セキュア・ソフトウェア開発フレームワーク(SSDF)
導入ガイダンス案（中間整理）

経済産業省　商務情報政策局
サイバーセキュリティ課

令和７年３月４日

目次

1. 背景と目的

1.1. 背景

サイバーセキュリティを確保する上で、ソフトウェアのセキュリティ確保が根幹となる。産業活動のサービス化に伴い、産業に占めるソフトウェアの重要性は高まる傾向にある。特に、近年は、産業機械や自動車等の制御にもソフトウェアの導入が進んでおり、IoT 機器・サービスや 5G 技術においても、汎用的な機器でハードウェア・システムを構築した上で、ソフトウェアにより多様な機能を持たせることで、様々な付加価値を創出していくことが期待されている。

ソフトウェアのセキュア開発については、2000 年代以前より、セキュア・コーディングやソフトウェア・セキュリティ・アシュアランスなどの長年の取組により実践知、経験則が蓄積されてきた。SSDF(Secure Software Development Framework)[1]はそのような取組を踏まえて整理された標準である。QUAD4 ヵ国共同原則においては、セキュア・ソフトウェア開発プラクティスを加盟国政府調達に於いて基本ポリシーとすることが合意され、そのベースとなる SSDF は政府調達においても重要な要件となっている。

SSDF は、幅広い分野で適用できるように、汎用性の高い、包括的なフレームワークを提供するものであるが、一方で、抽象的で、導入手順が示されていないことから、組織において実践導入する上で具体的な方法やプラクティスが明確ではないといった課題がある。

このようなことから、SSDF を組織・プロジェクトの現場に導入するための具体的な方法や手順が求められている。

1.2. 目的

本ガイダンスは、セキュア・ソフトウェア開発フレームワーク SSDF を組織・プロジェクトの現場に導入するための具体的な方法や手順を示すことで、産業界におけるセキュア・ソフトウェア開発の向上を図ることを目的とする。

実ソフトウェアを対象に、SSDF の導入実証に基づく知見から、SSDF の具体的な導入方法、達成の判断指針と具体例を示す。また、SSDF 導入の参考情報として、SSDF と国内ガイドラインの対応関係を整理した情報を示すことで、すでに対応済みの国内ガイドラインを参考とした導入方法について考え方を整理する。

1.3. 対象読者

SSDF は、産業分野、開発プロセスを問わず幅広く汎用的に適用できるセキュア開発・運用における

[1] NIST, SP 800-218 Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218.pdf

実践策（プラクティス）を提供する。

サプライチェーンのライフサイクルプロセス全体を通じて、様々なステークホルダーに対して適用可能であり、発注者にとっては、供給者や開発委託先が導入する開発方法の確認事項として活用でき、供給者にとっては、調達者に対して導入する開発方法についての可視化、説明責任を果たすために活用できる。

1.4. 対象ソフトウェア

システム、製品、サービス、ライブラリを構成するソフトウェアを対象に幅広く、汎用的に適用可能である。また、委託開発や既製品などのパッケージソフトウェアにもよらず、ソフトウェア全般に幅広く適用することができる。

2. SSDF の概要

2.1. SSDF の位置付け

NIST SSDF は、ソフトウェアの脆弱性を軽減するためのソフトウェア開発者向けの手法をまとめたフレームワークである [2]。セキュア・ソフトウェア開発に関して米国を中心に長年にわたり蓄積された実践知、経験則を体系化したもので、ステークホルダーや部署の間の共通言語として利用可能である。

米国 OMB 覚書 M-23-16 において、SSDF に関して政府調達における自己適合宣誓書の提出が求められている。

また、QUAD4 ヵ国国際共同原則において、SSDF と同様の 4 つのプラクティスに基づく実践を政府方針に取り入れることを共同合意している。

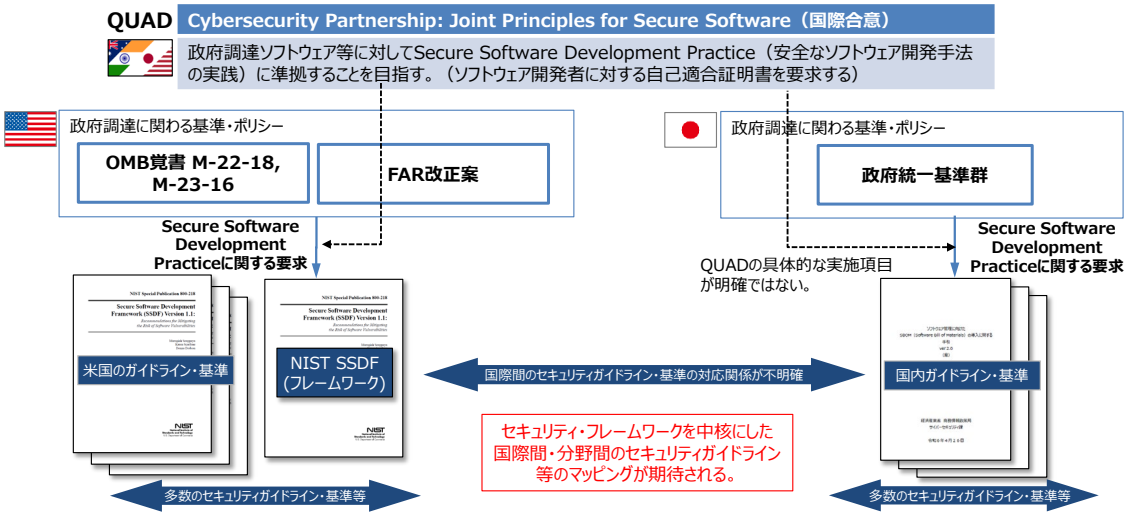


図 2-1 国際合意および日米政府調達における SSDF の位置付け

このように、SSDF は、セキュア・ソフトウェア開発における国際的な基盤として重要な役割を果たす標準である。

2.2. SSDF の全体像

2.2.1. プラクティス・タスクの全体像

SSDF の推奨事項は、プラクティスとタスクの 2 つの階層で整理されている。さらに、各推奨事項に対する実装例および参考文献が記載されているため、実装時にどのリソースを調査すべきかを簡単に参照することができる。

---

[2] NIST, SP 800-218 Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218.pdf

プラクティスは４つのカテゴリに大分類され、組織の準備（Prepare the Organization：PO）、ソフトウェアの保護（Protect the Software：PS）、安全なソフトウェアの作成（Produce Well-Secured Software：PW）、脆弱性への対応（Respond to Vulnerabilities：RV）から構成される。プラクティスは、さらに 2 階層のタスク（課題）に分類され、下記は、上位のタスクの要素を示している。

表 2-1 SSDF のプラクティス、タスクの上位構成

プラクティス（実践策）	タスク（課題）
1. **組織の準備** **（PO：Prepare the Organization）** ソフトウェアを開発する組織は、組織レベルで安全なソフトウェアの開発を行うために、適した人材、プロセス、技術を準備する必要がある。	PO.1: ソフトウェア開発におけるセキュリティ要件を定義する PO.2: ソフトウェア開発における内外組織の役割と責任を明確化する PO.3: ソフトウェア開発を支援するツールチェーンを明確化する PO.4: ソフトウェアのセキュリティを確認するための基準を定義し、活用する PO.5: ソフトウェア開発のための安全な環境を導入し、維持する
2. ソフトウェアの保護 **（PS：Protect the Software）** ソフトウェアを開発する組織は、ソフトウェアのすべてのコンポーネントを、改ざんや不正アクセスから保護する必要がある。	PS.1: あらゆる形態のコードを不正アクセスや改ざんから保護する PS.2: ソフトウェアリリースの完全性を検証する仕組みを提供する PS.3: 各ソフトウェアのリリースをアーカイブ化し、保護する
3. セキュア・ソフトウェアの開発 **（PW：Produce Well-Secured Software）** ソフトウェアを開発する組織は、脆弱性を最小限に抑え、十分なソフトウェアを備えたソフトウェアをリリースする必要がある。	PW.1: セキュリティ要件を満足するとともにセキュリティリスクを軽減できるよう、ソフトウェアを設計する PW.2: ソフトウェア設計をレビューし、セキュリティ要件やリスクへの適合性を検証する PW.3: サードパーティ製ソフトウェアのセキュリティ要件の準拠を確認する（ver.1.1 で改訂） PW.4: 可能な場合、機能を複製するのではなく、既存の保護されたソフトウェアを再利用する PW.5: セキュアコーディングのプラクティスを遵守してソースコードを作成する PW.6: 実行可能なセキュリティを向上させるために、コンパイル、インタプリター及びビルドプロセスを構築する PW.7: コードをレビュー・分析することで、脆弱性を特定し、セキュリティ要求事項への準拠を検証する PW.8: 実行コードをテストして脆弱性を特定し、セキュリティ要求事項への準拠を検証する PW.9: ソフトウェアをデフォルトで安全な設定とする
4. 脆弱性への対応 **（RV：Respond to Vulnerabilities）** ソフトウェアを開発する組織は、リリースするソフトウェアに残存する脆弱性を特定し、適切に対応する必要がある。	RV.1: 脆弱性に対する継続的な特定と確認を実施する RV.2: 脆弱性の評価、優先順位付け及び修正を実施する RV.3: 脆弱性を分析することで、その根本原因を特定する

これらにより、セキュア・ソフトウェア開発に関する組織の準備、関連ソフトウェアの保護、および開発プロセス、運用（脆弱性管理）プロセスに関する推奨事項全般を整理している。

2.2.2. 開発プロセス俯瞰イメージ

SSDF は、サプライチェーンのライフサイクルプロセス全体を通じて、様々なステークホルダーに対して適用可能であり、発注者にとっては、供給者や開発委託先が導入する開発方法の確認事項として活用でき、供給者にとっては、調達者に対して導入する開発方法についての可視化、説明責任を果たすために活用できる。

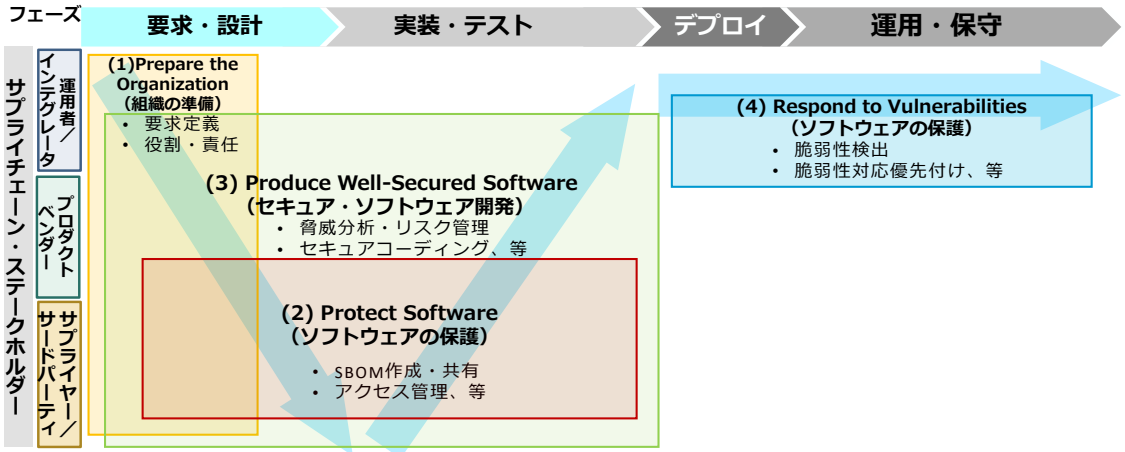


図 2-2 サプライチェーンを通じた開発プロセスと SSDF タスクの全体俯瞰

ステークホルダーごとにそれぞれのターゲット（システム、機器、ソフトウェア、部品等）に対して再帰的に同様の開発プロセスに応じた SSDF の適用スコープが存在する。

2.3. SSDF の基礎と特徴

セキュア・ソフトウェア開発プラクティスは、様々なソフトウェア開発ライフサイクル(software development life cycle: SDLC)に対してセキュリティ課題の解決策を付加・統合するものである。これにより、リリースされるソフトウェアの脆弱性件数を低減し、残留する脆弱性が悪用された場合の影響を低減し、脆弱性混入の根本原因を解消することなどに有効である。

SSDF に関するポイントと特徴を整理すると以下の通りである。

SSDF に関するポイントと特徴：

- 様々な開発プロセス、開発フロー、ツールチェーンに導入・統合可能な汎用なセキュア開発プラクティスである。
- ツール、技術、メカニズムなどによりどのように実現するか(How)ではなく、プラクティスの結果として何が達成されるか(What)に焦点を当てる。
- 既存の標準、ガイダンス、プラクティスをベースに、上位(High-level)のプラクティス（経験則）を体系化したもので、業種、組織規模、成熟度によらずあらゆる産業、コミュニティで汎用的に利用可能である。
- IT, ICS, CPS, IoT などのソフトウェアに幅広く適用できる。
- 特定の技術、プラットフォーム、開発言語、SDLC モデル、開発環境、OS、ツールなどの開発特性に依らず適用可能
- ウォーターフォールなど伝統的なソフトウェア開発モデルからアジャイル、DevOps など新しい開発

モデルに利用されるセキュア開発プラクティスへの移行を可能にする。

- ソフトウェアの調達者が、サプライヤにおいて導入するセキュア・ソフトウェア開発プラクティスを理解することを可能にする。調達者が、供給者の開発手法を把握するために用いることができる。
- SSDF は、セキュア・ソフトウェア開発プラクティスの基礎となる共通言語を提供する（NIST CSF と同様のアプローチ）。事業オーナー、開発者、PM、調達者など様々なステークホルダーが会話する共通言語となる。
- 開発者のセキュリティに対する前提条件に応じて、プラクティスの実装や形式化の度合いに違いが生じる。
- リスク、コスト、フィージビリティ、適用可能性は、どのプラクティスを利用するか決定する際に十分に検討が必要。
- SSDF を作成した意図は、リスクベース・アプローチの計画と実装の基礎を提供するものである。
- 機微情報、有資格者、セキュアなどの用語の意味は、各社の環境に応じて定義して用いる。
- 開発手法を文書化し、継続的改善プロセス、手法定義の一環として利用できる。

3. SSDF 導入の意義・メリット

SSDF は抽象的なセキュア開発フレームワークであり、抽象度が高いため、セキュア開発の経験が少ないと、その意図している内容や意義・メリットが分かりにくい。そのためここでは、SSDF 導入の意義・メリットについて明確化する。

- **体系的な対策による脆弱性の解消**

  セキュリティ・エンジニアリング分野における長年の実践知・経験則を集約した体系的なフレームワークによる網羅的な対策による弱点を解消する。

- **可視化を通じた説明責任の向上（アシュアランスの向上）**

  調達者、供給者の双方にとって、開発手法を可視化・把握できるようにし、説明責任の向上（不確実なリスクの低減）を図る。

- **共通言語によるコミュニケーションの促進**

  産業分野、開発言語、開発プロセスに依存しない共通言語を提供し、ステークホルダー間の理解・コミュニケーションを促進する。

- **プロセスの効率化**

  組織・ツール環境の整備によるセキュリティ・プロセスの効率化を実現する。

4. SSDF 導入の考え方と本書の位置付け

4.1. 本書の位置付け

本書は、SSDF に対して以下の価値を付加することにより、SSDF を開発組織・調達組織の現場に導入するための一助となるプロセスや方法を提供するものである。

SSDF に対する本書の付加価値

- SSDF に対して導入手順・プロセス、各タスクの具体策、留意事項を示す。
  - SSDF は、網羅的な対策の一覧であり、導入手順・プロセスやタスクの具体策を示していないため、本書ではそれらを付加している。
- アシュアランスとアカウンタビリティの考え方を示す。
  - SSDF は、達成レベルについて具体的に示しておらず、SSDF の適合性の判断が明確ではない。本書では、タスクごとの達成レベルの判断指針と具体的なプラクティス案を提示する。
- 各タスクにおけるツールや技術を具体的に例示した実践策を示す。
  - SSDF は、ツールや技術については限定せず、汎用的、抽象的なレベルでタスクを規定している。本書では、ツールや技術の選択肢や具体例を示すことで、導入者に具体的な指針を示す。
- ケーススタディを通じた具体例とコスト情報を示す。
  - SSDF では、タスクの具体例とコストについては情報を提示していない。本書では、具体的なソフトウェアを対象とした実証を通じてタスクの達成レベルの判断と選択したタスク達成レベルの実施に要した工数コストの具体例 [3]を示すことで、導入者の選択時の参考情報を提示する。

[3] 10 付録：SSDF 実証内容に示す。

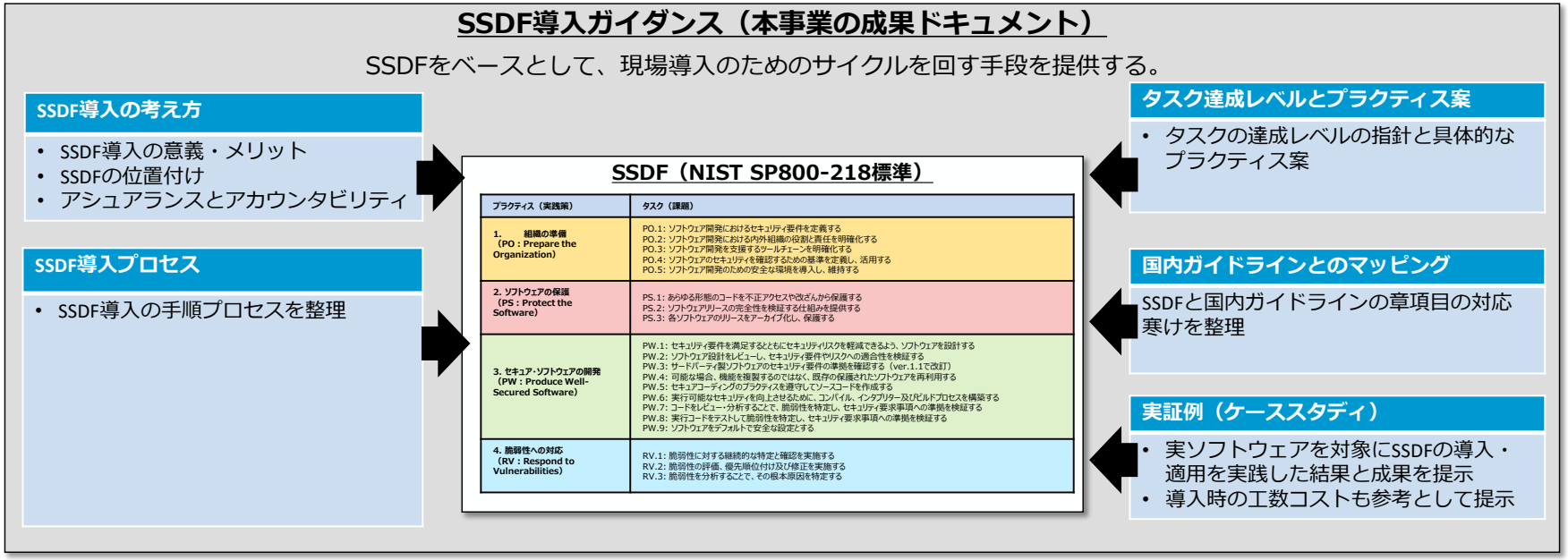


図 4-1 本書の位置付け（NIST SP800-218 との関係）

4.2. 本書における SSDF 導入の考え方

SSDF 導入の基本方針として以下の 3 点について順に説明する。

(1) 目標の明確化

(2) アカウンタビリティベース・アプローチ

(3) 残留リスクの把握

4.2.1. 目標の明確化

SSDF は、ソフトウェアのセキュア開発と脆弱性管理などのプラクティスを体系化したものであり、スコープが広範であることから、プロジェクトに導入するにあたっては、その目的を明確して適切な範囲を設定することが重要である。例えば、政府調達基準や、民間分野におけるソフトウェア開発委託において調達者が供給者のソフトウェア開発のセキュリティ手法の実施状況を把握するための達成レベルの判断指針として使うことができる。供給者にとっては、自社の製品・サービスのセキュリティ水準の高さを合理的に示す手段として用いることができる。求められるセキュリティ水準は、分野によって大きく異なるため、妥当なセキュリティ対策を選別して実施することが重要になる。例えば、重要インフラや人命に関わるソフトウェアについては、高いセキュリティレベルが要求される分野においては、SSDF の全てのタスクを実施することが期待されるが、それ以外の領域のソフトウェアに対しては、コストの関係から必ずしも全ての SSDF タスクを実施することが妥当とは言えない。また、全てのタスクについて、どこまで対策を徹底するかそのレベルには大きな幅があり、適正なレベルは、ソフトウェアの用途や利用環境によって決まるリスクに応じて適切に判断することが求められる。このように達成レベルの判断指針を利用する用途、産業分野、セキュリティ要求水準に応じて、タスクの範囲とレベルに関する目標を適切に設定して対策を実施することが重要になる。

4.2.2. アカウンタビリティベース・アプローチ

本ガイダンスでは、タスクの達成レベルの設定とその達成内容に関する説明責任（アカウンタビリティ）を果たすことでセキュリティを確保する「**アカウンタビリティベース・アプローチ」を新たに提案**する。本書では、タスクの達成レベルの指針や具体的なプラクティス案を示すことで、ソフトウェアに関わる事業者の対策について選択肢を示すが、その選択と実施した内容の説明責任は事業者が果たすという考え方を基本方針とする。理由は、ソフトウェアに対する責任は、製品・サービスを提供する事業者が自ら負うものであり、リスク、収益性を含む総合的な意思決定が可能なのは事業者自身であり、他のステークホルダーではありえないという考え方による。重要インフラ分野など、リスクの大きい分野においては、利用者保護の観点から、一定の義務的な基準を求めることは必要であるが、その基準を満たしたとしても十分という保証が得られるわけではなく、自らの意思で製品・サービスを提供する事業者は最終的な責任を負うことが基本原則である [4]。

本書では、SSDF のタスクの達成レベルを 3 段階に分けて、その指針と実施すべき主な対策を例示するが、どのタスクを対象にどのレベルまで実施するかについて決定するのは事業者自らである。行政機

[4] 重要インフラのサイバーセキュリティに係る行動計画、2022 年 6 月 17 日、サイバーセキュリティ戦略本部

関を含む事業者以外のステークホルダーが達成レベル一律に義務化するのではなく、責任を負う事業者が自らレベルを選択し、そのレベルの指針とプラクティス案に照らして実施した内容について客観的に説明責任を果たすことが求められる。このような対策アプローチを、本書では、「アカウンタビリティベース・アプローチ」と呼ぶ。「アカウンタビリティベース・アプローチ」は、**リスクベース・アプローチをベースとしつつ、その課題を解消し、発展形**として位置づけることができる。**リスクの定量化が困難であるという前提に立ち、リスク量に基づき対策を何段するのは難しく、対策の妥当性の説明責任を果たせるかどうかに基づき判断するアプローチ**であることがポイントとなる[5]。実施すべき対策は固定的ではなく、達成すべき要求に対して対策の妥当性を説明することで柔軟に運用できる点がポイントである。「アカウンタビリティベース・アプローチ」は、会計基準における**原則主義アプローチ**(**Principle-based Approach**)[6]の **"Comply or Explain"**の考え方を導入し、原則に対して準拠するか、そうでない場合には、その妥当性を説明することで柔軟性、実質性を確保するものである。原則に対して、一律に基準を強制するのではなく、企業が定められた原則を遵守するか、遵守しない場合はその理由を説明することで柔軟で、実質を重視した運用を可能にする。

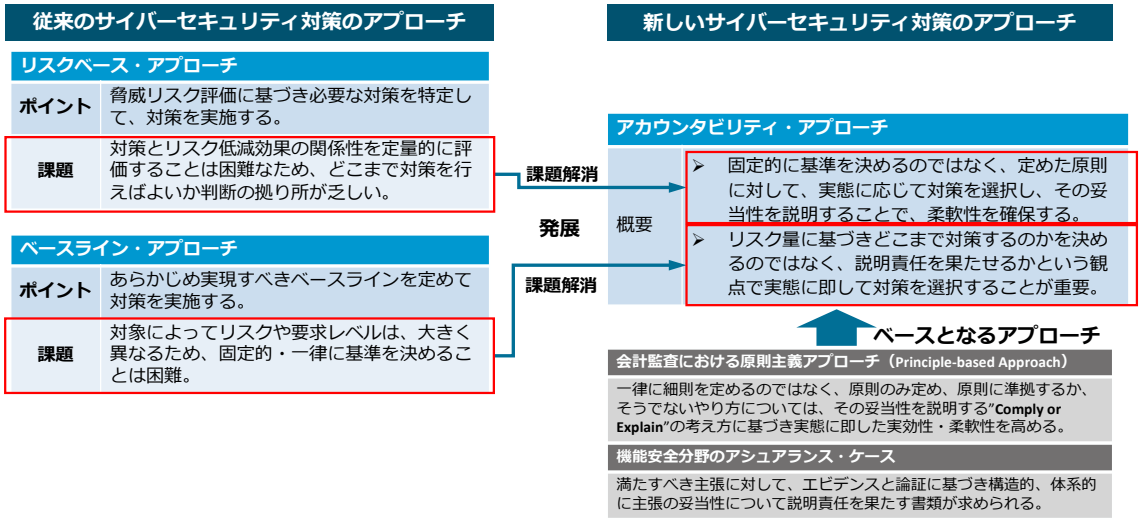


図 4-2 既存のセキュリティアプローチとアカウンタビリティベース・アプローチの関係（コンセプト）

4.2.3. 残留リスクの把握

SSDF は、ソフトウェア・セキュリティに係る実践則・経験知を網羅的に集約、体系化したものであるが、ハイレベルのプラクティスを示すものであり、これらのプラクティスを実施したからと言って、当然ではあるがリ

[5] ISO 21434 :2021 Road vehicles — Cybersecurity engineering などにおけるサイバーセキュリティ・アシュアランス・ケース（サイバーセキュリティ論証）の考え方と共通する。

[6] Principle-based Approach： IFRS（International Financial Reporting Standards）などの会計基準において、細部の規則を規定するのではなく、原則のみを示し、その原則に準拠していることについて妥当性を説明する考え方である。一律の外形的な基準に不要なコストをかけるのではなく、運用の柔軟性が高く、実質的な対策を優先的に確保できることが特徴である。

スクがゼロになることを保証するものではない。例えば、脆弱性検査については、動的検査、静的検査、構成解析ベースの既知脆弱性検査など含まれるが、いずれの技術も完全性は保証されておらず、技術的な課題が大きいことが知られている。これらの検査の完全性を高めようとすればするほど、コストは急速に増大し、どこまで実施するかについては、ソフトウェアの便益、リスク、コストを総合して判断することが必要である。このようなことから SSDF のタスクを網羅的に実施したり、最高レベルのタスク達成レベルを実施していても、技術的な課題や限界に伴うリスクは存在する。本書では、SSDF の全てのタスクについて３段階のレベル分けの指針と具体例を示すことで、その達成度を可能な限り具体的に説明する手段を提供しており、全体としての達成度の分布を把握することで、対象ソフトウェアのセキュリティレベルと残留リスクを可視化するための一つの手段を提供する。

5. SSDF 導入プロセス

5.1. SSDF 導入プロセス全体像

組織にSSDFを導入するための手順として、以下のようなプロセスとして全体を整理することができる。

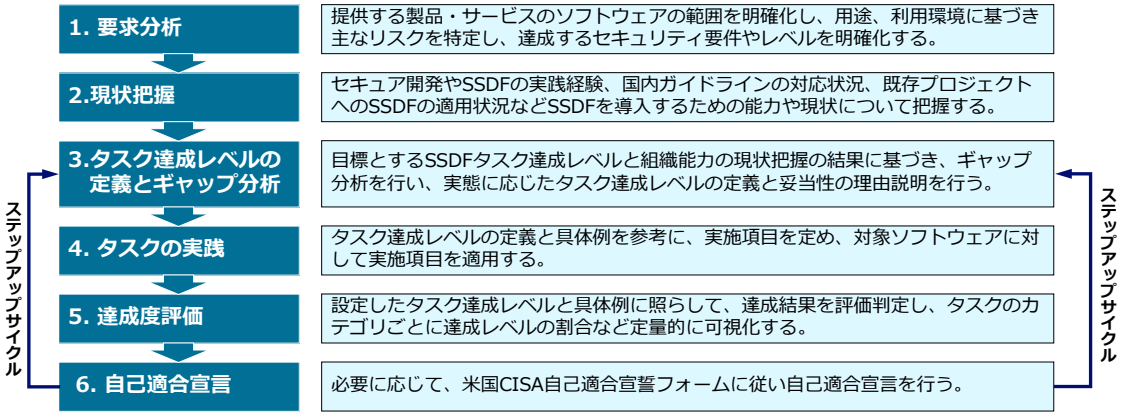


図 5-1 SBOM 導入プロセス

以下の節で導入プロセスの各フェーズの実施内容、手順等についてまとめる。

5.2. SSDF 導入プロセスのフェーズ実施

SSDF 導入プロセスの各フェーズについて以下の節で示す。

5.2.1. 要求分析

インプット：対象システムの分野、用途、リスク、利用環境、要求レベル 活動：各ステップに記載 アウトプット： SBOM 導入プロセスへのインプット（対象ソフトウェアの範囲、用途・環境、主なリスク、要求レベル、サイバーセキュリティゴール）

対象システムや調達・開発に関わる組織に関する情報をインプットとし、5.SSDF 導入プロセスに必要となるインプット情報をアウトプットとして整理する。タスクの達成レベルを判断する上で必要となる情報を整理する。アウトプットには以下の情報を含める。

要求分析のアウトプットと実施内容：

[1] 対象ソフトウェアの範囲

対象ソフトウェアは、インフラやサービスとして提供される事業者固有のシステムや、汎用品としての製品など多様であり、その境界を明確化する。また、OS、ミドルウェア、外部連携クラウドなども対象とするか、責任を負う対象範囲とその境界を明確化し、主要な構成要素を明確化して文書化する。

[2] 用途・利用環境

対象システムの用途、利用環境を明確に文書化する。リスクに関わる事項についても文書化する。

[3] 主なリスク

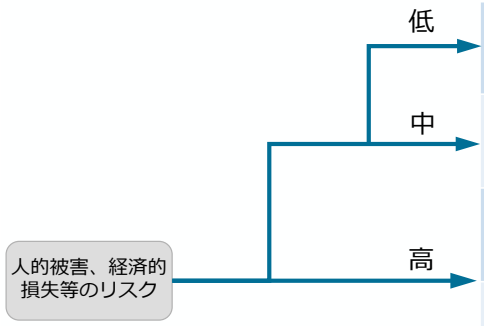
対象ソフトウェアが受ける可能性のある攻撃・脅威、その発生可能性、事故に至った場合の波及的な影響を特定し、文書化する。詳細については、5.SSDF 導入プロセスにおいて脅威リスク分析において実施するため、ここではシステム全体のハイレベルな要素について特定する。

[4] サイバーセキュリティ・ゴール（達成事項）

SSDF 適用により達成する目標を明確化する。主な脅威に対して防御により維持する状態や要求を明確化する。また、リスクに対して保証すべき事項についての言明（クレーム）を明確化する。

[5] 要求レベル

上記[1]～[4]に基づき対象システムに対する要求レベルを大枠で判定する。要求レベルは、システムに対するリスクをベースに以下のような例を参考に 4 段階程度で分類し、どのレベルに該当するか判断する。SSDF は政府調達に限定されるものではなく、あらゆる分野のソフトウェアを対象とした汎用的なフレームワークであるため下記の通り幅広い分野を対象とする。



リスク	分類分析	システムの分類	想定産業分野、システム等の例	要求レベル
人的被害、経済的損失等のリスク	低	**(Type I)** 社会的影響が殆どない・影響は、限定的。個社、個人に留まる。	一般コンシューマー製品等。	レベル1
	中	**(Type II)** 社会的・波及的な影響がある。	特定のインフラ産業、高シェア大手ベンダー等	レベル2
	高	**(Type III)** 社会的影響が極めて大きい。	基盤的重要インフラ、電力、金融、通信分野等。	レベル3
		(Type IV) 人命に影響、甚大な経済的損失。	医療機器、自動車、航空機、原子力、石油化学プラント	

（出所）高信頼化ソフトウェアのための開発手法ガイドブック、情報処理推進機構を元に加筆

図 5-2 リスク・影響度に基づくシステムの分類と要求レベル

5.2.2. 現状把握

対象システムと導入組織について、SSDF タスクの導入状況と組織の経験について現状を把握する。SSDF タスクごとに以下の 3 項目について現状を特定する。区分としては、組織・要員の経験と対象システムへの提供状況に分かれる。組織・要員については、これまでに該当項目について組織・要員としての経験を特定する。対象システムについては、既存システムか新規開発システムかによって確認方法はことなる。開発中または開発済のものに、SSDF を適用する場合、当該タスクをすでに適用済みかどうか

確認する。新規プロジェクトで開発する場合には、対象システムへの SSDF タスクは適用していないことになる。

表 5-1 タスクごとの現状把握項目

区分	項目	内容
組織・要員の経験	SSDF タスクの適用経験	組織・要員が当該タスクの適用経験がある。
	SSDF タスクに対応する国内ガイドラインの項目対応有無	当該タスクについて、の項目のうち、実施経験があるものを全て特定する。
対象システム	対象システムへの SSDF タスクの適用有無	対象システムが開発中または開発済の場合、当該タスクの適用有無

以上の状況を全てのタスクについて確認・判断することで、5.2.3 タスク達成レベルの定義とギャップ分析の適用フェーズにおいて、タスク達成レベルの決定の際に用いる。

タスク達成レベルの定義とギャップ分析に応じて、対象システムへのタスクの適用状況または適用可能性について判断する。対象システムについて既に開発中で、SSDF のタスク項目を適用を行っている場合には、それらの状況を確認する。また、適用可能性については、自組織におけるソフトウェアセキュリティの適用能力を判断する。その際に、自組織がこれまでに導入済のガイドライン等を特定し、9 付録：SSDF と国内ガイドラインのマッピングを参考に SSDF タスク項目への対応済/未済の状況を把握する。タスク項目ごとに以下のような分類で現状把握結果を整理する。タスクの達成レベルとプラクティス案を参考に 5.2 要求分析に基づき対象製品・サービスについて目指すタスクレベルを設定し、5.3 現状把握との比較からタスク実施能力の不足について明らかにするためギャップ分析を行う。

表 5-2 SSDF タスク項目ごとの対応状況の判断区分と観点

判断項目	対応状況	
(1)対象システムへの適用状況	適用済、部分適用、未適用	
(2)導入済ガイドラインによる能力有無	対応ガイドライン名と該当章（9 付録：SSDF と国内ガイドラインのマッピングを引用）	該当章の包含関係とその理由説明（9 付録：SSDF と国内ガイドラインのマッピングを引用）
(3)ギャップ分析	(1),(2)を元にタスクの対応状況を高、中、低などで判断し、理由をまとめる。	

以上のギャップ分析結果に基づき、5.2.4 においてタスクに必要な準備と実践を行う。

コラム

本ガイダンスでは、タスク毎のタスク達成レベルの決定において、まず、全体の目標レベルを設定し、それに対して、対象システムのリスク、組織の現状分析に基づき、実態に応じてタスク達成レベルの修正を行い、その妥当性の理由説明を行った。このような手続きを踏む理由は、タスク達成レベルの定量的な判断が困難であるためである。ここでは、タスク達成レベルの定量的な決定における困難について参考情報を提示する。

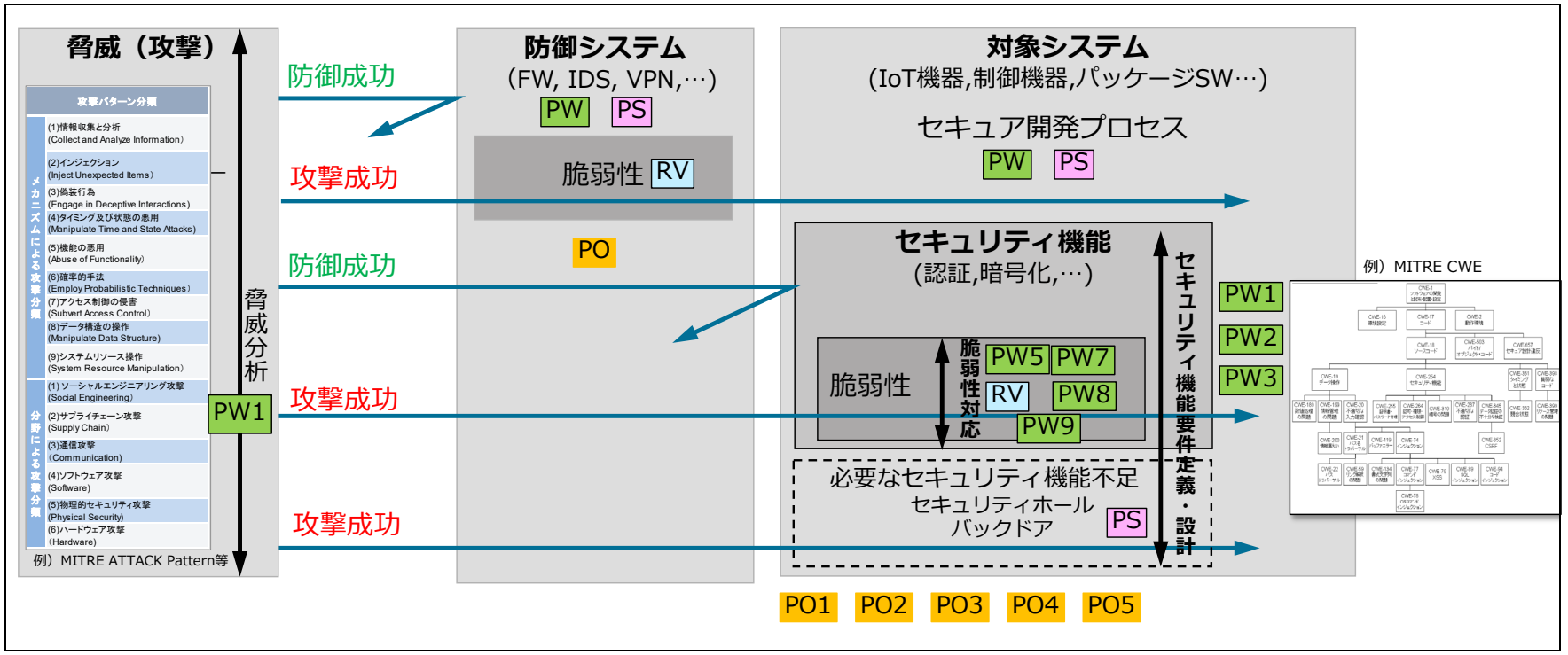
セキュア開発の最終目標は２つに分けられる：

(1)セキュリティ機能が正しく実現されていることの保証、

(2)セキュリティ機能およびその他機能の脆弱性が許容リスクに抑えられていること、

未知の脅威に対して対策（タスク）とリスク低減効果の関係性を定量的に示すことは困難である。また、セキュリティ機能が、脅威に対して十分に対応できるように網羅されているかについても、未知の脅威を考えると定量的に評価することは困難である。また、静的解析、動的解析、構成解析による脆弱性対策が未知の脅威を含めて必要十分であるか定量評価は困難である。以上のことから、対策とリスクの関係性を定量的に示すことは困難であるため、対策レベルは定量的に決定することは難しい。

参考に、脅威・攻撃、防御システム、対象システムにおける SSDF タスクによる対策の俯瞰的な関係を示したものが、下図である。


脅威（攻撃）
攻撃パターン分類
メカニズムによる攻撃分類
(1)情報収集と分析 (Collect and Analyze Information)
(2)インジェクション (Inject Unexpected Items)
(3)偽装行為 (Engage in Deceptive Interactions)
(4)タイミング及び状態の悪用 (Manipulate Time and State Attacks)
(5)機能の悪用 (Abuse of Functionality)
(6)確率的手法 (Employ Probabilistic Techniques)
(7)アクセス制御の侵害 (Subvert Access Control)
(8)データ構造の操作 (Manipulate Data Structure)
(9)システムリソース操作 (System Resource Manipulation)
分野による攻撃分類
(1)ソーシャルエンジニアリング攻撃 (Social Engineering)
(2)サプライチェーン攻撃 (Supply Chain)
(3)通信攻撃 (Communication)
(4)ソフトウェア攻撃 (Software)
(5)物理的セキュリティ攻撃 (Physical Security)
(6)ハードウェア攻撃 (Hardware)
例）MITRE ATTACK Pattern等
脅威分析
PW1
防御成功
攻撃成功
防御成功
攻撃成功
攻撃成功
防御システム
(FW, IDS, VPN,…)
PW
PS
脆弱性 RV
PO
対象システム
(IoT機器,制御機器,パッケージSW…)
セキュア開発プロセス
PW
PS
セキュリティ機能
(認証,暗号化,…)
脆弱性
脆弱性対応
PW5
PW7
RV
PW8
PW9
必要なセキュリティ機能不足
セキュリティホール
バックドア
PS
セキュリティ機能要件定義・設計
PW1
PW2
PW3
例）MITRE CWE
PO1
PO2
PO3
PO4
PO5

5.2.3. タスク達成レベルの定義とギャップ分析

本フェーズでは、タスク毎のタスク達成レベルを決定する。以下の入力、活動、出力により定義される。

入力：

(1)5.2.2 現状把握で特定した各タスクに関する能力・適用状況

(2)8 付録：SSDF に対するタスク達成レベルと具体例

(3)5.2.1 要求分析

活動：

(1) 8 付録：SSDF に対するタスク達成レベルと具体例に基づき、目標レベルを設定

(2)5.2.2 現状把握で特定した各タスクに関する能力・適用状況や、5.2.1 要求分析のリスクを考慮して、ギャップ分析を行い、タスク達成レベルを決定する。決定したタスク達成レベルの妥当性の理由説明を作成する。

出力：

タスク毎に決定したタスク達成レベル

タスク達成レベルの検討手順は以下のようになる。

1. **基本目標レベルを設定**
   「タスク達成レベル判断表」をもとに対象システムについてタスク全体の基本となる目標レベルを設定する。
2. **達成レベルの決定**
   タスク毎の達成レベルの判断指針と具体例を参考に、セキュア開発の経験や国内ガイドラインの対応状況をもとに、実態に合わせたレベルの見直しとその妥当性の理由説明を示して、各タスクの達成レベルを決定する。
3. **優先タスクレベル設定**
   レベル 1 はセキュア開発の先決課題であるため、現状でレベル 1 に達しないタスクがあれば優先対応する。
4. **レベル達成計画の設定（目標レベル達成）**
   目標レベル 2 以上で、達成しないタスクがあれば実施計画（POA&M)を検討する。
5. **レベル達成計画の設定（上位レベル達成）**
   現在の能力・リスク・コストに照らして目標レベル以上のレベルを狙う場合、その実施計画を策定し実施する。

(1)　タスク達成レベルの目標レベル設定

5.2.1 要求分析で明確化したリスク等情報をもとに、タスク達成レベルの標準的な目安を示す下記の「タスク達成レベル判断表」を元に基本とするタスク全体の目標レベルを設定する。

表 5-3 タスク達成レベル判断表

タスク達成レベル	（判断観点１） リスク	（判断観点２） システムの特徴と要求レベル	（判断観点３） 関連法令有無等	（判断観点４） 対象企業等
レベル３	人命に影響、経済的に甚大な影響 （航空機、医療機器、自動車、電気、金融、通信）	安全性、セキュリティ、信頼性等について最も要求レベルが高いシステム	国内外法令・業法等 （法令準拠は前提）	セーフティクリティカルシステムのベンダー、基幹インフラのベンダー、サプライヤー（中小企業を含む）
レベル２	社会・産業に波及的な影響 (上記以外の重要インフラ)	社会的基盤であり、産業や多数の国民に影響を与えるシステム	業界基準・ガイドラインなど	重要インフラのベンダー、サプライヤー（中小企業を含む）、事業範囲の広い大企業
レベル１	社会的影響が殆どない・限定的 (影響は個社、個人に限定的)	一般コンシューマ製品、経済的損失を伴わないシステム	任意基準等	上記以外の中小企業等

ここでは、目標レベルの判断目安として、リスク、システムの特徴と要求レベル、関連法令有無、対象企業などの判断観点を示しており、それらを総合してレベル１～レベル ３ のどれに最もよく当てはまるか判断し、目標レベルを設定する。

(2) ギャップ分析に基づくタスク達成レベルの決定

タスク毎の達成レベルの判断指針と具体例を参考に、セキュア開発の経験や国内ガイドラインの対応状況をもとに、実態に合わせたレベルの見直しとその妥当性の理由説明を示して、各タスクの達成レベルを決定する。

「タスク達成レベル判断表」に基づく目標レベルに対して、組織・要員のセキュア開発経験、国内ガイドライン対応実績、対象システムへの適用レベルなどの現状把握を行い、目標レベルとのギャップ分析に基づき、実態に応じて必要なタスク達成レベルを決定する。

目標レベルを達成しない場合、決定したレベルの妥当性の理由説明または、目標レベルを達成するまでの期限や方法を明確にする。

タスク	目標レベル	セキュア開発経験有無	国内ガイドライン対応有無	既存システムへの適用レベル	タスク達成レベル決定	妥当性の理由説明	目標レベル達成計画
PO.1.1	2	あり	METI CPSF CPS.GV-2,...等	1	2	（目標レベルに準拠）	―
PO.1.2	2	あり	METI CPSF CPS.GV-2,...	1	2	（目標レベルに準拠）	―
PO.1.3	2	なし	なし	なし	1	セキュリティ要求についてOSS提供者との合意は困難であるが、付属文書で確認する。	１年後の次期メジャーアップデートまでに、OSS部品の特定と付属文書確認を行う。
・・・	・	・・	・・・	・・	・	・・・	・・・
PS1.1	2	あり	METI SBOM導入手引き5.3	1	1	利用者側の検証環境の確保までは難しいが、完全性検証可能な署名データを提供可能。	―
・・・	・	・・	・・・	・・	・	・・・	・・・

「タスク達成レベル判断表」に基づく目標レベル

現状の組織・要員のレベル 対象システムへの適用レベル

ギャップ分析

実態を踏まえたタスク達成レベルと妥当性の理由説明

決定したタスク達成レベルが未達の場合の達成計画

図 5-3 目標レベルと現状のギャップ分析に基づく達成レベルの検討方法

8 付録：SSDF に対するタスク達成レベルと具体例、を参考にして、5.2.1 要求分析、に基づき、SSDF タスク項目ごとに対象システムにおいてするタスクレベルを決定する。5.2.3 タスク達成レベルの定義とギャップ分析においては、SSDF タスク項目ごとに３段階のレベル分けを行い、レベルの違いを決定する考え方を判断指針を明瞭に示すとともに、各レベルで実施する対策の具体例を示している。これらを参考に対象システムとそれに対するリスクや要求レベルを考慮して、妥当なレベルを選択する。達成レベルは、利用者、調達者に対してどのような対策を実施したか可視化、明確化することが重要であり、そ

の説明責任を果たすことができれば、レベルの定義を必要に応じて変更しても構わない。ただし、その場合、全体としての達成度を可視化する場合、標準とする達成レベルとは異なるため、比較ができなくなることに注意が必要である。

タスク達成レベルの目標設定は、いきなり最終目標を設定するのではなく、現状能力と能力向上、出荷スケジュールに応じて段階的な目標を設定する。

SSDF のタスク全体を知ってから、個社やシステムの用途の実態に応じて、必要性を判断することが重要である。

5.2.4. タスクの実践

設定したタスク達成レベルに基づきタスクの実践を行う。

(1) タスクの実施順序の定義

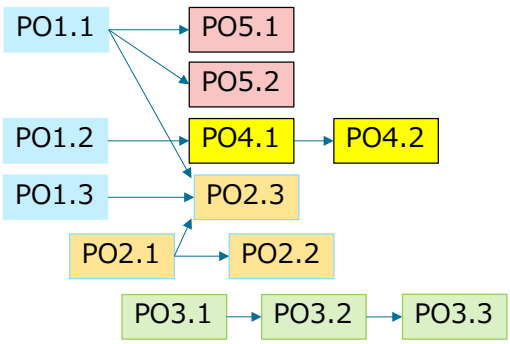
SSDF 標準においてはタスクの掲載順序は、実施順序や依存関係を示すものではないと記載されているが、実際には、タスクのカテゴリ間やカテゴリ内の各タスクには典型的な実施順序、依存関係があるため、以下のような順序で実践することが効果的である。

SSDF タスクカテゴリについては 1.組織の準備(PO)は、他のカテゴリの前提となる組織の基盤的な準備であるため、最も先に実施する。2.ソフトウェアの保護(PS)は、開発するソフトウェアの保護に関するもので、開発を実施する前に整える。それらを整備した後で、3.セキュア・ソフトウェアの開発(PW)、4.脆弱性への対応(RV)へと進む。これらのカテゴリの中の個々のタスクによっては、必ずしもカテゴリの順序に従わなくても実施できるものがあるが、概ね、このカテゴリの順序で実施するのが妥当である。

次に、個々のカテゴリ内のタスクの依存関係については概ね以下のように想定されるが、必ずしもこの順序が必須という訳ではない。

1) 組織の準備(PO)のタスクの順序関係

PO1.1,1.2,1.3 が他のタスクのベースとなる。タスクの依存関係としては一般的には以下のような関係が想定されるが、個別事例によって状況が異なることも考えられる。これらの関係を考慮して実施できるタスクから順に実施することでよい。

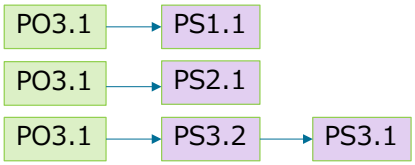


（SSDF 実証に基づき作成）

図 5-4 SDF タスクカテゴリ PO におけるタスクの標準的な順序関係・依存関係

2) ソフトウェアの保護（PS）のタスクに関わる順序関係

PS タスクは PO タスクのうち特に導入するツールに依存する部分が大きい。よって、PO3.1 タスクがベースとなり、それぞれ対応内容が決まってくる。タスク対応順としては PS1.1 が先行し、それ以外は順不同でよい。これについても個々の事例により状況は異なる可能性がある。

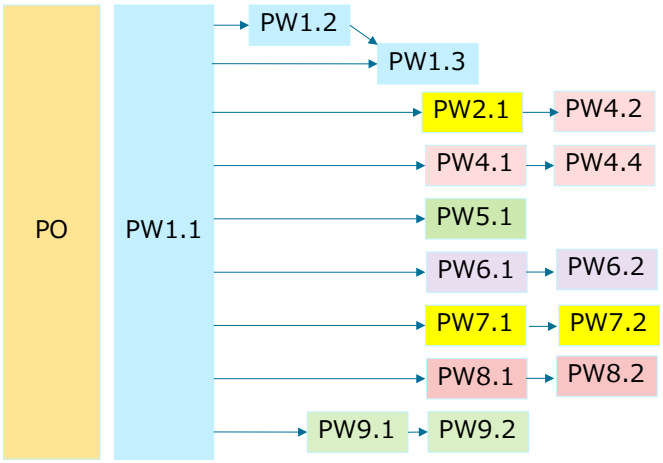


（SSDF 実証に基づき作成）

図 5-5 SSDF タスクカテゴリ PS のタスクに関わる標準的な順序関係・依存関係

3) セキュア・ソフトウェアの開発（PW）のタスクの順序関係

PW タスクはまず PW1.1 のリスク評価が全体の中で優先度が高く、PO もベースとして各 PW に対応することが想定される。

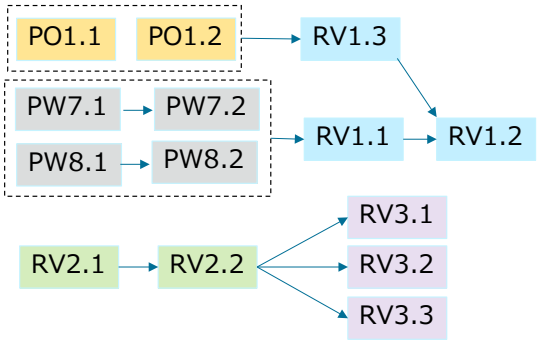


（SSDF 実証に基づき作成）

図 5-6 SSDF タスクカテゴリ PW のタスクに関わる標準的な順序関係・依存関係

4) 脆弱性への対応（RV）のタスクの順序関係

RV タスクでは、RV1.3 については製品リリース後の脆弱性対応の基本ポリシーとなるため優先的に対応してくのが望ましい。それ以外についてはタスクの記載順に対応していくことが想定される。



※点線枠は、全体として、矢印先の前に実施することが想定される。

（SSDF 実証に基づき作成）

図 5-7 SSDF タスクカテゴリ RV のタスクに関わる標準的な順序関係・依存関係

(2) アカウンタビリティベース・アプローチによるタスクの実施内容の定義と実施

5.2.3 タスク達成レベルの定義とギャップ分析において設定した達成レベルに対応する原則・指針となるタスク達成レベルの指針、具体例を参考にして、SSDF の実施項目を定義し、それらを実施する。下図において、①～④の情報を参考に、⑤において対象システムや組織の実態に合わせてテイラリングを行い、⑥において、定義した実施内容が、SSDF タスク説明や本事業のタスク達成レベルの基本指針に対して妥当性の理由説明を行う。ここでは、アシュアランスケースのような厳格な論証を必須とするのではなく、タスク達成レベルに示す具体例から取捨選択を行う場合の簡易な理由説明が示されることで、受発注者間での透明性の確保に有益と考えられる。必要に応じて⑦のステップアップサイクルを回す。

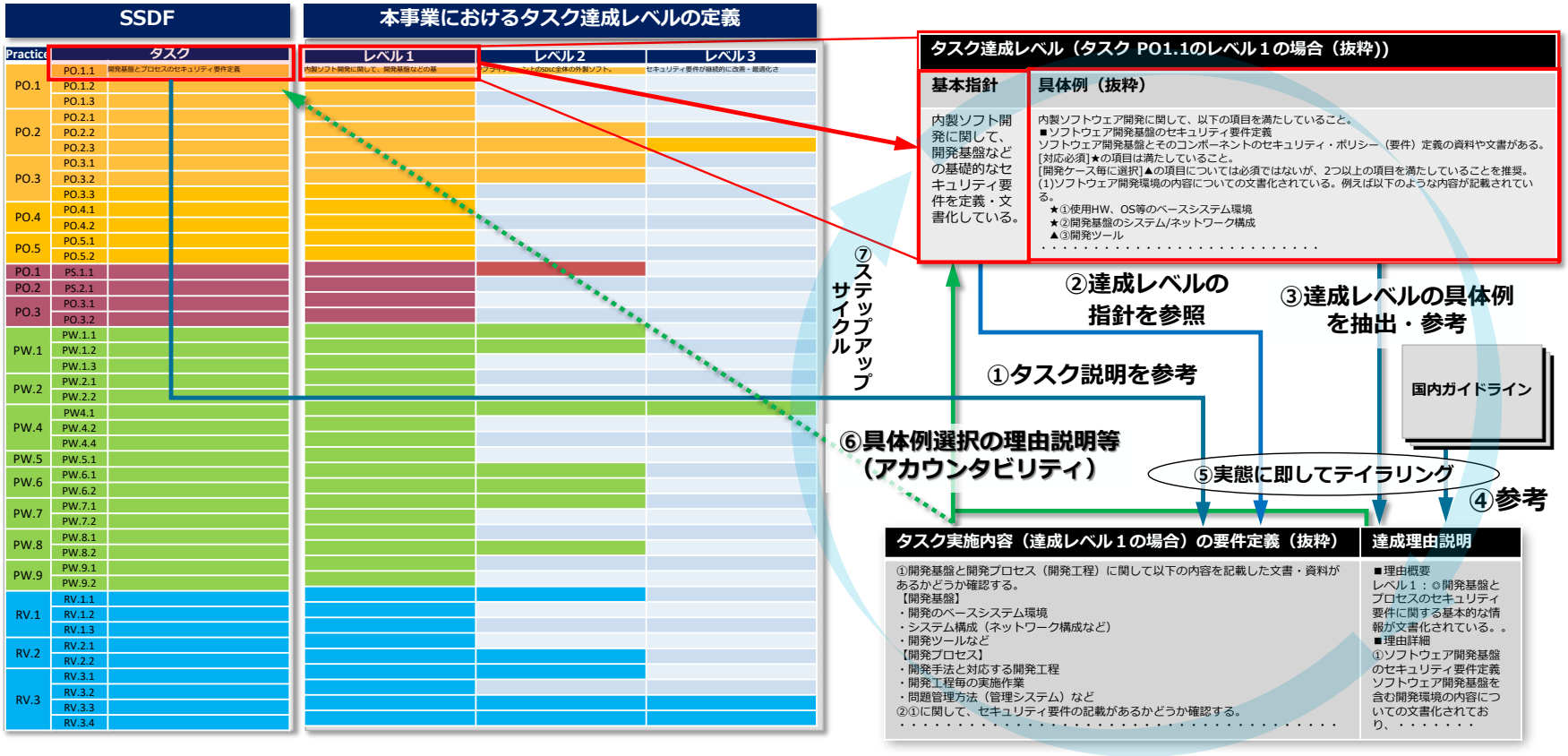


図 5-8 アカウンタビリティベース・アプローチの考え方の実施方法

アカウンタビリティベース・アプローチの実践においては、タスク達成レベルに示される定義と具体例を参考に対象、環境、個社の状況に合わせてテイラリングして適用することを前提とする。

⑤の実態に即してテイラリングにおいては、8 付録：SSDF に対するタスク達成レベルと具体例 および 9 付録：SSDF と国内ガイドラインのマッピングの読み方を参考にして、タスクの実施内容を定義する。

5.2.5. 達成度評価

タスク達成レベルとプラクティス案に基づき、タスクの実践結果を比較することにより、タスク達成レベルを評価判定し、タスク達成レベルの目標設定と乖離がある場合、妥当性の評価を行う。

タスク達成の合否判定は、以下の２通りの定義と比較することが考えられる。

- NIST SP800-218(SSDF)標準
  のタスクの記述に対して、タスクを実施した主体が定義した実施内容が、妥当（達成していること）の理由説明を行う。
- SSDF タスク達成レベル定義
  8 付録：SSDF に対するタスク達成レベルと具体例 に示されるレベルごとの定義に対応できているか判断する。レベルの選択は、調達者、供給者の要求に応じて決定するものとする。

どちらの場合も、求められる事項の定義に対して、実際に実施したことが妥当であることの理由説明を行い、（事後的であっても）確認（検証）可能な文書を残す。必要に応じて、実施した事項の入出力や活動記録などの客観的なデータや文書をエビデンスとして残す。

達成度評価全体の結果は、下図のようにタスク毎の達成度の分布として示される。

プラクティス	タスク	レベル1	レベル2	レベル3
PO.1	PO.1.1	内製ソフト開発に関して、開発基盤などの基	サプライチェーン上のSDLC全体の外製ソフト。	セキュリティ要件が継続的に改善・最適化さ
	PO.1.2			
	PO.1.3			
PO.2	PO.2.1			
	PO.2.2			
	PO.2.3			
PO.3	PO.3.1			
	PO.3.2			
	PO.3.3			
PO.4	PO.4.1			
	PO.4.2			
PO.5	PO.5.1			
	PO.5.2			
PS.1	PS.1.1			
PS.2	PS.2.1			
PS.3	PS.3.1			
	PS.3.2			
PW.1	PW.1.1			
	PW.1.2			
	PW.1.3			
PW.2	PW.2.1			
	PW.2.2			
PW.4	PW4.1			
	PW.4.2			
	PW.4.4			
PW.5	PW.5.1			
PW.6	PW.6.1			
	PW.6.2			
PW.7	PW.7.1			
	PW.7.2			
PW.8	PW.8.1			
	PW.8.2			
PW.9	PW.9.1			
	PW.9.2			
RV.1	RV.1.1			
	RV.1.2			
	RV.1.3			
RV.2	RV.2.1			
	RV.2.2			
RV.3	RV.3.1			
	RV.3.2			
	RV.3.3			
	RV.3.4			

図 5-9 SSDF タスク達成度の全体分布表示（イメージ）

また、各プラクティスやプラクティスのカテゴリごとに達成レベルの比率を求めることで、透明性を確保することができる。

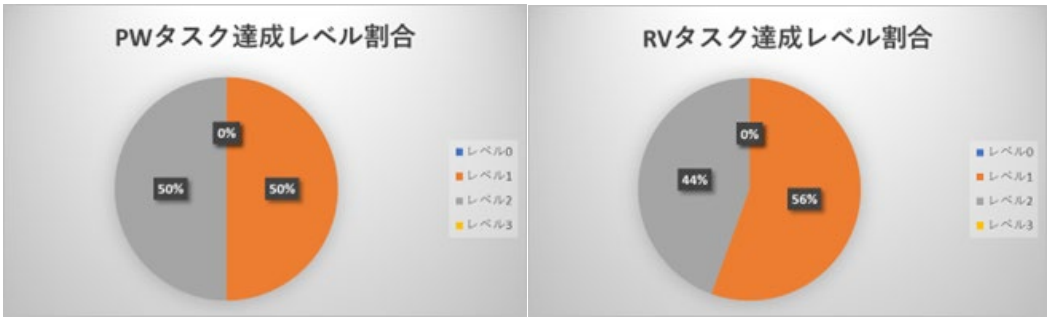


図 5-10 タスクカテゴリごとの達成レベルの比率表示（イメージ）

達成度評価は以下の 3 通りが考えられる。

- 第一者評価

  ソフトウェアの開発者、供給者など SSDF タスクを実践した主体が自ら行う。

- 第二者評価

  ソフトウェアの発注者など SSDF タスクの委託元が行う。

- 第三者評価

  ソフトウェアの調達者、供給者などソフトウェアの受発注に直接利害関係を持つものではなく、国や第三者機関が独立した立場で行う。

5.2.6. 自己適合宣言の作成

必要に応じて、政府調達において求められる SSDF 自己適合宣言書を作成する。ここでは、米国で求められる CISA 自己適合宣誓書に基づく場合を示す。将来的には、CISA 自己適合宣誓書に整合しつつ、国内で検討したタスク達成レベルに基づき、より解像度高く達成状況を可視化することも有効と考えられる。

米国においては、OMB 安全なソフトウェア開発手法の実装を通じたソフトウェアサプライチェーンの確保に関する覚書（M-22-18[7]）とその内容を更新する新たな覚書（M-23-16[8]）の規定に対応し、CISA が公開した SSDF 自己適合宣誓様式(Self-Attestation form)[9]に基づく宣誓書が求められる。日本においては、政府調達における SSDF 自己適合宣誓書についてはまだ規定はないが、QUAD

[7] OMB, ”Enhancing the Security of the Software Supply Chain through Secure Software Development Practices”
https://www.whitehouse.gov/wp-content/uploads/2022/09/M-22-18.pdf

[8] OMB, ”Update to Memorandum M-22-18, Enhancing the Security of the Software Supply Chain through Secure Software Development Practices”
https://www.whitehouse.gov/wp-content/uploads/2023/06/M-23-16-Update-to-M-22-18-Enhancing-Software-Security.pdf

[9] CISA, ”Secure Software Development Attestation Common Form”
https://www.cisa.gov/resources-tools/resources/secure-software-development-attestation-form

国際共同原則に合意しているため、同様の様式で SSDF 自己適合宣誓書が求められる可能性がある。

SSDF 自己適合宣誓書の記載項目とその詳細は次のとおりである。セクションⅢの項目は SSDF に準拠した内容となっており、いずれの要件にも適合する必要がある。

以下要件は EO 14028 のサブセクション(4)(e)に規定されており、8 付録：SSDF に対するタスク達成レベルと具体例の例示にて SSDF の項目とのマッピングを示している。

表 5-4 SSDF 自己適合宣誓書の記載項目

No	セクション	項目	詳細
1	Ⅰ	宣誓の種類	宣誓書の種類を以下から選択する。 ・新規宣誓 ・延長または免除後の宣誓 ・修正宣誓
2		適用範囲	適用範囲を以下から選択する。 ・企業全体 ・個別製品 ・複数製品/特定バージョン
3		対象ソフトウェア	宣誓書が個々の製品または複数の製品用である場合に記載する。 ・製品名 ・バージョン番号 ・リリース/公開日
4	Ⅱ	ソフトウェアの開発者情報	開発者について以下を記載する。 ・会社名 ・住所 ・会社のウェブサイト
5		担当者の情報	担当者の情報について以下を記載する。 ・名前（個人または担当部門） ・役職 ・住所（該当する場合） ・電話番号 ・メールアドレス（代表アドレス可）
6	Ⅲ	セキュア開発環境の確保	セキュアな開発環境であることを証明するために、以下項目の取組について記載する。 ・開発環境の分離 ・アクセス管理の強化（ログ監視、多要素認証の適用） ・不必要なソフトウェアの排除 ・機密データの暗号化 ・サイバーセキュリティの継続的監視と対応
7		ソフトウェアの信頼性確保	ソフトウェアの信頼性確保を行っていることを証明するために、以下項目の取組について記載する。 ・自動ツール等を利用し、内部コードとサードパーティコンポーネントのセキュリティ管理
8		内部コード及びサードパーティコンポーネントの信頼性確保	ソフトウェアの信頼性確保を行っていることを証明するために、以下項目の取組について記載する。 ・内部コードとサードパーティコンポーネントの元情報

			の管理
9		脆弱性管理と対応	脆弱性の管理と対応を証明するために、以下項目の取組について記載する。 ・自動ツールによる脆弱性スキャン ・リリース前に発見したセキュリティ問題へ対応するためのポリシーやプロセス ・脆弱性開示プログラム（VDP）を運用し、報告された脆弱性に迅速に対応
10		通知義務	ソフトウェア制作者が上記を使用しなくなった場合、本フォームを提出したすべての期間へ通知すること。
11		署名と提出	CEO または代理人の署名、もしくは FedRAMP に認定された第三者評価機関（3PAO:Third Party Assessor Organization）による評価結果を添付する。

以下は、SSDF 自己適合宣誓書の本文であるが、記載すべき箇所について上表の No と連動している。

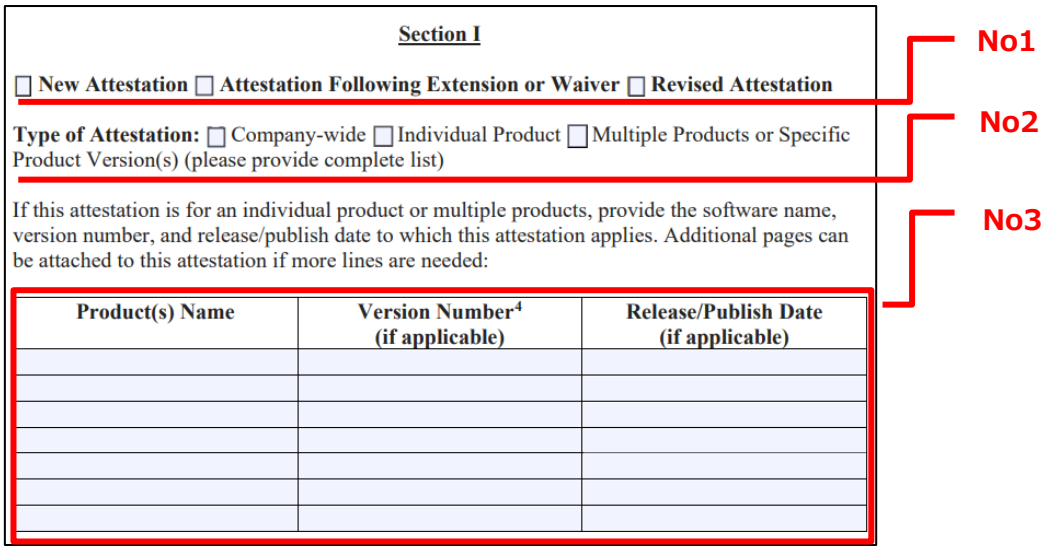
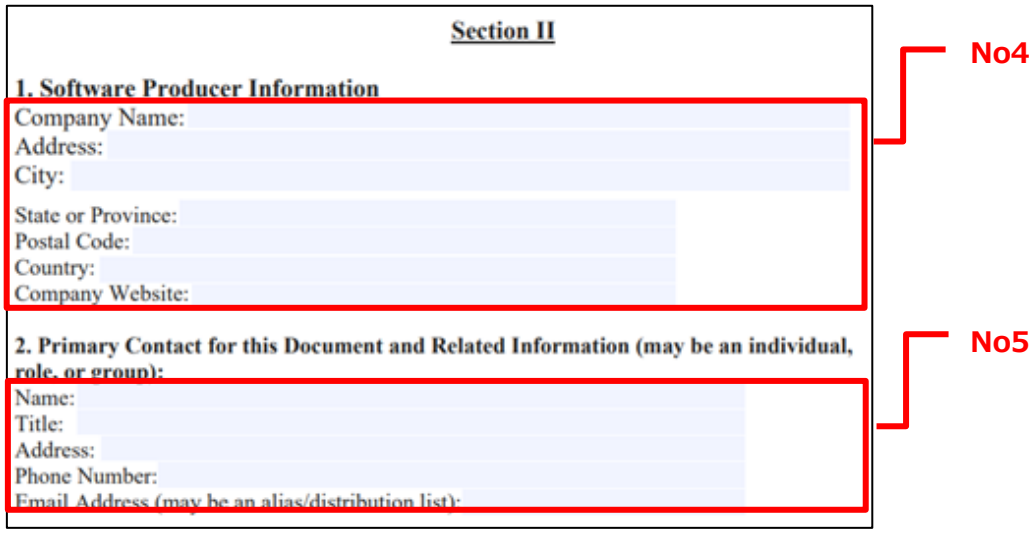


図 5-11 SSDF 自己適合宣誓書フォーム（セクションⅠ）



Section II

1. Software Producer Information

Company Name:
Address:
City:

State or Province:
Postal Code:
Country:
Company Website:

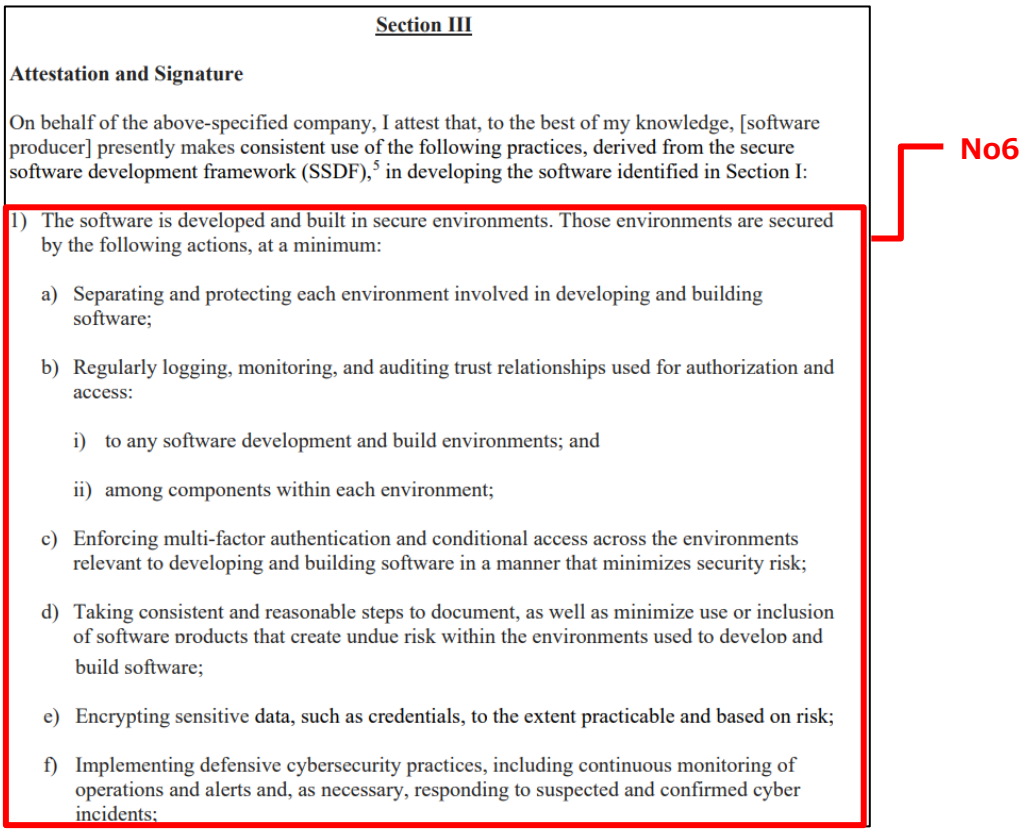
2. Primary Contact for this Document and Related Information (may be an individual, role, or group):

Name:
Title:
Address:
Phone Number:
Email Address (may be an alias/distribution list):

No4

No5

図 5-12 SSDF 自己適合宣誓書フォーム（セクションⅡ）



Section III

Attestation and Signature

On behalf of the above-specified company, I attest that, to the best of my knowledge, [software producer] presently makes consistent use of the following practices, derived from the secure software development framework (SSDF),[5] in developing the software identified in Section I:

1) The software is developed and built in secure environments. Those environments are secured by the following actions, at a minimum:

   a) Separating and protecting each environment involved in developing and building software;

   b) Regularly logging, monitoring, and auditing trust relationships used for authorization and access:

      i) to any software development and build environments; and

      ii) among components within each environment;

   c) Enforcing multi-factor authentication and conditional access across the environments relevant to developing and building software in a manner that minimizes security risk;

   d) Taking consistent and reasonable steps to document, as well as minimize use or inclusion of software products that create undue risk within the environments used to develop and build software;

   e) Encrypting sensitive data, such as credentials, to the extent practicable and based on risk;

   f) Implementing defensive cybersecurity practices, including continuous monitoring of operations and alerts and, as necessary, responding to suspected and confirmed cyber incidents;

No6

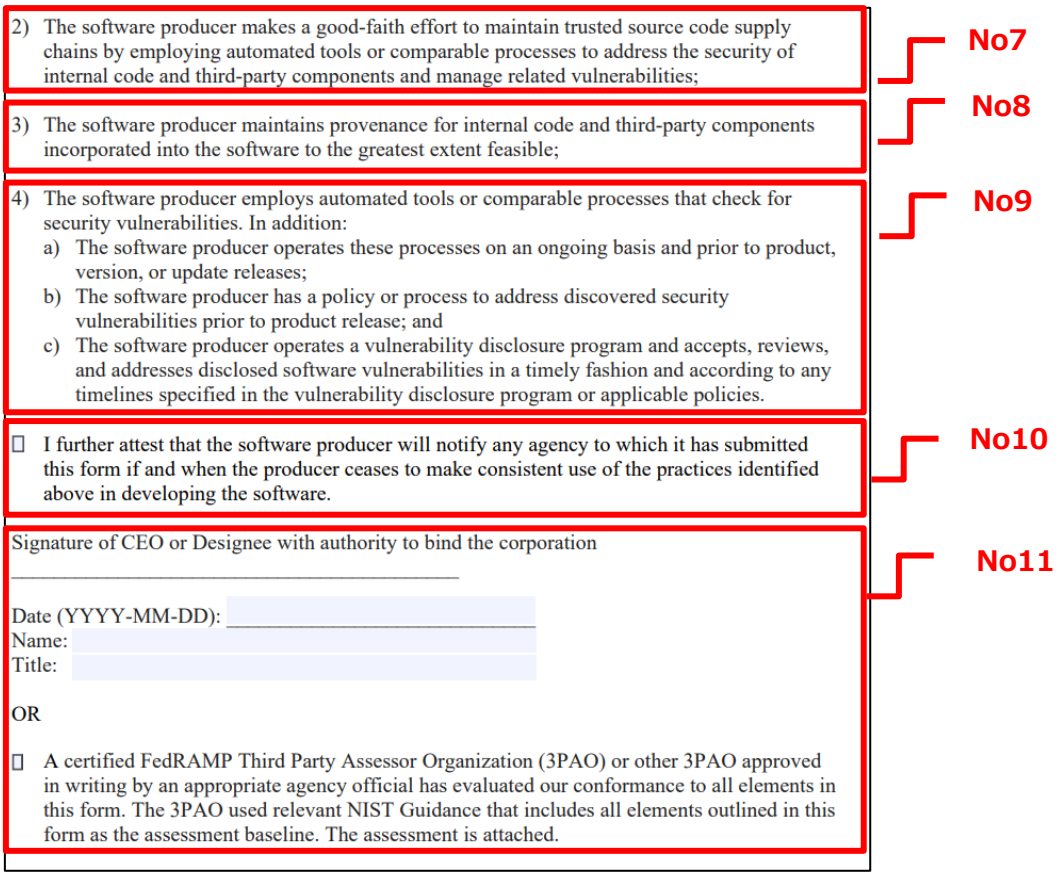


図 5-13 SSDF 自己適合宣誓書フォーム（セクションⅢ）

以下に該当する場合、本フォームの対象外となる。

- 連邦政府機関が開発したソフトウェア
- 連邦政府が直接入手したオープンソースソフトウェア
- ソフトウェア製品に組み込まれたサードパーティ製オープンソース / プロプライエタリコンポーネント
- 公開され、自由に入手できるソフトウェア

セクションⅢの要件が適合できていない場合には、実施に向けた**行動計画書（POA&M）**を作成することで、それが受け入れられれば条件を満たす可能性がある。適合できていない要件を明確に示し、リスク軽減措置として代替策や暫定的なセキュリティ対策について、行動計画とマイルストーンの記載を行う。なお、行動計画の実施期限や要件の適合期限について、本文書では明確に指定されていない。

5.2.7. ステップアップサイクルの推進

5.2.3 タスク達成レベルの定義とギャップ分析ステップに戻り、段階的な目標設定選択し、5.2.4、5.2.5 のサイクルを回し、最終的な目標レベルを到達する。

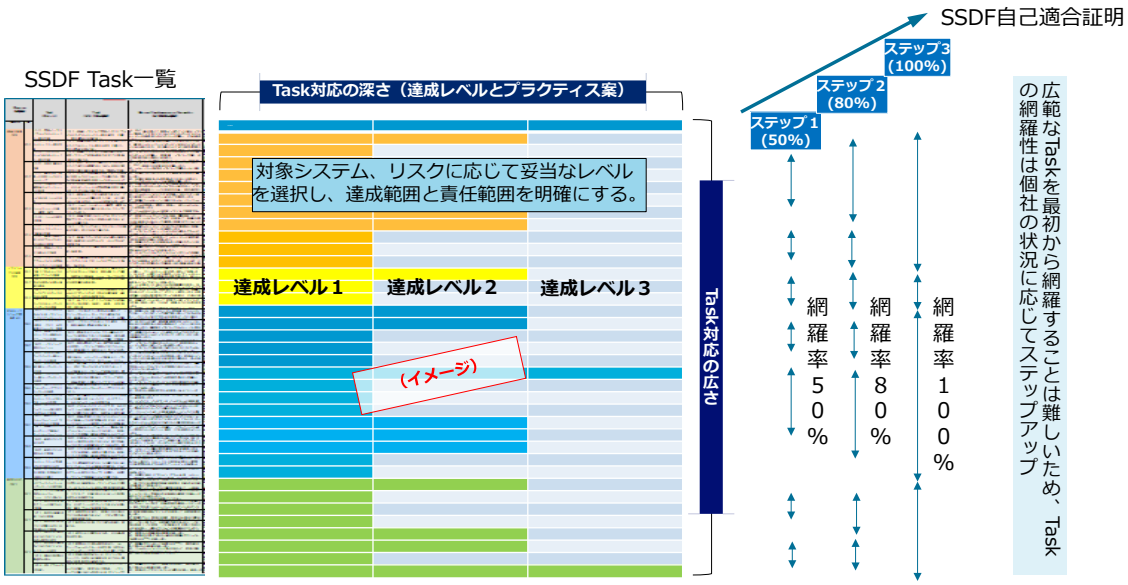


図 5-14 SSDF タスクレベルのステップアップサイクル

6. 参考文献

[1] NIST SP 800-218 Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities

[2] CISA, Secure-by-Design Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software[10]

[3] QUAD The Wilmington Declaration, Joint Statement from the Leaders of Australia, India, Japan, and the United States

[4] METI, ソフトウェア管理に向けた SBOMSBOM(Software Bill of Materials Materials)の導入に関する手引 ver.2.0, 2024, Guidance on Introduction of Software Bill of Materials (SBOM) for Software Management ver.2.0 (English)[11]

[5] CISA, Framing Software Component Transparency: Establishing a Common Software Bill of Materials (SBOM) September 3, 2024

[6] The Linux Foundation and its Contributors (OMG etc.). The System Package Data Exchange® (SPDX®) Specification Version 3.0.1

[7] OWASP, The CycloneDX specification, Version 1.6

[10] https://www.cisa.gov/sites/default/files/2023-10/Shifting-the-Balance-of-Cybersecurity-Risk-Principles-and-Approaches-for-Secure-by-Design-Software.pdf

[11] https://www.meti.go.jp/english/policy/safety_security/cybersecurity/downloadfiles/sbom_tebiki02_en.pdf

7. 付録：用語、定義及び略称

用語	出　所[12]	定義	略称
Practice（プラクティス・実践策）	[1]	セキュアなソフトウェアを実現する上で達成すべき状態と活動を抽象的に示すもので、ツール、技術、仕組みを規定するものではない。タスクの集合として規定される。	—
Task (タスク・対策・課題)	[1]	プラクティスを実現するための対策。	—
Notional Implementation Examples (概念的実装例)：	[1]	タスクを実現するためのツール、プロセス、方法などの概念的な例。いずれも必須ではなく、これらが選択肢の全てではない。組織や状況によっては、適用できない選択肢もある。	—
Task References（タスク参考文献）	[1]	確立されたセキュア開発プラクティス文書へのポインタと特定のタスクへのマッピングである。全ての参考文献が、ソフトウェア開発の全てのインスタンスに適用されるわけではない。	—

[12] 9.参考文献に記載の文献番号

8. 付録：SSDF に対するタスク達成レベルと具体例

別冊 Excel ファイル「SSDF 導入ガイダンスタスク整理シート.xlsx」のシート「SSDF タスク整理全体」の列「タスク達成レベルとプラクティス例」(OPQ 列)に示す。

1) SSDF に対するタスク達成レベルと具体例の読み方

(ア) タスク達成レベルの判断指針

SSDF タスクは、ソフトウェアの用途や利用環境のリスクに応じて実施すべき事項には大きな幅があることが想定される。例えば、「PO.1.1：組織のソフトウェア開発基盤とプロセスのすべてのセキュリティ要件を特定、文書化し、要件を最新の状態に保つ。」については、対象システムや組織によってプロセスの粒度や文書化の粒度や項目の件数は大きく異なっており、どこまで実施するかについては、目的と要求レベルに依存し、それに伴い効果やセキュリティレベルは大きく異なる。したがって、その達成レベルを分解し、具体的にどのようなプラクティスを実施しているかについて可視化しなければ、対象とするソフトウェアに対するセキュリティ要求レベルと実現されるセキュリティレベルに相違が生じる危険性がある。そのため、本書では、全てのタスクを 3 段階のレベルに分解し、調達者と供給者の間での認識に大きな相違が生じることを解消するための方法を提供する。レベルの数は、本質的な差が明確に区別できるように、3 段階に抑え、レベルを区別する本質的な違いを判断指針として明示した。また、実践するためには、具体的な実施項目が分かることが重要であるため、各レベルの具体例を例示するようにした。それらについて次節以降でその考え方と読み方を示す。

SSDF 達成レベルの本質的な違いが明確となるように、レベルは 3 段階に抑えて、その本質的な違いを判断指針として定義した。レベル定義の考え方として CMMI のプロセス成熟度モデルの考え方も参考にした。CMMI は 5 段階でレベルを区別しているが、明確性を重視し、より簡潔な 3 段階とした。CMMI とのレベルの対応関係は以下の通りである。

表 8-1 CMMI 成熟度レベルと本書におけるタスク達成レベルの関係

タスク達成レベル					想定産業分野等（高信頼性タイプ分類）	参考：CMMI	
レベル	目安	網羅性	再現性	最適化		成熟度レベル	定義概要
0	最低限の事項が未達。レベル 1 に未達。	網羅できていない	属人的	未最適	ー	レベル 1：初期レベル	プロセスが定義されておらず、場当たり的である状態。そのため、プロジェクトの成功は担当者の努力に依存する。
1	最低限の事項が実施されているが、部分的、属人的	最低限の項目対応	属人的	未最適	（Type I分野） 社会的影響が殆どない・限定的。一般コンシューマー製品等。	レベル 2：管理されたレベル	プロセスの基本が確立されている状態。基本的な手順が決まり、再現可能な状態。
2	主な項目が網羅され、標準化されているが、プロセス改善に基づく最適化は未実施	網羅的	標準化（体系化）	未最適	（Type II分野） 社会的・波及的な影響がある。特定のインフラ産業、高シェア大手ベンダー等	レベル 3：定義されたレベル	各プロセスが標準化されている状態。標準プロセスのマニュアル化やスタッフの熟練度の向上が進み、属人化解消や生産性向上も期待できる。プロセスにおける技術的な側面も標準化が進み、レベル 2 よりもスムーズに作業を進められる状態。
3	高度なレベルで網羅され、標準化され、かつ改善プロセスに基づき最適化を達成	網羅的かつ高度	標準化	最適化・改善プロセス確立	（Type III分野） 社会的影響が極めて大きい。基盤的重要インフラ、電力、金融、通信分野等。	レベル 4：定量的に管理されたレベル	レベル 4 はプロセスが明確な数値や数量で管理されている状態。作業の手順や進捗状況、成果物の品質を数値化することで、より客観的な管理を実現できる。
					（Type IV分野） 人命に影響、甚大な経済的損失。医療機器、自動車、航空機、原子力、石油化学プラント	レベル 5：最適化しているレベル	プロセスが最適化している状態。継続的なプロセス改善、トラブルの予測と対策、新しい提案や技術の費用対効果分析、組織全体での情報共有ができている。

（出所）三菱総合研究所作成

具体的なタスク達成レベルの例を示す。ここでは、組織の準備(PO)のタスク PO.1.1：開発基盤とプロセスのセキュリティ要件定義の例を示している。表に示す通り、レベルを分ける本質的な違いとしては、対象とするソフトウェアのうち、レベル 1 では、内製ソフトを対象としてセキュリティ要件定義を行うのに対し、レベル 2 では、サプライチェーンにおける SDLC 全体の外製ソフト（OSS やサードパーティ等の部品）までセキュリティ要件定義を行うことが本質的な違いとなる。さらに、レベル 3 では、セキュリティ要件定義が、継続的に改善・最適化される仕組みが確立していることを挙げ、プロセス成熟度モデルで最も高度なレベルに求められる条件を採用している。

レベル	PO1.1 タスクのレベル判断指針
1	**内製ソフト**開発に関して、開発基盤などの基礎的なセキュリティ要件を定義・文書化している。（内製ソフトの開発基盤の最低限のセキュリティ要件定義）
2	1.に加え、サプライチェーンにおける **SDLC 全体の外製ソフト（OSS やサードパーティ等の部品）**も含めた開発基盤と開発プロセスのセキュリティ要件全般が体系的に定義・文書化されている。（外製ソフトを含むソフト全体の開発基盤のセキュリティ要件の十分な定義）
3	2.についてセキュリティ要件が**継続的に改善・最適化される仕組みが確立**している。（セキュリティ要件が最適化されている）

このような考え方で、全てのタスクについてタスク全体を大きく 3 段階に分ける上で最も重要な観点を特定し、その違いが可能な限り明確になるように定義した。

(イ)　タスク達成レベルのプラクティス案

前節のレベル判断指針の考え方に準拠して、具体的に各レベルの実施項目や対象事例を示すことで、現場において実施する事項を定義するための参考情報を提示することを目的に整理した。その具体例を以下に示す。ここでもレベル判断指針と同じ PO.1.1：開発基盤とプロセスのセキュリティ要件定義の例を示す。

この例では、レベル 1 においては、内製ソフトについて、ソフトウェア開発基盤と開発プロセスについて、それぞれ対象とする項目、文書化する項目の具体例を示している。また、★、▲印により必須項目、任意項目を区別して示している。レベル 2 においては、外製ソフトに対しても同様の項目に関する要件定義を求めており、また、内製ソフトについては任意項目 12 件のうち、10 件以上を満たすなど要件を強化している。レベル 3 においては、見直しプロセスの定義やその実施履歴について要件化することで、最適化の仕組みが確立されているか判断でいるようにしている。

表 8-2 タスクレベルの具体例（タスク PO.1.1 の場合）

レベル	PO1.1 タスクのレベル具体例
1	内製ソフトウェア開発に関して、以下の項目を満たしていること。 ■ソフトウェア開発基盤のセキュリティ要件定義 ソフトウェア開発基盤とそのコンポーネントのセキュリティ・ポリシー（要件）定義の資料や文書がある。 [対応必須]★の項目は満たしていること。 [開発ケース毎に選択]▲の項目については必須ではないが、2 つ以上の項目を満たしていることを推奨。 ※以下の例示で同等の対策を実施している場合は、達成度としてカウント可能。その場合、同等性の説明が必要。 (1)ソフトウェア開発環境の内容についての文書化されている。例えば以下のような内容が記載されている。 ★①使用 HW、OS 等のベースシステム環境 ★②開発基盤のシステム/ネットワーク構成 ▲③開発ツール ▲④テスト環境 ▲⑤使用する管理システム ・ 構成管理システム（ソース管理、文書管理など） ・ 問題管理システム（バグ管理、インシデント管理など） ▲⑥使用する情報共有・伝達手段（メール、WEB、チャットなど） (2)上記に対するセキュリティ要件/ポリシーが定義されている 例えば以下のような内容が記載されている。 ★①システムのデフォルト設定をセキュアな設定に倒す ★②システムのアクセスコントロール方針 ▲③認証の強制 ▲④ネットワーク不正侵入回避のための対策 ▲⑤テスト環境の分離による保護 ▲⑥セキュリティアラートログの監視による保護 ■ソフトウェア開発プロセスのセキュリティ要件定義 ソフトウェア開発プロセスとそのセキュリティ確保のポリシー定義（セキュリティ要件）の資料や文書がある (1)ソフトウェア開発プロセスが定義されている。 例えば以下のような内容が記載されている。 ★①開発手法（ウォータフォール，アジャイル開発など） ★②開発工程プロセス（設計、実装、テストなど） ・開発工程毎のインプット/アウトプット -設計書 -ソース -テスト仕様書・報告書

	-レビュー報告書等 ・次工程移行判定基準 ・製品リリース判定基準 ・構成管理（ソース管理、リリース管理、文書管理等） ・各工程での障害・問題管理（バグ管理、インシデント管理、脆弱性管理等） ・製品リリース後の運用・管理（サポート体制、アップデート計画等） (2)上記に対するセキュリティ要件/ポリシーが定義されている 例えば以下のような内容が記載されている。 ★①設計時に開発機能のセキュリティ要件を明確化し、設計に盛り込むこと ★②設計時のセキュリティ要件が実装されているかチェックすること ★③各工程における生産物を保存し、完全性を担保する。 ★④各工程における問題点や課題について記録し、完全性を担保する。 ▲⑤教育計画を立てる。 ▲⑥各工程でのレビューは第三者によるレビューを実施し、その履歴を管理すること。 ▲⑦開発工程にセキュリティ品質のチェックポイントを置き、品質評価を行う。 ▲⑧品質評価に基づいて次工程への移行、リリース判定を行う
2	レベル 1 に加え、外製ソフトウェア（OSS やサードパーティ製ソフトウェア）も含めたセキュリティ要件が定義され、上記項目を満たしていること。内製ソフトについては任意項目 12 件のうち、10 件以上を満たす。
3	レベル 2 に加えて、以下の項目を満たしていること。任意項目は 2 件以上満たす。 ■ソフトウェア開発基盤や開発プロセスのセキュリティ確保のポリシー定義（セキュリティ要件）の見直しプロセスが定義された資料や文書がある。 (1)ソフトウェア開発基盤や開発プロセスのセキュリティ要件の見直しプロセスが定義されている。 例えば以下のような内容が記載されている。 ★①定期的にソフトウェア開発基盤のセキュリティ要件を見直す。 ・見直すトリガやタイミング ・見直しに伴う対応など ★②定期的にソフトウェア開発プロセスのセキュリティ要件を見直す。 ・見直すトリガやタイミング ・見直しに伴う対応など (2)見直し実施履歴がある。 例えば以下のような資料や情報がある。 ▲①ソフトウェア開発基盤のセキュリティ要件の見直し履歴 ▲②ソフトウェア開発プロセスのセキュリティ要件の見直し履歴 (3)見直しに伴う教育の実施履歴がある。 例えば以下のような資料や情報がある。 ▲①ソフトウェア開発基盤のセキュリティ要件の見直しに対応した教育計画と実施履歴 ▲②ソフトウェア開発プロセスのセキュリティ要件の見直しに対応した教育計画と実施履

	歴

このような考え方で、全てのタスクについて 3 段階のレベルについて具体的に実施する事項を豊富に挙げることで、選択肢やカスタマイズの材料を提供している。

SSDF 実施者は、タスクについてこのような判断指針と具体例を参考に、必要に応じて取捨選択、カスタマイズを行うことで、タスクの具体的な実施項目を定義することができる。

9. 付録：SSDF と国内ガイドラインのマッピング

別冊 Excel ファイル「SSDF 導入ガイダンスタスク整理シート.xlsx」のシート「SSDF タスク整理全体」のマッピング列（Y 列～AS 列)に示す。

1) SSDF と国内ガイドラインのマッピングの読み方

ソフトウェア・セキュリティに係る国内ガイドラインが多数存在する中で、QUAD 等の国際的な標準のベースとなりうる SSDF への対応においては、すでに対応済みの国内ガイドラインや業界にカスタマイズした国内ガイドラインを活用した SSDF への対応が効果的と考えられる。また、国際整合化の観点から、国内ガイドラインは、SSDF との関係性や整合性を意識した改訂が期待される。このようなことから、国内ガイドラインと SSDF のタスクレベルの項目間の対応関係を整理することで、各社のセキュリティ対策、国・業界等のガイドラインの改訂への参考情報を提供することを目的とする。

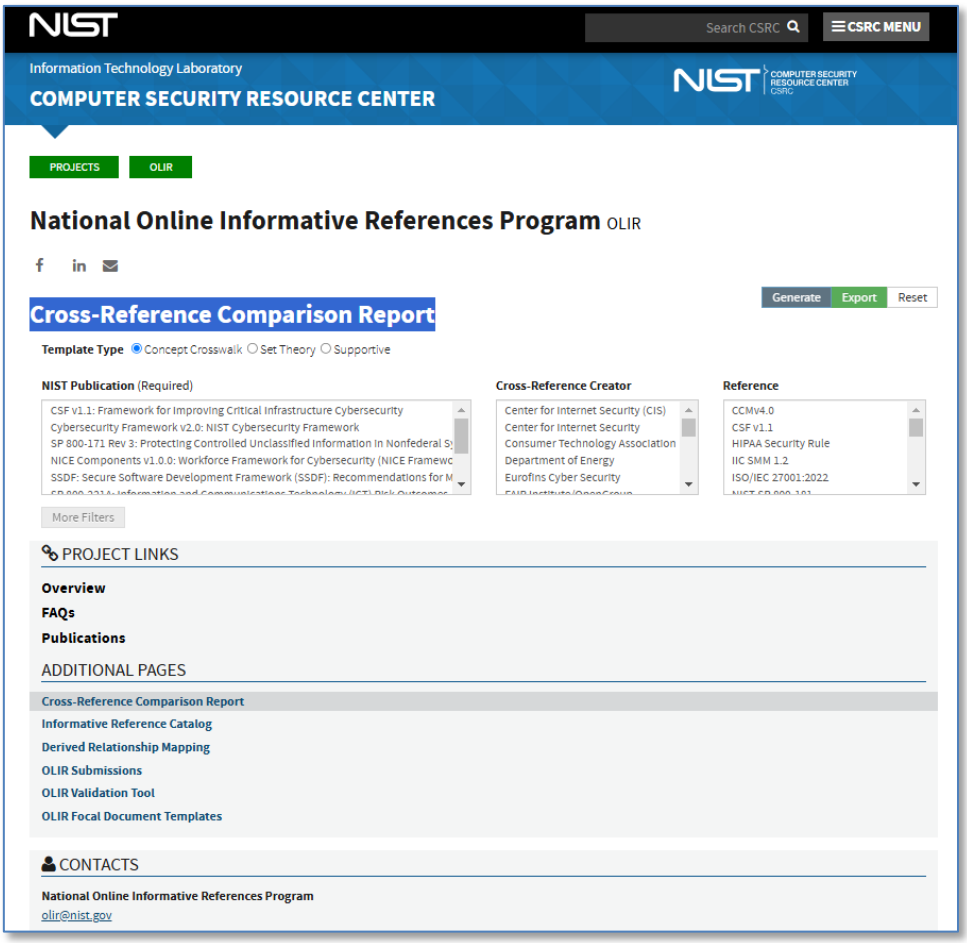


図 9-1　NIST 関連標準間の相互参照データベース（Web ページ）[13]

[13] NIST Cross-Reference Comparison Report においても、サイバーセキュリティに関する標準やガイドラインの章項目の対応関係を示すデータベースが公開されている。https://csrc.nist.gov/projects/olir/coverage-report#/olir/coverage-report

(ア)　マッピングの方針

SSDF タスクと国内ガイドラインの要素項目については、粒度や観点が多様であるため厳格な対応付けを行うことは困難である。ただし、関連性のある項目を対応付けたり、大まかな包含関係を示すことは、国内ガイドラインに対応済みの組織にとって、SSDF に対応する上で俯瞰的な理解を高めたり、効率化を図ることで有効である[14]。SSDF と国内ガイドラインのマッピングは、それらを策定した主体自身が実施することが好ましく、正確性が高まると考えられる。また、SSDF や国内ガイドラインの改訂を考慮すると、第三者が集中的にメンテナンスするのではなく、ガイドライン策定者が、ガイドラインの改訂に応じて自律分散的に行うことが現実的と考えられる。

本書では、その先鞭として、経済産業省事業を通じて、SSDF と国内ガイドラインのマッピングの見本を示すものである。SSDF との関連性や注目度の観点から国内ガイドラインの一部を選択し、それらについて SSDF タスク項目との対応関係を整理した。これらの国内外ガイドラインやその他の国内ガイドラインについて、それぞれのガイドライン策定主体が、必要性に応じて、マッピングの作成やガイドラインの改訂に応じてマッピングの改訂を行うことが期待される。

対象とするガイドラインは以下の通りである。

表 9-1 SSDF と国内ガイドライン等とのマッピング対象一覧

発行者	ガイドライン名
METI	SBOM 導入手引き　Ver.1.0,　Ver.2.0
Software-ISAC	情報システムにおけるセキュリティコントロールガイドライン
デジタル庁	政府情報システムにおけるセキュリティ・バイ・デザインガイドライン
METI	サイバーセキュリティ経営ガイドライン
NISC	政府機関等のサイバーセキュリティ対策のための統一基準群
METI	サイバー・フィジカル・セキュリティ対策フレームワーク（CPSF）
MIC	クラウドサービス提供における情報セキュリティ対策ガイドライン

[14] NIST Cross-Reference Comparison Report では、章項目間の対応関係を大まかに示すだけで、対応関係の根拠説明や包含関係までは示していない。https://csrc.nist.gov/projects/olir/coverage-report#/olir/coverage-report

う

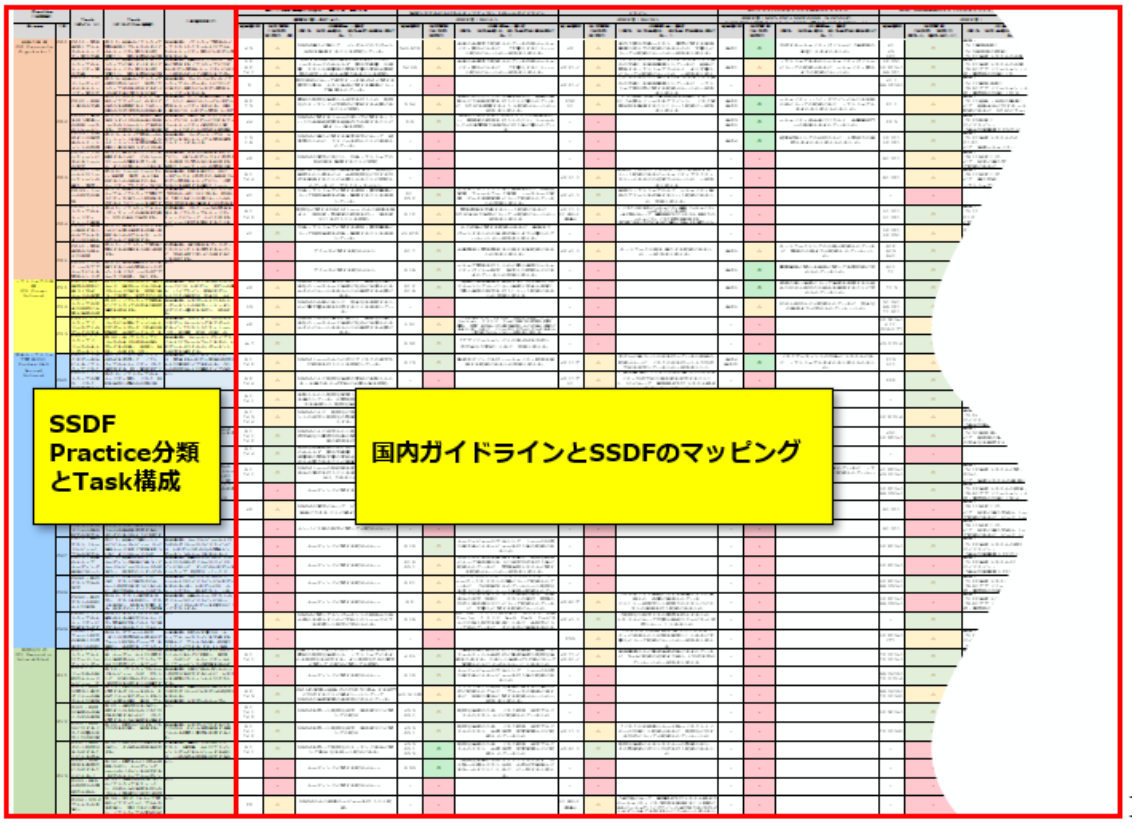
国内ガイドラインに対応している企業が、SSDF への対応を検討する上で、どの程度対応できているか対応状況を大まかに判断するための参考情報として SSDF タスクと国内ガイドラインの項目の対応関係を以下のような包含関係で示し、その判断理由も示す。

表 9-2 SSDF タスクと国内ガイドラインの項目の対応関係の区分

◎:包含	ガイドライン該当章は、SSDF taskを含みより広い
○:同等	ガイドライン該当章は、SSDF taskと同等の範囲
△:部分	ガイドライン該当章は、SSDF Taskの部分のみ
×:該当なし	SSDF Taskに該当する章は含まれない

(イ) マッピング表の読み方と具体例

前節に従い SSDF と国内ガイドラインのマッピング結果を示したものが 9 付録：SSDF と国内ガイドラインのマッピングである。マッピングの全体像（イメージ）は以下の通りである。



凡例

◎:包含	ガイドライン該当章は、SSDF taskを含みより広い
○:同等	ガイドライン該当章は、SSDF taskと同等の範囲
△:部分	ガイドライン該当章は、SSDF Taskの部分のみ
×:該当なし	SSDF Taskに該当する章は含まれない

図 9-2 SSDF と国内ガイドラインのマッピング全体像（イメージ）

マッピングの例（抜粋）政府統一基準

SSDF	NISC 政府統一基準		
Task	対応項目	包含関係	包含関係判断理由
PW.1.1：リスクモデリングの形式を使用して、ソフトウェアのセキュリティリスクを評価する（例：脅威モデリング、攻撃モデリング、攻撃面のマッピング）	2.1.3 5.1.1	△	「5.1.1 情報システムの分類基準等の整備」にて、リスクの評価を行い、対応方針を決定する旨の記載があるが、**リスク評価の詳細な記載はない**。また、 「2.1.3 情報セキュリティ関係規程の整備」ガイドライン： 「（解説）遵守事項 2.1.3(1)(a)」 にて、リスク評価の方法が記載されているが、**システム構築時のリスク評価ではない**ため、一部包含と判断した。
PS.1.1：ソースコード、実行コード、構成コードなどの全てのコード形式を、原則に基づいて保管し、承認された担当者、ツール、サービスなどからのみアクセスできるようにする。	7.1.3	○	「7.1.3 権限の管理」 にて、**権限の最小化に基づいた情報の保護**について記載があるため。 ※ソースコード等を管理するシステムを情報システムの一種と考える。

マッピングの例（抜粋）METI　CPSF

SSDF	METI, サイバー・フィジカル・セキュリティ対策フレームワーク（CPSF）		
Task	対応項目	包含関係	包含関係判断理由
PS.3.1：各ソフトウェアリリースに必要なファイルおよびサポートデータ（完全性検証情報、由来データなど）を安全に保管する。	CPS.IP-4	△	CPSFの「**構成要素**」**の一部としてソフトウェアも含まれる**と考えられるため。 根拠文書：NIST-CSFとSSDFの対応（NIST Cross-Reference Comparison Report）、NIST-CSFとCPSFの対応(METI CPSF Ver.1.0 添付D)
PS.3.2：各ソフトウェアリリースの全ての部品由来のデータを収集し、保護し、維持し、共有する（例：SBOM）	×	×	根拠文書：NIST-CSFとSSDFの対応（NIST Cross-Reference Comparison Report）、NIST-CSFとCPSFの対応(METI CPSF Ver.1.0 添付D)

10. 付録：SSDF 実証内容

別冊 Excel ファイル「SSDF 導入ガイダンスタスク整理シート.xlsx」のシート「SSDF タスク整理全体」の列「実証評価」(RSTUVWX 列)に示す。

11.　付録：SSDF 導入の実証例（参考資料）

別冊 pdf ファイル「11.付録　SSDF 導入の実証例（参考資料）.pdf」に示す。