3.3	システム要求分析.....	11
5.3.4	システム方式設計.....	12
5.3.5	ソフトウェア要求分析.....	13
5.3.6	ソフトウェア設計.....	15
5.3.7	欠番.....	18
5.3.8	ソフトウェア製作.....	18
5.3.9	欠番.....	20
5.3.10	ソフトウェア統合.....	20
5.3.11	ソフトウェア統合試験.....	21
5.3.12	ターゲットマシンへのインストール.....	24
5.3.13	システム統合およびシステム総合試験.....	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取得.....	28
5.4	運用プロセス.....	29

5.4.1	プロセス開始の準備.....	29
5.4.2	運用試験.....	30
5.4.3	運用.....	30
5.4.4	ユーザサポート.....	31
5.5	保守プロセス.....	31
5.5.1	プロセス開始の準備.....	31
5.5.2	問題把握および修正分析.....	32
5.5.3	修正の実施.....	32
5.5.4	ソフトウェアの書き換え.....	32
5.5.5	移行.....	32
5.5.6	ソフトウェア廃棄・リプレース.....	33
6	支援ライフサイクルプロセス.....	34
6.1	文書化プロセス.....	34
6.1.1	プロセス開始の準備.....	35
6.1.2	作成.....	35
6.1.3	発行.....	35
6.1.4	改訂.....	35
6.2	構成管理プロセス.....	36
6.2.1	プロセス開始の準備.....	36
6.2.2	構成識別.....	37
6.2.3	構成変更管理.....	37
6.2.4	構成変更状況の記録.....	38
6.2.5	構成変更状況の評価.....	38
6.2.6	リリース管理と出荷.....	38
6.3	品質保証プロセス.....	39
6.3.1	プロセス開始の準備.....	39
6.3.2	製品品質の保証.....	40
6.3.3	プロセスの保証.....	40
6.3.4	品質システムの保証.....	40
6.4	検証プロセス.....	42

6.4.1	プロセス開始の準備.....	42
6.4.2	検証.....	42
6.5	妥当性確認プロセス.....	44
6.5.1	プロセス開始の準備.....	44
6.5.2	妥当性確認.....	45
6.6	共同レビュープロセス.....	45
6.6.1	プロセス開始の準備.....	46
6.6.2	管理面のレビュー.....	46
6.6.3	技術面のレビュー.....	46
6.7	アセスメントプロセス.....	47
6.8	問題解決プロセス.....	47
6.8.1	プロセス開始の準備.....	48
6.8.2	問題識別.....	48
6.8.3	問題分析と改善／解決.....	48
6.8.4	改善／解決状況の監視.....	49
6.8.5	関係者への周知.....	49
	付録.....	a

## 1 適用範囲

本標準は、地上システム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、本標準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、本標準が準拠しているソフトウェア開発標準（JERG-0-049）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 準拠文書

- (1) JERG-0-049 ソフトウェア開発標準

### 2.2 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) ISO 9000:2000 品質マネジメントシステム—基本および用語
- (7) 共通フレーム 2007 ソフトウェアライフサイクルプロセス SLCP-JCF 2007

## 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-0-049 に準ずるものとする。本標準で再定義した用語、新規に定義した用語を以下に示す。

## 3.1 本標準で再定義した用語

用語	説明
システム	<p>開発する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・ネットワークを含む、ソフトウェア・プラットフォーム・ハードウェアの総体。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、その定義は一意でなければならない。</p> <p>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>

## 3.2 本標準で新規に定義した用語

用語	説明
運用者、ユーザ	<p>運用者：システムを運用する個人または組織。</p> <p>ユーザ：ある機能を果たす運用システムを利用する個人または組織。</p>
外部インタフェース、内部インタフェース	<p>開発対象システムの外側とのインタフェースを外部インタフェース、内側のインタフェースを内部インタフェースとすることを基本とする。システム・サブシステムの内側を内部インタフェース、外部を外部インタフェースとするなどの定義も統一されるならば、使用してよい。</p>
ターゲットマシン	<p>実運用時にソフトウェアをインストールし、稼働させるコンピュータ。</p>
品質システム	<p>品質管理を実施するために必要となる組織構造、手順、プロセスおよび経営資源のこと。</p>
リグレッション試験(回帰試験)	<p>ソフトウェアを変更した際に、その変更によって意図しない結果を引き起こしていないかどうかを検証する再試験。</p>
利用ケース	<p>システムと運用者およびユーザとのやりとりを、運用者およびユーザの立場から記述したものであり、利用・運用コンセプト、利用・運用シナリオが考慮されている。</p>

用語	説明
～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うる試験条件と、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必要がある。ソフトウェア製作プロセスで使用する「単体試験仕様」、ソフトウェア統合試験プロセスで使用する「ソフトウェア統合試験仕様」、システム総合試験プロセスで使用する「システム総合試験仕様」が該当する。
～への要求	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。

### 3.3 JERG-0-049 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。
アセスメント	本標準の要求事項に対する実施状況を確認し、改善項目を識別するプロセスである。
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。
受け入れ検査・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	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、システムの維持などが含まれる。
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	信頼性、使用性、効率性、保守性、移植性など、機能性以外の要求すべて。

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 (ISO 9000)。
構成管理	<p>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>たとえば、ソフトウェアが複数のモジュールから構成されるソフトウェアの場合には、ソフトウェアのバージョンの管理も必要であるが、その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>
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>
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。
試験計画性	試験対象項目に対して、適切な開発プロセス・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。
ソフトウェア	<p>システムの構成部品目的一种であり、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>

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および妥当性確認に関し、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、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、以降の仕様の変更可能性の小ささを示す指標。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。
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、管理的な観点から考慮される。
トレーサビリティ	上位文書との対応。
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。

用語	説明
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、状態の変化に対応するもの、アウトプットや作業の評価などを含む。
要求仕様	<p>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	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、地上システム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、地上システム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。

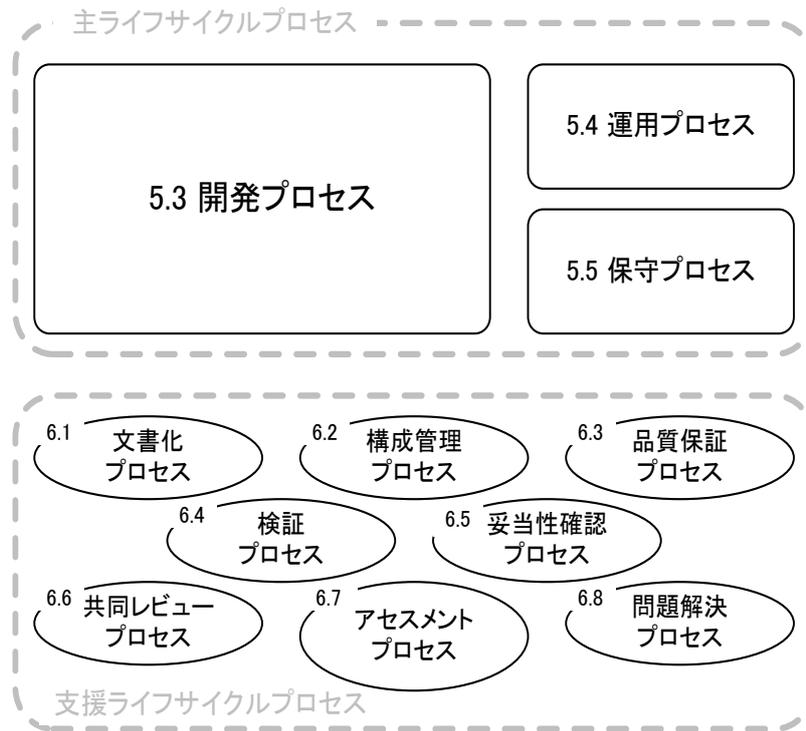


図 4.1 プロセス全体図

各プロセスおよび各アクティビティについては、本標準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	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・ターゲットマシンへのインスト

プロセス		説明
		ール・供給と導入・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・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	製品品質・プロセス・品質システムが、計画にしたがって行われていることを保証するプロセス。
	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	本標準の要求事項に対する実施状況を確認し、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、

システム要求分析プロセス

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

ターゲットマシンへのインストールプロセス

システム統合および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) 開発対象となるシステム・サブシステムの範囲
  - (b) 開発プロセスの定義(ソフトウェアをターゲットマシンにインストールしたシステムに対するアクティビティなども含む)。
  - (c) 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(品質保証プロセスの計画を含む)
  - (d) 開発文書体系の整備と、各開発プロセスにおける入出力の関係
  - (e) 適切な作業分割および作業単位ごとの作業計画・進捗管理の方法(開発スケジュール・担当組織を含む)。
  - (f) ソフトウェア開発・検証で使用する環境(シミュレータ、試験環境など)
  - (g) COTS および再利用ソフトウェア(JAXA が保有するソフトウェアおよびメーカーの保有するソフトウェアを含む)を使用する場合、その管理計画(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
      - ・COTS 品目に対しては、使用実績および使用予定機能に対して事前評価を行うこと。
      - ・再利用ソフトウェア品目に対しては、使用条件、仕様の適合性を評価するための設計情報と検証情報が活用できることを確認すること。
  - (h) 対象システムの他の部分との適合性評価計画
- (2) ソフトウェア開発計画を文書化し、構成管理対象文書として制定すること。

#### 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アクティビティを実施すること。

- (1) 開発の状況に応じてソフトウェア開発計画を更新、管理すること。
- (2) ソフトウェア開発進捗を把握すること。

### 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 システム要求分析

### 5.3.3.1 アクティビティ

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

開発する対象システムへの要求および運用コンセプトを分析し、考えられる利用・運用形態を検討し、対象システムが分担する機能要求、特に性能・品質などの機能外要求、および運用シナリオを抽出すること。

- (2) 要求仕様作成

抽出された要求に基づき、実現可能性および整合性を確認し、対象システムに対する要求仕様、外部インタフェース仕様を定義、文書化すること。また、要求仕様について、その外部システムの接続先組織と相互に内容の解釈を含めて合意を得ること。

なお、要求仕様の根拠を明確化し、対象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) 対象システムへの要求
- (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) 対象システムに対する要求仕様
- (3) 外部インタフェース仕様(上記 (2)に含めてもよい)

- (4) 上位要求に対するトレーサビリティの評価結果

### 5.3.4 システム方式設計

#### 5.3.4.1 アクティビティ

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) 運用シナリオ、対象システムに対する要求仕様および外部インタフェース仕様に基づき、対象システムに対するシステム方式(アーキテクチャ)を設計すること。構成する品目(システム・サブシステム)、開発要素(新規開発、COTS 利用、再利用ソフトウェア品目の改修/流用)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)を明確にし、システム全体の実現可能性を評価すること。設計結果を対象システム方式設計仕様、構成する品目間の内部インタフェース仕様としてまとめること。
- (2) 対象システムに対する要求仕様をシステムの各構成部品への要求として割り当てること。上記(1)で明確にした構成する品目に、対象システムに対する要求仕様から抽出された要求を、漏れなく割り当てること。
- (3) 運用シナリオを分解し、上記(1)で割り当てられたシステムの各構成部品に割り当てること。
- (4) 上記(1)～(3)の設計結果を対象システム方式設計仕様としてまとめるとともに、システム方式設計仕様について、設計根拠と運用前提および制約事項を明らかにし、適切な評価を実施すること。
- (5) 対象システム方式設計仕様の作成にさいし、システムに対する要求仕様など上位要求にあたる対象システムへの要求および運用シナリオとのトレーサビリティを評価すること。
- (6) 上記(1)において、ソフトウェア品目に割り当てられた要求を抽出してソフトウェアへの要求としてまとめること。
- (7) 設計の進捗に合わせて、ターゲットマシンおよび関連設備(ネットワーク、無停電電源装置など)に対する要求および制約事項をとりまとめること。場合によっては、ソフトウェア設計プロセスでまとめてもよい。

#### 5.3.4.2 インプット

- (1) 運用シナリオ
- (2) 対象システムに対する要求仕様
- (3) 外部インタフェース仕様(上記(2)に含めてもよい)

### 5.3.4.3 アウトプット

- (1) 対象システム方式設計仕様
- (2) 内部インタフェース仕様(上記(1)に含めてもよい)
- (3) ソフトウェアへの要求
- (4) 上位要求に対するトレーサビリティの評価結果
- (5) 運用前提および制約事項
- (6) ターゲットマシンの制約事項

### 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、以下の要求を満たすアクティビティを実施すること。

- (1) 対象システム方式設計仕様、内部インタフェース仕様、ソフトウェアへの要求、ターゲットマシンの制約事項を分析し、ソフトウェア要求仕様(特に性能・信頼性要求などの機能外要求を含む)を作成すること。
  - (a) 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。
  - (b) 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(想定データ件数・サイズ、検索速度、保存速度など)を含めること。
  - (c) 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。  
異常処理要求の例：  
異常処理に対する共通要求(ログ記録など)、想定される主要な異常に対する処理要求、想定外の異常が発生した場合の処理要求、インタフェースに関する異常が発生した場合の処理要求などの要求
  - (d) 必要に応じ、セキュリティ要求を分析し、ソフトウェア要求仕様を含めること。
  - (e) 利用ケースの検討を進め、ソフトウェア要求仕様を含めること。
- (2) 内部インタフェース仕様などの仕様を分析し、他のソフトウェア(COTS、再利用ソフトウェア品目の改修/流用を含む)、ハードウェア、人間などのシステムを構成する要素との内部インタフェース仕様を詳細化すること。また、詳細化された内部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。

- (3) ソフトウェア要求仕様に対し、対象システム方式設計仕様・内部インタフェース仕様・ソフトウェアへの要求とのトレーサビリティおよび整合性を評価すること。
- (4) ソフトウェア要求仕様に対し必要に応じ、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。
- (5) 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適合性（COTS、再利用ソフトウェアにより実現する要求項目の識別、妥当性評価など）を評価すること。
- (6) 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し、更新すること。
- (7) ソフトウェア要求仕様、内部インタフェース仕様など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を立案すること。
- (8) ソフトウェア検証計画に対しソフトウェアへの要求・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、内部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。
- (9) ソフトウェア検証計画において、実運用環境とは異なる試験環境での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。
- (10) ターゲットマシンおよび関連設備に対する必要な要求を見直し、それぞれの要求および制約事項に反映すること。
- (11) 必要に応じ、運用シナリオに反映を行うこと。

#### 5.3.5.2 計測活動

ソフトウェア要求分析に対し、計測に関する以下のアクティビティを実施すること。

- (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) 対象システム方式設計仕様
- (2) 内部インタフェース仕様(上記(1)に含めてもよい)

- (3) ソフトウェアへの要求
- (4) 運用前提および制約事項
- (5) ターゲットマシンの制約事項
- (6) 運用シナリオ

#### 5.3.5.4 アウトプット

- (1) ソフトウェア要求仕様
- (2) 内部インタフェース仕様(更新)
- (3) ソフトウェア要求仕様との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(更新)
- (7) ソフトウェア検証計画
- (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) ソフトウェア要求仕様安定度(成熟度)評価結果
- (10) 運用シナリオ(更新)
- (11) ターゲットマシンの制約事項(更新)

#### 5.3.5.5 審査

アウトプットに対して、ソフトウェア要求仕様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(期限、対処状況、影響度などに応じ)について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、特にソフトウェア基本設計およびソフトウェア詳細設計を分割したプロセスとしていないが、実際の開発プロセスでは、必要に応じソフトウェア基本設計、ソフトウェア詳細設計に分割して、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、以下の要求を満たすアクティビティを実施すること。

#### 【ソフトウェア基本設計】

- (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、ソフトウェア設計仕様をまとめること。
- (2) ソフトウェア設計仕様には、ソフトウェア要求仕様に定義された機能外要求(特に処理時間要求、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。
- (3) 内部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、内部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。
- (4) ソフトウェア設計仕様には、画面設計およびHMI (Human Machine Interface) 設計を含めること。画面設計およびHMI 設計については、特に画面遷移、応答速度などの仕様について、そのユーザとの合意を得ること。
- (5) 必要に応じ、利用ケースの検討を進め、ソフトウェア設計仕様および運用シナリオに反映すること。
- (6) 必要に応じ、データベース設計を含めること。

#### 【ソフトウェア詳細設計】

- (7) ソフトウェア機能分割、モジュール分割にしたがい、各モジュール内設計を実施し、ソフトウェア詳細設計を行うこと。
- (8) 内部インタフェース仕様について、モジュール内設計に基づいて詳細化を行うこと。また、内部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。

#### 【ソフトウェア基本設計、ソフトウェア詳細設計 共通】

- (9) ソフトウェア設計とソフトウェア要求仕様・内部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モジュール単位で評価すること。
- (10) 各ソフトウェア設計項目について、必要に応じ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。
- (11) 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。
- (12) 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。

(13) ソフトウェア検証計画にしたがって、ソフトウェア統合試験計画およびソフトウェア統合試験仕様を立案すること。ソフトウェア統合試験仕様には、評価基準を含めること。また、ソフトウェア統合試験仕様は、以下の観点を考慮すること。

(ア) 運用前提および制約事項

(イ) 最大負荷

(ウ) ソフトウェア要求仕様およびソフトウェア設計仕様、内部インタフェース仕様に対する網羅性

(エ) 例外や異常・故障などの事象

(オ) COTS/再利用ソフトウェア品目の対象システムに対する適合性

(カ) 画面設計およびHMI 設計に対する適合性

(キ) 必要に応じ、セキュリティ要求に対する適合性

(14) ターゲットマシンおよび関連設備に対する必要な要求を見直し、それぞれの要求および制約事項に反映すること。

(15) 必要に応じ、セキュリティ要求に合致する設計を行うこと。

(16) 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アクティビティを実施すること。

(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) 内部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
- (6) 運用シナリオ

#### 5.3.6.4 アウトプット

- (1) ソフトウェア設計仕様(ソフトウェア基本設計仕様、ソフトウェア詳細設計仕様に分けてもよい)
- (2) 内部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) ソフトウェア統合試験計画
- (8) ソフトウェア統合試験仕様
- (9) 運用シナリオ(更新)

#### 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) コーディング時に準拠す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内部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こと。
- (5) ソースコードから単体ロードモジュールを作成すること。
- (6) 単体試験仕様を作成し、それに沿って単体試験を実施し実施結果を記録すること。その場合は、網羅性および条件分岐も考慮しなければならない。モジュールを組み合わせて単体試験を行い、確認してもよい。
- (7) ソフトウェア検証計画に基づき、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、またはソースコードレビューを実施し、ソースコードの品質評価を実施すること。
- (8) ソースコードおよびソフトウェア設計仕様のトレーサビリティおよび整合性を解析すること。
- (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アクティビティを実施すること。

- (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管理すること。

### 5.3.8.3 インプット

- (1) ソフトウェア設計仕様
- (2) 内部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(単体試験後)
- (2) 単体ロードモジュール(単体試験後)
- (3) 運用前提および制約事項(更新)
- (4) 単体試験仕様

- (5) 単体試験記録
- (6) ソースコード品質評価結果
- (7) ソースコードおよびソフトウェア設計仕様トレーサビリティおよび整合性評価結果

#### 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、以下の要求を満たすアクティビティを実施すること。

- (1) ソースコード(単体試験後)、もしくは単体ロードモジュール(単体試験後)を用いて、ソフトウェアの統合を行うこと。
- (2) ソフトウェア統合後のベースラインを構成管理プロセス(6.2 参照)により確定すること。
- (3) ソフトウェア統合中のデバック情報(たとえば、コンパイル異常、リンク異常、環境設定に関する問題など)を記録すること。また、そのデバック情報に基づきソースコード、ソフトウェア、環境設定に対して調査、変更を行う必要がある場合、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アクティビティを実施すること。

- (1) ソフトウェア統合中のデバック情報について、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。たとえば、収集データの定義としては、ソフトウェアのリンク中に表示される警告メッセージや警告レベルなどが残留するケースについて考慮すること。評価方法の定義としては、ソフトウェア品質に影響があると想定される

ため許容できないメッセージ内容やソフトウェア統合中のソフトウェア動作などを考慮すること。

- (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) ソフトウェア統合において、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管理すること。

#### 5.3.10.3 インプット

- (1) ソースコード(単体試験後)
- (2) 単体ロードモジュール(単体試験後)

#### 5.3.10.4 アウトプット

- (1) ソースコード(ソフトウェア統合後)
- (2) ソフトウェア(ソフトウェア統合後)
- (3) デバッグ記録

#### 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、以下の要求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統合試験仕様を更新すること。
  - (b) ソフトウェア統合試験は、以下の観点を考慮すること。
    - (ア) 運用前提および制約事項
    - (イ) 最大負荷
    - (ウ) ソフトウェア要求仕様およびソフトウェア設計仕様、内部インタフェース仕様に対する網羅性
    - (エ) 例外や異常・故障などの事象
    - (オ) COTS/再利用ソフトウェア品目の対象システムに対する適合性
    - (カ) 画面設計およびHMI設計に対する適合性

- (キ) 必要に応じ、セキュリティ要求に対する適合性
  - (c) ソフトウェア検証計画、ソフトウェア統合試験計画およびソフトウェア統合試験仕様にしたがい、ソフトウェア統合試験手順を作成すること。
  - (d) ソフトウェア統合試験手順を受け、必要であればソフトウェア統合試験実施に必要な治具や試験用データを整備すること。
- (2) 試験の実施
- (a) ソフトウェア統合試験手順にしたが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、試験の途中段階で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(ソフトウェア統合後)・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) 評価基準に照らして合否判定を行った結果を記録すること。結果を適切に保管するとともに必要に応じ、結果を提示すること。
  - (e) ソフトウェア統合試験において問題が発生した場合、問題解決プロセス(6.8 参照)にしたがって処置すること。必要に応じ、ソフトウェアまたはソフトウェア統合試験仕様の変更要求の分析および評価を行うこと。また、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。
  - (f) ソフトウェア統合試験を実施した結果、新たな運用前提・制約事項が識別された場合、情報を更新すること。

### 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アクティビティを実施すること。

- (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・評価を行う場合には、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管理し、報告すること。

### 5.3.11.3 インプット

- (1) 内部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) ソフトウェア統合試験計画
- (6) ソフトウェア統合試験仕様
- (7) 運用前提および制約事項
- (8) ソフトウェア(ソフトウェア統合後)

### 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) ソフトウェア(ソフトウェア統合試験後)
- (5) ソフトウェア統合試験仕様(更新)

### 5.3.11.5 審査

アウトプットに対して、ソフトウェア統合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 ターゲットマシンへのインストール

#### 5.3.12.1 アクティビティ

ターゲットマシンへのインストールに対し、以下の要求を満たすアクティビティを実施すること。

- (1) ターゲットマシン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) ソフトウェア統合後の構成管理情報にしたがい、既開発品を含めたリリース対象を明確にすること。
- (3) リリースするソフトウェア(ターゲットマシンにインストール可能な形態)を準備し、そのソフトウェアについて構成管理を行うこと。
- (4) リリースするソフトウェアの構成管理情報、ターゲットマシンへのインストール手順を作成すること。インストール手順には、インストール前の事前確認、インストール方法、インストール結果の確認を含むこと。

確認項目の例：

- (a) ターゲットマシンの状態と動作
  - (b) リリースするソフトウェア
  - (c) COTS 品目
  - (d) 環境変数
  - (e) インストールするさいの制約事項(作業時間、新規または更新か、保全関係などを含む)
  - (f) ネットワークの状態
  - (g) ターゲットマシンの制約事項
- (5) インストール手順にしたがい、事前確認、インストール(すでにターゲットマシンにインストールされている場合は省略可能)、インストール結果の確認を実施し、インストール確認結果を記録すること。

### 5.3.12.2 インプット

- (1) ソフトウェア(ソフトウェア統合試験後)
- (2) 運用前提および制約事項
- (3) ターゲットマシンの制約事項

### 5.3.12.3 アウトプット

- (1) ソフトウェア(ターゲットマシンにインストール可能な形態)
- (2) インストール手順
- (3) ターゲットマシンにインストールするさいの制約事項
- (4) ソフトウェアをインストールしたターゲットマシン
- (5) インストール確認結果
- (6) 構成管理情報

## 5.3.13 システム統合およびシステム総合試験

### 5.3.13.1 アクティビティ

システム統合およびシステム総合試験に対し、以下の要求を満たすアクティビティを実施すること。

- (1) 試験準備
  - (a) システム統合として、ソフトウェアをインストールしたターゲットマシンに対して、ネットワークや他システムインタフェースを考慮して、運用環境と同等のシステム総合試験環境を構築すること。
  - (b) ソフトウェア検証計画にしたがい、システム総合試験仕様・システム総合試験手順(合否判定基準含む)を作成すること。なお、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) 最大負荷
    - (ウ) 対象システムに対する要求仕様および対象システム方式設計仕様に対する網羅性
    - (エ) 例外や故障などの異常事象
    - (オ) COTS/再利用ソフトウェア品目の対象システムに対する適合性
    - (カ) セキュリティ要求に対する適合性

- (キ) 外部インタフェース仕様、内部インタフェース仕様に関すること
- (c) 試験準備(試験手順確認など)で見つかった問題点を記録・管理すること。
- (d) 試験準備の段階において、前提となる運用前提を明らかにすること。
- (2) 試験の実施
  - (a) システム総合試験手順にしたがい、試験を実施すること。必要に応じ、試験の途中段階で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。
  - (b) 必要に応じ、シミュレータなどのツールを活用して運用シナリオベースの試験を実施し、運用シナリオに対する検証網羅性を確認すること。
  - (c) システム総合試験のソフトウェア・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) システム総合試験結果を記録し、適切に保管すること。
  - (e) システム総合試験中にソフトウェアまたはシステム総合試験仕様に修正が発生した場合、有効性を評価し必要ときは、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。
  - (f) システム総合試験を実施した結果、新たな運用前提・制約事項が識別された場合、情報を更新すること。

### 5.3.13.2 計測活動

品質を評価できるように、システム統合およびシステム総合試験に対し、計測に関する以下のアクティビティを実施すること。

- (1) 品質指標データの取得
  - (a) 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) システム総合試験準備(試験手順確認など)で発見された問題点は、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・評価を行う場合には、以下を実施すること。

  - (a) 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行うこと。
- (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管理し、報告すること。

#### 5.3.13.3 インプット

- (1) 対象システム方式設計仕様
- (2) ソフトウェア要求仕様
- (3) ソフトウェア設計仕様
- (4) 外部インタフェース仕様
- (5) 内部インタフェース仕様
- (6) ソフトウェア検証計画
- (7) 運用前提および制約事項
- (8) 運用シナリオ
- (9) 対象システムに対する要求仕様

#### 5.3.13.4 アウトプット

- (1) システム総合試験仕様
- (2) システム総合試験手順(合否判定基準を含む)
- (3) 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と導入に対し、以下の要求を満たすアクティビティを実施すること。

- (1) ソフトウェア使用マニュアル(インストールマニュアル、運用制約事項も含む)を作成すること。
- (2) ソフトウェア統合試験が終了していることなど、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) ソフトウェア製品の導入手順を作成すること。なお、必要に応じ、導入計画(既存システムとの置き換えおよびそれまでの一時的な並行運転を含む)を立案すること。

- (4) 導入計画および導入手順に基づいてソフトウェア製品を導入し、その結果を記録すること。
- (5) 識別された保守要求(予備品準備、アップデート情報の収集、リカバリ手順など)を整理し、保守プロセス(5.5 参照)へ引継ぎを実施すること。
- (6) 構成管理情報およびインストール手順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、構成管理情報およびインストール手順を更新すること。

#### 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) ソフトウェア(ターゲットマシンにインストール可能な形態)
- (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(インストールマニュアル、運用制約事項を含む)
- (2) ソフトウェア製品(供給形態)
- (3) 供給可能な状態にあることの確認の記録
- (4) 導入計画
- (5) 導入手順
- (6) 導入結果の記録
- (7) 保守要求
- (8) インストール手順(更新)

### 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

ソフトウェア製品の取得に対し、以下の要求を満たすアクティビティを実施すること。

- (1) 受け入れ検査および試験について、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、取得者がその試験内容の承認を行うこと。
- (2) 上記(1)の計画、仕様、手順にしたがい、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の記録を審査すること。

### 5.3.15.2 インプット

- (1) ソフトウェア使用マニュアル
- (2) ソフトウェア製品(供給形態)
- (3) 供給可能な状態にあることの確認の記録

### 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) 供給可能な状態にあることの確認の記録の審査記録

## 5.4 運用プロセス

以下のソフトウェアまたはシステムの運用およびユーザサポートを行うこと。

- (1) プロセス開始の準備
- (2) 運用試験
- (3) 運用
- (4) ユーザサポート

### 5.4.1 プロセス開始の準備

本プロセスを実施するための体制を含め計画を作成すること。必要に応じ、運用者およびユーザに対する訓練・教育についても計画を作成すること。また、以下の運用に関する規格(運用標準・運用要領)を定めること。

- (1) 運用時の問題管理手続き
- (2) ソフトウェアまたはシステムの運用およびユーザサポートにかかわる作業手順

#### 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)にしたがい、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 運用およびユーザサポートにかかわる作業手順の確立

ソフトウェアまたはシステムを運用環境下で実運用するための運用手順、試験手順、問題報告および修正依頼を保守プロセス(5.5 参照)に渡すための手順、ソフトウェアまたはシステムを実運用にリリースするための移行手順、およびユーザサポートにかかわる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システムがリリースされるごとに計画に従い、運用試験(仮運用、試験運用の形態で実施される場合も含む)を実施すること。実運用に移行するために、該当するソフトウェアまたはシステムが規定されている判定基準を満たすこと。

#### 5.4.3 運用

ソフトウェアまたはシステムの運用に対し、以下の要求を満たすアクティビティを実施すること。

- (1) ソフトウェア使用マニュアルおよび規格(運用標準、運用要領)にしたがって所定の環境において運用すること。
- (2) ソフトウェアまたはシステムへの問題点にかかわるデータを収集し、記録すること。ログデータやパフォーマンスをモニタし、障害の前兆を発見した場合には、事前防止措置をとること。
- (3) COTS の不具合情報/バージョンアップ/保守サービス提供状況をモニタし、ソフトウェアの更新の必要性を検討すること。
- (4) 運用において問題が識別された場合、問題解決プロセス(6.8 参照)にしたがい処理すること。必要に応じ、保守プロセス(5.5 参照)などの関連プロセスに引き渡すこと。また、問題が解決するまで、問題解決状況を監視すること。
- (5) 適用されるセキュリティ標準やセキュリティポリシーなどのセキュリティ要求がある場合、それにしたがうこと。不正な利用や不正な改変がないかログデータなどの情報をモニタすること。
- (6) ソフトウェア・データ・設定などの情報について、対象、重要度、採取タイミング、リストア時制約などの運用条件を考慮しバックアップを計画・実施すること。なお、バックアップを実施するさいには、ソフトウェアの利用の妨げとならないよう計画し、ユーザへの周知を行うこと。

#### 5.4.4 ユーザサポート

ユーザサポートに対し、以下の要求を満たすアクティビティを実施すること。

- (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、必要に応じ問題解決プロセス(6.8 参照)にしたがい処理すること。必要に応じ、保守プロセス(5.5 参照)などの関連プロセスに引き渡すこと。問題解決を行う場合、問題(解決課題)の明確化・解決の選択肢・解決の計画(解決方法の決定や解決までのスケジュールを含む)を、その問題に関してサポートを要求したユーザに報告すること。また、問題が解決するまで、問題解決状況を監視すること。
- (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。
- (4) 必要に応じ、運用者、ユーザへの教育を実行すること。

#### 5.5 保守プロセス

保守対象であるソフトウェア製品(ソースコードおよび関連文書)の修正およびCOTSに対し、以下のアクティビティを行うこと。

- (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・リプレース

##### 5.5.1 プロセス開始の準備

本プロセスを実施するために以下を含め、計画および手順を作成すること。なお、保守プロセスの開始時期は運用開始後ではなく、ソフトウェア製品の導入から対象とすること。

- (1) 問題報告および修正依頼の受理、記録、解決、追跡およびそれらの状況の通知を問題解決プロセス(6.8 参照)にしたがい処理すること。
- (2) 現行システムの修正作業を管理するために、構成管理プロセス(6.2 参照)にしたがい処理すること。

### 5.5.2 問題把握および修正分析

問題把握および修正分析に対し、以下の要求を満たすアクティビティを実施すること。

- (1) 修正について以下の観点で分析すること。
  - (a) 修正の目的／分類(訂正、改良、予防、新たな環境への適合など)
  - (b) 適用範囲(修正の規模、必要なコスト、修正時間など)
  - (c) 致命度(性能、安全性またはセキュリティへの影響など)
- (2) 分析に基づき、修正方法を検討すること。修正方法については複数検討し、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) 修正理由・修正に関する分析結果・修正方法を記録すること。
- (4) 修正について承認を得ること。

### 5.5.3 修正の実施

修正の実施に対し、以下の要求を満たすアクティビティを実施すること。

- (1) 修正するソフトウェア製品とそのバージョンを決定し、修正を実施すること。
- (2) 修正したソフトウェア製品またはシステムの試験を行い、試験結果を記録すること。なお、修正内容および修正の影響する範囲を分析し、修正していない部分の試験範囲としては、修正により影響を受ける部分のみに限定してもよい。
- (3) 修正を承認する権限をもつ組織と共同でレビューを行い、修正されたソフトウェアあるいはシステムの修正が過不足なく適切に行われていることを確認し、承認を受けること。
- (4) 必要に応じ、開発プロセス(5.3 参照)など該当するプロセスにしたがい実施すること。

### 5.5.4 ソフトウェアの書き換え

地上システムでは適用しない。

### 5.5.5 移行

このプロセス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の他、5.3.14の導入計画立案時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) 移行の実施(必要に応じ、並行運用を含む)
- (5) 移行結果の確認
- (6) 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・ログなどの記録は保管することが望ましい。

#### 5.5.6 ソフトウェア廃棄・リプレース

ソフトウェア廃棄・リプレースに対して、以下の要求を満たすアクティビティを実施すること。

- (1) 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への移行あるいは新しいシステムへの切替え(リプレース)を円滑にするために、新旧ソフトウェアまたは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) 予定した廃棄時期となったとき、関係者全員に通知すること。関連するすべての開発文書、ログおよび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、かつ、機能させる。

### 6.1 文書化プロセス

文書化プロセスは、文書の作成・発行・改訂に関するプロセスであり、ソフトウェアまたはシステムのライフサイクルを通じて作成する文書を対象とする。なお、構成管理を行う文書は、構成管理プロセス(6.2 参照)にしたがうものとする。

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) 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) 文書作成で適用する文書作成標準/要領を準備すること。

### 6.1.2 作成

文書の作成は定められた手続きにしたがって行うこと。下記について考慮すること。

- (1) 文書は文書作成標準/要領などの規約にしたがって作成すること。
- (2) 設計に使用する文書類は必ず文書番号(識別子)を付すこと。
- (3) 作成にあたり使用する文書は正規配付されたものであることを確認すること。
- (4) 作成した文書はあらかじめ定められた手続きにしたがっていること、文書作成標準にしたがっていること、技術的内容が適切であることを点検すること。
- (5) 作成した文書はあらかじめ定められた手続きにしたがい、承認を受けたのち、発行すること。

### 6.1.3 発行

文書の発行は定められた手続きにしたがって行うこと。下記について考慮すること。

- (1) 文書化計画にしたがい適切な時期に文書を発行すること。なお、文書は紙・電子または他の適切な媒体を使用すること。文書を発行する場合には、あらかじめ定められた配付先に配付すること。
- (2) 文書の原本は、関連する要求事項(セキュリティ管理・バックアップなど)にしたがい保管すること。

### 6.1.4 改訂

文書の改訂は定められた手続きにしたがって行うこと。下記について考慮すること。

- (1) 文書を改訂する場合には、必ず改訂符を付して識別した上で、再発行すること。
- (2) 文書を改訂する場合には、改訂理由などの情報を記述し、変更箇所がわかるようにすること。
- (3) 文書を改訂したときには、あらかじめ定められた配付先に配付すること。

- (4) 改訂前の文書を履歴管理などの目的で保管する場合には、誤って使用されないように旧版であることの識別を行うこと。
- (5) 文書を改訂する場合には、あらかじめ定められた手続きにしたがい承認すること。

## 6.2 構成管理プロセス

構成管理プロセスは、ソフトウェアの構成が正しいこと、変更が正しく行われていることを管理するプロセスであり、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

プロセスを開始するにあたり、以下を含む構成管理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

構成管理として実施するすべてのアクティビティについて目的とその内容を明確にすること。

構成管理のアクティビティ例：

技術変更、デビエーション、ウエーバなど

- (2) アクティビティを実施するための手続きとタイミング

各アクティビティを実施するための手続きを文章や図を用いて明確にすること。また、実施するタイミング(頻度、マイルストーン、条件など)を明確にすること。

- (3) アクティビティを実施する責任を負う組織および関係する他の組織

各アクティビティを実施する責任を負う組織および関係する他の組織を明確にすること。また、それぞれの組織の活動内容もあわせて明確にし、すべてのアクティビティがどこかの組織に割り当てられていることを確認すること。

## 6.2.2 構成識別

構成識別のために以下を実施すること。

### (1) 構成管理品目と管理単位の決定

ソフトウェア製品に対して構成管理の対象となる品目と管理単位を決定すること。また、品目と管理単位が一意に特定できるように識別子(名称や番号)を与えること。

構成管理品目の例：

ソフトウェア、ソースコード、ロードモジュール、データ(メイクファイル、リンクマップ、データファイル(環境変数を含む)、データベース)、関連文書(要求仕様、設計書)、電子データを格納した媒体など

### (2) バージョン管理体系の定義

構成管理品目と管理単位の変更履歴がわかるバージョン体系を定義すること。

### (3) ベースラインの確立

上記(1)および(2)で定義した構成管理品目および管理単位の中からベースラインとするものを決定すること。また、ベースラインを確立する時期を明確にすること。

## 6.2.3 構成変更管理

構成変更管理に対し、次のことを実施すること。

### (1) 変更依頼の識別と記録

構成管理品目の識別、検索および追跡を行えるようにするため、審査、試験、出荷、修正などのイベントごとに対象となるソフトウェア製品の構成およびバージョンを識別し、その記録を実施時期とともに管理すること。特に、修正にかかわる記録には修正理由と修正対象、承認者を明記し、追跡可能としておくこと。

### (2) 変更内容の分析と評価

変更内容に対して影響範囲および影響度(期間、コスト、実現可能性など)を分析すること。また、その分析結果について関係者により妥当であるか評価を行うこと。なお、分析は必要に応じ問題解決プロセス(6.8 参照)にて実施すること。

### (3) 修正の可否(承認または不承認)

分析結果の評価を基に修正可否を判断(承認または不承認)すること。判断にさいしては構成管理対象の識別と変更を決定する責任者が行うこと。

#### (4) 修正された品目の実装と検証およびリリース

ソフトウェア製品を修正する場合、分析により特定された影響範囲のすべてについて、品目のベースラインを修正し、修正が適切であることを検証(動作検証も含む)した上でリリースを行うこと。

なお、安全性またはセキュリティが要求される重要な機能にかかわる品目においては、構成管理にかかわるすべてのアクセスを管理し、監査すること。

### 6.2.4 構成変更状況の記録

構成管理品目の状況と構成管理履歴を示す管理記録を残すこと。

### 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および物理的な完全性を確認することを目的に、ソフトウェア品目の構成状況について、ソフトウェアライフサイクル中に発生した修正を以下の観点で評価すること。

- (1) 誤った品目が混入していないか
- (2) 意図していない品目がリリースされていないか
- (3) 修正されるべき文書はすべて反映がなされているか
- (4) ベースラインとされた仕様に基づいて製作されたコードになっているか
- (5) 必要とされる検証(リグレッション試験など)が行われているか

### 6.2.6 リリース管理と出荷

リリース管理と出荷に対し、次のことを実施すること。

- (1) ソフトウェア製品のリリースおよび出荷は、あらかじめ手続きを定め、その手続きにしたがって作業管理すること。
- (2) リリースするさいには、ソフトウェア製品がウィルスに感染していないことを確認すること。
- (3) リリースまたは出荷したソースコードおよび提出文書についてソフトウェア(またはシステム)のライフサイクルを通じて、保守すること。
- (4) 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、保管、包装、出荷するさいに、組織の方針にしたがって取り扱うこと。

## 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システム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したが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管理し、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

プロセスを開始するにあたり、以下を含む品質保証計画を立案すること。

- (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) 共同レビューなど品質保証活動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・廃棄などの作業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

製品品質の保証に対し、次のことを実施すること。

- (1) ソフトウェア製品またはシステム、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を保証すること。
- (2) ソフトウェア製品またはシステムの出荷時において、そのソフトウェアまたは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

プロセスの保証に対し、次のことを実施すること。

- (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したが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。

このアクティビティは、次のタスクからなる。

#### 6.3.4.1 教育・訓練

対象となるソフトウェア製品またはシステムの開発・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、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

購買管理・供給業者管理に対し、次のことを実施すること。

(1) 購買管理

組織における購買管理方針、規則にしたがい、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) バージョンなどのソフトウェア品目の識別と構成管理手法
- (h) 保守および将来的支援サポート
- (i) 知的財産権などの権利関係

#### 6.3.4.5 取扱、保管およびラベリング

ソフトウェアまたはシステムの取扱いおよび保管を確実に実施するために、次の項目を含む必要事項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、規定要求事項を満たしていることを確認するプロセスである。なお、検証プロセスに、試験／レビュー／解析などの確認作業を含めてもよい。

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) 検証

#### 6.4.1 プロセス開始の準備

プロセスを開始するにあたり、以下を含むアクティビティを立案すること。

- (1) 検証を実行する組織にどの程度の独立性と権限をもたせるかを決定し、保証すること。
- (2) 検証の対象とする作業およびソフトウェア品目を決定し、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。
- (3) 選択したタスクに基づき、検証計画を作成すること。検証計画には、検証の対象とする作業およびソフトウェア品目に適用するタスク、組織、およびスケジュールを定義すること。

#### 6.4.2 検証

検証計画にしたがって検証を行うこと。識別された問題点を問題解決プロセス(6.8 参照)に送ること。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

次に示す内容を考慮して、プロセスを検証すること。

- (1) 選択されたプロセスが適切であること。
- (2) ソフトウェア開発計画のプロセスの計画が適切であること。
- (3) プロセスに関する適切な基準・環境が準備されていること。

- (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こと。
- (5) プロセスがソフトウェア開発計画にしたがって適切に実行されていること。

#### 6.4.2.2 要求の検証

次に示す内容を考慮して、要求を検証すること。

- (1) 要求に一貫性があり、実現可能かつ検証可能であること。
- (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こと。(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、標準が満たされていること。
- (4) セキュリティなど特に注意を要する要求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こと。

#### 6.4.2.3 設計の検証

次に示す内容を考慮して、設計を検証すること。

- (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あること。
- (2) 入出力インタフェース・タイミング・計算機リソース(特にメモリ容量・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こと。
- (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こと。
- (4) 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、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こと。

#### 6.4.2.4 ソースコードの検証

次に示す内容を考慮して、ソースコードを検証すること。

- (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あること。
- (2) 入出力インタフェース・タイミング・計算機リソース(特にメモリ容量・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こと。
- (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こと。

- (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こと。
- (5) ソースコードは、定められたコーディング規約などの基準を満たしていること。

#### 6.4.2.5 統合の検証

次に示す内容を考慮して、統合を検証すること。

- (1) ソフトウェアを構成する品目やデータが適切で、正しいバージョンであること。
- (2) ソフトウェアを構成する品目やデータが過不足なく、完全に、正確にソフトウェアとして統合されていること。
- (3) ソフトウェア統合が、計画にしたがって実施されていること。

#### 6.4.2.6 文書化の検証

次に示す内容を考慮して、文書化を検証すること。

- (1) 文書の内容が正しく、一貫性があること。
- (2) 文書化が計画され、文書が適切な時期に発行されていること。
- (3) 構成管理が必要となる文書は、構成管理プロセス(6.2 参照)にしたがっていること。

### 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

プロセスを開始するにあたり、以下を含むアクティビティを立案すること。

- (1) 妥当性確認の要否を決定すること。
- (2) 妥当性確認を実施する組織にどの程度の独立性と権限をもたせるかを決定し、保証すること。

- (3) 妥当性確認を計画する。計画には、次の項目を含めること。
  - (a) 妥当性確認を必要とする品目
  - (b) 実行する妥当性確認タスク
  - (c) 妥当性確認のための資源、責任および予定
  - (d) 妥当性確認報告を配付する手続き

## 6.5.2 妥当性確認

妥当性確認計画にしたがって妥当性確認を行い、識別された問題点を問題解決プロセス(6.8 参照)に送ること。

- (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) 試験要求事項および試験ケースがソフトウェアまたはシステムが想定する使用状況・使用方法に対応していることを確認すること。
- (3) 必要に応じ次の観点を含めて、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(故障時の機能縮退や異常時の運用対処など)
  - (c) ユーザを想定した業務利用試験
- (4) ソフトウェアまたはシステムが、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。

## 6.6 共同レビュープロセス

共同レビュープロセスは、合意の目標に対する進捗の共通理解および成果が利害関係者を満足させていることを確認するためになされる成果物の確認作業である。

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) 管理面のレビュー
- (3) 技術面のレビュー

### 6.6.1 プロセス開始の準備

プロセスを開始するにあたり、以下を含むアクティビティを立案すること。

- (1) ソフトウェア開発計画などの計画であらかじめ計画された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信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、ソフトウェア製品およびツールを含むこと。
- (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点を記録し、必要に応じ問題解決プロセス(6.8 参照)に引き渡すこと。
- (5) レビュー結果を関係者に配付すること。
- (6) すべての当事者は、指摘された問題点に関し、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 管理面のレビュー

ソフトウェア開発状況をソフトウェア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評価し管理すること。また、計画に沿った開発完了が困難と推測された場合、ソフトウェアの仕様見直し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 技術面のレビュー

技術面のレビューにあたり、以下の要求を満たす活動を実施すること。

- (1) ソフトウェア品目を技術的な観点から評価し、要求仕様・標準を満たすソフトウェアの実現に対するリスクを明確化すること。また、要求仕様・標準を満たすソフトウェアの実現が困難と推測された場合、ソフトウェアの仕様見直しやハードウェアによる実現を含めた、要求仕様・標準の適用方法の見直しを検討すること。

- (2) 各技術レビュー完了後にレビュー記録を作成するとともに、技術レビュー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、技術レビューの品質評価を行うこと。

## 6.7 アセスメントプロセス

アセスメントプロセスは、本標準の要求事項に対する実施状況を確認し、改善項目を識別する活動プロセスである。

このプロセスは、次のアクティビティからなる。

- (1) アセスメント実施責任者(以下、スポンサーという)の指示にしたがい、アセスメントを計画し、実施する手順を文書に定め、それに基づいて実施すること。
- (2) アセスメントを実施するにあたって、次の内容を明確にすること。
  - (a) アセスメントの目標・ゴール
  - (b) 事業目標と改善のゴール
  - (c) アセスメントモデルとアセスメント手順
  - (d) アセスメントの範囲と対象プロセス
- (3) 本標準または本標準の要求事項に関わる規定に基づいて、対象プロセスの実施状況を確認し、開発の実施状況や開発の成果物を基に評価すること。また要改善事項を識別すること。
- (4) アセスメントは、アセスメントモデルとアセスメント手法および適用する標準、基準類に精通した者を含むアセスメントチーム(チームリーダー、アセスメント実施者)を結成し、実施すること。
- (5) 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。

## 6.8 問題解決プロセス

問題解決プロセスは、ソフトウェアのすべてのライフサイクルプロセスの実行過程で発見された問題を分析し、改善/解決するためのプロセスである。

本プロセスの目的は、発見されたあらゆる問題に対し、問題の分析・改善/解決・傾向把握のため、適応可能な対処法を提供することである。

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) 問題識別

- (3) 問題分析と改善／解決
- (4) 改善／解決状況の監視
- (5) 関係者への周知

#### 6.8.1 プロセス開始の準備

本プロセスを実施するための手順を準備すること。

- (1) 以下の事項を含めた手順を定めること。
  - (a) 発見された問題を本プロセスに取り込むこと。
  - (b) 関係者に問題の存在を通知すること。
  - (c) 問題を分析し、原因を特定して解決策を決定すること。解決策には、処置実施後の確認方法を含むこと。
  - (d) 適切な作業指示および技術指示を行い、決定された解決策を実施すること。
  - (e) 状況を追跡し、記録を作成・維持して適時、関係者に報告すること。
- (2) 本プロセス実施の各手順における組織間および組織内の責任および権限を定めること。
- (3) 問題の種類、重要度および緊急度など、分類の基準を定めること。
- (4) 問題を把握してから処置完了までの記録・報告形式を定めること。

#### 6.8.2 問題識別

各プロセスの実施の過程で見つかった様々な問題については、本プロセスに引き渡され、あらかじめ定められた管理手順に則り、識別、記録されること。

#### 6.8.3 問題分析と改善／解決

識別された問題について、改善／解決の必要度合いの評価と本質的な原因の分析を行い、必要度合いに応じて実施可能な改善／解決策を講じること。

- (1) 問題解決策を決定するに先立って、原因および発生メカニズムを究明し、製品品質に与える影響を評価し、適切な解決策を決定するための調査・解析を実施すること。

また、問題の重要度に応じて背後要因分析を実施し、根本原因を明らかにして是正処置に反映すること。

- (2) 問題の内容および重要度に応じた参加メンバによる組織体を設置し、問題の内容を確認して解決策を決定すること。また、必要に応じ暫定処置の実施を決定すること。これらは記録し、取られる解決策にかかわる組織に確実に周知すること。
- (3) 問題解決のために技術変更が必要となった場合は、構成管理プロセス(6.2 項参照)にしたがい処置すること。

また、識別された複数の問題から傾向分析などの分析を行い、可能であれば予測される問題に対して予防的な改善／解決策を講じること。

#### 6.8.4 改善／解決状況の監視

実施した改善／解決策について、決定された是正処置ならびに予防処置が継続的に実施されているか、期待した効果が得られているかなどの状況の監視を処置決定時に定めた方法で行うこと。

#### 6.8.5 関係者への周知

識別された問題や分析結果、実施する改善／解決策、改善／解決状況などの情報について、適時、関係者にあらかじめ定められた形式にて周知すること。

問題報告は、問題の発見から調査・分析、原因究明、解決ならびに、問題を横断する傾向の発見に用いることができる記述レベルであること。なお、一覧表程度の管理でよい場合は、品質指標の取得などに影響を与えない範囲で簡素化した形式としてもよい。また、重大な品質問題の場合は速やかに取得者に周知すること。

付録

付録1 開発プロセス(例)

ウォーターフォールモデル型の開発プロセスの流れおよび主なアウトプットの例を示す。

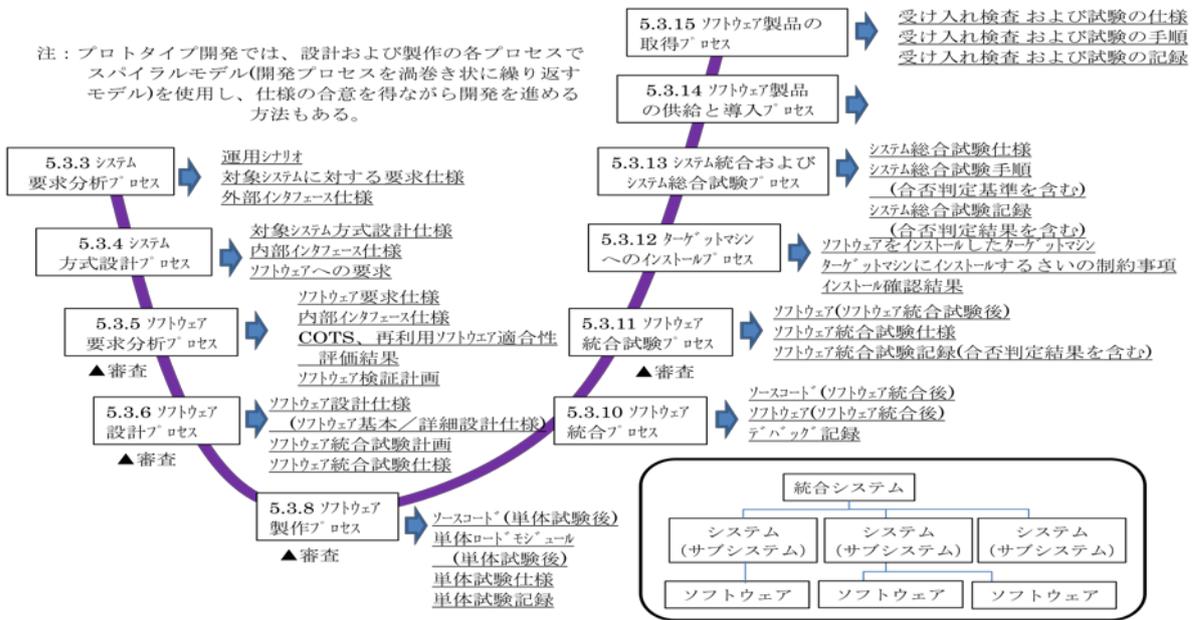


図-付録1 地上システムにおける開発プロセス(例) (ウォーターフォールモデル型)

表-付録1 各開発プロセスにおける主なインプットおよびアウトプット(例)

開発プロセス	インプット(主なもの)	アウトプット(主なもの)
5.3.3 システム要求分析プロセス	・対象システムへの要求 ・運用コンセプト	・運用シナリオ ・対象システムに対する要求仕様 ・外部インタフェース仕様
5.3.4 システム方式設計プロセス	・運用シナリオ ・対象システムに対する要求仕様 ・外部インタフェース仕様	・対象システム方式設計仕様 ・内部インタフェース仕様 ・ソフトウェアへの要求
5.3.5 ソフトウェア要求分析プロセス	・対象システム方式設計仕様 ・内部インタフェース仕様 ・ソフトウェアへの要求 ・運用シナリオ ・運用前提および制約事項	・ソフトウェア要求仕様 ・内部インタフェース仕様 ・COTS、再利用ソフトウェア適合性評価結果 ・ソフトウェア検証計画
5.3.6 ソフトウェア設計プロセス	・ソフトウェア要求仕様 ・COTS、再利用ソフトウェア適合性評価結果 ・ソフトウェア検証計画 ・内部インタフェース仕様 ・運用シナリオ	・ソフトウェア設計仕様 (ソフトウェア基本/詳細設計仕様) ・ソフトウェア統合試験計画 ・ソフトウェア統合試験仕様
5.3.8 ソフトウェア製作プロセス	・ソフトウェア設計仕様 ・内部インタフェース仕様 ・運用前提および制約事項	・ソースコード/単体ロードモジュール(単体試験後) ・単体試験仕様 ・単体試験記録
5.3.10 ソフトウェア統合プロセス	・ソースコード/単体ロードモジュール(単体試験後)	・ソースコード(ソフトウェア統合後) ・ソフトウェア(ソフトウェア統合後) ・デバッグ記録
5.3.11 ソフトウェア統合試験プロセス	・内部インタフェース仕様 ・ソフトウェア設計仕様 ・ソフトウェア要求仕様 ・運用前提および ・ソフトウェア統合試験計画 制約事項 ・ソフトウェア統合試験仕様	・ソフトウェア(ソフトウェア統合試験後) ・ソフトウェア統合試験仕様 ・ソフトウェア統合試験記録(合否判定結果を含む)
5.3.12 ターゲットマシンへのインストールプロセス	・ソフトウェア(ソフトウェア統合試験後) ・運用前提および制約事項 ・ターゲットマシンへの制約事項	・ソフトウェアをインストールしたターゲットマシン ・ターゲットマシンにインストールするさいの制約事項 ・インストール確認結果
5.3.13 システム統合およびシステム総合試験プロセス	・対象システム方式設計仕様 ・内部インタフェース仕様 ・外部インタフェース仕様 ・ソフトウェア要求仕様 ・運用シナリオ ・ソフトウェア設計仕様	・システム総合試験仕様 ・システム総合試験手順(合否判定基準を含む) ・システム総合試験記録(合否判定結果を含む)
5.3.14 ソフトウェア製品の供給と導入プロセス	・ソフトウェア要求仕様 ・インストール手順 ・ソフトウェア設計仕様 ・ソフトウェア(ターゲットマシンにインストール可能な形態)	・ソフトウェア製品(供給形態) ・ 導入結果の記録 ・ソフトウェア使用マニュアル (インストールマニュアル、運用制約事項を含む)
5.3.15 ソフトウェア製品の取得プロセス	・ソフトウェア製品(供給形態) ・ソフトウェア使用マニュアル	・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 ・受け入れ検査および試験の記録