RMD-110 Rev.01

国立研究開発法人新エネルギー・産業技術総合開発機構
（NEDO）
次世代空モビリティの社会実装に向けた実現プロジェクト
（ReAMo プロジェクト）

無人航空機の型式認証等の取得のためのガイドライン

安全基準セクション 110 ソフトウェア　解説書

2024 年 3 月

無人航空機の認証に対応した証明手法の事例検討
110 サブ WG ソフトウェア

目次

図 目次

表 目次

1　目的

本解説書は「無人航空機の型式認証等の取得のためのガイドライン」（以降、「航空局ガイドライン」という）安全基準「セクション 110 ソフトウェア（以降、「セクション 110」と呼ぶ）」に対する解説書である。

なお、解説対象とする文書は国土交通省航空局から 2022 年（令和 4 年）12 月 2 日発行時点の航空局ガイドラインとする。解説対象に関する詳細は本冊（RMD　Rev.01）1.2 を参照すること。

2　対象の基準「サーキュラー」（引用）

「サーキュラーNo.8-001“無人航空機の型式認証等における安全基準及び均一性基準に対する検査要領“（以降、「サーキュラーNo.8-001」と呼ぶ）」の「110 ソフトウェア」を以下に引用する。

> *・110 ソフトウェア*
>
> *残存するソフトウェアエラーを最小化するために、申請者は以下を行わなければならない。*
>
> *（a）　無人航空機の安全な運用に影響を与えるすべてのソフトウェアに対して試験による検証*
>
> *（b）　ソフトウェアの全ライフサイクルを通した変更に対する追跡、管理及び保存を行うための形態管理システムの使用*
>
> *（c）　ソフトウェアの修正及び欠陥を捕捉し記録するための PR（Problem Report）システムの導入及び活用*

3　「航空局ガイドライン」（引用）

「航空局ガイドライン」安全基準「110 ソフトウェア」の「基準の概要」、「適合性証明方法（MoC）」、「その他参考となる情報」を以下に引用する。

> *・110 ソフトウェア*
>
> *<u>基準の概要</u>*
>
> *本基準は、ソフトウェアエラーの残存を最小化するために必要となる活動を要求するものです。セクション 110 では、まずソフトウェアに対し試験（テスト）で要求が適切に実装されていることの確認を行います。なお、本テストはシステムレベルの要求に対して行います。続いてバージョン（Ver.）が刻々と変化する可能性の高いソフトウェアに対し、形態管理は重要であるため、どのようなソフトウェアが各型式に搭載されているのか（追跡）、適切なソフトウェアなのか（管理及び保存）について、ライフサイクルを通して維持・管理できることが要求されます。最後にソフトウェアのエラーを把握し記録、必要に応じた修正を行うための PR システムが必要となります。*
>
> *なお、有人航空機においてはソフトウェアに対し開発保証として多くの場合 RTCA DO-178 に基づいた活動が要求されます。適合性証明方法として DO-178（DAL D）を用いることもできます。*

適合性証明方法(MoC):1、2

(a),(b),(c): セクション 110 ソフトウェア適合性証明計画/完了報告書 (MoC 1, 2)

セクション 110 への適合性を証明するための計画をまとめたセクション 110 ソフトウェア適合性証明計画を作成します。計画は、最終的には計画どおりに完了したことを記す完了報告書となります(計画と完了報告書の 2 文書が必要)。

無人航空機ソフトウェア適合性証明計画書には以下を記載します:

まず(a)項に対し、「安全な運用に影響を与えるソフトウェア」とは何か、無人航空機に使用される全ソフトウェアから抽出する必要があります。抽出の方法は、セクション 135 を準用してソフトウェアが誤った挙動をした場合の影響の程度を評価する方法、FHA (Functional Hazard Analysis)、SSA (System Safety Assessment)、FMEA (Failure Mode and Effect Analysis)等の安全性解析手法を用いるなど、いくつかあります。また、対象となるソフトウェアの抽出はせず、無人航空機に使用されるすべてのソフトウェアに対し試験を行う方法もあります。

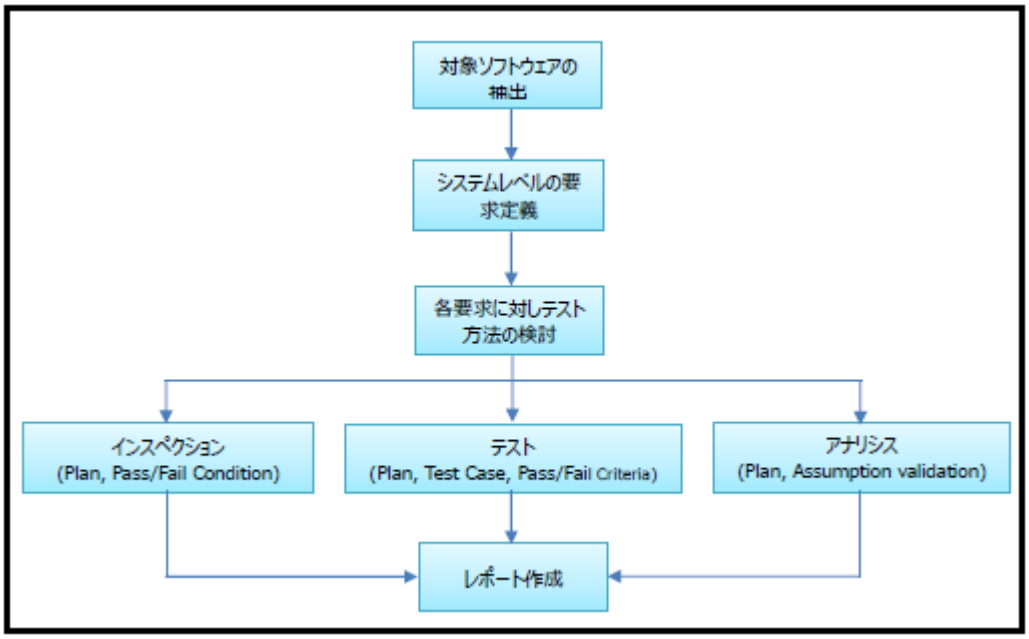
続いて、「安全な運用に影響を与えるソフトウェア」は「システムレベルのテスト」によりその動作を確認する必要があります。そのため、「システムレベルの要求」を定義する必要があります。システムは日本語では系統と訳せますが、無人航空機は様々なシステムから構成されています。例えば自機位置を把握する GNSS システム、飛行制御を行うフライトコントロールシステムなどがあります。システムレベルの要求は、それぞれのシステムが満たすべき一つひとつの要求です。例えば GNSS システムであれば位置情報を正しく出力することがひとつの要求となります。なお、ソフトウェアのテストには、ホワイトボックステスト、ブラックボックステスト、単体テスト、統合テスト、これ以外の分類も含め多種多様に存在しますが、要求ベースのテストとは、システムレベルの要求が正しくソフトウェアに実装されていることを確認するテストとなります。そのため、一般的にはブラックボックステストが該当し、試験装置によるテスト(ベンチテスト)及び機体レベルで行う地上試験が主体となります。このほか、セクション 300 で行う飛行試験及びベンチテスト、地上試験との組合せで確認できる要求もあります。

すべての要求は、基本テストにより確認される必要がある一方、テストでの確認が困難な要求に関しては解析(アナリシス)や検査(インスペクション)といった方法も許容されます。例えば、非機能要求(許容できるメモリ量、CPU 負荷など)は解析(アナリシス)により確認される要求となります。検査(インスペクション)は一般的に目視、聴覚や触覚などの感覚によって行う非破壊の評価となり、物理的な測定や操作などが該当します。

なお、要求からテストケース及び手順へはトレーサビリティを確保する必要があります。

システムレベルのテストについて、ASTM F3153-15 "Standard Specification for Verification of Avionics Systems"を参考にすることができます。

以上の活動を図示すると以下のようになります:



上記活動をもとに無人航空機ソフトウェア適合性証明計画書/完了報告書には以下を記載します(証明活動の結果は完了報告書のみに記載):

- *無人航空機のシステム概要(他の文書を引用可)*
- *ソフトウェア一覧(搭載、非搭載別)*
- *安全な運用に悪影響を与えるソフトウェアをどのように抽出するかの説明及びその結果*
- *システムレベル要求ベーステストをどのように実施するかの説明(計画)*
- *システムレベル要求ベーステストの結果概要*

また、完了報告書を補完する文書として以下が必要となります。

- *システムレベルの要求一覧*
- *システムレベル要求ベーステスト関連書類及び試験結果*

続いて(b)項に対し、ソフトウェアのライフサイクルを通した変更に対する追跡、管理及び保存を行うための形態管理システムが適用されることを提示します。

なお、対象となるソフトウェアは(a)項の対象と同じく安全な運用に影響を与えるソフトウェアです。

本項を満たすためには、形態管理がどのように行われるのか概要を説明するとともに、以下のアイテムについて形態管理されることの説明が必要になります。

i. 要求(Requirements)

ii.システム及びソフトウェアテスト環境の説明

(System and software test environment descriptions)

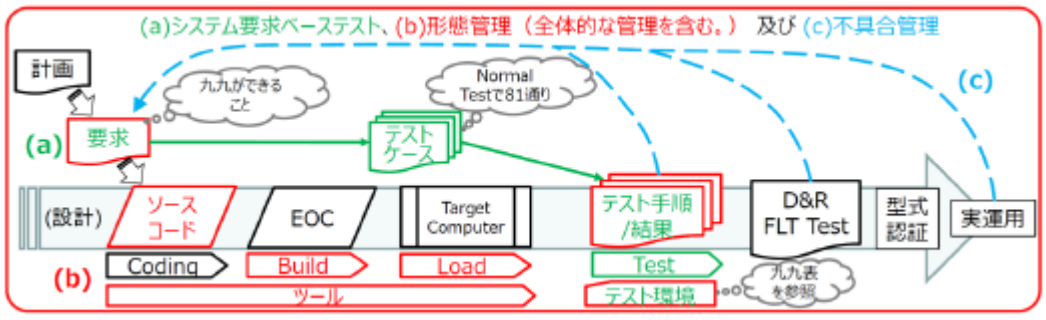
iii.要求からテストケース及び手順へのトレーサビリティを含むテスト手順及び結果
(Test procedures, and results with requirements traceability to test cases and procedures)
iv.ソースコード及び開発環境/ツール
(Source code and development environment/tools)
v. 実行オブジェクトコードの複製のためのビルド/ロード手順
(Build and load procedures for replication of the executable object code)
形態管理は、ベースラインとして形態管理のスタート地点を定める必要がありますが、遅くとも(a)項のテストの開始前にはスタートする必要があります。

最後に(c)項に対し、ソフトウェアの修正及び欠陥を捕捉し記録するための PR(Problem Report)システムが適用されることを提示します。
なお、対象となるソフトウェアは(a)項の対象と同じく安全な運用に影響を与えるソフトウェアです。
本項を満たすためには、不具合管理がどのように行われるのかの説明が必要となります。

自己宣言について
「サーキュラー」の表 1 で求める自己宣言について以下に記します。
自己宣言とは、一部の認証区分において、航空局又は登録検査機関によるセクション 110 ソフトウェアに対する適合性の検査は受けないものの、申請者自身がセクション 110(a)、(b)及び(c)の各要件に対して適合していることを確認した上で、適合している旨を記載した宣言書を型式認証申請後に提出することが必要です。

※「検査のポイント」および「検査者の関与度(LOI)」については引用記載しない

その他参考となる情報
以下は(a)、(b)及び(c)項についての参考図です。
緑色が(a)項、赤色が(b)項、青色が(c)項に関するものとなります。



産業規格

- *RTCA DO-178C, Software Considerations in Airborne Systems and Equipment Certification*
- *ASTM F3153-15, Standard Specification for Verification of Avionics Systems*

4　解説書

4.1　セクション 110 ソフトウェアの対象と範囲

（a）無人航空機の安全な運用に影響を与えるすべてのソフトウェアに対して試験による検証
［引用:サーキュラー No.8-001］

サーキュラーNo.8-001 によると、特にセクション 110（a）の範囲として、「*無人航空機の安全な運用に影響を与えるすべてのソフトウェアに対して*」、とある通り、セクション 110 の活動対象は、無人航空機の安全な運用に影響を与える機体およびすべての関連システムに含まれるソフトウェアに対して活動を行う。（フライトに関するソフトウェアにおいても、機体に搭載される場合や、機体に搭載されるかわりにクラウドなどに置かれ、そのクラウドに置かれたソフトウェアが運用の際に使用される場合もある。）

4.2　セクション 110 ソフトウェアの活動

（1）　基準に対する活動要求とその理由

本基準は、ソフトウェアエラーの残存を最小化するために必要となる活動を要求するものです。”
［引用: 航空局ガイドライン］

セクション 110 で行うべき認証としての活動はソフトウェアの品質を確保するために行われなければいけないソフトウェア開発プロセスの構築である。つまり、ソフトウェアの品質の確保を行うための「プロセス」を構築し、そのプロセスに基づいてソフトウェアを開発することがセクション 110 で認証される活動である。

電子的なプログラムで動く無人航空機にとって、ソフトウェアは非常に重要な部位である。セクション 110 は、システム全体からの要求を充足する品質を持つソフトウェアを構築し、かつ適正にそれらが構成されていることを証明するために必要となる活動を定義するものである。これらの要求や、品質の充足を管理するには、他のセクションから来る要求や他のセクションへの活動の委譲を理解して、型式認証全体の一活動であるとして対処されるべきである。一方で、無人航空機上のソフトウェアも、関連シ

ステムなどの機体外のソフトウェアも、完全なソフトウェアを目指して構築されるが、システム実装上の不具合が生じることが常である。このため、品質の充足のためには、システムズエンジニアリングに基づく開発プロセスの組織化が必要である。セクション 110 で要求される 3 つの活動は以下のとおりである。

(a)　システムレベル要求ベーステスト

（a）無人航空機の安全な運用に影響を与えるすべてのソフトウェアに対して試験による検証

［引用:サーキュラー No.8-001］

(b)　形態管理

（b）ソフトウェアの全ライフサイクルを通した変更に対する追跡、管理及び保存を行うための形態管理システムの使用

［引用:サーキュラー No.8-001］

(c)　不具合管理

（c）ソフトウェアの修正及び欠陥を捕捉し記録するための PR(Problem Report）システムの導入及び活用

［引用:サーキュラー No.8-001］

(2)　他セクションとの関係性

セクション 110 は、ソフトウェア開発におけるプロセスと管理手法に対する「プロセス認証」が主である。一方で、実際のソフトウェア開発では、航空局ガイドラインセクション 110 その他参考となる情報における参考図(図 4.2-1)で示される通り、計画を策定し、要求を導出し、その要求を充足するようにシステム、コンポーネント、コードへと細分化して設計し、コードレベルから対応する単体テスト、結合テストを経て、システム全体のテスト、最終的な飛行テストまで行って型式の認証が成される。この開発の流れの中で、(a)、(b)、(c)の項目について活動を定義し、ソフトウェアエラーを低減させることがセクション 110 の要求となっている。

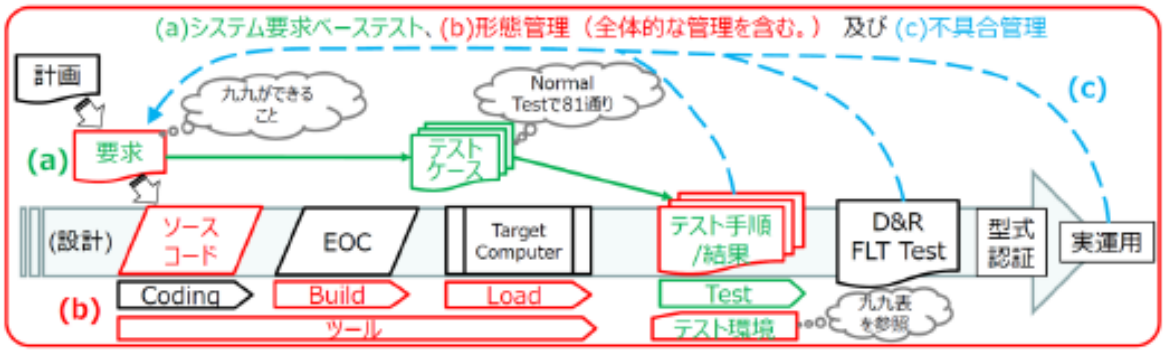


図 4.2-1　航空局ガイドラインセクション 110 その他参考となる情報における参考図

［引用:航空局ガイドライン］

このような無人航空機システム全体の計画と要求は他セクションから導かれる。(図 4.2-2 セクション 110 と他セクションとの関係)は、システム開発における V&V(Verification & Validation:検証と妥当性確認)プロセスに型式認証のプロセスを被せた形によるセクション 110 と他セクションとの関係を表した模式図になる。また、赤線はソフトウェア開発における V 字プロセスに関係する入力を表現している。

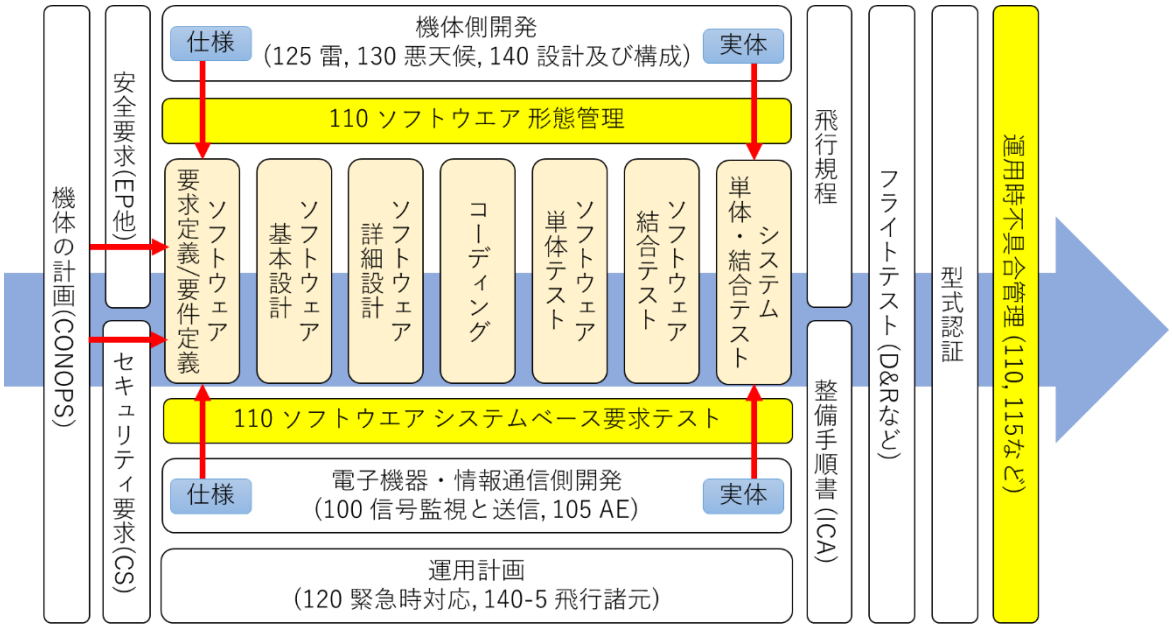


図 4.2-2 セクション 110 と他セクションとの関係

無人航空機システムに対するソフトウェアの開発は、機体の計画段階から始まり、必要な機能、やり取りされる情報などが設計概念書(CONOPS)として定義される。この機体の計画に対して事前に安全性およびセキュリティに対してリスクアセスメントを行い、それぞれ必要な要求が導出されて計画が更新される。これらの情報から、機体側開発、ソフトウェア開発、電子機器・情報通信機器開発および運用計画にそれぞれ分解されて、各パートでシステム開発が行われる。一方で、ソフトウェア開発は、機体を動作させるようなものや、センサーなど電子機器情報を受け取り動作するもの、情報通信機器を介してコミュニケーションを取りながら動作するものなどがあるので、それらハードウェアに起因する仕様を受けてソフトウェア開発が始まり、システムベースの要求が充足するかをハードウェアの実体と併せて用いてテストを行うことが必要であることから、他セクションで定義されているハードウェアの仕様または実体と連携しながらソフトウェアが構築されるべきである。

セクション 110 において、形態管理は、そういった他セクションからの情報を受けて、いかに要求を充足するようにソフトウェアを開発するかについて追跡するものである。また、システムベース要求テストは、他セクションから発生する部分も含めた要求について考察され、それを充足するシステムの挙動が何かを定義し、開発されたソフトウェアとともにテストを行うものである。

不具合管理については、現実にはソフトウェア起因ではない不具合の可能性も存在する。ソフトウェア開発中においては、開発中に行われるソフトウェア単体・結合テストに対する不具合のレポーティングとして、供与後においては、ソフトウェア起因ではない不具合も含めてレポートとして受け取り、形態

管理とともにソフトウェア起因による不具合であるかを解析するものとして用いることができる。
他セクションとの関連性についての例示を以下の通りに記述する。

- セクション 001 概念設計書(CONOPS)(以降、「セクション 001」と呼ぶ)
  - セクション 110 への入力
    - ユースケースから機能要求の導出
    - シナリオから品質要求の導出
    - 初期的な安全性解析から安全要求の導出
    - 初期的なセキュリティ解析からセキュリティ要求の導出
- 安全性リスクアセスメント – セクション 135 重要な部品(フライトエッセンシャルパーツ)(以降、「セクション 135」と呼ぶ)
  - セクション 110 への入力
    - 機能安全アセスメントによる安全要求の導出と追加要求の導出
    - エッセンシャルパーツにおけるシステムレベルでの単一故障における安全性解析
    - 機能ハザード解析や HAZOP ガイドワードによる解析結果
  - セクション 110 とのギャップ
    - セクション 135 で提示される安全性要求は、当該するシステムが安全に関与するか、どのような故障モードが存在するかまでである。そのため、詳細なソフトウェアに対する安全性解析はソフトウェア要求の段階で行う必要がある。この部分はソフトウェア基本設計のフェーズで行うこと
- セキュリティリスクアセスメント – セクション 115 サイバーセキュリティ(以降、「セクション 115」と呼ぶ)
  - セクション 110 への入力
    - セキュリティ要求に資するシステム開発項目の列挙
    - セキュリティ要求に資するシステムテスト項目の列挙
  - セクション 110 からの出力
    - システムコンポーネント間におけるデータフローの記述およびインターフェース要求の列挙
    - セクション 110 で開発される各コンポーネントにおけるセキュリティ要素の列挙
  - セクション 110 とのギャップ
    - 「悪意のある行動」に基づく安全性の侵害の考察はセクション 115 で行うことを前提とする。解析の結果導出された要求事項のうちソフトウェアに関するものはセクション 110 に

関する追加要求として受理すること
- 開発時における「悪意」もセクション 115 で解析され、ソフトウェア開発時の組織やコーディング規約などの必要な要求が導出される場合、その要求をソフトウェア開発要求に追加すること

- 機体以外のソフトウェア – セクション 105 無人航空機の安全な運用に必要な関連システム（以降、「セクション 105」と呼ぶ）

  ■ セクション 110 への入力

  - 関連システムとされる他のシステムに対してのデータフロー、およびソフトウェア機能要求
  - 関連システムと機体システムのインターフェース要求
  - 関連システム内ソフトウェアのコンポーネント情報
  - 安全性、品質の要求

  ■ セクション 110 からの出力

  - 関連システムとされる他のシステムが 110 ソフトウェアで管理するソフトウェアシステムの範疇外となる場合、その場合における保証の責任関係とインターフェース要求・データの流出入の要求を記述

- 機体側・電子機器側開発 – セクション 100 無人航空機に係る信号の監視と送信（以降、「セクション 100」と呼ぶ）、セクション 105、セクション 125 雷（以降、「セクション 125」と呼ぶ）、セクション 130 悪天候（以降、「セクション 130」と呼ぶ）、セクション 140 その他必要となる設計及び構成（以降、「セクション 140」と呼ぶ）

  ■ セクション 110 への入力

  - 機体側で定義される物理的なパラメータの定義（角度、回転数、制御パラメータ）
  - 電子機器側で定義されるセンサー系・動力系パラメータの定義
  - 機体側インターフェースの仕様（電気的特性含む）
  - 自動運転などの構成を含む場合、自動運転のためのソフトウェア機能要求
  - カメラなどの構成を含む場合、カメラなどのデータの受け渡し、およびソフトウェア機能要求
  - 搭載されるセンサー、ハードウェアの値域の定義
  - 機体本体のシステムテストへの供用

  ■ セクション 110 からの出力

  - ソフトウェアで出力される物理コンポーネントへの制御パラメータの列挙
  - 単体テスト項目として、物理コンポーネントへの入力データに対する特性をグラフなどで解析した結果

- 運用計画 – セクション 120 緊急時の対応計画（以降、「セクション 120」と呼ぶ）、セクション

200 無人航空機飛行規程(以降、「セクション 200」と呼ぶ)、セクション 205 ICA(以降、「セクション 205」と呼ぶ)

■ セクション 110 への入力

- 他セクションから導出される運用計画、飛行規程に関してセクション 110 の要求仕様が合致しているかの確認
- 整備のために必要な制御パラメータ入出力確認のための要求事項の追加
- 不具合管理に紐づく整備時点の異常の詳細な記述方法(ログ)

■ セクション 110 からの出力

- ソフトウェアから出力される物理コンポーネントや論理コンポーネントへのデータ変動と、インターフェースの関係に基づく飛行方法、制限の詳細な記述
- ソフトウェア開発における不具合修正アップデートの方法の提示
- ソフトウェアの管理方法についての教示

■ セクション 110 とのギャップ

- 200 飛行規程はソフトウェアの「使い方」であるので、基本的にオペレータ側ユーザーインターフェース要求は 200 飛行規程に入力すること
- また、機体側などの項目が存在する場合、200 飛行規程で、関連システムとの連携で利用される API などインターフェースの項目・値域と、値域の変動幅の許容量に対しての項目が記述されること
- 機体内部の OS や内部ツールなどに関して、一般的なフライトに資するものではなく、パラメータ調整など整備に関わる内容は 205 ICA に移譲
- 運用全体に対して、ユーザーインターフェースと管理方法についてソフトウェア開発者が言及すること

● フライトテスト – セクション 300 耐久性及び信頼性(以降、「セクション 300」と呼ぶ)セクション 305 起こり得る故障(以降、「セクション 305」と呼ぶ)

■ セクション 110 への入力

- セクション 300 におけるテストで不具合が発生した場合に対して、ソフトウェアに起因する不具合である場合のフィードバックと追加要求
- システムアーキテクチャが更新されない場合(ソフトウェア入出力の値域などの変更にとどまる場合)はゼロリセットではなく、更新による修正で対応
- システムアーキテクチャが更新される必要がある場合(ソフトウェアのみならずハードウェアの追加などが必要になり、結果的にソフトウェアコンポーネントの追加、更新も必要となる場合)はゼロリセットとなる可能性
- ソフトウェアで制限「されていない」、セクション 200 で「指定されていない」値域や値域の変動幅の許容量についてのリストを提示し、セクション 305 で「起こり得る故障」としてテ

ストすべき値
- システムハングアップなど、データ流通がストップする場合においての制御フローの提示

■ セクション 110 からの出力

- ソフトウェア本体、セクション 200 などで扱われるユースケース
- セクション 110 で管理している関連システムソフトウェアなど

■ セクション 110 とのギャップ

- セクション 135、セクション 305 では機能に対する単一故障を見るが、ソフトウェア機能は場合によっては単一的に故障が発生したときに連鎖的に故障が発生するケースがみられる(メモリリークなど含む)。そのため、ソフトウェアシステム全体の連携の故障解析などは、あらかじめ結合テストなどで把握すること

4.3 セクション 110(a)に対する解説(ソフトウェアに対する試験による検証)

セクション 110(a)は、ソフトウェアに対するテストによる検証、しかもシステムレベルの要求ベーステストを求めている。各システムは連携して無人航空機システムを構築し、安全な運用のために相互作用するが、各システム(系統)は情報の流通(インターフェースを通したインプットとアウトプット)と内部の機能が機能要求や安全要求、セキュリティなどの他の要求によって定義されることから、これらの要求を満たす機能、入力に対する出力があることを、システム要求に応じてテストを行うことが規定される。この要求は網羅的に整理され、要求を満たす機能、出力が実現されていることを証明する必要がある。セクション 110 のタイトルは「ソフトウェア」であり、航空局ガイドラインでは、*「まずソフトウェアに対し試験(テスト)で要求が適切に実装されていることの確認を行います。」*と記述されているとおり、目的はソフトウェアに対する要求の実装確認である。この実装確認の手段として、ソフトウェア単体ではテストが難しいものも存在することから、「システムレベルの要求」に対して実施することでよいとしており、要求が適切にカバレッジされていれば、ソフトウェア単体でのテストに依らなくてもよいと解釈できる。

〔引用:航空局ガイドライン 110 ソフトウェア基準の概要〕

ソフトウェアに対する検証の目的は、故障の発生に繋がるエラーが残存するのを最小化し、無人航空機を安全に運用するためである。ソフトウェア検証は、システムレベルの要求に対し、検証(テスト、アナリシスおよびインスペクション)により、各要求が適切に実装されていることの確認を行う。航空局ガイドラインでは、「斜体部の引用:*ASTM F3153-15"Standard Specification for Verification of Avionics Systems"を参考にすることができる*」としている。これらの活動を通じて、システムレベル要求ベーステストの結果を「無人航空機ソフトウェア適合性証明完了報告書」から呼び出されるように引用し、形態管理すべき文書として、「システムレベルの要求一覧」にトレース可能な状態で保持する。航空局ガイドラインでは、*システムレベル要求の実装確認が対象*であるが、本解説書では、ソフトウェア要求のインプットとアウトプットを確認する方法についても取り扱っている。

〔引用:航空局ガイドライン 110 ソフトウェア適合証明方法(MoC)〕

(1) 「無人航空機の安全な運用に影響を与えるすべてのソフトウェア」の特定

セクション 110(a)の対象は、無人航空機の「安全な運用に影響を与えるソフトウェア」が対象である。無人航空機開発時には多数のソフトウェアが搭載されているため、それらを把握し、機能や関連性を理解することが非常に重要である。以下に手順を示す。

1) 「ソフトウェア一覧」の作成

無人航空機に使用される全ソフトウェアを次のインプットを参考に「ソフトウェア一覧」にリストアップする。ソフトウェアは、インストールされたソフトウェア(ソフトウェアオブジェクト)単位で記載する。

＜インプット＞

- 設計書、図面目録、三面図インターフェース管理図面
- 材料・部品・装備品等の部品表
- 関連システムの最低限の仕様

次の記入要領を参考に「ソフトウェア名称」および「外部のシステムやユーザーとの相互作用の詳細」を「ソフトウェア一覧」に記載する。

表 4.3-1 記入要領

記載項目	記載事項
P/N および ID	ソフトウェアのインストールされたソフトウェア(ソフトウェアオブジェクト)単位での Parts No.または ID を記載する
ソフトウェア名称	ソフトウェアの正確な名称や識別子を記載する
主な機能	ソフトウェアにより行われる主要な機能やタスクを記述する
搭載／非搭載	型式に搭載されるか、されないかを記述する
関連するハードウェア要素またはインターフェース	ソフトウェアがどのハードウェアコンポーネントやインターフェースと直接的に関連があるかを確認する
関連システムやユーザーとの相互作用	ソフトウェアが外部のシステムやオペレータとどのようにインタラクトするか特定する

2) ソフトウェアの影響評価

ソフトウェアのリストアップを通じて作成した「ソフトウェアの一覧」に対して、ソフトウェアが誤った挙動をした場合の無人航空機の運用の安全性に直接影響を及ぼす可能性のあるソフトウェアを抽出するための手順を以下に示す。

a. 安全性解析手法の選択

航空局ガイドラインでは、*セクション 135 を準用してソフトウェアが誤った挙動をした場合の影響度を評価する方法*、*FHA*、*SSA*、*および FMEA などの安全性解析手法*が紹介されており、これらの実

施が困難な場合は簡易版 FMEA の実施が紹介されている。また、安全性解析を使用するための規格としては、航空機開発で利用されている安全性評価プロセス(SAE ARP4761)、およびシステム開発保証プロセス(SAE ARP4754A)などが参照可能である。

〔引用:航空局ガイドライン 110 ソフトウェア適合証明方法(MoC)〕

b. ソフトウェアの評価と抽出

選定した安全性解析手法を用いて、「ソフトウェア一覧」に対して評価を行う。この評価の様式および記入例を表 4.3-2 に例示する。評価の結果、ソフトウェアの動作不良などにより「S/W エラーで無人航空機の安全な運用に影響を与えるか①」が「No」の場合は「①が No の理由」を記載し、「セクション 110/135(b)適用 S/W」は「No」となる。一方、「S/W エラーで無人航空機の安全な運用に影響を与えるか①」が「Yes」の場合はセクション 110/135 適用 S/W に「○」を記入し、評価対象として抽出する。

なお、ソフトウェアの動作不良が発生した場合に、安全な運用に影響を与えないよう別のソフトウェアによってカバーすることができるならば、どちらか一方のソフトウェアのみをセクション 110 の対象とすることができる。(カバーするソフトウェアとカバーされるソフトウェアの両方をセクション 110 の対象としても良い)

表 4.3-2 フライトエッセンシャル S/W 特定解析書の記入例

No.	P/N	Nomen	Ver.	S/W エラーで無人航空機の安全な運用に影響を与えるか①	①が No の理由	セクション 110/135(b)適用 S/W
1	XXXXX-XXXXX	フライトモード制御ソフトウェア	A	Yes	-	○
2	XXXXX-XXXXX	帰還モード制御ソフトウェア	B	Yes	-	○
3	XXXXX-XXXXX	安全機能制御ソフトウェア	A	Yes	-	○
4	XXXXX-XXXXX	操作ソフトウェア	B	Yes	-	○
5	XXXXX-XXXXX	機器制御ソフトウェア	A	No	別の S/W がカバー	No

（2） システムレベル要求ベーステスト

1） システムレベル要求の導出

航空機開発におけるソフトウェア開発の流れは、RTCA DO-178C および SAE ARP4754A がガイドラインとして規定され、V&V プロセスとして知られるプロセスが運用されている。

航空局ガイドラインでは、システムレベルのテストによりその動作を確認する必要がありそのため、「システムレベルの要求」を定義する必要がある。要求の定義は、JISX0020:1992 では、要求とは、「*システムが満たさなければならない必須条件*」とされている。なお、SAE ARP4754A では、要求の定義は「*検証可能で、実装を検証できる機能仕様の特定可能な要素*」とされている。具体的な機能要求の導出については、システムの構造や振舞いを記述する記法を定めた標準の 1 つである SysML(System Modeling Language)の要求図などを用いて網羅的に導出する必要があるが、その中で、ソフトウェアに関わる要求であるかを考察すること、さらに、非機能要求を含めてソフトウェアに対する要求を導出する必要がある。

〔引用:JISX00220:1992 20.01 一般概念 20.01.02 要件、要求、要求事項〕

〔引用:SAE ARP4754A〕

は、組織におけるシステムの位置づけと、それに基づく要求の種類である。実際のところ、CONOPS で記載される内容は組織全体の計画である。そのため、から紐解くと、4 つの要求が CONOPS から読み取れる。

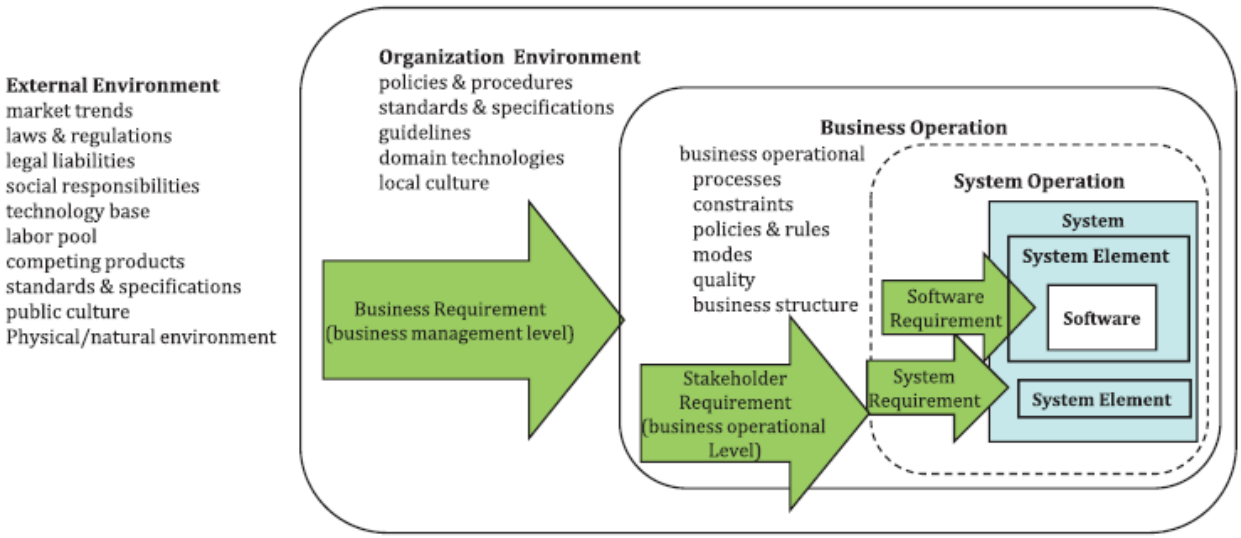


図 4.3-1 ビジネス・コンテキストにおける要求範囲の例

［引用:ISO/IEC/IEEE 29148:2018 Figure3-Example of requirements scope in a business context］

① ビジネス要求
② 利害関係者要求
③ システム要求
④ ソフトウェア要求

このうち、③のシステム要求がソフトウェアを含む箇所の要求である。

セクション 110 で管理する部分は基本的に④の部分である。一方で、システムレベルの要求ベーステストを鑑みると、テストの設計段階では、これらシステム全体の構成を含めて設計する必要がある。

第二種型式認証で考えると、これらの「システム」とは、ハードウェアに起因する 1 つの系統として見ることができる。無人航空機システムは、大きく分解すれば、機体システムと地上システムに分解できるが、より細かく分解すると、通信システム、補助システムなどに分解でき、さらに、各サブシステム内には、一連のインストーラブルなソフトウェアのまとまりとして、「地上局ソフトウェア」や「フライトコントローラソフトウェア」などとして解釈することができる。一方で、地上局ソフトウェアやフライトコントローラソフトウェアは、ソフトウェア単体で動くものではなく、実機を伴うものであるため、この部分の構成の管理を行う必要がある。また、狭義のシステムではコンピュータシステムの中のハードウェアとソフトウェアが主体となるが、ここでいうシステムでは PID 制御のパラメータ変更といった「手作業への要求」や機体の整備要求（ICA など）および地上局システムに対するアップデートなどの「設備への要求」も含まれたシステムが定義されている。

これらの抽出された要求は、システムレベルでまとめられ、システムレベルテストの計画で利用することとなる。このため、システムレベルの要求は網羅的に抽出される必要があり、可能であれば、要求管理から無人航空機システム全体に対する形態管理を行うべきであると考えられる。また、初期設計から実装、テストに至るまで、開発のあらゆる段階で潜在的なリスクを特定し、軽減するため、航空機システムの複雑さ、ソフトウェアとハードウェアの相互依存性を考慮すると、システム要求の全体的な視点が必要となる。したがって、要求を抽出する際は、要求抽出に関わる開発者やメンテナンス関係者などのステークホルダーを交えて検討することを推奨する。なお、ソフトウェアの航空機安全性に対する要求に関しては、SAE ARP4754A や RTCA DO-178C などのドキュメントを参照されたい。なお、DO-178C では、ソフトウェア要求フェーズにおいて、システムレベル要求をブレイクダウンして、HLR（ハイレベル要求）を作成するプロセスが定義されている。「航空局ガイドライン」では、ソフトウェアレベルの要求ベーステストについては規定されていないが、ソフトウェア単位でのテストが必要な場合は、システムレベル要求とソフトウェア要求の双方向のトレースを確保しておくことが重要である。

2） システムレベル要求ベーステストの計画

次に、「システムレベルの要求一覧」に記載された各要求に対してどのようにテストを行うかを検討する。「システムレベルの要求一覧」に定義した各システム要求を元にテストケースを作成する。テストケースは、各要求の確認を漏れなくダブりなく設計し、テストケースに落とし込む。表 4.3-3 では、GNSS システムおよびフライトコントローラに対する要求ベーステストを計画するために、システム要求から導出されたテストケースを例示したものである。テストケースは、同表のような項目を含めることが推奨される。

表 4.3-3 テストケースの記入例

テストケースID	目的	前提条件（実行前の状態）	テスト手順	発生させるイベント	期待する結果（実行後の状態）	合否判定基準
GNSS_TC_001	GNSSシステムが、衛星から受け取った情報を適切に解釈し、正確な位置情報を生成できること	GNSSシステムが起動し、衛星からのシグナルを受信可能な状態であること	GNSSテスト手順書	衛星からの信号を模擬した信号を送信	GNSSシステムが正確な位置情報を表示	GNSSシステムが正確な位置情報を5秒以内に表示すればテストは合格（PASS）、それ以外の場合は不合格（FAIL）
FC_TC_001	フライトコントローラがパイロットからの指示を適切に解釈し、これに基づいた無人航空機の運動を制御できること	無人航空機が離陸し、操縦可能な状態となっていること	フライトコントローラテスト手順書	パイロットが操縦桿を操作し、フライトコントローラに指示を送信	フライトコントローラが出力した指示に従って無人航空機が適切に飛行	無人航空機がフライトコントローラの指示に従い、予定した動きをすぐに実行すればテストは合格（PASS）、それ以外は不合格（FAIL）

3） テスト手順書の作成

実際にテストを実行するためには、詳細なテスト手順が必要となる。そのため、各テストケースに対して詳細なテスト手順を作成し、それに基づいてテストを行う。手順は、実際にテストを行う担当者が理解できるレベルで明確に記載し、テスト手順書の項目例、および記載内容は下表を参照のこと。

表 4.3-4 テスト手順書の項目例

項目	項目タイトル	記載要領
1	はじめに	
1.1	目的	テストの全体的な目的と方針を明確に記載。テストが必要とする主な理由とその背景を説明する
1.2	適用範囲	テスト手順が適用する具体的なシステムの部分や設備を定義。特定のモジュールや機能に限定される場合がある
1.3	参照文書	使用される資料、システム要求仕様、デザイン文書など、テストケースを理解するために必要な参考文献をリストアップする
1.4	用語と略語	手順書内で使用する重要な専門用語や略語の説明を記載する
2	テストの概要	
2.1	テストするシステム、機器、装置などの概要	テストの対象である装置やシステムの基本機能や特性を記載する
2.2	テストレベル	システムテスト、統合テスト、単体テストなど、実施するテストの種類を記載する
2.3	テスト目標	テストを通じて実現したい具体的な目標を記載する
3	テストの前提条件	
3.1	環境条件	テストを行うために必要な環境、設定、またはその他の前提条件を記載する
3.2	機器セットアップ	システムが適切にセットアップされ、テストの準備が整った状態であることの確認方法を記載
3.3	テストデータ	テストの際に必要なデータの準備とその内容を記載する
3.4	テストツールと設備	テスト運用に必要なハードウェア、ソフトウェア、ツールをリスト化し、そのセットアップガイドを記載する
4	テスト手順	
4.1	テスト項目	テストする機能の詳細なリストを記載。各項目は明確に説明し、理解しやすいように構造化する
4.2	操作手順および確認項目	各テスト項目について具体的なテスト手順を記載し、テスト実施者がそれに従ってテストを正確に実行できるようにする
4.3	発生させるイベント	どのようなイベントを発生させ、その反応をテストするか具体的に記載する
5	合否判定基準	
5.1	期待する結果	各テスト項目に対する期待される出力または結果を記載する
5.2	合否判定(PASS/FAIL)基準	テストパス(成功)の基準を明確に記述。これは、システムが期待通りに動作したことを確認するための基準である
6	附録	
6.1	テストに使用したツールのリスト	テストで使用したすべてのツール、およびバージョン情報をリストアップする
6.2	参考資料	仕様書、デザイン資料、既存のテストケースなど、参考になる可能性のある他の資料を記載する

4） ソフトウェアレベルの要求ベーステスト

これまで、「システムレベルの要求」に対するテスト方法を解説したが、航空局ガイドラインでは、ソフトウェアに対するテストを否定しているわけではない。航空局ガイドラインでは、*「ソフトウェアのテストには、ホワイトボックステスト、ブラックボックステスト、単体テスト、統合テスト、これ以外の分類も含め多種多様に存在しますが」*と記載されているため、代表的なソフトウェア要求ベーステストの分類について解説する。

前述のとおり、無人航空機のソフトウェア開発は V&V プロセスで実施され、テストは、いくつかの工程に分かれており、解説書により表現の違いはあるが、JSTQB Foundation Level シラバスでは、下表のように分類される。

表 4.3-5 テストレベルによる分類（JSTQB Foundation Level シラバス）

テストレベル	説明
単体テスト	個別にテスト可能なコンポーネントに焦点をあてるテスト （JSTQB Foundation Level シラバスでは「コンポーネントテスト」と定義されているが「航空局ガイドライン」の用語を優先し、セクション 110 では「単体テスト」として統一している）
統合テスト	コンポーネントまたはシステム間の相互処理に焦点をあてるテスト
システムテスト	システムが実行するエンドツーエンドのタスクと、タスクの実行時にシステムが示す非機能的振る舞いといったシステムやプロダクト全体の振る舞いや能力に焦点をあてるテスト
受け入れテスト	システム全体の振る舞いや能力に焦点をあてるテスト システムが、ユーザーのニーズ、要求、ビジネスプロセスを満足するかをチェックするための公式なテスト

〔引用：JSTQB Foundation Level シラバス 2.2 テストレベル・JSTQB・2024 年 2 月 13 日・https://jstqb.jp/dl/JSTQB-SyllabusFoundation_Version2018V31.J03.pdf〕

5） テストケース、テスト手順書およびテスト結果のトレーサビリティ

航空局ガイドラインでは、*要求からテストケース及び手順へはトレーサビリティを確保する必要があります。*と記述されており、システム要求に対して、抜け・漏れなく「システムレベルのテストケース」が設定され、それを実行するための「テスト手順書」が設定され、「システムレベルのテストケース」および「テスト手順書」通りに実施した結果が「テスト結果」として完了していることが確認できる仕組みが必要である。また、各テストが要求を網羅しているか確認することもあり、ガイドでは求められていないが、適合性証明計画などで計画することも有効であると考えられる。なお、ここでいうトレーサビリティは、要求-テストが必ずしも「1:1」にはならず、「1:N」または「N：M」（複数の要求：複数のテスト）関係も存在すること、ならびに階層間の関係、および各階層でトレースすべき項目を定義しておくことに留意する必要がある。

図 4.3-2 は、要求ベーステストにおいて、ソフトウェアを含むシステムのレベルでテストを実施したトレーサビリティの例である。また、図 4.3-3 は、ソフトウェアのレベルでテストを実施したトレーサビリティの例である。実際の開発においては、要求文書の作成方針や利用可能なテスト環境などを考慮しながら、どのようなトレーサビリティの方式を選択するのかについて検討する必要がある。

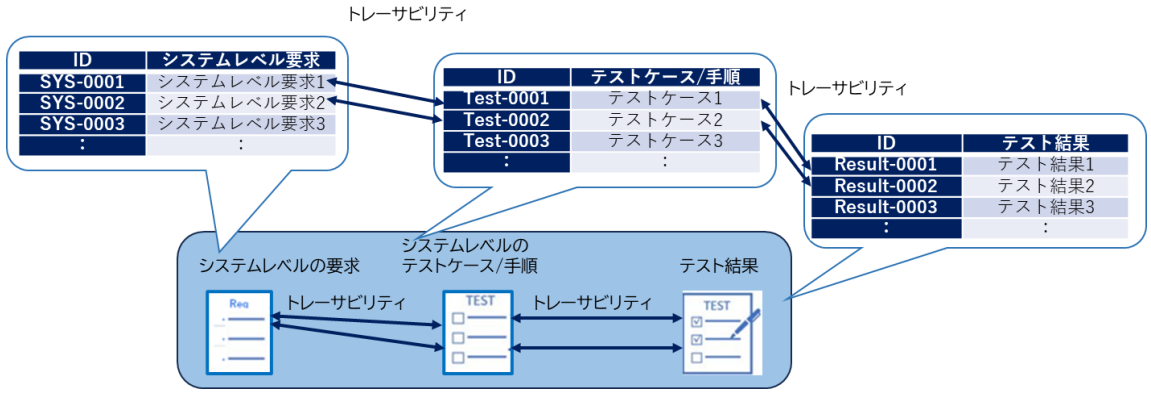


図 4.3-2 ソフトウェアを含むシステムのレベルでテストを実施したトレーサビリティの例

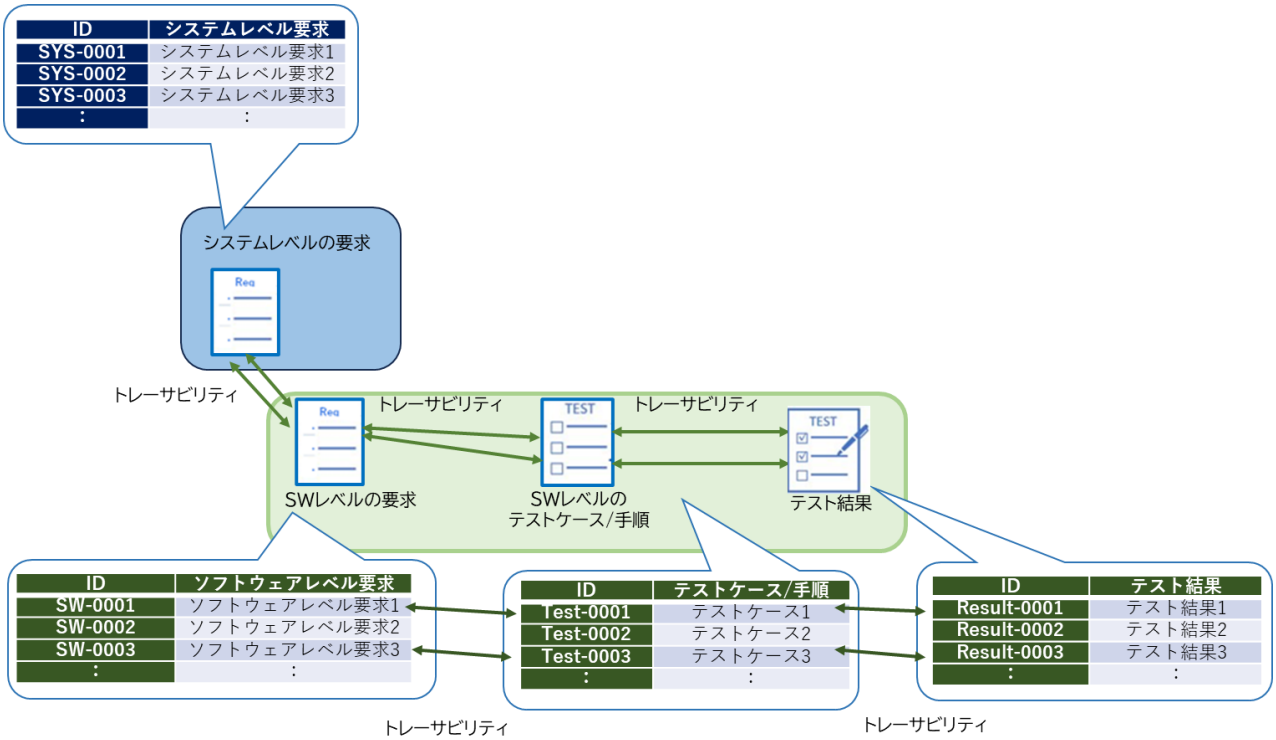


図 4.3-3 ソフトウェアのレベルでテストを実施したトレーサビリティの例

6） テストでの確認が困難な要求への対応方法

すべての要求は、基本的にはテストにより確認される必要がある一方、航空局ガイドラインでは「*テストでの確認が困難な要求に関しては解析(アナリシス)や検査(インスペクション)といった方法も許容されます。*」とあるため、システムレベルの要求は、テスト、解析またはアナリシスのどれを実施するかを決定し、計画する必要がある。テストでの確認が困難な要求に関してはこのステップでは、各要求に対しては確認方法がテストだけに限らない可能性があることに注意が必要である。例えば、非機能要求(許容できるメモリ量、CPU の負荷など)については、テストだけでなく、解析や検査なども確認方法として採用される。解析および検査について、ASTM F3153-22 の定義を引用し、事例を下表に例示した。

表 4.3-6 解析および検査の例示

要求	検査(インスペクション)	解析(アナリシス)
定義(ASTM F3153-22)	検査とは、1つ以上の感覚(例えば、視覚、聴覚、触覚など)を用いたシステムの非破壊検査である。単純な物理的操作や測定を含むが、これに限定されない	解析とは、モデル、計算、テスト装置、またはそれらの組み合わせを用いてシステムを検証することである。分析により、既知のデータおよび仮定の外挿に基づき、期待されるシステム性能について予測的な言明を行うことができる
パフォーマンス	リソースの消費速度、CPU 負荷、メモリ使用量、バッテリ消耗などのパフォーマンス関連のデータを収集し、分析する	実際にシステムを観察し、温度上昇、異音、異臭などがないか確認する
耐久性	使用回数・時間、テストパターン(気温、湿度、振動、衝撃)に基づいて設計寿命を拡大推定する	長期使用後、システムに異常摩耗、損傷、変形が起きていないか、外観および振動・音を確認する
ロード	同時に多くの指示を与えるシミュレーションを通じて、フライトコントローラの制約や上限性能を理解する	高負荷環境下で稼働した後のコントローラを確認し、異音、異臭などがないか確認する
可用性	実際の稼働時間と全体の稼働可能時間をもとに、システムの可用性を統計分析する	長期にわたり連続稼働させたハードウェアを視察し、熱問題、劣化などを確認する
セキュリティ	暗号化、認証などのセキュリティ機能が正しく実装されているかを確認するために、コードレビューやセキュリティ解析ツールを使用する	物理的なセキュリティ特性(例えば、システムに物理的アクセスを遮断するケースなど)を視察上、損傷や変形を確認する

4.4 セクション 110(b)に対する解説(ソフトウェアの形態管理)

セクション 110(b)は、ソフトウェアエラーを減少させるため、形態管理システムの使用を求めている。

無人航空機のソフトウェア開発には、多くのステークホルダーが関わり、それぞれが異なる役割および責任を持つ。また、日々の作業では、それらステークホルダーにより、ソフトウェアが頻繁に変更される。このような状況において形態管理システムを使用すると、以下の利点が得られる。

- あるソフトウェアが変更されたときに、その変更内容(例えば、誰が変更を行ったのか、いつ変更が行われたのか、なぜ変更が行われたのか、どのデータのどの箇所が変更されたのかなど)について追跡できるようになる。その結果、ソフトウェアの変更から生じる現場の混乱が抑えられ、エラーがソフトウェアに入り込むリスクを低減させることができる。
- 開発に関わる人が、最新のバージョンのソフトウェアを容易に特定できるようになる。その結果、誤って古いバージョンのソフトウェアを使用するリスクを低減することができる。
- ソフトウェアの不適切な変更により新たなエラーが発生した場合、エラーが発生していない過去のバージョンを特定し、素早く戻ることができる。その結果、エラーの影響を最小限に抑えることが可能となる。
- 形態管理システムにより管理されているソフトウェアへのアクセス制御を行うことができる。これにより、特定の開発者のみがデータを変更することが保証され、不適切な変更が抑制され、ソフトウェアの整合性を保つことが可能になる。

なお、形態管理の目的は、ソフトウェアエラーの減少だけでなく、多岐にわたる。例えば、各型式に搭載されているソフトウェアの追跡、適切なソフトウェアの管理および保存、ソフトウェア開発プロセスの効率化、セキュリティの向上、ソフトウェアの再利用性の向上、データ喪失からの復旧などである。

(1) 形態管理の対象

形態管理の対象は、無人航空機の安全な運用に影響を与えるソフトウェアである。航空局ガイドラインには、以下のデータが形態管理の対象として明示されている。

> *i. 要求(Requirements)*
> *ii. システム及びソフトウェアテスト環境の説明*
> *(System and software test environment descriptions)*
> *iii. 要求からテストケース及び手順へのトレーサビリティを含むテスト手順及び結果*
> *(Test procedures, and results with requirements traceability to test cases and procedures)*
> *iv. ソースコード及び開発環境/ツール*
> *(Source code and development environment/tools)*
> *v. 実行オブジェクトコードの複製のためのビルド/ロード手順*
> *(Build and load procedures for replication of the executable object code)*
>
> *[引用:航空局ガイドライン]*

航空局ガイドラインは、型式認証が適切かつ円滑に行われるよう、安全基準に対する適合性証明方法をとりまとめたものであり、基本的には航空局ガイドラインに明示されているデータについての形態管理が必要となる。表 4.4-1 に、形態管理が必要となる各データの概要を示す。

表 4.4-1 形態管理が必要となる各データの概要

データ	概要
要求	開発対象(例えば、機体、システム、ソフトウェアなど)の期待される振る舞いを定義するデータ。機能(何をすべきか)を定義する機能要求や性能(どの程度の性能で実行すべきか)を定義する性能要求などがある。航空局ガイドラインでは、システムレベルの要求の定義が求められている。
ソースコード	要求を具体化するための指示となるコード。アセンブリ言語や高水準言語(例えば、C や Python など)で記述される。
ビルド手順	ソースコードから実行オブジェクトコードを生成するためのビルド手順。実行オブジェクトコードとは、ターゲットコンピュータで実行できるコードの形式である。

データ	概要
ロード手順	実行可能オブジェクトコードをターゲットコンピュータにロードするための手順である。
開発環境／ツール	開発(すなわち、要求の作成からロードまで)で使用する環境やツール。例えば、ソースコードの統合開発環境、ビルドツール、ロードツールなどが含まれる。
テスト手順	要求が実装されていることを確認するための手順。要求とテスト手順の間のトレーサビリティを含む必要がある。
テスト結果	テスト手順の実行結果。
テスト環境	テストを実行する環境やツール。例えば、テストツールなど。

しかし、明示されているデータの形態管理が難しい場合、その代替として他のデータの形態管理を行うことが適切な場合がある。例えば、

- COTS ソフトウェアを使用する場合、ソースコードは提供されず、オブジェクトコードまたは実行可能オブジェクトコードのみが提供されることがある。そのような場合、オブジェクトコードまたは実行可能オブジェクトコードを形態管理することが適切な場合がある。詳細については、本解説書の 4.7(1)を参照のこと。
- 特定のテストが実行できない場合、代わりにインスペクションまたはアナリシスを行うことがある。そのような場合、テストケース、テスト手順およびテスト結果の代わりにそれらのデータを形態管理することが適切である。
- ソフトウェアのテストを別のテスト(例えば、セクション 300 のテスト)と合わせて行うことがある。そのような場合、別のテストのデータも形態管理することが適切である。

また、パラメータデータアイテムを作成する場合、その形態管理を行うことが適切である。パラメータデータアイテムについては、本解説書の 4.7(2)を参照のこと。

また、航空局ガイドラインに明示されているデータに加え、他のデータについても形態管理を行うことが適切な場合がある。例えば、以下に示すデータは、形態管理の対象とすることが適切な場合がある。

- ソフトウェア適合性証明計画書
- ソフトウェア設計
- 形態管理の記録(例えば、ベースラインの記録など)
- 不具合管理の記録(例えば、不具合の記録、不具合の分析結果など)
- ソフトウェア品質保証結果(注:航空局ガイドラインでは求められていないが、品質保証活動を実施する場合)
- ソフトウェア適合性証明完了報告書
- 宣言書
- 他のセクションの活動で生成されたデータ
- その他、社内資料やプロジェクト固有のニーズにより作成したデータ

（2） 形態管理の活動

航空局で求められている形態管理の活動を実施する必要がある。ただし、組織の規模／カルチャー／経験、ソフトウェアの規模／複雑性／開発プロセス、使用する形態管理ツール、その他のツールとの連携などを考慮して、それらの条件下で最適な形態管理の活動を行う必要がある。

参考として、表 4.4-2 に RTCA DO-178C のソフトウェア形態管理プロセス（Software Control Management Process）に要求されている活動の概要、および、それら活動が航空局ガイドラインで求められているか否かを示す。ただし、活動および用語の詳細は DO-178C を参照のこと。

表 4.4-2 DO-178C におけるソフトウェア形態管理プロセスの活動

活動	概要	DO-178C の参照	航空局ガイドラインとの対応
形態識別（Configuration Identification）	形態アイテムを選定し、それぞれを一意に識別するための方法を確立する活動である	Section 7.2.1	航空局ガイドラインで明確に求められている活動ではないが、形態管理を行う上で必須の活動であり、実施する必要がある
ベースライン設定（Baseline）	成熟した形態アイテムをベースラインとしてソフトウェアライブラリ（注：リポジトリ）に保存する活動である。ベースラインとして設定した形態アイテムを変更する場合、変更の手続きが必要となる	Section 7.2.2	航空局ガイドラインで求められている活動である
不具合管理（Problem Reporting）	不具合を、識別し、分析し、解決するまで管理する活動である	Section 7.2.3	航空局ガイドラインでは、形態管理とは異なる活動として求められている。不具合管理については、本解説書の 4.5 を参照のこと
変更管理（Change Control）	ソフトウェア開発の過程で発生する変更を管理する活動である。変更は、不具合の解決、新機能の追加、性能の改善などのために行われる	Section 7.2.4	航空局ガイドラインで明確に求められている活動ではないが、不具合の解決などのために実施する必要がある
変更レビュー（Change review）	変更が適切に取り扱われていることを確認する活動である	Section 7.2.5	航空局ガイドラインで明確に求められている活動ではない。必要に応じて実施することが推奨される
形態状況報告（Configuration status accounting）	形態管理についての状況（例えば、形態アイテムの状況、ベースラインの状況、不具合管理の状況、変更履歴、リリースの状況など）について記録し、報告する活動である	Section 7.2.6	航空局ガイドラインで明確に求められている活動ではない。必要に応じて実施することが推奨される
保存、復元、リリース（Archive, Retrieval, and Release）	ソフトウェア製品に関連するデータ（形態アイテムを含む）を保存し、必要に応じて復元する活動である。「保存」はデータを長期間保存し、保護することを意味する。「復元」は保存したデータを取り出すことを意味する。「リリース」は形態アイテムを正式に利用可能にすることを意味し、認証のために利用する前や顧客または他のプロジェクトに提供する前に行われる	Section 7.2.7	航空局ガイドラインで明確に求められている活動ではない。必要に応じて実施することが推奨される
ソフトウェアロード管理（Software Load Control）	ソフトウェアをターゲットコンピュータにロードするためのロード手順を作成する活動である	Section 7.4	航空局ガイドラインで明確に求められている活動である
ソフトウェア環境の管理（Software Life Cycle Environment Control）	ソフトウェア環境（例えば、ツール、ツールを使用するコンピュータなど）を管理する活動である	Section 7.5	航空局ガイドラインで明確に求められている活動である

なお、形態管理の活動は、セクション 110(b)に記載されている通り、全ライフサイクルを通して実施される。全ライフサイクルとは、一般的には、プロジェクトなどの立ち上げフェーズから始まり、開発フェーズ、テストフェーズ、運用および保守フェーズを経て、製品リタイアフェーズを指す。立ち上げフェーズでは、チームが発足され、形態管理システムの計画とセットアップが行われる。これには、形態管理ツールの選定、設定、トレーニングおよび適用が含まれる。また、製品リタイアフェーズにおいて、形態管理システムが終了し、形態管理の対象が破棄または保存される。

注:形態管理の活動のうち、ベースラインと不具合管理は、プロジェクトなどの立ち上げフェーズから実施する必要はない。ベースラインを開始するタイミングについては、本解説書の 4.4(3)を参照。不具合管理については、本解説書の 4.5 を参照。

(3) ベースラインを開始するタイミング

ベースラインとは、形態管理の活動の 1 つであり、特定の時点で形態管理対象の形態(例えば、文書番号とバージョンなど)を記録し、固定する活動である。これにより、その時点から先の活動における形態管理対象の変更および不具合の管理が明確になる。

航空局ガイドラインでは、ベースラインについて、「遅くともセクション 110(a)のテストの開始前にスタートする」ことを求めている。一般的には、プロジェクトなどの立ち上げフェーズにおいて、形態管理の責任者がベースラインを開始するタイミングを決定する。

(4) 形態管理記録

形態管理が行われた結果として、以下の記録などを保持する。

- 形態管理対象のリスト
- ベースラインのリスト
- 文書の改訂履歴
- バージョン管理ツールに自動的に記録された変更履歴

4.5 セクション 110(c)に対する解説(PR(Problem Report)システムの導入および活用)

サーキュラーNo.8-001 には、"*PR(Problem Report)システムの導入と活用*"とあるが、PR システムを利用する目的は不具合管理であるため、不具合管理とそれを実現するための PR システムについての解説としている。(PR システムは、不具合管理システムやバグトラッキングシステムなどと呼ばれていることもある)

不具合管理の目的は、不具合の対応漏れを防ぎ、ソフトウェアエラーを最小化することである。そのために、発見された不具合を記録し、分析し、解決するまで確実に管理する必要がある。PR システムを導入、活用し、確実に不具合管理できるようにすることや、開発中においては開発者間、運用中においては開発者と運用者間のコミュニケーションに役立てることも重要である。

(1)　不具合管理の対象

セクション 110(a)の対象と同じく安全な運用に影響を与えるソフトウェアが対象である。

テストのみでなく、解析(アナリシス)、検査(インスペクション)で発見された不具合も対象となり、ソースコードの不具合のみでなく、システムレベルの要求、テストケースおよび手順、テスト環境の不具合も含まれる。

プロジェクト管理に関する問題や品質保証活動によるプロセスの問題はここでは扱わない。

(2)　不具合管理の対象テスト、フェーズ

以下のテスト、フェーズで発見されたソフトウェアに関する不具合が対象となる。ベースラインを設定し、テスト、実運用を開始する必要がある。(以下のテスト、フェーズよりも前に、ベースラインを設定し、ソフトウェアの不具合管理を始めても構わない)

- セクション 110(a)のテストで発見されたソフトウェアに関する不具合
- 各種 D&R のテストで発見されたソフトウェアに関する不具合(例、安全基準を満たすために実施する地上テスト、飛行テストなど)
- 型式認証後の実運用時に発見されたソフトウェアに関する不具合

発見される不具合には、ソフトウェアに関する不具合だけでなく、ハードウェアなど他の不具合の可能性もある。それらを同じ管理方法で管理しても構わない。セクション 110(c)での対象は、ソフトウェアに関する不具合のみとなる。

(3)　不具合管理活動

以下が、不具合発見後の不具合管理活動になる。(4)の記録する不具合情報も参照。

1)　不具合の記録

テスト、実運用時に、ソフトウェアの不具合が発見された場合、不具合が一意に識別できるよう ID などを付与し、不具合内容を記録する。

不具合の現象だけでなく、不具合を再現させるための情報も記録する。テスト時であれば、不具合が発生したテストケース、ベースラインなどの情報、実運用時であれば、運用者にヒアリングし収集した情報などの記録が考えられる。

2)　不具合の分析

不具合の原因、影響範囲、重大度などを分析し、修正対象と修正内容を決定する。分析には、テストケースとシステムレベルの要求間のトレース情報などが参考となる。

不具合から修正対象、修正内容までトレースを取り、不具合の修正を確実にする。

3） 不具合の修正

不具合を修正し、修正内容をテスト、レビューなどで確認する。
既存の機能が正常に動作するかを確認するための回帰テスト（リグレッションテスト）を実施する。

4） 再テスト

再設定されたベースラインを用いて、再度テストを実施する。

5） 不具合の解決の追跡

各不具合にステータス（オープン、クローズなど）を付与してステータスを管理し、解決まで追跡する。再テストに合格することで不具合のクローズとなる。

6） 未解決の不具合の正当化

今回の型式認証では解決しない“未解決の不具合”がある場合、その不具合を残存させても、“無人航空機の安全な運用”への影響が許容範囲であることを評価し、未解決でも良い理由を記録する。すべての“未解決の不具合”とその正当化の情報を、ソフトウェア適合性証明完了報告書に記述する。

例えば、サプライヤが開発したソフトウェア、COTS ソフトウェア、オープンソースソフトウェア、関連システムの不具合が発見された場合、サプライヤ、COTS ソフトウェアの提供元、関連システムの開発元と調整し修正することや、オープンソースソフトウェアを修正することが望ましいが、修正できない場合があるかもしれない。

（4） 記録する不具合情報

不具合を一意に識別し、再現できるようにするために、以下の不具合情報などを記録する。

- 不具合 ID
- 不具合発見ベースライン
- 不具合発見テスト／フェーズ
- 不具合を発見したテストケース、手順
- 不具合内容
- ステータス（オープン、クローズなどの不具合のステータス）
- 重大度（安全な運用に影響を与えるかなど）
- 修正対象と修正内容
- 再テストケース、手順
- 再テスト結果
- 未解決の不具合の場合、正当化の情報など
  （他にも日付、発見者、不具合タイトル、優先順位なども記録することがある）

（5） PR システム

上記の不具合管理を実施可能な PR システムを利用して、不具合管理を確実にする。不正なアクセスや不正な変更などのセキュリティの観点からも、アクセス権限の管理、変更ログが残る PR システムが良い。PR システムのデータの保護のため、バックアップも考慮する。

利用者が PR システムを正しく使用することができるよう、使用手順を定める。

4.6 ソフトウェア適合性証明計画、完了報告書、宣言書に対する解説

（1） ソフトウェア適合性証明計画と完了報告、自己宣言

申請者が、開発されるソフトウェアに要求される厳密さに見合ったソフトウェアライフサイクルを提案しているかについて、検査者へ示すための計画をまとめたものをソフトウェア適合性証明計画と呼ぶ。第二種機体の型式認証においては、申請者はここで自ら定めた計画に沿って証明活動を実施し、結果をまとめた完了報告書を作成、検査者へ提出し確認を仰ぐ。了承を得たところで自己宣言書を提出し証明活動は完了となる。

（2） ソフトウェア適合性証明計画書

ソフトウェア適合性証明計画書とは、(1)の証明活動計画をまとめた書類一式である。

セクション 110 ではセクション 110(a)で「安全な運用に影響を与えるすべてのソフトウェア」の抽出とテストによる安全性への影響評価について、セクション 110(b)で形態管理の方策について、セクション 110(c)で PR システムの導入と活用について提示している。ソフトウェア適合性証明計画書は、申請者がセクション 110(a)にて抽出した「安全な運用に影響を与えるソフトウェア」についてセクション 110(a)～(c)それぞれの項目に対してどのように適合性証明を実施するかなどの計画を示した文書となる。以下に検証項目の例を示す。また、Appendix1～3 に記載例を示す。

表 4.6-1 セクション 110 各項におけるソフトウェア適合性証明計画における検証項目の例

項	活動	記載項目
(a)	要求ベーステスト	● 無人航空機のシステム概要 ● ソフトウェア一覧(搭載、非搭載別) ● 「安全な運用に影響を与えるソフトウェア」をどのように抽出するかの説明およびその結果 ● 「システムレベルの要求」定義(要求一覧) ● システムレベル要求ベーステストをどのように実施するかの説明(計画) ● システムレベル要求ベーステスト関連書類要求×テストケース×手順マトリクス)テストツール、テスト環境 ● 各要求に対するテスト方法概要 ● 解析(アナリシス) ● 検査(インスペクション)
(b)	形態管理	● 形態管理対象、それらの識別方法 ● ベースラインを開始するタイミング、ベースラインの対象およびそれらの識別方法 ● 変更管理の方法 ● ビルド手順およびロード手順の作成方法 ● ソフトウェア環境(例えば、使用するツール、ツールを使用するコンピュータなど)の管理方法
(c)	不具合管理	● 不具合管理の対象 ● 不具合管理の対象テスト、フェーズ ● 不具合管理の活動内容(不具合の記録、分析、修正、再テスト、不具合解決の追跡、未解決不具合の扱いなど) ● 使用する PR システム

1) 申請の流れ

申請者はソフトウェア適合性証明計画(案)および必要書類を作成し、検査者から合意を得た後に、証明活動を開始する。ソフトウェア適合性証明計画は申請者の計画が示された文書となるので、以降の過程で計画の見直しが発生した場合、申請者は当該計画を変更し、再度検査者より合意を得る必要がある。

また、110 ソフトウェアの証明活動は、その機体区分に応じて 2 つのパターンに大別される。
以下にサーキュラーNo.8-001 第 1 章より、「区分に応じて適用される規定」を引用する。

区分	第二種				第一種
	機体認証を受けようとする無人航空機／型式認証を受けようとする型式の無人航空機				機体認証を受けようとする無人航空機／型式認証を受けようとする型式の無人航空機
			最大離陸重量25kg以上のもの		人口密度の低い地域その他の無人航空機の飛行により航空機の航行の安全並びに地上及び水上の人及び物件の安全を損なうおそれが少ないと認められる地域を飛行するもの
	最大離陸重量4kg未満のもの	最大離陸重量4kg以上25kg未満のもの	特定飛行を行うもののうち、無人航空機の飛行により航空機の航行の安全並びに地上及び水上の人及び物件の安全を損なうおそれが少ないと認められるもの※2	その他のもの※3	
001 設計概念書(CONOPS)	✓	✓	✓	✓	✓
005 定義	✓	✓	✓	✓	✓
100 無人航空機に係る信号の監視と送信	✓※4	✓	✓	✓	✓
105 無人航空機の安全な運用に必要な関連システム	✓	✓	✓	✓	✓
110 ソフトウェア	✓※5	✓※5	✓※5	✓	✓
115 サイバーセキュリティ	✓	✓	✓	✓	✓
120 緊急時の対応計画	✓✓※6	✓	✓	✓	✓
125 雷	✓	✓	✓	✓	✓
130 悪天候	✓	✓	✓	✓	✓
135 重要な部品（フライトエッセンシャルパーツ）	✓	✓	✓	✓	✓
140 その他必要となる設計及び構成	✓※7	✓※7	✓※7	✓※7	✓※7
200 無人航空機飛行規程	✓	✓	✓	✓	✓
205 ICA	✓	✓	✓	✓	✓
300 耐久性及び信頼性	✓	✓	✓	✓	✓
305 起こり得る故障	✓✓※8	✓✓※8	✓	✓	✓
310 能力及び機能	✓※9	✓	✓	✓	✓
315 疲労試験	N/A	N/A	✓	✓	✓
320 制限の検証	N/A	N/A	✓	✓	✓

図 4.6-1 区分に応じて適用される規定

［引用：サーキュラーNo.8-001］

また併せて、型式認証取得までの全体フローを航空局ガイドラインより引用する。

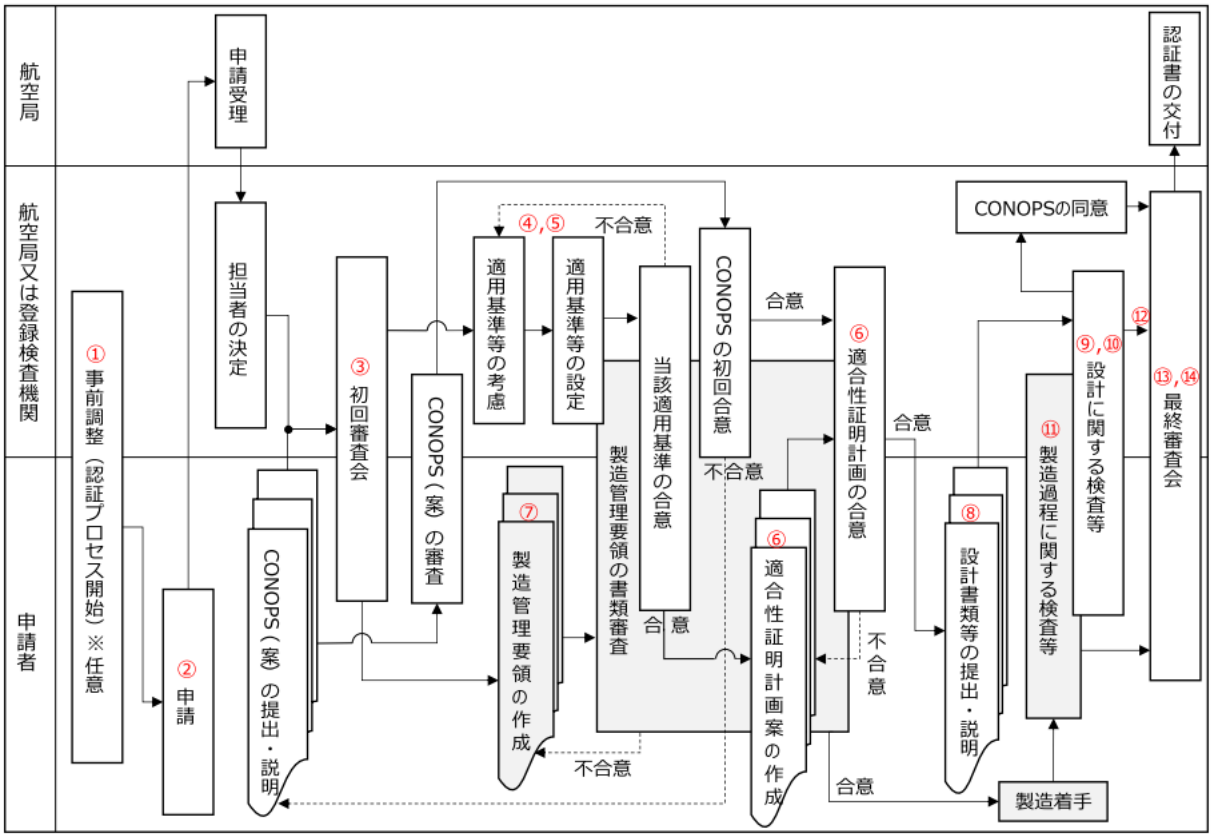


図 4.6-2 型式認証取得までの全体フロー

［引用：航空局ガイドライン］

第二種機体のうち図 4.6-1 で示される、最大離陸重量 25kg 未満のものおよび最大離陸重量 25kg 以上かつ「特定飛行を行うもののうち、無人航空機の飛行により航空機の航行の安全並びに地上及び水上の人及び物件の安全を損なう恐れが少ないと認められるもの」については、検査者の立会検査を必要としない。図 4.6-2 中の「*⑥適合性証明計画の合意*」以降に申請者が開発行為の一環として証明計画を立案、実行し、「*⑨⑩設計に関する検査等*」までに適合性証明結果および自己宣言書を検査者へ提出、受理されることで活動完了となる。

以下に上記のソフトウェア適合性証明計画の完了までの流れの一例を示す。実際の設計／証明活動は並行で進む場合や手戻りも発生するため一概に当てはまるものではない。

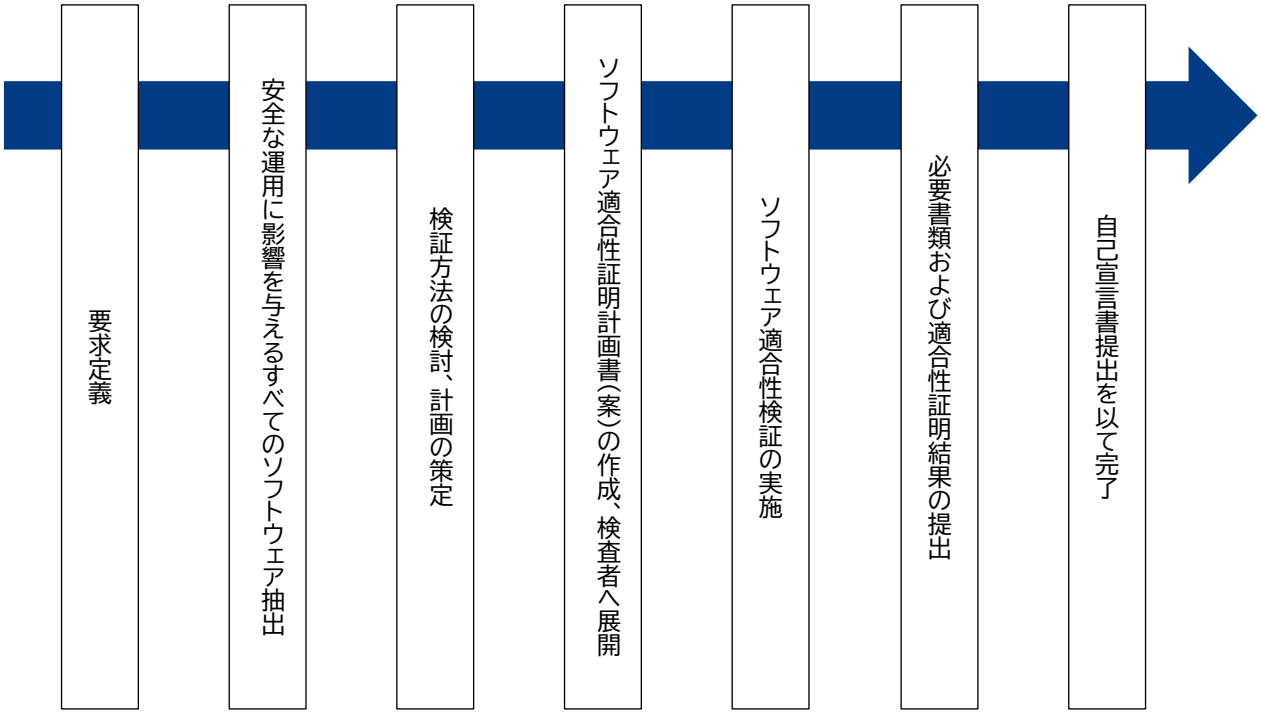


図 4.6-3 ソフトウェア開発に伴う適合性証明実施の一例

また、第二種機体の中でも図 4.6-1 における「最大離陸重量 25kg 以上のもの」のうち「その他のもの」および第一種機体については、その他セクションと同様に通常の CP に則った証明活動が必要となる。図 4.6-2 中の「*⑥適合性証明計画の合意*」までに開発者がソフトウェア適合性証明計画を立案、検査者と合意し、「*⑨⑩設計に関する検査等*」までに検査者立ち合い検査を含めた証明活動を実施する。適合性証明結果および必要書類を提出し、検査者の承認を以て完了となる。

2） 必要書類

下記 a～e をまとめた書類一式を適合性証明計画書とする。

a. システム概要

ソフトウェアに限らず、必要となる場合はハードウェア含めた機体仕様などシステムの概要を把握するための書類を作成する。

b. 安全に影響を与えるソフトウェア一覧

「安全に影響を与えるソフトウェア」の抽出基準およびその一覧を用意する。

c. 日程概略

申請者が希望する申請書の提出から証明活動完了までの日程の概略に係る情報を記載する。日程は四半期単位など大まかなものでも構わない。なお、ソフトウェア適合性証明計画の改訂の頻度などを勘案し、別途スケジュールの管理が可能な資料が存在する場合は当該資料を呼び出すことも可能と

する。

d. ソフトウェア適合性証明計画表

Appendix 4 に記載例を示す。

ア） 適用基準などに関する事項

「安全な運用に影響を与えるソフトウェア」ついて、セクション 110(a)～(c)の適用基準、ソフトウェア適合性証明の必要性の有無、解析または実証の選択を含む適合を示す方法、実施時期などを記載する。

イ） 証明活動などに関連する人員に関する事項

本証明活動に関連する人員について、それぞれの責任および権限を明確にする。

e. その他必要となりうる書類

申請者が従来の無人航空機の設計にはないと考える新設計、新技術を採用する場合は、その概要ならびに検討事項およびその解決方法をまとめる。また、共同開発者がある場合は、共同開発者の名称、分担に関する情報を記載し、設計開発行為の一部を外部に委託する場合は、当該委託先（外注先）の名称、委託内容に関する情報を記載する。

3） 議事録

ソフトウェア適合性証明計画の説明および調整を行った際は、その説明内容、指摘およびその改善事項、調査事項、問題点などを明確にし、認識を共有する目的から、申請者において議事録を作成し、双方で記載の内容を確認する。

議事録には特に定まった様式はないが、「サーキュラーNo.8-002”無人航空機の型式認証等の手続き”（以降、「サーキュラーNo.8-002」と呼ぶ）」の別添 3（JCAB FORM 8-002-3）の様式を使用することも可能である。

（3） 完了報告書

ソフトウェア適合性の証明活動における完了報告書は、申請者が適合性証明計画に沿って実施した結果を明記し、完了したことを証明するものである。完了を承認する検査者からのソフトウェア適合性証明計画書、完了報告書および自己宣言書への署名を以て該当する項目の証明活動完了となる。以下に記載項目の例を示す。また、Appendix5 に記載例を示す。

表 4.6-2 セクション 110 各項におけるソフトウェア適合性証明完了報告書の記載内容

項	活動	記載項目
(a)	要求ベーステスト	● システムレベル要求ベーステストの結果概要 ● システムレベル要求ベーステスト結果 ● 各要求に対するテスト方法結果概要 ● ソフトウェアテスト結果 ● ソフトウェア解析(アナリシス)結果 ● ソフトウェア検査(インスペクション)結果
(b)	形態管理	● 形態管理した対象、それらを識別した方法 ● ベースラインを開始したタイミング、ベースラインの対象としたデータおよびそれらを識別した方法 ● 変更管理した方法 ● ビルド手順およびロード手順を作成した方法 ● ソフトウェア環境(例えば、使用するツール、ツールを使用するコンピュータ)を管理した方法 ● ソフトウェア適合性証明計画書からの相違点
(c)	不具合管理	● 不具合管理の対象 ● 不具合管理の対象テスト、フェーズ ● 不具合管理の活動内容 ● 使用した PR システム ● ソフトウェア適合性証明計画書からの相違点 ● 未解決の不具合のリストとその正当化の情報

(4) 自己宣言書について

サーキュラーNo.8-001 にて記載される通り、機体認証および型式認証を受けようとする無人航空機のうち一部を除く第二種に分類される機体は、宣言書を以て検査の代替とする。

機体認証または型式認証申請者は、自身が当該要件の適合性を確認した上でその旨を自己宣言書に記載し検査者へ提出する。Appendix5 にその書き方の例を示す。

4.7 その他参考となる情報

(1) ソースコードを入手できない場合の扱い

航空局ガイドラインのセクション 110(b)においては、ソースコードの形態管理が求められている。一方で、申請者はフリーウェアや COTS ソフトウェアなどを入手することが可能である。一般的にこれらのソフトウェアについて、申請者がソースコードを入手することは難しいと考えられる。そこでこのような場合においても、申請者が型式認証を正しいプロセスで取得できるように解説を行う。はじめに申請者が入手可能なソフトウェアを一覧化し、ソースコードの入手可否を分類した上で、対応内容を記載する。

1) 申請者が入手可能なソフトウェア一覧

a. オープンソースソフトウェア(OSS)

特定のライセンスにしたがって、ソースコードを使用、調査、再利用、修正、拡張、再配布が可能なソフトウェアの総称である。脆弱性調査を行った結果、問題がある場合はセクション 115 に則り対策を行

う。

b. 自由ソフトウェア

ユーザーが自由に共有、研究、変更できるソフトウェアである。

c. 自社開発ソフトウェア

ア） 申請者がソフトウェアの情報を把握・入手できる場合

イ） 申請者がソフトウェアの情報を把握・入手できない場合（例：現在、組織に存在しない人物が過去に作成した場合など）

この場合はテストを実施した結果、問題を抽出してもソースコードが無く、対応ができないため使用することは望ましくない。

d. COTS

ア） （COTS）システム

ハードウェア、ソフトウェアを組み合わせたシステムとして、販売されている場合を指す。

イ） （COTS）ソフトウェア

COTS ソフトウェアとして、ソフトウェア単品で、販売されている場合を指す。

e. フリーウェア

無料で利用することができるソフトウェアである。

f. 他社開発

申請者が所属する組織以外で開発されたソフトウェアを指す。ここでは特に、申請者が所属する組織が、ソフトウェア仕様書を作成して他社に開発を依頼する場合を指す。

2） ソースコードの入手可否

図 4.7-1 は、ソースコードの入手可否について一般的な考えに基づき、分類した図である。以降は図 4.7-1 で分類した例を基に解説を行う。

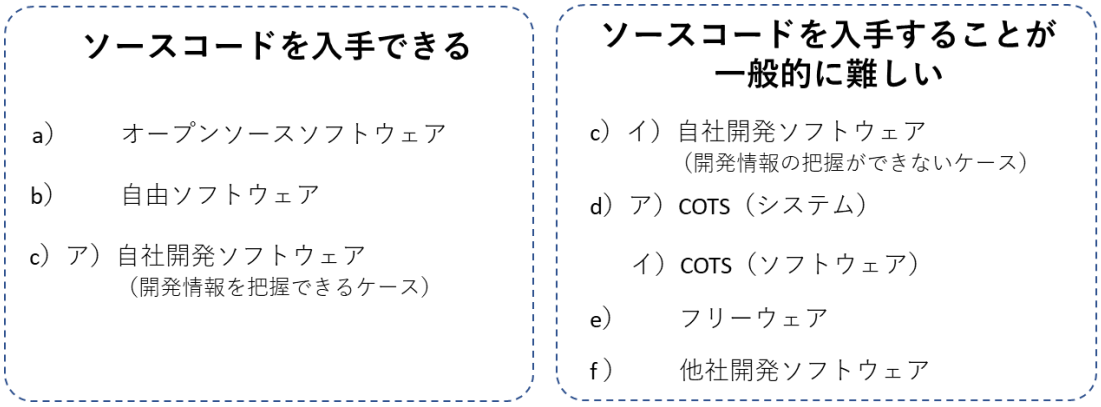


図 4.7-1 ソースコードの入手可否を一般的に整理したソフトウェアの分類

3） セクション 110(a)に対する解説(ソフトウェアに対する試験による検証)

a. ソースコードを入手できるソフトウェア

ソースコードの入手が可能なためセクション 110(a)に則り、テストを実施する。

b. ソースコードを入手することが一般的に難しいソフトウェア

ソースコードが入手できない場合、セクション 110(a)に則り、テストを実施する。ソースコードレベルのテストが必要な場合は、ソースコードを開発する組織に対応を依頼する。

4） セクション 110(b)に対する解説(ソフトウェアの形態管理)

a. ソースコードを入手できるソフトウェア

ソースコードの入手が可能なため、ソースコードレベルでセクション 110(b)に則った形態管理を実施する。

b. ソースコードを入手することが一般的に難しいソフトウェア

一般的にソースコードの入手が困難なため、オブジェクトコードレベルでセクション 110(b)に則った形態管理を行う。オブジェクトコードを入手できない場合は、実行可能オブジェクトコード(EOC)の形態管理を実施する。EOC は不具合対応で更新される可能性があるため、最新の EOC のバージョン情報を常に把握しておくことが必要である。さらに COTS においては形態管理と同時に、製品の販売とサポートが終了するタイミングを把握しておくことが重要である。サポート終了後のソフトウェアを使用して問題が発生した場合、対応ができないためである。

またソースコードの形態管理についてはサプライヤ管理の一環で、申請者はサプライヤにソースコードの形態管理を行うことを求めることとする。

5） セクション 110(c)に対する解説(PR システムの導入および活用)

a. ソースコードを入手できるソフトウェア

ソースコードの入手が可能なため、セクション 110(c)に則った不具合管理を実施する。

b. ソースコードを入手することが一般的に難しいソフトウェア

サプライヤがソフトウェアを開発している場合、サプライヤが実施している不具合管理がセクション 110 ソフトウェアのセクション 110(c)を満たすことを示すまでは不要と考える。申請者側でセクション 110 ソフトウェアのセクション 110(a)で求められているシステムレベルのテストを実施し、不具合が発見された場合は、サプライヤによる不具合修正の後、申請者側で再テストによる確認を行うことで不具合管理は可能である。申請者側の不具合管理活動を示すことで十分と考える。

（2） PDI File の扱い

PDI File とはパラメータを変更することによりターゲットコンピュータが直接利用できるデータであり、実行可能オブジェクトコード(EOC)の振る舞いを変えることができるファイルである。また PDI File は、EOC から独立したファイルであり、EOC とは別に形態管理するデータである。

1） セクション 110(a)に対する仮説(ソフトウェアに対する試験による検証)

PDI File を使用する場合、システムレベルの要求において、データの値、範囲、フォーマットなどを定義する。その上で、EOC と合わせてセクション 110(a)に則りテストを実施する。

2） セクション 110(b)に対する解説(ソフトウェアの形態管理)

PDI File に変更があった場合は、セクション 110(b)に則り形態管理を実施する。

3） セクション 110(c)に対する解説(PR システムの導入および活用)

PDI File に不具合があった場合は、セクション 110(c)に則り不具合管理を実施する。

（3） ソフトウェア変更時の対応

1） 開発中

ソフトウェアの不具合が発生した場合、対策後にセクション 110(a)～(c)の対応を行った上で、セクション 300 の飛行テスト時間を 0 時間にリセットするか否かについて申請者は検査者と協議すべきである。

2） 型式認証取得後

申請者の対応は以下の通りとなるが、図 4.7-2 に補足として対応フローを記述する。大変更、軽微変更の判断については、ソフトウェアの変更内容を航空局審査官と共有して調整した上で進める。

型式認証等を既に取得した型式の無人航空機にあっては、型式認証書類の内容に変更が生じた 場合、当該型式の型式認証等保有者は速やかに変更の内容を航空局に連絡し、型式認証の変更に 係る承認申請を行うこと。 型式認証書類に変更が生じるような変更は、原則として型式認証の変更の承認対象となる。型 式認証の変更の区分及び内容は、次の表に定めるとおりとする。

変更の区分	*変更の内容*
その他の変更（大変更）	*下記に掲げる変更以外の変更*
軽微変更	*当該型式の無人航空機に係る塗装の変更その他これに類する 安全性及び均一性に影響しない設計又は製造過程の変更*

［引用:国土交通省 サーキュラー No.8-002］

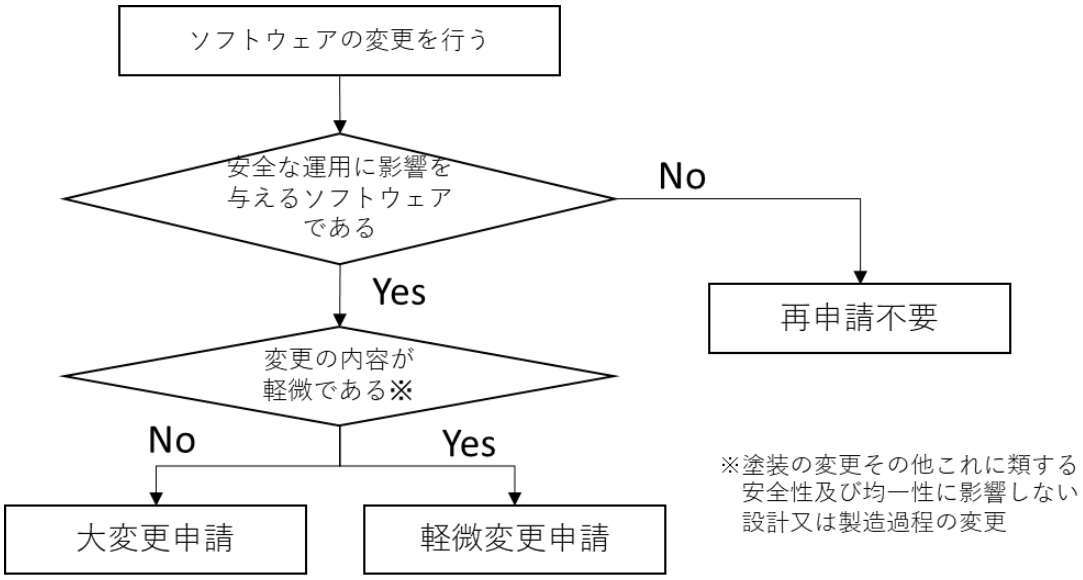


図 4.7-2 ソフトウェア変更時の対応フロー

3） セクション 110(a)に対する解説(ソフトウェアに対する試験による検証)

開発中および型式認証後に、ソフトウェアを変更する場合、まずは変更に伴う影響範囲を明らかする必要がある。確認の結果、必要なテストを検討して(a)項に則りテストを実施する。

4） セクション 110(b)に対する解説(ソフトウェアの形態管理)

開発中および型式認証後に、ソフトウェアを変更する場合、セクション 110(b)に則り形態管理を実施する。

5） セクション 110(c)に対する解説(PR システムの導入および活用)

開発中および型式認証後に、ソフトウェアを変更する場合、(c)項に則り不具合管理を実施する。

（4） 未認証の既存機体を型式認証取得する際の対応について

申請者が未認証の既存機体について、型式認証を取得することがあると想定して解説を行う。

1） セクション 110(a)に対する解説(ソフトウェアに対する試験による検証)

過去のテスト結果を引用できない場合は、セクション 110(a)に則り新たにテストを実施する。

2） セクション 110(b)に対する解説(ソフトウェアの形態管理)

未認証の機体においては、形態管理が実施されていないことがある。過去の再現が求められた場合は、セクション 110(b)に則った形態管理を開始した時点からの内容を提示する。

3） セクション 110(c)に対する解説(PR システムの導入および活用)

未認証の機体において、テスト記録は残っており、不具合が存在した記録があるが、その“不具合管理の記録”がない場合、再テストで不具合の修正が確認されていても、確実に修正されていることの確認が重要と考える。再テストなどを実施して不具合が修正されていることを確認し、レポート作成などで示せれば良いと考える。

5　今後の課題(未議論項目)

WG 活動で議論があったが未対応(今後対応予定)項目は以下の通り。

5.1 ソースコードを入手できない場合の扱いについて

4.7(1)において、現時点での記載を行ったが、ソースコードが入手できない場合の脆弱性対策、検証時に不具合が発生した際の対応などについて課題が残っている認識である。それ故、セクション 110(b)だけでなく、セクション 110(a)、(c)項における影響も併せて引き続き議論する必要がある。

さらに COTS(システム、ソフトウェア)、フリーウェアなどは市場実績も踏まえて、採用可否を検討する必要があると考えているが、深い議論まで行えていないため引き続き議論する必要がある。

5.2 PDI File の扱い

4.7(2)において、現時点での記載を行ったが、ライター間でも深い議論ができておらず、一般的なことは記載できている認識であるが、PDI 自体の詳細解説について図なども活用しながら解説を行う必要があると考えている。

5.3 セクション 110 ソフトウェアにおける PLD および ASIC の扱いについて

PLD および ASIC はロジックを持つハードウェアであるため、ソフトウェアと同様の性質を持つ。したがって、セクション 110 の範疇であるというレビュアーコメントも頂いているため、今後、セクション 110 の対象とするか否かについて引き続き議論が必要である。

Appendix 1　ソフトウェア適合性証明計画書　表紙記載例

※あくまで一事例であり、必ずしもこの通りに作成する必要はない

ソフトウェア適合性証明計画書

〇〇年●●月〇〇日

〇〇〇〇株式会社
〇〇〇部〇〇〇課
担当：（申請者　氏名）

記

- 安全に影響を与えるソフトウェア一覧　1 部
- 適合性証明計画日程概略　1 部
- 適合性証明計画表　1 部
- 自己宣言書　1 部

改定来歴

Eddition No.	変更箇所	変更内容	発行日
1.00	-	初版	〇〇年●●月〇〇日

Appendix 2　安全に影響を与えるソフトウェア一覧　記載例

※あくまで一事例であり、必ずしもこの通りに作成する必要はない

ソフトウェア名	バージョン	開発	搭載ハードウェア	適否	判断基準	備考
フライトコントロール	v2.1	AAAA Inc.	フライトコントローラ	適用	ソフトウェアの誤動作は、制御不能に陥り、危険性の高い状況を招く可能性があるため。	
グランド・コントローラ	v1.5	BBBB Technologies	GCS	適用	ソフトウェアの誤作動は、無人航空機の制御不能や誤操作につながる可能性があるため。	
モータ・コントローラ	v3.0	CCCC Solutions	モータ	適用	ソフトウェアにエラーや不具合が発生すると、高度制御ができなくなったり、飛行を維持できなくなったりする可能性があるため。	
バッテリー・マネジメント・システム	v1.2	DDDD Systems	バッテリーシステム	適用	ソフトウェアに不具合が生じた場合、電源関連の事故や飛行中の電源喪失につながる可能性があるため。	
ラジコンシステム	v4.0	EEEE Electronics	プロポ	適用	ソフトウェアに障害や干渉が発生すると、通信が途絶え、制御や状況認識ができなくなる可能性があるため。	
ホイストコントローラ	v1.1	FFFF Automation	ホイスト	非適用	ソフトウエアの欠陥や誤動作は、ホイストの安全でない操作につながるが、機体運航への影響は限定的であり、無人航空機の安全な運用に影響を与えないため。	

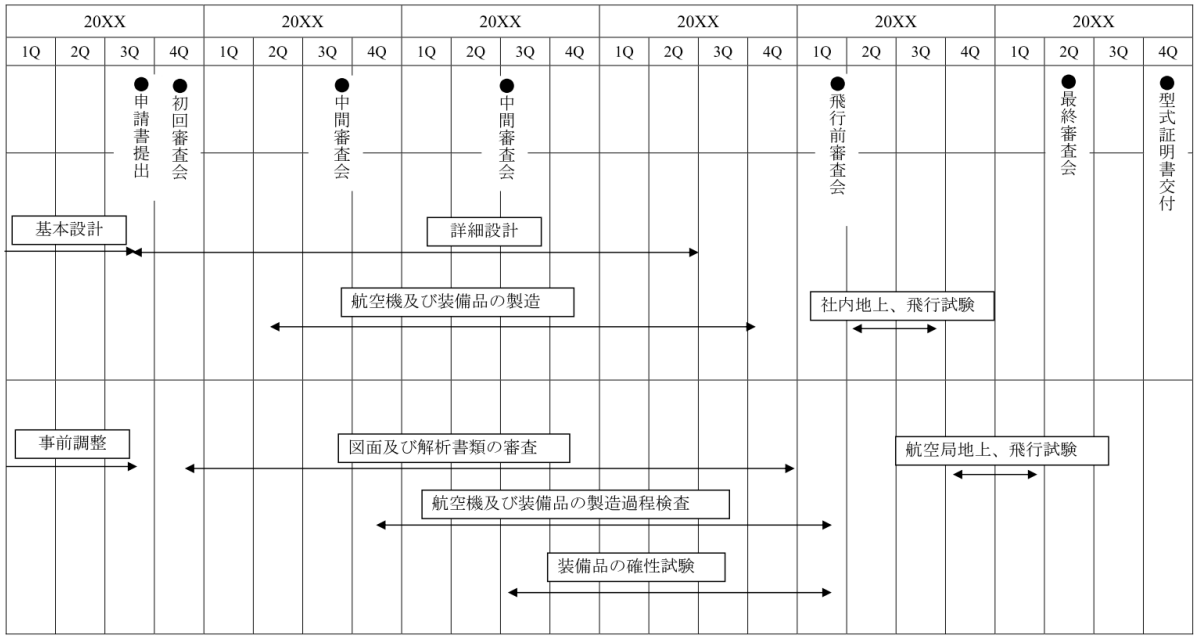
Appendix 3　ソフトウェア適合性証明計画日程概略　記載例

※あくまで一事例であり、必ずしもこの通りに作成する必要はない

○○ソフトウェア　適合性証明計画　日程概略

（設計及び製造に関する概略の日程の一例）　　別添 1

20XX 年 XX 月 XX 日　現在



1－307 (6)

Appendix 4　ソフトウェア適合性証明計画表　記載例

※あくまで一事例であり、必ずしもこの通りに作成する必要はない

○○ソフトウェア　適合性証明計画表

YYYY 年 MM 月 DD 日　現在

基準項目	活動	適否	証明方法	適合性証明文書		関連文書		備考	進捗状況	担当
				管理番号	名称	管理番号	名称			
①	②	③	④	⑤	⑥			⑦		⑧
(a)	要求ベーステスト	適	実技テスト	M001	xxx テスト仕様書	S001	xx システム仕様書			申請者
						S002	xx 要求定義書			申請者
(b)	形態管理	適								
(c)	不具合管理	否								

①　：適用基準などに掲げられた項目番号を記載すること。

②　：①の項目番号に対応した活動項目名を記載すること。

③　：今回の型式証明または型式設計変更にて証明する場合、以前に証明した内容が有効と判断する場合または同等性により証明する場合は「適」を記載し、未装備などの理由により証明が必要でない場合は「否」を記載すること。

④　：③にて「適」と判断した場合、図面、解析またはテストなどの証明方法を記載すること。

⑤,⑥　：④にて提案した証明方法にしたがって作成した文書または図面番号およびそれらの名称を記載すること。ただし、適合性証明計画には当該内容は未記入でよい。

⑦　：③にて「適」と判断した場合はその証明概要を記入し、「否」と判断した場合はその理由を記載すること。

⑧　：適合検査およびテスト立会を航空局と設計検査認定事業場で役割分担する場合、希望する分担を記載すること。

Appendix 5　宣言書記載例

※あくまで一事例であり、必ずしもこの通りに作成する必要はない

自己宣言書

文書番号：No.001-0001
文書発行日：YYYY.MM.DD

【発行者の名称】　xxx 株式会社
【発行者の住所】　愛知県名古屋市 aaa　123-456　〇〇ビル 5F

【宣言の対象】
製品名：　第二種　無人航空機 zzz
型式：　AAA-01

上記の宣言の対象はサーキュラーNo.8-001 にて規定されるセクション 110 ソフトウェア要件へ適合している。

〇〇ソフトウェア　Ver.aaaa（開発元：xxx 株式会社）

△△ソフトウェア　Ver.bbbb（開発元：yyy 株式会社）

□□ソフトウェア　Ver.cccc （開発元：株式会社 zzz）

署名

【問い合わせ先】　xxx 株式会社 yyy 部
【代表者役職・氏名】　yyy 部部長　テスト受け太郎

Appendix 6　ソフトウェア適合性証明完了報告書　記載例

※あくまで一事例であり、必ずしもこの通りに作成する必要はない

○○ソフトウェア　適合性証明完了報告書

YYYY 年 MM 月 DD 日　現在

基準項目	活動	適否	証明方法	適合性証明文書		備考	確認結果
				管理番号	名称		
①	②	③	④	⑤	⑥		
(a)	要求ベーステスト	適	実技テスト	M001	xxx テスト仕様書		
(b)	形態管理	適					
(c)	不具合管理	否					

署名

Appendix 7 各セクション特有の用語集

#	用語	解説	参考
1	オブジェクトコード	コンピュータプログラムの低水準な表現形式のこと。通常はターゲットコンピュータが直接利用できる形式ではない。プロセッサへの命令情報に加えて、リロケーション情報も含まれている。	DO-178C, ANNEX B, GLOSSARY
2	形態アイテム(またはアイテム)	(1)形態管理において 1 つの単位として扱われる 1 つ以上のハードウェアまたはソフトウェアのコンポーネントのこと。(2)形態管理において 1 つの単位として扱われるソフトウェア開発において作成されるデータのこと。	DO-178C, ANNEX B, GLOSSARY
3	欠陥	ソフトウェアエラーが顕在化したもの。欠陥があると、故障が引き起こされる可能性がある。	DO-178C, ANNEX B, GLOSSARY - Fault
4	Commercial-Off-The-Shelf (COTS) software	ベンダーが公開カタログリストを通じて販売する市販のアプリケーションのこと。また、特定のアプリケーション向けに開発された契約交渉済のソフトウェアは COTS ソフトウェアではない。	DO-178C, ANNEX B, GLOSSARY
5	実行可能オブジェクトコード (EOC)	Executable Object Code のこと ターゲットコンピュータの処理装置が直接使用できる形式のコードである。それ故、ターゲットコンピュータのハードウェアにロードされるビルドされたバイナリイメージである。	DO-178C, ANNEX B, GLOSSARY
6	ソフトウェアサプライヤ	ソフトウェアの供給者を指す。	
7	ソースコード	アセンブリ言語や高水準言語などのソース言語で書かれたコードのこと。ソースコードは、アセンブラやコンパイラに入力するため、機械可読形式である。	DO-178C, ANNEX B, GLOSSARY
8	ソフトウェア	コンピュータシステムの運用に関連するコンピュータプログラムおよびのこと。無人航空機および関連システムにロードされる実行可能オブジェクトコードとパラメータファイルのこと。場合によっては、関連する文書とデータを含む。	「DO-178C, ANNEX B, GLOSSARY」をベースに、本解説書向けにアレンジ
9	ソフトウェアテスト	ソフトウェアテスティングは、プログラムが、日常生まれる無限の実行ドメインから適切に選択されたテストケースの有限集合のうえで行われる、期待される振る舞いを提示するための動的検証から構成される。	SWEBOK V3.0 テストの定義
10	パラメータ データ アイテム ファイル (PDI ファイル)	Parameter Data Item File のこと。 ターゲットコンピュータの処理装置 (processing unit)によって直接使用できて、パラメータ データ アイテムを有するファイルである。	DO-178C, ANNEX B, GLOSSARY

#	用語	解説	参考
11	パラメータ データ アイテム(PDI)	実行可能オブジェクトコードを変更せずにソフトウェアの動作に影響を与えることができるデータの集合体のこと。	DO-178C, ANNEX B, GLOSSARY
12	ベースライン	ある時点における承認された 1 つ以上の形態アイテムのこと。ベースラインは、それ以降の開発の基準となる。	DO-178C, ANNEX B, GLOSSARY -
13	PLD (Programmable Logic Gate Array)	電子部品として購入され、アプリケーションの特定の機能を実行するように変更されたコンポーネントのこと。	DO-254, Appendix C, Glossary of Terms
14	ASIC (Application Specific Integrated Circuit)	特定の機能を実行するために開発された集積回路のこと。	DO-254, Appendix C, Glossary of Terms
15	API (Application Programming Interface)	あるソフトウェアが他のソフトウェアやハードウェアなどとやりとりをするために備えているインタフェースのことである。ここでいうところのインタフェースとは, 機能の呼び出し手順やコードの記述方法などを定めた仕様のこと。	API｜IS ディジタル辞典－重要用語の基礎知識－第二版(ipsj-is.jp)(一部抜粋)

Appendix 8 関連文書

(1) サーキュラーNo.8-001 無人航空機の型式認証等における安全基準及び均一性基準に対する検査要領、2022 年 9 月 7 日(国空機第 456 号。同年 12 月 2 日付け国空機第 645 号までの改正を含む。)、
https://www.mlit.go.jp/koku/content/001520547.pdf

(2) サーキュラーNo.8-002 無人航空機の型式認証等の手続き、2022 年 12 月 2 日(国空機第 645 号)、https://www.mlit.go.jp/koku/content/001574424.pdf

(3) 無人航空機の型式認証等の取得のためのガイドライン、2022 年 11 月 2 日、
https://www.mlit.go.jp/common/001574425.pdf

(4) RTCA DO-178C/EUROCAE ED-12C – Software Considerations in Airborne Systems and Equipment Certification

(5) ASTM F3153-15, Standard Specification for Verification of Avionics Systems

(6) ASTM F3153-22, Standard Specification for Verification of Avionics Systems

(7) RTCA DO-254/EUROCAE ED-80 – Design Assurance Guidance For Airborne Electronic Hardware

(8) JISX0020:1992-情報処理用語(システム開発)

(9) ISO/IEC/IEEE 29148:2018 Systems and software engineering Life cycle processes-Requirements engineering

(10) JSTQB Foundation Level シラバス 2.2 テストレベル、JSTQB・2024 年 2 月 13 日、https://jstqb.jp/dl/JSTQB-SyllabusFoundation Version2018V31.J03.pdf

(11) ARP4754A, Guidelines for Development of Civil Aircraft and Systems

(12) ARP4761, GUIDELINES AND METHODS FOR CONDUCTING THE SAFETY ASSESSMENT PROCESS ON CIVIL AIRBORNE SYSTEMS AND EQUIPMENT

Appendix 9　サブ WG の構成員名簿

無人航空機の第二種認証に対応した証明手法の事例検討 WG におけるサブ WG セクション 110 ソフトウェアの構成員名簿（サブ WG 主査およびライター）を以下に示す。なお、レビュワーの講成員名簿は本冊（RMD、Rev.01）Appendix4 を参照すること。

タイトル	氏名	所属
主査	南　貴紘	株式会社電通総研（旧株式会社電通国際情報サービス）
ライター	金子　達哉	DNV ビジネスアシュアランスジャパン
ライター	榧野　尊	株式会社電通総研（旧株式会社電通国際情報サービス）
ライター	中川西　拓	株式会社 SClabAir
ライター	西岡　亮	株式会社ベリサーブ
ライター	矢口　勇一	公立大学法人会津大学

無人航空機の型式認証等の取得のためのガイドライン解説書

発行日：2024 年 3 月

この成果は、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務（JPNP22002）
の結果得られたものです。