



ロケット搭載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
1.1 適用対象ソフトウェア	1
1.2 他の要求事項との関係	1
2 関連文書	2
2.1 準拠文書	2
2.2 適用文書	2
2.3 参考文書	2
3 用語、定義および略語	2
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	3
3.3 JERG-2-600 および JERG-0-049 で定義された用語	4
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 システム検討プロセス	8
5.3.1 システム要求分析	8
5.3.2 システム方式設計	8
5.4 ソフトウェア開発プロセス	9
5.4.1 プロセス開始の準備	11
5.4.2 ソフトウェア開発管理	11
5.4.3 ソフトウェア要求分析	12
5.4.4 ソフトウェア設計	14
5.4.5 欠番	15
5.4.6 ソフトウェア製作	15
5.4.7 欠番	16
5.4.8 ソフトウェア統合	16
5.4.9 ソフトウェア統合試験(CSCI 試験)	17
5.4.10 目標プラットフォームへのインストール(組込み)	18

5.4.11	ソフトウェア製品の供給と導入	19
5.4.12	ソフトウェア製品の取得	20
5.5	システム検証プロセス	21
5.5.1	システム試験	21
5.5.2	運用試験	22
5.6	号機毎製作・打上げ運用プロセス	23
5.6.1	ソフトウェアの号機毎製作	24
5.6.2	ソフトウェアの取得	24
5.6.3	ソフトウェアの使用	24
5.6.4	ユーザサポート	24
5.6.5	ソフトウェアの書き換え	24
5.7	保守プロセス	24
5.7.1	プロセス開始の準備	24
5.7.2	ソフトウェア製品および開発環境の変更	25
5.7.3	移行	25
5.7.4	ソフトウェア廃棄	25
6	支援ライフサイクルプロセス	26
6.1	文書化プロセス	26
6.1.1	プロセス開始の準備	26
6.1.2	作成	26
6.1.3	発行(配付)	26
6.1.4	維持	26
6.1.5	取得者による承認など	27
6.2	構成管理プロセス	27
6.2.1	プロセス開始の準備	27
6.2.2	構成識別	27
6.2.3	構成変更管理	27
6.2.4	構成管理状況評価	28
6.2.5	リリース管理と出荷	28
6.3	品質保証プロセス	28
6.3.1	プロセス開始の準備	28

6.3.2	製品の保証	29
6.3.3	プロセスの保証	29
6.3.4	品質システムの保証	29
6.4	検証プロセス	31
6.4.1	プロセス開始の準備	31
6.4.2	検証	31
6.5	妥当性確認プロセス	32
6.5.1	プロセス開始の準備	32
6.5.2	妥当性確認	32
6.6	レビュープロセス	33
6.6.1	プロセス開始の準備	33
6.6.2	管理面のレビュー	33
6.6.3	技術面のレビュー	33
6.7	アセスメントプロセス	34
6.7.1	プロセス開始の準備	34
6.7.2	アセスメントの実施	34
6.8	問題解決プロセス	34
6.8.1	プロセス開始の準備	34
6.8.2	問題解決	35
	付録	a

1 適用範囲

本標準は、ロケット(人工衛星および宇宙探査機などの無人宇宙機の打上げに用いる無人宇宙輸送機)に搭載するソフトウェアの開発・運用・保守に関する活動並びに関連するシステム開発活動および支援活動に適用する。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。

1.1 適用対象ソフトウェア

本標準では「ソフトウェア」として、以下を適用対象とする。

(1) 計算機プログラム

記憶装置および演算処理装置を有するハードウェアにおいて、演算処理装置で処理される一連の手順およびデータを記述したもので、記憶装置に格納される。記憶装置の種類は問わない。

(2) アプリケーション機能を果たすマイクロプログラム

計算機プログラム(マクロ命令)と同様にシステム構成目目(コンポーネント)が果たすシステム機能(アプリケーション機能)の主要部分を司るマイクロプログラム。マイクロプログラム制御方式の演算処理装置におけるマイクロ命令を用いて、本マイクロプログラムを開発する場合、プロジェクトの判断で、コンポーネント構成目目の1つとしてハードウェアとは切り離して適用する。

(3) FPGA などのプログラマブル・ロジック・デバイス

システム構成目目(コンポーネント)の機能を司る、FPGA などのプログラマブル・ロジック・デバイス。そのプログラマブル・ロジック・デバイスに焼き込む回路情報が複雑で膨大なため計算機プログラムと同等の管理が必要とプロジェクトが判断した場合、コンポーネント構成目目の1つとしてハードウェアとは切り離して適用する。

1.2 他の要求事項との関係

本標準を適用する場合、契約書や調達仕様書など(以下「契約書など」と記述する)の要求事項と本標準の要求事項との間に相違がある場合は、契約書などが優先する。

また、本標準は他の標準と重複した作業を要求するものではなく、互いに補完し合うものである。ソフトウェア開発計画書において他の標準との関係を述べ、その旨を明記する。

2 関連文書

2.1 準拠文書

本標準は、下記の文書の要求をロケットに適用するため、その要求内容を詳細化および具体化した文書である。

- (1) JERG-0-049 ソフトウェア開発標準

2.2 適用文書

次の文書は、本標準に規定する範囲において本標準の一部をなす。適用文書は原則として、ソフトウェア開発計画書を作成する時点の最新版とし、改訂などの変更が生じた場合は、影響の度合いに応じて適用要否および適用時期を決定する。

- (1) JMR-006、「コンフィギュレーション管理標準」

2.3 参考文書

適用しない。

3 用語、定義および略語

本標準で再定義(詳細化)した用語を 3.1 に、新規に定義した用語を 3.2 に示す。また、JERG-0-049 に定義された用語の内、本標準で使用するもののみを 3.3 に示す。

3.1 本標準で再定義した用語

用語	説明
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、メンテナンスなどを含む、開発完了後に行われる作業全般を含む。
ソフトウェア	1.1 適用対象ソフトウェア参照

3.2 本標準で新規に定義した用語

用語	説明
CSC	コンピュータ・ソフトウェア・コンポーネント(Computer Software Component) CSCI を分解した場合の、CSCI より下位レベルのソフトウェア単位。一般には、プログラムレベル、サブプログラムレベルに相当する。CSC は(開発管理上ではなく)機能的な分割単位であり、ひとつのまとまった機能を実現する単位である。CSC (プログラム)を、さらに下位レベルのCSC (サブプログラム)に分割してもよい。最下位レベルのCSC は、複数のCSU (モジュールに相当)から構成される。通常は、プログラム(実行単位)やタスク(並列処理の実行単位)はCSC レベルに相当する。複数のCSU を結合したCSC の試験は、「組合せ試験(CSC試験)」と呼ばれる。
CSCI	コンピュータ・ソフトウェア・コンフィギュレーション・アイテム(Computer Software Configuration Item) ソフトウェア開発を行う場合に、コンフィギュレーション管理など開発管理を行う単位。通常、ソフトウェア開発仕様書はCSCI 単位に制定される。詳細は、付録.1 CSCI 分割ガイドラインを参照のこと。複数のCSC を結合して行うCSCI の試験は、「統合試験(CSCI試験)」と呼ばれる。
CSU	コンピュータ・ソフトウェア・ユニット CSU (Computer Software Unit) ソフトウェアを分割した場合の最小単位(モジュール)である。一般には、モジュールレベルに相当する。CSU は、1つの機能を持ち、それ自体だけを分離した試験が可能なものではない。つまり、分離した試験ができないものは、CSU ではない。また、ドキュメントを作成する最小単位でもある(モジュール設計書に相当)。FORTRAN の場合は、サブルーチン(または、ファンクション)に相当する。CSU の試験は「単体試験(CSU試験)」と呼ばれる。
GTV	GTV : Ground Test Vehicle 地上試験機
システム	システムとは、ソフトウェアの上位要求となるもの。例として、ロケットシステム、サブシステム、コンポーネントなどがあり得る。何をシステムとするかは、システム要求分析の過程で、明確にされるものである。 JERG-0-049 で定義された用語(3.3 参照)の中で“コンピュータシステム”と記載されているものについて、本標準では“システム”に読み替えること。
ソフトウェア試験計画	ソフトウェア検証計画を基に、ソフトウェア試験に関する内容・環境(試験装置など)を詳細化するもの。
マイクロプログラム	機械語よりも低いレベルでハードウェアの動作を細かく制御するプログラム。ハードウェアとソフトウェアの間に位置し、ファームウェアと呼ばれることもある。
マイクロ命令	命令セットアーキテクチャによって定められた命令を、プロセッサ内でさらに分割した単純な命令のこと。
マクロ命令	命令を実行する順番に定義したもの。マクロ命令には言語が定義した命令と、プログラマーが定義した命令とがある。

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	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。
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。
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目目を定義し、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。
識別子	アウトプットやインプットの各項目に付与され、各項目を特定可能な、比較的短い文字列。項目番号のようなものが、識別子に該当する。識別子は数字だけでなくアルファベットや記号などを組み合わせてもよく、また、必ずしも連番である必要はない。 各項目に識別子を付与することにより、要求などの識別(要求管理)やトレーサビリティ確認などの利便性が向上するといったメリットが期待される。
試験計画性	試験対象項目に対して、適切な開発フェーズ・試験環境を用いて、試験が可能である仕様記述であり、計画できること。
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。

用語	説明
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。
独立検証及び有効性確認 (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	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。
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> <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	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 5 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 7 つのプロセスに分類している(図 4.1 にプロセスの構成を示す)。

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・運用・保守で実行するプロセスである。

支援ライフサイクルプロセスとは、開発・運用・保守を通して、主ライフサイクルプロセスを補い、特定の目的のための具体的な方法を示すものである。

各プロセスの活動内容は「アクティビティ」として記載している。

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主ライフサイクルと支援ライフサイクルの両方で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。

また、各プロセスのインプット/アウトプットに文書名を記載しているものがあるが、それらは目的を満たしていれば、文書名は問わない。

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義し、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、各プロセスは適用するソフトウェアおよびプロジェクトの特徴を考慮して、適切に修整されることを想定している。

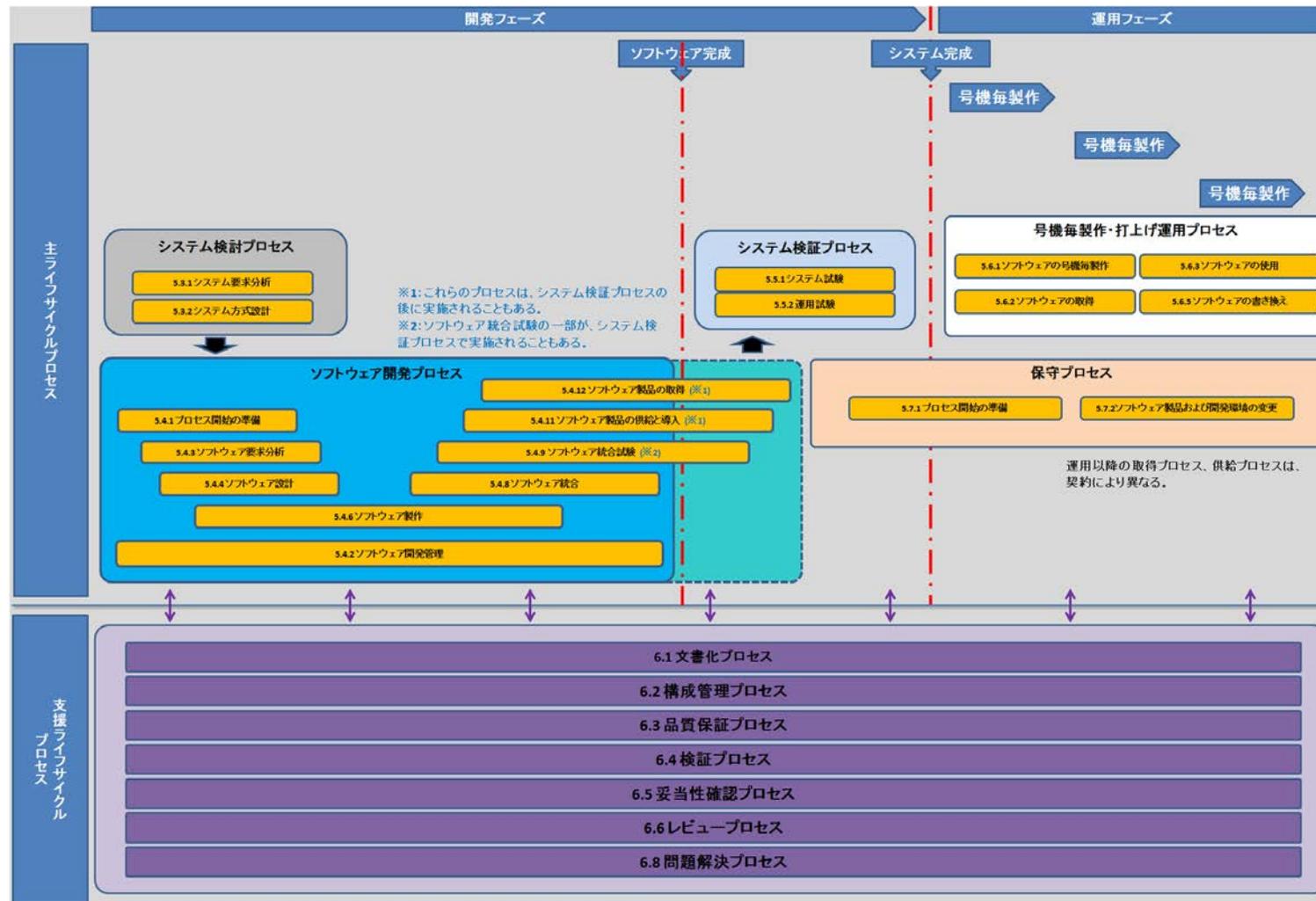


図 4.1 プロセス全体図

5 主ライフサイクルプロセス

5.1 欠番

5.2 欠番

5.3 システム検討プロセス

ソフトウェア開発の準備として、ソフトウェアの上位要求となるシステムについて検討する。

5.3.1 システム要求分析

システムに対する要求を明確化する。

5.3.1.1 アクティビティ

システム開発担当組織と協同して、以下の要求を満たすアクティビティを実施する。

(1) 要求抽出

開発するシステムに対する要求(システムの必要性、利用目的、利用・運用コンセプトなど含む上位要求)を分析し、システムの運用要求・機能要求・機能外要求を抽出する。

(2) 要求仕様作成

運用要求・機能要求・機能外要求に基づき、実現可能性・整合性を確認し、システム要求仕様を定義し文書化する。

(3) システム要求仕様について、上位要求(取得者のニーズまたは上位のシステム要求仕様)とのトレーサビリティおよび設定根拠や前提条件を明確化し、システム要求仕様に漏れ・矛盾などの誤りがなく正確であることを評価する。

5.3.1.2 インプット

(1) 上位要求(取得者のニーズまたは上位のシステム要求仕様)

5.3.1.3 アウトプット

(1) システム要求仕様書(サブシステムの場合「サブシステム要求仕様書」、コンポーネントの場合「コンポーネント要求仕様書」)

(2) 各システム要求仕様の根拠

(3) 上位要求に対するトレーサビリティの評価記録

5.3.2 システム方式設計

システム要求を実現するための方式を設計する。

5.3.2.1 アクティビティ

システム開発担当組織と協同して、下記のアクティビティを行う。

(1) システム要求仕様書に基づき、システム方式(アーキテクチャ)を設計し、構成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、手作業)を明確にする。

(2) システム要求仕様書を各構成品目に配分する。

(3) 上記(1)および(2)の結果をシステム仕様書として文書化する。

(4) システム間/サブシステム間/構成品目間のインタフェース方式を設計してインタフェース管理仕様書を定める。

(5) 各構成品目に割り当てられた要求の実現に対するリスク評価を行う。

- (6) システム仕様の設計根拠や前提条件を明確化し、システム仕様に漏れ・矛盾などの誤りがなく正確であることを評価する。
- (7) システム仕様書とシステム要求仕様書などの上位要求とのトレーサビリティを明確化し、システム仕様に漏れ・矛盾などの誤りがなく正確であることを評価する。
- (8) インタフェース管理仕様書とシステム要求仕様書などの上位要求とのトレーサビリティを明確化し、インタフェース管理仕様書に漏れ・矛盾などの誤りがなく正確であることを評価する。

5.3.2.2 インプット

- (1) システム要求仕様書

5.3.2.3 アウトプット

- (1) システム仕様書(機体システムの場合「機体システム仕様書」、サブシステムの場合「サブシステム開発仕様書」、コンポーネントの場合「コンポーネント開発仕様書」)
- (2) 各構成品目への要求(サブシステムおよびコンポーネントの開発仕様書、ソフトウェアに対する要求)
- (3) インタフェース管理仕様書
- (4) 各構成品目への要求の実現に対するリスク評価記録
- (5) システム仕様の設計根拠(前提条件など含む)
- (6) システム仕様の設計根拠評価記録
- (7) システム仕様書とシステム要求仕様書とのトレーサビリティ評価記録
- (8) インタフェース管理仕様書とシステム要求仕様書とのトレーサビリティ評価記録

5.4 ソフトウェア開発プロセス

ソフトウェア開発プロセスには、開発したソフトウェア製品をシステムに組み込んで最終的に開発が完了したことを確認するまでに、開発者が行うアクティビティが含まれる。ソフトウェア開発プロセスは基本的には、取得者との契約に基づいて供給者が実施するアクティビティであるが、取得者自身が実施するアクティビティも含まれる。

アクティビティの実施順序は本書に記載する順序である必要はないが、ソフトウェア開発計画書に各アクティビティの実施順序やプロセス間の関係などプロセス全体構成を定義したうえ、プロセス全体の管理方法を定め、プロセスの管理を行う。(図 5.4.1 にプロセス全体構成の例をイメージで示す)



図 5.4.1 ソフトウェア開発プロセス全体構成の例

5.4.1 プロセス開始の準備

ソフトウェア開発プロセスを実施するための準備として、ソフトウェア開発計画書を作成する。

5.4.1.1 アクティビティ

(1) ソフトウェア開発計画の立案

ソフトウェア開発計画を立案し、文書化する。別途、プロジェクト管理計画書が作成されている場合、重複した内容は含めなくてよい。また、両文書を一体化したソフトウェア開発計画書を作成してもよい。ソフトウェア開発計画書には、以下の情報を含める。

- (a) 対象となるシステムの範囲
- (b) ソフトウェア開発ライフサイクル(ソフトウェア開発完了を含む)およびプロセスの定義
- (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
- (d) 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
- (e) 審査計画
- (f) 命名規約、コーディング規約、使用する用語や略語の定義など、設計・製作時に従うべき規準の適用計画
- (g) 作業単位ごとの作業計画・進捗管理の方法(開発スケジュールを含む)
- (h) 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)
- (i) 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
- (j) システムとの適合性評価計画

5.4.1.2 アウトプット

- (1) ソフトウェア開発計画書

5.4.2 ソフトウェア開発管理

開発プロセス全体を通じて、ソフトウェア管理計画書に基づいた管理を実施する。

※付録2 ソフトウェア開発管理ガイドライン参照

5.4.2.1 アクティビティ

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

5.4.2.2 アウトプット

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

5.4.3 ソフトウェア要求分析

開発するソフトウェアに対する要求を分析してソフトウェア要求仕様書としてまとめる。また、ソフトウェア単体開発などの作業でソフトウェア開発仕様書を要する場合は、ソフトウェア要求仕様書の要求を実現するための方式(機能)を設計し、ソフトウェア開発仕様書(機能仕様)を定める。

なお、ソフトウェア開発仕様書を作成する場合は、ソフトウェア要求仕様書に相当する内容をソフトウェア開発仕様書に含めて作成してもよい。また、ソフトウェア開発仕様書を作成しない場合は、ソフトウェア開発仕様書に相当する内容を 5.4.4 項にて作成するソフトウェア設計書に含める。

※ソフトウェア要求仕様書に相当する内容をソフトウェア開発仕様書に含めて作成する場合は、本書内の「ソフトウェア要求仕様書」を「ソフトウェア開発仕様書」と読み替えること。また、ソフトウェア開発仕様書を作成しない場合は、本書内の「ソフトウェア開発仕様書」を「ソフトウェア設計書」と読み替えること。

5.4.3.1 アクティビティ

システム要求分析およびシステム方式設計のアクティビティを通じ、またそれらアクティビティの成果を用いて、以下のアクティビティを実施する。

- (1) システム仕様書、各構成品目への要求およびインタフェース管理仕様を分析し、ソフトウェアに対する要求事項をソフトウェア要求仕様書として文書化する。
- (2) ソフトウェア要求仕様書の各要求項目には、構成管理プロセス(6.2 参照)における構成識別のため、また、設計書や試験結果とのトレーサビリティの確認などの作業に活用できるよう、個別に識別子(項番含む)を付与する。
- (3) ソフトウェア要求仕様書には、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要求を含める。
- (4) ソフトウェア要求仕様書には、システムの特性変動や飛行環境の変動などの変動要素の定義およびシステムで想定される例外や異常・故障などの事象とその対策を含める。
- (5) ソフトウェア要求仕様書とシステム仕様書・インタフェース管理仕様書とのトレーサビリティおよび整合性を確認し、文書化する。トレーサビリティの詳細さは、ソフトウェア要求仕様書の各項目に付与した識別子の単位とする。
- (6) ソフトウェア要求仕様書の各項目について、実現可能性を評価し、必要に応じて要求根拠を明らかにする。この実現可能性としては、新規性や技術的難易度、要求精度の高さ、想定される目標プラットフォームの能力(速度、容量など)を考慮する。
- (7) ソフトウェア要求仕様の確定状況について評価する。
- (8) ソフトウェア要求仕様書に基づき方式(機能)設計を行い、ソフトウェア開発仕様書を作成する。なお、組込みソフトウェアの場合は、ソフトウェア開発仕様書の内容を上位システム(コンポーネント)の開発仕様書に含めてもよい。ソフトウェア開発仕様書の各項目は、識別子(項番含む)によって個別に識別可能とする。また、アルゴリズム解説などの文書はソフトウェア開発仕様書とは別に定めてもよい。
- (9) ソフトウェア開発仕様書とソフトウェア要求仕様書とのトレーサビリティを確認し、文書化する。
- (10) COTS または再利用ソフトウェアを利用する場合には、ソフトウェア開発仕様書、システム仕様書およびインタフェース管理仕様書との適合性を評価する。
- (11) ソフトウェア開発仕様書の各要求項目について、検証方法を含む検証計画(妥当性確認計画を含む)を立案する。
- (12) 検証計画書について、検証可能性および検証網羅性を評価する。

- (13) ソフトウェア検証計画書において、試験環境と実ハードウェアの挙動の差異の影響を受ける試験を実施する場合、あるいは、試験を実施せずレビュー／解析などでの確認を実施する場合は、その確認対象の識別と確認方法が適切であるかの評価も含める。

5.4.3.2 計測活動

適用しない。^(※)

※ロケットの開発で必要な計測活動に関する要求はアクティビティに含まれるため。

5.4.3.3 インプット

- (1) システム仕様書(機体システムの場合「機体システム仕様書」、サブシステムの場合「サブシステム開発仕様書」、コンポーネントの場合「コンポーネント開発仕様書」)
- (2) インタフェース管理仕様書
- (3) ソフトウェアに対する要求 (各構成目への要求に含まれる)

5.4.3.4 アウトプット

- (1) ソフトウェア要求仕様書(ソフトウェア開発仕様書に含まれる場合がある)
- (2) ソフトウェア要求仕様書とシステム仕様書・インタフェース管理仕様書とのトレーサビリティ評価記録
- (3) ソフトウェア要求仕様書の根拠
- (4) ソフトウェア開発仕様書(組み込みソフトウェアではコンポーネント開発仕様書に含まれる場合がある)
- (5) ソフトウェア開発仕様書とソフトウェア要求仕様書とのトレーサビリティ評価記録
- (6) COTS・再利用ソフトウェアとソフトウェア開発仕様書との適合性評価記録
- (7) ソフトウェア検証計画書
- (8) ソフトウェア検証計画書の検証可能性および検証網羅性評価記録
- (9) ソフトウェア要求審査記録

5.4.3.5 審査

- (1) 本アクティビティのアウトプット(5.4.3.4 項の(1)～(8))に基づいて審査対象を選択し、ソフトウェア要求審査の実施計画を定める。
- (2) 実施計画に基づいて審査を実施し、結果を記録する。
- (3) 審査において明らかになっ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。

5.4.4 ソフトウェア設計

ソフトウェア要求分析結果に基づき設計を行う。

なお、本標準では、基本設計と詳細設計を分割していないが、実際の開発プロセスでは、必要に応じ、基本設計と詳細設計を分割して実施する。

5.4.4.1 アクティビティ

ソフトウェア要求仕様書(ソフトウェア開発仕様書に含まれる場合がある)に基づき、以下の事項を満たすアクティビティを実施する。

【基本設計】

- (1) 機能階層構造などの機能分割結果に基づいてモジュール分割を行い、各機能を構成するモジュール間の関係を明確にし、ソフトウェア構造を設計する。
- (2) ソフトウェア構造の各階層(CSCI/CSC)のモジュールに開発仕様書の機能仕様の各項目を配分し、ソフトウェア(基本)設計書を定める。
- (3) ソフトウェア(基本)設計書には、ソフトウェア要求仕様書に定められた機能外要求(処理時間要求、メモリのリソース要求など)に対する設計、配分を含める。
- (4) ソフトウェア(基本)設計書には、モジュール間インタフェースの設計を含める。

【詳細設計】

- (5) ソフトウェア(基本)設計書に基づき、CSU レベルに分割し、各 CSU 内の設計を実施し、ソフトウェア(詳細)設計書を定める。

【基本設計、詳細設計共通】

- (6) 基本設計書および詳細設計書は設計書として文書化する。設計書はソフトウェア規模などの状況に応じ、1つの設計書として文書化してもよい。
- (7) ソフトウェア(基本、詳細)設計書の各項目は、トレーサビリティを確保できるレベルで、識別子(項番含む)によって個別に識別可能とする。
- (8) ソフトウェア(基本、詳細)設計書とソフトウェア要求仕様書とのトレーサビリティおよび適合性を解析し、記録する。
- (9) 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。
- (10) COTS または再利用ソフトウェアを利用する場合には、ソフトウェア設計書との適合性を解析する。
- (11) ソフトウェア検証計画書に従って、ソフトウェア試験計画書、組合せ試験(CSC 試験)仕様書およびソフトウェア統合試験(CSCI 試験)仕様書を作成する。
ソフトウェア統合試験(CSCI 試験)仕様書は、以下の観点を考慮する。
 - (ア) 最大負荷
 - (イ) ソフトウェア開発仕様書に対する網羅性
 - (ウ) ソフトウェア設計書の必要な項目
 - (エ) 例外や異常・故障などの事象
 - (オ) COTS・再利用ソフトウェアのシステムに対する適合性
- (12) 耐故障性向上のための実装指針(関数からの異常戻り値などの確実なエラー処理、無限ループや無限待ちの回避、ポインタ範囲外参照の回避など)を設定する。

5.4.4.2 計測活動

適用しない。^(※)

※ロケットの開発で必要な計測活動に関する要求はアクティビティに含まれるため。

5.4.4.3 インプット

- (1) ソフトウェア要求仕様書(ソフトウェア開発仕様書に含まれる場合がある)
- (2) ソフトウェア開発仕様書(組込みソフトウェアではコンポーネント開発仕様書に含まれる場合がある)
- (3) ソフトウェア検証計画書

5.4.4.4 アウトプット

- (1) ソフトウェア(基本、詳細)設計書
- (2) ソフトウェア設計書とソフトウェア要求仕様書のトレーサビリティおよび適合性評価記録
- (3) ソフトウェア設計根拠および実現可能性評価記録
- (4) COTS・再利用ソフトウェアとソフトウェア設計書との適合性評価記録
- (5) ソフトウェア試験計画書
- (6) 組合せ試験(CSC 試験)仕様書
- (7) ソフトウェア統合試験(CSCI 試験)仕様書
- (8) ソフトウェア設計審査記録

5.4.4.5 審査

- (1) 本アクティビティのアウトプット(5.4.4.4 項の(1)～(7))に基づいて審査対象を選択し、ソフトウェア設計審査の実施計画を定める。
- (2) 実施計画に基づいて審査を実施し、結果を記録する。
- (3) 審査において明らかになっ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。

5.4.5 欠番

5.4.6 ソフトウェア製作

設計書(詳細設計)に基づいてソースコードを作成し、デバッグおよび単体試験(CSU 試験)を行う。

5.4.6.1 アクティビティ

ソフトウェア製作では、以下の事項を満たすアクティビティを実施する。

- (1) 設計書(詳細設計)およびコーディング規約に基づき、ソースコードを作成する。
- (2) コードチェックツールなどのツールで静的解析を実施し、ソースコードの品質評価を実施する。
- (3) ソースコードとソフトウェア設計書(詳細設計)のトレーサビリティを評価し、その結果を記録する。
- (4) ソフトウェア試験計画書に従い、単体試験(CSU 試験)仕様書を作成する。
- (5) 単体試験(CSU 試験)仕様書に従って単体試験(CSU 試験)を実施し、実施結果を合否判定可能な形式で記録する。
- (6) 単体試験(CSU 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。少なくともソースコードの分岐を十分に網羅させる。

5.4.6.2 計測活動

適用しない。(※)

※ロケットの開発で必要な計測活動に関する要求はアクティビティに含まれるため。

5.4.6.3 インプット

- (1) ソフトウェア(詳細)設計書
- (2) ソフトウェア試験計画書

5.4.6.4 アウトプット

- (1) ソースコード
- (2) 単体試験(CSU 試験)仕様書
- (3) 単体試験(CSU 試験)記録
- (4) 静的解析結果
- (5) ソースコードとソフトウェア設計書のトレーサビリティ評価記録
- (6) ソフトウェア製作審査記録

5.4.6.5 審査

- (1) 本アクティビティのアウトプット(5.4.6.4 項の(1)～(5))に基づいて審査対象を選択し、ソフトウェア製作審査の実施計画を定める。
- (2) 実施計画に基づいてソフトウェア製作審査を実施し、結果を記録する。
- (3) 審査において明らかになっ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。

5.4.7 欠番

5.4.8 ソフトウェア統合

各モジュールを統合し、CSCI とする。なお、ソフトウェア統合はソフトウェア規模などの状況に応じ、ソフトウェア統合試験(CSCI 試験)の準備活動として 5.4.9 項のアクティビティに含めて実施してもよい。

5.4.8.1 アクティビティ

ソフトウェア統合では、以下の事項を満たすアクティビティを実施する。

- (1) モジュール階層構造の最下位の CSU を統合して CSC とし、さらに CSC を統合して最上位の CSCI として、ソフトウェア統合試験(CSCI 試験)に供する CSCI の版を確定する。
- (2) 統合過程において、必要に応じて、CSU 間および CSC 間のインタフェースを確認する。
- (3) 統合過程での確認事項およびその結果を記録する。
- (4) CSC ごとに機能・性能を検証する組合せ試験(CSC 試験)は、必要に応じて実施するものとし、実施要否については、ソフトウェア開発計画書に定める。実施する場合は下記のように行う。
 - (a) ソフトウェア試験計画書および組合せ試験(CSC 試験)仕様書を必要に応じて見直し、組合せ試験(CSC 試験)手順を文書化する。
 - (b) ソフトウェア試験計画書、組合せ試験(CSC 試験)仕様書および組合せ試験(CSC 試験)手順に従って組合せ試験(CSC 試験)を実施する。
 - (c) 組合せ試験(CSC 試験)結果およびその評価記録を組合せ試験(CSC 試験)成績書にまとめ、合否判定を行う。

5.4.8.2 計測活動

適用しない。(※)

※ロケットの開発に必要な計測活動に関する要求はアクティビティに含まれるため。

5.4.8.3 インプット

- (1) ソースコード(CSU)
- (2) ソフトウェア設計書
- (3) ソフトウェア試験計画書
- (4) 組合せ試験(CSC 試験)仕様書

5.4.8.4 アウトプット

- (1) ソースコード(CSCI)
- (2) オブジェクト(実行形式)コード
- (3) ソフトウェア統合記録
- (4) ソフトウェア試験計画書(更新)
- (5) 組合せ試験(CSC 試験)仕様書(更新)
- (6) 組合せ試験(CSC 試験)手順書
- (7) 組合せ試験(CSC 試験)成績書

5.4.9 ソフトウェア統合試験(CSCI 試験)

ソフトウェア統合試験(CSCI 試験)は、CSCI レベル(1 個または複数の CSCI)での機能および性能を確認する(このアクティビティの中で組合せ試験(CSC 試験)を実施する場合は、それを含む)。ソフトウェア統合試験(CSCI 試験)では検証マトリクスなどの確認用資料を作成し、機能要求、機能外要求および機能仕様のすべてを網羅していることを示す。

なお、ソフトウェアをシステム(コンポーネント)に組み込んでから実施する場合でも、ソフトウェア要求仕様書の確認のために実施する場合は本試験に含めることができる。

5.4.9.1 アクティビティ

ソフトウェア統合試験(CSCI 試験)では、以下の事項を満たすアクティビティを実施する。

- (1) ソフトウェア統合試験(CSCI 試験)準備
 - (a) ソフトウェア試験計画書およびソフトウェア統合試験(CSCI 試験)仕様書を必要に応じて見直し、ソフトウェア統合試験(CSCI 試験)手順を文書化する。
 - (b) ソフトウェア統合試験(CSCI 試験)は、以下の観点を考慮する。
 - (ア) 最大負荷
 - (イ) ソフトウェア開発仕様書に対する網羅性
 - (ウ) ソフトウェア設計書の必要な項目
 - (エ) 例外や異常・故障などの事象
 - (オ) COTS・再利用ソフトウェアのシステムに対する適合性
 - (c) ソフトウェア試験計画書は以下の観点を考慮する。
 - (ア) 試験コンフィギュレーション
 - (イ) 試験設備
 - (ウ) スケジュール
 - (エ) 試験体制

- (2) ソフトウェア統合試験(CSCI 試験)の実施
 - (a) ソフトウェア試験計画書、ソフトウェア統合試験(CSCI 試験)仕様書およびソフトウェア統合試験(CSCI 試験)手順書に従って試験を実施する。
 - (b) 必要に応じて試験結果のクイックレビューを実施し、ソフトウェア統合試験(CSCI 試験)継続の可否の判断を行う。
 - (c) 試験の手順、ログ、試験結果、入力条件、供試ソフトウェアの構成やバージョンなどの構成情報など、試験に必要な情報を記録することにより、再現性を確保する。試験結果を記録し、評価基準に照らして評価し、合否の判定を行う。
 - (d) ソフトウェアまたは試験仕様書に修正が発生した場合、以前実施した各種試験の有効性を評価し、必要な場合は再度実施する。

5.4.9.2 計測活動

適用しない。^(※)

※ロケットの開発に必要な計測活動に関する要求はアクティビティに含まれるため。

5.4.9.3 インプット

- (1) オブジェクト(実行形式)コード
- (2) ソフトウェア開発仕様書
- (3) ソフトウェア試験計画書
- (4) ソフトウェア統合試験(CSCI 試験)仕様書

5.4.9.4 アウトプット

- (1) ソフトウェア試験計画書(更新)
- (2) ソフトウェア統合試験(CSCI 試験)仕様書(更新)
- (3) ソフトウェア統合試験(CSCI 試験)手順書
- (4) ソフトウェア統合試験(CSCI 試験)成績書
- (5) ソフトウェア試験審査記録

5.4.9.5 審査

- (1) 本アクティビティのアウトプット(5.4.9.4 項の(1)～(4))に基づいて審査対象を選択し、ソフトウェア試験審査の実施計画を定める。
- (2) 実施計画に基づいてソフトウェア試験審査を実施し、結果を記録する。
- (3) 審査において明らかになった要処置事項について、期限、対処状況、影響度などの優先度に応じ、適切にフォローアップを行う。

5.4.10 目標プラットフォームへのインストール(組込み)

適用しない。^(※)

※目標プラットフォームへのインストールは、ソフトウェア製品の供給と導入のアクティビティ(5.4.11.1 参照)の(5)に含まれるため。

5.4.11 ソフトウェア製品の供給と導入

開発が完了したソフトウェア製品^(※)の内、別途指定された対象物について、供給者が取得者へ引き渡す準備を行う。号機ごとに製作するソフトウェア製品の供給については、号機毎製作・打上げ運用プロセス(5.6 参照)に定める。

ソフトウェア製品の供給と導入は、システム検証プロセス(5.5 参照)完了後に実施してもよい。

※ソフトウェアとソフトウェアに関する手順、文書、関連データの集合体の呼称。

5.4.11.1 アクティビティ

ソフトウェア製品の供給と導入では、以下の事項を満たすアクティビティを実施する。

- (1) 供給の準備
 - (a) ソフトウェア(基本、詳細)設計書およびソフトウェア構成管理並びに製作プロセスおよび試験プロセスを確立し、製品仕様書に定義する。
 - (b) 使用に必要な取扱手順、運用手順を明確にし、文書化する。これにはソフトウェアをハードウェアにインストールする手順、インストール結果の確認手順、システム上での動作確認手順および運用に関する規格などの情報を含める。
- (2) 供給するソフトウェア製品の開発完了の審査
 - (a) 審査対象のソフトウェアに関し、ソフトウェア開発完了審査(コンポーネントなどの開発完了審査に含めてもよい)の実実施計画を定める。審査では下記を確認する。
 - (ア) ソフトウェア製品が供給可能な状態にある。
 - (イ) ソフトウェア検証計画書で計画している開発完了に必要な検証がすべて終了している。
 - (ウ) ソフトウェア試験がすべて合格している(5.4.12 の取得者による受入検査・試験および(3)供給するソフトウェア製品の試験を除く)。
 - (エ) ソフトウェア製品の不適合がすべて処置され、その結果が確認されている。
 - (オ) システムの不適合修正や設計変更などのソフトウェア要求仕様の変更によるソフトウェア製品への影響がすべて処置され、その結果が確認されている。
 - (カ) 計画された妥当性確認がすべて行われている。
 - (b) 実施計画に基づいて審査を実施し、結果を記録する。
 - (c) 審査において明らかになっ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。
- (3) 供給するソフトウェア製品の製作・試験

ソフトウェア製品仕様書に基づき、下記を行う。

 - (a) ソフトウェア収録媒体を作成する。
 - (b) ソフトウェア製品の試験計画、試験仕様および試験手順を文書化し、それに基づいて試験を実施する。
 - (c) 試験結果およびその評価記録をソフトウェア製品試験成績書にまとめ、合否判定を行う。

- (4) 下記の内容を含むリリース文書(文書パッケージを含む)の作成
 - (a) ソフトウェア製品の識別
 - (b) ソフトウェア製品の構成管理情報
 - (c) ソフトウェア製品の使用・検査・試験の履歴
 - (d) ソフトウェア製品の不適合履歴
 - (e) ソフトウェア製品の試験成績
 - (f) ソフトウェア製品に組み込まれている運用データの仕様
- (5) 目標プラットフォームへのインストール
目標プラットフォームへのインストール後に取得者に引き渡す場合、取扱説明書に従ってインストールを行う。

5.4.11.2 インプット

- (1) ソースコード
- (2) オブジェクト(実行形式)コード
- (3) ソフトウェア(基本、詳細)設計書
- (4) ソフトウェア検証結果
- (5) ソフトウェア試験成績書
- (6) 要処置事項ステータス
- (7) 妥当性確認結果

5.4.11.3 アウトプット

- (1) 運用手順書(運用データ設定手順などを含む)
- (2) ソフトウェア製品収録媒体作成手順書(コンパイラオプションなどの手順を含む)
- (3) ソフトウェア製品仕様書
- (4) ソフトウェア製品取扱説明書
- (5) ソフトウェア製品(収録媒体またはハードウェア組み込み状態)
- (6) ソフトウェア製品試験計画書
- (7) ソフトウェア製品試験仕様書
- (8) ソフトウェア製品試験手順書
- (9) ソフトウェア製品試験成績書(合否判定結果を含む)
- (10) ソフトウェア製品リリース文書(文書パッケージなど)

5.4.12 ソフトウェア製品の取得

取得者はソフトウェア製品受け入れ前に、受入検査・試験を実施し、ソフトウェア製品が要求に適合していることを確認する。受入検査・試験は、5.4.11 項の試験に重複して実施する必要はなく、試験内容を明確に定めて実施する。

受入検査・試験に合格したソフトウェア製品を取得する。

ソフトウェア製品の取得は、システム検証プロセス(5.5 参照)完了後に実施してもよい。

5.4.12.1 アクティビティ

ソフトウェア製品の取得では、以下の事項を満たすアクティビティを実施する。

- (1) 開発仕様書および製品仕様書に基づいて受入検査・試験の計画を立案し、受入検査・試験の仕様および手順を文書化する。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。
- (2) 受入検査・試験の計画、仕様および手順に従い、受入検査・試験を実施する。
- (3) 受入検査・試験の結果およびその評価記録に基づいて、合否判定する。
- (4) ソフトウェア製品およびリリース文書が取得可能な状態にあることを確認する。

5.4.12.2 インプット

- (1) ソフトウェア開発仕様書
- (2) ソフトウェア製品仕様書
- (3) ソフトウェア製品(収録媒体またはハードウェア組み込み状態)
- (4) リリース文書

5.4.12.3 アウトプット

- (1) 受入検査実施要領書(受入検査・試験計画、受入検査・試験仕様などを含む)
- (2) 受入検査・試験成績書
- (3) リリース文書(受入検査記録を含む)
- (4) ソフトウェア製品(受入検査済み)

5.5 システム検証プロセス

5.5.1 システム試験

システム試験とは、ソフトウェアをハードウェアと組み合わせた状態での機能確認を行うコンポーネント試験および機体システム、サブシステムに対する機能確認の試験をいう。この試験において、開発したソフトウェアがシステム仕様に適合していることを確認・評価する。

5.5.1.1 アクティビティ

システム試験では、以下の事項を満たすアクティビティを実施する。

- (1) 試験準備
 - (a) ソフトウェア検証計画書を反映したシステム試験計画書、システム試験仕様書およびシステム試験手順書を作成する。
 - (b) システム試験仕様書は以下の観点を考慮する。
 - (ア) 最大負荷
 - (イ) システム要求仕様、システム仕様およびインタフェース管理仕様に対する網羅性
 - (ウ) ソフトウェア要求仕様書に対する網羅性
 - (エ) 例外や異常・故障などの事象
 - (オ) COTS/再利用品目のシステムに対する適合性。

(2) 試験の実施

- (a) システム試験計画書、システム試験仕様書およびシステム試験手順書に従ってシステムを統合し、システム試験を実施する。
- (b) 必要に応じて試験結果のクイックレビューを実施し、試験継続の可否の判断を行う。
- (c) 試験の手順、ログ、試験結果、入力条件、供試ソフトウェアの構成やバージョンなどの構成情報など、試験に必要な情報を記録することにより、再現性を確保する。試験結果を記録し、評価基準に照らして評価し、合否の判定を行う。
- (d) 供試体(ハードウェア、ソフトウェア)またはシステム試験仕様に修正が発生した場合は、以前実施した各種試験の有効性を評価し、必要な場合は再度実施する。

5.5.1.2 計測活動

適用しない。(※)

※ロケットの開発で必要な計測活動に関する要求はアクティビティに含まれるため。

5.5.1.3 インプット

- (1) システム要求仕様書
- (2) システム仕様書
- (3) インタフェース管理仕様書
- (4) ソフトウェア要求仕様書
- (5) ソフトウェア検証計画書

5.5.1.4 アウトプット

- (1) システム試験計画書
- (2) システム試験仕様書
- (3) システム試験手順書
- (4) システム試験成績書

5.5.2 運用試験

運用試験とは、システムでの実運用手順でソフトウェアを運用して行う試験である。本試験には、GTVなどの設備を用いて地上設備を含めた検証を行う試験も含まれる。手順はシステムの運用手順として準備したものを使用する。運用試験の動作評価としては下記とする。

- (1) ソフトウェアが設計通りに動作する。運用の前提がソフトウェア設計の前提と相違ないこと。
- (2) システムで意図した操作が行えること。

5.5.2.1 アクティビティ

運用試験では、以下の事項を満たすアクティビティを実施する。

- (1) 試験準備
 - (a) ソフトウェア検証計画書を反映した運用試験計画書、運用試験仕様書および運用試験手順書を作成する。
 - (b) 運用試験仕様書は以下の観点を考慮する。
 - (ア) システム要求、システム仕様およびインタフェース管理仕様に対する網羅性
 - (イ) ソフトウェア要求仕様書に対する網羅性
- (2) 試験の実施
 - (a) 運用試験計画書、運用試験仕様書および運用試験手順書に従って運用試験を行う。
 - (b) 必要に応じて試験結果のクイックレビューを実施し、試験継続の可否の判断を行う。
 - (c) 試験の手順、ログ、試験結果、入力条件、供試ソフトウェアの構成やバージョンなどの構成情報など、試験に必要な情報を記録することにより、再現性を確保する。試験結果を記録し、評価基準に照らして評価し、可否の判定を行う。
 - (d) 試験結果を記録し、評価基準に照らして評価し、可否の判定を行う。
 - (e) 供試体(ハードウェア、ソフトウェア)または運用試験仕様に修正が発生した場合は、以前実施した各種試験の有効性を評価し、必要な場合は再度実施する。

5.5.2.2 計測活動

適用しない。(※)

※ロケットの開発で必要な計測活動に関する要求はアクティビティに含まれるため。

5.5.2.3 インプット

- (1) システム要求仕様書
- (2) システム仕様書
- (3) インタフェース管理仕様書
- (4) ソフトウェア要求仕様書
- (5) ソフトウェア検証計画書

5.5.2.4 アウトプット

- (1) 運用試験計画書
- (2) 運用試験仕様書
- (3) 運用試験手順書
- (4) 運用試験成績書

5.6 号機毎製作・打上げ運用プロセス

号機毎製作・打上げ運用プロセスは、号機ごとに必要となるソフトウェア製品を製作し、使用するためのプロセスである。

問題解決を行うためにソフトウェア製品の維持設計(保守)が必要な場合は保守プロセス(5.7 参照)に従う。

5.6.1 ソフトウェアの号機毎製作

号機、コンポーネントごとに運用データを設定してソフトウェアを製作する場合の作業について定める。ソフトウェア製品の供給と導入(5.4.11 参照)で定めた手順に従ってソフトウェアの製作および試験を実施し、成績書にまとめる。ソフトウェア製品を改修する必要がある場合には、保守プロセス(5.7 参照)に従い、開発プロセス(5.4 参照)と同様の処置を行う。

5.6.2 ソフトウェアの取得

号機、コンポーネントごとに運用データを設定して製作するソフトウェアに対し、5.4.12 項と同等の要求を満たすような受入検査・試験を計画し、実施する。また、号機、コンポーネントごとの定数表などの文書は、受入検査・試験成績書に含める。

5.6.3 ソフトウェアの使用

システムに組み込まれたソフトウェアは、ソフトウェア製品取扱説明書に従い、所定の動作環境において使用する。

5.6.4 ユーザサポート

適用しない。(※)

※ロケットではユーザサポートは存在しないため。

5.6.5 ソフトウェアの書き換え

システムに組み込まれた状態のソフトウェアを運用現場で書き換える場合に適用する。

- (1) 運用現場での書き換えは原則として定数を対象とする。
- (2) ソフトウェア製品の供給と導入(5.4.11 参照)で定めた手順に従ってソフトウェアの製作および試験を実施し、成績書にまとめる。
- (3) 定数の書き換え手順を定める。必要に応じて、運用試験時に手順を確認する。

5.7 保守プロセス

ソフトウェア開発完了後の保守プロセスとして、ソフトウェアの改修に必要な、ソフトウェア製品および開発環境の、維持または変更作業について定める。

5.7.1 プロセス開始の準備

- (1) 保守サービスの供給者は必要な場合、保守の実施計画を立案する。
- (2) 必要な場合(運用サービス供給者と保守サービス供給者が異なる場合など)、運用プロセスからの作業指示および技術指示の手順を文書化する。

5.7.1.1 ソフトウェア製品および開発環境の維持

開発したソフトウェアを維持可能なように、必要となるソフトウェア製品や開発環境を識別し期限を定めて維持する。

維持の対象(ソフトウェア製品、開発環境)に問題が判明した場合または維持の継続が困難となる場合などの問題が生じた場合には取得者に報告する。

5.7.2 ソフトウェア製品および開発環境の変更

- (1) ソフトウェア製品に変更が必要となった場合、開発プロセス(5.4 参照)について、変更内容に応じて必要な作業を定め、ソフトウェア製品の変更を行う。
 - (a) 問題解決については、問題解決プロセス(6.8 参照)に従い処置すること。
 - (b) 修正の実施については、構成管理プロセス(6.2 参照)に従い処置すること。
- (2) ソフトウェアの開発環境に変更が必要となった場合、ソフトウェア製品の品質に影響のないことが確認されている開発環境を使用する。

5.7.3 移行

適用しない。(※)

※ロケットの場合、ソフトウェアを別環境に移行することがないため。

5.7.4 ソフトウェア廃棄

適用しない。(※)

※ロケットの場合、ソフトウェア製品の廃棄がないため(アンインストールなどの必要がないため)。

6 支援ライフサイクルプロセス

6.1 文書化プロセス

文書化プロセスは、ライフサイクル全体に渡って作成する文書およびデータのためのプロセスである。

構成管理対象については、本項に加え、構成管理プロセス(6.2 参照)で定められた管理手順に従って管理される。

6.1.1 プロセス開始の準備

- (1) ソフトウェアのライフサイクルを通じて作成する文書およびデータ並びにそれらの作成時期を明確にし、審査においてそれらが準備されていることを確認する。作成する文書およびデータ並びにそれらの作成時期は、ソフトウェア開発計画書に定める。
- (2) 文書およびデータの作成、発行、維持に関する手続きを定める。
- (3) 文書で適用する様式(記載内容、フォーマットなど)を定める。
- (4) 外部から受け入れた文書およびデータの管理手順を定める。

6.1.2 作成

- (1) 各文書およびデータは文書標準に従って作成する。
- (2) 設計に使用する文書類は必ず文書番号(識別子)を付ける。
- (3) 文書およびデータ作成時に使用・参照する他の文書およびデータは正規版であることを確認する。止むを得ず、仮版を使用する場合には、文書およびデータ中にその旨を記載するなどの方法で識別し、正規版が発行されたときに、仮版と差異がないことを確認する。
- (4) 作成した文書およびデータは手順に従い承認する。

6.1.3 発行(配付)

- (1) 文書およびデータはソフトウェア開発計画書に定められた時期に発行する。
- (2) 文書およびデータを発行する場合には、定められた配付先に配付する。
- (3) 仮版として発行する場合には、配付先を記録し、正規版を必ず配付する。
- (4) 文書およびデータの発行および配付には、紙の媒体、電子媒体または他の媒体を使用してもよい。原本の媒体は、セキュリティ、保守およびバックアップの要求事項に従って保管する。

6.1.4 維持

- (1) 文書およびデータの内容を変更した場合には、必ず改訂符(リビジョン)を付して再発行する。(同一改訂符で内容の異なる文書およびデータを存在させないこと)
- (2) 文書改訂時には、改訂理由を記述(文書に記載してもよい)し、改訂の履歴が追えるようにする。
- (3) 文書を改訂したときには、定められた配付先に配付する。
- (4) 改訂前の文書およびデータ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管理する。不要な場合には廃棄する。
- (5) 改訂した文書およびデータは手順に従い承認する。

6.1.5 取得者による承認など

取得者と供給者の契約に基づいて供給者が取得者に提出・提示する文書およびデータは、次のいずれかの区分で承認などを行う。

この区分は各文書およびデータごとに契約書などあるいはソフトウェア開発計画書に定める。区分を変更する場合は、取得者および供給者で調整して取り決める。

(1) 承認

適用に先だって取得者の承認が必要なもので、供給者は契約書などに定められた期限内に取得者に提出する。取得者は承認可否の結果を供給者に通知する。

(2) 審査

適用に先だって取得者の審査が必要なもので、供給者は契約書などに定められた期限内に取得者に提出する。取得者は審査結果を供給者に通知する。

(3) 通知

供給者は契約書などに定められた期限内に取得者に提出する。

6.2 構成管理プロセス

構成管理プロセスは、主ライフサイクルプロセス(5 参照)で作成されるアウトプットの中から構成管理対象を識別し、管理するためのプロセスである。

ここでは、JMR-006 の要求をソフトウェアとして補完し、具体化する。

6.2.1 プロセス開始の準備

- (1) 主ライフサイクルプロセス(5 参照)で作成されるものについて、6.2.2 に従って構成管理対象を識別する。
- (2) (1)で識別した構成管理対象(文書、ソースコードなど)に対し、管理開始時期、管理方法を定める。管理開始時期は、主ライフサイクルプロセス(5 参照)で定められた適切な時期を起点とする。
- (3) 構成管理対象の識別と変更を決定する責任者を決め、権限を明確にする。
- (4) 上記(1)～(3)を構成管理計画書として文書化する。なお、JMR-006 で作成するコンフィギュレーション管理計画書に含めてもよい。

6.2.2 構成識別

- (1) 機能・性能を定めたソフトウェア製品を構成管理対象として識別する。
- (2) 管理するソフトウェアを構成するモジュール(CSU 単位)ごとに識別子を付して、ソフトウェアの版に含まれるモジュールを識別できるようにする。
- (3) 文書類は、文書番号(識別子)と版番号を付して、管理する。詳細は文書化プロセス(6.1 参照)に従う。
- (4) 構成管理対象は、ソフトウェア製品修正時に修正箇所と識別子が対応するように適度に細分化された粒度(単位)にする。

6.2.3 構成変更管理

6.2.3.1 識別、検索および追跡

構成管理対象の識別、検索および追跡を行えるようにするため、試験、審査、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・版の識別ができるように管理する。

6.2.3.2 変更管理

- (1) 変更要求に対する検討結果(要求分析結果、設計書変更案)について、関連する設計者(要求設定担当、設計担当、製作担当)で下記を確認する。
 - (a) 要求と設計結果のトレーサビリティと適合性
 - (b) 設計文書間の整合性
 - (c) 設計変更と影響箇所
 - (d) 修正前後の版の識別(変更内容・理由を含む)
- (2) 協議の結果を含め、構成管理対象の識別と変更を決定する責任者が、設計変更を承認する。特に変更による検証結果の有効性、再検証範囲と方法の適切性を確認する。
- (3) 構成管理対象変更の内容は、変更要求と設計文書(設計結果)との対応の履歴が追えるように記録、保管する。
- (4) 構成管理対象の反映結果を確認する。

6.2.4 構成管理状況評価

構成管理計画書に従って構成管理が実施されていることを確認する。

6.2.5 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。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。

6.3 品質保証プロセス

品質保証プロセスは、ソフトウェア製品またはシステムのライフサイクルおよびプロジェクトのライフサイクルにおける各プロセスが規定要求事項に適合し、設定した計画に従って行われていることを保証するために実施する。

6.3.1 プロセス開始の準備

6.3.1.1 品質保証活動体制

品質保証活動を適切に実施するための体制を明確にする。

- (1) 技術部門から独立した責任と権限を持ち、品質保証活動の管理、実施状況の確認および改善などの活動を実行する責任者を任命する。
- (2) 品質保証活動に係わる組織の長および各担当者の役割と責任を明確にする。

6.3.1.2 ソフトウェア品質保証プログラム計画書

品質保証活動の計画として下記を明確にする。

- (1) 品質保証活動の体制
- (2) 品質保証活動およびその実施時期
- (3) 品質保証活動のレビュー対象
- (4) 品質保証活動の結果(品質記録)保管期限

ただし、他のソフトウェア開発と共通する事項は、共通のソフトウェア品質保証プログラム計画書を作成して適用してもよい。

6.3.2 製品の保証

- (1) ソフトウェア製品の構成が、ソフトウェア設計書で定められたものと整合することを確認する。
- (2) ソフトウェア製品の機能・性能が開発仕様書で規定する要求に適合していることを確認する。
- (3) 上記(1)(2)の結果は品質記録として保管する。
- (4) 上記(1)(2)は、計画した審査、ウォークスルーなどのレビューにおいても確認する。
- (5) 上記(1)(2)は、設計担当が作成した設計文書、試験成績書などの文書をレビューすることで実現してもよい。品質記録としては、レビューした文書類を認可することでもよい。
- (6) ソフトウェア製品に問題が発見された場合には、問題解決プロセス(6.8 参照)により処置する。
- (7) 必要に応じて独立検証・有効性確認 (IV&V) を実施し、製品品質を保証すること。

6.3.3 プロセスの保証

- (1) ソフトウェア開発計画書、ソフトウェア製品仕様書または保守の実施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。
- (2) ソフトウェア開発計画書、ソフトウェア製品仕様書または保守の実施計画で定義されたプロセスに従って、プロセスが実行されていることを確認する。各種の設計文書類が、計画どおりに作成されていることを確認する。
- (3) ソフトウェア製品の不適合処置、構成管理などのアクティビティが支援ライフサイクルプロセス(6 参照)に従って実行されていることを確認する。
- (4) 上記の(1)～(3)を実施する担当者が、必要な技術および知識を持ち、必要な教育・訓練を受けていることを確認する。

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。

6.3.4.1 教育・訓練

適用しない。(※)

※教育・訓練は、プロセスの保証(6.3.3 参照)の(4)に含まれるため。

6.3.4.2 購買管理

購買管理は、供給者がソフトウェアを外部に発注する場合(COTS を含む)、その信頼性・品質が品質要求事項に適合することを確認するアクティビティを定め、実施する。購買先の選定にあたっては、能力評価を実施する。

6.3.4.3 再利用ソフトウェア品目、COTS 品目の管理

- (1) 再利用ソフトウェア品目
既存ソフトウェア(COTS 品を除く)を再利用する場合には、下記を適用する。
 - (a) モジュール単位の再利用であるか、一部を改修して再利用するのかを明確にする。
 - (b) 再利用するソフトウェアに対する検証結果を有効とする範囲を明確にする。
 - (c) ソフトウェアの改修に影響しない機能の試験を省略し、実施済みの試験結果を適用する場合は、適用できる根拠を明確化する。
 - (d) 供給者が開発した既存ソフトウェアは、構成管理プロセス(6.2 参照)と同等の

管理がされているものに限って再利用可能である。

- (e) 既存ソフトウェアの再利用を可能にする評価項目は次の通りである。
 - (ア) 適用される要求事項とのトレーサビリティ
 - (イ) 入手可能または再利用可能な関連文書
 - (ウ) バージョンなどの識別と登録構成管理状況
- (2) COTS 品目

COTS 品目を利用する場合には、下記を適用する。

 - (a) 開発するソフトウェアの検証において、COTS 品目の動作をとまなう機能を試験する。
 - (b) COTS 品目の利用を可能にする評価項目は次の通りである。
 - (ア) 利用する COTS 品目の製品情報から得られるリスク
 - (イ) 受け入れおよび保証条件
 - (ウ) 入手可能または利用可能な関連文書
 - (エ) 保守および将来的支援
 - (オ) 専有権、使用权、所有権、ライセンス付与権、著作権
 - (カ) サポート終了後の取扱い

6.3.4.4 支給品などの管理

支給品などの物品を受け入れる場合には、媒体に記録されたソフトウェア製品の識別が、リリース文書(文書パッケージなど)に記載されているものと一致していることを確認する。

支給品などの物品を利用する場合には、再利用ソフトウェア品目、COTS 品目の管理(6.3.4.3 参照)を適用する。

6.3.4.5 取扱い、保管およびラベリング

ソフトウェア製品の取扱いおよび保管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。

- (1) 媒体に収録する場合には、作業に使用するソフトウェア製品のみを記録する。
- (2) 媒体に収録する場合には、記録したソフトウェア製品を識別するためのラベリング(名称と識別子など)をする。
- (3) 媒体からソフトウェア製品を読み込んだときに、ソフトウェア製品の識別を確認する。

6.3.4.6 スタンプ管理

署名あるいは承認印・検査印などのスタンプについて、保管方法・使用者などの管理運用規則を定め、規則に従ってスタンプを使用する。

6.3.4.7 品質記録

品質保証プログラムが適切に実行されたことおよびそれが効果的であることが検証できるように、ソフトウェア開発の全プロセスに渡って記録(品質記録)を作成し維持管理する手順を文書に定め、それに基づいて実行する。この手順には次の事項を含む。

- (1) 品質記録の種類

管理の対象となる品質記録の種類・名称
- (2) 品質記録の保管・維持

品質記録の保管および維持を行う担当者または組織

品質記録は、容易に検索できるように台帳または同等機能で管理する
- (3) 保管期間の設定

品質記録の保管期間

品質記録は、取得者の評価に利用できるように、適切に保管する。

6.4 検証プロセス

検証プロセスは、あるアクティビティの成果(アウトプット)が、先行するアクティビティで決められた要求(インプット)を満たしているかどうかを確認するためのプロセスである。

ここでは検証の観点を記述しているが、主ライフサイクルプロセス(5 参照)と重複する作業を改めて要求するものではない。

6.4.1 プロセス開始の準備

開発、運用および保守のプロセスにおいて品質要求事項を検証するアクティビティの計画を立案し文書化する。検証計画には、実施担当者を明確にする。必要であれば、組織の独立性を考慮する。

検証対象のプロセス、ソフトウェアまたはシステム、文書、データを示し、必要な資源、実施組織と責任、実施方法および技術、実施スケジュールなどの前提条件を明確にする。

6.4.2 検証

検証計画を実行する。検証作業で判明した問題および不適合（以降、「問題」という）は、問題解決プロセス(6.8 参照)で取り扱う。問題はすべて解決する。検証作業の結果は、取得者および関係する組織が利用できるようにする。

6.4.2.1 プロセスの検証

次に示す基準を考慮して、プロセスを検証する。

- (1) 選択されたプロセスが適切であることを確認する。
- (2) プロセスの記録をソフトウェア開発計画書に照らして、以下の観点で確認する。
 - (a) スケジュールが適切に計画されている。
 - (b) スケジュール通り作業が進捗している。
 - (c) 基準、手続き、開発環境の準備状況が整っている。
 - (d) 担当者は教育訓練されており、必要数確保されている。

6.4.2.2 要求事項の検証

次に示す基準を考慮して、要求事項を検証する。

- (1) 要求事項は、一貫性(矛盾が無いこと)があり、実現可能かつ検証可能である。
- (2) 要求事項は、ソフトウェア品目に対して適切に配分されている。(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) 要求事項は、上位の要求や品目に適用される基準に適合している。
- (4) 品質特性(運用性、可読性、運用環境・ミッション適応性、ロバスト性、安全性およびセキュリティなど)に関するソフトウェア要求仕様書が、ソフトウェアが果たす機能および重要性に照らしてふさわしいものとなっている。安全性・セキュリティなど特に注意を必要とする要求事項は、上位の要求や品目に適用する基準に適合していることが、適切な手法により示されている。

6.4.2.3 設計の検証

次に示す基準を考慮して、設計を検証する。

- (1) 設計仕様は要求に適合し、要求に対してトレーサビリティがある。
- (2) 設計仕様は、入出力インタフェース・タイミング・ハードウェアリソース(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・終了など)に関し、適切に設定されている。
- (3) 設計仕様は、運用性・保守性などの特性について、適切に考慮されている。

- (4) 設計仕様は品質特性(運用性、可読性、運用環境・ミッション適応性、ロバスト性、安全性およびセキュリティなど)の観点で、それぞれの要求仕様に照らして過不足がなく、適切に設定されている。
- (5) アルゴリズムやデータなどの設定根拠が文書化されており、それらはシステムなどの上流側の意図に合致して妥当な内容である。
- (6) 設計仕様は、オフノミナルな変動要因などの異常事象に遭遇した場合も含め、想定されるすべての要因を考慮している。

6.4.2.4 ソースコードの検証

次に示す基準を考慮して、ソースコードを検証する。

- (1) ソフトウェア設計書とソースコードのトレーサビリティが確認されている。(ソフトウェア設計書の改訂が漏れなくソースコードに反映されている)
- (2) コーディング規約に従ってコーディングされている。

6.4.2.5 統合の検証

適用しない。(※)

※統合の検証は、ソフトウェア統合試験(CSCI 試験)(5.4.9 参照)に含まれるため。

6.4.2.6 文書化の検証

適用しない。(※)

※文書化の検証は、プロセスの検証(6.4.2.1 参照)に含まれるため。

6.5 妥当性確認プロセス

妥当性確認プロセスは、ソフトウェアに対する要求を設定する組織に、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。その手段は、設計書のレビュー、試験計画書、試験成績書の確認などで行ってもよく、試験を別途に実施することは必須ではない。

6.5.1 プロセス開始の準備

ソフトウェア開発計画書において、妥当性確認の実施時期、実施方法、実施体制を定める。

6.5.2 妥当性確認

計画に従い下記を行う。

(1) 確認方法

想定されるオフノミナルな変動要因などの異常事象への対応が十分であるかの確認も含める。

開発仕様書などの設計書のレビュー、検証項目のレビュー、検証試験成績書のレビューなどの方法で行ってよい。ソフトウェアに対する要求を設定した部門の独立した検証を要求するものではない。

レビューの観点は、要求する機能が、ソフトウェア製品で実現されていることを確認する。

(2) 確認結果

(a) 確認結果は品質記録として保管する。

(b) 問題は問題解決プロセス(6.8 参照)により処置する。

6.6 レビュープロセス

レビュープロセスは、ソフトウェア開発の状況およびアウトプットを、適宜、評価するためのプロセスである。レビューでは、ソフトウェア開発の管理面および技術面の両面を確認する。

6.6.1 プロセス開始の準備

6.6.1.1 レビュー計画の立案

管理面および技術面のレビュー計画を立案する。このレビュー計画はソフトウェア開発計画書に含めてもよい。システムのレビューにソフトウェアを含めて実施する場合、ソフトウェアに関する事項はシステムのレビュー計画に含めてもよい。

レビュー計画には以下の観点を含め、レビュー実施者、受審者の両当事者は、レビュー計画について合意する。

購買先が行うレビューについてもレビュー計画を立案する。

- (1) 実施時期
- (2) 実施形式や方法
- (3) 参加メンバーおよびその役割
- (4) 指摘事項などのレビュー結果および実施記録の処理手順
- (5) レビュー対象物、レビュー項目および評価基準

6.6.2 管理面のレビュー

- (1) 作業状況を、ソフトウェア開発計画書、標準および指針と照らして評価する。評価記録を文書化する。
- (2) 評価記録に基づき、必要に応じて、リソース再配分などの是正策を決定する。

6.6.3 技術面のレビュー

6.6.3.1 審査

ソフトウェア開発プロセスにおける審査は、該当するプロセスの作業が完了したことを確認し、ソフトウェア開発計画書で定義されたライフサイクル上の次のプロセスに移行できることを判断するためのものである。

- (1) 準備
 - (a) 審査を実施するにあたり、審査員を選定する。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)を含むように選定する。
 - (b) 審査対象、目的を明確にして、事前に審査に必要な資料が完成していることを確認する。
 - (c) 審査目的、審査対象、審査員を明確にし、文書化する。
- (2) 実施
 - (a) ソフトウェアについて、開発進捗に応じて段階的に評価し、次の事項を明らかにする。
 - (ア) 取得者が求める機能・性能を発揮する。
 - (イ) 要求仕様、設計仕様および製造文書に整合している。
 - (b) ソフトウェアに関する技術的活動を評価し、次の事項を明らかにする。
 - (ア) 作業指示文書および技術指示文書に従っている。
 - (イ) 適用される標準類に従っている。
 - (ウ) 次のプロセスに移行可能になっている。

※審査の例としては下記のようなものが挙げられる。

- (1) ソフトウェア要求審査(Software Specification Review : SSR)
- (2) 基本設計審査(Preliminary Design Review : PDR)
- (3) 詳細設計審査(Critical Design Review : CDR)
- (4) 認定試験後審査(Post Qualification Test Review : PQR)

6.6.3.2 ウォークスルー

ウォークスルーは、設計や製作の問題を早期に発見・除去し、生産性と品質を向上させる狙いから、必要に応じ、実作業担当者が中心に実施する。

- (1) 準備

作成中のものも含み、対象となる文書・ソースコードを提供する。
- (2) 実施方法
 - (a) 必要な関係者(前後のプロセス実行者など)が、対象となる文書・ソースコードに問題がないことを確認する。確認は下記のような観点で行う。
 - (ア) 関係者間の理解が一致していること。
 - (イ) 適用されている基準に適合していること。
 - (ウ) 要求に対して、対象となる文書・ソースコードに漏れ・矛盾などの誤りがないこと。
 - (b) ウォークスルーにより判明した質問事項、問題は記録に残し、対策が完了するまでフォローする(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)。
 - (c) 対象となる文書・ソースコードの作成期間中であっても、ソフトウェアの機能ごとなど、細かな単位の作業完了時ごとに実施してもよい。
 - (d) ウォークスルーは審査会時に実施してもよい。

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。

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。

- (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。
- (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。
- (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。

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。

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判明したあらゆる種類の、あらゆる発生源の問題(不適合を含む)を分析し、解決するためのプロセスである。

6.8.1 プロセス開始の準備

- (1) 必要に応じて問題の種類、重要度など、分類の基準を定める。
- (2) 問題の分類に応じた解決策の決定組織、責任および権限を定める。

6.8.2 問題解決

ソフトウェアの機能・性能に関する問題発生に関して下記を実施する。

- (1) 問題の内容を分析・調査し、設計誤り箇所を特定する。
- (2) 設計誤りに至った要因を分析し、類似の設計誤りが想定される箇所について調査・確認する。
- (3) 上記(1)(2)の調査の結果、判明した設計誤りの修正方法を検討し、設計変更が必要となる設計文書と修正内容を明確にする。
- (4) 上記(3)の結果より、恒久対策を検討し、要すれば手順書、設計規約などの文書に反映し再発防止を図る。
- (5) 試験を実施済みの場合には、設計誤りが影響する試験項目を明確にし、再試験内容を試験要求としてまとめる。
- (6) 上記(5)の試験要求に基づき試験を実施する。試験は、初度を実施した開発段階の設計基準に従い実施する。
- (7) 設計文書およびソースコードの修正は、ソフトウェア開発プロセス(5.4 参照)の該当する定義に従い実施する。
- (8) 上記(1)～(7)の結果は、レビュープロセス(6.6 参照)に従い審査する。審査は、(1)～(7)の中で適切な時期に分割して実施してもよい。処置規模に応じて、審査の実施時期、回数は決定してもよい。
- (9) 問題が判明した場合は問題報告書を作成する。また、問題の分類に応じて、品質指標の取得などのアクティビティに影響を与えない範囲で簡素化した形式としてもよい。
- (10) 不適合の内容は、品質記録として保管するとともに、発生傾向を分析し、一定の傾向が見られた場合には、対策を講じる。
対策は、設計規約の修正、作業員への教育、プロセスの変更、使用ツールの改善など様々が考えられるので、必要な対策は複数講じる。
- (11) 取得者および供給者は問題の重要性に応じ、問題の内容および解決策などの情報を不適合データベースあるいは教訓データベースなどのデータベースに蓄積し、活用する。

付録

付録1 CSCI 分割ガイドライン

非常に小さなシステムではソフトウェア要素の全体が1つのCSCIとなりうるが、通常はシステムには複数のCSCIが割り当てられる。不適切なCSCIの選択は、システム全体のコスト、納期、品質に対して悪影響を与えるため、CSCIの分割には十分な留意を払う必要がある。CSCI分割のガイドラインを以下に述べる。

1. 複数のCSCIに分割するケース
 - (1) 異種プロセッサ上の機能
異種プロセッサ上に搭載される機能は、別のCSCIとする。ただし、同一のソフトウェアの複製であれば、1つのCSCIとなる。
 - (2) 複数の開発業者
複数の開発業者に割り当てられた機能は、それぞれ別個のCSCIとする。

2. 1つのCSCIを設定するケース
 - (1) クリティカルな機能
1つの主要機能は、その故障がセキュリティや安全性に悪影響を及ぼす場合、その機能は1個のCSCIとする。
 - (2) 複数のシステムに対して共通な機能
複数のシステムに対して共通な機能は、それがシステムの主要なコンポーネントであれば1個のCSCIとする。
 - (3) 複数CSCI間のインタフェース管理
複数のCSCI間のインタフェースを制御する機能は、1つのCSCIとする。
 - (4) 新技術の利用
最新の技術(例、エキスパートシステム)を使用して実現される機能は、1つのCSCIとする。
 - (5) 頻繁な変更
今後の頻繁な変更または開発後に変更が予想される主要機能は、1つのCSCIとする。
 - (6) 凝結した機能
データや制御に関して高度に相互依存性がある機能は、1つのCSCIとする。

3. CSCI分割結果の評価
分割されたCSCIは、次の条件を満たしていることが望ましい。(満足していない場合は、再分割または分割のやり直しの必要性が高い。)
 - (1) CSCI間の関係(インタフェースの最小化)
CSCI間のインタフェースは、極小化され簡潔でなければならない。即ち、CSCIの機能間にはできるだけ少ないインタフェースが望ましい。
 - (2) CSCI内の機能(凝結した機能)
1つのCSCIに割当てられた下位の機能は、密接に関連していなければならない。

付録2 ソフトウェア開発管理ガイドライン

管理手順は次の事項が実現できる仕組みとする。

- (1) 定義したプロセスモデルに従い、各工程の開始時には準備が整っていること、終了時には次工程に移行できることを確認し、工程の進行を管理する。
- (2) 各工程の進行状況などのプロジェクトの進行を定量的に把握し、個別スケジュール及びマスタースケジュールの計画と照らし合わせて評価する。
- (3) 計画とずれが生じている場合、その原因を究明し、是正処置を講ずる。
- (4) スケジュールに影響を及ぼすシステム等の外部の進行状況を、6.9.4 項のインタフェース管理におけるインタフェース調整などの場を通して的確に把握し、プロジェクトの計画に適時適切に反映する。
- (5) システム等の外部へ影響を与える問題を識別し、的確に通知する。
- (6) 進行状況の報告先を予め定め、定期的に報告する。

付録3 関連文書一覧

5.3 システム検討プロセス

プロセス	インプット	アウトプット
5.3.1 システム要求分析	(1) 上位要求(取得者のニーズまたは上位のシステム要求仕様)	(1) システム要求仕様書(サブシステムの場合「サブシステム要求仕様書」、コンポーネントの場合「コンポーネント要求仕様書」) (2) 各システム要求仕様の根拠 (3) 上位要求に対するトレーサビリティの評価記録
5.3.2 システム方式設計	(1) システム要求仕様書	(1) システム仕様書(機体システムの場合「機体システム仕様書」、サブシステムの場合「サブシステム開発仕様書」、コンポーネントの場合「コンポーネント開発仕様書」) (2) 各構成品目への要求(サブシステムおよびコンポーネントの開発仕様書、ソフトウェアに対する要求) (3) インタフェース管理仕様書 (4) 各構成品目への要求の実現に対するリスク評価記録 (5) システム仕様の設計根拠(前提条件など含む) (6) システム仕様の設計根拠評価記録 (7) システム仕様書とシステム要求仕様書とのトレーサビリティ評価記録 (8) インタフェース管理仕様書とシステム要求仕様書とのトレーサビリティ評価記録

5.4 ソフトウェア開発プロセス

プロセス	インプット	アウトプット
5.4.3 ソフトウェア要求分析	(1) システム仕様書(機体システムの場合「機体システム仕様書」、サブシステムの場合「サブシステム開発仕様書」、コンポーネントの場合「コンポーネント開発仕様書」) (2) インタフェース管理仕様書 (3) ソフトウェアに対する要求(各構成品目への要求に含まれる)	(1) ソフトウェア要求仕様書(ソフトウェア開発仕様書に含まれる場合がある) (2) ソフトウェア要求仕様書とシステム仕様書・インタフェース管理仕様書とのトレーサビリティ評価記録 (3) ソフトウェア要求仕様書の根拠 (4) ソフトウェア開発仕様書(組込みソフトウェアではコンポーネント開発仕様書に含まれる場合がある) (5) ソフトウェア開発仕様書とソフトウェア要求仕様書とのトレーサビリティ評価記録 (6) COTS・再利用ソフトウェアとソフトウェア開発仕様書との適合性評価記録 (7) ソフトウェア検証計画書 (8) ソフトウェア検証計画書の検証可能性および検証網羅性評価記録 (9) ソフトウェア要求審査記録

5.4.4 ソフトウェア設計	(1) ソフトウェア要求仕様書(ソフトウェア開発仕様書に含まれる場合がある) (2) ソフトウェア開発仕様書(組込みソフトウェアではコンポーネント開発仕様書に含まれる場合がある) (3) ソフトウェア検証計画書	(1) ソフトウェア(基本、詳細)設計書 (2) ソフトウェア設計書とソフトウェア開発仕様書のトレーサビリティおよび適合性評価記録 (3) ソフトウェア設計根拠および実現可能性評価記録 (4) COTS・再利用ソフトウェアとソフトウェア設計書との適合性評価記録 (5) ソフトウェア試験計画書 (6) 組合せ試験(CSC 試験)仕様書 (7) ソフトウェア統合試験(CSCI 試験)仕様書 (8) ソフトウェア設計審査記録
5.4.6 ソフトウェア製作	(1) ソフトウェア(詳細)設計書 (2) ソフトウェア試験計画書	(1) ソースコード (2) 単体試験(CSU 試験)仕様書 (3) 単体試験(CSU 試験)記録 (4) 静的解析結果 (5) ソースコードとソフトウェア設計書のトレーサビリティ評価記録 (6) ソフトウェア製作審査記録
5.4.8 ソフトウェア統合	(1) ソースコード(CSU) (2) ソフトウェア設計書 (3) ソフトウェア試験計画書 (4) 組合せ試験(CSC 試験)仕様書	(1) ソースコード(CSCI) (2) オブジェクト(実行形式)コード (3) ソフトウェア統合記録 (4) ソフトウェア試験計画書(更新) (5) 組合せ試験(CSC 試験)仕様書(更新) (6) 組合せ試験(CSC 試験)手順書 (7) 組合せ試験(CSC 試験)成績書
5.4.9 ソフトウェア統合試験(CSCI 試験)	(1) オブジェクト(実行形式)コード (2) ソフトウェア開発仕様書 (3) ソフトウェア試験計画書 (4) ソフトウェア統合試験(CSCI 試験)仕様書	(1) ソフトウェア試験計画書(更新) (2) ソフトウェア統合試験(CSCI 試験)仕様書(更新) (3) ソフトウェア統合試験(CSCI 試験)手順書 (4) ソフトウェア統合試験(CSCI 試験)成績書 (5) ソフトウェア試験審査記録
5.4.11 ソフトウェア製品の供給と導入	(1) ソースコード (2) オブジェクト(実行形式)コード (3) ソフトウェア(基本、詳細)設計書 (4) ソフトウェア検証結果 (5) ソフトウェア試験成績書 (6) 要処置事項ステータス (7) 妥当性確認結果	(1) 運用手順書(運用データ設定手順などを含む) (2) ソフトウェア製品収録媒体作成手順書(コンパイラオプションなどの手順を含む) (3) ソフトウェア製品仕様書 (4) ソフトウェア製品取扱説明書 (5) ソフトウェア製品(収録媒体またはハードウェア組み込み状態) (6) ソフトウェア製品試験計画書 (7) ソフトウェア製品試験仕様書 (8) ソフトウェア製品試験手順書 (9) ソフトウェア製品試験成績書(合否判定結果を含む) (10) ソフトウェア製品リリース文書(文書パッケージなど)
5.4.12 ソフトウェア製品の取得	(1) ソフトウェア開発仕様書 (2) ソフトウェア製品仕様書 (3) ソフトウェア製品(収録媒体またはハードウェア組み込み状態) (4) リリース文書	(1) 受入検査実施要領書(受入検査・試験計画、受入検査・試験仕様などを含む) (2) 受入検査・試験成績書 (3) リリース文書(受入検査記録を含む) (4) ソフトウェア製品(受入検査済み)

5.5 システム検証プロセス

プロセス	インプット	アウトプット
5.5.1 システム試験	(1) システム要求仕様書 (2) システム仕様書 (3) インタフェース管理仕様書 (4) ソフトウェア要求仕様書 (5) ソフトウェア検証計画書	(1) システム試験計画書 (2) システム試験仕様書 (3) システム試験手順書 (4) システム試験成績書
5.5.2 運用試験	(1) システム要求仕様書 (2) システム仕様書 (3) インタフェース管理仕様書 (4) ソフトウェア要求仕様書 (5) ソフトウェア検証計画書	(1) 運用試験計画書 (2) 運用試験仕様書 (3) 運用試験手順書 (4) 運用試験成績書