Open Source Software(OSS)
に係るヒヤリング調査

オープンソースソフトウェアの脆弱性情報管理に関する戦略策定報告書（抜粋）

令和２年３月

一般財団法人　機械システム振興協会

委託先　一般社団法人コンピュータソフトウェア協会

※本資料用に項番や体裁の修正を行っております。内容は報告書と同様です。

OSS に係るヒヤリング調査結果

Software ISAC や今回の戦略策定委員会の委員の協力を賜り、合計 21 社に対するヒヤリングを実施した。本章ではヒヤリングの結果をまとめて、現状や課題を明らかにする。

1. 調査方法

調査を行うにあたって、ヒヤリング項目の精査を行い、以下の項目の設問を用意（表 2-1 参照）し、ヒヤリングを行った。次からヒヤリングを行うにあたって作成した質問表について、背景や概要を解説する。

まずは、ソフトウェアの開発を行っているとはいえ、自社製品だけの開発を行っているわけではなく、受託開発を行っている企業も非常に多いことを想定し、受託開発実施状況に関する確認を行うための質問を含めた。さらに、開発も自組織だけで完結しているのではなく、日本の多重下請け構造を踏まえ、外部委託や外部からの支援を受けて開発がしていることを確認するための質問も盛り込んだ。当該割合が多ければ多いほど、日本の多重下請け構造が証明され、ソフトウェアサプライチェーンの課題は益々深まることとなる。

また、本ヒヤリングの主たる目的ともいえる、OSS の開発における使用状況に関する質問も盛り込んだ。さらに、OSS 使用状況を確認するために、具体的なソフトウェアの名称をあげて、各社における開発に関する OSS の情報をヒヤリングした。これにより、国内における OSS の活用状況が可視化でき、国内の OSS 活用状況や傾向を認識することができる。一方で、OSS の活用はこれまでの情報や経験からも多いと考えられるが、使用していない場合の質問も盛り込み、使用しない理由の確認も行った。

OSS に関しては具体的な契約を踏まえた対応や、会社としての対応を行っていないことを想定し、契約や事前の協議状況なども質問に盛り込んだ。また、ソフトウェアコンポーネント管理は、通常の事業・業務にひっ迫していることを想定して、適切に実施している企業がどれほどあるか、SBOM というキーワードを含めることや、OSS の選定そのものを会社として行っているかどうかを確認するための質問も含めた。

さらには、納品形態の確認を行う質問を含めた。これは、日本のソフトウェア産業の構造的問題ともいえる仮説で、納品形態がバイナリ納品かソースコード納品かによって、ソフトウェア自体の構造把握や、脆弱性が確認された際の対応も大きく異なるからである。ソースコード納品は、アプリケーションのプログラムコードを納品する方法で、著作権を納品時に委託元に引き渡す契約形態が多い。言わば開発した自動車のエンジンがどのようなパーツ

で構成されていて、どのような形で動作するのかを示し、利用方法や自動車への実装方法、設定方法を適切に示すような納品である。一方でバイナリ納品とは、コンピュータが直接的に処理するために表現される 2 進数のデータ（バイナリデータ）を納品する方法である。これは著作権の権利を渡さずに開発元が今後も活用できるようにするために用いられる納品方法であり、中小企業のソフトウェア開発企業は、自分たちの権利を守り、事業を継続させるために当該納品形態をとっていることも推察した。なお、推察した背景には、実際にソースコード納品を行い、ソースを確認したところ、使用している OSS によっては工数の見直しを行うような指摘を受けることがあるといったケースが確認されているからである。受託する側としては、ソースコード納品はあまり取りたくない形態ではないかと考え、当該質問を含めた。

表 2-1：ソフトウェア開発に関するヒヤリングシート

設問	選択肢
右項目の情報についてご記入をお願い致します。	業態： 従業員数：
ソフトウェアの受託開発などを行っていますか？	□ 行っている □ 行っていない
ソフトウェア開発にあたって外部委託などの外部の協力を得ていますか？	□ 得ている □ 得ていない
現在、自社の開発において OSS を使用していますか？	□ 使用している □ 使用していない
（使用している場合）どのような OSS を使用していますか？	※表 2-2 参照
（使用していない場合）なぜ開発において OSS を使用しないのですか？	自由記入
委託企業でどのような OSS を使用しているか知っていますか？	□ 知っている □ 知らない
委託企業との間で OSS を含んだソフトウェアに脆弱性等セキュリティ上の問題があった場合に関する契約事項はありますか？	□ ない □ ある（対応はしない） □ ある（無償で納品時点における既知の脆弱性に対応する） □ ある（有償で対応する） □ ある（協議のうえ対応を検討する） □ ある（その他　　）

（契約がある場合）どのように対応をしていますか	□ ない □ ある（対応はしない） □ ある（無償で納品時点における既知の脆弱性に対応する） □ ある（有償で対応する） □ ある(協議のうえ対応を検討する） □ ある（その他　　）
使用するOSSの選定にあたって	□会社としてOSSの選定を行う □プロジェクトチームに選定を委ねている □エンジニアに OSS の選定を委ねている □その他（　　）
コンポーネント管理表（SBOMなど）を作成していますか？	□ 作成している □ 作成していない
（作成していない場合）作成しないのはなぜですか？	自由記入
依頼元とOSSの利用に関する取り決めや協議を事前に行いますか？	□ 行う □ 行わない
委託元との間で OSS を含んだソフトウェアに脆弱性等セキュリティ上の問題があった場合に関する契約事項はありますか？	□ ない □ ある（対応はしない） □ ある（無償で納品時点における既知の脆弱性に対応する） □ ある（有償で対応する） □ ある(協議のうえ対応を検討する） □ ある（その他　　）
使用するOSSの選定にあたって	□ 会社としてOSSの選定を行う □ プロジェクトチームに選定を委ねている □ エンジニアにOSSの選定を委ねている □ その他（　　）
ソフトウェアの納品方法として「ソースコード」と「バイナリ」それぞれどれくらいの割合で対応していますか？	ソースコード納品：　　％ バイナリ納品：　　％

PSIRT を知っていますか？	□ 知っている □ 知らない
PSIRT を構築していますか？	□ 構築している □ 構築していない
（未構築の場合）PSIRT を構築する予定はありますか？	□ 1 年以内に予定 □ 2−3 年以内 □ 将来的に構築したい □ 構築する予定はない
（PSIRT を知っていて、構築予定がない場合）PSIRT を構築しないのはなぜですか？	□ ヒト □ モノ □ カネ □ その他（　　）
脆弱性情報の収集を行っていますか？	□ 行っている □ 行っていない
（行っている場合）脆弱性情報の収集にどれくらいの頻度で実施していますか？	毎日／毎週／毎月／四半期／不定期
（行っている場合）脆弱性情報の収集に 1 日あたりどれくらい時間を費やしていますか？	5 分／10 分／15 分／30 分／60 分以内／2 時間以上／それ以上（　　）
開発したソフトウェアの検証を行っていますか？	□ 行っている □ 行っていない □ 案件によって異なる
自社で検証環境（人員や端末、ライセンス等）を整備できていますか？	□ 常設している □ 都度構築している □ 案件によって整備する □ 整備できていない
OSS の管理ができるポータルを準備したいと思っています。当該ポータルを使用したいですか？	□ 使用したい □ 使用したくない
当該ポータルからどのような情報を得たいですか？	□ OSS に関する全般的な情報（最新バージョンなど） □ 脆弱性情報／最新の脅威情報 □ 不確定な脆弱性情報(脆弱性の噂) □ OSS の学習／セキュア開発の学習 □ パッチ適用による弊害情報 □ その他（　　）

収集してほしい情報はありますか？	自由記入
ポータルやデータベースに期待することを教えてください。	自由記入

また、製品セキュリティを適切に維持管理するための組織である PSIRT の認知や設置状況を質問に含め、組織的に製品セキュリティに取り組むことができているのか確認を行ったり、まだ構築が進んでいないことを想定して、構築する時期に関する質問を追加したりした。

そもそも PSIRT は Product Security Incident Response Team の略称で、製品セキュリティに特化した組織であり、Secure Development Lifecycle（安全な開発ライフサイクル）を組織に実装するための組織である。当該ライフサイクルは、モノが作り上げられた後、または検証に近い段階でセキュリティを考えるのではなく、開発の各段階でセキュリティを考え実装しようとする考えである。図 2-1 に示す通り、セキュア開発ライフサイクルは、製品の企画段階から、設計、実装、検証、出荷、提供・保守、そして廃棄に至るプロセスでセキュリティを考えるものである。

図 2-1：セキュリティ開発ライフサイクル

企画、設計段階でセキュリティを考えることができず、重大なインシデントを起こしてしまった組織が出ているだけでなく、製品を組み上げ検証フェーズでセキュリティを考えたとしても、今更開発をやり直すことができないといったプロジェクトも存在する。また、適切な廃棄が行われておらず、廃棄した製品から重要な情報が漏えいするといったインシデント（HDD からの情報抜き取り）も発生している。そのため、PSIRT のような組織を設け、組織として製品やサービスに対するセキュリティをいかなるフェーズでも考えることは極めて重要であり、PSIRT の重要性は増している。

重要な組織となってきている PSIRT ではあるが、それでも構築できない組織は多数あることに鑑みて、構築できない理由を確認するための質問も追加する工夫も行った。

ヒヤリングの最後に、OSS の脆弱性管理ポータルやデータベースの構築概念や計画を示し、具体的にどのような要望があるか、そもそも使用したいかなどの質問を盛り込み、戦略策定委員会で検討するデータベースやポータルの要件定義に繋げるための質問も盛り込んでいる。

このように、製品セキュリティに関する取り組みを、組織的な視点や技術的な視点を考慮しながらヒヤリングシートを作成し、ヒヤリングを実施した。

また、具体的に使用しているソフトウェアについては、以下の製品調査表（表 2-2 参照）を用意し、ヒヤリングを行った。具体的には、OSS のみではないが使用されている傾向が強いと思われるソフトウェアについて、これまでの知見や有識者からの意見、JPCERT コーディネーションセンターの情報を参考にしながら絞り込みを行い、製品調査表を作成した。

表 2-2：製品調査表

分野	名称
OS	Android OS
	iOS
	macOS
	Debian
	Fedora
	Ubuntu
	RedHat/CentOS
	Windows
	その他（　　）
プロトコル	FTP (TLS)
	HTTP
	HTTPS
	IMAP (TLS)
	POP (TLS)
	SMTP (TLS)
	LDAP (TLS)
	NTP
	PPTP
	RDP/ICA
	RPC
	SMB
	SOAP
	その他（　　）
ミドルウェア／ソフトウェア	

Web／AP サーバ	Apache HTTP Server
	Apache Tomcat
	IIS
	Nginx
	その他（　　）
POP／IMAP／MTA	Courier-IMAP
	Cyrus IMAP
	Postfix
	Sendmail
	Sendgrid
データベース	MySQL
	PostgreSQL
	SQLite
	SQL Server
	Oracle
	その他（　　）
開発言語／マークアップ言語	C
	C++
	Java
	Javascript
	PHP
	Python
	Ruby
	Swift
	HTML
	HTML5
	.Net Framework（C#,VB.NET,Powershell 等）
その他	自由記載（　　）

なお、本来であればこのような開発に関わる情報は提供したくない企業が多く、データの取り扱いの徹底、利用範囲の限定（プロジェクトマネージャー、プロジェクトリーダー、ヒヤリング実施者、事務局のみ）、必ず個社の情報については公開しないことを約束（図 2-2 参照）して、ヒヤリングを行った。

ヒヤリング実施に伴う機密保持誓約及び統計データフィードバックについて

先方訪問企業名：
先方訪問者名：

一般社団法人コンピュータソフトウェア協会 Software ISAC （以下、「SW ISAC」という）では、本ヒヤリング実施にあたり、個人情報を含む機密情報（以下、「機密情報」という）について、下記の事項を誓約します。また、ヒヤリングにご協力いただきました企業には、統計データをフィードバックさせていただきます。

記

1. 次に掲げる機密情報は、業務遂行上必要最小限の範囲で取扱い、第三者に提供しません。また、機密情報を記録した文書、記録媒体、データ等は厳重に保管します。ただし、SW ISAC の責めによらずに公知となった情報、及び貴社 と SW ISAC が機密情報として扱わない旨を別途合意した情報についてはこの限りではありません 。

（１） 貴社の運営、業務、財務等に関する事実で、経営、評価等に多大な影響を及ぼす、未公表の情報
（２） 個人情報のうち、貴社が機密扱いとする情報
（３） その他、貴社が機密保持の対象として特定した情報

2. ヒヤリングした情報は、統計情報として利用させていただき、個社名および個社情報は公開しません。

以上

図 2-2：ヒヤリング実施に伴う機密保持誓約書

2-1. ヒヤリング対象の企業規模

組織の規模によって組織的な体力や対応なども異なることを想定して、できる限り幅広い企業規模からヒヤリングができるよう、従業員規模で 100 名以下、101～500 名、501～1,000 名以下、1,000 名以上の企業で、バランスがとれるよう配慮して実施した。(図 2-3 参照)

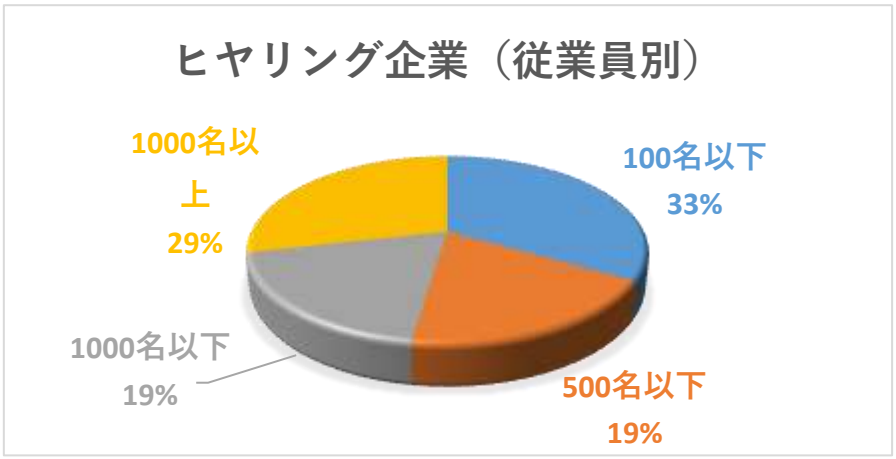

図 2-3：ヒヤリング企業従業員別割合

2-2. ソフトウェアの受託開発比率

ヒヤリングをした企業において、ソフトウェアの開発を行っているといっても、自社ですべての開発が完結しているわけではない。先にも述べた通り、日本は多重下請け構造であり、ソフトウェアの開発も同様に外部委託し、ソフトウェア、製品やサービスを完成させている。

今回ヒヤリングを行った企業の約 7 割でソフトウェアの受託開発を行っていることが分かった（図 2-4 参照)。なお、ヒヤリングをした企業によっては、事業の割合は異なるものの自社製品の開発や販売を行っている一方で、部分的に受託開発を行ってリレーション継続や強化、さらには企業の工数を有効活用している企業もあった。

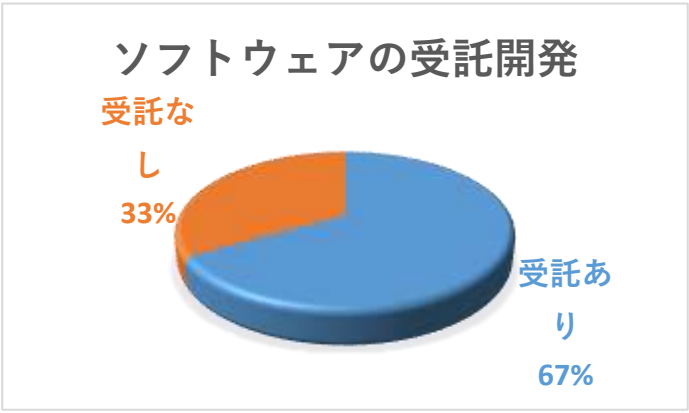

図 2-4：受託開発受け入れの現状

2-3.　ソフトウェア開発における外部委託の有無

企業において自社のみで開発を行うのではなく、外部の支援を活用（外部委託）している割合が 8 割に迫っている（図 2-5 参照）。なお、外部委託を行っていない企業においても人的支援を受けて業務の遂行をしている企業もある。自社で完結した開発の方がプロジェクトメンバーやコンポーネントやモジュールの管理など体制としては望ましいが、全ての開発を完結できない組織も多く、ソフトウェア産業界が支え合って事業を行っているともいえる。

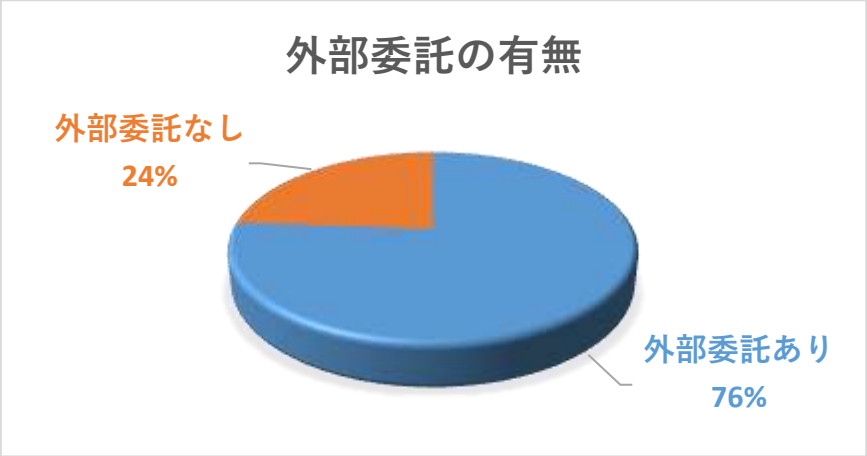

図 2-5：ソフトウェア開発における外部委託状況

2-4.　自社開発における OSS 使用有無

ソフトウェアの開発においては、OSS を使用して開発をしている企業がおよそ 7 割を占めている（図 2-6 参照）。今回の結果からみて、ソフトウェア開発における OSS の使用割合は高く、開発には OSS を欠かせないといえる。なお、自社サービスや製品としては使用しているが、受託開発案件には OSS を使用しないといった企業や、マイクロソフト社の開発環境をベースにした開発を行っている企業においては OSS を使用していないといった回答もあった。

図 2-6：OSS の使用状況

また、OSS の選定にあたっては、会社としての対応を行っているのは 1 組織しかなく、プロジェクトチームに選定を委ねているという組織が大半を占め、さらにはエンジニアに選定を委ねている組織を含めると 6 割を超える割合となっている。これは、OSS が組織的な課題として捉えられていない現実があるためと考えられる。OSS の使い方や脆弱性が生じた場合、自組織で開発し納入した成果物や、開発・販売している自組織の製品に影響を及ぼす可能性がある。OSS の課題は経営にも影響を及ぼす課題であることを戒め、OSS の選定は組織として責任をもって決定していく必要がある。

ソフトウェア開発の一部を外部委託等する場合に、OSS の使用状況を委託元として把握できている企業は概ね半分といった結果（図 2-7 参照）だが、当該質問についてはより深掘りをすべき設問であった。なぜならば、ヒヤリングを実施している中で、把握しているが細かいバージョンは分からない、そもそも OSS の情報まで公開しているメーカーのほうが少なくて分からないといった回答もあったからである。把握している企業においてもどのソフトウェアを使っているか概要を知っているだけで、具体的なバージョン情報などは持ち得ていない可能性の方が高いのではないかと考える。

また、OSS の選定については、委託元や企業が主体的に選定をしているわけではなく、プロジェクトに依存していることが大半を占めている。企業として選定している企業は 1 社しかなかった。これまでの開発は現場に任せて経営者が関与してこなかった面も強いと推察するが、経営者もある程度技術的な理解を深め、OSS の課題がやがて経営にも影響を生じる可能性があることを、まず理解する必要がある。

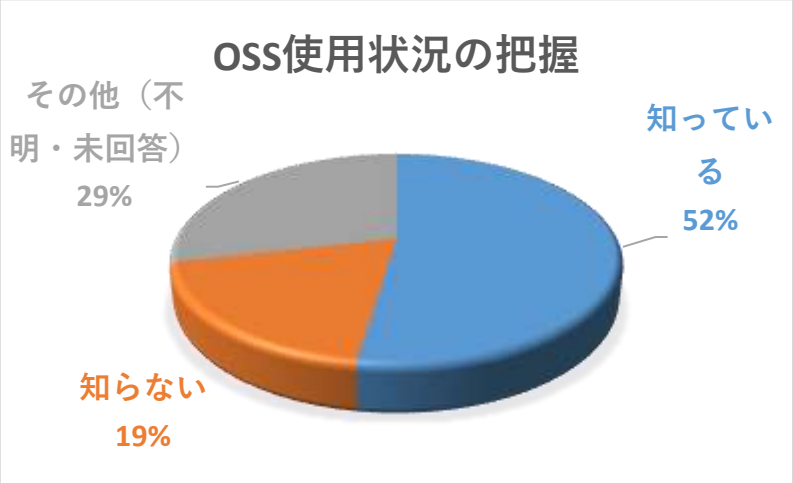

図 2-7：OSS 使用状況の把握

2-5.　OSS 等のソフトウェアの脆弱性に対応するための体制

2.5.1　ソフトウェア脆弱性に関する契約と管理

OSS をはじめソフトウェアには必ず脆弱性が生じる。その際に企業はどのように対応できるのか、また対応を行うのかを契約面でどれくらい考慮しているのかをヒヤリングした（図 2-8 参照）。契約にはない、不明という割合を合わせると約 8 割の企業において脆弱性の対処に係る契約に課題があることが分かる。

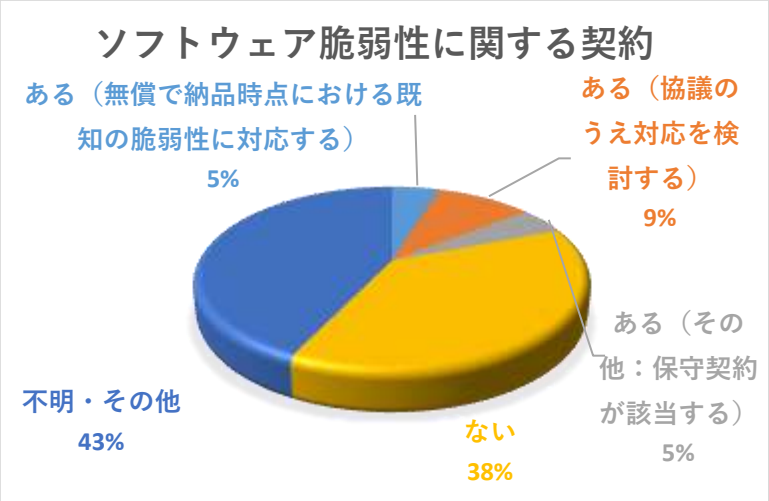

図 2-8：脆弱性に関する契約上の取り決め状況

また、ソフトウェアの開発・提供、受託などを行っているのであれば、コンポーネント管理を行い、SBOM などの管理表を作っておくことが望ましい。しかし、クラウドベースだから開発環境は常に最新であるといった回答もあり、多くの企業では作成していないことが分かった（図 2-9 参照）。委託元との契約があるから必要であれば作成して提出するといった回答も見受けられたように、実際に全く管理表がないわけではないが、きちんと文書化できていないのが現状である。文書化をしておかないと、プロジェクトメンバーが抜ければ過去のプロジェクトを追いきれなくなるだけでなく、提供しているソフトウェアのサポートができなくなってしまうリスクもある。無論、コード解析すればよいが、その金額的・人的コストは無駄にかかることになる。

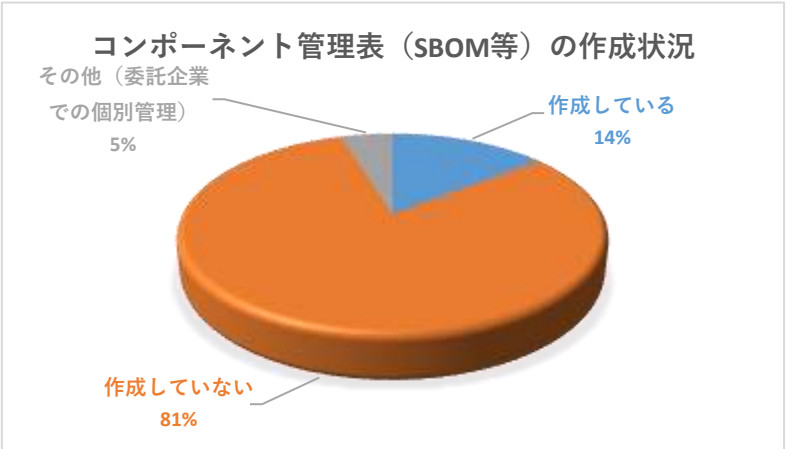

図 2-9：ソフトウェア管理の現状

また、作成していない理由は様々だが「知見がない」「必要性は感じているができていない」「都度調査することでカバーできると判断している」といったような回答が得られた。実施したいがノウハウやリソースがないからできない、必要な時には対応するなど対応できない理由は各社によって異なるが、組織としての管理ができていない現状も明るみになった。

さらに、納品方法に関する調査も行ったところ、各社の回答から割合を算出すると**ソースコード納品約 3 割、バイナリ納品約 7 割**といった現状が明るみになった。回答内容は百社百様とはいえ、契約によってソースコード納品となっている企業はバイナリ納品を行わず、特段の取り決めがない場合は、バイナリ納品のみを行っている企業もあった。これは、委託先としては、自社で開発したものであればその権利を守りたいという視点や、継続して仕事を得るために当該納品を行っている企業もある。本回答は日本のソフトウェア産業にとって大きな課題であり、会社の規模に関係なく、委託元、委託先それぞれが理解を深める必要がある。委託元は開発に関する取り決めを前もって厳密に行うべきであり、その際に、使用している OSS によって費用変更を行うようなことはあってはならないし、権利関係もどのように保持するのかは開発前に適切に議論を行う必要がある。

使用するソフトウェアによっては汎用的に使用されているものもあれば、希少性の高いソフトウェアもあるかもしれず、どのようなソフトウェアを利用するかによっても対応できる技術者が少ない場合や多い場合があるので、本来であればどのようなソフトウェアを使用するかによって、完成物をどのように保守していくのかを議論し、保守費用についても一律で考えるのではなく、ソフトウェアの違いによっても価格や体制などを考える必要がある。

2.5.2 ソフトウェア脆弱性検証に関する体制

開発したソフトウェアに関する検証を行っていないと回答した企業はなかったが、約 4 分の 1 で「案件によって異なる」といった回答があった。検証を行っている企業においても、動作検証していてもセキュリティ検証までは行っていないとの回答もあった。委託先としてのソフトウェア検証の仕方や深さも差が生じている。これらの背景には、委託元がソフトウェアの検証まで考えずに契約や費用を決定してしまっている現状もあると考えられる（図 2-10 参照）。

図 2-10：ソフトウェア検証の現状

また、検証を行っている企業の多くは概ね検証環境を有しているが、そもそも整備ができていなかったり、案件毎や都度検証環境を構築したりしている企業も一定割合あった。（図 2-11 参照）。これは自社開発や販売を行っている組織や、毎年必ず受ける仕事が決まっていれば常設もできるが、不定期な案件や検証環境を維持するコストなどの観点から、必要性は理解しているが対応できていない現実があると言える。

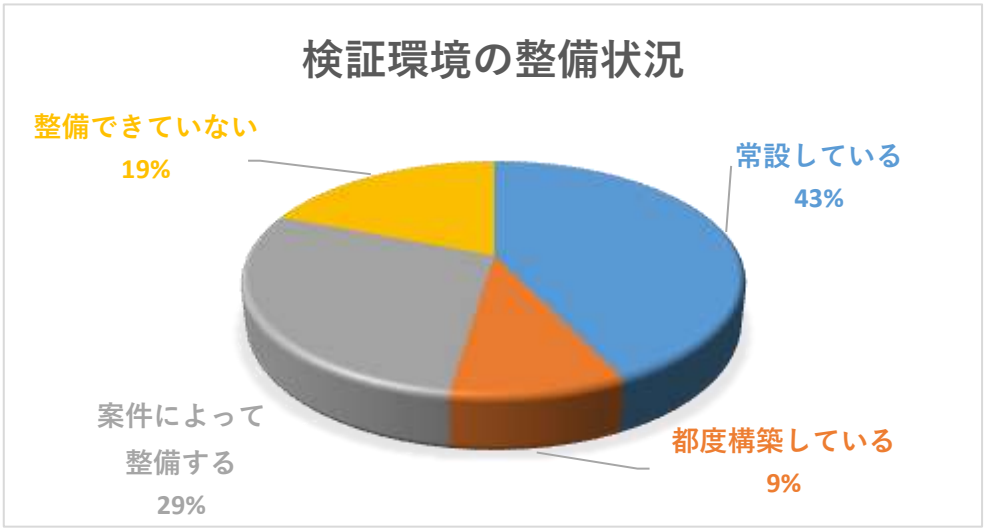

図 2-11：検証環境の整備状況

2.6　使用しているソフトウェアの調査

各社で開発に使用しているソフトウェアの調査を行った。以降、ソフトウェア毎にヒヤリング結果をまとめる。

2.6.1　使用しているソフトウェア（OS）

OSS に限らず Windows を含めて調査を行った（図 2-12 参照）。各社が使用しているソフトウェアでは Windows ベースが多く、ついで Android OS や iOS、RedHat/Cent OS といった結果となっている。また、今回のヒヤリングでは Debian と Fedora も選択肢に含めたが、使用している組織は存在しなかった。

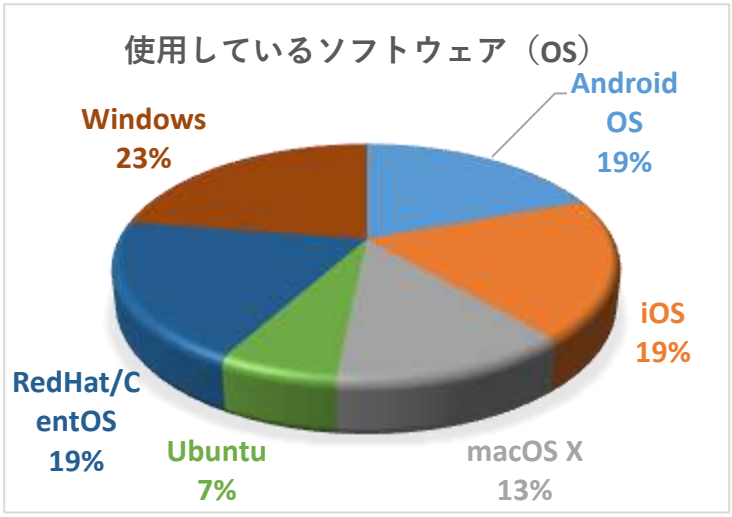

図 2-12：使用しているソフトウェア（OS 編）

2.6.2　使用しているソフトウェア（プロトコル）

Web やメールが多く利用されていることに対応するために、プロトコルは当然ながら実装されている。プロトコル別（図 2-13 参照）にみると HTTP、HTTPS、次いで SMTP が割合として高くなっている。しかし、HTTP と HTTPS の割合は変わらないため、開発時に委託元、委託先において、セキュリティについてより一層の議論と配慮が必要である。

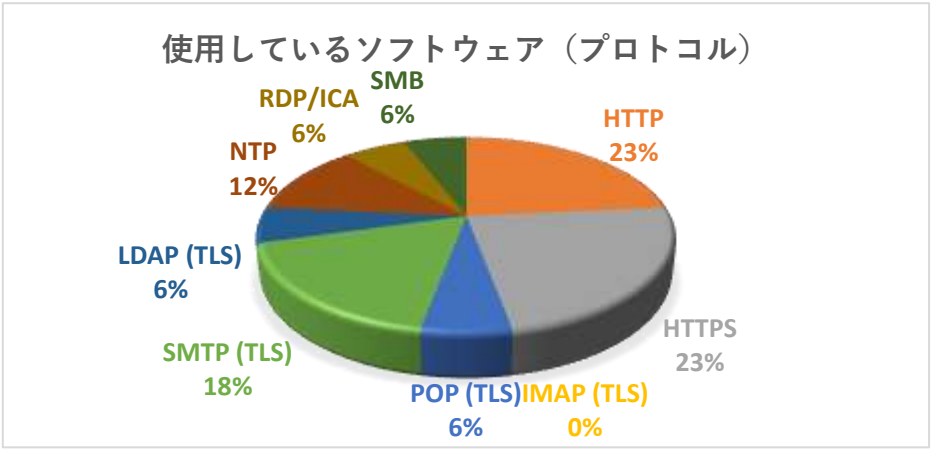

図 2-13：使用しているソフトウェア（プロトコル編）

2.6.3　使用しているソフトウェア（WEB/AP サーバ、POP/IMAP/MTA）

Web/AP サーバ（図 2-14 参照）においては Apache が多く、Apache HTTP Server、Apache Tomcat の順となっており、また、POP/IMAP/MTA[1]（図 2-15 参照）は Postfix と sendmail が、ともに多く使われている。

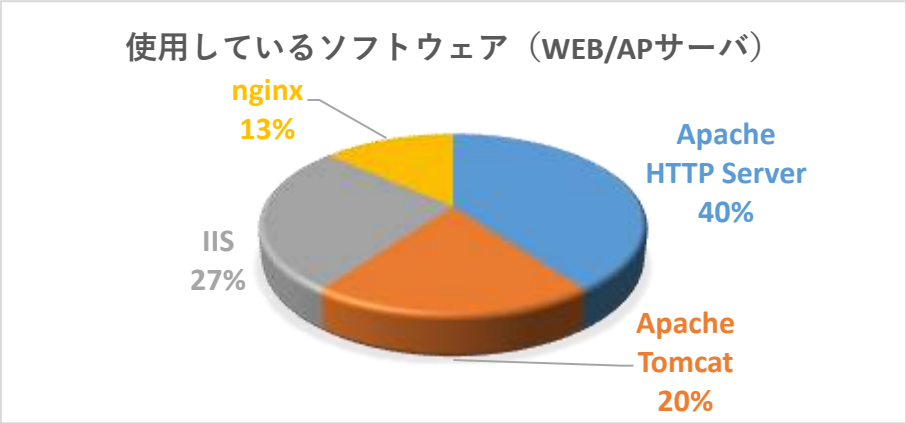

図 2-14：使用しているソフトウェア（Web/SP サーバ編）

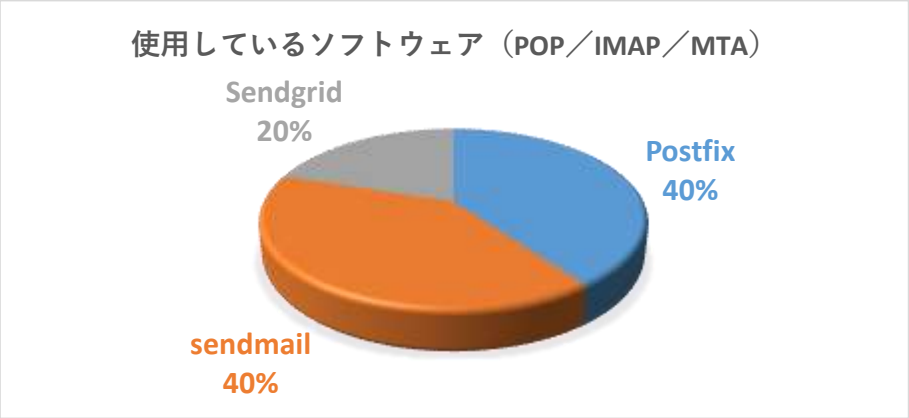

図 2-15：使用しているソフトウェア（POP/IMAP/MTA 編）

[1] MTA とは、Mail Transfer Agent の略で、メールの転送や配送を行うためのソフトウェアのこと。POP は Post Office Protocol の略で IMAP（Internet Message Access Protocol）と同様、メールを受信するための仕組みのこと。POP はサーバにあるメールをダウンロードして、端末でメールを管理する仕組みで、IMAP はサーバにあるメールをダウンロードせずに、サーバ上でメールを管理する仕組み。

2.6.4　使用しているソフトウェア（DB）

DB（図 2-16 参照）では、MySQL に次いで SQL Server の利用割合が高い結果となっている。

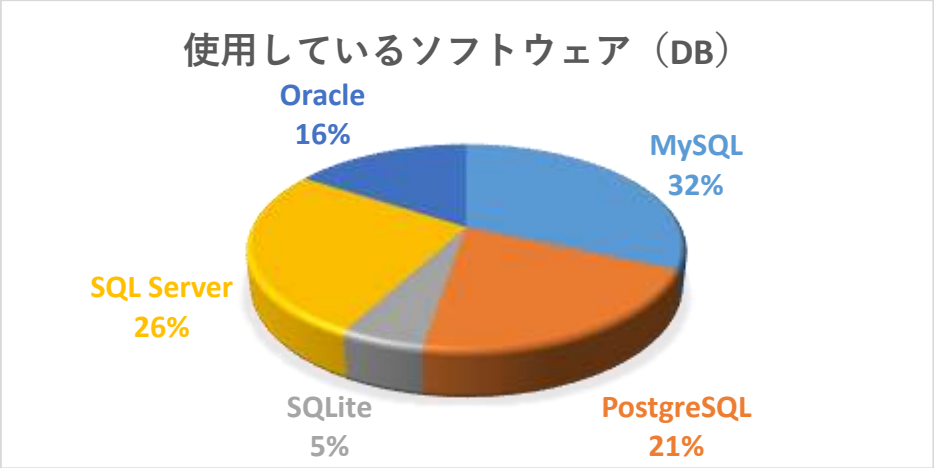

図 2-16：使用しているソフトウェア（DB 編）

2.6.5　使用しているソフトウェア（開発言語）

開発言語（図 2-17 参照）としては様々な言語が利用されており、分散傾向にあるが、PHP、.Net、Java、JavaScript が高い割合となっている。

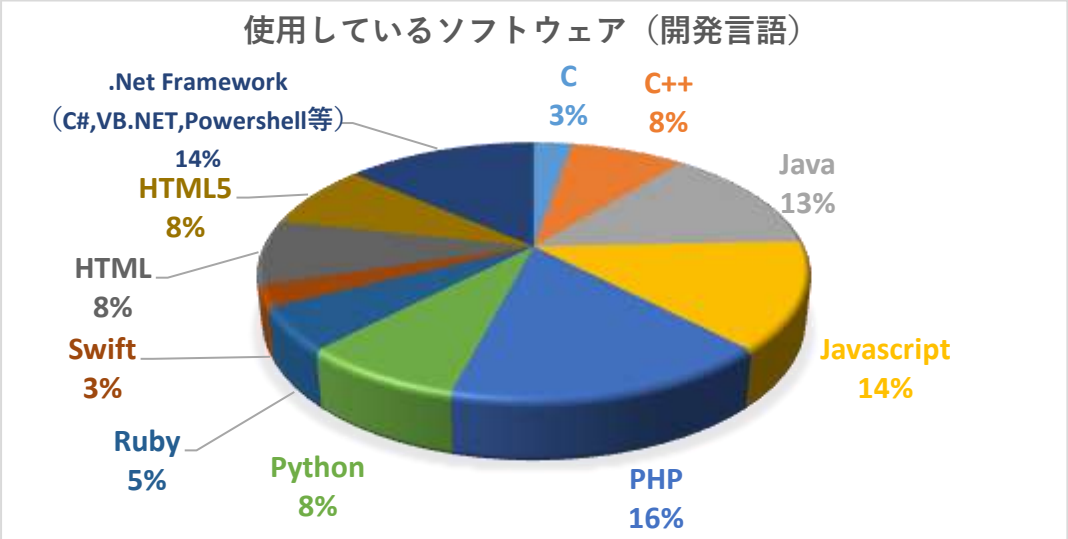

図 2-17：使用しているソフトウェア（開発言語編）

2.7　脆弱性情報の収集有無

約 7 割の組織で脆弱性情報の収集を行っていることが分かった（図 2-18 参照）が、組織的に対応できている組織は少なく、ヒヤリングの中では属人的な収集によって対応している組織もあった。1 組織あたりの収集時間は平均すると 20 分／日程度の時間を使用していることも分かった。

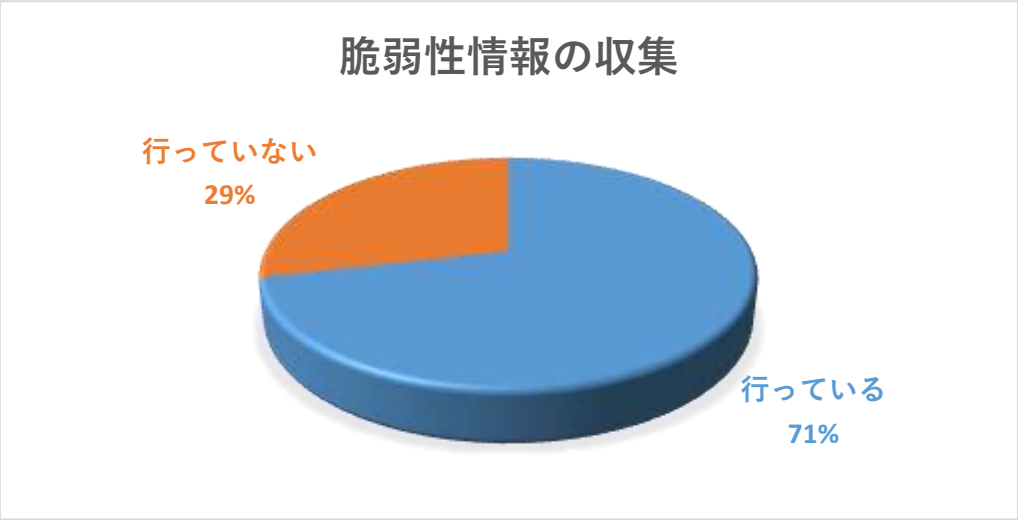

図 2-18：脆弱性情報の収集状況

2.8　PSIRT について

製品の開発（受託開発含む）を行っている組織だからこそかもしれないが、PSIRT を知っている割合は比較的高く、7 割近い組織が認知をしていた（図 2-19 参照）。しかし、PSIRT を知っているからといって企業で設置が進んでいるわけではなく、PSIRT を構築している企業は 2 割にも満たなかった（図 2-20 参照）。さらに、将来的に構築をしたいとしている企業も 2 割強ほどしかなく、認知はしているが構築をできていない現状が明るみになった。これはセキュリティを組織として考えることができていないと言える一方で、セキュリティよりも事業を優先し、日々の開発業務に集中せざるを得ない現状もあるといえるであろう（図 2-21 参照）。

図 2-19：PSIRT の認知度

それではPSIRTをなぜ構築、検討すらしていないのか。それはノウハウがないためという意見もあったが、基本的にはリソースの不足であり、特に、「ヒト」がいないために実施できていないといった回答が多く見られた。現状はセキュリティにコストを掛けるだけの余裕はなく、組織的な対応が取れていないことが、本調査によって明るみになったと言える。

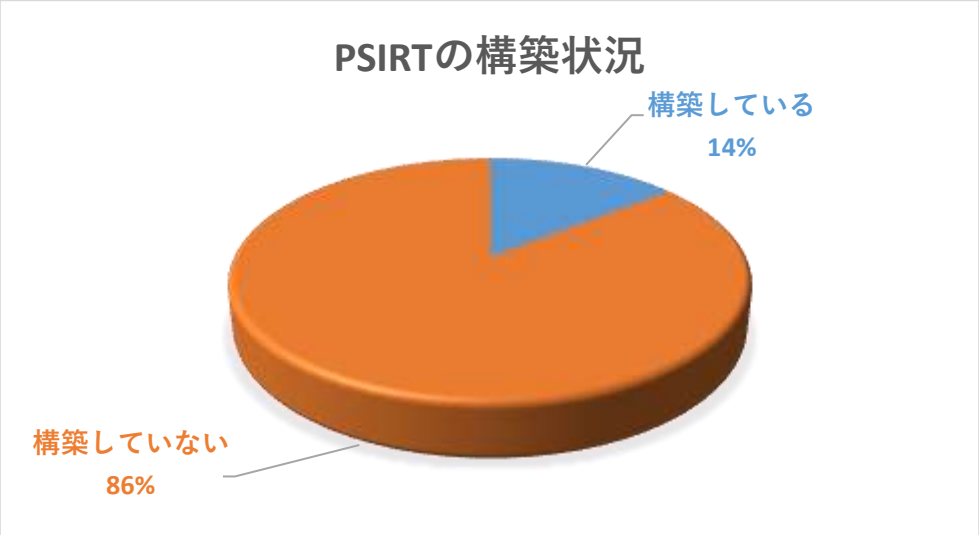

図 2-20：PSIRT の構築状況

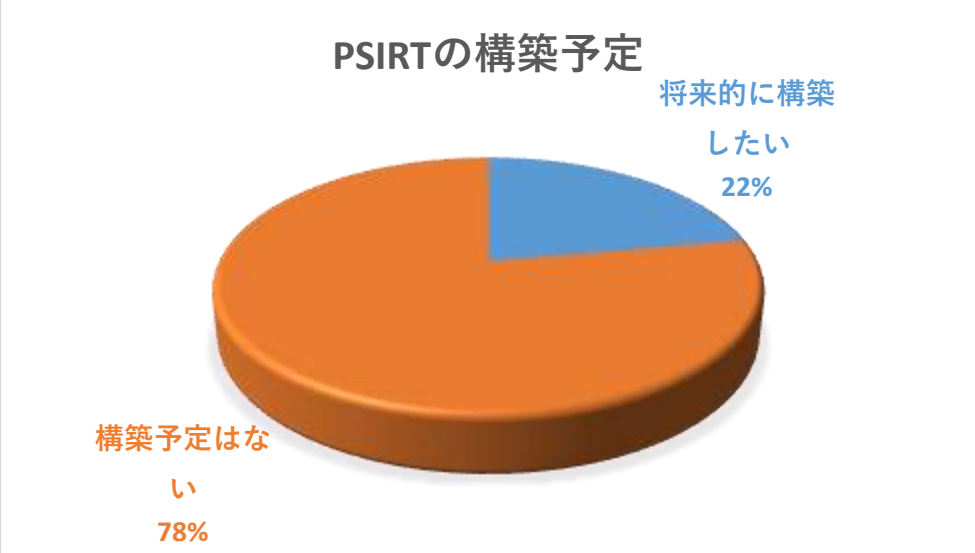

図 2-21：PSIRT の構築予定

2.9　脆弱性管理データベース及び脆弱性ポータルへの期待

自組織での脆弱性情報の管理や収集、脆弱性対応などには限界があり、属人的な要素も強い。また、中小企業においては、そもそも脆弱性に適切に対応できる組織や人もいない状況なので、組織を超えた連携に対する期待は大きい。当該ヒヤリングで構想している脆弱性情報管理のための OSS データベースに求めることを問うたところ、以下のような回答があった。

- 溢れる情報の集約
- 公開前の脆弱性情報や脅威情報の共有
- 即時性の高い情報連携
- 希望する OSS の継続監視　など

ポータルやデータベースの利用について約 6 割の企業で、脆弱性情報管理の OSS データベースサービスを利用したいと回答があった一方で、未回答や不明といった回答も次いで多かった。そもそもの重要性に気づいていない組織については適切に説明していく必要があるとともに、既存のサービスの集約だけであれば不要といった意見も存在し、要望を適切に吸い上げ、反映、説明する必要がある。

2.10　ヒヤリング調査のまとめ

我が国における OSS の使用状況と脆弱性管理の実態について、実際のソフトウェア開発を行っている企業にヒヤリングを行い、実情を定量的に明らかにすることができた。その主要なポイントは次の通りである。

① 約 7 割の組織が開発にあたっては外部委託を行っており、OSS も 8 割近い組織が製品や受託開発において使用している
② バイナリ納品の割合は総じて高く、SBOM に代表されるようなコンポーネント管理が適切に行えていない現状がある。
③ PSIRT の認識は高まっているが、PSIRT を構築している組織は少なく、特に「ヒト」の課題によって構築が行えていない。

このように、我が国のソフトウェア開発においては、脆弱性管理が十分ではなく、個々の企業の努力のみに任せていては、ソフトウェアサプライチェーンの問題を根強く残し、今後 OSS によって日本の産業や IoT 機器を使用している家庭に影響を及ぼす可能性もある。

特に、中小企業においては、日本の多重下請け構造による受託開発によって支えられている面もあり、また脆弱性管理を始めとした PSIRT のような組織や能力を持つことは極めて

難しく、日本全体で横断的にソフトウェア開発を行う特に中小企業を支えるような体制検討が必要である。

このため、Software ISAC において新たに脆弱性管理データベース及び脆弱性管理ポータルを作成し、ソフトウェア開発企業の脆弱性管理を支援する必要が明らかになった。