サイバーインフラ事業者に求められる役割等に関する

ガイドライン（案）

― ソフトウェアの開発、供給、運用におけるサイバーセキュリティ確保とレジリエンス向上のための

顧客とサイバーインフラ事業者の適切な役割分担と責務の在り方について―

令和７年 10 月

経済産業省産業サイバーセキュリティ研究会ＷＧ１・

内閣官房国家サイバー統括室合同ワーキンググループ

サイバーインフラ事業者に求められる役割等の検討会

目次

1. 総論

1.1. 背景と目的

(1) サイバーセキュリティ[1]に係るレジリエンス向上の必要性

近年、サイバー攻撃の起点は多様化し、社会活動の基盤を担うソフトウェアとそのサプライチェーン上の潜在的なあらゆる脆弱性を狙うサイバー攻撃が相次いでいる。デジタル社会の活動は様々な情報・通信システムやサービスを構成するあらゆるソフトウェアに深く依存しているため、サイバー攻撃は、重要インフラを含め国民生活・経済活動に甚大な被害をもたらしデジタル社会の信頼性を損なうことになる。表 1 に代表的な事例を示す。

表 1 サイバー攻撃の代表的な事例

事例	概要
Apache Log4J の脆弱性	Apache Log4J は、ログを出力するソフトウェアライブラリであり、世界中で利用されている。2021 年に遠隔から任意の処理を実行できるという深刻な脆弱性が発見され、悪用された事例である。 多層のソフトウェアサプライチェーンの中で、様々なソフトウェアに組み込まれ利用されていることから、脆弱性を発見、追跡、改修するという管理の重要性を示唆している。
ソフトウェアベンダー A 社のソフトウェアアップデートの改ざん	正規のソフトウェアアップデートが改ざんされたことで、本ソフトウェアの利用組織全体に影響が及んだ事例である。 ソフトウェアベンダーに侵入されたことで、正規のソフトウェアアップデートが改ざんされたものであり、ソフトウェアサプライチェーンの開発・運用環境のセキュリティ確保の重要性を示唆している。
B 病院が保有する患者情報の暗号化・漏えい	脆弱性が改修されないままの VPN 装置を介して、院内ネットワークへ侵入されたことで、診療業務に支障がでた事例である。 ソフトウェアセキュリティに関する病院（顧客）の主体的な管理と、事業者による情報提供の重要性を示唆している。

このように、あまたのソフトウェアのサイバー攻撃の要因となるものは、システムやサービスの設計を含めた

[1] サイバーセキュリティとは、電磁的な方式による情報の漏えい、滅失、毀損を防ぎ、かつそれらの情報を扱うシステムやネットワークの安全性と信頼性を確保するために講じられる措置、及びその維持管理のことをいう。サイバーセキュリティ基本法第 2 条を参照。

開発フェーズ、構築や保守・運用フェーズ、利用ユーザとソフトウェア開発者や供給者間のサプライチェーンとその契約フェーズなど多種多様なところに広く潜在しているため、全ての対策を網羅するのは容易なことではない。これらのリスク要因を適切に対処するためには、サイバーインフラ事業者に係る官民が連携した取組や、コストとのバランス、及びサイバーセキュリティリスクを考慮したリスクマネジメントが求められている。この点、米国においては、近年、ソフトウェア開発やソフトウェアサプライチェーンのセキュリティ強化に向けた基準やガイドラインが作成された。また、EU においてはサイバーレジリエンス法が 2024 年に発効、2027 年に全面的に施行予定であるなど、デジタル製品・サービスにおけるサイバーセキュリティ対策の強化に関する制度整備が加速している。また、セキュアバイデザインの概念が国際的に支持を集める[2]など、企業は自社をサイバー攻撃から守ることのみならず、自社が提供するソフトウェア製品・サービスのサイバーセキュリティ対策についても問われる時代になりつつある。

一方、我が国のサイバーセキュリティ基本法第 7 条においてはサイバー関連事業者・情報システム等供給者[3]等の責務が規定されており、特に情報システム等供給者は、情報システム等の利用者によるサイバーセキュリティ確保に必要な支援を行う努力義務を負うこととされているところ、これらの事業者のうちソフトウェアの設計を含めた開発、供給、運用において一定の社会インフラの機能を提供している事業者（以下、「サイバーインフラ事業者[4]」という。）が、ソフトウェア製品・サービスのサイバーセキュリティ対策について、開発、供給、運用の各フェーズで果たすべき役割等を整理したドキュメントは存在しない状況にある。

本ガイドライン（案）はサイバーインフラ事業者に求められる役割等について整理・解説することにより、これら事業者によるレジリエンスの向上、及びサイバーセキュリティの根本的確保を促進することを目的とするものである。

[2] 2023 年 10 月、日米含む 13 か国の政府機関等が、設計段階から IT 製品（特にソフトウェア）の安全性を確保する際の推奨事項をまとめたガイダンスに共同署名している。これらのガイダンスは米国 CISA のサイトから公開している（https://www.cisa.gov/securebydesign）。

[3] サイバー関連事業者とは、インターネットその他の高度情報通信ネットワークの整備、情報通信技術の活用又はサイバーセキュリティに関する事業を行う者である。情報システム等供給者とは、情報システム若しくはその一部を構成する電子計算機若しくはプログラム、情報通信ネットワーク又は電磁的記録媒体の供給者である。

[4] サイバーインフラ事業者とは、サイバーセキュリティ基本法において、サイバー関連事業者（インターネットその他の高度情報通信ネットワークの整備、情報通信技術の活用又はサイバーセキュリティに関する事業を行う者）等の責務が規定されている事業者のうち、政府機関等及び重要インフラ事業者を始め広く社会で活用される情報・通信システム、ソフトウェア製品及び ICT サービスを開発し提供する事業者並びに当該情報・通信システム等のソフトウェアのライフサイクルとサプライチェーンに関わる事業者をいう。

1.2. ガイドライン（案）の位置付け

(1) 整備体系

本ガイドライン（案）は、顧客（政府機関等及び重要インフラ事業者[5]等を含む）に IT/OT システム、ソフトウェア製品又は ICT サービスを提供するサイバーインフラ事業者とそのサプライチェーンにおいて、ソフトウェアを対象とした効果的なサプライチェーン上のサイバーセキュリティ対策を進めるため、事業者と顧客との間での適切な役割分担の下で、サイバーインフラ事業者及び顧客に求められる責務（基本理念に類する事項）を示すものである。また、サイバーセキュリティに関わるリスクを把握・評価した上で、適切なリスク対応の実践を通じて残留リスクを許容範囲以下に抑制するリスクマネジメントにおいて不可欠となる体系的な対応策を、責務を果たすための要求事項として整理した。これらの事業者及び顧客に求められる責務、及び責務を果たすための要求事項に基づき、事業者と顧客が互いの役割を認識し、正確な情報を共有してセキュリティを確保する対策を共に講じることで、サイバー攻撃への対応力の強化につながることが期待される。

なお、諸外国の例では、ソフトウェアサプライチェーンのセキュリティ対策として、技術的な取組のみならず、直接的に事業者に規律を課すことも行われているところではあるが、現在、国内にはソフトウェアサプライチェーンに関わるサイバー関連事業者等を直接規律する法律がない中で、本ガイドライン（案）は、事業者及び関係者がサイバーセキュリティ対策の実効性を確保するために参考となる考え方を示すものである。[6]

(2) 利用方法

本ガイドライン（案）は、サイバーインフラ事業者、及び顧客による利用を想定している。利用の際には、対象とするソフトウェアの特性、ソフトウェアの利用に係る契約形態などに基づいて、ソフトウェアライフサイクル上の開発・提供・運用における各々の役割区分を定める。サイバーインフラ事業者は、必要に応じて顧客との合意形成を図り、自らに求められる責務の範囲を認識する。

- サイバーインフラ事業者

  本ガイドライン（案）の要求事項をチェック項目として、自組織及びソフトウェアサプライチェーンに関連する事業者の取組の過不足を確認することで、ソフトウェアサプライチェーンのセキュリティ対

[5] 重要インフラ事業者とは、サイバーセキュリティ基本法第 3 条第 1 項に規定する重要社会基盤事業者をいう。国民生活及び経済活動の基盤であって、その機能が停止し、又は低下した場合に国民生活又は経済活動に多大な影響を及ぼすおそれが生ずるものに関する事業を行う者である。

[6] 本ガイドラインは、サイバーセキュリティ基本法 第 7 条第 1 項（サイバー関連事業者その他の事業者の責務）の「サイバー関連事業者その他の事業者は、基本理念にのっとり、その事業活動に関し、自主的かつ積極的にサイバーセキュリティの確保に努めるとともに、国又は地方公共団体が実施するサイバーセキュリティに関する施策に協力するよう努めるものとする」及び第 2 項（情報システム等供給者の責務）の「情報システム若しくはその一部を構成する電子計算機若しくはプログラム、情報通信ネットワーク又は電磁的記録媒体（以下この項において「情報システム等」という。）の供給者は、サイバーセキュリティに対する脅威により自らが供給した情報システム等に被害が生ずることを防ぐため、情報システム等の利用者がその安全性及び信頼性の確保のために講ずる措置に配慮した設計及び開発、適切な維持管理に必要な情報の継続的な提供その他の情報システム等の利用者がサイバーセキュリティの確保のために講ずる措置を支援する取組を行うよう努めるものとする。」に関連する考え方を整理したものであり、法的に新たな責任や規制を課すことを意図するものではない。

策の成熟度を向上させるツールとして活用できる。

この取組を進めるには、サプライチェーン全体（ソフトウェアコンポーネントの調達先、及びソフトウェア開発の委託先（開発委託の末端まで）を含む）でセキュアなソフトウェア開発・保守のプロセスを整備する必要があり、自組織及びソフトウェアサプライチェーンに関連する事業者においてもソフトウェア開発規約の変更といったプロセス変革などに相応の投資が必要になる。中長期的な目線での投資効果も念頭に、取組を進めるスタンスが不可欠である。

一方で、このようなソフトウェアの脆弱性を低減する開発を行うことで、短期的には脆弱性修正プログラムの作成コストを、長期的にはソフトウェアの保守コストを低減することが期待できる。また、セキュリティに配慮したソフトウェアを顧客が利用することで、設定ミスなどのリスクが低減すれば、顧客のセキュリティが向上し、サイバーインフラ事業者に対する信頼向上にもつながる。

- 顧客

特にソフトウェア製品やサービスの調達段階での利用が想定される。本ガイドライン（案）の要求事項を、自組織においてソフトウェアの開発、供給、運用を行う際の仕様としたり、本ガイドライン（案）の要求事項をチェック項目として、あらかじめソフトウェアやサービスの調達先であるサイバーインフラ事業者の取組を把握したりすることで、適切なソフトウェア開発事業者を選定することが可能となる。これらの取組を通じ、適切な事業者を選定することで、サイバーセキュリティのリスクを管理し、脆弱性修正プログラム導入等の運用負担軽減につながることが期待できる。

また、顧客が自ら利用するソフトウェアの開発部門、提供部門、運用部門を持つ場合、顧客としての責務に加え、自らサイバーインフラ事業者相当の責務と役割分担に基づく活動を独自に実施することで、ソフトウェアのライフサイクル全体のサイバーセキュリティのリスクに対応することができる。

サイバーインフラ事業者とともに進めるリスク対応のコストには、ソフトウェアサプライチェーンにおいて他の関連事業者が実施するセキュリティ対策が組み込まれる価値に対する対価が含まれる点は留意が必要である。また、顧客自らのリスク管理、及びセキュアな調達・運用のプロセスやリソースを整備する必要があるなど、相応の投資が必要になる。顧客においても、事業を進める上でのソフトウェアのセキュリティ確保の重要性を認識するとともに、本ガイドライン（案）の要求事項に関連した取組を特に意識し進めることで、リスク対応のコストの膨張を適正にコントロールしつつセキュリティを強化する姿勢が重要である。

1.3. 適用対象

(1) ソフトウェアの範囲

本ガイドライン（案）では、ソフトウェア製品、ソフトウェアサービス、IT/OT/IoT 機器などに組み込まれるファームウェア、IT/OT システム又は ICT サービスを構成するソフトウェアなど、ソフトウェアライフサイクル上で開発・保守される表 2 のソフトウェアを対象とする。（以下、IT/OT システム、ICT システムの各々、又は総称として、「システム」、「サービス」、「システム・サービス」の用語を使用する。）

表 2 対象とするソフトウェアの分類

名称	説明
ソフトウェア製品	製品として顧客に提供されるソフトウェア
ソフトウェアサービス	クラウドサービスなど顧客が直接利用する IT サービス
組み込みソフトウェア	IT/OT/IoT 機器などのハードウェア製品[7]として提供される組み込みソフトウェア及びファームウェア
システム・サービスを構成するソフトウェア	IT/OT システム又は ICT サービスを構成するソフトウェア。Web プログラム[8]などを専用に開発するアプリケーションソフトウェアのほか、OS、ソフトウェアパッケージ、ソフトウェアライブラリ、オープンソースソフトウェアなど、開発者が手配しシステムに統合した状態で、システムの構成要素として提供されるソフトウェア

(2) 想定する事業者

本ガイドライン（案）では、広くソフトウェアの設計を含めた開発・供給・運用に関わる「サイバーインフラ事業者」を対象として想定している。ソフトウェアのサイバーセキュリティに関わるレジリエンスを向上するためには、サイバーインフラ事業者は、インシデントの防御を対象とした関わりだけではなく、インシデントの事前対処と事後対処における情報収集、分析、対処調整の協力者として、様々な面で関係を強化していくことが求められる。本ガイドライン（案）では、サイバーインフラ事業者を、開発者、供給者、運用者の 3 つの主な役割で分類している。なお、ソフトウェアの効果的なサプライチェーン上のサイバーセキュリティ対策を進めるためには、サイバーインフラ事業者と顧客の適切な役割分担や、サイバーインフラ事業者が

[7]様々な接続機器が対象となる（ネットワーク機器、IoT 機器、制御装置、検査装置、輸送機器、医療機器等の各種接続機器など）。「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」では、調達する「機器等」を「サーバ装置、端末、通信回線装置、複合機、特定用途機器等、ソフトウェア等」と定義し、これら情報システム基盤を管理又は制御するソフトウェアのセキュリティ対策を求めている。

[8] Web デザイン業務において、スクリプト言語等を利用したプログラミングを実施することがある。そのようなケースでは、開発者に準じた責務が求められる。

関連する業界団体等、その他の関係機関との協力関係等も求められることから、その他のステークホルダーも対象として考慮する。これらの関係者の分類を表 3 に示す。

表 3 サイバーインフラ事業者及びステークホルダーの分類

分類	名称	説明
サイバーインフラ事業者	開発者	ソフトウェア製品、ソフトウェアサービス、組み込みソフトウェア、あるいはこれらのソフトウェアで構成されるシステム・サービスの設計を含めた開発又はインテグレーションに従事する事業者・人員 ソフトウェア開発ベンダー、ソフトウェアサービスプロバイダ、機器開発ベンダー、ソフトウェアやシステムの開発請負事業者、ソフトウェアコンポーネント開発事業者、インフラ事業者、自社開発ソフトウェアの開発部門などにおいて、ソフトウェアの開発又はインテグレーションを行う事業者等が対象となる。
	供給者	顧客にソフトウェア製品、ソフトウェアサービス、組み込みソフトウェア（ハードウェア製品を含む）、あるいはこれらのソフトウェアで構成されるシステム・サービスを提供する事業者・人員[9] ソフトウェア製品やソフトウェアを含む機器の販売会社、ソフトウェアサービスプロバイダ、システムの開発運用請負事業者、インフラ事業者、ソフトウェア開発ベンダーなどにおいて、ソフトウェアやシステム・サービスを提供する事業者等が対象となる。
	運用者	顧客に対して主にシステム・サービスの運用を支援する役務を提供する事業者・人員[10]
ステークホルダー	顧客	政府機関等及び重要インフラ事業者を始め、ソフトウェアの利用主体となる事業者等
	その他関係機関	サイバーレジリエンス向上の支援を担う組織

(3) システムを対象とした一般的な役割分担の想定

本ガイドライン（案）が対象とするサイバーインフラ事業者が扱うソフトウェアの資産について、ソフトウェアで構成するシステムの開発・契約形態・利用形態を踏まえた関係を図 1 に示す。ここでは、サイバーインフラ事業者を、システムの開発・契約・利用の観点から、以下の 2 つの役割を想定している。

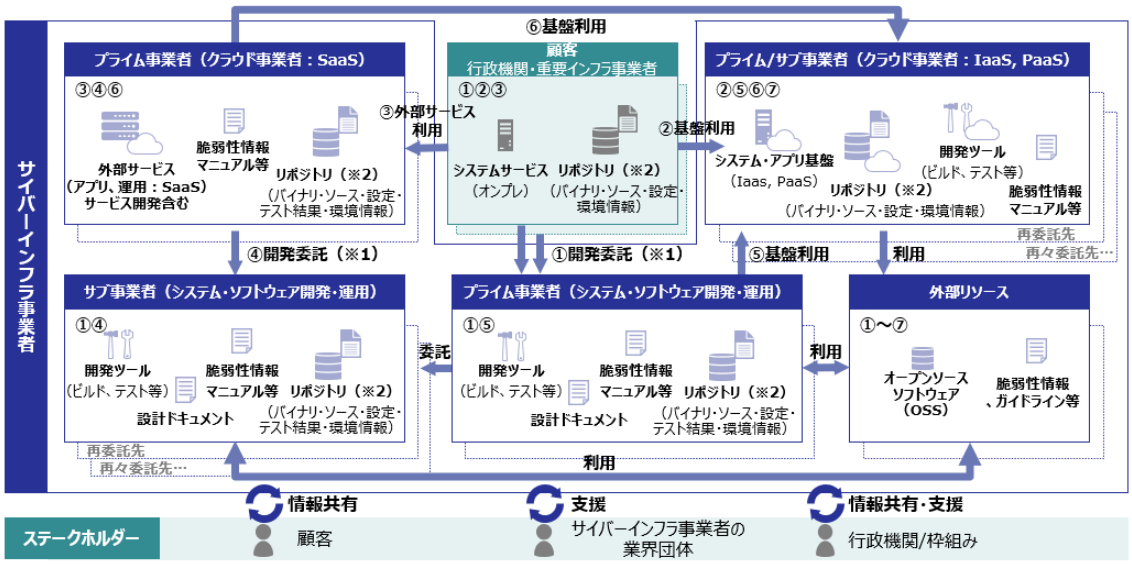
[9] 供給者内に、開発者・運用者が含まれるケースもある。また、サイバーインフラ事業者に販売会社が含まれるケースでは、供給者に準じた責務が求められる。

[10] ソフトウェアの利用主体である顧客がソフトウェアを運用することが一般的であるが、システム・サービス又はこれらを構成するソフトウェアの運用には専門的な知識や技能が必要な場合も多い。ここでは、顧客との契約により、サイバーインフラ事業者がソフトウェアの運用（又はその一部）を支援する場合を想定する。

- プライム事業者
  顧客と直接契約を結びシステムやクラウドサービスの開発・供給・運用を実施する 1 次請け事業者

- サブ事業者
  プライム事業者と契約を結びシステム・クラウドサービスの開発・供給・運用を実施する 2 次請け以降の事業者[11]

プライム事業者とサブ事業者の関係は、グループ会社と資本関係のない外部委託先のいずれかを想定する。サブ事業者のサプライチェーンは、複数層の委託構造、かつ各層内においても複数の階層構造を成していることもある。また、外部リソースとは、OSS のようなソフトウェアの公開リポジトリであり、脆弱性情報等を含めて公開するなど、有志組織等により運用されているリソース群である。



※1 ①④にて下請けを使用しないケースもある。顧客がサイバーインフラ事業者として自ら開発するケースもあり。
※2 サービス及びシステム開発や契約形態等によりソフトウェアの所有権が異なるが、所有権については表記していない。

[システムの開発・運用・利用形態の例]
①：オンプレミスのシステム（プライム事業者に開発委託）　②：顧客がIaaS, PaaSを利用
③：SaaSを利用したシステムの利用　④：クラウド事業者（SaaS）が開発委託
⑤：プライム事業者（SIer等）がIaaS, PaaSを利用　⑥：クラウド事業者（SaaS）がIaaS等を利用
⑦：クラウド事業者（IaaS, PaaS）が相互利用

図 1 ソフトウェアで構成するシステムの関係者の概念図

(4) 想定するリスク

本ガイドライン（案）が対象とする、ソフトウェアが関係すると想定されるサイバーセキュリティリスクとは、ソフトウェアへの悪意のある攻撃や設計を含めた開発上の不備・設定ミス等による電磁的な情報の漏えい、滅失又は毀損などの安全管理上の懸念の度合い、あるいは電磁的な情報を扱うシステムやネットワークへの攻撃や開発上の不備・設定ミス等による安全性・信頼性の低下又は停止などの維持管理上の懸念の度合いをいう。これらのサイバーセキュリティのリスクが顕在化する要因となるものは、ソフトウェア製品、あるいはソフトウェアで構成されるシステムやサービスの分析・計画フェーズに始まり、要件定義、設

[11] SaaS 事業者間では一般に「請負」の表記を用いないが、ここでは表記上ソフトウェア開発とサービスの両方に「請負」の用語を使用している。

計・開発・テスト、リリース、運用、廃棄にいたるまで多種多様なところに広く潜在している。分析・計画フェーズでのリスク分析の不足、要件定義フェーズでのセキュリティ要件の合意不足、開発フェーズでの不正なコードやコンポーネントの挿入、レビュー不足、ソフトウェア配布フェーズでの改ざん、運用中のサービス停止、全てのフェーズに関わるヒト・モノ・カネの整備不足やサプライチェーンの管理不足などである。本ガイドライン（案）では、これらソフトウェアの全てのフェーズにおいてソフトウェアのなりすまし、改ざん、否認、情報漏えい、サービス拒否、権限昇格に関わる脅威を想定している。

1.4. 役割分担の考え方

ソフトウェアのライフサイクルに関わる実務においては、ソフトウェアの特性、ソフトウェアの開発・供給の体制、ソフトウェアの利用、運用及び開発に係る契約形態などに基づいて、各関係者の責務と役割分担を定め、顧客であるソフトウェアの利用主体におけるサイバーセキュリティリスクへの対応を推進することが重要である。ここでは、各ソフトウェアの対象範囲における主な役割分担の例を表で示すとともに、代表的な役割分担の考え方を解説する。

本ガイドライン（案）は、ソフトウェアの供給と利用に関与する事業者等の責務区分として、「**サイバーインフラ事業者**」と「**顧客**」を区分し、かつサイバーインフラ事業者の役割分類を、開発者、供給者、運用者に分類する。自らの責務区分と役割分担を特定する際には、図 2 に示すように、対象とするソフトウェアの特性（ソフトウェアの対象範囲、各役割の分担の方針など）を念頭に、どのような立場、役割の範囲を担当するのかを理解する。その上で、担当するソフトウェアの構成上の位置付けと関連する他の開発・供給体制との役割分担、あるいは契約に基づいて規定される役割などを踏まえて、責務区分と役割分担を特定する。

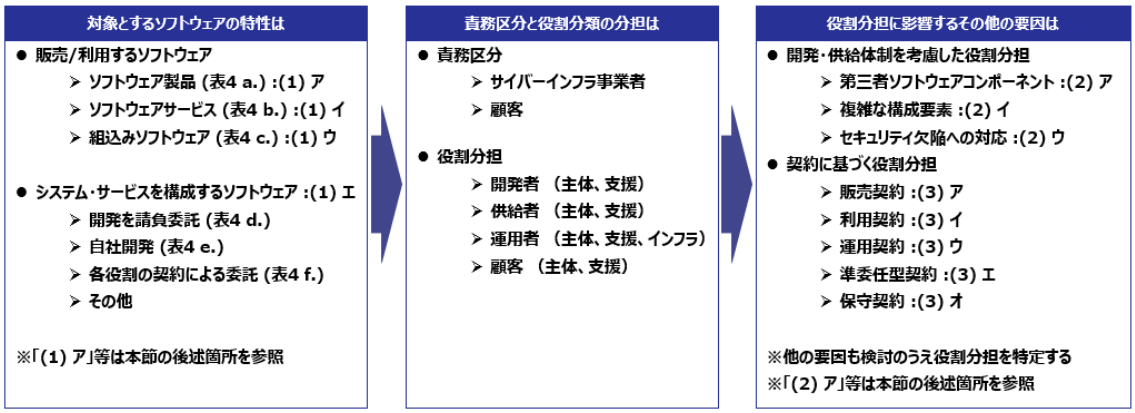


図 2 責務区分と役割分担を特定するための考え方

表 4 は、対象とするソフトウェア特性における各サイバーインフラ事業者と顧客の想定を例示するとともに、各々の責務区分において該当する役割を「✓」で示したものである。一般的に主体的な立場と支援的な立場で役割を分担することが想定されるところについては、「(主体)」、「(支援)」を付記した。また、システムが動作する基盤を提供する役割には、「(インフラ)」を付記した。なお、以下の各役割分担の考え方において、想定する関連事業者が「顧客」と「運用者」の役割を兼ねる場合、役割ごとに区別して記載している。顧客と同じ関連事業者であっても、役割が「運用者」である部門・担当の場合、サイバーインフラ事業者の「運用者」相当の責務を負うものと考える。また、顧客が自組織において開発、供給を実施する場合は、顧客自身が責務区分「サイバーインフラ事業者」としての各役割である「開発者」、「供給者」の責務を自ら「(主体)」として負うものと考える。

表 4 関連事業者の想定と責務区分・役割の例示

関連事業者の想定		役割分類 / 責務区分	開発者	供給者	運用者	顧客
a. ソフトウェア製品						
ソフトウェア開発ベンダー		**サイバーインフラ事業者**	✓			
販売会社		**サイバーインフラ事業者**		✓		
購入者(運用担当者)		**サイバーインフラ事業者**			✓	
購入者(利用者)		**顧客**				✓
b. ソフトウェアサービス（サービス間連携を含む場合）						
サービスプロバイダ(プライム事業者)		**サイバーインフラ事業者**	✓(主体)	✓	✓	
サービスプロバイダ(サブ事業者)		**サイバーインフラ事業者**	✓(支援)			
サービス開発支援(サブ事業者)		**サイバーインフラ事業者**	✓(支援)			
インフラ事業者(サブ事業者)		**サイバーインフラ事業者**	✓			
サービス利用者(アプリ運用担当者)		**サイバーインフラ事業者**			✓	
サービス利用者(アプリ利用者)		**顧客**				✓
c. 組み込みソフトウェア						
機器開発ベンダー		**サイバーインフラ事業者**	✓			
組み込みソフトウェア開発部門		**サイバーインフラ事業者**	✓			
販売会社		**サイバーインフラ事業者**		✓		
購入者(運用担当者)		**サイバーインフラ事業者**			✓	
購入者(利用者)		**顧客**				✓
d. システム（システムオーナーが企画、開発・運用・インフラサービスを調達）						
開発運用請負事業者		**サイバーインフラ事業者**	✓(主体)	✓(主体)	✓(支援)	
開発支援		**サイバーインフラ事業者**	✓(支援)	✓(支援)		
ソフトウェアコンポーネント開発		**サイバーインフラ事業者**	✓	✓		
インフラ事業者(IaaS/PaaS)		**サイバーインフラ事業者**	✓	✓(インフラ)	✓(インフラ)	
調達者(システム運用者)		**サイバーインフラ事業者**			✓(主体)	
調達者(システムオーナー)		**顧客**				✓
e. システム（自社開発、系列事業者が開発・供給・運用を支援）						
親事業者(開発部門)		**サイバーインフラ事業者**	✓(主体)	✓(主体)		
系列事業者		**サイバーインフラ事業者**	✓(支援)	✓(支援)	✓(支援)	
親事業者(運用部門)		**サイバーインフラ事業者**			✓(主体)	
親事業者(利用部門)		**顧客**				✓
f. システム（顧客である事業者の利用部門、運用部門、開発部門が各役割を(主体)として取り組む場合において、各役割の業務の一部分(支援)を、それぞれ調達者として準委任型委託契約で他の事業者に委託するケースの例）						
調達者(開発部門)		**サイバーインフラ事業者**	✓(主体)	✓(主体)		
調達者(運用部門)		**サイバーインフラ事業者**			✓(主体)	
調達者(利用部門)		**顧客**				✓(主体)
コンサル(システム化構想)	ケース	**顧客**				✓(支援)
調査事業者(PMO 支援)	ケース	**顧客**				✓(支援)
開発ベンダー(開発)	ケース	**サイバーインフラ事業者**	✓(支援)	✓(支援)		
運用ベンダー(運用保守)	ケース	**サイバーインフラ事業者**			✓(支援)	

(1) ソフトウェアの特性による役割分担の考え方

対象とするソフトウェアの特性による役割分担の考え方を示す。

ア. 対象がソフトウェア製品の場合

対象がソフトウェア製品の場合、開発者と顧客は別の事業者であり、供給者はソフトウェア製品の販売代理店又は開発者による直販（開発者が供給者を兼ねる）となるのが一般的である。また、ソフトウェア製品の運用者は、ソフトウェア製品の利用者である顧客自身若しくは顧客の運用部門等が担当する場合が一般的である。

イ. 対象がソフトウェアサービスの場合

対象がソフトウェアサービスの場合、開発者と顧客は別の事業者であり、サービスプロバイダが、開発者、供給者、及び供給するサービス階層の運用者を兼ねるのが一般的である。顧客であるサービス利用者がソフトウェアサービスを基盤として独自にアプリケーションを設定・動作させる場合、例えばクラウドサービスを利用する場合などでは、責任共有の考え方に基づいて運用者としての各ステークホルダーの責任範囲を定めるのが一般的である。その場合、サービスプロバイダがインフラ上に構築したシステムの運用を担当し、サービスの利用者はアプリケーションの運用を担当するなど、運用の責任を分担・共有する。

ウ. 対象が組み込みソフトウェアの場合

対象が組み込みソフトウェアの場合、そのソフトウェアを組み込んだ機器として販売し利用されることを想定し、開発者はソフトウェアの開発部門を擁する機器の開発者と捉えるのが一般的である。組み込みソフトウェアを含む機器の運用者は、顧客自身若しくはユーザである顧客の運用部門等が担当する場合が一般的である。

エ. 対象がシステム・サービスを構成するソフトウェアの場合

対象がシステム・サービスを構成するソフトウェアの場合、システム・サービスの利用若しくは提供の主体となる事業者等（一般的にシステムオーナーと呼ばれる事業者）が、本ガイドライン（案）の顧客の役割を担当する。システム・サービスの開発・供給・運用は、その規模や求められる専門知識・技能により、システムオーナーとは別の事業者群が担当する場合が想定され、プライム事業者、サブ事業者、あるいはインフラ環境を提供するクラウド事業者など複数層の委託構造かつ各層内において複数の階層構造をなすことも想定される（図 1 を参照）。このような場合は、システム・サービスの構成要素や開発・供給プロセス、運用プロセスの体制を元に、開発者としての階層構造、供給者としての階層構造、運用者としての階層構造、及びそれらの相互連携の構造を踏まえ、それぞれの役割分担を定める必要がある。IT システムの運用に関しては、システムオーナー（顧客）の運用部門が、運用者としての役割の全体又は運用を外部若しくは引き入れ委託する事業者群の取りまとめを担当することが一般的であり、このような運用体制全体として運用者の役割を分担する。

(2) ソフトウェアの開発・供給体制による役割分担の考え方

ソフトウェアの開発・供給体制と役割分担の考え方の例を示す。

ア. 第三者のソフトウェアコンポーネントを含む場合の開発・供給の役割分担

ソフトウェアの構成要素に第三者のソフトウェアコンポーネントを含む場合、この第三者はコンポーネントの開発者、及びそのコンポーネントの供給先となる開発者に対するコンポーネント供給者の役割を分担する各事業者の位置付けとなる。

イ. ソフトウェアが複雑な構成要素を持つ場合の開発・供給の役割分担

ソフトウェアの構成要素の構造が複雑化し、第三者のソフトウェアコンポーネントを複数含む場合やこのようなソフトウェアのコンポーネント構造が階層的に複雑化する場合などが想定される。システム・サービスにおいては、このような構造を持つソフトウェアを更に複数組み合わせて構成する場合などが想定される。ソフトウェアの各構成要素単位で責務を担う開発者が存在し、複数の構成要素を組み合わせて動作させるソフトウェアに対しても各ソフトウェア単位で責務を担う開発者が存在する。このようなソフトウェアの開発に関わる全ての事業者は、本ガイドライン（案）の開発者（及び担当するコンポーネントの供給者）としての責務を認識し、役割分担を果たすことが期待される。少なくとも、ソフトウェアの構成要素の全ては、いずれかの事業者に開発者としての責務が生ずる状態とすることが原則である。このような開発・供給体制の下、ソフトウェアの開発、供給、及び欠陥修正の対応の体制を定め、適切に役割を分担する。

ウ. セキュリティ欠陥への対応に関わる開発・供給の役割分担

ソフトウェアの一次的な利用者である顧客との接点において、開発者によるテスト済みソフトウェアをセキュアにリリースする役割を供給者が担い、運用時に発見されたセキュリティ欠陥のおそれ（脆弱性を含む可能性が予見されるもの）の通知受付窓口、セキュリティ勧告の提供を含む欠陥修正の一連のプロセスの責務を開発者が担う役割分担とする。

(3) ソフトウェアの利用、運用及び開発に係る契約形態による役割分担の考え方

ソフトウェアの利用、運用及び開発に係る幾つかの契約形態による役割分担の考え方の例を示す。

ア. 販売契約による製品購入の場合

顧客が製品として購入したソフトウェアの利用及び運用の主体は、原則として顧客が担当する。ソフトウェアの利用に関しては、一般的に顧客とソフトウェアの供給者との間で使用許諾（ライセンス）契約及び保守契約が締結される。開発者と供給者（主に販売者）との間は、販売契約を締結し、ソフトウェアの販売権、保守に対する役割分担などが規定される。

イ.　利用契約によるサービス利用の場合

サービスプロバイダが提供するソフトウェアサービス（クラウドサービスなど）を顧客が利用する場合、顧客の役割としては、サービスを利用する側面と、運用の一部をアウトソーシングする側面がある。特に、運用に関しては、責任共有モデルの考え方に基づき、顧客が責任を持って主体的に運用する範囲、サービスの運用責任の一部をサービス供給者であるサービスプロバイダに委ねる範囲、及びこれらの範囲の分界点を特定する。その上で、利用規約と SLA（Service Level Agreement）による利用契約を締結する。

ウ.　運用契約による運用委託の場合

顧客がソフトウェアを含むシステムの運用の全部若しくはその一部をサイバーインフラ事業者に委託する場合は、運用者となるサイバーインフラ事業者と運用契約（必要に応じて保守契約を含む）を締結し、その契約に基づいて対象ソフトウェアを含むシステムの運用の役割を分担する。

エ.　準委任型契約による役務委託の場合

顧客がソフトウェア開発を委託する場合は、窓口となるサイバーインフラ事業者（開発・供給）との間で請負型あるいは準委任型のソフトウェア開発委託契約を締結する。顧客側でセキュリティ要件を含む開発仕様が作成され、ソフトウェア開発を含むシステム開発（設計、プログラミング、テスト、設置、導入、データ移行、教育、リリース準備など）の完成責任を負う場合は請負型の契約を締結する場合が多い。一方、システム化構想など仕様検討段階の役務提供などでは準委任型の契約を締結する場合もある。また、準委任型契約により開発や運用の役務提供を受ける場合もある。請負型の開発委託契約の場合、請け側の事業者が開発者及び供給者の役割を担当する。準委任型の委託契約の場合、役務提供をどの役割範囲で実施するのかを定め、役割分担とその役割に応じた責務があることを認識し、契約において責務に応じた実施内容を明確化することが望ましい。

オ.　保守契約によるソフトウェアの開発・運用の役割分担

ソフトウェア開発を含むシステム開発の完了後、顧客はソフトウェアを含む開発システムの受入れを行い、そのシステムの運用を開始するに当たり、ソフトウェアの不具合対応を含む保守契約を開発者との間に締結する（開発の契約不適合責任に関しては個別に整理が必要）。

保守契約には、問合せ対応、不具合の原因調査、規定に基づく不具合修正対応（あるいは更新したソフトウェアの提供）などが一般的に含まれるが、ソフトウェアの保守契約においては、保守に関わる実際の業務やサービスの内容に適した契約形態を選択する。情報提供を主とする契約の場合は、開発・供給側からの情報提供やバージョンアップの提供が主となり、提供された情報や更新の適用は顧客側の運用部門が担当する。一方、ソフトウェアの問合せ（使用方法、不明点、技術的問題に対する確認・質問など）や不具合修正を所定の範囲で対応するような保守の場合、準委任契約として締結する場合が多い。ただし、ソフトウェアの不具合対応において修復に関する完成責任を負う場合は請負契約として締結するケースもある。

本ガイドライン（案）では、ソフトウェア製品の保守契約には、欠陥修正を含む脆弱性対応の開発者責務を含むことを想定する。また、開発委託をしたソフトウェアに対しては、請負型の保守契約、若しくは準委任型の保守契約に加え、脆弱性対応の開発者責務を含む請負契約相当の保守契約を締結し、合意の上仕様及び費用等の変更に関する覚書を締結するなどの運用を想定する。

1.5. 代表的なユースケース例

ソフトウェアのライフサイクルに関わる実務においては、ソフトウェアの特性、ソフトウェアの開発・供給の体制、ソフトウェアの利用及び運用に係る契約形態などに基づいて、各関係者の責務と役割分担を定め、顧客であるソフトウェアの利用主体におけるサイバーセキュリティリスクへの対応を推進する。ここでは、以下の４つのユースケース例において、複数のサイバーインフラ事業者による役割分担について例示し解説する。

- ソフトウェア製品・組み込みソフトウェアにおける役割のユースケース例
- ソフトウェアサービスにおける役割のユースケース例
- システム（開発を請負委託）における役割のユースケース例
- システム（自社開発）における役割のユースケース例

① ソフトウェア製品・組み込みソフトウェアにおける役割のユースケース例

ここでは、ソフトウェア製品の開発・供給のユースケース例として、購入者である顧客がソフトウェア製品を調達する場合を示す（図 3 参照）。ソフトウェア製品は販売会社から調達し、販売会社は注文に応じて（又は仕入れとして）ソフトウェア開発ベンダーにソフトウェア製品を発注する。ソフトウェア製品の開発、製品化、出荷等はソフトウェア開発ベンダー（プライム事業者）が担当し、ソフトウェアコンポーネントの開発を社外のソフトウェア開発会社（サブ事業者）が担当する。

また、IoT 機器（組み込みソフトウェアを含む）の開発・供給のユースケース例として、購入者である顧客が組み込みソフトウェアを含む IoT 機器を調達する場合を示す。IoT 機器は販売会社から調達し、販売会社は注文に応じて（又は仕入れとして）機器開発ベンダーに IoT 機器を発注する。IoT 機器の開発、組み込みソフトウェアの実装、製品化、出荷等は機器開発ベンダーが担当し、組み込みソフトウェアの開発を機器開発ベンダーの組み込みソフトウェア開発部門が担当する。

仮に、ソフトウェア製品を提供するソフトウェア開発ベンダーが、ソフトウェアのアップデートサービスをクラウド経由で提供する際に、そのシステム基盤としてクラウド事業者が提供する SaaS を利用する場合、このクラウド事業者は供給者であるとともに開発者・運用者の役割を担当する。また、いずれかの事業者が外部リソースを利用する場合も、その利用形態に応じて開発者、供給者、又は運用者が適切に管理することになる。

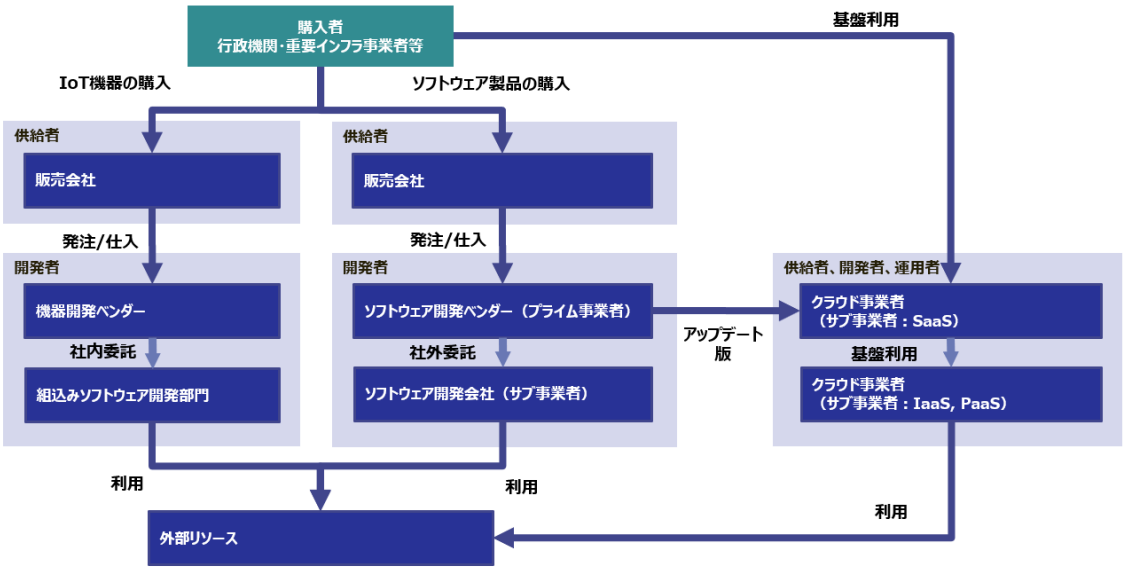


図 3 ソフトウェア製品・組み込みソフトウェアにおける役割のユースケース例の概念図

② ソフトウェアサービスにおける役割のユースケース例

ここでは、ソフトウェアサービスとして、SaaS を利用したクラウドサービスを調達する場合のユースケース例を示す（図 4 参照）。SaaS サービスをプライム事業者であるサービスプロバイダが供給するとともに、開発者、運用者の役割を担当し、SaaS サービスを構成するソフトウェアの開発をサブ事業者であるサービス開発会社が担当、SaaS サービスを動作させる IaaS サービスの供給、及び開発・運用を同じ若しくは別のクラウド事業者が担当（図の①基盤利用の「インフラ事業者（サブ事業者:IaaS、PaaS）」）するユースケースなどが想定される。更にクラウド事業者若しくはサブ事業者が外部リソースを利用する場合は、その利用形態に応じて開発者、供給者、又は運用者が適切に管理することになる。

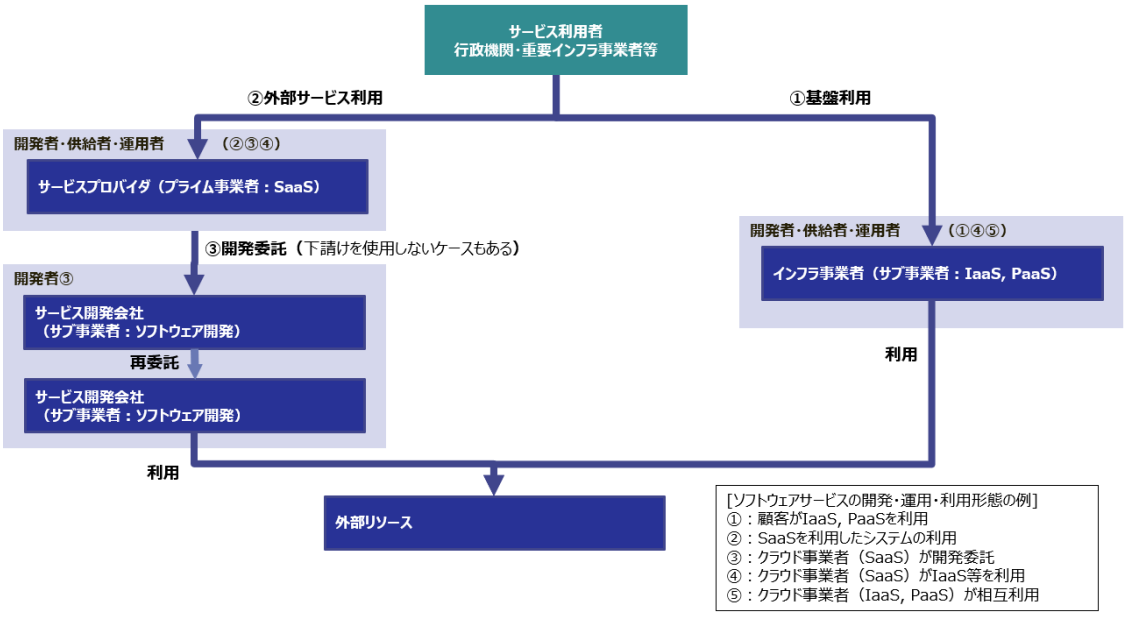


図 4 ソフトウェアサービスにおける役割のユースケース例の概念図

③ システム（開発を請負委託）における役割のユースケース例

ここでは、ガバメントクラウドを利用した業務システムの設計・開発・運用・保守の調達などで一般的にみられる IT システムの開発を請負委託する場合のユースケース例を示す（図 5 参照）。IT システムを開発・デプロイし、その IT システムの運用・保守作業の支援・代行を含めて調達する場合のユースケース例を示す。IT システムを構成するソフトウェアに対する役割として、SIer であるプライム事業者が供給者であるとともに開発者、運用者の役割を担当し（図の①開発委託先の「プライム事業者：システム・ソフトウェア開発・運用」）、開発の一部を請け負うサブ事業者や、IT システムの構成要素であるソフトウェア製品や IoT 製品の開発・製造を行うサブ事業者（図の委託先である「サブ事業者：システム・ソフトウェア開発」とその再委託先など）が、開発の役割を担当するユースケースなどが想定される。

仮に、システム基盤として、プライム事業者経由でサブ事業者であるクラウド事業者の PaaS を契約・利用する場合（図の②基盤利用の「インフラ事業者（サブ事業者:IaaS、PaaS）」）、このインフラ事業者は供給者であるとともに開発者、運用者の役割を担当する。また、外部リソースを利用する場合は、その利用形態に応じて開発者、供給者、又は運用者が適切に管理することになる。

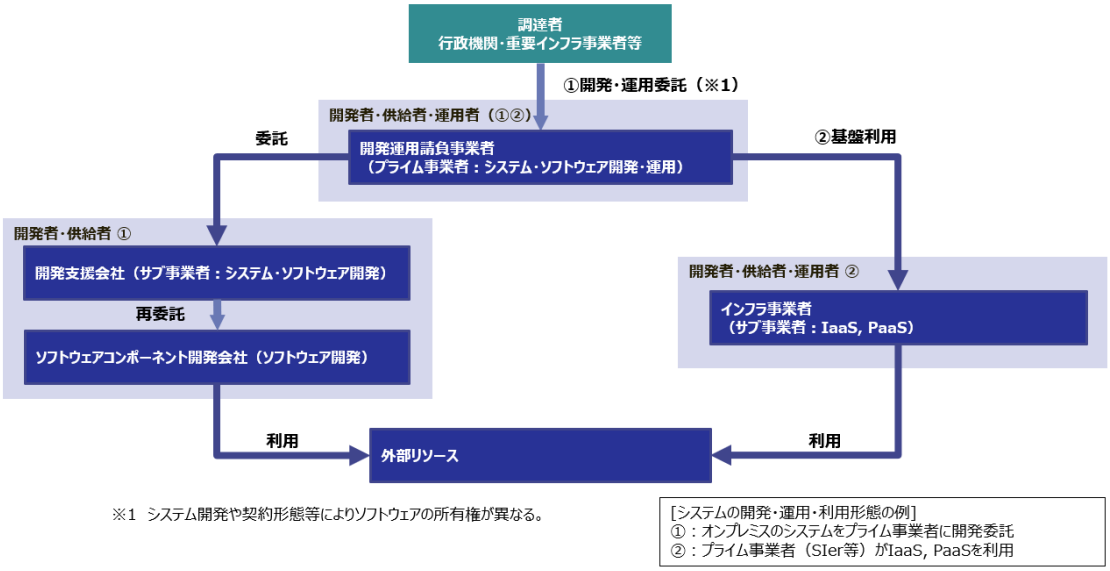


図 5 システム（開発を請負委託）における役割のユースケース例の概念図

④ システム（自社開発）における役割のユースケース例

社内で利用する IT システムのシステムオーナーである事業者が自ら開発を行う場合のユースケース例を示す（図 6 参照）。その事業者内に開発部門があり、利用部門（本ガイドライン（案）の顧客に相当）において利用する IT システムの運用をサポートする運用部門を擁するケースがある。このような場合の責務と役割分担は、利用部門（顧客）、開発部門（開発者）、運用部門（運用者）をそれぞれ担当するなどが想定される。

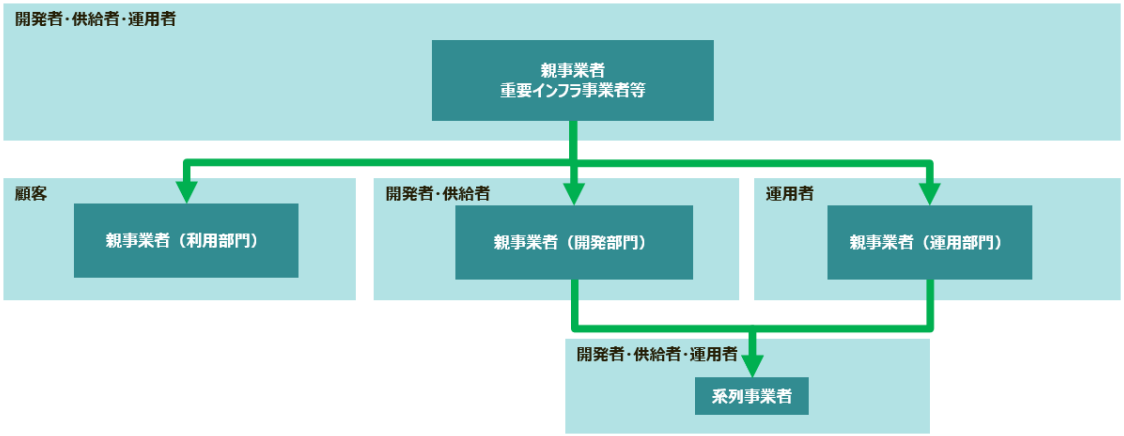


図 6 システム（自社開発）における役割のユースケース例の概念図

2. サイバーインフラ事業者と顧客の責務と役割分担

2.1. 責務と役割分担の考え方

ソフトウェアサプライチェーン上でのセキュリティリスクを軽減するには、サイバーインフラ事業者一社の単独での取組には限界があり、サプライチェーンを構成するサイバーインフラ事業者が、それぞれ又は連携して顧客と協調しつつ、それぞれの責務を果たす必要がある。要件定義フェーズを例に挙げると、事業者が適切なリスク分析を行うとともに、顧客にも自らがオーナーシップを持つシステム全体のリスク管理に関する責務があり、リスクを把握しなければ、適切なセキュリティ要件の合意は難しく、また、ソフトウェアの脆弱性が残存する一因にもなる。

つまり、顧客は、経営層のリーダーシップの下、自組織のシステムに関するリスク管理についてサイバーインフラ事業者との役割分担を明確化するとともに、ソフトウェア製品・サービスの利用者として自組織で判断や調整を行わなければならない事項を把握し、適切な製品の購入等ができるよう、サイバーインフラ事業者にセキュリティに関する要求事項を提示し、依頼した業務の結果の品質を自社で評価できる体制を整備する必要がある。

一方、サイバーインフラ事業者は、自社製品・サービスのセキュリティ対策に関する責務があり、顧客だけにセキュリティの責務を負わせないよう、経営層が主導して対策を進めることが求められていると言える。

これらの概念を責務として以下にまとめている。

2.2. 責務

サイバーセキュリティに関するレジリエンスを向上させるためには、サイバーインフラ事業者と顧客がそれぞれの責務を果たすことで、相補的な効果を得ることができる。

＜サイバーインフラ事業者が認識すべき責務＞

サイバーインフラ事業者は、サイバーセキュリティに関するレジリエンス向上のために、以下の５つの責務を認識することが求められる。これらの全ての責務は、サイバーインフラ事業者の経営層が認識し、経営層のリーダーシップにより責務を果たす取組を実施することが求められる。

（１） セキュリティ品質を確保したソフトウェアの設計・開発・供給・運用

- **セキュアなソフトウェアの提供と対策評価**
  「セキュアバイデザイン」及び「セキュアバイデフォルト」の原則に則り、リスクベースのアプローチによりソフトウェア開発・運用に対する脅威を軽減するための対策を実施し、その有効性を判断する。また、ソフトウェアに対して最低限のソフトウェアセキュリティ標準を実施する。

- **ソフトウェアのライフサイクル全体でのサイバーセキュリティの考慮**
  セキュリティ要件の合意に始まり、セキュアなビルド、テスト、運用等、顧客と合意したソフトウェアライフサイクル全体にサイバーセキュリティを考慮する。

（２） ソフトウェアサプライチェーンの管理

- **セキュリティ管理策の実施状況の共有**
  利用者がソフトウェアの調達と導入に関して、リスクに基づいたソリューションの選択を含む意思決定を行えるよう、供給者はソフトウェア開発の取組状況を開示する。
  顧客に周知する必要があるサイバーセキュリティの側面について透明性を確保する。

- **ソフトウェア構成情報の共有**
  利用者による脆弱性対策のため、ソフトウェア部品表（SBOM）、設定情報をはじめとする OSS も含めたソフトウェアの構成管理による情報を活用する。

- **サプライチェーンを含むリスクマネジメントの推進**
  供給者（システムインテグレーター、外部システムサービスプロバイダ、パートナー等）、開発者、その他の関連する IT/OT/ICT 関連の事業者全てをソフトウェアサプライチェーン・リスクマネジメントの活動範囲に含める。

（３） 残存脆弱性への速やかな対処

- **脆弱性と脅威情報の共有と対処**
  脆弱性開示ポリシーを整え、脆弱性対応に関する体制を整備する。ベンダーは、クラウドサービス・ソフトウェアの脆弱性の特定と開示、セキュアなサービス構成と運用に必要な情報の提供、サービスのバージョンアップ、パッチの開発と配布に責務を有すること、ベンダーはバージョンアップ・パッチ適用プロセスを文書化して、顧客がプロセスへの参加方法を理解できるようにする。また、顧客に確実に通知する仕組みを整える。

（４） ソフトウェアに関するガバナンスの整備

- **ソフトウェアサプライチェーン・リスク管理を企業のリスク管理に統合**
  ソフトウェアサプライチェーン・リスク管理は、ソフトウェアライフサイクル全体にわたる活動を対象とし、企業のリスク管理プロセスの一部として集約する。
  自組織として許容可能なレベルまでリスクを低減するために必要なリソース（ヒト、モノ、カネ）を整える。サイバーセキュリティを経営の重要事項として位置付け、トップマネジメントがリスクマネジメント実施の責務を負う。
  法令を遵守する。

（５） サイバーインフラ事業者・ステークホルダー間の情報連携・協力体制の強化

- **関係者間での脅威・脆弱性情報の共有と対処**
  脅威情報・脆弱性情報を、政府及び産業界のパートナーとの間で迅速かつ時宜を得た形で共有する。供給者がソフトウェアの脆弱性情報を所管する機関と共有する。

- **サイバーセキュリティに関わる関係者の協働**
  コミュニティを含む全てのステークホルダーが健全に連携し合う。
  潜在的なリスクを特定し、サイバーセキュリティに関連するサプライチェーン・リスクの依存関係を評価するための枠組みを開発するために協働する。
  セキュリティ対策は、プラットフォームプロバイダや消費テナント組織等も含むサプライチェーン全体で責務を共有して取り組む。
  民間部門が、政府と協力しながら、必要な要件に継続的に適応し、重要インフラを提供する事業者が依存する技術、製品及びサービスのセキュリティを改善する。
  ステークホルダーの適切で時宜を得た参加により、知識、認識等を共有でき、適切なリスクマネジメントにつながる。

＜顧客に求められる責務＞

顧客がオーナーシップを持つシステムを構成するソフトウェアのセキュリティに関わる活動において、顧客には以下の責務が求められる。

（６） 顧客の経営層のリーダーシップによるリスク管理とソフトウェア調達・運用

- **顧客の経営層のリーダーシップによるリスク管理**
  顧客の独立した主体的な取組及びサイバーインフラ事業者との契約に基づく協力的な取組によるリスク管理
  既知の脆弱性への対処及び緩和策を主体的に実施するためのリソースの割当てと整備
  セキュリティ改善を目的とするコミュニティや協力体制の活用

- **顧客の経営層のリーダーシップによるソフトウェア調達・運用**
  ソフトウェア設計計画にセキュリティ機能を組み込むためのセキュリティ要件の提示
  ソフトウェアの調達・導入におけるセキュリティ慣行の要求の開示
  ソフトウェアの調達・導入におけるリスク評価に基づいた意思決定
  ライフサイクルを考慮したソフトウェアの運用、リスク対応及び契約に係る予算確保

サイバーインフラ事業者が認識すべき責務に基づく活動において、顧客との接点を持つ活動は、上記の顧客に求められる責務を顧客が認識し、顧客がその責務を果たす活動を合理的な合意に基づいて支援することで、サイバーセキュリティのレジリエンス向上に貢献する姿勢が重要である。

3. 責務を果たすための要求事項

3.1. 要求事項の全体像

サイバーインフラ事業者と顧客は、サイバーセキュリティに関するレジリエンス向上の責務を果たすために、対象となるソフトウェアの特性や組織に適した方法で、以下のサイバーセキュリティ対策の要求事項（6 のカテゴリ、21 の要求事項）を実施することが求められる。これらを実現するためには、組織のリスクマネジメントを担う経営層のリーダーシップの下、リスクに応じた対策の実施方針、予算や人材の割当て、実施状況の確認や問題の把握と対応、その他の関係機関との協力等を的確に進めることが求められる。

自組織での対応が困難又は専門事業者による実施が適切と判断される取組については、その一部を外部委託によって実施することの検討も求められる。

これらの考え方に基づくサイバーセキュリティのレジリエンス向上の要求事項を以下に示す。なお、各要求事項の識別は “S(n1)-n2” の形式（n1 はカテゴリ番号、n2 はカテゴリ内の連番）とし、各要求事項の個別要求の識別は “S(n1)-n2.n3” の形式（n3 は当該要求事項内の個別要求の連番）とする。

＜サイバーインフラ事業者に求められる要求事項＞

（１） セキュアな設計・開発・供給・運用

脆弱性を抑え、セキュリティを備えたソフトウェアを開発・供給・運用する

S(1)-1 設計時のリスク評価と対策の追跡

S(1)-2 セキュアなビルド

S(1)-3 テスト

S(1)-4 サービスのモニタリング

（２） ライフサイクル管理、透明性の確保

ソフトウェア管理の透明性をライフサイクル全体で確保しサプライチェーンを含むリスク管理を行う

S(2)-1 セキュアなコンポーネントの手配

S(2)-2 リリースファイルやデータのセキュアなアーカイブ

S(2)-3 関係者間のセキュリティ要件の確立

S(2)-4 利用者への適切な情報提供

（３） 残続する脆弱性の速やかな対処

リリースしたソフトウェアに残存する脆弱性を特定し、速やかに対応する

S(3)-1 継続的な脆弱性調査

S(3)-2　検知した脆弱性への対処

S(3)-3　対処結果を組織のプロセス改善に活用

（４）　人材・プロセス・技術の整備

組織レベルでソフトウェアに関わる人材・プロセス・技術を整備する

S(4)-1　人材：経営層のコミットメントと人員の整備

S(4)-2　プロセス：開発ポリシーの確立と法令順守

S(4)-3　プロセス：運用ポリシーの確立と法令順守

S(4)-4　プロセス：開発・運用基準の策定

S(4)-5　技術：セキュアな開発ツールの整備

S(4)-6　技術：セキュアな開発環境の整備

（５）　サイバーインフラ事業者・ステークホルダー間の関係強化

サイバーインフラ事業者・ステークホルダー間の情報連携・協力体制を強化する

S(5)-1　情報連携のための組織体制

S(5)-2　協力体制の強化

＜顧客に求められる要求事項＞

（６）　顧客によるリスク管理とセキュアなソフトウェアの調達・運用

顧客経営層のリーダーシップによるリスク管理とセキュアなソフトウェア調達、運用を行う

S(6)-1　顧客経営層のリーダーシップによるリスク管理

S(6)-2　顧客経営層のリーダーシップによるソフトウェアの調達、運用

これらの要求事項の 6 カテゴリと一般的なセキュリティ対策の体系との関係の概念図を図 7 に示す。

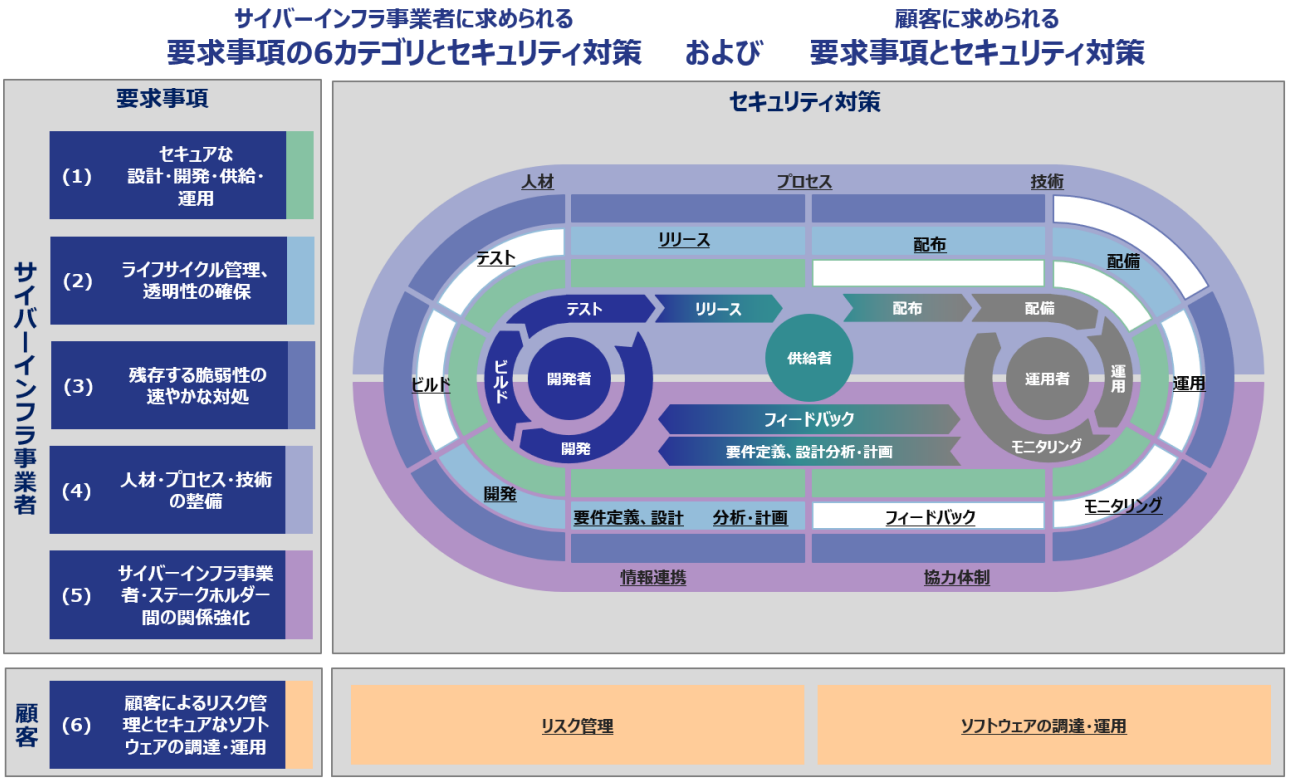


図 7 要求事項の概念図

参考情報には、本ガイドライン（案）で示す要求事項のチェックリスト、プラクティス例、関係する参照情報や用語の説明等を記載している。

3.2. 要求事項

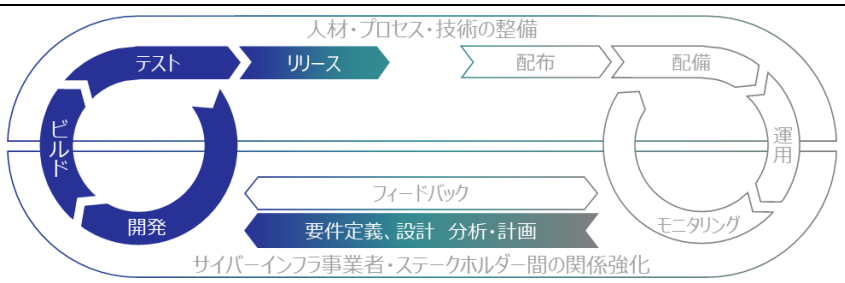
本ガイドライン（案）において設定した各要求事項は、以下の構成で記載している。

- **識別**
  要求事項を識別するために、「S」の後にカテゴリ番号、カテゴリ内の連番を「(1)-1」のように示す。

- **要求事項のタイトル、対象役割、概要、ライフサイクル上の該当箇所**
  各要求事項のタイトル、その要求事項が求められる役割、タイトルに対する概要説明を示す。
  下段には、上記「要求事項の概念図」のソフトウェアライフサイクルの中で、当該要求事項が該当するサイクルを「塗りつぶし」で示している。

- **個別要求**
  各要求事項に対して、対象者に求められる具体的な取組を促す個別要求の内容を示す。

（１）セキュアな設計・開発・供給・運用

S (1)-1	開発者	
	供給者	
	運用者	
	顧客	

設計時のリスク評価と対策の追跡

「セキュアバイデザイン」及び「セキュアバイデフォルト」の原則に則り、開発するソフトウェアのリスクを分析・評価し、リスク対応、セキュリティ要件、設計上の決定事項を追跡し、対策を維持する。

個別要求

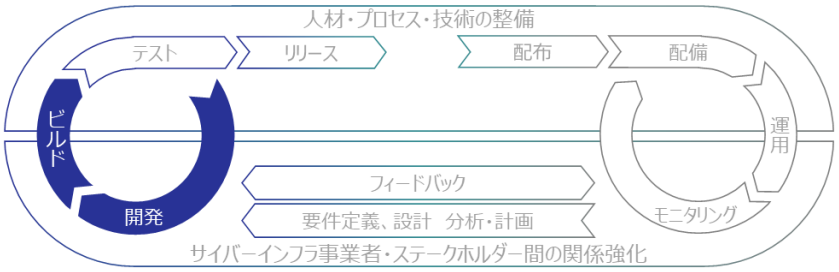
- ☐ **S(1)-1.1 リスクベースのセキュリティ要件の定義**
  開発するソフトウェア、あるいはソフトウェアで構成されるシステム・サービスに対して、リスクベースの分析・評価を実施し、緩和策となるセキュリティ要件を定義する。
- ☐ **S(1)-1.2 設計レビュー**
  ソフトウェアの設計のレビューを通じて、全てのセキュリティ要件を満たし、識別されたリスク情報に十分に対応していることを確認し、レビュー結果を反映する。
- ☐ **S(1)-1.3 リスク対応記録**
  設計上の決定事項、リスクへの対応、承認された例外措置に関する記録を保持し、ソフトウェアのライフサイクル全体を通じて監査や保守の目的で使用できるように維持する。
- ☐ **S(1)-1.4 リスクベースの定期的確認**
  セキュリティ要件に対して承認された全ての例外とソフトウェア設計、及びソフトウェアの設計時に作成したリスクベースの分析・評価結果をレビューし、リスクへの対処が適切か定期的に確認する。

S(1)-1 は、ソフトウェアの開発者に対して、セキュリティ要件を満たし、かつセキュリティリスクを軽減するように、ソフトウェアを設計することを求めている。

セキュリティ要件が既に特定されている場合、ソフトウェアの設計をレビューし、セキュリティ要件やリスクへの適合性を検証することは、ソフトウェアがセキュリティ要件を満たし、特定されたリスク情報に十分に対処できることを確認するのに役立つ。ソフトウェアの設計時からセキュリティ要件とリスクに対応すること（セキュアバイデザイン）、及びソフトウェアのセキュリティをデフォルトで組み込むこと（セキュアバイデフォルト）は、ソフトウェアのセキュリティを向上させるためのキーファクタであるとともに、開発効率の向上にもつながる。

ソフトウェアのセキュリティ要件を導出する必要がある場合は、リスクに基づく分析を行い、セキュリティ要件を特定・評価することを求めている。ソフトウェアの運用中に直面する可能性のあるセキュリティリスクと、

それらのリスクをソフトウェアの設計とアーキテクチャによってどのように軽減すべきかを判断する。また、リスクに基づく分析によって、セキュリティ要件を緩和又は免除すべきであると判断することで、その正当性を示すことができる。

S (1)-2	開発者 / 供給者 / 運用者 / 顧客	

セキュアなビルド

開発言語や開発環境に適したセキュアコーディング及びシステム構築のプロセスを定義し、これに従いコードを生成・ビルドする。設定を含むコードのレビュー及び分析を実施し、対応結果をプロセスにフィードバックする。

個別要求

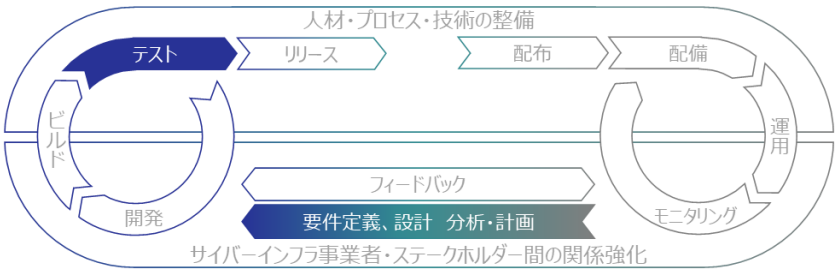
- ☐ **S(1)-2.1 セキュア開発プロセスの定義**
  セキュアコーディングの観点、ビルド実施タイミングと方式、自動化ツールの利用、トレーニングなど、セキュアコーディング、セキュアビルド及びデフォルトセキュアに関するプロセスを定義する。
- ☐ **S(1)-2.2 セキュアビルド**
  実行可能形式のセキュリティを向上させる機能を提供するコンパイラ、インタプリタ、及びビルドツールを使用し、コードを生成・ビルドする。
- ☐ **S(1)-2.3 検証とフィードバック**
  レビュー及び分析による検証により発見された問題の根本原因を特定し、その対応結果をプロセスにフィードバックする。
- ☐ **S(1)-2.4 コードベース**
  レビュー及び分析の対象は、ソースコードのみでなく、可読性があると組織が決定した様々な形式のコード（設定ファイル等）も対象とする。

S(1)-2 は、ソフトウェアの開発者に対して、ソフトウェアのコードベースをセキュアに生成し、かつセキュアにビルドすることを求めている。

セキュアコーディングの慣行を遵守して、ソースコードやセキュアな設定のコードベースを生成することで、ソフトウェアに含まれるセキュリティ脆弱性を低減する。また、コードベースの生成時に混入する脆弱性のうち、組織で定義された脆弱性許容度以下であることを確実にする、若しくはそれを超えるものを最小化するためのプロセスを適用することで、コストの削減にもつながる。実行可能形式のセキュリティを向上させるためには、コンパイル、リンク及びビルドプロセスを構築し、テスト実施前に脆弱性を排除することで、ソフト

ウェアのセキュリティ脆弱性を減少させ、コストを削減することにもつながる。コードをレビュー及び分析することで、セキュリティ要件への準拠を検証することができる。また、その過程で、脆弱性を特定することができた場合、ソフトウェアがリリースされる前に脆弱性を修正し、悪用されないようにすることにも役立つ。

これらのコードベースの生成やビルドの工程に自動化された手段を適用することで、脆弱性を検出するために必要な労力とリソースを削減することができる。

S (1)-3	開発者 / 供給者 / 運用者 / 顧客	

テスト

ビルドフェーズまでのレビュー及び分析で特定されなかった脆弱性を発見するために、機能テストに加え、脆弱性テスト、侵入テストを設計・実施し、発見された脆弱性への対策を実施する。

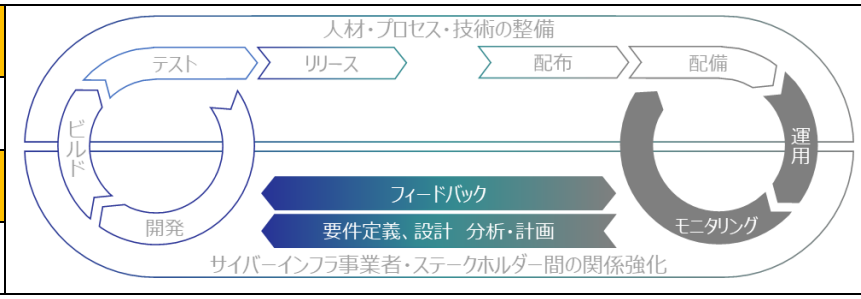
個別要求

- ☐ **S(1)-3.1 テスト計画**
  脅威モデルとリスク分析に基づき、テスト範囲及びテスト方式を決定し、テスト計画を立案する。
- ☐ **S(1)-3.2 テスト方式**
  テスト方式には、機能テスト、脆弱性テスト、ファジング、侵入テストなどを含める。
- ☐ **S(1)-3.3 テスト実施**
  テスト計画に従ってテストを設計、実施し、結果を文書化する。
- ☐ **S(1)-3.4 問題への対応**
  テストの結果、発見された全ての問題と推奨される対応策を開発チームのワークフローに組み込み、対処する。

S(1)-3 は、ソフトウェアの開発者に対して、テストによって脆弱性を発見し、対策することを求めている。

実行コードをテストすることで、セキュリティ要求事項への準拠を検証することができる。また、その過程で、脆弱性を特定することができた場合、ソフトウェアがリリースされる前に脆弱性を修正し、悪用されないようにすることにも役立つ。テストの工程に自動化された手段を適用し、実施形態に応じて適切な証跡や環境を整備することで、脆弱性の検出に必要な労力とリソースを低減し、追跡可能性と再現性を向上させることができる。なお、テスト方式は、対象とするソフトウェアが、自ら開発する製品やサービスなのか、受託開発するシステム・サービスなのか、あるいはその開発方法（ウォーターフォール開発、アジャイル開発）

などにより方針が異なる。リスクベースで定義されたセキュリティ要件、及び定義されたセキュア開発プロセスに基づいてテスト方式の方針を定め、テスト計画を立案する。

S (1)-4	開発者	
	供給者	
	運用者	
	顧客	

サービスのモニタリング

ソフトウェアがその導入環境（ネットワーク、プラットフォーム、サービスなど）と整合性をもって情報資産を保護、維持することをモニタリングするプロセス及びシステムを整備し、実施する。

個別要求

☐ **S(1)-4.1 資産管理**

運用者は、システム・サービスが扱う資産、及びシステム・サービスを構成する資産に関する資産管理手順と資産リストを整備する。

☐ **S(1)-4.2 モニタリング環境の整備**

運用者は、リスク発生時の潜在的な影響を最小化するためにシステムを適切に分離し、ソフトウェアによる資産保護上重要なリスクを監視するモニタリング環境を整備する。

☐ **S(1)-4.3 セキュリティメカニズムの整備**

ソフトウェア及びソフトウェアを適用するシステム・サービスが、動作環境又はデジタルインフラなどのリソース上にある情報資産及びデータの機密性・完全性を保護し、監視可能とするための適切なセキュリティメカニズムを整備する。

☐ **S(1)-4.4 モニタリングと評価**

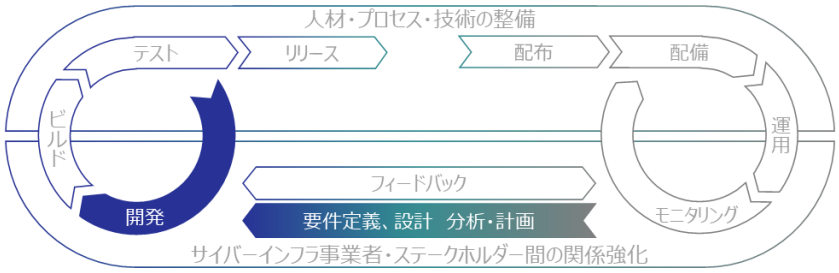
運用者は、重要なサービスを提供するソフトウェアに適用したメカニズムの動作状況をモニタリングするとともに、定期的にセキュリティ評価を実施し、組織のリスク管理の枠組みに統合する。

S(1)-4 は、ソフトウェアの運用者に対して、ソフトウェアによるサービスがセキュアな運用状態であるかどうかをモニタリングし、情報資産及びデータがサービスにより保護、維持されることを求めている。S(1)-4 の要求事項を満たすための運用（利用ソフトウェアのモニタリングシステムの整備、及びモニタリングと評価の支援など）は、ソフトウェア利用主体である顧客が実施するのが一般的であるが、システム・サービス又はこれらを構成するソフトウェアの運用には専門的な知識や技能が必要な場合を想定し、サイバーインフラ事業者が契約に基づいて実施する運用支援を想定する。

ソフトウェアによるシステム・サービスが扱う資産、及びシステム・サービスを構成する資産をリスト化して管理することで、インストール時及び運用時のソフトウェアのセキュリティを向上させ、ソフトウェアが脆弱なセキュリティ設定のまま導入及び運用され、危険にさらされる可能性を低減させる。

ソフトウェアの運用のためのセキュアな環境を導入し、維持することで、ソフトウェアの運用環境の全ての構成要素が、内部、外部の脅威から適切に保護されていることを確認し、環境又はその中で運用・維持されているソフトウェアの危殆化を防止することができる。また、動作状況をモニタリング及びセキュリティ評価することで、重要なサービスの運用におけるリスク管理にも効果が期待される。モニタリングする動作状況としては、ソフトウェアの持つ保護メカニズムが有効に働き、リソース上にある情報資産及びデータが保護されているか、意図的又は不用意にかかわらず、ソフトウェアの意図したセキュリティ特性が回避又は無効化されていないか、などを想定する。なお、セキュリティメカニズムを適切に設計・実装し、そのメカニズムの動作を監視可能とするためには、必要に応じて、開発者とも役割分担して対応することが望ましい。

（２）ライフサイクル管理、透明性の確保

S (2)-1	開発者	
	供給者	
	運用者	
	顧客	

セキュアなソフトウェアコンポーネントの手配

外部から手配した商用、オープンソース、その他のサードパーティのソフトウェアコンポーネントが、そのライフサイクルを通じて、組織が定義した要件に準拠していることを検証する。

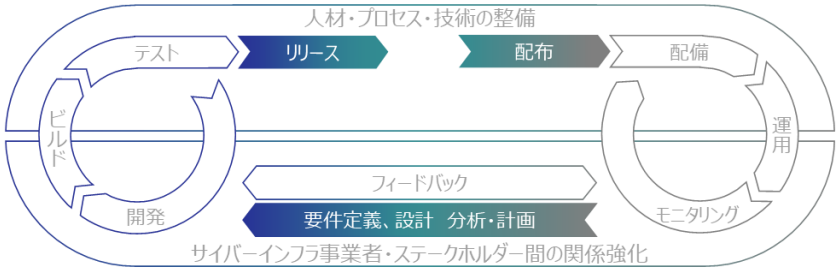
個別要求

- □ **S(2)-1.1 ソフトウェアコンポーネントの手配**
  外部から手配する商用、オープンソース、その他のサードパーティのソフトウェアコンポーネントは、組織が定義した要件を満たす安全性の高いものを採用する。
- □ **S(2)-1.2 ソフトウェアコンポーネントの開発・維持**
  外部からソフトウェアコンポーネントを手配しない場合、組織で確立されたセキュリティ基準・慣行に従い、安全性の高いソフトウェアコンポーネントを社内で開発、維持する。
- □ **S(2)-1.3 ソフトウェアコンポーネントのリスク評価**
  各ソフトウェアコンポーネントの出所情報を取得・分析し、そのコンポーネントがもたらすリスクを評価する。
- □ **S(2)-1.4 ソフトウェアコンポーネントの公知脆弱性の確認**
  各ソフトウェアコンポーネントの公知脆弱性、サポート期間を定期的にチェックする。
- □ **S(2)-1.5 ソフトウェアコンポーネントの更新**
  各ソフトウェアコンポーネントを新しいバージョンにセキュアに更新するプロセスを導入する。

S(2)-1 は、ソフトウェアの開発者に対して、サードパーティのソフトウェアコンポーネントを組織の要件に準拠するように扱うことを求めている。

できる限り開発する機能を重複させず、既存のセキュアなソフトウェアコンポーネントを利用する。セキュリティがチェック済みであり、脆弱性に対応する更新プロセスが適切に働くソフトウェアモジュールやサービスを再利用することで、ソフトウェア開発コストの削減、ソフトウェア開発の迅速化、ソフトウェアに新たなセキュリティ脆弱性を持ち込む可能性の低減を図ることができる。暗号モジュールやプロトコルなど、セキュリティ機能を実装するソフトウェアにおいては特に重要である。なお、公知脆弱性の確認においては公的機関が提供する脆弱性情報等も積極的に活用する。特に、サイバー対処能力強化法では、国は、重要電子計算機として用いられる電子計算機やプログラムにおける脆弱性を認知したときには、当該脆弱性に

関する情報や対応方法について、公表その他の適切な方法により周知することができることとされており、ソフトウェアの開発等に際して当該情報等を積極的に活用することが期待される。

S (2)-2	開発者 供給者 運用者 顧客	

リリースファイルやデータのセキュアなアーカイブ

ソフトウェアのリリースごとに保持すべき必要なファイルやデータをアーカイブし、必要な人員、ツール、サービスのみにアクセスを制限する。ソフトウェア部品表（SBOM）の段階的な採用などを通じて、各リリースの全てのコンポーネントについて、出所データを収集、保護、維持、共有する。

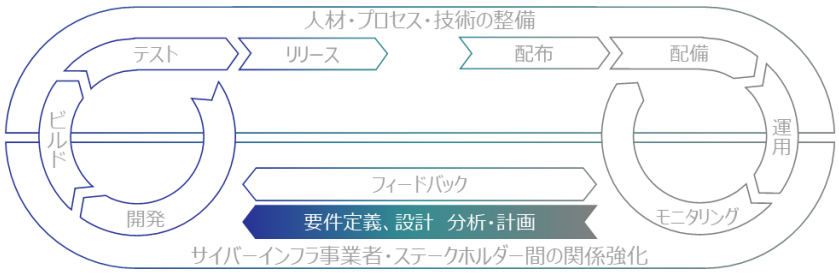
個別要求

- □ **S(2)-2.1 コードベースの保護**
  全ての形式のコードベースを不正アクセスや改ざんから保護するために、リポジトリにコードや設定情報を保管し、承認された担当者、ツール、サービスなどのみがアクセスできるよう最小権限の原則に基づいたアクセス制御を実施する。
- □ **S(2)-2.2 リリースのアーカイブ**
  リリース後に発見された脆弱性を分析、特定できるようにするために、各ソフトウェアのリリースをアーカイブ化して保護する。
- □ **S(2)-2.3 リリースの出所データの共有**
  各ソフトウェアリリースの全てのコンポーネントの出所データを収集、保護、維持、共有する。

S(2)-2 は、ソフトウェアの開発者及び供給者に対して、ソフトウェアのリリースごとに、ファイルやデータをセキュアにアーカイブして保護することを求めている。

あらゆる形態のコードベースを不正アクセスや改ざんから保護することで、意図的、不用意にかかわらず、ソフトウェアの意図されたセキュリティ特性を回避又は無効化するような、コードベースへの不正な変更を防ぐのに役立つ。公開することを意図していないコードについては、ソフトウェアの盗難を防ぐのに役立ち、攻撃者がソフトウェアの脆弱性を発見するのをより困難にすることができる。

各ソフトウェアのリリースをアーカイブ化し、保護することで、リリース後のソフトウェアに発見された脆弱性の特定、分析、除去などのアクションを支援することができる。なお、ソフトウェアのリリースごとに保持すべき必要なファイル及びサポートデータ（完全性検証情報、出所データなど）をセキュアにアーカイブ化し、関係者と共有可能とするためには、SBOM 等を用いたコンポーネントリストの生成・維持・共有に関するタスクなどが有効である。

S (2)-3	開発者 供給者 運用者 顧客	

関係者間のセキュリティ要件の確立

関係者間で合意すべきセキュリティ要件を確立し、契約又は共有するポリシーに盛り込む。

個別要求

□ **S(2)-3.1 セキュリティ要件の合意**

IT 製品（自社のソフトウェアで再利用するための商用ソフトウェアコンポーネントを含む）又はサービスを提供するサードパーティとの契約又は共有するポリシーに、明示的なセキュリティ要件を盛り込む。

□ **S(2)-3.2 サプライチェーンセキュリティ要求への対応**

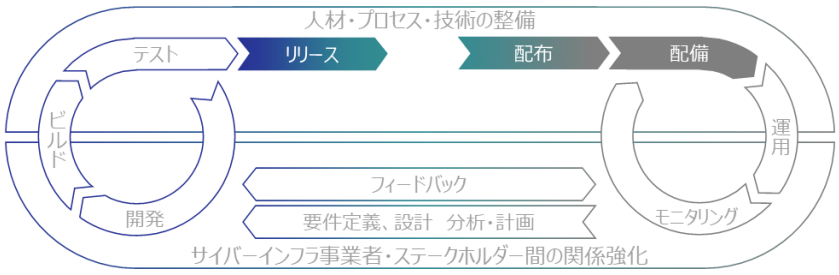
提供する IT 製品又はサービスを受領・取得する組織が採用するサプライチェーンセキュリティ要件と同等のサプライチェーンセキュリティ要件に対応する。

□ **S(2)-3.3 セキュリティ要件を満たさないリスクへの対処プロセスの整備**

受領・取得するサードパーティ製の IT 製品又はサービスが満たさないセキュリティ要件がある場合のリスクに対処するプロセスを整備する。

S(2)-3 は、ソフトウェアの開発者、供給者、及び運用者に対して、関係者間で共有すべきセキュリティ要件を確立することを求めている。

定義したソフトウェア開発及び運用のセキュリティ要件（サプライチェーンを対象に含む）を、サードパーティとの契約又は共有するポリシーにおいて明示化し、関係者が常に把握できるようにすることで、SDLC 全体を通じてセキュリティ要件（サプライチェーンを対象に含む）を考慮することができる。また、要件を完全かつ確実に共有することで、労力の重複を最小化することができる。なお、S(2)-3 の要求事項を満たすための運用（ソフトウェアの運用に必要な IT 製品やサービス等のセキュリティ要件の合意、及び関連するリスク対処プロセスの整備支援など）は、ソフトウェア利用主体である顧客が実施するのが一般的であるが、システム・サービス又はこれらを構成するソフトウェアの運用にはサードパーティが持つ専門的な知識や技能が必要な場合を想定し、サイバーインフラ事業者がサードパーティとの契約又は共有するポリシーに基づいて実施する運用支援を想定する。

S (2)-4	開発者 供給者 運用者 顧客	

利用者への適切な情報提供

ソフトウェアの導入・インストールから操作、利用終了までのライフサイクル全体でセキュアな利用を容易にするガイダンスをソフトウェア利用者が確実に利用できるようにする。

個別要求

☐ **S(2)-4.1 セキュアな導入・設定・操作・変更・廃棄・終了**

ソフトウェアをセキュアに導入・設定・操作するための情報、及び変更の影響・廃棄・提供終了・利用終了に係る情報をソフトウェア利用者が継続的に利用できるようにする。

☐ **S(2)-4.2 整合性検証情報の提供**

ソフトウェアの整合性・完全性の検証に必要な情報をソフトウェア利用者が継続的に利用できるようにする。

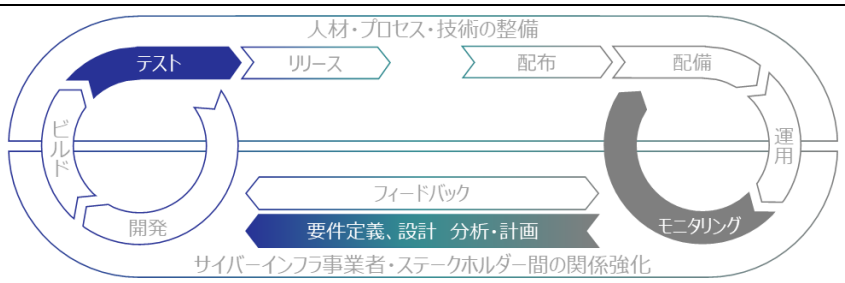
S(2)-4 は、ソフトウェアの開発者、及び供給者に対して、ソフトウェアのセキュアな利用方法を保証するための情報を利用者に提供することを求めている。

ソフトウェアをセキュアに導入・設定・操作するための情報を提供することで、インストール時にソフトウェアのセキュリティを向上させ、ソフトウェアが脆弱なセキュリティ設定のまま導入されてセキュアでない使用法によって操作されるなど、危険にさらされる可能性を低減させる。また、製品・サービスの販売終了（EOS: End of Sale）/製品・サービスのライフサイクル終了（保守やサポートの終了）（EOL: End of Life）に関する情報[12]、及び変更の影響や廃棄・提供終了・利用終了に関する情報を利用できるようにすることで、ソフトウェア利用者による資産管理やそのソフトウェアのセキュアな運用に役立つ。ソフトウェアの供給後も、こうした開発者による利用者への情報提供は、継続的になされる必要がある。

さらに、ソフトウェアのセキュアなデフォルト設定（又は該当する場合は、デフォルトコンフィギュレーションや相互に関連するデフォルト設定のグループ）を実装し、ソフトウェア管理者向けに各設定に関する情報を提供する。また、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ことで、ソフトウェア利用者が、取得したソフトウェアが正規のものであり、改ざんされていないことを確認するのに役立つ。

[12] ここでは、製品・サービスの販売終了を EOS、製品・サービスのライフサイクル終了を EOL と表している。ライフサイクル終了に類似する表現として、EOSL: End of Service Life、EOS: End of Support、EOS: End of Service、EOE: End of Engineering などが利用されることもある。

（３）残続する脆弱性の速やかな対処

S (3)-1	開発者 / 供給者 / 運用者 / 顧客	

継続的な脆弱性調査

ソフトウェアの脆弱性の開示と是正に関する方針を定め、その方針に必要な役割、責務、プロセスを定義し、実施する。

個別要求

□ **S(3)-1.1 脆弱性対応体制の設置**

ソフトウェア製品の脆弱性の開示と修復に対処するポリシーを定め、そのポリシーをサポートするための脆弱性対応（インシデント対応を含む）に関する体制を設置し、必要な役割、責務、プロセスを定義する。

□ **S(3)-1.2 コミュニケーション計画**

全ての利害関係者に対するコミュニケーション計画を定める。

□ **S(3)-1.3 脆弱性情報の収集**

公知情報の探索、ソフトウェア利用者からの通知、外部脅威情報の取得、システム構成データのレビュー、その他の方法を通じて、新たな脆弱性情報を収集する。

□ **S(3)-1.4 未検出の脆弱性の特定**

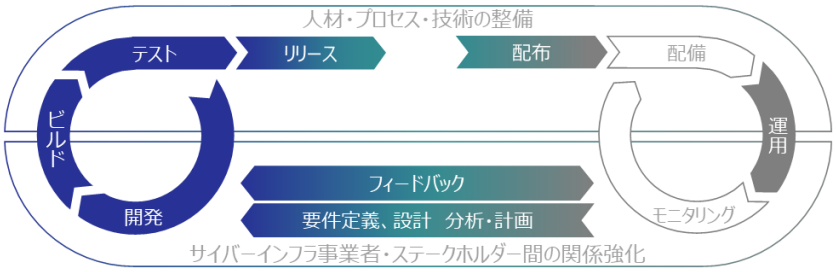
継続的又は定期的に、ソフトウェアのコードのレビュー、分析、テストを実施し、今まで未検出の対処すべき脆弱性（不適切な設定などを含む）を特定する。

S(3)-1 は、ソフトウェアの開発者、及び運用者に対して、ソフトウェアのインシデント対応を含む脆弱性対応に関する体制を整備し、脆弱性の開示と是正に関する方針に基づく継続的な脆弱性調査を求めている。特に開発者においては、継続的に自らが設計・開発を行ったソフトウェアの脆弱性に対応していくことが必要である。

なお、運用に関し S(3)-1 の要求事項を満たすための取組（利用ソフトウェアのインシデント対応の支援、利用ソフトウェアの脆弱性に関する情報収集の支援など）は、ソフトウェア利用主体である顧客が実施するのが一般的であるが、システム・サービス又はこれらを構成するソフトウェアの運用には専門的な知識や技能が必要な場合を想定し、サイバーインフラ事業者が契約に基づいて実施する運用支援を想定する。

脆弱性に対する継続的な把握と確認を実施することで、脆弱性をより迅速に特定し、リスクに応じて速やかに是正するなどの対応が可能となり、結果として攻撃者による攻撃機会の隙を減らすことに貢献する。ソフトウェアの開発者は、脆弱性の開示と是正に関する方針を定め、その方針に基づいて対応を推

進するために必要な役割、責務、プロセスを実施する。ソフトウェアの運用者は、ソフトウェア及びソフトウェアが使用するサードパーティ製コンポーネントに潜在する脆弱性の懸念について、ソフトウェア開発者に情報を提供する。

S (3)-2	開発者 供給者 運用者 顧客	

検知した脆弱性への対処

リリースしたソフトウェアに残存する脆弱性に対するリスク対応を定期的に計画し、実施する。

個別要求

☐ **S(3)-2.1 脆弱性の分析**

開発者は、残存する各脆弱性の影響に伴うリスクを把握するために必要な情報を収集し、修復又はその他のリスク対応を計画するために、各脆弱性を分析する。

☐ **S(3)-2.2 脆弱性へのリスク対応**

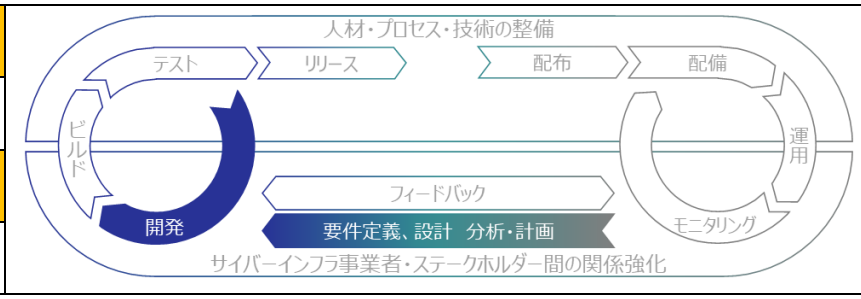
開発者は、各脆弱性に対するリスク対応を計画し、実装する。

☐ **S(3)-2.3 セキュリティ勧告**

開発者は、セキュリティ勧告を作成し、リリースしたソフトウェアの供給先にその情報を提供するとともに、関連する制度の指定に従って報告する。また、運用者はセキュリティ勧告に従った配備を実施する。

S(3)-2 は、ソフトウェアの開発者に対して、脆弱性の評価、優先順位付け及び修正を実施することを求めている。

各脆弱性を分析し、リスクに関する十分な情報を収集し、その是正又はその他のリスク対応を計画、及びソフトウェアの修正を実装することにより、リスクに応じて脆弱性を確実に是正し、攻撃者による攻撃機会の隙を減らすことを支援する。特に、例えばサイバー対処能力強化法では、国は、重要電子計算機として用いられる電子計算機やプログラムにおける脆弱性を認知したときには、当該電子計算機等の供給者に対し情報を提供することとされており、このような公的機関等から悪用された脆弱性等の情報提供があった際には、パッチ開発を含めて適正かつ積極的に対応することが求められる。なお、脆弱性が基幹インフラ事業者が使用する特定重要電子計算機に用いられる電子計算機等に関連するものである場合は、国は、当該電子計算機等の供給者に対し、サイバー攻撃による被害を防止するために必要な措置を講ずるよう要請することができることとされている点に留意する必要がある。また、セキュリティ勧告と是正対応済みのソフトウェアを供給先に提供し、それを適用することで、セキュアなソフトウェアの運用の維持につながる。

S (3)-3	開発者 供給者 運用者 顧客	

対処結果を組織のプロセス改善に活用

ソフトウェアに発見された問題の根本原因が再発しない、若しくはその可能性を低減するよう、脆弱性に基づき、開発と運用のプロセスを見直す。

個別要求

□ **S(3)-3.1 根本原因の特定**

根本原因を決定するために、識別された脆弱性を分析し、プロアクティブに対策する。

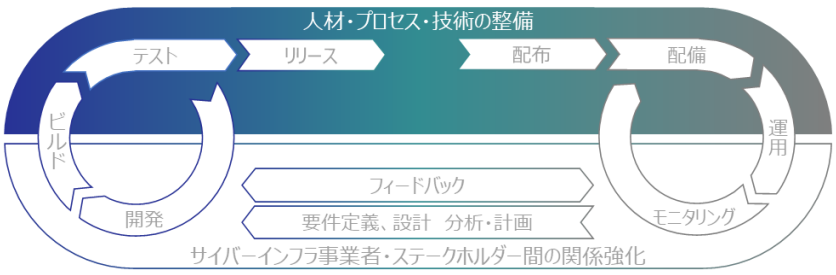
□ **S(3)-3.2 プロセス改善**

ソフトウェアの更新又は作成された新しいソフトウェアにより、根本原因の再発を防止又はその可能性を低減するために、ソフトウェアライフサイクル全体の開発と運用のプロセスをレビューし、必要に応じて見直す。

S(3)-3 は、ソフトウェアの開発者、及び運用者に対して、脆弱性を分析することで、その根本原因を特定し、対策することを求めている。なお、S(3)-3 の要求事項を満たすための運用（ソフトウェアの利用プロセスの改善や根本原因の分析に係る支援など）は、ソフトウェア利用主体である顧客が実施するのが一般的であるが、システム・サービス又はこれらを構成するソフトウェアの運用には専門的な知識や技能が必要な場合を想定し、サイバーインフラ事業者が契約に基づいて実施する運用支援を想定する。

特定した脆弱性を分析し、その根本原因を特定し対策することで、今後、脆弱性が発生する頻度を減らすことに貢献する。さらに、SDLC プロセスを見直し、ソフトウェアの更新や新規に作成されるソフトウェアに根本原因が再発しないよう（又はその可能性を減らすよう）に更新することで、根本原因の再発防止又はその可能性を低減し、結果として脆弱性の発生頻度を減らすことに貢献する。

（4）人材･プロセス･技術の整備

S (4)-1	開発者 供給者 運用者 顧客	

人材：経営層のコミットメントと人員の整備

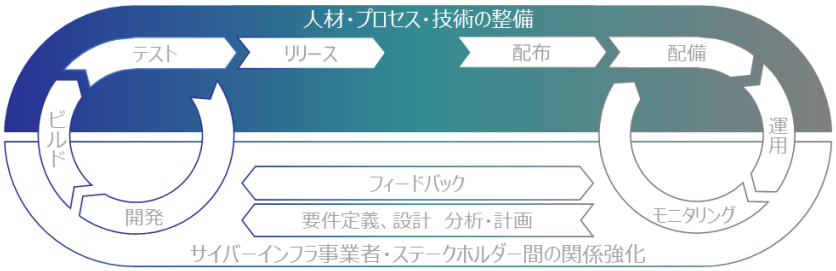
ソフトウェアのライフサイクル全体を網羅した役割と責務を定義する。セキュア開発に対する経営層のコミットメントを周知し、セキュリティ対策のための人材を確保し、セキュアな開発･運用に関連する全要員に、要員の習熟度と役割に応じたトレーニングを提供し、定期的に見直す。

個別要求

- ☐ **S(4)-1.1 役割と責務の定義**
  ソフトウェア開発ライフサイクルを網羅する役割と責務を定義する。
- ☐ **S(4)-1.2 経営層のコミットメント**
  全要員に対してセキュア開発に対する経営層のコミットメントを周知し、組織にとってのセキュアな開発･運用の重要性を教育する。
- ☐ **S(4)-1.3 役割と責務の同意**
  各要員が、役割と責務を認識･同意していることを確認する。
- ☐ **S(4)-1.4 各役割のトレーニング**
  各役割のトレーニング計画を作成し、全要員が習熟度と役割に応じてトレーニングを実施できるように提供する。
- ☐ **S(4)-1.5 役割とトレーニングの見直し**
  役割やトレーニングは定期的に見直す。

S(4)-1 は、ソフトウェアの開発者、供給者、及び運用者に対して、SDLC 全体に関わる人員への役割と責務を明らかにし、役割に応じた適切なトレーニングを実施することを求めている。なお、S(4)-1 の要求事項を満たすための運用（運用者の役割のトレーニングなど）は、ソフトウェア利用主体である顧客が実施するのが一般的であるが、システム･サービス又はこれらを構成するソフトウェアの運用には専門的な知識や技能が必要な場合を想定し、サイバーインフラ事業者が契約に基づいて運用支援を実施する場合を想定する。

ソフトウェア開発における役割と責務を明確化し、役割に応じたトレーニングを実施することで、SDLC に関わる組織内外の全員が、SDLC 全体を通じて SDLC に関連する役割と責務を果たすための準備が整う。また、役割と責務を定期的に見直し、及び要員の習熟度と役割に応じたトレーニングを定期的に見直し、必要に応じて更新することで、SDLC 全体のセキュリティ対応能力を維持することに貢献する。

S (4)-2	開発者 / 供給者 / 運用者 / 顧客	

プロセス：開発ポリシーの確立と法令順守

法令を遵守し、組織の開発インフラ及びプロセスに関するセキュリティポリシーを文書化・維持し、セキュリティ確保に必要な予算を確保する。

個別要求

□ **S(4)-2.1 ソフトウェア開発ポリシーの定義**

ソフトウェア開発のインフラ及びプロセスの全てのセキュリティ要件（EOL に係る要件を含む）を特定し、法令遵守の下 SDLC 全体を通じて維持するためのセキュリティポリシーを定義する。

□ **S(4)-2.2 ソフトウェア・セキュリティポリシーの定義と維持**

組織が開発するソフトウェアが満たすべき全てのセキュリティ要件を規定したポリシーを定義し、これらの要件を SDLC 全体にわたって維持する。

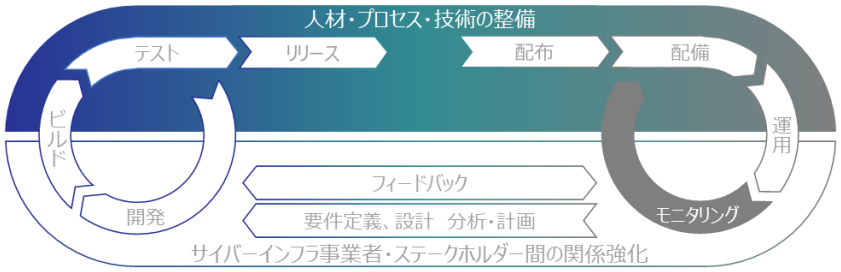
□ **S(4)-2.3 費用認識の共有と予算化**

ポリシーに基づいてセキュリティを確保するために必要な予算を確保する。

S(4)-2 は、ソフトウェアの開発者に対して、組織の開発インフラ及びプロセスに関するセキュリティポリシーを定め、SDLC 全体にわたって、法令を遵守した上で維持すること（予算確保を含む）を求めている。

ソフトウェア開発のインフラ及びプロセスにおけるセキュリティ要件、及びソフトウェアが満たすべきセキュリティ要件を特定する。SDLC 全体でこれらの要件を維持するためのポリシーを定義し、ソフトウェア開発のセキュリティ要件（EOL に係る要件を含む）を常に把握できるようにすることで、SDLC 全体を通じて考慮することができる。また、ソフトウェアの開発に関わる要件を共有することで、労力の重複を最小化することができる。また、セキュリティの確保に必要な予算の検討時において、ポリシーが関係者の理解共有の拠り所となる。

S (4)-3	開発者
	供給者
	運用者
	顧客



プロセス：運用ポリシーの確立と法令遵守

法令を遵守し、ソフトウェアを適用したサービス運用インフラ及びプロセスに関する全てのセキュリティポリシーを文書化し、維持する。

個別要求

☐ **S(4)-3.1 ソフトウェアサービス運用ポリシーの定義**

ソフトウェアを適用したサービス運用インフラ及びプロセスの全てのセキュリティ要件（EOS 及び廃棄に係る要件を含む）を特定し、法令遵守の下 SDLC 全体を通じて維持するためのセキュリティポリシーを定義する。

☐ **S(4)-3.2 サービスのセキュリティポリシーの定義と維持**

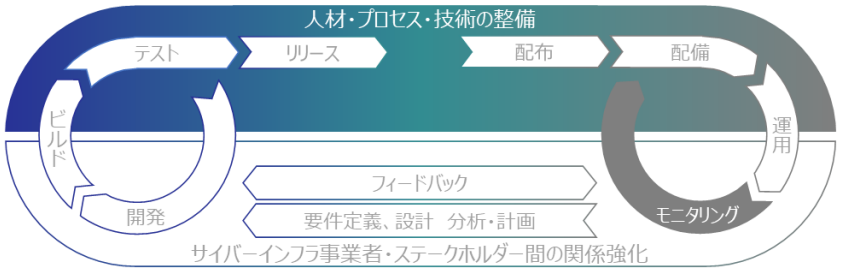
ソフトウェアを適用したサービスが満たすべき全てのセキュリティ要件を規定したポリシーを定義し、これらの要件を SDLC 全体にわたって維持する。

☐ **S(4)-3.3 運用ポリシーに基づく監査**

ポリシーに基づくガバナンスにより、サービス運用インフラ及びプロセスの保護、及びサービスのセキュリティ要件が SDLC 全体にわたって維持されていることを監査により確認する。

S(4)-3 は、ソフトウェアの運用者に対して、ソフトウェアの運用インフラ及びプロセスに関するセキュリティポリシーを定め、SDLC 全体にわたって、法令を遵守した上で維持すること（予算確保を含む）を求めている。なお、S(4)-3 の要求事項を満たすための運用（運用ポリシーの定義、維持、ポリシーに基づく監査支援など）は、ソフトウェア利用主体である顧客が実施するのが一般的であるが、システム・サービス又はこれらを構成するソフトウェアの運用には専門的な知識や技能が必要な場合を想定し、サイバーインフラ事業者が契約に基づいて実施する運用支援を想定する。

ソフトウェアを適用したサービス運用におけるセキュリティ要件、及びソフトウェアを適用したサービスが満たすべきセキュリティ要件を特定する。SDLC 全体でこれらの要件を維持するためのポリシーを定義し、ソフトウェア運用のセキュリティ要件（EOS 及び廃棄に係る要件を含む）を常に把握できるようにすることで、SDLC 全体を通じて考慮することができ、ソフトウェアの運用に関わる要件を共有できるため、労力の重複を最小化することができる。また、監査を実施することで、運用ポリシーに基づくガバナンス状況を把握し、SDLC 全体のセキュリティ要件の長期にわたる維持を実現する。

S (4)-4	開発者 / 供給者 / 運用者 / 顧客	

プロセス：開発・運用基準の策定

ソフトウェアの開発に関わるセキュリティ上の確認基準を定め、基準の裏付けに必要な情報を収集し、適合するためのプロセス、仕組みを実装する。ライフサイクル全体を通じて適合状況を追跡する。

個別要求

□ **S(4)-4.1 セキュリティ確認基準の定義と追跡**

ソフトウェアのセキュリティ確認基準を定義し、SDLC 全体を追跡する。

□ **S(4)-4.2 セキュリティ確認基準に基づく意思決定のサポート**

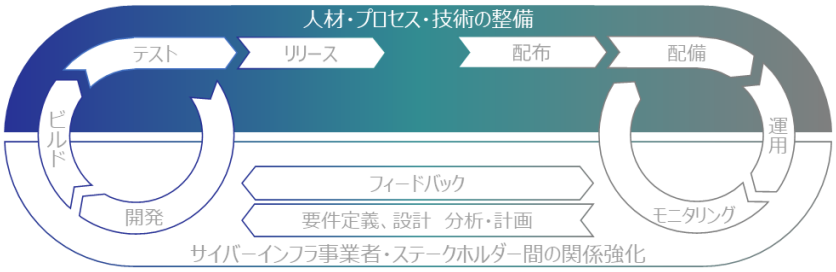
セキュリティ確認基準に基づく意思決定をサポートするために必要な情報を収集し保護するためのプロセスや仕組みなどを実装する。

□ **S(4)-4.3 セキュリティ確認基準に基づく監査**

セキュリティ上の確認基準への適合を遵守するためのガバナンスにより、SDLC 全体を追跡し意図する効果を得ていることを監査により確認する。

S(4)-4 は、ソフトウェアの開発者、及び運用者に対して、ソフトウェアのセキュリティを確認するための基準に基づき、情報を収集してその基準への適合状況を追跡することを求めている。なお、S(4)-4 の要求事項を満たすための運用（運用に関するセキュリティ確認基準に基づく意思決定支援や監査支援など）は、ソフトウェア利用主体である顧客が実施するのが一般的であるが、システム・サービス又はこれらを構成するソフトウェアの運用には専門的な知識や技能が必要な場合を想定し、サイバーインフラ事業者が契約に基づいて実施する運用支援を想定する。

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、セキュリティの実現状況を SDLC（ソフトウェア開発ライフサイクル）全体にわたって追跡することで、開発及び維持するソフトウェアのセキュリティをチェックする際の基準として使用することができる。この基準を満たすこと（保証）によりソフトウェアが SDLC から得られる組織の期待に継続して応えることを支援し、そのセキュリティを確保（保障）する。また、監査を実施することで、確認基準への適合遵守のためのガバナンス状況を把握し、SDLC 全体のセキュリティ水準の長期にわたる維持を実現する。

S (4)-5	開発者	
	供給者	
	運用者	
	顧客	

技術：セキュアな開発ツールの整備

ソフトウェアの開発ライフサイクル全体のリスクを分析し、開発ツールにセキュリティ対策を実施する。

個別要求

☐ **S(4)-5.1 ツールとツールチェーンの指定**

特定されたリスクを軽減するために有効なツールを特定し、どのツールチェーンに含めることが必須若しくは必要であるか、及びツールチェーンのコンポーネントを相互に統合する方法を指定する。

☐ **S(4)-5.2 ツールとツールチェーンの配備・運用・保守**

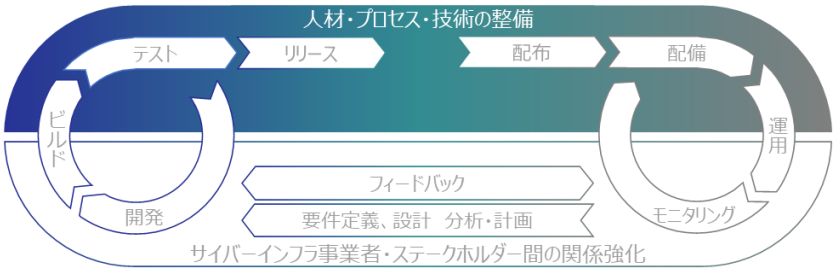
セキュリティ慣行に従ってツールとツールチェーンを配備、運用、及び保守する。

☐ **S(4)-5.3 ツール構成と証跡生成**

組織によって定義されたセキュアなソフトウェア開発の慣行のサポートに関する証跡を生成するようにツールを構成する。

S(4)-5 は、ソフトウェアの開発者に対して、ソフトウェアの開発ツールにセキュリティ対策を実施することを求めている。

ソフトウェア開発を支援するツールチェーンを使用して自動化を促進することで、人間の労力を削減することができる。また、これらの対策の使用を文書化して実証する方法を提供することで、SDLC 全体のセキュリティ対策の正確性、再現性、使いやすさ、及び包括性（開発の全体的なつながり具合）を向上させることができる。さらに、ツールチェーンとツールは、組織全体又はプロジェクト固有など組織の様々なレベルで使用することができる上、その一部はソフトウェア開発の実施状況の証跡の自動生成にも使用でき、SDLC の特定の部分の自動化対応のみならず、プロセス見直しのためのフィードバック効果にも貢献する。

S (4)-6	開発者	
	供給者	
	運用者	
	顧客	

技術：セキュアな開発環境の整備

ソフトウェアの開発ライフサイクル全体のリスクを分析し、開発に関わる環境を保護強化する。

個別要求

☐ **S(4)-6.1 環境の分離保護**

ソフトウェア開発に関係する各環境を分離して保護する。

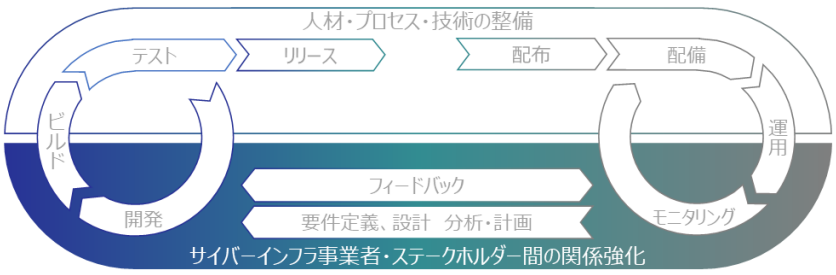
☐ **S(4)-6.2 開発用エンドポイントの保護**

リスクベースのアプローチを使用して開発関連のタスクを実行するために、各開発者向けのエンドポイントを保護、強化する。

S(4)-6 は、ソフトウェアの開発者に対して、セキュアな開発環境を整備することを求めている。

ソフトウェア開発のためにセキュアな開発環境を導入し、リスクベースのアプローチにより、開発関連のタスクを実行するための開発用エンドポイント（ソフトウェア設計者、開発者、テスターなどのためのエンドポイント）の保護された状態を維持することで、ソフトウェア開発環境の全ての構成要素が、内外の脅威から十分に保護されていることが確認でき、開発環境及びその中で開発・維持されているソフトウェアの危殆化を防止することに貢献する。

（５）サイバーインフラ事業者・ステークホルダー間の関係強化

S (5)-1	開発者 供給者 運用者 顧客	

情報連携のための組織体制

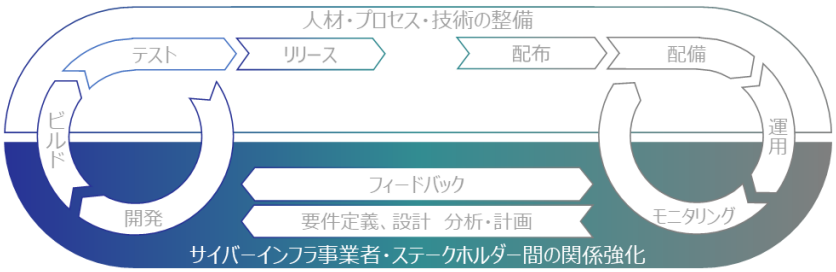
ソフトウェアの製品及びサービスのセキュリティを改善するために、民間企業同士、関係当局、専門組織との情報連携のための組織体制を構築する。

個別要求

- □ **S(5)-1.1 情報連携のための組織体制の構築**
  ソフトウェアの製品及びサービスのセキュリティを改善するために、民間企業同士、関係当局、専門組織との情報連携のための組織体制を構築する。
- □ **S(5)-1.2 重要なセキュリティ関連情報の提供**
  業界固有の必須かつ重要なセキュリティ関連情報を選別・識別して、サプライチェーン先に提供する。
- □ **S(5)-1.3 脆弱性情報の通知サービスの利用**
  効率的に脆弱性情報の共有を図るため、脆弱性情報の通知サービスを利用する。

S(5)-1 は、ソフトウェアの開発者、供給者、及び運用者に対して、ソフトウェアのセキュリティを改善することを目的とする情報連携のための組織体制を構築することを求めている。

ソフトウェアのセキュリティに関わる情報は、脆弱性に係る情報（攻撃による影響や対処に関する情報、被害事例など）、法的要求事項、業界のベストプラクティスなどがあり、これらの情報を取得・提供・共有するための組織体制を整備し、ステークホルダーとの情報連携のための関係を強化することで、ソフトウェアのセキュリティの継続的な改善に役立つ。

S (5)-2	開発者 供給者 運用者 顧客	

協力体制の強化

ソフトウェアの製品及びサービスのセキュリティを改善するために、民間企業同士、関係当局、専門組織との協力体制と枠組みを活用する。

個別要求

☐ **S(5)-2.1 協力体制の活用**

ソフトウェアの製品及びサービスのセキュリティを改善するために、外部の事業者、顧客、及び専門機関が参加するソフトウェアセキュリティの改善を目的とするコミュニティや協力体制を活用する。

☐ **S(5)-2.2 協力体制への貢献**

コミュニティや協力体制に参加する場合には、積極的に活動に関与し、協力体制に対して貢献する。

S(5)-2 は、ソフトウェアの開発者、供給者、及び運用者に対して、ソフトウェアのセキュリティを改善することを目的とする協力体制を活用することを求めている。

ソフトウェアのセキュリティを改善するためのコミュニティや協力体制に参加し、その活動に貢献することで、セキュリティの確保・維持・向上のための責務と役割に関する相互理解が深まり、セキュリティを改善するためのアクションの効果・効率が高まることで、ソフトウェアのセキュリティ確保とレジリエンス向上に役立つ。

（６）顧客によるリスク管理とセキュアなソフトウェアの調達・運用

S (6)-1	開発者	供給者	運用者	顧客
顧客経営層のリーダーシップによるリスク管理 顧客経営層のリーダーシップにより、顧客独自のリスク管理をサイバーインフラ事業者と協力して実施するリスク管理を統合する。				

個別要求

□ **S(6)-1.1 リスク管理**

顧客の独立した主体的な取組とサイバーインフラ事業者との契約に基づく取組を統合したリスク管理を実施する。

□ **S(6)-1.2 リソース整備**

既知の脆弱性への対処、及び緩和策を主体的に実施するためのリソースを割り当て、整備する（SBOM 活用を含む）。

□ **S(6)-1.3 協力体制の活用**

ソフトウェアセキュリティの改善を目的とするコミュニティや協力体制を活用する。

S(6)-1 は、顧客に対して、経営層のリーダーシップにより、顧客独自のリスク管理をサイバーインフラ事業者と協力して実施することを求めている。

顧客経営層のリーダーシップによりソフトウェアのセキュリティに関するリスク管理を推進することにより、顧客による主体的なソフトウェアのセキュリティ確保とレジリエンス向上につなげる。そのためには、取引先であるサイバーインフラ事業者とリスク対応の責務と役割を明確化し、契約により取り決めた手続に従って統合的にリスクを管理することが求められる。また、ソフトウェアの利用ライフサイクルを踏まえ既知脆弱性への対処と緩和策に必要なリソースの整備が求められる。また、ソフトウェアのセキュリティを改善するためのコミュニティや協力体制に参加し、その活動に貢献することで、セキュリティの確保・維持・向上のための責務と役割に関する相互理解が深まり、セキュリティを改善するためのアクションの効果・効率が高まることで、ソフトウェアのセキュリティ確保とレジリエンス向上に役立つ。

S (6)-2	開発者	供給者	運用者	顧客

顧客経営層のリーダーシップによるソフトウェアの調達、運用

顧客経営層のリーダーシップにより、セキュアにソフトウェアを調達、運用する。

個別要求

- ☐ **S(6)-2.1 セキュリティ要件の定義**
  ソフトウェア設計計画にセキュリティ機能を組み込むためのセキュリティ要件を定義し、ソフトウェアを調達・導入する前に、サイバーインフラ事業者に提示する。
- ☐ **S(6)-2.2 セキュリティ慣行の要求開示**
  ソフトウェアの調達・導入前に、サイバーインフラ事業者に求めるセキュリティ慣行の要求を開示する。
- ☐ **S(6)-2.3 リスク評価に基づく意思決定**
  ソフトウェアを調達・導入する際に、リスク評価に基づいた意思決定を行う。
- ☐ **S(6)-2.4 予算確保**
  ソフトウェアのライフサイクルを考慮した導入・運用・移行・廃棄、リスク対応、及び関連する契約に係る予算を継続的に確保する。

S(6)-2 は、顧客に対して、経営層のリーダーシップにより、ソフトウェアの調達及び運用をセキュアに実施することを求めている。

顧客経営層のリーダーシップの下、ソフトウェアを調達、運用する際には、これらの委託先候補となるサイバーインフラ事業者に対して、定義したセキュリティ要件とサイバーインフラ事業者に求めるセキュリティ慣行の要求を示すこと、適正なリスク評価に基づいてソフトウェアの調達・導入の意思決定を行うこと、及び適正かつ継続的なリスク対応を含むソフトウェアのライフサイクルを考慮した導入・運用・移行・廃棄の各フェーズに必要な予算を確保することにより、ソフトウェアのセキュリティ確保とレジリエンス向上につながる。なお、サイバーインフラ事業者に求めるセキュリティ慣行（サプライチェーンセキュリティ対策を含む）は、調達・導入するソフトウェアの特性やリスク対策の許容判断に基づいて設定する。

4. 要求事項の利活用

4.1. 要求事項の要求パッケージ化

サイバーインフラ事業者と顧客（政府機関等及び重要インフラ事業者をはじめとするソフトウェア製品・サービスの利用者）に求められる、ソフトウェアを対象としたサイバーセキュリティに関するレジリエンス向上の責務を果たすために取り組むべき要求事項は、その要求の目的・目標に応じて以下の２つに分類し、要求事項（個別要求）の要求パッケージとして利用することができる。

- **最低限要求パッケージ**
  全てのサイバーインフラ事業者及び顧客が最低限（ミニマム）実施すべき要求事項（個別要求）群。ソフトウェアの供給前及び運用中の脆弱性への対応、セキュアな調達、必要最小限の情報共有に限定したもの。

- **標準要求パッケージ**
  標準的に実施が求められる要求事項（個別要求）群。セキュアな開発体制の確立、リスク対応体制の確立、ステークホルダー間連携を含むもの。ソフトウェアが扱う情報やその保護の仕組みの維持、及び脆弱性への対応の即応性、確実性などの観点から、これらの欠如が対策すべきリスクと捉えられる場合は、標準要求パッケージを適用する。

要求事項と要求パッケージの対応関係を、表 5（サイバーインフラ事業者向け）及び表 6（顧客向け）に示す。

表 5 サイバーインフラ事業者に求められる要求事項と要求パッケージの対応関係

サイバーインフラ事業者に求められる要求事項（個別要求）	開発者	供給者	運用者	最低限要求パッケージ	標準要求パッケージ
S(1)-1.1 リスクベースのセキュリティ要件の定義	✓			○	○
S(1)-1.2 設計レビュー	✓			○	○
S(1)-1.3 リスク対応記録	✓				○
S(1)-1.4 リスクベースの定期的確認	✓				○
S(1)-2.1 セキュア開発プロセスの定義	✓			○	○
S(1)-2.2 セキュアビルド	✓			○	○
S(1)-2.3 検証とフィードバック	✓			○	○
S(1)-2.4 コードベース	✓			○	○
S(1)-3.1 テスト計画	✓			○	○
S(1)-3.2 テスト方式	✓			○	○

サイバーインフラ事業者に求められる要求事項（個別要求）	開発者	供給者	運用者	最低限要求パッケージ	標準要求パッケージ
S(1)-3.3 テスト実施	✓			○	○
S(1)-3.4 問題への対応	✓			○	○
S(1)-4.1 資産管理			✓	○	○
S(1)-4.2 モニタリング環境の整備			✓		○
S(1)-4.3 セキュリティメカニズムの整備	✓		✓		○
S(1)-4.4 モニタリングと評価			✓	○	○
S(2)-1.1 ソフトウェアコンポーネントの手配	✓			○	○
S(2)-1.2 ソフトウェアコンポーネントの開発・維持	✓			○	○
S(2)-1.3 ソフトウェアコンポーネントのリスク評価	✓			○	○
S(2)-1.4 ソフトウェアコンポーネントの公知脆弱性の確認	✓			○	○
S(2)-1.5 ソフトウェアコンポーネントの更新	✓			○	○
S(2)-2.1 コードベースの保護	✓	✓		○	○
S(2)-2.2 リリースのアーカイブ	✓	✓		○	○
S(2)-2.3 リリースの出所データの共有	✓	✓		○	○
S(2)-3.1 セキュリティ要件の合意	✓	✓	✓	○	○
S(2)-3.2 サプライチェーンセキュリティ要求への対応	✓	✓			○
S(2)-3.3 セキュリティ要件を満たさないリスクへの対処プロセスの整備	✓	✓	✓		○
S(2)-4.1 セキュアな導入・設定・操作・変更・廃棄・終了	✓	✓		○	○
S(2)-4.2 整合性検証情報の提供	✓	✓		○	○
S(3)-1.1 脆弱性対応体制の設置	✓		✓	○	○
S(3)-1.2 コミュニケーション計画	✓		✓	○	○
S(3)-1.3 脆弱性情報の収集	✓		✓	○	○
S(3)-1.4 未検出の脆弱性の特定	✓		✓	○	○
S(3)-2.1 脆弱性の分析	✓			○	○
S(3)-2.2 脆弱性へのリスク対応	✓			○	○
S(3)-2.3 セキュリティ勧告	✓	✓	✓	○	○
S(3)-3.1 根本原因の特定	✓		✓		○
S(3)-3.2 プロセス改善	✓		✓		○
S(4)-1.1 役割と責務の定義	✓	✓	✓		○
S(4)-1.2 経営層のコミットメント	✓	✓	✓	○	○
S(4)-1.3 役割と責務の同意	✓	✓	✓		○
S(4)-1.4 各役割のトレーニング	✓	✓	✓		○
S(4)-1.5 役割とトレーニングの見直し	✓	✓	✓		○
S(4)-2.1 ソフトウェア開発ポリシーの定義	✓			○	○

サイバーインフラ事業者に求められる要求事項（個別要求）	開発者	供給者	運用者	最低限要求パッケージ	標準要求パッケージ
S(4)-2.2 ソフトウェア・セキュリティポリシーの定義と維持	✓			○	○
S(4)-2.3 費用認識の共有と予算化	✓			○	○
S(4)-3.1 ソフトウェアサービス運用ポリシーの定義			✓		○
S(4)-3.2 サービス・セキュリティポリシーの定義と維持			✓		○
S(4)-3.3 運用ポリシーに基づく監査			✓		○
S(4)-4.1 セキュリティ確認基準の定義と追跡	✓		✓	○	○
S(4)-4.2 セキュリティ確認基準に基づく意思決定のサポート	✓		✓	○	○
S(4)-4.3 セキュリティ確認基準に基づく監査	✓		✓		○
S(4)-5.1 ツールとツールチェーンの指定	✓			○	○
S(4)-5.2 ツールとツールチェーンの配備・運用・保守	✓			○	○
S(4)-5.3 ツール構成と証跡生成	✓			○	○
S(4)-6.1 環境の分離保護	✓			○	○
S(4)-6.2 開発用エンドポイントの保護	✓			○	○
S(5)-1.1 情報連携のための組織体制の構築	✓	✓	✓		○
S(5)-1.2 重要なセキュリティ関連情報の提供	✓	✓	✓	○	○
S(5)-1.3 脆弱性情報の通知サービスの利用	✓	✓	✓	○	○
S(5)-2.1 協力体制の活用	✓	✓	✓		○
S(5)-2.2 協力体制への貢献	✓	✓	✓		○

表 6 顧客に求められる要求事項と要求パッケージの対応関係

顧客に求められる要求事項（個別要求）	最低限要求パッケージ	標準要求パッケージ
S(6)-1.1 リスク管理	○	○
S(6)-1.2 リソース整備	○	○
S(6)-1.3 協力体制の活用		○
S(6)-2.1 セキュリティ要件の定義	○	○
S(6)-2.2 セキュリティ慣行の要求開示	○	○
S(6)-2.3 リスク評価に基づく意思決定	○	○
S(6)-2.4 予算確保	○	○

4.2. 役割分担に応じた要求事項の適用に関する注意点

サイバーインフラ事業者の役割分担と要求事項の関係に基づき、対応すべき要求事項の基本的な設定方法、及び対応すべき要求事項を適切に適用するための注意点を以下に示す。

- サイバーインフラ事業者の役割分担に応じた要求事項の基本的な設定方法

サイバーインフラ事業者は、対象とするソフトウェアに対する組織としての担当の役割範囲（開発者、供給者、運用者の各役割の有無）を明らかにし、求められる要求事項の達成度合い（要求パッケージの標準、最低限）を定め、役割に対する個別要求を満たすサイバーセキュリティ対策を検討し、実施することが期待される。一般的に、サプライチェーン全体（ソフトウェアコンポーネントの調達先、及びソフトウェア開発の委託先（開発委託の末端まで）を含む）で同一の達成度合い（要求パッケージの標準、最低限）を定めることが望ましいことから、求められる要求事項の達成度合いは、顧客自身が定める達成度合いを満たすために、サイバーインフラ事業者の役割範囲として整合するように設定する。主体的な立場で役割を担当する場合、設定された達成度合い（要求パッケージの標準、最低限）の全ての要求事項を満たすことが求められる。支援的な立場で役割を担当する場合でも、その役割の主体者と同等の達成度合い及び要求事項を満たすことが望ましいが、支援する範囲の責務に応じて、主体者に割り当てられた要求事項に限定した対応とすることも許容されるものとする。

- 顧客による CSIRT 設置

システム開発が完了し、顧客受入れを経てシステムが運用フェーズに入ると、顧客がシステムインシデント対応の CSIRT を設置し、ソフトウェアの脆弱性対応を顧客が主導し、一部のソフトウェアの脆弱性対応の実働をサイバーインフラ事業者に委託する場合がある。このようなケースでは、顧客の運用部門が要求事項 S(3) を運用者として実施し、一部のソフトウェアの脆弱性対応を対象とする要求事項 S(3) の運用を委託先のサイバーインフラ事業者が運用者として実施するものと考えられる。

- 顧客によるコード生成ツールの適用

サイバーインフラ事業者が提供する開発用コード生成ツールを利用して、顧客がコードを生成するノーコードプラットフォームを用いた開発において、サイバーインフラ事業者が想定していないプログラムが生成される恐れがある。このような手順で生成されたソフトウェアの場合、コード生成ツールは開発者を代行して開発行為の一部を代行するものと考え、顧客仕様に基づくソフトウェアの適切な動作を保証するためのテストは顧客自身（顧客の開発部門など）が実施する位置付けと考えられる。この場合、サイバーインフラ事業者は、コード生成ツール（又はこれを含むノーコードプラットフォーム）をソフトウェア製品としてその開発者・供給者としての役割を担当し、このツールを適用して生成するソフトウェアの開発者としての責務は顧客自身（顧客の開発部門など）の役割を担当するなど、個別に役割分担を整理することが望ましい。

5. 参考情報

5.1. 要求事項チェックリスト

要求事項に関する情報を一覧で確認できるチェックリスト 別紙「要求事項チェックリスト」及び「要求事項チェックリスト（役割・フェーズ）」を参照のこと。

5.2. セキュリティインシデントと要求事項との対応関係例

社会に大きな影響をもたらしたセキュリティインシデントの事例に対して、本ガイドライン（案）で整理した要求事項がどのようにリスクを軽減するのか、その対応関係を参考情報として以下に示す。

■ Apache Log4J の脆弱性

Apache Log4J は、ログを出力するソフトウェアライブラリであり、世界中で利用されている。2021年に遠隔から任意の処理を実行できるという深刻な脆弱性が発見され、悪用された事例である。多層のサプライチェーンの中で、様々なソフトウェアに組み込まれていることから、脆弱性を簡単には発見、追跡、改修できないことが要因である。長期にわたり脆弱性が残存しているケースもある。

本インシデントでは、要求事項 S(3) の脆弱性情報を収集し対処すること、要求事項 S(5) の情報収集体制の整備を通じて、脆弱性情報を把握・対処することでリスクを軽減できる。また、脆弱性情報が公開されて以降にソフトウェア開発を開始するケースであっても、要求事項 S(2) の適切なソフトウェアの採用を通じて、脆弱性が残存するソフトウェアの利用を排除できる。

■ ソフトウェアベンダーA 社のインシデント

正規のソフトウェアアップデートが改ざんされたことで、本ソフトウェアの利用組織全体に影響が及んだ事例である。ソフトウェア開発企業の開発・運用環境に侵入されることで、ソフトウェアアップデートを改ざんされており、ソフトウェアサプライチェーンの上流から下流までの開発・運用環境のセキュリティが十分に確保されていないことが要因である。

本インシデントは、要求事項 S(1) と S(4) のセキュアな開発・運用環境の維持を通じて、攻撃者の侵入を困難にすることでリスクを軽減できる。

■ B 病院が保有する患者情報の暗号化・漏えいインシデント

VPN 装置の脆弱性を悪用し、攻撃者に院内ネットワークへ侵入され、病院が保有する患者情報が暗号化・漏えいし、診療業務に支障がでた事例であり、VPN 装置の脆弱性情報が放置されていたことが要因である。

本インシデントは、要求事項 S(3) の脆弱性情報を収集し対処すること、要求事項 S(5) の情報収集体制の整備を通じて、脆弱性情報を把握・対処することでリスクを軽減できる。また、顧客側も、要求事項 S(6) のセキュアなソフトウェアの調達・運用を通じて、事業者と協力することでリスクを低減できる。

5.3. システムライフサイクルにおける脅威と要求事項の対応関係

システムライフサイクル上の脅威に対抗するために本ガイドライン（案）で整理した要求事項が必要である理由とともに、脅威と要求事項の主な対応関係を参考情報として表 7 に示す。

表 7 システムライフサイクルと脅威及び要求事項の対応関係

システムライフサイクル	脅威の概要	脅威に対抗するために要求事項が必要な理由
分析・計画	• **現状分析の不足** 顧客と開発・供給者間で現状のシステム・サービスのリスク分析が十分になされないまま、現状の脆弱性やセキュリティを考慮しないシステム・サービスが構築されてしまう。	事業者は、S(1)-1 により、リスク分析により適切にセキュリティ要件を定義し、その分析結果を確認する。 顧客は、S(6)-2 により、リスク評価に基づいて意思決定を行う。 これらにより、適切な現状分析を実施する。
要件定義	• **要件の不合意** 顧客及び事業者間でセキュリティ要件の合意が十分になされておらず、意図しないセキュリティ要件が定義されてしまう。セキュリティ要件の誤解、欠如を含む。	事業者は、S(1)-1 により、適切なセキュリティ要件を定め、S(2)-3 により、セキュリティ要件を事業者間で合意するとともに、要件を満たさないリスクにも対処する。 顧客は、S(6)-2 により、セキュリティ要件の定義に主体的に関与する。 これらにより、顧客と事業者間で適切なセキュリティ要件を合意する。
設計～テスト	• **要件の不理解・不適切な実装** ソフトウェアのセキュリティ品質の観点から、セキュリティ要件が完全に理解されていない、若しくは適切に実装されない。	事業者は、S(1)-1 により、リスク対応を記録し、リスク対応策を継続的にレビューすることで、適切なリスク対応を維持する。また、S(1)-2 により、セキュア開発プロセスを活用し適切な実装を行う。 これらにより、要件を適切に実装する。
	• **意図的なコード操作** セキュアでない開発環境を悪用し、攻撃者によって将来の不正アクセスを可能にするバックドア等の悪意のあるコードやコンポーネントを意図的に注入される。あるいは、攻撃者によってソースコード等の機密情報が窃取される。	事業者は、S(2)-2 により、開発環境のコードにアクセス制御を実施する。S(4)-5 と S(4)-6 により、開発ツールにセキュリティ対策を実施し、開発環境を保護する。 これらにより、開発環境の保護を通じて攻撃者からコードを保護する。
	• **不正なサードパーティソフトの組み込み** 脆弱性のあるサードパーティのソースコードやバイナリ、出所が不明のソフトウェア又はコンポーネントが、意図的又は偶発的に組み込まれてしまう。	事業者は、S(2)-1 により、セキュアなソフトウェアコンポーネントを調達する。S(2)-2 により、各ソフトウェアリリースのコンポーネントの出所を管理する。 これらにより、不適切なサードパーティソフトの組み込みを回避する。

システムライフサイクル	脅威の概要	脅威に対抗するために要求事項が必要な理由
設計～テスト	• **ビルド時の不正組み込み** 攻撃者によって、ビルドプロセス内の欠陥が悪用され、製品のコンポーネントに不正なソフトウェアが組み込まれてしまう。 （例：不適切なコンパイラオプション）	事業者は、S(1)-2 により、セキュアなビルドツールを利用しビルドを実施する。S(4)-5 により、ビルド環境の開発ツールにセキュリティ対策を実施する。 これらにより、ビルド時の不正組み込みを回避する。
	• **正確性・再現性の低い属人的な開発プロセス** PDCA 手順が順守されず、熟練していない個人作業に過度に依存した開発（実装）プロセスにより、正確性や再現性が低下しビルド環境におけるセキュリティ上の潜在的な問題が発生する。 （例：個人依存のローカルビルド）	事業者は、S(1)-2 により、セキュアなビルドツールを利用しビルドを実施する。S(4)-5 により、適切な開発ツールチェーンを利用する。 これらにより、属人的な開発作業を回避する。
	• **レビュー・分析の漏れ** 脆弱性の特定や基準適合のためのコードのレビューや分析が十分なされていないため、脆弱性が残存する。 （例：脆弱性テストやスキャンの不足）	事業者は、S(1)-1 により、設計時にレビューを行う。S(1)-3 によりテストを実施する。S(1)-2 により、様々な形式のコードをレビューし、プロセスにフィードバックする。 これらにより、適切な範囲でレビュー・分析を実施する。
	• **不適切な開発プロセス** PDCA 手順が確立しておらず、低品質な開発プロセスを採用している。 また、市場投入までの時間短縮、コスト削減等を優先する余り、新たな開発プロセス、アプローチが無理に採用されることで、セキュリティ上の潜在的な問題が発生する。 （例：脆弱な開発基準）	事業者は、S(4)-2 により、開発ポリシーを整備し、S(4)-4 により、開発基準を整備する。 これらにより、セキュアな開発プロセスを維持する。
	• **意図的でない情報漏えい** 意図せず、情報が漏えいする。 （例：開発者の不注意や、不適切な開発環境等）	事業者は、S(4)-1 により、トレーニング等を通じた要員のスキル向上を図る。S(4)-2 により、セキュアなソフトウェア開発インフラを整備する。S(4)-3 により、ソフトウェアサービス運用ポリシーを整備する。S(4)-5 と S(4)-6 により、開発ツールにセキュリティ対策を実施し開発環境を保護する。 これらにより、人的面及び環境面での情報漏えいを低減する。
	• **不適切なサービス利用（※クラウドのみ）** （スケジュールやコストの制約から）セキュリティを考慮しない SaaS サービスが導入若しくは実装されてしまう。 （例：単一の SaaS ソリューション内の異なるモジュール間でセキュリティ要件が異なるため全コンポーネントの検証が不十分になる）	事業者は、S(2)-1 により、適切なソフトウェアコンポーネントからなるサービスを導入する。S(2)-3 により、セキュリティ要件を満たさないリスクへの対処プロセスを整備する。 これらにより、適切なサービスを導入・利用する。

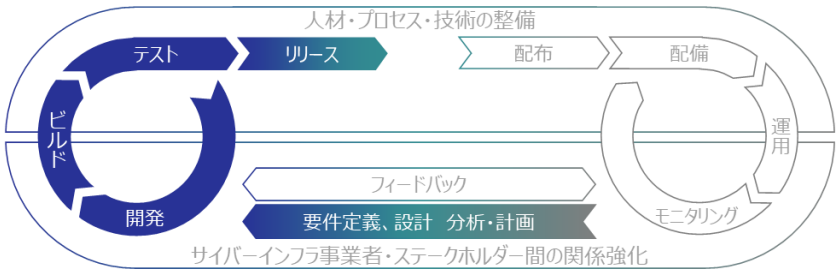
システムライフサイクル	脅威の概要	脅威に対抗するために要求事項が必要な理由
配布	• **配布時の不正組み込み** ソフトウェアの配布経路・配信メカニズムにおいて、顧客に配布されたオリジナルのソフトウェアパッケージ、更新プログラム、アップグレード製品内に悪意のあるソフトウェアが注入される。（例：プログラムの改ざん、ドキュメントの改ざん）	事業者は、S(2)-4 により、利用者がセキュアにソフトウェアの利用を開始するための情報と仕組みを提供する。 これにより、配布時の不正を回避する。
運用	• **サービス妨害** 攻撃者等によって、SaaS 等の外部サービスが停止させられる、セキュリティパッチ等の供給が停止させられる。	事業者は、S(4)-3 により、ソフトウェアサービス運用ポリシーを整備する。S(1)-4 により、サービス運用のモニタリング環境を整備する。S(3)-2 により、脆弱性のリスク対応を実施する。 これらにより、要因となる脆弱性に対処するとともに、サービス妨害を検知しその影響を低減する。
	• **アーカイブの不正操作** 開発者により意図せず、あるいは攻撃者により意図して、アーカイブが不正操作、上書き・破壊される。リリース後に発見された脆弱性の分析と対処が困難になる。	事業者は、S(2)-2 により、アーカイブを保護する。 これにより、アーカイブの不正操作を回避する。
	• **脆弱性の放置** 発見されたソフトウェアの脆弱性情報が顧客に伝わらないまま、ソフトウェアの利用が継続されてしまう。	事業者は、S(3)-1 により、脆弱性の開示方針を定める。S(2)-4 により、利用者に脆弱性対策に必要な情報を提供する。S(3)-2 と S(3)-3 により、根本原因を含めて脆弱性に対処する。S(2)-1 によりソフトウェアコンポーネントの更新プロセスを導入する。 これらにより、顧客に適切な情報と脆弱性の対応策を提供することで、脆弱性の解消を図る。
	• **構成・設定不全** 適切な構成や設定を完了しないまま、ソフトウェアが使用されてしまうことで、脆弱性が顕在化してしまう。 （例：デフォルトでフルアクセスを許可、真正性が確認できないソフトウェアの実行 等）	事業者は、S(1)-2 により、デフォルトセキュアなソフトウェア開発を推進する。S(2)-4 により、利用者にセキュアな構成・利用方法を提供する。 これらにより、適切に構成されたソフトウェアの利用を進める。
廃棄	• **廃棄物を介した情報漏えい・不正アクセス** 機密情報が残存する廃棄機材を通じて、攻撃者によってソースコード等の機密情報が窃取される、残置された廃棄予定の機器を介して不正アクセスされる。	事業者は、S(4)-2、S(4)-3 により、ソフトウェアサービス運用ポリシーを整備する。 これにより、適切に廃棄対象機材を処分する。

システムライフサイクル	脅威の概要	脅威に対抗するために要求事項が必要な理由
ライフサイクル共通	• **プロセス・リソースの未整備** プロセス・リソース（ヒト・モノ・カネ）が整備されていないため、セキュアなソフトサイクルやソフトウェアサプライチェーンが維持されない。 （例：脅威又はリスクの評価を実施するための訓練が不十分である、セキュリティが経営課題となっていない、過剰な業務量等）	事業者は、S(4)-1 により、人材を整備する。S(4)-2 により開発ポリシーを整備する。S(4)-3 により、ソフトウェアサービス運用ポリシーを整備する。S(4)-4 により、プロセスとリソース体制を監査する。顧客は、S(6)-1 により、事業者の体制を適切に判断する。S(6)-2 により、適切な予算を確保する。 これらにより、開発インフラ、開発プロセスのセキュリティ要件や基準といった、セキュリティ確保に必要なヒト・モノ・カネの整備をはかる。
	• **情報・資産管理の不備** 必要な情報が把握されていないため、セキュリティの取組に着手できていない。	事業者は、S(5)-1 により、情報連携を進める。S(5)-2 により、協力体制の活用を進める。 これらにより、セキュリティ対策に必要な情報の収集・管理を進める。
	• **不完全な合意連鎖** 供給者、サードパーティの供給者、及び顧客間のセキュリティに関わる契約合意事項が適切に連鎖していないため、セキュリティ要件が満たされない、若しくは要求を拒否される。	事業者は、S(2)-3 により、関係者間で適切なセキュリティ要件の合意を図る。S(1)-1 により、リスク対応を維持する。 顧客は、S(6)-1 により、主体的にリスク管理を行う。S(6)-2 により、顧客は事業者に求めるセキュリティ慣行をあらかじめ開示する。 これらにより、関係者間で適切なセキュリティ要件を合意する。
	• **選定条件の不備** 供給者（再委託先）及びソフトウェアの選定時に、供給者の特性を考慮されない。 （例：過去の実績等）	事業者は、S(2)-3 により、関係者間でセキュリティ要件を合意し、契約に盛り込む。 これにより、適切な供給者を選定する。

5.4. 要求事項に対する取組例

要求事項（個別要求）を実現するために対応すべき実施事項の例を参考情報として以下に示す。要求事項を満たすために、対象ソフトウェアに対して組織として実現すべき手段は、組織に適した方法を選択・適用することになる。これらの取組例は、そのような手段の候補がイメージできるように参考情報として示している。したがって要求事項（個別要求）に対して対応すべき取組を網羅的に示したものではないことに留意すること。要求事項を実現する手段を検討する際には、必要に応じて 5.5 などを参考とすることを推奨する。

（1）セキュアな設計・開発・供給・運用

<table>
<tr><td rowspan="4">S (1)-1</td><td>開発者</td><td rowspan="4"> 
</td></tr>
<tr><td>供給者</td></tr>
<tr><td>運用者</td></tr>
<tr><td>顧客</td></tr>
<tr><td colspan="3">設計時のリスク評価と対策の追跡
「セキュアバイデザイン」及び「セキュアバイデフォルト」の原則に則り、開発するソフトウェアのリスクを分析・評価し、リスク対応、セキュリティ要件、設計上の決定事項を追跡し、対策を維持する。</td></tr>
<tr><td colspan="3">□　S(1)-1.1 リスクベースのセキュリティ要件の定義
開発するソフトウェア、あるいはソフトウェアで構成されるシステム・サービスに対して、リスクベースの分析・評価を実施し、緩和策となるセキュリティ要件を定義する。</td></tr>
<tr><td>取組例</td><td colspan="2">• リスクベースのアプローチの適用効果を高めるために、攻撃モデリング、脅威モデリングなどのリスクモデリングを使用したリスク分析手法によるリスク評価を実施する。
• リスクベースのアプローチの適用効果を高めるために、開発チームをトレーニングするか、リスクモデリングの専門家の協力を得る。
• 守るべき機密データや個人情報、認証、アクセス制御、クレデンシャル管理など、リスクの高い領域に対してより厳密にリスク評価を実施する。

（システム・サービスのソフトウェアの場合）
• 顧客と合意すべきセキュリティ要件を全社ルールとして事前に定める。要件の提示がない顧客に対しては、ヒアリングを通じて適切なセキュリティ要件を合意する。
• セキュリティ対策に伴う費用に関する顧客の理解を得るために、顧客のセキュリティ向上による効果を提案する。その際、増加費用の必要性を明細に基づいて顧客に説明する。
• 事業者と顧客間の役割分担、開発環境や用語の整備による業界内の共通化・標準化、コミュニケーションの方法を含む、システムライフサイクル全体の協力体制を構築する。</td></tr>
</table>

□ **S(1)-1.2 設計レビュー** ソフトウェアの設計のレビューを通じて、全てのセキュリティ要件を満たし、識別されたリスク情報に十分に対応していることを確認し、レビュー結果を反映する。	
取組例	• ソフトウェアを設計の観点（アーキテクチャ、設計、重要コード、脆弱性など）から、それぞれに適した方法（ピアレビュー、リードレビュー、ウォークスルーなど）でレビューする。 • ソフトウェアを開発の様々な観点（統合開発環境（IDE）、ビルドパイプライン、静的・動的セキュリティなど、ツールチェーンでインスタンス化された自動プロセス）から、それぞれに適した方法（ピアレビュー、リードレビュー、ウォークスルー、静的・動的スキャン、脆弱性スキャンなど）でレビューする。
□ **S(1)-1.3 リスク対応記録** 設計上の決定事項、リスクへの対応、承認された例外措置に関する記録を保持し、ソフトウェアのライフサイクル全体を通じて監査や保守の目的で使用できるように維持する。	
取組例	• 設計上の決定事項、リスク軽減の達成方法やセキュリティ要件に対する承認された例外の根拠など、各リスクへの対応を記録する。 • 各リスクへの対応に関する記録を維持する。
□ **S(1)-1.4 リスクベースの定期的確認** セキュリティ要件に対して承認された全ての例外とソフトウェア設計、及びソフトウェアの設計時に作成したリスクベースの分析・評価結果をレビューし、リスクへの対処が適切か定期的に確認する。	
取組例	• 承認された全ての例外措置を定期的に再評価し、必要に応じて然るべき変更を実装する。 • リスクモデルをレビューし、リスクへの対処が適切か定期的に確認し、必要に応じて変更を実装する。（sp800-218 PW1.2 notional implementation example）

■ 脅威モデリングとリスク管理

脅威モデリングとは、ソフトウェアの潜在的な脅威と脆弱性を特定し、実施すべきセキュリティ対策を検討しリスク管理につなげるための分析手法である。

対象のソフトウェアと守るべき資産を明確化し、資産に悪影響を与える脅威と脆弱性を分析する。

様々なフレームワークが公開されており、代表的なものにマイクロソフト社が開発した STRIDE モデルがある。このモデルは、「Spoofing（なりすまし）、Tampering（改ざん）、Repudiation（否認）、Information disclosure（情報漏えい）、Denial of service（サービス妨害）、Elevation of Privilege（特権の昇格）」の観点で脅威を特定し、セキュリティ対策を検討するための方法論として活用されている。（関連する要求事項：S(1)-1.1）

コラム	“悪しき慣行”のガイダンス

様々な組織において、ソフトウェアセキュリティについての Best Practice（優れた慣行）が整理されているが、この対極の概念として、米国では、CISA（サイバーセキュリティ・社会基盤安全保障庁）と FBI（連邦捜査局）から、「Product Security Bad Practices Guidance（製品セキュリティの悪しき慣行）」が公開されている。このガイダンスでは、ソフトウェアベンダーにとって、リスクが高いと考えられる製品セキュリティに係る不適切な行為を示し、ソフトウェアベンダーがこれらのリスクを軽減するための推奨事項を以下の３つにカテゴライズして提供している。本ガイドライン（案）との関連が深いものは、１番目と２番目である。

① 製品特性：

ソフトウェアの観察可能なセキュリティ品質に関わる事項であり、メモリセーフでない言語での開発などが該当する。本ガイドライン（案）に直結するものとして、既知の脆弱性を含むコンポーネントを含むソフトウェア製品をリリースすること、脆弱性のあるオープンソースソフトウェアを利用することが該当する。

② 組織的プロセスとポリシー：

ソフトウェアセキュリティのための透明性確保に関する事項であり、脆弱性開示ポリシーの公表を怠ること等が該当する。

③ セキュリティ機能：

ソフトウェア製品が保持すべきセキュリティ機能に関わる事項であり、多要素認証やログ機能の欠落等が示されている。

■ 開発するソフトウェアのリスク評価を実施する主体

この文書が対象とする「開発するソフトウェア」は、システム・サービスの機能性を実現するソフトウェア、IoT 機器等に組み込まれるソフトウェア、チップに搭載されるファームウェアなどである。

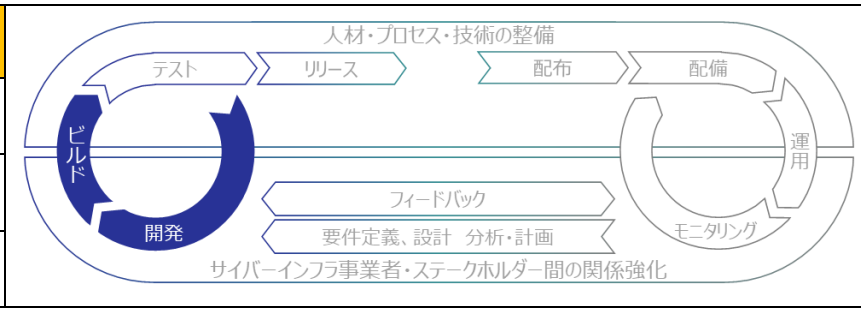
システム・サービスレベルのリスクモデリング及び分析・評価は、システム・サービスの運用主体が実施する。また、IoT 機器やチップレベルのリスクモデリング及び分析・評価は、IoT 機器やチップの設計・製造主体が実施する。

一方、S(1)-1 でフォーカスするリスクは、開発するソフトウェアに関するリスクであり、開発者が主体的にリスクモデリング及び分析・評価を実施する。専用のソフトウェアの場合、利用環境が特定された上位レベルのリスクを考慮することが求められる。また、汎用のソフトウェアの場合、想定される利用環境におけるソフトウェアの使用法に基づいたリスクを考慮することが求められる。

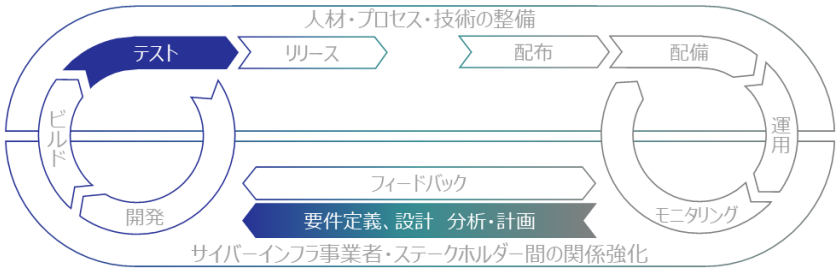
■ セキュリティ対策に伴う費用の考え方

セキュリティ対策に伴う費用について顧客の理解を得るためには、サイバーインフラ事業者自身の利益配分だけではなく、顧客のセキュリティ向上を踏まえた提案が必要となる。また、顧客への費用に関する啓発活動とともに、増加費用の明細（システム開発又は改修の場合は、かかるコストの見積り、クラウドの場合は適用するサービスメニューの追加）と必要性に関する顧客への説明責任が求められる。

顧客とサイバーインフラ事業者の両者が、セキュリティ対策の必要性やコストについて、相互の認識を合わせることが重要であり、顧客とサイバーインフラ事業者との間の役割分担、開発環境や用語整備による業界内の共通化・標準化、対象システムのライフサイクルにわたるコミュニケーションなどにより、理解を醸成することが望まれる。（関連する要求事項：S(1)-1 全般）

<table>
<tr><td rowspan="4">S (1)-2</td><td>開発者</td><td rowspan="4"> 
</td></tr>
<tr><td>供給者</td></tr>
<tr><td>運用者</td></tr>
<tr><td>顧客</td></tr>
<tr><td colspan="3">セキュアなビルド
開発言語や開発環境に適したセキュアコーディング及びシステム構築のプロセスを定義し、これに従いコードを生成・ビルドする。設定を含むコードのレビュー及び分析を実施し、対応結果をプロセスにフィードバックする。</td></tr>
<tr><td colspan="3">□　S(1)-2.1 セキュア開発プロセスの定義
セキュアコーディングの観点、ビルド実施タイミングと方式、自動化ツールの利用、トレーニングなど、セキュアコーディング、セキュアビルド及びデフォルトセキュアに関するプロセスを定義する。</td></tr>
<tr><td>取組例</td><td colspan="2">• 開発言語や開発環境に共通する脆弱性をチェックし、プロセスにこれらの脆弱性が組み込まれないようにする。
• 実行可能形式のセキュリティを向上させる機能を提供するコンパイラ、インタプリタ、及びビルドツールを使用する。（sp800-218 PW.6.1 task）
• 開発手法及び開発環境における自動化支援（品質向上）を導入する。
• 開発における中間成果物、及び構成ベースラインを管理するための構成管理ツールを適用する。
• セキュアコーディング手法の使用、及び自動化された機能を備えた開発環境を使用する前に、適切なトレーニングを行う。（sp800-218 PW.5.1 notional implementation example）
• セキュアなデフォルト設定を実装するとともに、デフォルトの構成を使用可能な形式で保存し、変更管理の慣行に従った変更を可能とする。
• ソフトウェアを利用するユーザ（システム管理者など）向けにセキュアな設定とその操作に関するガイダンスを文書化する。
• 開発手法及び開発環境における AI サポートによる自動化支援（静的解析への AI 適用による品質向上など）を導入する。</td></tr>
</table>

□ **S(1)-2.2 セキュアビルド** 実行可能形式のセキュリティを向上させる機能を提供するコンパイラ、インタプリタ、及びビルドツールを使用し、コードを生成・ビルドする。	
取組例	• コンパイラ、インタプリタ、ビルドツールの機能と構成を決定し、承認された構成を CaC (Configuration-as-Code)として使用可能とする。 • コンパイラ、インタプリタ、ビルドツールを導入・更新するための変更管理プロセスを整備し、真正性と整合性を定期的に検証する。 • ソフトウェアのデプロイに使用するコンテナなどの仮想化技術は、保護された構成を適用する。
□ **S(1)-2.3 検証とフィードバック** レビュー及び分析による検証により発見された問題の根本原因を特定し、その対応結果をプロセスにフィードバックする。	
取組例	• ソフトウェアのライフサイクルの段階に応じて、コードのレビュー及び分析の方法を選択する。（sp800-218 PW.7.1 notional implementation example） • バックドアやその他の悪意のあるコードの有無を検査するために、専門のレビュアーの協力を得る。 • 静的コード分析ツールなど、ツールを使用した場合は、その分析結果を文書化する。 • 静的分析ツールを使用して、コードの脆弱性と組織のセキュアコーディング標準への準拠を自動的に点検し、ツールによって報告された問題を人間がレビューし、必要に応じて是正する。（sp800-218 PW.7.2 notional implementation example） • レビュー及び分析によるセキュリティ要件準拠性を検証し、発見された問題の根本原因を特定、文書化し、その対応結果を、セキュアコーディング、セキュアビルド及びデフォルトセキュアに関するプロセスにフィードバックする。（statement から派生）
□ **S(1)-2.4 コードベース** レビュー及び分析の対象は、ソースコードのみでなく、可読性があると組織が決定した様々な形式のコード（設定ファイル等）も対象とする。	
取組例	• 組織のセキュアコーディング標準に基づいて、コードのレビュー及び分析を実行し、発見された全ての問題と推奨される対応案を、開発チームのワークフロー又は問題追跡システムに記録し、優先順位付けする。（sp800-218 PW.7.2 task） • レビュー対象とする開発環境の設定対象には、一般的な脆弱性と弱点を防ぐためのコンパイラ構成、開発言語や環境、必要に応じて、サードパーティのコードや組織内で書かれた再利用可能なコードモジュールも含める。

S (1)-3	開発者 供給者 運用者 顧客	

テスト

ビルドフェーズまでのレビュー及び分析で特定されなかった脆弱性を発見するために、機能テストに加え、脆弱性テスト、侵入テストを設計・実施し、発見された脆弱性への対策を実施する。

□　S(1)-3.1 テスト計画

脅威モデルとリスク分析に基づき、テスト範囲及びテスト方式を決定し、テスト計画を立案する。

取組例	• レビュー、分析、又はテストで特定されていない脆弱性を見つけるために、実行可能コードのテストを実行する必要があるかどうかを判断する。（sp800-218 PW.8.1 task） • 実行可能コードのテストを実行する場合、テスト範囲及びテスト方法を決定し、テスト計画を立案する。（sp800-218 PW.8.1 task） • テストの対象には、バイナリ、直接実行されるバイトコード、ソースコード、及び組織が実行可能であるとみなすその他の形式のコード、ソフトウェアを含める。 • コードのテストの対象には、サードパーティの実行可能コード、及び社内で作成された再利用可能な実行可能コードモジュールを必要に応じて含める。 （システム・サービスのソフトウェアの場合） • 委託元企業がテストに関するガイドラインやテンプレートを配布する場合は、これらを含めてテスト計画を立案する。

□　S(1)-3.2 テスト方式

テスト方式には、機能テスト、脆弱性テスト、ファジング、侵入テストなどを含める。

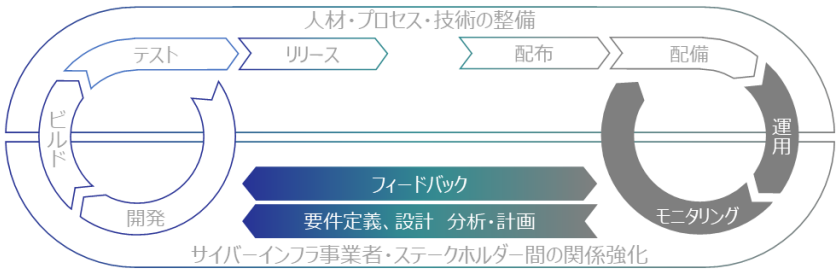
取組例	• デフォルト設定を含む各設定が期待通りに動作し、不用意にセキュリティ上の弱点や運用上の問題などを引き起こしていないことを確認するためのテスト方式を含める。 • ソフトウェア内部の入出力処理の問題を発見するためには、ファジングテストをテスト方式に含める。 • 攻撃者がリスクの高いシナリオでソフトウェアを侵害しようとする方法をシミュレートする侵入テスト（ペネトレーションテスト）をテスト方式に含める。 • 静的・動的脆弱性テストや、改修の悪影響を除去する回帰テストを、プロジェクトの自動テストスイートに統合する。

□　S(1)-3.3 テスト実施

テスト計画に従ってテストを設計、実施し、結果を文書化する。

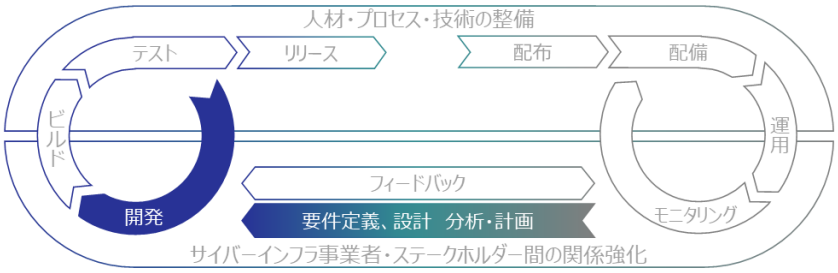
取組例	• デフォルト設定を含む各設定が期待通りに動作することを確認できるように、マニュアルに従い、本番の利用環境を考慮したテストを実施する。

☐ **S(1)-3.4 問題への対応** テストの結果、発見された全ての問題と推奨される対応策を開発チームのワークフローに組み込み、対処する。	
取組例	• テストの結果、発見された全ての問題と推奨される対応策を、開発チームのワークフロー又は問題追跡システムに記録し、優先順位付けして文書化する。（sp800-218 PW.8.2 task） • 発見された問題の根本原因を特定して記録する。（sp800-218 PW.8.2 notional implementation example） • 脆弱性分析、脆弱性のリスク軽減策及びツールの分析結果を、レビュアーの求めに応じて提示できるように整理する。

S (1)-4	**開発者** 供給者 **運用者** 顧客	
サービスのモニタリング ソフトウェアがその導入環境（ネットワーク、プラットフォーム、サービスなど）と整合性をもって情報資産を保護、維持することをモニタリングするプロセス及びシステムを整備し、実施する。		
☐ **S(1)-4.1 資産管理** 運用者は、システム・サービスが扱う資産、及びシステム・サービスを構成する資産に関する資産管理手順と資産リストを整備する。		
取組例	• システム・サービスの資産管理に、変更管理と構成管理を統合し、セキュアな構成を維持する。 • セキュリティポリシーに基づき、システム・サービスのセキュリティを保守するための手順を整備、維持する。	
☐ **S(1)-4.2 モニタリング環境の整備** 運用者は、リスク発生時の潜在的な影響を最小化するためにシステムを適切に分離し、ソフトウェアによる資産保護上重要なリスクを監視するモニタリング環境を整備する。		
取組例	• サービスの管理用システムは、専用で使用し、他の業務と混在しないようにする。 • 重要なリスクを判定するための診断ツールを適用する。	
☐ **S(1)-4.3 セキュリティメカニズムの整備** ソフトウェア及びソフトウェアを適用するシステム・サービスが、動作環境又はデジタルインフラなどのリソース上にある情報資産及びデータの機密性・完全性を保護し、監視可能とするための適切なセキュリティメカニズムを整備する。		
取組例	• 機密性・完全性を保護するためのメカニズムとして、ファイアウォール、暗号化、署名等を用いる。	

☐ **S(1)-4.4 モニタリングと評価** 運用者は、重要なサービスを提供するシステムに適用したメカニズムの動作状況をモニタリングするとともに、定期的にセキュリティ評価を実施し、組織のリスク管理の枠組みに統合する。	
取組例	• 重要なサービスに適用したソフトウェアのメカニズムの動作状況をモニタリングし、ソフトウェアの動作に関連するネットワーク、プラットフォーム及び連動する他のサービスと整合して保護、維持されていることを確認する。（statement から派生） • システム・サービスに関するセキュリティ評価の実施タイミングに、新システム導入時、及び運用システムの大きな変更時を含める。

（２）ライフサイクル管理、透明性の確保

S (2)-1	**開発者** 供給者 運用者 顧客	
セキュアなソフトウェアコンポーネントの手配 外部から手配した商用、オープンソース、その他のサードパーティのソフトウェアコンポーネントが、そのライフサイクルを通じて、組織が定義した要件に準拠していることを検証する。		
☐ **S(2)-1.1 ソフトウェアコンポーネントの手配** 外部から手配する商用、オープンソース、その他のサードパーティのソフトウェアコンポーネントは、組織が定義した要件を満たす安全性の高いものを採用する。		
取組例	• サードパーティのソフトウェアコンポーネント（暗号モジュールや標準プロトコルなど、標準化されたセキュリティ機能・サービスの提供を含むソフトウェアライブラリ、モジュール、ミドルウェア、フレームワークなどを含む）を、利用環境を想定してレビュー及び評価する。 • サードパーティのソフトウェアコンポーネントの安全な構成を決定し、開発者がその構成を configuration-as-code などにより簡単に使用できるようにする。（sp800-218 PW.4.1 notional implementation example） • サードパーティのソフトウェアコンポーネントの想定利用環境における安全性を検証するために、ソースコードからのビルド（セキュリティスキャンを含む）、静的解析（バイナリスキャン）、動的解析などを必要に応じて実施する。 （システム・サービスのソフトウェアの場合） • 顧客の要求に応じて、SSDF への準拠を示す自己適合証明書の提出と、その準拠を示す成果物として必要に応じて SBOM を提出する。	

□ **S(2)-1.2 ソフトウェアコンポーネントの開発・維持** 外部からソフトウェアコンポーネントを手配しない場合、組織で確立されたセキュリティ基準・慣行に従い、安全性の高いソフトウェアコンポーネントを社内で開発、維持する。	
取組例	• コンポーネントを開発及び維持する際は、組織が確立したセキュリティ慣行に従って、安全なソフトウェア開発を行う。（sp800-218 PW.4.2 notional implementation example） • 開発したソフトウェアコンポーネントの安全な構成を決定し、開発者がその構成を configuration-as-code などにより簡単に使用できるようにする。（sp800-218 PW.4.2 notional implementation example）
□ **S(2)-1.3 ソフトウェアコンポーネントのリスク評価** 各ソフトウェアコンポーネントの出所情報を取得・分析し、そのコンポーネントがもたらすリスクを評価する。	
取組例	• 組織が承認した商用ソフトウェアコンポーネントとコンポーネントバージョンの一覧を、それらの出所データ（SBOM など）とともに維持する。（sp800-218 PW.4.1 notional implementation example） • 各ソフトウェアコンポーネントの構成分析（ソースコード、バイナリコード）を実施し、安全な構成を簡単に使用できるようにリポジトリを整備する。 • 各ソフトウェアコンポーネントの出所情報（SBOM、ソース構成分析、バイナリソフトウェア構成分析など）を取得、分析して、コンポーネントがもたらす可能性のあるリスクを評価する。（sp800-218 PW.4.1 notional implementation example） • デジタル署名又は他のメカニズムにより、ソフトウェアコンポーネントの完全性を検証・確認する。その際、使用される証明書を識別・検証し、使用する暗号規格・標準についても検証する。 • サプライチェーン上でのソースコード、構成情報、変更情報の管理の共有、及び必要に応じて SBOM を共有する。
□ **S(2)-1.4 ソフトウェアコンポーネントの公知脆弱性の確認** 各ソフトウェアコンポーネントの公知脆弱性、サポート期間を定期的にチェックする。	
取組例	• 各ソフトウェアコンポーネントの公知脆弱性、サポート期間の定期的なチェックは、必要に応じて、外部の診断サービス又は審査サービスの活用も検討する。 • 導入したソフトウェアコンポーネントの既知の脆弱性の自動検出をツールチェーンに組み込む。
□ **S(2)-1.5 ソフトウェアコンポーネントの更新** 各ソフトウェアコンポーネントを新しいバージョンにセキュアに更新するプロセスを導入する。	
取組例	• 導入した各ソフトウェアコンポーネントを新しいバージョンに更新するプロセスを実装し、それらのバージョンからの全ての移行が正常に完了するまで、古いバージョンのソフトウェアコンポーネントを保持する。（sp800-218 PW.4.1 notional implementation example） • 導入した各ソフトウェアコンポーネントに存在する、発見された対応すべき脆弱性（公知脆弱性を含む）は、当該ソフトウェアコンポーネントのサプライチェーン上で情報共有し、迅速なパッチ適用等により対処する。 • 取得したバイナリの整合性又は出所を確認できない場合は、ソースコードの整合性と出所を検証した後に、ソースコードからバイナリをビルドする。（sp800-218 PW.4.1 notional implementation example）

■ セキュアなソフトウェアコンポーネントを調達する目的と情報共有の必要性

ソフトウェアで実現する機能は、その機能を個別にスクラッチ開発する代わりに、十分にセキュリティが保護された既存のシステムコンポーネントや標準化されたソフトウェアコンポーネント（標準に準拠したログ管理、アクセス制御など）といったセキュリティが確保された既存のソフトウェアを再利用することで、脆弱性がもたらされるリスクを低減することができる。

ソフトウェアの脆弱性は、開発段階において可能な限り対策することとし、その後発見された残存脆弱性も以降のライフサイクルとサプライチェーンにおいてパッチ適用等により迅速に対処する必要がある。これを実現するためには、安全なソフトウェア開発手法（SSDF）やサイバーセキュリティ―サプライチェーン・リスクマネジメント（C-SCRM）に基づいた、持続的に維持されたソフトウェアの開発・維持環境下における構成管理・変更管理を通じてソフトウェアに関する情報共有を実現する必要がある。（関連する要求事項：S(2)-1.1）

■ SBOM の活用・共有によるセキュアなソフトウェア流通の仕組み構築に向けて

開発者又は供給者から調達したソフトウェアに脆弱性が発見された場合、調達者側（顧客側）でそのソフトウェアを修正するなどの対策を自ら行うためには、そのソフトウェアのソースコードや SBOM を入手し、これらの構成管理及び変更管理を自ら実施することが望ましい。

一方、供給・開発者側にソースコードの権利を留保したまま、供給・開発者がソースコードと SBOM を管理し、適切に構成管理及び変更管理することを契約で定め、必要に応じて証跡を追跡できるようにするなど、脆弱性が管理されたソフトウェア（バイナリやソフトウェアが組み込まれた IoT 機器など）が流通する標準的な仕組みが実現できるのであれば、納品とともにソースコードや SBOM を流通させることを求める必要はないかもしれない。また、脆弱性の影響範囲の迅速な特定を検討するために、自社製品のトレーサビリティ（組み込み先）を追跡する仕組みや、CISA による拘束力のある運用指令（BOD）や EU Cyber Resilience Act のように、政府から重要インフラ事業者、あるいは顧客・運用組織からサイバーインフラ事業者に対して、脆弱性有無の報告を求めるような枠組みができれば、より効果的となるかもしれない。

このように、SBOM の活用・共有の必要性は高まってきており、一部の業界では統一ルールの検討を進めている。SBOM を導入し組織間で本格的に共有するために、経済産業省が 2024 年 8 月 29 日に公表した「ソフトウェア管理に向けた SBOM(Software Bill of Materials)の導入に関する手引 ver2.0」等を参考に仕組みを検討し、構築することが望まれる。具体的には、SBOM を含むセキュアなソフトウェア流通に関する仕組みの実現可能性や制約事項、及び取組の優先度などを十分に把握した上で、セキュアなソフトウェア流通の仕組みを定め、契約などで合意することが重要である。（関連する要求事項：S(2)-1.3）

■ **欧米における政府関係機関が導入するソフトウェアへの要件**

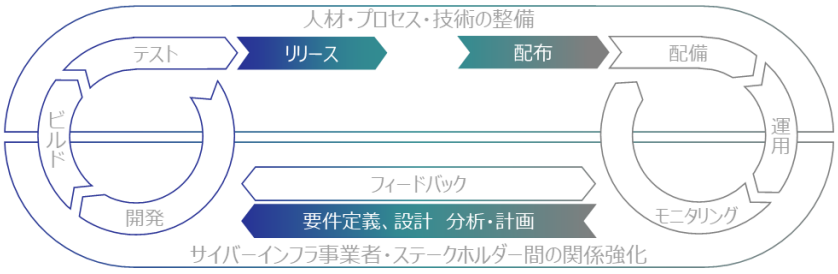
欧米を中心に、安全なソフトウェア開発手法（SSDF）に関する取組が進みつつあり、米国では、米国行政管理予算局(OMB)から「M-22-18「安全なソフトウェア開発の実践によるソフトウェアサプライチェーンのセキュリティの強化」（及び M-23-16 同更新版）が発表されている。

この文書では、連邦政府関係機関に、安全なソフトウェア開発手法（SSDF：SP800-218）を実装していることを証明できるソフトウェアベンダーを採用することを求めており、ソフトウェアベンダーに、SSDF の実装の適合性を証明する自己適合証明書の提出と、その準拠を示す成果物として必要に応じて SBOM を提出することを求めている。

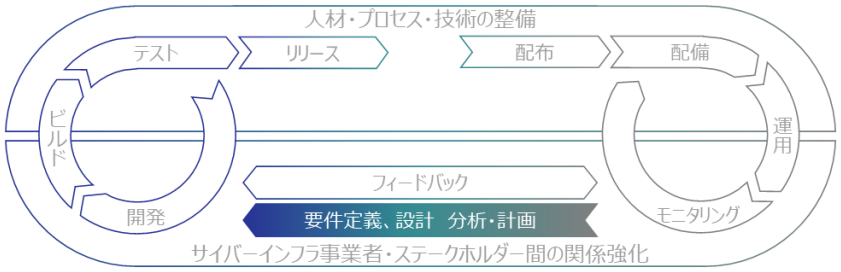
自己適合証明書とは、EO14028 に基づく SSDF への準拠を示す書面であり、脆弱性開示やその対応等、安全なソフトウェア開発が継続的に行われたプロセスと手順を証明するものである。（関連する要求事項：S(2)-1.1）

（参考）以下 URL、経済産業省のソフトウェアタスクフォース資料の p31 には、OMB 覚書(M-22-18)において、SSDF 自己適合証明書、SBOM 要求に関する記載がある。

https://www.meti.go.jp/shingikai/mono_info_service/sangyo_cyber/wg_seido/wg_bunyaodan/software/pdf/010_03_00.pdf

<table>
<tr><td rowspan="4">S (2)-2</td><td>開発者</td><td rowspan="4"> 
</td></tr>
<tr><td>供給者</td></tr>
<tr><td>運用者</td></tr>
<tr><td>顧客</td></tr>
<tr><td colspan="3">リリースファイルやデータのセキュアなアーカイブ
ソフトウェアのリリースごとに保持すべき必要なファイルやデータをアーカイブし、必要な人員、ツール、サービスのみにアクセスを制限する。ソフトウェア部品表（SBOM）の段階的な採用を通じて、各リリースの全てのコンポーネントについて、出所データを収集、保護、維持、共有する。</td></tr>
<tr><td colspan="3">□ S(2)-2.1 コードベースの保護
全ての形式のコードベースを不正アクセスや改ざんから保護するために、リポジトリにコードや設定情報を保管し、承認された担当者、ツール、サービスなどのみがアクセスできるよう最小権限の原則に基づいたアクセス制御を実施する。</td></tr>
<tr><td>取組例</td><td colspan="2">• コードベースのリポジトリには、ソースコード、実行可能コード、設定、リソースファイル、コンテナイメージ、コードとしての構成（CaC）などの全ての形式のコードを保存する。また、オープンソースや言語クラスの部品群、完全性検証情報、出所データなども対象とする。
• ソースコードや実行可能コードのファイルの正当性や整合性を保護するために暗号化技術（コード署名、コミット署名、ハッシュなど）を使用する。
• コードに加えられた全ての変更を別の人がレビューし、コード所有者が承認する。</td></tr>
</table>

□ **S(2)-2.2 リリースのアーカイブ** リリース後に発見された脆弱性を分析、特定できるようにするために、各ソフトウェアのリリースをアーカイブ化して保護する。	
取組例	• 確立された組織ポリシーに従って、リリースファイル、関連するイメージなどをリポジトリに保存する。必要な担当者による読み取り専用アクセスを許可し、他の人によるアクセスを禁止する。（sp800-218 PS.3.1 notional implementation example） • 機能拡張時には関連するコードと実行可能ファイルを保存し、全ての変更を確認して承認する。
□ **S(2)-2.3 リリースの出所データの共有** 各ソフトウェアリリースの全てのコンポーネントの出所データを収集、保護、維持、共有する。	
取組例	• SBOM などを使用して、ソフトウェアを受領・取得した組織の運用・対応チームが出所データを利用できるようにする。 • 開発者が直接取得してソフトウェアに組み込んだ全てのサードパーティ製コンポーネントを文書化し、可能な限り元のソースを追跡する手段を採用する。 • サードパーティ製バイナリのスキャン、及びサードパーティ製コンポーネントのセキュリティに関するリスク評価を行う。 • オープンソースライブラリのチェックシステムを導入し、定期的にチェックする。 • 供給者と開発者の両者が協力して、完全性検証のための署名サーバの整合性を確保する。

S (2)-3	**開発者** **供給者** **運用者** **顧客**	

関係者間のセキュリティ要件の確立

関係者間で合意すべきセキュリティ要件を確立し、契約又は共有するポリシーに盛り込む。

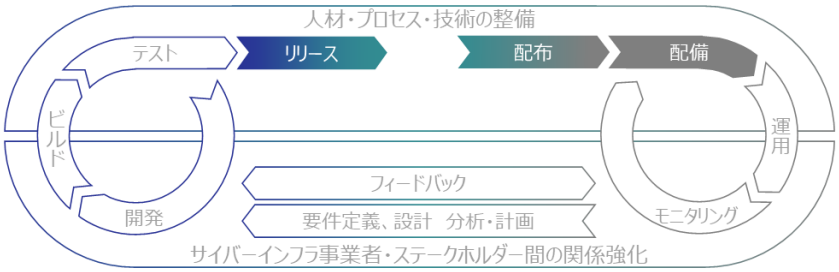
□ **S(2)-3.1 セキュリティ要件の合意** IT 製品（自社のソフトウェアで再利用するための商用ソフトウェアコンポーネントを含む）又はサービスを提供するサードパーティとの契約又は共有するポリシーに、明示的なセキュリティ要件を盛り込む。	
取組例	• サードパーティ（供給者）との契約又はポリシーに盛り込む要件の例は以下の通り。 ➢ 供給者の情報セキュリティ順守事項の監視と開示（ソフトウェアのセキュリティ要件） ➢ 供給者との間で生じる可能性のある問題の情報を共有する規則 ➢ セキュア開発プロセスの実行（第三者によるプロセス検証、侵入テスト等の脆弱性の検証を含む） ➢ 脆弱性の管理（脆弱性の開示、パッチ管理を含む） ➢ SBOM の提供 ➢ 供給者からの構成部品が真正であることを確実にするプロセスの実行（サプライチェーン関連のデータの転送中に不正アクセスから保護する） ➢ 供給者の製品及びサービスに関連する脆弱性への対処 ➢ 供給者の可用性確保と回復の対策 ➢ サポートの提供、SLA の定義、苦情対応 ➢ その他、両者の責務と役割の定義、契約完了・終了の要件など
□ **S(2)-3.2 サプライチェーンセキュリティ要求への対応** 提供する IT 製品又はサービスを受領・取得する組織が採用するサプライチェーンセキュリティ要件と同等のサプライチェーンセキュリティ要件に対応する。	
取組例	• サプライチェーンセキュリティ要件に基づいたサードパーティ製コンポーネントの供給者の選定プロセスを整備し、選定のエビデンスを取得する。 （システム・サービスのソフトウェアの場合） • ソフトウェア品質に対する顧客からの要求仕様を満たすことを示すために、顧客が指定するソースコード診断ツールを実行し、エビデンスを提出する仕組みを採用する。
□ **S(2)-3.3 セキュリティ要件を満たさないリスクへの対処プロセスの整備** 受領・取得するサードパーティ製の IT 製品又はサービスが満たさないセキュリティ要件がある場合のリスクに対処するプロセスを整備する。	
取組例	• サプライチェーン・リスクを軽減するための取得戦略、取得方法を定める（購入用途を明確化しない、信頼できる配布先の選定、契約内容及び管理状態の良い供給者へのインセンティブ付与など）。 • サードパーティから入手した SBOM を検証、更新する。 （システム・サービスのソフトウェアの場合） • サプライチェーンを管理監督するために、構成管理、変更管理、開発・維持環境のハードニングなどを実施・提供する。

■ 委託先である供給者・開発者への支援・協力

サプライチェーン全体のセキュリティの確立・維持を目指すために、委託元である顧客は、供給者や開発者への支援・協力の一環として、合意したサプライチェーンセキュリティ要件を満たすことができるように、可能な支援・管理を行う。例えば、以下のような支援・管理が考えられる。

- 供給者・開発者に開発環境を提供する、又は利用を許可する。その際、委託先人員の受講とテストの合格をもって開発環境の利用を許可する。
- 重要なセキュリティ要件から段階的に合意する。
- プライム事業者が委託先を支援・管理する。サイバーインフラ事業者が複数関わるケースでは、全体の脆弱性は、顧客あるいは統括し全体の上流工程を担うコンサル会社が関わって、脆弱性が盛り込まれないように管理する。
- サブ事業者が子会社の場合、親会社がサプライチェーンセキュリティを統制する形態が考えられるが、利益供与に注意する必要があることから、財務部門と IT 部門が関与して全体バランスを図る。
- 管理基準を改定してもすぐには対応できない委託先企業に対して猶予期間を設ける。

このように、サプライチェーンセキュリティ要件に対応するに当たっては、プライム事業者、サブ事業者、販売代理店など、サプライチェーンの階層と各階層で取組レベルにバラつきがあり、現状を踏まえた基本的な取組のレベルアップが必要である。その際には、役割分担と実施事項を契約時に決めることが、ステークホルダー間の協力関係を深めるためにも重要となる。今後は、業界団体や公的機関とも議論を深めてガイドラインを整備し、関係者の将来的な取組を通じて、責任範囲の定め方、観点などを整理していくことが望まれる。（関連する要求事項：S(2)-3 全般）

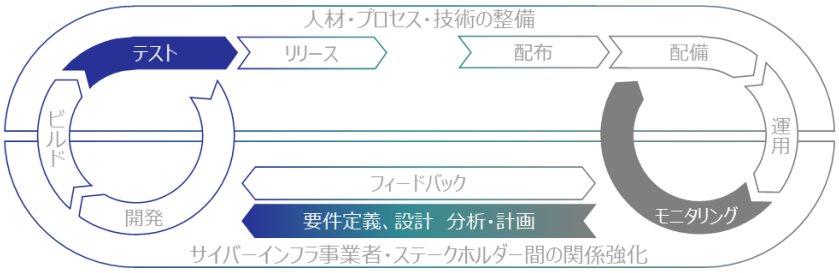
<table>
<tr><td rowspan="4">S (2)-4</td><td>開発者</td><td rowspan="4"> 
</td></tr>
<tr><td>供給者</td></tr>
<tr><td>運用者</td></tr>
<tr><td>顧客</td></tr>
<tr><td colspan="3">利用者への適切な情報提供
ソフトウェアの導入・インストールから操作、利用終了までのライフサイクル全体でセキュアな利用を容易にするガイダンスをソフトウェア利用者が確実に利用できるようにする。</td></tr>
</table>

□ **S(2)-4.1 セキュアな導入・設定・操作・変更・廃棄・終了** ソフトウェアをセキュアに導入・設定・操作するための情報、及び変更の影響・廃棄・提供終了・利用終了に係る情報をソフトウェア利用者が継続的に利用できるようにする。	
取組例	• セキュアなデフォルト設定（又は該当する場合はデフォルト設定のグループ）を実装する。（sp800-218 PW.9.2 task） • ソフトウェア利用者（システム管理者など）向けガイダンスに以下の内容を含める。 ➢ 初期インストール、追加コンポーネント、アップデート、及びパッチのインストールのためのセキュアな導入手順、及びセキュアな設定とするための手順 ➢ ソフトウェアの完全性検証情報、及び構成ガイド ➢ セキュアな設定情報の詳細（各設定の目的、デフォルト値、セキュリティ上の関連性、潜在的な運用上の影響、他の設定との関係など） • ソフトウェアのサポート終了の決定をユーザ（顧客）に明確に伝え、サポート終了予定日を特定する。 • ソフトウェアの供給後においても、例えば、自ら開発したプログラム等に脆弱性が発見された場合には、迅速にサプライチェーン上の関係者に広く情報を提供するなど、ソフトウェア利用者の安全性の確保のために必要となる情報を提供する。 （システム・サービスのソフトウェアの場合） • 初期のシステム提供と並行して、製品サポートの内容と期間に関する現実的な想定を伝達する。 • サービスを提供するクラウド事業者は、利用者にわかりやすく情報提供し、運用者やシステムの構築・設置者に対して具体的な情報ガイドなどを提供する。
□ **S(2)-4.2 整合性検証情報の提供** ソフトウェアの整合性・完全性の検証に必要な情報をソフトウェア利用者が継続的に利用できるようにする。	
取組例	• ソフトウェア利用者（システム管理者など）向けに以下を提供する。 ➢ 供給するソフトウェアの SBOM 又はこれと同等の情報 ➢ 供給者から顧客への流通経路での改ざん対策が適切であることを検証するための情報（リリースファイルの暗号化ハッシュ、コード署名など） • ソフトウェア利用者（システム管理者など）向けに以下を提供する。 ➢ 配付システムの保護対策（信頼できる認証局をコード署名に使用、コード署名プロセスの定期的な見直し、その他署名環境の保護対策など）

■ セキュアなガイダンスを求める要件

IT 製品のセキュリティ評価のための国際標準である Common Criteria（ISO/IEC 15408, ISO/IEC 18045）では、顧客である IT 製品の利用者がセキュアに導入・運用が開始できるように、マニュアルに対してセキュアなインストール・利用に関する利用者への適切な指示を記載することを要求している。また、CISA の「ソフトウェアサプライチェーン攻撃に対する防御」では、入手したソフトウェアが改ざんされていないことを顧客が確認できるよう、ソフトウェアのリリースの完全性を検証する仕組み（コード署名証明書の保護など）を提供することが示されている。（関連する要求事項：S(2)-4.1）

（３）残続する脆弱性の速やかな対処

S (3)-1	開発者 供給者 運用者 顧客	

継続的な脆弱性調査

ソフトウェアの脆弱性の開示と是正に関する方針を定め、その方針に必要な役割、責務、プロセスを定義し、実施する。

☐ **S(3)-1.1 脆弱性対応体制の設置** ソフトウェア製品の脆弱性の開示と修復に対処するポリシーを定め、そのポリシーをサポートするための脆弱性対応（インシデント対応を含む）に関する体制を設置し、必要な役割、責務、プロセスを定義する。	
取組例	• ソフトウェアの製品セキュリティインシデント対応チーム（PSIRT）を設置し、製品セキュリティに関わるインシデント対応プロセスを整備する。（sp800-218 RV.1.3 implementation example） • ソフトウェアの製品セキュリティを侵害する一般的な脅威に対する明確な手順と手続、インシデント対応のトリガー、ステップ、復旧時間目標、サービスに関するコンティンジェンシープランを定める。 • インシデント対応プロセスの演習を定期的に実施する。
☐ **S(3)-1.2 コミュニケーション計画** 全ての利害関係者に対するコミュニケーション計画を定める。	
取組例	• 全ての利害関係者に対するコミュニケーション計画を含む、脆弱性の開示プロセスを整備する。 • 開示した脆弱性情報への容易なアクセスと取り込みをサポートするメカニズム（メーリングリスト、ポータルなど）を整備する。
☐ **S(3)-1.3 脆弱性情報の収集** 公知情報の探索、ソフトウェア利用者からの通知、外部脅威情報の取得、システム構成データのレビュー、その他の方法を通じて、新たな脆弱性情報を収集する。	

取組例	• 脆弱性情報の収集プロセスを整備する。（statement から派生） • 公開情報源（CVE やサードパーティのサポートチャネルを監視）から、ソフトウェア及びソフトウェアに組み込まれているサードパーティ製コンポーネントの脆弱性に関する情報を収集する。 • ソフトウェア及びソフトウェアに組み込まれているサードパーティ製コンポーネントのベンダー、ソフトウェアの取得者/ユーザ（顧客など）、第三者の研究者が特定したソフトウェアの脆弱性の疑いに関する情報を収集し調査する。 • ソフトウェアに組み込まれたサードパーティ製コンポーネントを特定し、修正の有無とサポートの終了期限を定期的に確認する。 • 脅威インテリジェンスからの情報源を使用して、一般的な脆弱性がどのように悪用されているかをよりよく理解する。 • 全てのソフトウェアコンポーネントの出所とソフトウェア構成データを自動的にレビューし、それらに含まれる新しい脆弱性を特定する。（sp800-218 RV.1.1 notional implementation example） （システム・サービスのソフトウェアの場合） • 対処優先度に基づく脆弱性管理を行うに当たり、SSVC 等のツールの活用を検討する（自組織の資産管理と一体の取組として有効に活用する）。
□ **S(3)-1.4 未検出の脆弱性の特定** 継続的又は定期的に、ソフトウェアのコードのレビュー、分析、テストを実施し、今まで未検出の対処すべき脆弱性（不適切な設定などを含む）を特定する。	
取組例	• ソフトウェアの潜在的な脆弱性に関する全てのレポートを記録及び追跡するシステムを維持する。 • S(1)-2.3（検証とフィードバック）の慣行を適用する。 • サポートする全てのリリースに対して定期的又は継続的に自動コード分析とテストを実行するようにツールチェーンを構成する。（sp800-218 RV.1.2 notional implementation example）

■ 脆弱性対応と注意喚起に関する連携

脆弱性の注意喚起、及びパッチ作成と適用を含む脆弱性対応は、事業者の立場（SIer、運用者、販売代理店など）に基づく対応や連携上の工夫が求められる。以下に例を示す。（関連する要求事項：S(3)-1.2）

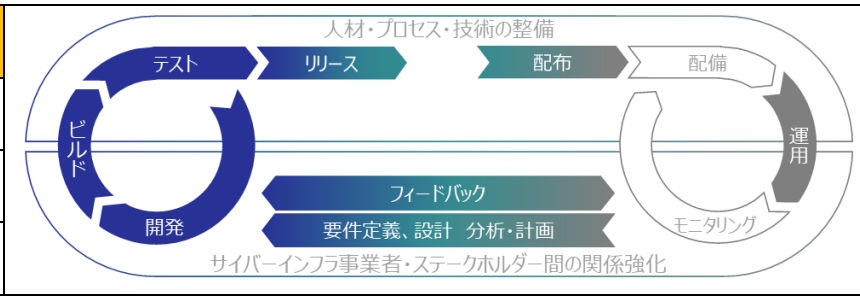
- ソフトウェア製造業者・IoT 製造業者は、製品の重大な脆弱性対応（修正パッチ作成）や顧客、SIer、販売代理店への注意喚起を実施する。
- SIer は、ソフトウェア製造業者・IoT 製造業者から顧客へ情報提供があった場合には、製品の重大な脆弱性対応（修正パッチを顧客に渡し、顧客の求めに応じて適用）を実施する。
- 運用保守ベンダーは、ソフトウェア製造業者・IoT 製造業者又は SIer から情報提供があった場合には、製品の重大な脆弱性対応（修正パッチを顧客に渡し、顧客の求めに応じて適用）を実施する。
- 販売代理店は、ソフトウェア製造業者/IoT 製造業者から情報提供があった場合には、製品の重大な脆弱性通知を顧客に行う。

■ システム・サービスのソフトウェアの脆弱性への対応が不十分な場合

クローズドな環境で運用されるレガシーシステムの中には、構成要素であるソフトウェアの脆弱性情報の収集や脆弱性への対策の必要性が認識されていないケースがある。また、システムを構成するソフトウェアの脆弱性管理に前提として必要なソフトウェア構成要素等の資産管理が不十分なケースも見受けられる。

このようにシステム・サービスにおいて、ソフトウェアの脆弱性対応が不十分な場合、既存のインシデント対応やコンティンジェンシープランの対応とあわせた脆弱性対応への取組が考えられる。（関連する要求事項：S(3)-1 全般）

- 供給者とのインシデント対応計画の提供、実装、テストに関する連携の体制を整備する。
- 提供するサービスの継続性を確保するためのコンティンジェンシープラン（電力などユーティリティの安全性を確保するための手段を含む）や継続性戦略を策定し、体制を整備する。
- これらの体制の中に、脆弱性情報の収集や脆弱性への対応を統合する。

<table>
<tr><td rowspan="4">S (3)-2</td><td>開発者</td><td rowspan="4"> 
</td></tr>
<tr><td>供給者</td></tr>
<tr><td>運用者</td></tr>
<tr><td>顧客</td></tr>
<tr><td colspan="3">検知した脆弱性への対処
リリースしたソフトウェアに残存する脆弱性に対するリスク対応を定期的に計画し、実施する。</td></tr>
<tr><td colspan="3">☐　S(3)-2.1 脆弱性の分析
残存する各脆弱性の影響に伴うリスクを把握するために必要な情報を収集し、修復又はその他のリスク対応を計画するために、各脆弱性を分析する。</td></tr>
<tr><td>取組例</td><td colspan="2">• 残存する各脆弱性について、悪用可能性、悪用された場合の影響、その他の関連する特性の推定に基づいてリスクを定量的に分析する。
• 各脆弱性を記録するために、利用可能な問題追跡ソフトウェアを使用する。（sp800-218 RV.2.1 notional implementation example）

（システム・サービスのソフトウェアの場合）
• リスク対応に伴う修正の実効性を確保するため、コード生成者以外でも改修が可能なように共通仕様書、設計書及び中間生成物を残すとともに、瑕疵担保期間を具体的に合意し、改修費用は個別に契約する。</td></tr>
</table>

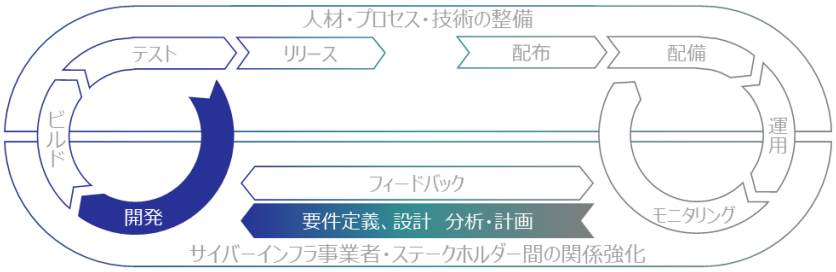
□ S(3)-2.2 脆弱性へのリスク対応 各脆弱性に対するリスク対応を計画し、実装する。	
取組例	• リスク対応は、脆弱性の回避策とするか、又はソフトウェアを修復する実装を伴う対策（恒久的な解決策が提供されるまで脆弱性を一時的に緩和する方法を含む）とするか、などリスクベースで決定し、実行する対応に優先順位を付けて計画する。（sp800-218 RV.2.2 notional implementation example） • 実装を伴う対応の場合、テスト及び検証結果を文書化する。 • 公的機関等からソフトウェアに含まれる脆弱性等の情報提供があった際には、必要なパッチ開発等を含めて適正かつ積極的に対応する。 • サイバー対処能力強化法に基づき、電子計算機等供給事業所管大臣から、サイバー攻撃による被害を防止するために必要な措置を講ずるよう要請があった際に積極的に対応する。
□ S(3)-2.3 セキュリティ勧告 セキュリティ勧告を作成し、リリースしたソフトウェアの供給先にその情報を提供するとともに、関連する制度の指定に従って当局に報告する。	
取組例	• ソフトウェアに含まれる脆弱性とコンポーネントを特定し、ソフトウェア構成情報とともに文書化する。（CRA Annex Ⅰ 2) • 脆弱性に対処するために必要な、ソフトウェアの修正セキュリティアップデートを作成する。（CRA Annex Ⅰ 2) • セキュリティアップデートが利用可能になった後、脆弱性の説明、影響を受けるソフトウェアをユーザが特定できる情報、脆弱性の影響、脆弱性の重大性、及びユーザが脆弱性を修正するのに役立つ情報を含む、修正された脆弱性に関するセキュリティ勧告を作成し、開示する。（CRA Annex Ⅰ 2) • セキュリティ勧告に示されたパッチ・対策手続を含むセキュリティアップデートは、完全性と信頼性を確保するためにセキュアに配布する。（S(3)-2.3, CRA Annex Ⅰ 2) • インシデントの影響を軽減するために、ユーザが実施可能な是正措置について、不当な遅滞なくユーザに通知する。 • 脆弱性届出制度に基づいて情報提供を受けた脆弱性については、情報セキュリティ早期警戒パートナーシップガイドラインを踏まえた対応を行う。 • セキュリティ分析官がプログラムを分析し、起こり得る脆弱性について報告できるようにする。（sp800-218 RV.1.3 notional implementation example） • 開示した脆弱性情報及びセキュリティアップデートへの容易なアクセスと取り込みをサポートするセキュアな配信メカニズムを整備する。 • 脆弱性を迅速に是正するために、脆弱性によるリスク評価、対応の優先順位付け、是正のサイクルを適性かつ効率化するアプローチを確立するとともに、顧客の更新戦略に基づいて設定可能なソフトウェアの自動的な更新メカニズムを提供する。 • 報告された一般的な脆弱性、ゼロデイ脆弱性の報告、実際に悪用されている脆弱性、及び複数の関係者とオープンソースソフトウェアコンポーネントが関与する重大な進行中のインシデントを処理するセキュリティ対応のプレイブックを用意する。（sp800-218 RV.1.3 notional implementation example）

■ システム・サービスのソフトウェアの脆弱性対応の課題と対応

システム・サービスを構成するソフトウェアの脆弱性対応には様々な課題が考えられ、状況に応じた対策を講じることが求められる。

- サービスを提供するシステムの場合、サービスが停止できるタイミングが限られるため、パッチ適用などを即時実施することが困難な場合がある。対策としては、パッチ適用は（許容される場合）定期的なバージョンアップとともに実施し、それまではサービスの周辺対策によって脆弱性が悪用できないように保護するなどが考えられる。
- 古いソフトウェアパッケージを維持する場合、互換性等の観点からパッチ適用が困難なため、あらかじめサポート期間を設けるなどして対策が可能なソフトウェアパッケージへの移行を促す。
- EOL（End of Lifecycle）が短いソフトウェアを採用する場合、技術の追随とバージョンアップが困難なケースもある。このような場合を見越して、中長期的にコンテナ等の技術を活用しながら、手間をかけずにバージョンアップするための環境整備を進めることなどが対策として考えられる。

セキュリティ対策に伴うソフトウェアの修正には、委託先へ修正を依頼することを含めて、必要なリソース確保が課題となる。例えば、コストの準備が十分できず、本番環境にパッチを適用する前の検証環境が準備できないことは、リスクを高める要因にもなる。このようなリスクを考慮して、必要な内容とコストの手当を行い、適切な検証環境を整備する。セキュリティ対策の予算化を申請する際には、経営層に対して、コストだけでなくリスクとその影響を説明することが不可欠であり、リスクと影響を踏まえたコストの必要性・妥当性の根拠を示すことが望まれる。その際、関係者と費用配分の考え方についても事前に合意しておくことが肝要である。（関連する要求事項：S(3)-2 全般）

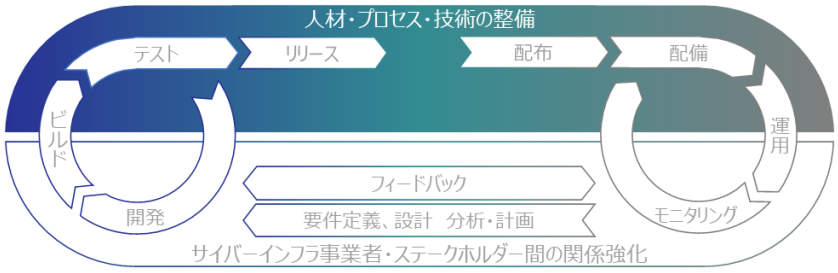
S (3)-3	開発者 供給者 運用者 顧客	
対処結果を組織のプロセス改善に活用 ソフトウェアに発見された問題の根本原因が再発しない、若しくはその可能性を低減するよう、脆弱性に基づき、開発と運用のプロセスを見直す。		

□ S(3)-3.1 根本原因の特定 根本原因を決定するために、識別された脆弱性を分析し、プロアクティブに対策する。	
取組例	• 特定済みの識別された脆弱性を分析し、発見された問題の根本原因を分析し、記録する。（sp800-218 RV.3.1 task/notional implementation example） • 特定のセキュアコーディング規則が一貫して守られていないなどのパターンを特定するために、原因となり得る事象の発生を自動的に検出する仕組みをツールチェーンに追加するなどして、根本的な原因が何かを時間をかけて分析する。 • 自動化されたツールを使用して、可読コードがリポジトリにチェックインされるときに検証されたセキュアでないソフトウェア慣行を継続的に観察する。 • 特定の系統の脆弱性を根絶するために、類似の脆弱性がないかソフトウェアをレビューし、外部からの報告を待たずに事前に修正する。
□ S(3)-3.2 プロセス改善 ソフトウェアの更新又は作成された新しいソフトウェアにより、根本原因の再発を防止又はその可能性を低減するために、ソフトウェアライフサイクル全体の開発と運用のプロセスをレビューし、必要に応じて見直す。	
取組例	• 根本原因の再発を防止・低減するために、自組織への影響の調査、及び脆弱性の緩和策を実施する。 • 根本原因の分析を通じて学んだ教訓を元に、ソフトウェアライフサイクル全体の開発と運用のプロセスをレビューし、必要に応じて更新する。（sp800-218 RV.3.4 task/notional implementation example） • 特定した根本原因及び是正処置は、開発者の能力向上につながるトレーニングに活用する。

■ 脆弱性対応のために開発と運用が協調すべきこと

開発したソースコードを更新した場合やミドルウェアを更新した際にソースコードに影響が出る場合などに備えて、対象ソフトウェアの十分な動作検証が必要になる。開発と運用を異なる組織で構成する企業では、この動作検証における役割分担などが課題となる。特に、運用フェーズに引き継ぐ際には、運用体制を十分なスキルやノウハウを持つ人員で構成するように整備することが求められる。また、開発組織が作成した構成管理情報を、運用組織が引き継いで管理するための標準（テンプレートやフレームワーク）の整備も必要となる。（関連する要求事項：S(3)-3.2）

（4）人材･プロセス･技術の整備

S (4)-1	開発者 供給者 運用者 顧客	

人材：経営層のコミットメントと人員の整備

ソフトウェアのライフサイクル全体を網羅した役割と責務を定義する。セキュア開発に対する経営層のコミットメントを周知し、セキュリティ対策のための人材を確保し、セキュアな開発･運用に関連する全要員に、要員の習熟度と役割に応じたトレーニングを提供し、定期的に見直す。

□　S(4)-1.1 役割と責務の定義

ソフトウェア開発ライフサイクルを網羅する役割と責務を定義する。

取組例	• ソフトウェア開発チームにセキュリティの役割を統合する。（statement から派生） • サプライチェーン管理及びリスク管理を、ソフトウェア開発プロセスに統合するように、役割と責務を整備する。 • サイバーセキュリティスタッフ、セキュリティチャンピオン、プロジェクトマネージャとリーダ、上級管理職、ソフトウェア開発者、ソフトウェアテスター、ソフトウェア保証リーダとスタッフ、製品オーナー、運用とプラットフォームエンジニア、調達と在庫管理リーダとスタッフなど、ソフトウェア開発ライフサイクルに関与する全ての役割と責任を定義する。

□　S(4)-1.2 経営層のコミットメント

全要員に対してセキュア開発に対する経営のコミットメントを周知し、組織にとってのセキュアな開発･運用の重要性を教育する。

取組例	• 経営層（経営トップ、上級マネジメント、承認権限者など）によるセキュアな開発･運用に対するコミットメントを、開発･運用に関連する役割と責任を持つ全ての人に周知･理解させるための教育を実施する。（sp800-218 PO.2.3 task） • セキュアなソフトウェア開発プロセス全体を担当するリーダ又はリーダシップチームを任命し、リスク及びリスク軽減に対する認識を高めるための教育を実施する。（sp800-218 PO.2.3 notional implementation example） • セキュアな開発･運用を実現するための経営層の取組と、組織としてのセキュアな開発･運用の重要性について、開発･運用関連の役割と責任を持つ全ての担当者に教育を実施する。（sp800-218 PO.2.3 notional implementation example）

□　S(4)-1.3 役割と責務の同意

各要員が、役割と責務を認識･同意していることを確認する。

取組例	• 役割を任命した個人、及び役割と責任に関する近々の変更について影響を受ける個人に対して教育を実施し、個人が役割と責任を理解しそれに従うことに同意することを確認する。（sp800-218 PO.2.1 notional implementation example）

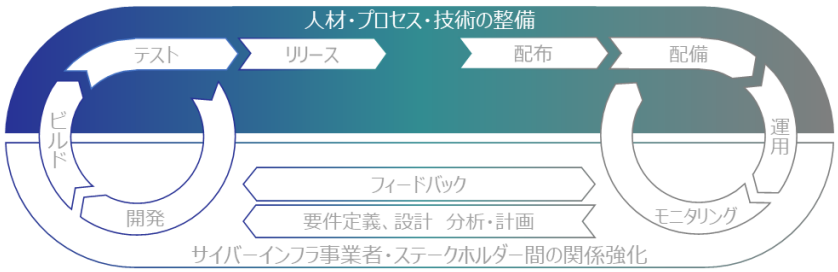
□ S(4)-1.4 各役割のトレーニング 各役割のトレーニング計画を作成し、全要員が習熟度と役割に応じてトレーニングを実施できるように提供する。	
取組例	• セキュアな開発に寄与する責任を持つ全ての要員に対して、役割ベースのトレーニングを提供する。（sp800-218 PO.2.2 task） • セキュリティ対策に関わる要員の候補者とその要員の経歴を継続的に確認する。 • ソフトウェア開発者には、コードの所管を割り当て、セキュアなソフトウェア開発手法（標準化された開発手法、自動化を活用した開発ツールの使用法、AI を活用したプログラミング手法を含む）、セキュアコーディング標準、ロール固有のベストプラクティス、AI サポートによる自動化支援（品質向上）の手法などを理解・共有するためのトレーニングを計画する。 • トレーニング計画には、習熟度と役割に応じた目標と成果の測定プロセスを含める。
□ S(4)-1.5 役割とトレーニングの見直し 役割やトレーニングは定期的に見直す。	
取組例	• 定義された役割と責任を定期的（年次等）にレビューし、必要に応じて更新する。（sp800-218 PO.2.1 task） • 要員の習熟度と役割ベースのトレーニング及びその成果の測定結果を定期的にレビューし、必要に応じてトレーニングを更新する。(sp800-218 PO.2.2 task/notional implementation example) • DevSecOps 開発パラダイムをベースとする CI/CD パイプラインなど新たな開発手法やツールチェーンを実装して利用する前に、ソフトウェアサプライチェーンのセキュリティ保証対策とツールの使用法に関するトレーニングを見直す。

■ 開発者へのセキュア開発に関するトレーニングの重要性

自動化を活用することで人の労力を減らし、ライフサイクル全体の取組の正確性、再現性を向上させる。自動化による効果を得るためには、要員によるアクションは自動化を前提とする必要があり、各役割のトレーニングも自動化効果を最大化するように調整していく必要がある。

開発者は、ソフトウェアのセキュリティ対策に直接関与することから、各役割のトレーニングの中でも開発者へのセキュア開発に関するトレーニングは特に重要である。ウォーターフォール開発の場合、設計の上流工程でのセキュリティ対策の見逃しは、下流工程からの手戻りに多くのコストが発生するリスクが残る。一方、アジャイル開発において、専任者や十分にトレーニングされた要員の配置が困難であると、開発チェックを必要なフェーズに割り当てることが困難になり、ウォーターフォール開発と比較して、技術者個人の能力や判断による意図しないライブラリを使用するなどのリスクが懸念される。

（関連する要求事項：S(4)-1.4）

S (4)-2	開発者 供給者 運用者 顧客	

プロセス：開発ポリシーの確立と法令順守

法令を遵守し、組織の開発インフラ及びプロセスに関するセキュリティポリシーを文書化・維持し、セキュリティ確保に必要な予算を確保する。

□　S(4)-2.1 ソフトウェア開発ポリシーの定義

ソフトウェア開発のインフラ及びプロセスの全てのセキュリティ要件（EOL に係る要件を含む）を特定し、法令遵守の下 SDLC 全体を通じて維持するためのセキュリティポリシーを定義する。

取組例	• SDLC 全体でソフトウェア開発インフラとそのコンポーネント（開発エンドポイントを含む）を保護し、そのセキュリティを維持するためのポリシーを定義する。（sp800-218 PO.1.1 notional implementation example） • SDLC 全体でソフトウェア開発プロセスとそのコンポーネント（他のサードパーティのソフトウェアコンポーネントを含む）を保護し、そのセキュリティを維持するためのポリシーを定義する。（sp800-218 PO.1.1 notional implementation example） • 組織の開発インフラ及びプロセスを維持、復旧するための計画（情報セキュリティ管理策、支援システム、既存の情報セキュリティ管理策を維持するためのプロセス、維持できない情報セキュリティ管理策を補う管理策）を策定し、実施の試験、レビュー及び評価を行う。 • 国内及び事業を行う地域の法的要求事項、業界のベストプラクティスや標準に対する社内ポリシーの適合性をチェックし、実施するための方針を整備する。 （システム・サービスのソフトウェアの場合） • システム・サービスの取得に関するポリシー（目的、範囲、役割、責務、組織間の調整、コンプライアンスへの対処など）を定義する。

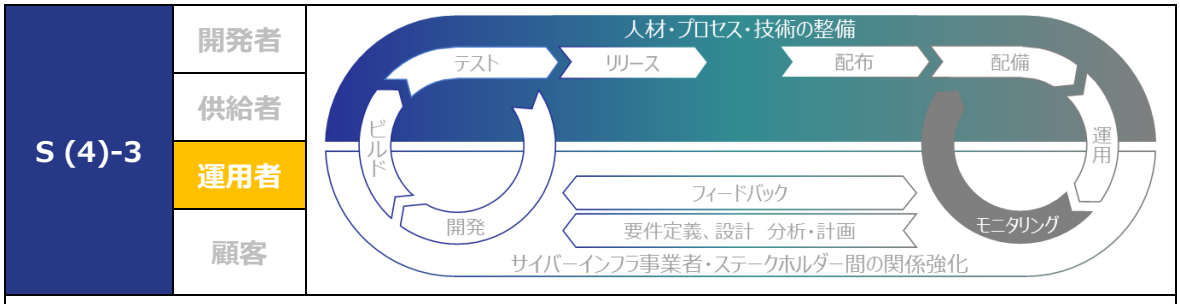
□ S(4)-2.2 ソフトウェア・セキュリティポリシーの定義と維持 組織が開発するソフトウェアが満たすべき全てのセキュリティ要件を規定したポリシーを定義し、これらの要件を SDLC 全体にわたって維持する。	
取組例	• ソフトウェアが満たすべきセキュリティ要件には、リスクを緩和するアーキテクチャ及び設計要件、ライフサイクルの適切なゲート（チェックポイント）における検証フロー要件、技術スタック（言語、環境、配備モデル等を含む）のリスク対応要件などを含める。 • ソフトウェアのリリースごとにアーカイブすべきもの（コード、ソフトウェアパッケージファイル、サードパーティライブラリ、設定、文書、データインベントリ等の関連成果物）、及び SDLC モデルやソフトウェアの寿命（EOL）、その他の要因に基づいて保存する必要がある期間などをポリシーとして定める。 • 定期的、あるいは追加の要件やインシデントの発生（組織、リリースソフトウェアの脆弱性発見を含む）に併せてポリシーを見直し、関係者に周知する。 • 要件の例外要求の処理プロセス（承認された例外の定期的なレビューを含む）、及びサプライチェーンの弱点を特定し対処するプロセスを確立する。 • 要件には、将来的にパッチを適用できるようなアーキテクチャとすることを含める。 （システム・サービスのソフトウェアの場合） • セキュリティ要件を検討する際には、内部（組織の方針、ビジネス目標、リスク管理戦略など）及び外部（適用される法令や規制など）からの要件を含める。
□ S(4)-2.3 費用認識の共有と予算化 ポリシーに基づいてセキュリティを確保するために必要な予算を確保する。	
取組例	• サイバーセキュリティに関する残存リスクを許容範囲以下に抑制するための方策を検討し、その実施に必要となる資源（予算、人材等）を確保した上で、具体的な対策に取り組む。（サイバーセキュリティ経営ガイドライン v3 指示 3） （システム・サービスのソフトウェアの場合） • 関係者間で費用認識の共有を促進する前提として、脆弱性の放置が将来的な負債（サイバー攻撃による損害など）につながることを共通の経営リスクとして把握する。

■ ポリシーとして整備すべき重要なポイント

ポリシーとして整備すべき重要なポイントとしては以下が挙げられる。

- 外部組織から提供を受けた情報の自組織内での取扱いに係るプロセスを決定する（情報の受取窓口の整備、受け取った情報と自社資産との関わり度合いや緊急度などの判断、対処方針の策定などの情報取扱いなど）。
- 自組織で活用する情報資産を管理する（自組織の状況を踏まえたリスク管理に基づき、受け取った情報の優先度を決定する）。
- セキュリティ投資などの業界別統計情報を整理する（経営層が、他社との比較を通じて自社取組の位置付けを把握することができ、取組を一層推進させることが期待される）。
- セキュリティ投資の有効性に関する評価指標を定義する（投資したツールのレポーティング機能を活用し投資効果の可視化を進めるなど、顧客と事業者間で効果を定量的に共有できることが期待される。ただし、わかりやすい「投資・コストの見える化」は、現時点では検討課題となることが多い）。

達成すべき数値目標が整理されれば、より現実的で精緻な費用算出が可能となることから、CISAによる拘束力のある運用指令（BOD）や IT ベンダーが共同で策定したセキュリティチェックリストである「実用最小限の製品（MVSP）」のような、セキュリティ要求の公的な数値目標やセキュリティベースラインへの適合を根拠に、コスト見積りの妥当性を高める取組なども、今後は重要になると考えられる。（関連する要求事項：S(4)-2 全般）

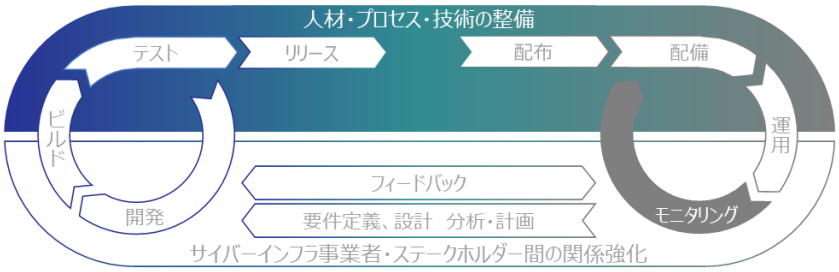


プロセス：運用ポリシーの確立と法令遵守

法令を遵守し、ソフトウェアを適用したサービス運用インフラ及びプロセスに関する全てのセキュリティポリシーを文書化し、維持する。

□ **S(4)-3.1 ソフトウェアサービス運用ポリシーの定義** ソフトウェアを適用したサービス運用インフラ及びプロセスの全てのセキュリティ要件（EOS 及び廃棄に係る要件を含む）を特定し、法令遵守の下 SDLC 全体を通じて維持するためのセキュリティポリシーを定義する。	
取組例	• SDLC 全体でソフトウェアを適用したサービス運用インフラ、プロセス、及びそのコンポーネント（他のサードパーティのソフトウェアコンポーネントを含む）を保護し、そのセキュリティを維持するためのポリシーを定義する。（sp800-218 PO.1.1 notional implementation example から派生） • 組織のサービス運用インフラ及びプロセスを維持、復旧するための計画（情報セキュリティ管理策、支援システム、既存の情報セキュリティ管理策を維持するためのプロセス、維持できない情報セキュリティ管理策を補う管理策）を策定し、実施の試験、レビュー及び評価を行う。 • 国内及び事業を行う地域の法的要求事項、業界のベストプラクティスや標準に対する社内ポリシーの適合性をチェックし、実施するための方針を整備する。 • 企業規模や業種を踏まえたリスク分析に基づき、保護対策を実施するための方針を整備する。 • 顧客が他のデジタルサービスと連携し、必要に応じて同様のサービスを提供する他のプロバイダに移行することを可能にするプロセスも含む。
□ **S(4)-3.2 サービスのセキュリティポリシーの定義と維持** ソフトウェアを適用したサービスが満たすべき全てのセキュリティ要件を規定したポリシーを定義し、これらの要件を SDLC 全体にわたって維持する。	
取組例	• セキュリティ要件を検討する際には、内部（組織の方針、ビジネス目標、リスク管理戦略など）及び外部（適用される法令や規制など）からの要件を含める。 • 定期的、あるいは追加の要件、インシデント発生（組織、リリースしたソフトウェアサービスの脆弱性発見を含む）に併せてポリシーを見直し、関係者に周知する。 • 要件の例外要求の処理プロセスを確立し、それに従う（承認された全ての例外の定期的なレビューを含む）。 （システム・サービスのソフトウェアの場合） • 関係者間で費用認識の共有を促進する前提として、脆弱性の放置が将来的な負債（サイバー攻撃による損害など）につながることを共通の経営リスクとして把握する。
□ **S(4)-3.3 運用ポリシーに基づく監査** ポリシーに基づくガバナンスにより、サービス運用インフラ及びプロセスの保護、及びサービスのセキュリティ要件が SDLC 全体にわたって維持されていることを監査により確認する。	
取組例	• 監査が形骸化しないよう、体制を整備し予算を確保する。最重要かつ必須の監査項目の監査を通じて受けた指摘に対応することで、ガバナンスの確立を図る。 • ガバナンスの観点から、監査人の技能及び能力の検証のためのメカニズムを整備する。

※S(4)-2 の参考情報「ポリシーとして整備すべき重要なポイント」を参照。

S (4)-4	開発者 供給者 運用者 顧客	

プロセス：開発・運用基準の策定

ソフトウェアの開発に関わるセキュリティ上の確認基準を定め、基準の裏付けに必要な情報を収集し、適合するためのプロセス、仕組みを実装する。ライフサイクル全体を通じて適合状況を追跡する。

□　S(4)-4.1 セキュリティ確認基準の定義と追跡

ソフトウェアのセキュリティ確認基準を定義し、SDLC 全体を追跡する。

取組例	• セキュリティエンジニアリングに基づいたソフトウェアのセキュリティ評価指標（重要業務評価指標（KPI）、重要リスク指標（KRI）脆弱性深刻度スコアなど）を定義し、開発プロセスに導入する。 • 過去のプロジェクトの脅威、脆弱性情報、及び教訓をセキュリティ確認基準に取り込む。 • 品質指標（コンパイラーエラーなしなど）を併せて定め、品質基準を満たしていることのエビデンスを残す。 • ワークフロー及び追跡システムの一部として、セキュリティ点検の承認、拒否、及び例外要求を記録する。（sp800-218 PO.4.1 notional implementation example） • 開発ワークフローの結了判断にセキュリティの確認基準を組み込み、成果物の準拠状況を確認し、確認結果を、開発プロセス全体の改善に活用する。 （システム・サービスのソフトウェアの場合） • 重要なサービスを支えるシステムにおいて、常に有効性を評価できるような KPI を導入する。

□　S(4)-4.2 セキュリティ確認基準に基づく意思決定のサポート

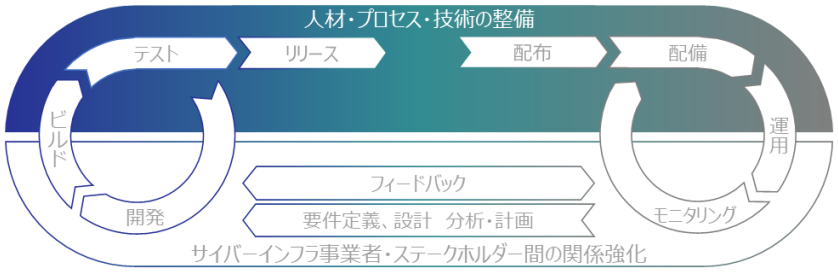
セキュリティ確認基準に基づく意思決定をサポートするために必要な情報を収集し保護するためのプロセスや仕組みなどを実装する。

取組例	• ツールチェーンを利用して基準クリアの確認に必要なデータを収集し、セキュリティ上の意思決定に活用するプロセスを整備する。（S(4)-4.2） • 基準をサポートする情報の生成と収集をサポートするために、必要に応じて追加のツールを配備する。（sp800-218 PO.4.2 notional implementation example） • 承認された担当者のみが収集した情報にアクセスできるようにし、情報の変更又は削除を防止する。（sp800-218 PO.4.2 notional implementation example） • 意思決定プロセスを自動化し、これらのプロセスを定期的に見直す。

□　S(4)-4.3 セキュリティ確認基準に基づく監査

セキュリティ上の確認基準への適合を遵守するためのガバナンスにより、SDLC 全体を追跡し意図する効果を得ていることを監査による確認する。

取組例	• 監査が形骸化しないよう、体制を整備し予算を確保する。最重要かつ必須の監査項目の監査を通じて受けた指摘に対応することで、ガバナンスの確立を図る。 • ガバナンスの観点から、監査人の技能及び能力の検証のためのメカニズムを整備することを含める。

S (4)-5	開発者 / 供給者 / 運用者 / 顧客	

技術：セキュアな開発ツールの整備

ソフトウェアの開発ライフサイクル全体のリスクを分析し、開発ツールにセキュリティ対策を実施する。

☐ S(4)-5.1 ツールとツールチェーンの指定

特定されたリスクを軽減するために有効なツールを特定し、どのツールチェーンに含めることが必須若しくは必要であるか、及びツールチェーンのコンポーネントを相互に統合する方法を指定する。

取組例	• ツールチェーンのカテゴリを定義し、各カテゴリに使用する必須ツール又はツールの種類を指定する。（sp800-218 PO.3.1 notional implementation example） • セキュリティツールをプロセスとツールチェーンに統合する。 • ツール間で受け渡される情報と使用されるデータ形式を定義し、ツールチェーンや既存のソフトウェア開発プロセス及びワークフローと統合する。 • ビルドの再現性を実現するなど、目的に応じてツールの管理やオーケストレーションに自動化技術を採用する。

☐ S(4)-5.2 ツールとツールチェーンの配備、運用、保守

セキュリティ慣行に従ってツールとツールチェーンを配備、運用、及び保守する。

取組例	• ツール及びツールチェーンにより組織が定める要件を満たしていることを定期的にレビューする。 • ツールのセキュリティ実現への効果を評価し、有効性を判断する。期待する効果としては、コードベースの構成を使用するツールチェーンの実現可能性、ビルドの再現性、脆弱性対応などのアップグレード対応、出所情報など整合性の検証に必要な情報の有無、ツールチェーンの自動化への対応、過去のプロジェクトの脅威、脆弱性情報及び教訓への対応など、適用目的に合わせて定める。 • ツールの出所、整合性、脆弱性や新機能を継続的に調査及び検証し、必要に応じてツールを更新する。 • ツールを評価する際には、脅威モデリングと脆弱性分析を実施する。 • ツールのセキュリティ対策として、セキュアなサードパーティ製ソフトウェアのツールチェーンとの互換性ライブラリを使用する。

☐ S(4)-5.3 ツール構成と証跡生成

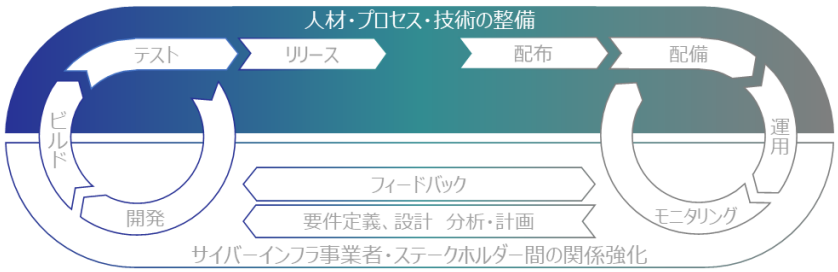
組織によって定義されたセキュアなソフトウェア開発の慣行のサポートに関する証跡を生成するようにツールを構成する。

取組例	• ツール利用時のログを継続的に生成・監視し、ポリシー違反や異常な動作を含む運用上及びセキュリティ上の潜在的な問題を発見する。 • 継続的な改善の目的で実行されるセキュアな開発関連の活動の監査証跡を作成するために、既存のツール（ワークフロー追跡、問題追跡、バリューストリームマッピングなど）を使用する。 • 収集した情報を監査する頻度を決定し、必要なプロセスを実装する。（sp800-218 PO.3.3 notional implementation example）

■ 開発ツールに対するセキュリティ慣行の補足

開発ツールに対するセキュリティ慣行について、以下に補足的なポイントを示す。（関連する要求事項：S(4)-5.2）

- 開発環境（開発ツールを含む）として OSS などのサードパーティ製ソフトウェアコンポーネントを使用する際には、脆弱性情報の収集や出所の確認を実施する。
- 受託開発の場合、委託先が保有する開発機材の利用制限などの統制上の課題をクリアにする。
- 構成や設定を一元的に管理・活用するツールを使い、使いこなす技術を開発チームとして醸成する。

<table>
<tr><td rowspan="4">S (4)-6</td><td>開発者</td><td rowspan="4"> 
</td></tr>
<tr><td>供給者</td></tr>
<tr><td>運用者</td></tr>
<tr><td>顧客</td></tr>
<tr><td colspan="3">技術：セキュアな開発環境の整備
ソフトウェアの開発ライフサイクル全体のリスクを分析し、開発に関わる環境を保護強化する。</td></tr>
<tr><td colspan="3">☐ S(4)-6.1 環境の分離保護
ソフトウェア開発に関係する各環境を分離して保護する。</td></tr>
<tr><td>取組例</td><td colspan="2">• 開発環境と運用環境を分離する。
• ソフトウェア開発のための環境やネットワークを分離する（開発環境、ビルド環境、テスト環境、配付環境など）。
• 本番環境のソフトウェア及びサービスの、非本番環境からの使用を最小限に抑える。（sp800-218 PO.5.1 notional implementation example）
• 環境間及び各環境内のコンポーネント間の認可とアクセスについて、信頼関係を定期的にログに記録、監視、及び監査する。（sp800-218 PO.5.1 notional implementation example）
• 環境の動作のアーティファクトを生成するために、環境の分離と保護に関連するセキュリティコントロールとその他のツールを構成する。（sp800-218 PO.5.1 notional implementation example）
• 各環境に展開されたコンポーネントの脆弱性を継続的に監視し、環境ごとにリスクベースの対処を行う。
• ゼロトラストアーキテクチャに準拠した環境のホスティングインフラストラクチャを保護するための対策を構成し実装する。</td></tr>
</table>

□ S(4)-6.2 開発用エンドポイントの保護 リスクベースのアプローチを使用して開発関連のタスクを実行するために、各開発者向けのエンドポイントを保護、強化する。	
取組例	• ソフトウェア開発のための環境やネットワークを分離するために、リスク分析に基づき適切なシステムの保護方法（適切なアーキテクチャ、技術など）を選定する。 • 開発環境・開発用エンドポイント（ソフトウェアの設計者、開発者、テスター、ビルダー向けなど）のセキュリティ保護を堅牢化し（環境ごとに多要素認証、リスクベース認証、条件付アクセスを使用、機密データの標準準拠に基づく暗号化、など）、特権アクセスやアクセス試行などを監視し、サイバーインシデントを検出、対応、回復する。 • 開発環境は、ユーザやサービスが必要とする最小限の機能を提供し、最小特権の原則を実施するように構成する。 • 開発環境への接続を厳格に制限する（インターネットへのアクセスを必要最小限に制限することを含む）。 • 構成管理、変更管理、及び開発・維持環境及び管理者権限の保護などのハードニングを実施し、悪意のあるソフトウェアの作成、混入を防止する。 （システム・サービスのソフトウェアの場合） • 開発・事務作業の効率化のため、構成管理を含む共通開発基盤を用意し、委託事業者に提供する（事業部門に費用負担が発生することを考慮）。

コラム　ソフトウェア開発運用における AI 活用のメリット

2023 年の米国の調査では、米国を拠点とする開発者の 92%は、既に仕事の内外で AI コーディングツールを使用しており、生成 AI がソフトウェア開発運用の現場でも広く活用されていることが確認できる。

また、米国でのリサーチ会社による GitHub Copilot ユーザの分析結果では、最初の 1 年間で、平均して GitHub Copilot からのコード提案を 30%近く受け入れ、この受入れによって生産性が向上したと報告されている。さらに、開発者がツールに慣れるにつれて、受入れ率が上昇していることが判明しており、ユーザが GitHub Copilot を使ったソフトウェア開発に慣れるにつれて、開発者の生産性に影響を与え続ける可能性が大きいことが示唆されている。

また、別の調査では、経験の浅い開発者ほど GitHub Copilot の恩恵を受けることも報告されており、生成 AI をソフトウェア開発運用にうまく活用することが望まれる。

コラム　ソフトウェア開発運用における AI 活用の負の側面

米国スタンフォード大学の調査では、AI アシスタントに権限を与えすぎる（例えば、パラメータ選択を自動化する）と、セキュリティ脆弱性への取組の熱意が低下する可能性があること、AI アシスタントは、開発者がライブラリドキュメントから API やセキュアな実装の詳細を注意深く検索する積極性を低下させる可能性があると報告している。

セキュリティ脆弱性の要因の一部が、不適切なライブラリの選択や使用に関連することを踏まえると、開発者は AI アシスタントの取扱い（プロンプトを含む対話方式等）に注意を払うとともに、生成物のテスト方法等についての学びが必要であると考えられる。

コラム	AI 活用における倫理的・法的・社会的な課題への対応

1980 年代の「ヒトゲノム計画」において、ELSI（Ethical, Legal and Social Implications）という取組が進められた。これは、技術的な課題とともに、倫理的・法的・社会的な影響を併せて対応していく考え方を示すものであり、現在、急速に進展する AI 活用においても重視されるべき観点である。AI 活用の ELSI で議論が進められている一つに、「信頼できる責任ある AI」への取組がある。

例えば、機械学習モデルの精度には、学習データの量とともに質（バリエーション）が大きく影響するため、データの偏りがないか、予測したい事象への網羅性は十分か、ノイズが含まれていないか、といった性能やセキュリティに影響するデータの質に関する特性が重要であるが、これらの学習データの収集、あるいは機械学習モデルの活用の前に、倫理的（人権侵害など）、法的（著作権、不正競争防止、営業秘密、個人情報とプライバシーなど）、社会的（AI の公平性、透明性、説明責任への対応など）な影響をリスクと捉え、「信頼できる責任ある AI」の開発・維持に対して適切なリスクマネジメント体制を整備することが求められる。

EU では、AI を包括的に規制する法案（Artificial Intelligence Act）について、2023 年 12 月時点で暫定的な合意が得られた。今後は EU 域内で開発・利用される AI システムに対してリスクベースの対応が求められ、違反への高額な罰金が定められるなど、各主体においても対策が迫られる状況になってきた。

また、米国の NIST では、AI に関連する個人、組織、社会へのリスクをより適切に管理するためのフレームワーク（NIST AI Risk Management Framework（AI RMF））を開発し、Trustworthy and Responsible AI Resource Center（NIST AIRC）がその活用のサポートを開始している。

https://airc.nist.gov/Home

コラム	生成 AI の安全なソフトウェア開発のプラクティス例

米国の NIST は、SP800-218 で規定するセキュアなソフトウェア開発のフレームワーク（SSDF）の派生として、SP800-218A（生成 AI とデュアルユースの基盤モデルのための安全なソフトウェア開発プラクティス）を公開している。これは、AI モデル開発に特有のタスク、プラクティス、推奨事項等を SSDF に補足し、AI モデルの開発者、AI システムの開発者、及び AI システムの購入者が AI モデルの安全なソフトウェア開発手法をより深く理解するために有用な情報として提供するものである。

以下のような事項が、SSDF に補足されている。

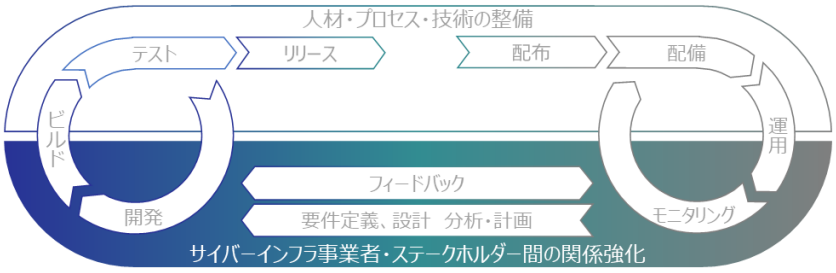
- データの保護（PS.1.2 タスクとして追加）
  全てのトレーニング、テスト、ファインチューニング、アライニング（調整）のためのデータを不正なアクセスや変更から保護する。
- モデルの保護（PS.1.3 タスクとして追加）
  全てのモデルの重みと構成パラメータのデータを不正なアクセスや変更から保護する。
- ソフトウェアアーティファクトのサプライチェーンレベル（SLSA）を通じた SBOM（PS.3.2 タスクの変更）
  各ソフトウェアリリースの全てのコンポーネントの出所データを収集、保護、維持、共有する(例:SLSA を通じた SBOM)。
- 実行パフォーマンスと動作の継続的監視（PO.5.3 タスクとして追加）

ソフトウェア開発環境におけるソフトウェアの実行パフォーマンスと動作を継続的に監視し、潜在的な不審なアクティビティやその他の問題を特定する。

- データの分析（PW.3.1 タスクとして追加）
  AI モデルのトレーニング、テスト、ファインチューニング、アライニングの目的でデータを使用する前に、データポイズニング、バイアス、均質性、タンパリングの兆候がないかデータを分析し、必要に応じてリスクを軽減する。
- データの出所の追跡（PW.3.2 タスクとして追加）
  AI モデルに使用される全てのトレーニング、テスト、ファインチューニング、アライニングのためのデータの出所を追跡する。
- 敵対的サンプル（PW.3.3 タスクとして追加）
  攻撃の検出を向上させるために、トレーニングデータとテストデータに敵対的なサンプルを含める。

これらのタスク及びプラクティス例を参考に、生成 AI モデルの開発をセキュアに行うための体制、プロセス、手続の策定に活用することができる。
https://csrc.nist.gov/pubs/sp/800/218/a/final

（５）サイバーインフラ事業者・ステークホルダー間の関係強化

S (5)-1	開発者 供給者 運用者 顧客	

情報連携のための組織体制

ソフトウェアの製品及びサービスのセキュリティを改善するために、民間企業同士、関係当局、専門組織との情報連携のための組織体制を構築する。

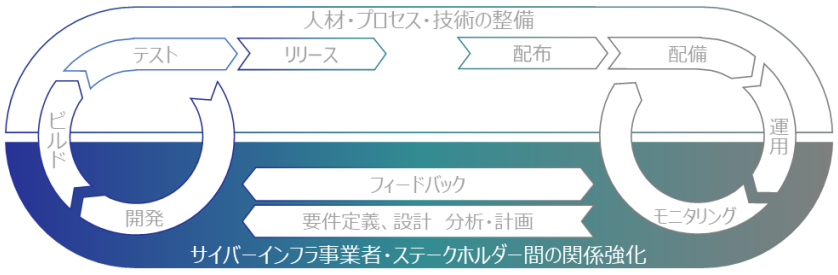
☐　**S(5)-1.1　情報連携のための組織体制の構築** ソフトウェアの製品及びサービスのセキュリティを改善するために、民間企業同士、関係当局、専門組織との情報連携のための組織体制を構築する。	
取組例	• ソフトウェアに関連する法的要求事項、業界のベストプラクティス、及び標準に対する社内ポリシーの適合性をチェックし、実施するための方針を整備する。 • ソフトウェアのセキュリティに関する民間企業同士、関係当局、専門組織との情報連携を活用するために、CSIRT と連絡責任者（POC）を設置するとともに、関係者のスキルアップ、コミュニケーションツールの普及による効率化を進める。
☐　**S(5)-1.2　重要なセキュリティ関連情報の提供** 業界固有の必須かつ重要なセキュリティ関連情報を選別・識別して、サプライチェーン先に提供する。	
取組例	• サイバーインフラ事業者（開発者、供給者、運用者）と顧客（発注者）の間に、ソフトウェアセキュリティに関する情報連携の仕組みを構築する。供給者は、開発者と顧客の間の連絡役又は仲介者として情報連携に寄与する。 • 同じ被害が繰り返されないように、被害事例（特に脅威と対処の情報）を積極的に共有する。 （システム・サービスのソフトウェアの場合） • 顧客のセキュリティ向上のための関係ベンダー間での情報共有の仕組みを構築し、インシデント発生時の対応は、契約に基づいて実施する。
☐　**S(5)-1.3　脆弱性情報の通知サービスの利用** 効率的に脆弱性情報の共有を図るため、脆弱性情報の通知サービスを利用する。	
取組例	• 発見された脆弱性がどのような攻撃と影響を受けるのか等の情報連携は、業界団体などが運営する仕組みを活用する。 • 設定の脆弱性等の情報連携は、専門組織などが運営する制度・仕組みを活用する。 • 業界団体経由で、業界の共通的な構成管理ツールなどの推奨情報の利用を促進する。

■ 関係者間で情報共有する仕組みへのニーズ

顧客のセキュリティ向上に資するインシデント発生時の対応に関する契約の在り方など、専門組織間、関係ベンダー間又は関係者間で情報を共有化できるような仕組みへのニーズが高まっている。以下にその例を示す。

- ソフトウェアの設定に起因する脆弱性などの情報連携は、現状では企業間と人のつながりの中での個別共有に留まる場合も多いため、公的機関（例えば IPA など）への届出制度のような共有の仕組みを活用する。
- 発見された脆弱性がどのような攻撃と影響を受けるのか、といった情報は、一般的に開発者は入手が難しいため、「サイバー攻撃被害に係る情報の共有・公表ガイダンス」（2023 年 3 月 8 日 サイバー攻撃被害に係る情報の共有・公表ガイダンス検討会策定）などを利用できるようにする。（「情報の共有」とは、非公開にて情報共有活動の場や専門組織との間で行われる、主にサイバー攻撃の手法等に関する技術的情報のやりとりのことを指す。一方、「公表」とは、被害組織が、自組織が受けたサイバー攻撃被害の状況や対応内容について広く外部に示すものを意図する。なお、同ガイダンスは、対策の重要度や期限を付すことで利便性が向上すると考えられる。）
- サイバー攻撃が高度化し、単独組織による攻撃の全容解明は困難となっている中、被害拡大の防止等の観点から、「情報の共有」は被害組織自身ではなく、被害組織を支援する専門組織を通じて、他の専門組織等との間で速やかに行われることが重要である。
  「サイバー攻撃による被害に関する情報共有の促進に向けた検討会」における提言の趣旨に添って、「攻撃技術情報の取扱い・活用手引き」及び「秘密保持契約に盛り込むべき攻撃技術情報等の取扱いに関するモデル条文案」（2024 年 3 月 11 日 同検討会策定）も活用しつつ、専門組織同士の円滑な情報共有及びその促進のための枠組みを構築する。
- サイバーインフラ事業者（開発者、供給者、運用者）と顧客（発注者）全てを対象とした官民連携による情報共有の仕組みを構築する。

このような仕組み構築するためには、情報を共有する場において情報共有の資格をどのように定めるのか、また、情報を有効活用するためのフォーマットやマニュアルの整備などが課題として挙げられる。（関連する要求事項：S(5)-1.2）

S (5)-2	開発者 供給者 運用者 顧客	

協力体制の強化

ソフトウェアの製品及びサービスのセキュリティを改善するために、民間企業同士、関係当局、専門組織との協力体制と枠組みを活用する。

☐ **S(5)-2.1 協力体制の活用** ソフトウェアの製品及びサービスのセキュリティを改善するために、外部の事業者、顧客、及び専門機関が参加するソフトウェアセキュリティの改善を目的とするコミュニティや協力体制を活用する。	
取組例	• セキュリティに関する情報を共有・分析する ISAC などの業界団体へ参加する。
☐ **S(5)-2.2 協力体制への貢献** コミュニティや協力体制に参加する場合には、積極的に活動に関与し、協力体制に対して貢献する。	
取組例	• 被害情報の共有にとどまらず、以下のような幅広い協力の枠組みを活用する。 ➢ 親会社が CSIRT 協議会に参加し、親会社の CSIRT にグループ会社の従業員を派遣し、情報を共有 ➢ グループ会社内で、MISP を活用し IoC 情報を共有 ➢ ISAC（SoftwareISAC など）や CSIRT 協議会に参加 ➢ 民間企業の有志団体が集まったコミュニティ、地域のセキュリティコミュニティを通じた連携政府機関等が立ち上げた会議体に参加し、地域の企業、商工団体、自治体との脆弱性情報を共有 ➢ 進行中のプロジェクトがない委託先とも、勉強会を開催し、情報を共有 ➢ プライム事業者が主催するユーザ会、インシデント事例やその原因に関する業界を横断した説明会 ➢ IPA などの国主導の状況共有枠組みの活用

■ 関係者間の協力体制を強化する取組への期待

関係者間に跨がって製品やサービスのセキュリティを改善するための協力体制を強化するためには、以下のような取組が期待される。（関連する要求事項：S(5)-2.2）

- 業界団体の協力は不可欠であり、業界団体において課題を取り上げることなどを通じて、事業者の積極的な参加を促し、様々な面で貢献することが期待される。
- 守秘義務を飛び越えてサイバー脅威に対応する情報を共有する、できる仕組み、インシデント発生時の契約の在り方、情報開示のレベルとルール整備等のサプライチェーンの情報の共有方法、また、既知となった脆弱性や対処情報を顧客あるいは保守運用側へ迅速に報告、共有する枠組みの構築が期待される。
- セキュリティ要件の実装に関する事業者レベルアップのサポート方法の一つとして、ガイドラインの整備、業界単位でのセキュリティベースラインの整理などが期待される。

（６）顧客によるリスク管理とセキュアなソフトウェアの調達・運用

S (6)-1	開発者	供給者	運用者	顧客
顧客経営層のリーダーシップによるリスク管理 顧客経営層のリーダーシップにより、顧客独自のリスク管理をサイバーインフラ事業者と協力して実施するリスク管理を統合する。				
□　**S(6)-1.1 リスク管理** 顧客の独立した主体的な取組とサイバーインフラ事業者との契約に基づく取組を統合したリスク管理を実施する。				
取組例	• 顧客はオーナーシップを持つシステム全体のリスク管理に責任を持つ。その上で、システム全体のリスクを踏まえた適切な対策（管理者権限のあるユーザに対する多要素認証の適用、シングルサインオンの適正運用による効率化実現など）を実施する。 • 組織のセキュリティ態勢を支える重要なサイバーインフラ事業者を重要なビジネス機能とみなし、サイバーインフラ事業者からの提案及び費用内訳の妥当性の確認結果などを踏まえ、組織の成功にとっての重要性に応じて、対象とするシステム及びソフトウェアの運用及びリスク対応に係るライフサイクル全体の資金拠出を行う。 • サイバーインフラ事業者に対して、内部統制に関する態勢やセキュアバイデザインとセキュアバイデフォルト慣行を取り入れるためのロードマップに対する透明性を要求する。 • 顧客がオーナーシップを持つシステムの運用においてインシデントが発生する場合を想定し、そのシステムの保守を委託するサイバーインフラ事業者とインシデント対応とその役割分担を含む保守契約を締結する。 • セキュアバイデザインやセキュアバイデフォルトの慣行を取り入れるサイバーインフラ事業者を活用するための能力向上に向けた計画を作成する。 • サイバーセキュリティスタッフ、セキュリティチャンピオン、セキュリティテスター、運用とプラットフォームのエンジニア、調達スタッフなど、ソフトウェア運用ライフサイクルに関与する全ての役割と責任を定義する。 • クラウドシステムを利用する場合、責任共有モデルに基づいて顧客と供給者のセキュリティに関する責任を明確にし、セキュリティ態勢、内部統制、責任共有モデルの下で責任を果たす能力について、透明性の高いクラウド事業者を優先する。			

□ S(6)-1.2 リソース整備 既知の脆弱性への対処、及び緩和策を主体的に実施するためのリソースを割り当て、整備する（SBOM 活用を含む）。	
取組例	• ソフトウェア製品のサポート期限を確認し、サポート切れのソフトウェアを使用しない運用計画とする。 • ソフトウェア製品のセキュリティ実装に係る証明情報（SBOM、SSDF の実装の適合性を証明する自己適合証明書など）を要求・検証する。 • ソフトウェアの受入れ、又は展開前に、整合性メカニズムチェック、セキュリティテスト、環境テスト、機能テストを実行する。 • 導入するソフトウェアの品質を担保するため、顧客とサイバーインフラ事業者の協議により品質の検証方法と基準を定め、エビデンスを要求する。 • 導入したソフトウェアのセキュリティ監視を継続的に実施し、ソフトウェアの脆弱性の疑いが特定された場合、サイバーインフラ事業者に報告する。 • ソフトウェアの更新戦略に基づく更新ポリシーを定め、必要に応じて自動化更新メカニズムを採用する。
□ S(6)-1.3 協力体制の活用 ソフトウェアセキュリティの改善を目的とするコミュニティや協力体制を活用する。	
取組例	• コミュニティや協力体制に参加する場合には、積極的に活動に関与し、協力体制に対して貢献する。

■ ライフサイクルの違い

ソフトウェアを利用する顧客が認識するライフサイクル（利用期間）と、ソフトウェアを提供するサイバーインフラ事業者が認識するライフサイクル（サポート期間）は、異なるものである。ソフトウェアを利用する際には、利用するソフトウェアのバージョンのサポート期限を定期的に確認し、サポート期限が終了したソフトウェアを利用することがないようなソフトウェアの運用計画とすることが肝要である。

（関連する要求事項：S(6)-1.2）

S (6)-2	開発者	供給者	運用者	顧客
顧客経営層のリーダーシップによるソフトウェアの調達、運用 顧客経営層のリーダーシップにより、セキュアにソフトウェアを調達、運用する。				
☐ **S(6)-2.1 セキュリティ要件の定義** ソフトウェア設計計画にセキュリティ機能を組み込むためのセキュリティ要件を定義し、ソフトウェアを調達・導入する前に、サイバーインフラ事業者に提示する。				
取組例	• 業界のカウンターパートと協力し、サイバーインフラ事業者が今後セキュアバイデザインとセキュアバイデフォルトの取組を優先するように要望を提示する。			
☐ **S(6)-2.2 セキュリティ慣行の要求開示** ソフトウェアの調達・導入前に、サイバーインフラ事業者に求めるセキュリティ慣行の要求を開示する。				
取組例	• IT 部門に、セキュアバイデザインとセキュアバイデフォルトの慣行に力点を置いた購入基準を作成する権限を与える。			
☐ **S(6)-2.3 リスク評価に基づく意思決定** ソフトウェアを調達・導入する際に、リスク評価に基づいた意思決定を行う。				
取組例	• IT 部門がソフトウェア購入前にそのセキュリティを評価すること、及び必要な情報源を要求することを定めたポリシーを作成し、IT 部門に必要に応じて拒否する権限を明確に与える。 • 特定の技術製品に関するリスクを受け入れる決定を行う場合には、正式に文書化し、経営層幹部が承認し、定期的に取締役会に報告する。 • デジタルサービスを導入する際には、他のデジタルサービスへの移行可能性をリスクの観点から評価し、導入の意思決定を行う。			
☐ **S(6)-2.4 予算確保** ソフトウェアのライフサイクルを考慮した導入・運用・移行・廃棄、リスク対応、及び関連する契約に係る予算を継続的に確保する。				
取組例	• サイバーセキュリティに関する残存リスクを許容範囲以下に抑制するための方策を検討し、その実施に必要となる資源（予算、人材等）を確保した上で、具体的な対策に取り組む。（サイバーセキュリティ経営ガイドライン v3 指示 3） （システム・サービスのソフトウェアの場合） • 関係者間で費用認識の共有を促進する前提として、脆弱性の放置が将来的な負債（サイバー攻撃による損害など）につながることを共通の経営リスクとして把握する。			

5.5. 統一基準群と本ガイドライン（案）との関係

(1) 統一基準群の利用の枠組み、統一基準の位置付け

国の行政機関及び独立行政機関（以下、政府機関等という）は、内閣官房国家サイバー統括室から公開される「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」[13]（以下、統一基準群という）の利用の枠組みに則り、それぞれの組織の情報セキュリティを確保することとしている。

この枠組みにおいて、政府機関等は、統一規範及びその実施のための要件である統一基準に準拠するために、「政府機関等の対策基準策定のためのガイドライン」（以下、対策基準策定ガイドラインという）を参照しつつ、組織及び取り扱う情報の特性を踏まえて情報セキュリティポリシーを策定し、ポリシーに定めた対策事項に係る運用規程や実施手順を定め、計画的に対策を実施することが求められる。

統一基準群は、統一規範、統一基準、対策基準策定ガイドラインにより構成される。統一基準は、政府機関等が実施すべき対策について、目的別に 3 階層（部、節、款）の対策項目を分類し、3 階層目（款）に対して目的、趣旨、遵守事項を示している。対策基準策定ガイドラインは、遵守事項を満たすためにとるべき基本的な対策事項を例示し、情報セキュリティポリシーの策定や実施に関する考え方等を解説している。

(2) 統一基準群が扱うソフトウェアとの関係

本ガイドライン（案）が扱う「ソフトウェア」とは、本ガイドライン（案）の目的である「ソフトウェアを対象とした効果的なサプライチェーン上のサイバーセキュリティ対策を進める」に則り、サイバーインフラ事業者が扱う以下のソフトウェアを対象としている。（詳細は本ガイドライン（案）の「1.3. 適用対象 (1)ソフトウェアの範囲」を参照）

- ソフトウェア製品
- ソフトウェアサービス
- 組み込みソフトウェア
- システム・サービスを構成するソフトウェア

一方、統一基準群が外部委託の調達や情報システムの開発・運用の委託において対策強化を求めるソフトウェアの範囲は以下としており、本ガイドライン（案）が対象とするソフトウェアに合致しているものと想定する。

＜外部委託（調達）＞

- クラウドサービス
- 機器等（サーバ装置、端末、通信回線装置、複合機、特定用途機器等、ソフトウェア等）
  ※ソフトウェアについては特にサプライチェーン・リスクに対応する必要があると判断されるソフトウェアを＜情報システムの基盤を管理又は制御するソフトウェアの例＞として以下を例示
  - 端末やサーバ装置、通信回線装置等を制御するソフトウェア

[13] https://www.nisc.go.jp/policy/group/general/kijun.html
なお、統一基準群のうち「統一規範」及び「統一基準」はサイバーセキュリティ戦略本部が決定する。

- 統合的な主体認証を管理するソフトウェア
- ネットワークを制御・管理するソフトウェア
- 資産を管理するソフトウェア
- 監視に関連するソフトウェア
- 情報システムのセキュリティ機能として使用するソフトウェア

＜情報システム（開発・運用の委託）＞

- アプリケーション・コンテンツ

また、直近の統一基準群の改定においては、ソフトウェアのセキュリティ対策やサプライチェーン・リスク対策に関する強化が進められている。統一基準群（令和５年度版）改定のポイントとして、以下の項目が示されていることから、本ガイドライン（案）の目的にも合致する。

- 統一基準群（令和 5 年度版）改定のポイント[14]
  - 情報セキュリティに関するサプライチェーン対策の強化
  - クラウドサービスの利用拡大を踏まえた対策の強化
  - ソフトウェア利用時の対策の強化
  - サイバーレジリエンスの強化や脅威・技術動向を踏まえての対策の強化
  - 組織横断的な情報セキュリティ対策の強化と情報システムの重要度に応じた対策の確保

(3) 統一基準群と本ガイドライン（案）との関係性

本ガイドライン（案）は、ソフトウェアのサイバーセキュリティの確保とレジリエンスの向上のためのサイバーインフラ事業者と顧客の適切な役割分担と責務の在り方を示すものである。統一基準群との関係性において、顧客とは政府機関等であり、統一基準群を遵守・参照する主体そのものである。一方、サイバーインフラ事業者は、政府機関等からみると外部委託者となるため、主体である政府機関等（顧客）から、情報システムの開発・運用あるいは機器等の調達先としてソフトウェアを含む業務委託を請ける立場となる。統一基準群は、顧客が遵守事項を実施できるようにするために、サイバーインフラ事業者が受託者として果たすべき役割と責務を直接明示しているわけではないため、統一基準群の内容を受託者の立場で読み替えて理解する必要がある。

統一基準の１階層目の部の分類は第２部から第８部の７分類であり、本ガイドライン（案）が示す責務に直接関係する内容を含む部（以下の表で「〇」とする部分）、及び本ガイドライン（案）が示す責務としての要求事項に関連する記載を含む部（同表で「△」とする部分）が存在する。

表 8 統一基準の 1 層目（部）との対応関係

統一基準の 1 層目（部）	サイバーインフラ事業者	顧客
第 1 部 総則		
第 2 部 情報セキュリティ対策の基本的枠組み		〇
第 3 部 情報の取扱い		△

[14] https://www.nisc.go.jp/pdf/policy/general/rev_pointr5.pdf

統一基準の 1 層目（部）	サイバーインフラ事業者	顧客
第 4 部 外部委託	○	○
第 5 部 情報システムのライフサイクル	○	○
第 6 部 情報システムの構成要素	○	○
第 7 部 情報システムのセキュリティ要件	○	○
第 8 部 情報システムの利用	△	△
付録		

また、2 階層目の節の分類は、統一基準と本ガイドライン（案）との関係において、統一基準の以下の節（以下の表で赤枠で示す部分）は、顧客とサイバーインフラ事業者が共に「○」であることから、本ガイドライン（案）が示す責務と役割分担について特に深い関係性が認められる。

4.1 業務委託
4.2 クラウドサービス
4.3 機器等の調達
5.2 情報システムのライフサイクルの各段階における対策
6.5 ソフトウェア
6.6 アプリケーション・コンテンツ
7.2 情報セキュリティの脅威への対策

表 9 統一基準の 2 層目（節）との対応関係

統一基準の 2 層目（節）		サイバーインフラ事業者	顧客
第 1 部 総則			
	1.1 本ガイドラインの目的等		
	1.2 情報の格付の区分・取扱制限		
	1.3 統一基準における用語定義		
	1.4 一般用語の解説		
	1.5 基本対策事項及び解説の読み方		
第 2 部 情報セキュリティ対策の基本的枠組み			○
	2.1 導入・計画		○
	2.2 運用		○
	2.3 点検		△
	2.4 見直し		△
	2.5 独立行政法人及び指定法人		△
第 3 部 情報の取扱い			△
	3.1 情報の取扱い		△
	3.2 情報を取り扱う区域の管理		△
第 4 部 外部委託		○	○
	4.1 業務委託	○	○

統一基準の 2 層目（節）		サイバーインフラ事業者	顧客
	4.2 クラウドサービス	○	○
	4.3 機器等の調達	○	○
第 5 部 情報システムのライフサイクル		○	○
	5.1 情報システムの分類		△
	5.2 情報システムのライフサイクルの各段階における対策	○	○
	5.3 情報システムの運用継続計画	△	△
	5.4 政府共通利用型システム	△	△
第 6 部 情報システムの構成要素		○	○
	6.1 端末	△	△
	6.2 サーバ装置	△	△
	6.3 複合機・特定用途機器	△	△
	6.4 通信回線	△	△
	6.5 ソフトウェア	○	○
	6.6 アプリケーション・コンテンツ	○	○
第 7 部 情報システムのセキュリティ要件		○	○
	7.1 情報システムのセキュリティ機能	△	△
	7.2 情報セキュリティの脅威への対策	○	○
	7.3 ゼロトラストアーキテクチャ	△	△
第 8 部 情報システムの利用		△	△
	8.1 情報システムの利用	△	△
付録			

5.6. 重要インフラのサイバーセキュリティに係る安全基準等策定指針と本ガイドライン（案）との関係

(1) 重要インフラのサイバーセキュリティに係る安全基準等策定指針の利用の枠組みと位置付け

重要インフラ事業者等は、当該事業分野に関する法制度の下、関係する基準に従い、業を営んでいる。「重要インフラのサイバーセキュリティに係る行動計画」[15]（以下、「行動計画」という。）において、重要インフラ事業者等は、後述の安全基準等を踏まえ自組織の障害対応体制の強化等に努めるものとされており、これらを通じて、重要インフラに関わる情報セキュリティ対策が総合的に進められている。「重要インフラのサイバーセキュリティに係る安全基準等策定指針」[16]（以下、「策定指針」という。）においては、サイバーセキュリティの確保に関して、各重要インフラ事業者等の判断や行為に関するこれらの基準又は参考となる文書類を「安全基準等」[17]と定義し、各重要インフラ分野に共通して求められるサイバーセキュリティ確保に向けた取組が分類されて整理されており、これらの取組を原則として重要インフラの業界別に策定される安全基準等に記載することが期待されている。

また、策定指針の参考文書として手引書が整備されており、リスクマネジメントといったセキュリティ対策を進める際の基本的な考え方や具体的な手順を解説している。

(2) 策定指針が扱うソフトウェアとの関係

本ガイドライン（案）が扱う「ソフトウェア」とは、本ガイドライン（案）の目的である「ソフトウェアを対象とした効果的なサプライチェーン上のサイバーセキュリティ対策を進める」に則り、サイバーインフラ事業者が扱う以下のソフトウェアを対象としている。（詳細は本ガイドライン（案）の「1.3. 適用対象 (1)ソフトウェアの範囲」を参照）

- ソフトウェア製品
- ソフトウェアサービス
- 組み込みソフトウェア
- システム・サービスを構成するソフトウェア

一方、策定指針を通じて安全基準等に規定すべき対象範囲は、行動計画「別紙１　対象となる重要インフラ事業者等と重要システム例」に記載された「対象となる重要システム例」や、「別紙２　重要インフラサービスとサービス維持レベル」に記載された「重要インフラサービス（手続を含む）」、「重要インフラサービス障害の例」、「サービス維持レベル」等の内容を踏まえることとされている。対策すべきサプライチェーンの事例としてクラウドサービスが示されており、リスク管理の対象として、情報システム、制御システムや汎用機器が示されている。ソフトウェアはシステムを構成する要素であることを踏まえると、対象とするソフトウェアとして以下が想定される。したがって、本ガイドライン（案）が対象とするソフトウェアのうち、ソフトウェア製品以外は、合致しているものと想定する。

[15] https://www.nisc.go.jp/pdf/policy/infra/cip_policy_2024.pdf
[16] https://www.nisc.go.jp/pdf/policy/infra/shishin202307.pdf
[17] 関係法令に基づき国が定める「強制基準」、関係法令に準じて国が定める「推奨基準」及び「ガイドライン」、関係法令や国民からの期待に応えるべく業界団体等が定める業界横断的な「業界標準」及び「ガイドライン」、関係法令や国民・利用者等からの期待に応えるべく重要インフラ事業者等が自ら定める「内規」に分類される。

＜外部委託（調達）＞

- クラウドサービス
- 制御システム（汎用機器を含む）

＜情報システム（開発・運用の委託）＞

- 情報システム

また、策定指針において以下のとおり記載されており、本ガイドライン（案）の目的にも合致する。

4.4. サプライチェーン・リスクマネジメント
自組織の重要システムや機能とサプライチェーンの依存関係の把握、供給者のセキュリティ対策の状況の把握を行う。 サプライチェーン・リスクに関するリスクアセスメント及びリスク対応を行う。（中略） 直接の供給者を対象に、事業者間の契約において、サイバーセキュリティリスクへの対応に関して担うべき役割と責任範囲を明確化する。さらに、リスクに応じて直接の供給者に連なる供給者への関与の程度を決定しつつ、各供給者がその先の供給者を対象にサプライチェーン・リスクマネジメントの実施状況を把握することで、サプライチェーン全体のリスクマネジメントを実施することが望ましい。また、セキュリティ対策の導入支援や共同実施等により、サプライチェーン全体での方策の実効性を高めることが望ましい。

(3) 策定指針と本ガイドライン（案）との関係性

本ガイドライン（案）は、ソフトウェアのサイバーセキュリティの確保とレジリエンスの向上のためのサイバーインフラ事業者と顧客の適切な役割分担と責務の在り方を示すものである。策定指針との関係性において、顧客とは重要インフラ事業者等であり、策定指針を踏まえた安全基準等に基づき対策を実施する主体そのものである。一方、サイバーインフラ事業者は、重要インフラ事業者等からみると通常は外部委託者となる[18]ため、主体である重要インフラ事業者等（顧客）から、情報システムの開発・運用あるいは機器等の調達先としてソフトウェアを含む業務委託を請ける立場となる。策定指針は、顧客が各重要インフラ分野に共通して求められるサイバーセキュリティ確保に向けた取組を実現できるようにするために、サイバーインフラ事業者が受託者として果たすべき役割と責務を直接明示しているわけではないため、策定指針の内容を受託者の立場で読み替えて理解する必要がある。

策定指針は、各重要インフラ分野に共通して求められるサイバーセキュリティ確保に向けた取組が分類されている。本ガイドライン（案）が示す責務に直接関係する内容を含む章（以下の表で「〇」とする部分）、及び本ガイドライン（案）が示す責務としての要求事項に関連する記載を含む章（同表で「△」とする部分）が存在する。

[18] ただし、重要インフラ事業者が自組織において開発、供給を実施する場合は、重要インフラ事業者自身が責務区分「サイバーインフラ事業者」としての各役割である「開発者」、「供給者」の責務を自ら「(主体)」として負うものと考えられる。具体例は「1.4 役割分担の考え方」の表 4 の e を参照。

表 10 策定指針の 1 層目との対応関係

策定指針の 1 層目	サイバーインフラ事業者	顧客
1. 目的及び位置付け		
2. 総則		△
3. 組織統治におけるサイバーセキュリティ		△
4. リスクマネジメントの活用と危機管理	○	○
5. 対策項目	○	○

2 階層目の節の分類は、策定指針と本ガイドライン（案）との関係において、策定指針の以下の節（以下の表で赤枠で示す部分）が顧客とサイバーインフラ事業者が共に「○」であることから、本ガイドライン（案）が示す責務と役割分担について特に深い関係性が認められる。

4.2 リスクアセスメント
4.3 サイバーセキュリティリスク対応
4.4 サプライチェーン・リスクマネジメント
4.8 平時の運用
5.1 組織的対策

表 11 策定指針の 2 層目との対応関係

策定指針の 2 層目		サイバーインフラ事業者	顧客
1. 目的及び位置付け			
	1.1. 重要インフラにおけるサイバーセキュリティの確保の重要性		
	1.2.「安全基準等」とは何か		
	1.3. 安全基準等策定 指針の位置付け		
2. 総則			△
	2.1. 策定目的		
	2.2. 対象範囲		
	2.3. 関係主体の役割		△
3. 組織統治におけるサイバーセキュリティ			△
	3.1. 組織方針		△
	3.2. 組織内外のコミュニケーション		△
	3.3. 経営リスクとしてのサイバーセキュリティリスクの管理		○
	3.4. 責任及び権限の割当て		△
	3.5. 資源の確保		△
	3.6. 監査・モニタリング		△
	3.7. 情報開示		
	3.8. 継続的改善		△
4. リスクマネジメントの活用と危機管理		○	○

策定指針の 2 層目		サイバーインフラ事業者	顧客
	4.1. 組織状況の理解	△	△
	4.2. リスクアセスメント	○	○
	4.3. サイバーセキュリティリスク対応	○	○
	4.4. サプライチェーン・リスクマネジメント	○	○
	4.5. 事業継続計画等		
	4.6. 人材育成・意識啓発		△
	4.7. CSIRT 等の整備	△	△
	4.8. 平時の運用	○	○
	4.9. 危機管理	△	△
	4.10. 演習・訓練	△	△
5. 対策項目		○	○
	5.1. 組織的対策	○	○
	5.2. 人的対策	△	△
	5.3. 物理的対策		
	5.4. 技術的対策	△	△
	5.5. 動向を踏まえた対策	△	△

5.7. サイバー対処能力強化法と本ガイドライン（案）との関係

(1)サイバー対処能力強化法の利用の枠組みと位置付け

令和７年５月に成立したサイバー対処能力強化法では、内閣総理大臣及び電子計算機等供給事業所管大臣は、重要電子計算機として用いられる電子計算機やプログラムにおける脆弱性を認知したときには、必要に応じ、当該電子計算機等の供給者に対し情報を提供するとともに、当該情報等について、公表その他の適切な方法により周知できることとされている（第 42 条第 1 項）。この枠組みにおいて、電子計算機等供給者は、提供・公表等された情報を踏まえ、脆弱性への対応等を行うことが期待される。

また、同法では、電子計算機等供給事業所管大臣は、脆弱性が基幹インフラ事業者が使用する特定重要電子計算機に用いられる電子計算機等に関連するものである場合には、当該電子計算機等の供給者に対し、サイバー攻撃による被害を防止するために必要な措置を講ずるよう要請することができることとされている（第 42 条第 2 項）。

さらに、同法では、内閣総理大臣及び電子計算機等供給事業所管大臣は、上記の脆弱性に係る公表等の周知や協力要請の実施に当たって、脆弱性に関する適切な情報把握を行う必要があることから、電子計算機等供給者に対し、必要な報告又は資料提出を求めることができることとされている（第 42 条第 4 項）。この枠組みにおいて、電子計算機等供給者は、政府に対して報告又は資料の提出を行う努力義務を負うこととされている（第 42 条第 5 項）。

(2)サイバー対処能力強化法が扱う電子計算機等供給者との関係

サイバー対処能力強化法第 42 条第 1 項に基づき情報提供の対象となる電子計算機等供給者とは、重要電子計算機に用いられる電子計算機や当該電子計算機に組み込まれるプログラムの供給者であり、同条第２項で要請の対象となるのは基幹インフラ事業者が使用する特定重要電子計算機に用いられる電子計算機等に関連するものであるため、サイバーインフラ事業者がこれらの規定の適用を受けるかはソフトウェアの供給先に応じて判断されることとなる。ただし、当該ソフトウェアは、現に特定電子計算機等に用いられているもののみならず、将来的に用いられる蓋然性が高いものも含まれることに留意が必要である。

5.8. 参照情報

（1）参照情報のリスト

略称	文書名
NSA	SECURING THE SOFTWARE SUPPLY CHAIN / Recommended Practices Guide for Developers https://media.defense.gov/2022/Sep/01/2003068942/-1/-1/0/ESF_SECURING_THE_SOFTWARE_SUPPLY_CHAIN_DEVELOPERS.PDF
NSA-S	SECURING THE SOFTWARE SUPPLY CHAIN / Recommended Practices Guide for Suppliers https://media.defense.gov/2022/Oct/31/2003105368/-1/-1/0/SECURING_THE_SOFTWARE_SUPPLY_CHAIN_SUPPLIERS.PDF
NSA-C	SECURING THE SOFTWARE SUPPLY CHAIN / Recommended Practices Guide for Customers https://media.defense.gov/2022/Nov/17/2003116445/-1/-1/0/ESF_SECURING_THE_SOFTWARE_SUPPLY_CHAIN_CUSTOMER.PDF
SP800-218	NIST SP800-218 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
BSA	The BSA Framework for Secure Software: A New Approach to Securing the Software Lifecycle https://www.bsa.org/files/reports/bsa_software_security_framework_web_final.pdf
CISA-D	Defending Against Software Supply Chain Attacks https://www.cisa.gov/sites/default/files/publications/defending_against_software_supply_chain_attacks_508_1.pdf
CISA-SBD	Secure-by-Design - Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software https://www.cisa.gov/sites/default/files/2023-10/SecureByDesign_1025_508c.pdf
SP800-161	NIST SP800-161 Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-161r1-upd1.pdf
ISMS	ISO/IEC 27002:2022 - Information security, cybersecurity and privacy protection Information security controls https://www.iso.org/standard/75652.html

略称	文書名
ISO 15408	Common Criteria for Information Technology Security Evaluation ISO/IEC 15408:2022 - Information security, cybersecurity and privacy protection Evaluation criteria for IT security Part1～3 https://www.iso.org/standard/72891.html https://www.iso.org/standard/72892.html https://www.iso.org/standard/72906.html
DSP	ENISA Guidelines on assessing DSP and OES compliance to the NISD security requirements https://op.europa.eu/en/publication-detail/-/publication/78f2a620-f909-11e8-9982-01aa75ed71a1/language-en
総務省	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 https://www.soumu.go.jp/main_content/000771515.pdf クラウドサービスの安全・信頼性に係る情報開示指針 https://www.soumu.go.jp/main_content/000477838.pdf
CRA	The European Cyber Resilience Act https://www.europarl.europa.eu/RegData/etudes/BRIE/2022/739259/EPRS_BRI(2022)739259_EN.pdf REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on horizontal cybersecurity requirements for products with digital elements and amending Regulation (EU) 2019/1020 https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L_202402847
統一基準群	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 https://www.nisc.go.jp/pdf/policy/general/kijyunr5.pdf 政府機関等のサイバーセキュリティ対策のための統一規範 https://www.nisc.go.jp/pdf/policy/general/kihanr5.pdf 政府機関等の対策基準策定のためのガイドライン https://www.nisc.go.jp/pdf/policy/general/guider6.pdf
策定指針	重要インフラのサイバーセキュリティに係る行動計画 https://www.nisc.go.jp/pdf/policy/infra/cip_policy_abst_2024.pdf 重要インフラにおける情報セキュリティ確保に係る安全基準等策定指針 https://www.nisc.go.jp/pdf/policy/infra/shishin5.pdf
日米豪印サイバーセキュリティ・パートナーシップ	日米豪印首脳会合共同声明（QUAD 共同原則） https://www.mofa.go.jp/mofaj/fp/nsp/page1_001188.html
UN-R155	UN Regulation No. 155 - Cyber security and cyber security management system https://unece.org/sites/default/files/2023-02/R155e%20%282%29.pdf
ISO 21434	ISO/SAE 21434:2021 - Road vehicles Cybersecurity engineering https://www.iso.org/standard/70918.html
UN-R156	UN Regulation No. 156 - Software update and software update management system https://unece.org/sites/default/files/2024-03/R156e%20%282%29.pdf

略称	文書名
ISO 24089	ISO 24089:2023 - Road vehicles Software update engineering https://www.iso.org/standard/77796.html ISO 24089:2023/Amd 1:2024 https://www.iso.org/standard/87522.html
OMB M-23-16	Update to Memorandum M-22-18, Enhancing the Security of the Software Supply Chain through Secure Software Development Practices https://bidenwhitehouse.archives.gov/wp-content/uploads/2023/06/M-23-16-Update-to-M-22-18-Enhancing-Software-Security.pdf Secure Software Development Attestation Form Instructions https://www.cisa.gov/sites/default/files/2024-03/Self-Attestation-Common-Form-03082024-FINAL.pdf
SP800-218A	NIST SP 800-218A Secure Software Development Practices for Generative AI and Dual-Use Foundation Models An SSDF Community Profile https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218A.pdf
DSIT	DSIT – The Code of Practice for Software Vendors https://www.gov.uk/government/calls-for-evidence/a-code-of-practice-for-software-vendors-call-for-views/call-for-views-on-the-code-of-practice-for-software-vendors
サイバーセキュリティ経営ガイドライン	サイバーセキュリティ経営ガイドライン https://www.meti.go.jp/policy/netsecurity/downloadfiles/guide_v3.0.pdf

（２）他の標準・ガイドライン等との関係

本ガイドライン（案）は、ソフトウェアのセキュリティやソフトウェア開発保証を対象とする各種ガイドラインやフレームワークとの間に、次のような関係を有する。具体的な取組において、方針検討及び実現する手段を更に検討する際に、これらのガイドラインも活用することが可能である。本ガイドライン（案）のベースとなる範囲と主な他の標準・ガイドラインとの関係を 図 8 に示すとともに、主な標準・ガイドライン等との関係を以下に説明する。

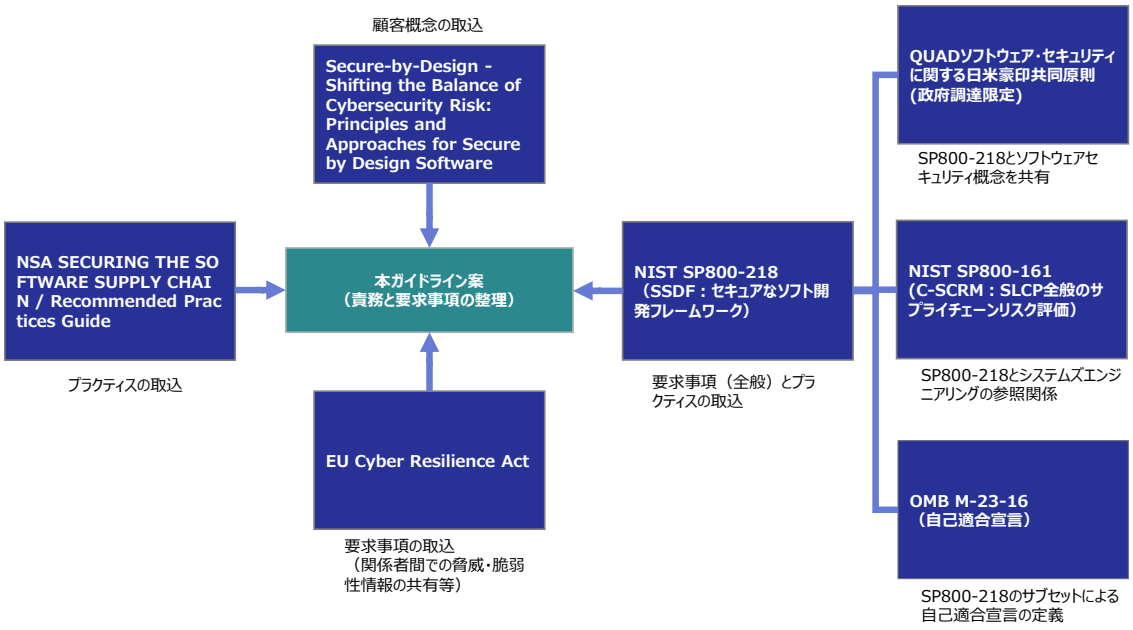


図 8 本ガイドライン（案）と他の標準・ガイドライン等との関係

① NIST SP800-218

NIST が発行した SP800-218 は、ソフトウェアサプライチェーンのセキュリティを強化するためのガイダンスである。ソフトウェア開発ライフサイクル（SDLC）モデルに対して、開発中のソフトウェアのセキュリティを確保するために、セキュアなソフトウェア慣行を追加する SSDF: Secure Software Development Framework と呼ばれるフレームワークを提唱する。本ガイドライン（案）では、SP800-218 に示された各慣行を実行するために必要なタスクを「要求事項」として網羅的に扱い、各タスクに対する実装例を「取組例」の一部の参考としている。

② NSA Software Supply Chain Guidance (for Developers, Suppliers, Customers)

NSA が発行した３つのソフトウェアサプライチェーンガイダンス（開発者編、供給者編、顧客編）は、ソフトウェア開発者、ソフトウェア供給者、及び顧客が参照すべき業界のベストプラクティスと原則を提供するガイダンスである。開発者編に示される原則には、セキュリティ要件の計画、セキュリティの観点からソフトウェアアーキテクチャの設計、セキュリティ機能の実装、ソフトウェア開発の基盤（開発環境、ソースコードレビュー、テストなど）のセキュリティ維持が含まれる。本ガイドライン（案）では、3 つの文書に示された原則及びベストプラクティスのエッセンスを主に「取組例」の一部の参考としている。

③ CISA Secure-by-Design - Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software

CISA が発行したセキュアバイデザインの原則とアプローチに関するガイダンスは、ソフトウェア開発者に対して製品の設計、開発、提供の際にセキュリティを最優先に求めることを目的としたガイダンスである。3 つの原則「顧客のセキュリティ成果のオーナーシップを持つ」、「根本的な透明性と説明責任を受け入れる」、「トップからリードする」を提唱し、セキュアバイデザイン、セキュアバイデフォルトの観点から、それぞれの原則の解説と主要なプラクティス、及びタクティクス（手法）を示している。本ガイドライン（案）では、セキュアバイデザイン及びセキュアバイデフォルトの主旨に則った責務の要求事項を構成するとともに、ソフトウェア製品のセキュリティ原則のプラクティス及びタクティクス（手法）のエッセンスを「取組例」の一部の参考としている。

④ EU Cyber Resilience Act.

欧州サイバーレジリエンス法は、EU 域内におけるデジタル要素を備えたハードウェア及びソフトウェア製品に対するサイバーセキュリティ要件を規定した法的枠組みであり、2024 年に発効、2027 年に全面的に施行予定である。この法的枠組みは、一部の例外を除いたデジタル要素を備えた広範な製品を対象としており、欧州市場に製品を投入する製造業者は、製品のライフサイクル全体を通じてセキュリティ要求への適合（SBOM の作成、セキュリティ更新の提供、脆弱性発見時やインシデント発生時の当局への報告など）を保証する義務を負うことになる。本ガイドライン（案）では、サイバーセキュリティ必須要件（製品の特性に関する要件、及び脆弱性処理要件）及びユーザへの情報と指示の一部を「個別要件」又は「取組例」の一部の参考としている。

⑤ その他の標準・ガイドライン

本ガイドライン（案）では、上記の他にも以下の標準・ガイドラインを参考としている。

- The BSA Framework for Secure Software: A New Approach to Securing the Software Lifecycle
- CISA Defending Against Software Supply Chain Attacks
- NIST SP800-161 Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations
- ENISA Guidelines on assessing DSP and OES compliance to the NISD security requirements
- Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408)
- ISO/IEC 27002:2022
- 総務省 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
- 総務省 クラウドサービスの安全・信頼性に係る情報開示指針
- 国家サイバー統括室 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
- 国家サイバー統括室 政府機関等の対策基準策定のためのガイドライン
- 国家サイバー統括室 重要インフラのサイバーセキュリティに係る行動計画

- 国家サイバー統括室 重要インフラにおける情報セキュリティ確保に係る安全基準等策定指針
- 日米豪印サイバーセキュリティ・パートナーシップ（日米豪印首脳会合共同声明:QUAD 共同原則）
- UN-R155
- UN-R156
- ISO/SAE 21434:2021
- ISO 24089:2023
- OMB M-23-16
- NIST SP800-218A
- DSIT The Code of Practice for Software Vendors

その他、CISA から公開されている、ソフトウェアに関するサイバーサプライチェーンのリスク管理（C-SCRM）のフレームワークと SSDF に基づいてリスクを特定、評価、軽減する方法に関する推奨事項なども参考としている。

（3）NIST SP800-218との対応関係

要求事項	NIST SP800-218 の対応する箇所
S(1)-1	PW 1.1, PW 1.2, PW 2.1
S(1)-2	PW 5.1, PW 6.1, PW 6.2, PW 7.1, PW 7.2, PW 9.2
S(1)-3	PW 8.1, PW 8.2, PW 9.1
S(1)-4	PW 9.1, (PS 1.1, PO 5.1, PO 5.2)
S(2)-1	PW 1.3, PW 4.1, PW 4.2, PW 4.4, (RV 2.1)
S(2)-2	PS 1.1, PS 3.1, PS 3.2
S(2)-3	PO 1.3, (PW 4.4)
S(2)-4	PS 2.1, PW 9.2
S(3)-1	RV 1.1, RV 1.2, RV 1.3
S(3)-2	RV 2.1, RV 2.2
S(3)-3	RV 3.1, RV 3.2, RV 3.3, RV 3.4, PW 7.2
S(4)-1	PO 2.1, PO 2.2, PO 2.3, (PO 3.1, PO 3.2, PO 3.3)
S(4)-2	PO 1.1, PO 1.2, (PO 2.3)
S(4)-3	(PO 1.1, PO 1.2, PO 2.3)
S(4)-4	PO 4.1, PO 4.2
S(4)-5	PO 3.1, PO 3.2, PO 3.3
S(4)-6	PO 5.1, PO 5.2
S(5)-1	―
S(5)-2	―
S(6)-1	―
S(6)-2	―

（4）NSA Software Supply Chain Guidance の 3 文書との対応関係

要求事項	NSA (for Developers)	NSA-S (for Suppliers)	NSA-C (for Customers)
S(1)-1	2.3.2	2.3.1	—
S(1)-2	2.2.1.4, 2.2.2, 2.2.6, 2.2.3.2, 2.3.2, 2.3.3, 2.4.1	2.2.2, 2.3.3, 2.3.4, 2.3.6	—
S(1)-3	2.2.1.3, 2.2.3.2, 2.3.2, 2.4.1	2.2.2, 2.3.5, 2.3.6	—
S(1)-4	—	—	—
S(2)-1	2.2.3, 2.3.2, 2.3.3, .2.3.4, 2.3.5	2.3.1, 2.3.2	2.1, 2.2
S(2)-2	2.2.1.1, 2.2.1.2, 2.2.1.4, 2.2.6, 2.3.2, 2.3.3, 2.4.1, 2.5.3	2.2.1, 2.2.2, 2.2.3	—
S(2)-3	2.2.3	2.1.1	—
S(2)-4	—	—	—
S(3)-1	2.3.4, 2.4.1	2.4.1	—
S(3)-2	—	—	—
S(3)-3	—	—	—
S(4)-1	—	—	—
S(4)-2	2.2.3	2.1.1	—
S(4)-3	2.2.3	2.1.1	—
S(4)-4	—	—	—
S(4)-5	—	—	—
S(4)-6	—	—	—
S(5)-1	—	—	—
S(5)-2	—	—	—
S(6)-1	—	—	2.1, 2.2, 2.3
S(6)-2	—	—	2.1

（５）CISA Secure-by-Design- Shifting the Balance of Cybersecurity Risk

との対応関係

要求事項	原則１	原則２	原則３	セキュアバイデザインの手法	セキュアバイデフォルトの手法
S(1)-1	[SBD]-1 [SPD]-5	—	—	11,12	—
S(1)-2	[SBD]-5	—	—	1,4,5,6,7	1
S(1)-3	[SBD]-2	—	—	6	—
S(1)-4	—	—	—	—	4
S(2)-1	[PSB]-3,4 [SPD]-4	—	—	3	—
S(2)-2	—	[SPD]-5	—	8	—
S(2)-3	—	—	—	—	—
S(2)-4	[SBD]-1,4	—	—	—	1,5,8
S(3)-1	—	[SPD]-6	—	9	—
S(3)-2	[PSB]-4	—	—	10	6
S(3)-3	[SPD]-3	—	—	10	—
S(4)-1	[SPD]-6	[PSB]-1	3,5	—	—
S(4)-2	[SPD]-1	—	—	—	—
S(4)-3	[SPD]-1	—	—	2	—
S(4)-4	[SPD]-5,6	—	—	12	—
S(4)-5	—	—	—	1,6	—
S(4)-6	—	—	—	—	—
S(5)-1	—	—	—	—	—
S(5)-2	—	—	6	—	—
S(6)-1	—	—	[RFC]	—	2,3
S(6)-2	—	—	[RFC]	—	—

※[SBD] : SECURE BY DEFAULT PRACTICES
[SPD] : SECURE PRODUCT DEVELOPMENT PRACTICES
[PSB] : PRO-SECURITY BUSINESS PRACTICES
[RFC] : RECOMMENDATIONS FOR CUSTOMERS

（6）EU Cyber Resilience Act. ANNEX I/IIとの対応関係

要求事項	ANNEX I Part I	ANNEX I Part II	ANNEX II
S(1)-1	(1), (2)-a/g/h/i/j/k	(3)	—
S(1)-2	(1)	—	—
S(1)-3	(1)	—	—
S(1)-4	(1), (2)-d/e/f	—	—
S(2)-1	—	(6)	—
S(2)-2	(2)-l	(1)	—
S(2)-3	—	—	—
S(2)-4	(2)-b/f/m	(7)	4,5,7,8-a/b/d
S(3)-1	(2)-a/j	(1),(2),(3),(6)	2
S(3)-2	(2)-a/c/j/l	(2),(4),(5),(7),(8)	8-c
S(3)-3	(2)-a/j/k	(2)	—
S(4)-1	—	—	—
S(4)-2	(2)-a/d/e/f/g/h/i	—	—
S(4)-3	(2)-a/g/h/i	—	—
S(4)-4	—	—	—
S(4)-5	—	—	—
S(4)-6	—	—	—
S(5)-1	—	—	—
S(5)-2	—	—	—
S(6)-1	—	—	—
S(6)-2	—	—	—

（７）その他の文書との対応関係

各要求事項に付記した取組例は、以下の文書も参考としている。

要求事項	その他の関連する文書
S(1)-1	BSA
S(1)-2	BSA, CISA-D
S(1)-3	SP800-161, CISA-D, ISO15408
S(1)-4	ISMS, DSP
S(2)-1	SP800-161, CISA-D, BSA
S(2)-2	BSA, CISA-D, NSA
S(2)-3	BSA, SP800-161, 総務省, CISA-D, ISO15408, DSP, ISMS
S(2)-4	BSA, ISO15408, CISA-D
S(3)-1	SP800-161, BSA, ISMS, DSP, CISA-D
S(3)-2	CISA-D, BSA
S(3)-3	SP800-161, ISMS
S(4)-1	SP800-161, ISMS, CISA-D, BSA
S(4)-2	SP800-161, CISA-D, DSP, 日米豪印サイバーセキュリティ・パートナーシップ
S(4)-3	DSP, 日米豪印サイバーセキュリティ・パートナーシップ
S(4)-4	SP800-161, CISA-D, 日米豪印サイバーセキュリティ・パートナーシップ
S(4)-5	SP800-161, CISA-D, 日米豪印サイバーセキュリティ・パートナーシップ
S(4)-6	SP800-161, 日米豪印サイバーセキュリティ・パートナーシップ
S(5)-1	NSA, DSP, ISMS
S(5)-2	DSP, 日米豪印サイバーセキュリティ・パートナーシップ
S(6)-1	CISA-D
S(6)-2	BSA, CISA-D

5.9. 用語

アジャイル開発	ソフトウェア開発ライフサイクル（SDLC）のフェーズを複数の開発サイクルに配置し、各フェーズを迅速に繰り返すことで、ソフトウェアを段階的に更新する開発プロセスのこと。
回帰テスト	プログラムを変更したことで、変更していない箇所に不具合が出ていないかを確認するためのテストのこと。
ソフトウェア開発ライフサイクル SDLC （Software Development Life Cycle）	短期間で、高品質、低コストのソフトウェアの生産を可能にする開発プロセスのこと。ウォーターフォール開発や、アジャイル開発等の種類がある。
ソフトウェアサプライチェーン	ソフトウェアの設計、開発、供給、運用の全てに関わるライフサイクルと、関連する組織及びソフトウェアの相互依存関係のこと。
脆弱性深刻度スコア CVSS（Common Vulnerability Scoring System）	情報システムの脆弱性に対するオープンで汎用的な評価手法のこと。CVSS を用いることで、特定の条件下で、脆弱性の深刻度を定量的に比較することができる。
セキュアバイデザイン SBD （Secure by Design）	ソフトウェアの設計段階から情報セキュリティを確保するための理念又は方策のこと。セキュリティバイデザインと表現する場合もあるが同義である。セキュアバイデザインの用語は、セキュアバイデフォルトを包含する。セキュアバイデザインの原則を通して主体的に顧客のセキュリティ確保に取り組むために、CISA が国家サイバー統括室を含む国際的パートナーと提携して発行した「Secure by Design ソフトウェアの原則とアプローチ」では、以下の 3 つのソフトウェア製品のセキュリティ原則が提起されている。 原則 1 :顧客にもたらされるセキュリティの結果に責任を負う 原則 2 :徹底的な透明性と説明責任を果たす 原則 3 :トップ主導 なお、類似用語である「シフトレフト」は、ソフトウェア開発の上流でセキュリティ対策を組み込むことである。
セキュアバイデフォルト	ソフトウェアのセキュリティ機能や設定を最初から（デフォルトで）組み込んだ状態にする理念又は方策のこと。 例えば、製品を購入し使用開始する最初の段階で、十

	分な強度のパスワードを設定しなければ他の機能を利用できないようにする、一般ユーザが通常使う必要のない機能は、デフォルトでは利用できないようにする、などは、セキュアバイデフォルトの理念に則った具体例である。
セキュリティ要件	製品やシステムの開発時や導入時などに満たすことが求められるセキュリティの目標に対する具体的な要求のこと。
ツールチェーン	ソフトウェア開発において必要となる機能を有するソフトウェアツールの一連のセットのこと。各ツールを連携させることで、開発作業の効率化を図ることを目的とする。
ハードニング	システムの脆弱性や、不要な機能を減らすことを通じて、セキュリティを強化すること。
バリューストリームマッピング VSM （Value Stream Mapping）	ソフトウェアなどの製品を顧客に提供するための開発・運用プロセスにおける必要な材料と情報等の流れを分析、設計、管理するためのリーン生産方式の手法のこと。
ビルドパイプライン	ビルド作業を複数のテスト等のプロセスに分割し、段階的に実行すること。CI：継続的インテグレーション（DevSecOps 参照）のプラクティスの１つ。
ピアレビュー・リードレビュー	経験やノウハウを活用しながら、同レベルの開発者やリーダーが成果物を診断、評価する活動のこと。
ウォークスルー	仕様書等の成果物の品質向上のため、成果物の作成者に加えて、開発関係者などを集めて行う机上のレビューのこと。システムの仕様やプログラムの問題を発見する手法。
リスクモデリング	発生する可能性のある脅威、危険、イベントなどの見込みを把握し、その望ましくない結果や問題を特定する分析手法のこと。ソフトウェアのリスクモデリングでは、脅威モデリング（情報資産を保護する観点から、ソフトウェアの特性や潜在的な攻撃者、攻撃手法を仮定した分析によりセキュリティ対策を検討する手法）を活用し、その過程で攻撃モデル（攻撃者、攻撃サーフェス、攻撃手法などによる攻撃者が取りうる行動のモデル）を用いる。
レジリエンス	「弾力」「回復力」「復元力」「耐久力」などと訳される用語。サイバーセキュリティにおいては、システムが攻撃を受けても、適切な対応により、被害を限定しそこから復旧できる能力のこと。
CSIRT （Computer Security Incident Response Team）	インシデント発生に対応する組織のこと。

DevSecOps	開発（Development）、セキュリティ（Security）、運用（Operations）の頭文字を組み合わせた造語であり、ソフトウェア開発プロセスのあらゆるフェーズでセキュリティテストを統合するプラクティスのこと。 「CI/CD パイプライン」は、その概念の一部を実装しており、ソフトウェアを継続的・段階的に更新し、自動的なビルドとテストによる検証を経て、自動的に配備される仕組みである。なお、CI/CD は（Continuous Integration / Continuous Delivery 又は Deployment）の略。
IaaS （Infrastructure as a Service）	情報システムを稼働するために必要なネットワーク、ストレージなどの基盤をインターネットを介してサービスとして提供すること。
ICT （Information and Communication Technology）	情報や通信に関する技術の総称。
統合開発環境 IDE （Integrated Development Environment）	効率的にソフトウェアコードを開発するために必要な機能を統合したソフトウェアのこと。
IoC （Indicator of Compromise）	サイバー攻撃などの侵害痕跡や指標のこと。
IoT （Internet of Things）	センサー機器などの「モノ」をインターネットに接続する仕組みのこと。
ISAC （Information Sharing and Analysis Center）	米国で国家の重要情報ネットワークを保護するために、重要インフラを構成する民間の各業種において設置が促されたのが始まりで、業界ごとのセキュリティなどの情報共有の推進を図る組織のこと。日本では、SoftwareISAC、金融 ISAC や交通 ISAC などが設立されている。
KPI （Key Performance Indicator）	組織の目標の達成度合いを観察するために用いられる定量的な指標のこと。
重要リスク指標 KRI （Key Risk Indicator）	組織内のリスクレベルを観察するために用いられる指標のこと。
MISP （Malware Information Sharing Platform）	マルウェアの通信先の IP アドレス等のサイバー攻撃の痕跡である IoC（Indicator of Compromise）の蓄積と共有を目的としたオープンソースの脅威共有プラットフォームのこと。 https://www.misp-project.org/

OSS （Open Source Software）	ソースコードが公開され、改変等が認められているソフトウェアのこと。
OT （Operational Technology）	工場、プラント、ビルなどの物理的なシステムや設備を制御し運用する技術の総称。
PaaS （Platform as a Service）	情報システムを稼働するために必要なアプリケーション用のプラットフォーム機能を、インターネットを介してサービスとして提供すること。
PSIRT （Product Security Incident Response Team）	自社で開発する製品やサービスのセキュリティの向上及びインシデント対応に取り組む組織のこと。
SaaS （Software as a Service）	情報システムサービスを、インターネットを介してサービスとして提供すること。
SBOM （Software Bill Of Materials）	ソフトウェアを構成するコンポーネントや互いの依存関係、ライセンスデータなどの一連の関連要素をリスト化して管理する手法のこと。
SLA （Service Level Agreement）	サービスを提供する事業者とサービス利用者間で、サービスの範囲、内容、達成目標等について合意した内容のこと。
SOC （Security Operation Center）	ネットワーク機器やサーバのログ等を監視し、サイバー攻撃やその予兆を検知、分析する専門組織のこと。

6. 本ガイドライン（案）の検討体制

サイバーインフラ事業者に求められる役割等の検討会

「サイバーインフラ事業者に求められる役割等の検討会」は、経済産業省 産業サイバーセキュリティ研究会 ワーキンググループ１ 分野横断サブワーキンググループ 及び サイバーセキュリティ戦略本部 重要インフラ専門調査会 の合同ワーキンググループとして、2024 年 9 月より開催し、ソフトウェアサプライチェーンのレジリエンス向上を図ることを目的としたサイバーインフラ事業者に求められる役割等に関して幅広い議論を行ってきた（令和７年７月重要インフラ専門調査会の廃止に伴い、経済産業省 産業サイバーセキュリティ研究会 ワーキンググループ１ 分野横断サブワーキンググループ 及び 内閣官房国家サイバー統括室 の合同ワーキンググループとして位置付け変更）。

検討会での議論を通じ、ソフトウェアの設計、開発・供給・運用に関わるサイバーインフラ事業者と顧客に求められる責務、及び責務を果たすための要求事項（役割別の具体的な取組の在り方）をまとめたガイドライン（案）の策定を進めるとともに、その普及策（自己適合宣言の仕組み化等）の検討を進めた。

＜構成員一覧＞

※敬称略、五十音順、2025 年 2 月 18 日時点

	阿部 恭一	ANA システムズ株式会社セキュリティマネジメント部エグゼクティブマネージャー 兼 株式会社レオンテクノロジー相談役
	稲垣 隆一	稲垣隆一法律事務所 弁護士
	鴨田 浩明	株式会社 NTT データ ソリューション事業本部セキュリティ＆ネットワーク事業部長
	木谷 浩	一般社団法人 情報サービス産業協会 サイバーセキュリティ部会 部会長 （キヤノン I T ソリューションズ株式会社 サイバーセキュリティ技術開発本部先進技術グループ）
	立石 聡明	一般社団法人 IT 団体連盟理事 （日本インターネットプロバイダー協会副会長・専務理事）
	津田 宏	富士通株式会社 富士通研究所 フェロー
座長	土居 範久	慶應義塾大学 名誉教授
	板東 直樹	一般社団法人 ソフトウェア協会（SAJ）フェロー（Software ISAC 共同代表）
	日高 昇治	一般社団法人 日本クラウド産業協会（ASPIC）執行役員
	淵上 真一	日本電気株式会社 Corporate Executive CISO 兼 サイバーセキュリティ戦略統括部長

古田 朋司　　トヨタ自動車株式会社 情報セキュリティ・トラスト部長

山口 雅史　　NRI セキュアテクノロジーズ株式会社 コンサルティング事業統括本部長

（事務局）

経済産業省、内閣官房 内閣サイバーセキュリティセンター

（オブザーバー）

警察庁、総務省、厚生労働省、防衛装備庁、デジタル庁、一般社団法人 日本医療機器産業連合会

＜各回概要＞

検討会日程	主な議題及び会議要旨
第 1 回検討会 （令和 6 年 9 月 24 日）	【議題】 責務と要求事項、及び検討の進め方に関する議論 【会議要旨】 サイバーインフラ事業者に求められる「責務」と「要求事項」、及びガイドライン（案）としての検討の進め方の議論を実施。
第 2 回検討会 （令和 6 年 12 月 17 日）	【議題】 文献調査及びヒアリング結果を踏まえたガイドライン（案）の審議、及びガイドライン附属書方針に関する議論 【会議要旨】 ガイドライン（案）、及びガイドラインの普及施策の議論を実施。
第 3 回検討会 （令和 7 年 2 月 18 日）	【議題】 ガイドライン（案）の承認、及び今後の普及方針の検討に関する議論 【会議要旨】 ガイドライン（案）の更新内容に関する審議、及びガイドライン（案）の活用促進に向けた取組及び普及施策の議論を実施。

要求事項チェックリスト									
	要求パッケージ		責務を果すための要求事項						
Check!	最低限	標準	要求ID	個別要求タイトル	個別要求	開発者	供給者	運用者	顧客
☐	○	○	S(1)-1.1	リスクベースのセキュリティ要件の定義	開発するソフトウェア、あるいはソフトウェアで構成されるシステム・サービスに対して、リスクベースの分析・評価を実施し、緩和策となるセキュリティ要件を定義する。	✓			
☐	○	○	S(1)-1.2	設計レビュー	ソフトウェアの設計のレビューを通じて、全てのセキュリティ要件を満たし、識別されたリスク情報に十分に対応していることを確認し、レビュー結果を反映する。	✓			
☐		○	S(1)-1.3	リスク対応記録	設計上の決定事項、リスクへの対応、承認された例外措置に関する記録を保持し、ソフトウェアのライフサイクル全体を通じて監査や保守の目的で使用できるように維持する。	✓			
☐		○	S(1)-1.4	リスクベースの定期的確認	セキュリティ要件に対して承認された全ての例外とソフトウェア設計、及びソフトウェアの設計時に作成したリスクベースの分析・評価結果をレビューし、リスクへの対処が適切か定期的に確認する。	✓			
☐	○	○	S(1)-2.1	セキュア開発プロセスの定義	セキュアコーディングの観点、ビルド実施タイミングと方式、自動化ツールの利用、トレーニングなど、セキュアコーディング、セキュアビルド及びデフォルトセキュアに関するプロセスを定義する。	✓			
☐	○	○	S(1)-2.2	セキュアビルド	実行可能形式のセキュリティを向上させる機能を提供するコンパイラ、インタプリタ、及びビルドツールを使用し、コードを生成・ビルドする。	✓			
☐	○	○	S(1)-2.3	検証とフィードバック	レビュー及び分析による検証により発見された問題の根本原因を特定し、その対応結果をプロセスにフィードバックする。	✓			
☐	○	○	S(1)-2.4	コードベース	レビュー及び分析の対象は、ソースコードのみでなく、可読性があると組織が決定したさまざまな形式のコード（設定ファイル等）も対象とする。	✓			
☐	○	○	S(1)-3.1	テスト計画	脅威モデルとリスク分析に基づき、テスト範囲及びテスト方式を決定し、テスト計画を立案する。	✓			
☐	○	○	S(1)-3.2	テスト方式	テスト方式には、機能テスト、脆弱性テスト、ファジング、侵入テストなどを含める。	✓			
☐	○	○	S(1)-3.3	テスト実施	テスト計画にしたがってテストを設計、実施し、結果を文書化する。	✓			
☐	○	○	S(1)-3.4	問題への対応	テストの結果、発見された全ての問題と推奨される対応策を開発チームのワークフローに組み込み、対処する。	✓			
☐	○	○	S(1)-4.1	資産管理	運用者は、システム・サービスが扱う資産、及びシステム・サービスを構成する資産に関する資産管理手順と資産リストを整備する。			✓	
☐		○	S(1)-4.2	モニタリング環境の整備	運用者は、リスク発生時の潜在的な影響を最小化するためにシステムを適切に分離し、ソフトウェアによる資産保護上重要なリスクを監視するモニタリング環境を整備する。			✓	
☐		○	S(1)-4.3	セキュリティメカニズムの整備	ソフトウェア及びソフトウェアを適用するシステム・サービスが、動作環境またはデジタルインフラなどのリソース上にある情報資産及びデータの機密性・完全性を保護し、監視可能とするための適切なセキュリティメカニズムを整備する。	✓		✓	
☐	○	○	S(1)-4.4	モニタリングと評価	運用者は、重要なサービスを提供するソフトウェアに適用したメカニズムの動作状況をモニタリングするとともに、定期的にセキュリティ評価を実施し、組織のリスク管理の枠組みに統合する。			✓	
☐	○	○	S(2)-1.1	ソフトウェアコンポーネントの手配	外部から手配する商用、オープンソース、その他のサードパーティのソフトウェアコンポーネントは、組織が定義した要件を満たす安全性の高いものを採用する。	✓			
☐	○	○	S(2)-1.2	ソフトウェアコンポーネントの開発・維持	外部からソフトウェアコンポーネントを手配しない場合、組織で確立されたセキュリティ基準・慣行にしたがい、安全性の高いソフトウェアコンポーネントを社内で開発、維持する。	✓			
☐	○	○	S(2)-1.3	ソフトウェアコンポーネントのリスク評価	各ソフトウェアコンポーネントの出所情報を取得・分析し、そのコンポーネントがもたらすリスクを評価する。	✓			
☐	○	○	S(2)-1.4	ソフトウェアコンポーネントの公知脆弱性の確認	各ソフトウェアコンポーネントの公知脆弱性、サポート期間を定期的にチェックする。	✓			
☐	○	○	S(2)-1.5	ソフトウェアコンポーネントの更新	各ソフトウェアコンポーネントを新しいバージョンにセキュアに更新するプロセスを導入する。	✓			
☐	○	○	S(2)-2.1	コードベースの保護	全ての形式のコードベースを不正アクセスや改ざんから保護するために、リポジトリにコードや設定情報を保管し、承認された担当者、ツール、サービスなどのみがアクセスできるよう最小権限の原則に基づいたアクセス制御を実施する。	✓	✓		
☐	○	○	S(2)-2.2	リリースのアーカイブ	リリース後に発見された脆弱性を分析、特定できるようにするために、各ソフトウェアのリリースをアーカイブ化して保護する。	✓	✓		
☐	○	○	S(2)-2.3	リリースの出所データの共有	各ソフトウェアリリースの全てのコンポーネントの出所データを収集、保護、維持、共有する。	✓	✓		
☐	○	○	S(2)-3.1	セキュリティ要件の合意	IT製品（自社のソフトウェアで再利用するための商用ソフトウェアコンポーネントを含む）またはサービスを提供するサードパーティとの契約または共有するポリシーに、明示的なセキュリティ要件を盛り込む。	✓	✓	✓	
☐		○	S(2)-3.2	サプライチェーンセキュリティ要求への対応	提供するIT製品またはサービスを受領・取得する組織が採用するサプライチェーンセキュリティ要件と同等のサプライチェーンセキュリティ要件に対応する。	✓	✓		
☐		○	S(2)-3.3	セキュリティ要件を満たさないリスクへの対処プロセスの整備	受領・取得するサードパーティ製のIT製品またはサービスが満たさないセキュリティ要件がある場合のリスクに対処するプロセスを整備する。	✓	✓	✓	
☐	○	○	S(2)-4.1	セキュアな導入・設定・操作・変更・廃棄・終了	ソフトウェアをセキュアに導入・設定・操作するための情報、及び変更の影響・廃棄・提供終了・利用終了に係る情報をソフトウェア利用者が継続的に利用できるようにする。	✓	✓		
☐	○	○	S(2)-4.2	整合性検証情報の提供	ソフトウェアの整合性・完全性の検証に必要な情報をソフトウェア利用者が継続的に利用できるようにする。	✓	✓		
☐	○	○	S(3)-1.1	脆弱性対応体制の設置	ソフトウェア製品の脆弱性の開示と修復に対処するポリシーを定め、そのポリシーをサポートするための脆弱性対応（インシデント対応を含む）に関する体制を設置し、必要な役割、責務、プロセスを定義する。	✓		✓	
☐	○	○	S(3)-1.2	コミュニケーション計画	全ての利害関係者に対するコミュニケーション計画を定める。	✓		✓	
☐	○	○	S(3)-1.3	脆弱性情報の収集	公知情報の探索、ソフトウェア利用者からの通知、外部脅威情報の取得、システム構成データのレビュー、その他の方法を通じて、新たな脆弱性情報を収集する。	✓		✓	
☐	○	○	S(3)-1.4	未検出の脆弱性の特定	継続的または定期的に、ソフトウェアのコードのレビュー、分析、テストを実施し、今まで未検出の対処すべき脆弱性（不適切な設定などを含む）を特定する。	✓		✓	
☐	○	○	S(3)-2.1	脆弱性の分析	開発者は、残存する各脆弱性の影響に伴うリスクを把握するために必要な情報を収集し、修復またはその他のリスク対応を計画するために、各脆弱性を分析する。	✓			
☐	○	○	S(3)-2.2	脆弱性へのリスク対応	開発者は、各脆弱性に対するリスク対応を計画し、実装する。	✓			
☐	○	○	S(3)-2.3	セキュリティ勧告	開発者は、セキュリティ勧告を作成し、リリースしたソフトウェアの供給先にその情報を提供するとともに、関連する制度の指定にしたがって報告する。また、運用者はセキュリティ勧告にしたがった配備を実施する。	✓	✓	✓	
☐		○	S(3)-3.1	根本原因の特定	根本原因を決定するために、識別された脆弱性を分析し、プロアクティブに対策する。	✓		✓	
☐		○	S(3)-3.2	プロセス改善	ソフトウェアの更新または作成された新しいソフトウェアにより、根本原因の再発を防止またはその可能性を低減するために、ソフトウェアライフサイクル全体の開発と運用のプロセスをレビューし、必要に応じて見直す。	✓		✓	
☐		○	S(4)-1.1	役割と責務の定義	ソフトウェア開発ライフサイクルを網羅する役割と責務を定義する。	✓	✓	✓	
☐	○	○	S(4)-1.2	経営層のコミットメント	全要員に対してセキュア開発に対する経営層のコミットメントを周知し、組織にとってのセキュアな開発・運用の重要性を教育する。	✓	✓	✓	
☐		○	S(4)-1.3	役割と責務の同意	各要員が、役割と責務を認識・同意していることを確認する。	✓	✓	✓	
☐		○	S(4)-1.4	各役割のトレーニング	各役割のトレーニング計画を作成し、全要員が習熟度と役割に応じてトレーニングを実施できるように提供する。	✓	✓	✓	
☐		○	S(4)-1.5	役割とトレーニングの見直し	役割やトレーニングは定期的に見直す。	✓	✓	✓	
☐	○	○	S(4)-2.1	ソフトウェア開発ポリシーの定義	ソフトウェア開発のインフラ及びプロセスの全てのセキュリティ要件（EOLに係る要件を含む）を特定し、法令遵守のもとSDLC全体を通して維持するためのセキュリティポリシーを定義する。	✓			
☐	○	○	S(4)-2.2	ソフトウェア・セキュリティポリシーの定義と維持	組織が開発するソフトウェアが満たすべき全てのセキュリティ要件を規定したポリシーを定義し、これらの要件をSDLC全体にわたって維持する。	✓			
☐	○	○	S(4)-2.3	費用認識の共有と予算化	ポリシーに基づいてセキュリティを確保するために必要な予算を確保する。	✓			

要求事項チェックリスト									
	要求パッケージ		責務を果すための要求事項						
Check!	最低限	標準	要求ID	個別要求タイトル	個別要求	開発者	供給者	運用者	顧客
☐		○	S(4)-3.1	ソフトウェアサービス運用ポリシーの定義	ソフトウェアを適用したサービス運用インフラ及びプロセスの全てのセキュリティ要件（EOS及び廃棄に係る要件を含む）を特定し、法令遵守のもとSDLC全体を通じて維持するためのセキュリティポリシーを定義する。			✓	
☐		○	S(4)-3.2	サービス・セキュリティポリシーの定義と維持	ソフトウェアを適用したサービスが満たすべき全てのセキュリティ要件を規定したポリシーを定義し、これらの要件をSDLC全体にわたって維持する。			✓	
☐		○	S(4)-3.3	運用ポリシーに基づく監査	ポリシーに基づくガバナンスにより、サービス運用インフラ及びプロセスの保護、及びサービスのセキュリティ要件がSDLC全体にわたって維持されていることを監査により確認する。			✓	
☐	○	○	S(4)-4.1	セキュリティ確認基準の定義と追跡	ソフトウェアのセキュリティ確認基準を定義し、SDLC全体を追跡する。	✓		✓	
☐	○	○	S(4)-4.2	セキュリティ確認基準に基づく意思決定のサポート	セキュリティ確認基準に基づく意思決定をサポートするために必要な情報を収集し保護するためのプロセスや仕組みなどを実装する。	✓		✓	
☐		○	S(4)-4.3	セキュリティ確認基準に基づく監査	セキュリティ上の確認基準への適合を遵守するためのガバナンスにより、SDLC全体を追跡し意図する効果を得ていることを監査により確認する。	✓		✓	
☐	○	○	S(4)-5.1	ツールとツールチェーンの指定	特定されたリスクを軽減するために有効なツールを特定し、どのツールチェーンに含めることが必須もしくは必要であるか、及びツールチェーンのコンポーネントを相互に統合する方法を指定する。	✓			
☐	○	○	S(4)-5.2	ツールとツールチェーンの配備・運用・保守	セキュリティ慣行にしたがってツールとツールチェーンを配備、運用、及び保守する。	✓			
☐	○	○	S(4)-5.3	ツール構成と証跡生成	組織によって定義されたセキュアなソフトウェア開発の慣行のサポートに関する証跡を生成するようにツールを構成する。	✓			
☐	○	○	S(4)-6.1	環境の分離保護	ソフトウェア開発に関係する各環境を分離して保護する。	✓			
☐	○	○	S(4)-6.2	開発用エンドポイントの保護	リスクベースのアプローチを使用して開発関連のタスクを実行するために、各開発者向けのエンドポイントを保護、強化する。	✓			
☐		○	S(5)-1.1	情報連携のための組織体制の構築	ソフトウェアの製品及びサービスのセキュリティを改善するために、民間企業同士、関係当局、専門組織との情報連携のための組織体制を構築する。	✓	✓	✓	
☐	○	○	S(5)-1.2	重要なセキュリティ関連情報の提供	業界固有の必須かつ重要なセキュリティ関連情報を選別・識別して、サプライチェーン先に提供する。	✓	✓	✓	
☐	○	○	S(5)-1.3	脆弱性情報の通知サービスの利用	効率的に脆弱性情報の共有を図るため、脆弱性情報の通知サービスを利用する。	✓	✓	✓	
☐		○	S(5)-2.1	協力体制の活用	ソフトウェアの製品及びサービスのセキュリティを改善するために、外部の事業者、顧客、及び専門機関が参加するソフトウェアセキュリティの改善を目的とするコミュニティや協力体制を活用する。	✓	✓	✓	
☐		○	S(5)-2.2	協力体制への貢献	コミュニティや協力体制に参加する場合には、積極的に活動に関与し、協力体制に対して貢献する。	✓	✓	✓	
☐	○	○	S(6)-1.1	リスク管理	顧客の独立した主体的な取組とサイバーインフラ事業者との契約に基づく取組を統合したリスク管理を実施する。				✓
☐	○	○	S(6)-1.2	リソース整備	既知の脆弱性への対処、及び緩和策を主体的に実施するためのリソースを割り当て、整備する（SBOM活用を含む）。				✓
☐		○	S(6)-1.3	協力体制の活用	ソフトウェアセキュリティの改善を目的とするコミュニティや協力体制を活用する。				✓
☐	○	○	S(6)-2.1	セキュリティ要件の定義	ソフトウェア設計計画にセキュリティ機能を組み込むためのセキュリティ要件を定義し、ソフトウェアを調達・導入する前に、サイバーインフラ事業者に提示する。				✓
☐	○	○	S(6)-2.2	セキュリティ慣行の要求開示	ソフトウェアの調達・導入前に、サイバーインフラ事業者に求めるセキュリティ慣行の要求を開示する。				✓
☐	○	○	S(6)-2.3	リスク評価に基づく意思決定	ソフトウェアを調達・導入する際に、リスク評価に基づいた意思決定を行う。				✓
☐	○	○	S(6)-2.4	予算確保	ソフトウェアのライフサイクルを考慮した導入・運用・移行・廃棄、リスク対応、及び関連する契約に係る予算を継続的に確保する。				✓

要求事項チェックリスト（役割・フェーズ）

	要求パッケージ		責務を果すための要求事項						ライフサイクルフェーズ														
Check!	最低限	標準	要求ID	個別要求タイトル	開発者	供給者	運用者	顧客	要件定義	設計	開発	ビルド	テスト	リリース	配布	配備	運用	モニタリング	フィードバック	分析	計画	人材・プロセス・技術の整備	サイバーインフラ事業者・ステークホルダー間の関係強化
☐	○	○	S(1)-1.1	リスクベースのセキュリティ要件の定義	✓				✓												✓		
☐	○	○	S(1)-1.2	設計レビュー	✓					✓													
☐		○	S(1)-1.3	リスク対応記録	✓					✓	✓	✓	✓	✓						✓	✓		
☐		○	S(1)-1.4	リスクベースの定期的確認	✓															✓	✓		
☐	○	○	S(1)-2.1	セキュア開発プロセスの定義	✓						✓	✓											
☐	○	○	S(1)-2.2	セキュアビルド	✓							✓											
☐	○	○	S(1)-2.3	検証とフィードバック	✓						✓	✓											
☐	○	○	S(1)-2.4	コードベース	✓						✓	✓											
☐	○	○	S(1)-3.1	テスト計画	✓								✓								✓		
☐	○	○	S(1)-3.2	テスト方式	✓								✓										
☐	○	○	S(1)-3.3	テスト実施	✓								✓										
☐	○	○	S(1)-3.4	問題への対応	✓								✓										
☐	○	○	S(1)-4.1	資産管理			✓										✓						
☐		○	S(1)-4.2	モニタリング環境の整備			✓										✓	✓					
☐		○	S(1)-4.3	セキュリティメカニズムの整備	✓		✓										✓	✓					
☐	○	○	S(1)-4.4	モニタリングと評価			✓											✓	✓	✓			
☐	○	○	S(2)-1.1	ソフトウェアコンポーネントの手配	✓				✓	✓	✓												
☐	○	○	S(2)-1.2	ソフトウェアコンポーネントの開発・維持	✓						✓												
☐	○	○	S(2)-1.3	ソフトウェアコンポーネントのリスク評価	✓					✓										✓			
☐	○	○	S(2)-1.4	ソフトウェアコンポーネントの公知脆弱性の確認	✓															✓	✓		
☐	○	○	S(2)-1.5	ソフトウェアコンポーネントの更新	✓						✓												
☐	○	○	S(2)-2.1	コードベースの保護	✓	✓								✓									
☐	○	○	S(2)-2.2	リリースのアーカイブ	✓	✓								✓	✓					✓			
☐	○	○	S(2)-2.3	リリースの出所データの共有	✓	✓								✓	✓								
☐	○	○	S(2)-3.1	セキュリティ要件の合意	✓	✓	✓		✓														
☐		○	S(2)-3.2	サプライチェーンセキュリティ要求への対応	✓	✓			✓	✓													
☐		○	S(2)-3.3	セキュリティ要件を満たさないリスクへの対処プロセスの整備	✓	✓	✓		✓	✓													
☐	○	○	S(2)-4.1	セキュアな導入・設定・操作・変更・廃棄・終了	✓	✓								✓	✓	✓							
☐	○	○	S(2)-4.2	整合性検証情報の提供	✓	✓								✓	✓	✓							
☐	○	○	S(3)-1.1	脆弱性対応体制の設置	✓		✓														✓		
☐	○	○	S(3)-1.2	コミュニケーション計画	✓		✓														✓		
☐	○	○	S(3)-1.3	脆弱性情報の収集	✓		✓											✓		✓			
☐	○	○	S(3)-1.4	未検出の脆弱性の特定	✓		✓			✓			✓							✓			
☐	○	○	S(3)-2.1	脆弱性の分析	✓														✓	✓			
☐	○	○	S(3)-2.2	脆弱性へのリスク対応	✓					✓	✓	✓	✓							✓	✓		
☐	○	○	S(3)-2.3	セキュリティ勧告	✓	✓	✓							✓	✓		✓						
☐		○	S(3)-3.1	根本原因の特定	✓		✓			✓	✓									✓			
☐		○	S(3)-3.2	プロセス改善	✓		✓													✓	✓		
☐		○	S(4)-1.1	役割と責務の定義	✓	✓	✓															✓	
☐	○	○	S(4)-1.2	経営層のコミットメント	✓	✓	✓															✓	
☐		○	S(4)-1.3	役割と責務の同意	✓	✓	✓															✓	
☐		○	S(4)-1.4	各役割のトレーニング	✓	✓	✓															✓	
☐		○	S(4)-1.5	役割とトレーニングの見直し	✓	✓	✓															✓	
☐	○	○	S(4)-2.1	ソフトウェア開発ポリシーの定義	✓																	✓	
☐	○	○	S(4)-2.2	ソフトウェア・セキュリティポリシーの定義と維持	✓																	✓	
☐	○	○	S(4)-2.3	費用認識の共有と予算化	✓																	✓	
☐		○	S(4)-3.1	ソフトウェアサービス運用ポリシーの定義			✓															✓	

□		○	S(4)-3.2	サービス・セキュリティポリシーの定義と維持			✓															✓	
□		○	S(4)-3.3	運用ポリシーに基づく監査			✓											✓				✓	
□	○	○	S(4)-4.1	セキュリティ確認基準の定義と追跡	✓		✓															✓	
□	○	○	S(4)-4.2	セキュリティ確認基準に基づく意思決定のサポート	✓		✓															✓	
□		○	S(4)-4.3	セキュリティ確認基準に基づく監査	✓		✓											✓				✓	
□	○	○	S(4)-5.1	ツールとツールチェーンの指定	✓																	✓	
□	○	○	S(4)-5.2	ツールとツールチェーンの配備・運用・保守	✓																	✓	
□	○	○	S(4)-5.3	ツール構成と証跡生成	✓																	✓	
□	○	○	S(4)-6.1	環境の分離保護	✓																	✓	
□	○	○	S(4)-6.2	開発用エンドポイントの保護	✓																	✓	
□		○	S(5)-1.1	情報連携のための組織体制の構築	✓	✓	✓																✓
□	○	○	S(5)-1.2	重要なセキュリティ関連情報の提供	✓	✓	✓																✓
□	○	○	S(5)-1.3	脆弱性情報の通知サービスの利用	✓	✓	✓																✓
□		○	S(5)-2.1	協力体制の活用	✓	✓	✓																✓
□		○	S(5)-2.2	協力体制への貢献	✓	✓	✓																✓
□	○	○	S(6)-1.1	リスク管理				✓															
□	○	○	S(6)-1.2	リソース整備				✓															
□		○	S(6)-1.3	協力体制の活用				✓															
□	○	○	S(6)-2.1	セキュリティ要件の定義				✓															
□	○	○	S(6)-2.2	セキュリティ情行の要求開示				✓															
□	○	○	S(6)-2.3	リスク評価に基づく意思決定				✓															
□	○	○	S(6)-2.4	予算確保				✓															