

# 変化に追隨するデータ連携へ： Software-Defined Society (SDS) 入門

独立行政法人情報処理推進機構

デジタル基盤センター デジタルエンジニアリング部

ソフトウェアエンジニアリングG

渋谷 健介

# 自己紹介

名前：渋谷 健介（しぶや けんすけ）

所属・役職：

IPA デジタル基盤センター デジタルエンジニアリング部  
ソフトウェアエンジニアリンググループ グループリーダー



略歴：

- ◆ 2008年4月 日本電気株式会社入社
  - ソフトウェア検証技術を中心に、社内のソフトウェアエンジニアリングの推進に従事
- ◆ 2024年1月よりIPA出向、現在に至る
  - Software-Defined Societyの到来に向けて、ソフトウェアを効率的かつ高品質に整備するための手法の確立と普及に従事

# レーンA ラインナップ つなげる基礎力 ～データ連携の技術エッセンス～

## 1. 「つなぐ」ための基礎力（信頼できる／意味が通じる）

### ①データ品質

なぜデータ品質が“つながり”を左右するのか

### ②データ辞書

セマンティックレイヤーをつなぐ（データ辞書）

## 2. つないだ後に「価値を回し続ける」ための基礎力（変化に追随する）

### ③Software-Defined

変化に追随するデータ連携へ：Software-Defined Society (SDS) 入門

本セッション

## ◆ お題

- データをつないだ後に、その価値を「**継続的に引き出し続ける**」ために必要なことは何か？

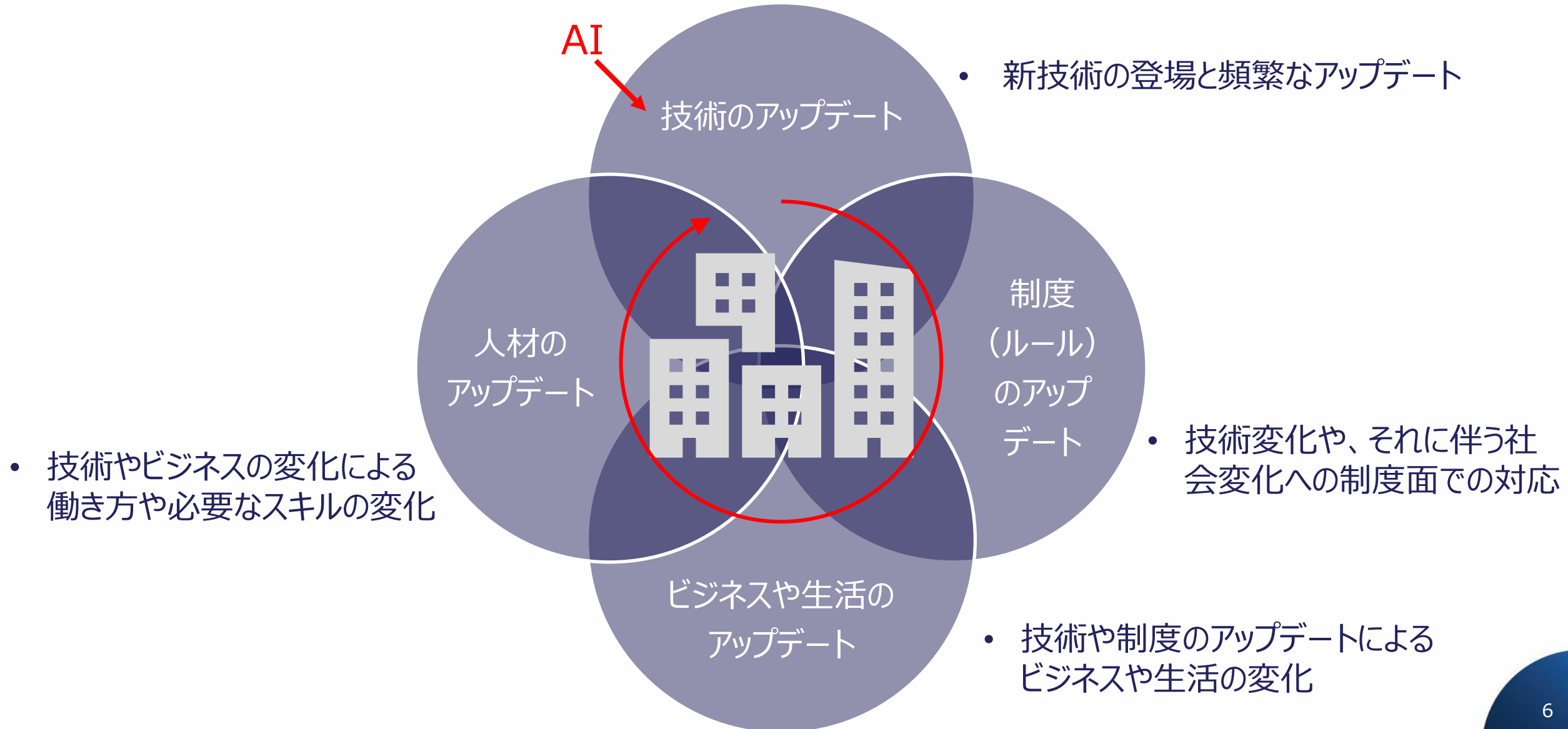
## ◆ ポイント

- ① 継続的価値創出の前提条件：**デジタル基盤の重要性**
- ② 変化への対応力：**アジリティの3つの柱**
- ③ アジリティの高い社会に向けて：**Software-Defined Society (SDS) のコンセプト**

## 継続的価値創出の前提条件

- デジタル基盤の重要性

# 背景：アップデートし続ける社会の到来

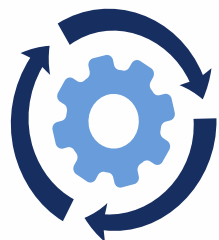


- ◆ データ連携後のよくある課題
  - 連携はできたけれど、制度変更や業務変更が来た瞬間に**改修が重くなる**
  - データは集まるけれど、意思決定や改善につながらず、**結局「見るだけ」になる**
  - 最初は成果が出ても、運用が回らず、**いつの間にか使われなくなる**
  - など
- ◆ また、社会変化が高速であり、データ連携したまま放置すると、**データ連携の価値がすぐに陳腐化してしまう**

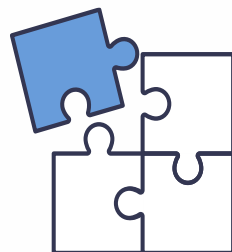
「つなぐ」はスタートで、次に出てくる壁は、  
「**変化に合わせて、使い方や仕組みを更新し続けられるかどうか**」

- ◆ 高速な社会変化に追随し、円滑な更新サイクルを回すための鍵が「ソフトウェア」の活用
    - データ：更新のための**燃料**
    - ソフトウェア：燃料を価値に変える**エンジン**
    - 燃料が良くても、エンジンが整備されていないと性能が出にくい（逆も同様）
- ⇒ データとソフトウェアの両輪で整備していく必要がある（デジタル基盤）

## ソフトウェアの3つの特長



使っても減らない



柔軟に改変可能



継続的な改善

# AI時代におけるデジタル基盤の重要性

- ◆ 最近ではAIの「見える成果」に注目が集まっているが、冰山と同じで、水面下にあるデジタル基盤（データを整え、変化に追従できる仕組み）が揃うほど、AIの精度や効率は伸びやすくなる



## AIの効果は、デジタル基盤で増幅される

- 同じ土俵で学習・推論できる
- 継続改善がしやすい

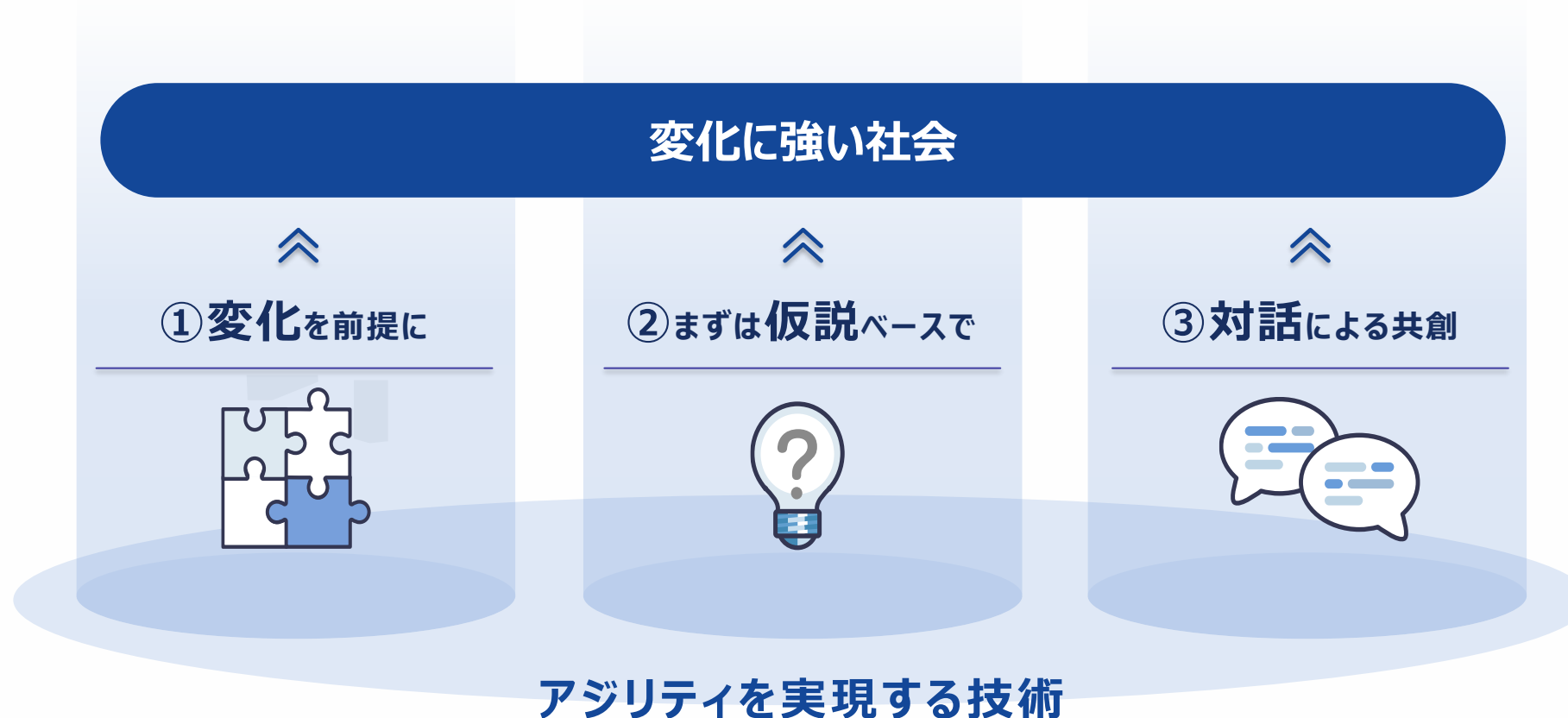
観点	デジタル基盤なし	デジタル基盤あり
精度	ばらつく	安定しやすい
効率	手戻りが多い	前処理が減る
スケール	横展開しにくい	横展開しやすい
ガバナンス	説明・監査が重い	説明・監査が通しやすい

## 変化への対応力

- アジリティの3つの柱

- ◆ アジリティとは：端的には「状況に合わせて柔軟に素早く対応する力」のこと

## アジリティを実現する考え方



# アジリティの柱① 変化を前提に

- ◆ **ポイント：将来的な変化を前提に考える**
  - 「一度つないで終わり」ではなく、**制度・業務・組織変更が来ても壊れにくい連携**にする
- ◆ そのキーとなる考え方が「**Software-Defined**」
  - 仕組みをソフトウェアで定義して、状況に応じて更新できるようにする考え方

## Software-Definedの例

### 従来の自動車



### ハードウェア主導の設計

購入時点で性能・機能が決まり、  
機能追加には部品交換やディーラーでの修理が必要。



### 最新の自動車 (SDV※)

※ Software-Defined Vehicleの略。



### ソフトウェア主導の設計

ソフトウェアをアップデートすることで、  
リモートで機能追加や性能向上が可能。

# アジリティの柱② まずは仮説ベースで

## ◆ ポイント：仮説検証と改善のループを回す

- 「全部つないでから価値を出す」ではなく、**価値仮説を置いて、まずは最小限の連携で検証する**ところから始めてみる



### 例えば...

- **審査・処理時間短縮仮説：**  
「1工程だけ」自動参照にして、処理時間を計測する
- **不正・ミス検知仮説：**  
連携したデータで簡単なルール検知から始め、精度を改善する

# アジリティの柱③ 対話による共創

## ◆ ポイント：多様な関係者との対話とフィードバックループ

- 様々な価値観を持つ人々との対話が、**自分たちだけでは生み出せない価値を創出**
- 連携は技術だけで完結せず、**意味・責任・例外対応**は関係者の合意が必要  
⇒ 単に対話を増やすのではなく、対話が回る「型」を作る（仕組み化）

他業種  
(ベンダーなど)



他業種との  
共創活動

提供者

(企業・公官庁、など)



利用者からの  
フィードバック

利用者



例えば...

- **データ定義レビュー会**：  
「顧客」「対象者」「契約」など曖昧語を  
モデルで合わせる
- **品質・利用状況の見える化ダッシュボード**：  
欠損率、遅延、利用回数、問い合わせ発生  
などを共有
- **変更の受付と優先順位付けのルール**：  
仕様変更の入り口、影響範囲、リリース  
サイクルを決める

# アジリティを実現する技術

- ◆ これら3つの考え方は、アジリティを実現する技術がベースにある
- ◆ これらの技術を知っておくことが、アジリティをより一層高めることにつながる

## アジャイル

変化に合わせてこまめに改善

小さく試して学び、こまめに作り直していく進め方です。「一度決めて終わり」ではなく、変化に合わせて更新し続けるという考え方は、アジリティの土台です。

## DevOps

作る人と使う人の団結

作る人と使う人が一体となって改善していくという考え方です。運用しながら機能変更を素早く安全に届けるといふ、対話が重要な仕組みです。

## モデリング

仕組みの見える化

複雑なことを「要素」と「関係」に整理・可視化する方法です。目に見えることで設計のブレを減らし、共通理解を得やすくします。

## デジタルツイン

デジタル空間での仮説検証

現実の状態をデジタル上で再現し、状況を把握しやすくする考え方です。デジタル上で仮説を検証できるので、改善や最適化を安心して進められます。

## ビルディングブロック

部品の交換・組合せ

よく使う機能やルールを「部品」としてそろえ、組み合わせてモノを作る考え方です。変更があってもゼロから作り直さずに済み、修正や機能拡張を素早く安全に実施できます。

## AI活用

人間の作業・判断のサポート

様々な工程やタスクで、人の作業や判断を補助する技術です。人間の負担を減らして改善のスピードを上げることで、変化への対応力が高まります。

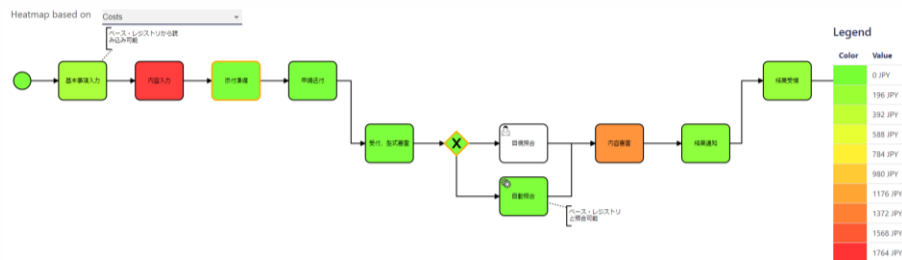
## データ利活用

データに基づく判断

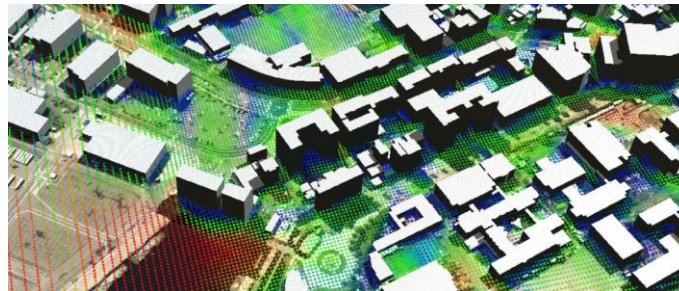
データを集めて整理し、状況を理解するための根拠とする取り組みです。データに基づいて判断することで、意思決定と改善を適切に実施できます。

# (参考) モデリング & シミュレーション

- ◆ 基本は可視化である。可視化されているから、
  - 共通の理解ができ、課題が明確化されるとともに、効率的に議論ができる
  - さらに、解決策を作り、検証することができる
- ◆ AIに指示を的確に出すにしても、要件定義の考え方が重要になる



業務プロセスモデルの可視化



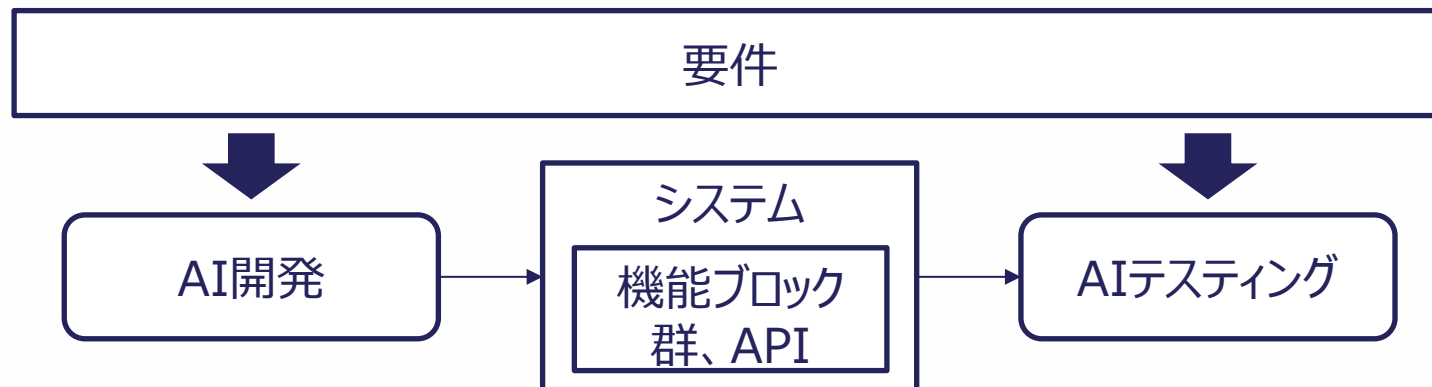
都市モデルの可視化 (PLATEAU)

## モデリングで活動を可視化

- 可視化することで事前検証が容易になる
- トラブル発生時に原因究明が容易になる
- 事前検証や予見が可能になる

# (参考) 要件定義がベースになる

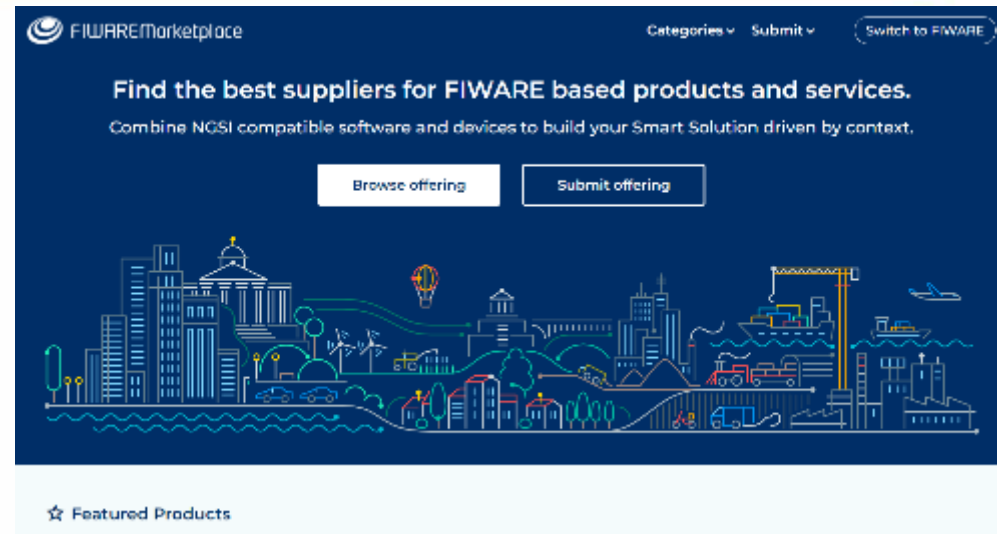
- ◆ AI時代においても、要件が定義されるから、プロンプトも明確に書け、AI開発やAIテストができる
  - もちろん、要件定義自体をAIでサポートすることも可能
- ◆ 要件が定義できたからと、なんでも作ればよいというものではない
  - 再利用できる部品を活用することが必要



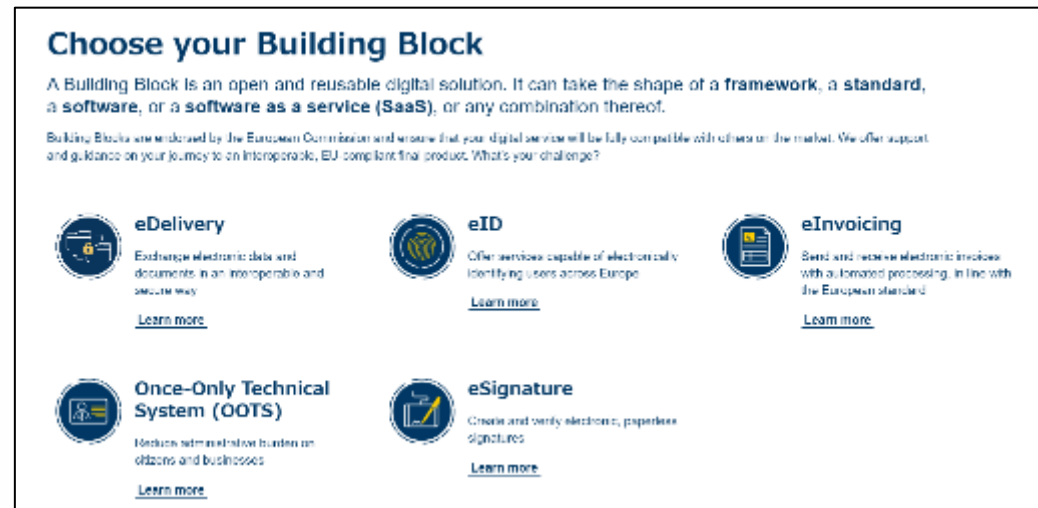
※AI開発以外に、ノーコード、サービス購入など様々な選択肢がある

# (参考) ビルディングブロック

- ◆ ソフトウェアやサービスを作りやすいように各種部品（OSS等）の供給が行われている
- ◆ 部品を組み合わせることで、迅速にシステムの開発ができる  
（ビルディングブロック型開発）
  - 部品を入れ替えることで変化への追従もしやすい
- ◆ ビルディングブロック型開発においては、どのようなソフトウェアを使っているか、SBOMが重要になる



出典：[FIWARE – Marketplace](#) (FIWARE)



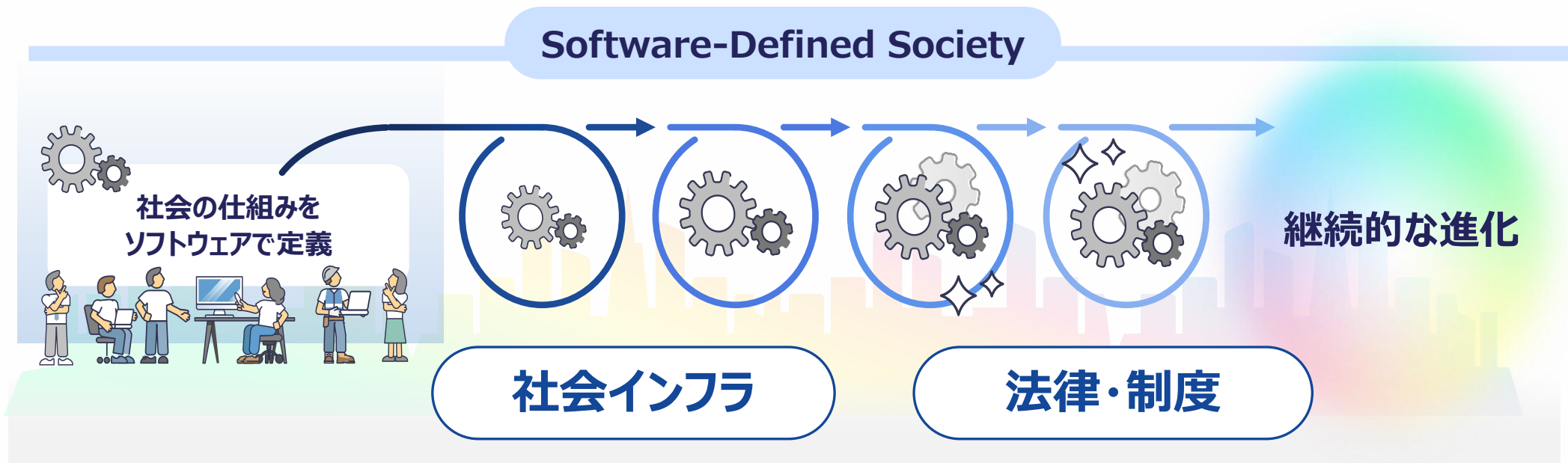
出典：[Digital Homepage](#) (Europe Commission)

## アジリティの高い社会に向けて

- Software-Defined Society (SDS) のコンセプト

## ◆ Software-Defined Society (SDS)

- Software-Definedの考え方を社会全体に広げた、アジリティの高い社会
- 社会・産業の主要な機能がソフトウェアで実装され、社会状況の変化に合わせて継続的に更新



# SDSの例① 社会インフラ

- ◆ 道路・橋梁・上下水道・電力などの社会インフラは、老朽化や気象条件、利用状況の変化などにより、状態が常に変動
- ◆ SDSでは、**センサー等で状態を観測し、データから兆候を捉え、点検・補修の計画や運用を継続的に更新**（観測→判断→対応→再観測のフィードバックループ）

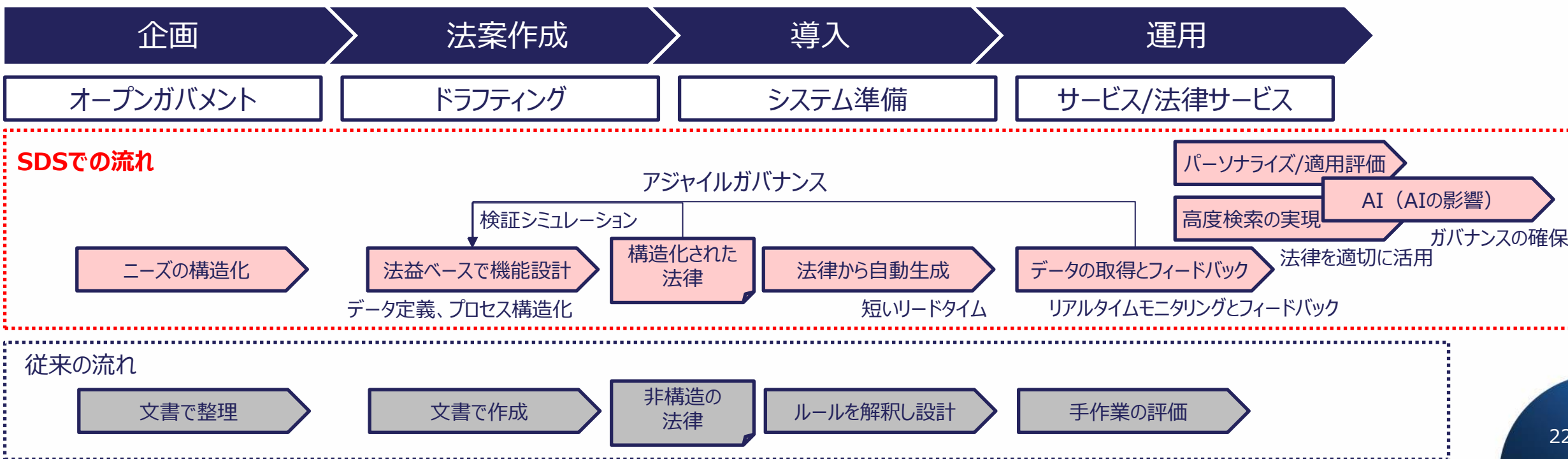


例えば...

- **遠隔監視（IoTセンサー）**：  
橋梁・トンネルにセンサーを取り付けてデータを収集。劣化をリアルタイムで検知することで迅速な異常検知や予兆保全を可能にする
- **AI予兆保全**：  
収集したデータをAIで解析し、故障を事前に予測。補修コストの削減やインフラの長寿命化を実現

# SDSの例② 法律・制度

- ◆ AIなどの技術変化に追随するには、従来の文章をベースとした法令管理では対応が困難
- ◆ SDSでは、**法律・制度がソフトウェアに適した形で制定**されることで、**法改正による事前シミュレーション**や**個人最適化**が可能になるだけでなく、**継続的に制度を改善**できる
  - そのような技術を**LegalTech（リーガルテック）**と呼び、近年注目を集めている



- ◆ SDSは「**参加型**」の社会設計
  - 変化を前提に、仮説検証とフィードバックを回して更新する
  - その担い手は「誰か」ではなく、「**私たち一人ひとり**」

## ◆ あなたの次の一歩（例）

### 次の一歩①：

#### 意味をそろえる（共通理解）

- 例：用語・定義・粒度を1枚にまとめる（モデル/データ辞書の最小版）
- 例：データの「責任者」と更新ルールを決める

### 次の一歩②：

#### 小さくつないで試す（仮説検証）

- 例：価値仮説を1つ置き、必要な3項目だけ連携して効果を測る
- 例：利用状況・品質を可視化して改善サイクルを作る

### 次の一歩③：

#### 越境の準備をする（協調）

- 例：外部連携を見据えた共有条件・権限・来歴を整理する
- 例：標準・ルール・事例を参照し、使えるところから取り入れる

参考) 社会全体での「協調の器」：データスペース（参考：[データスペースの推進 | IPA](#)）

- 信頼（ルール・権限・監査）を担保しながら、組織・業界を越えてデータを共有・活用する枠組み
- 相互運用性／公平性／データ主権などの考え方が重要

## まとめ

- データをつないだ後に、その価値を「**継続的に引き出し続ける**」ために必要なこと

- ① 継続的価値創出の前提条件：**デジタル基盤の重要性**
  - ① データとソフトウェアの両輪で整備していく必要がある
  - ② AIの効果は、デジタル基盤で増幅される
  
- ② 変化への対応力：**アジリティの3つの柱**
  - ① 変化を前提に
  - ② まずは仮説ベースで
  - ③ 対話による共創
  
- ③ アジリティの高い社会に向けて：**Software-Defined Society (SDS) のコンセプト**
  - ① SDSは「参加型」の社会設計。その担い手は「私たち一人ひとり」

## 付録

- 本セッションに関連するIPAの活動紹介

- ◆ IPAではデータ利活用やソフトウェアに関する情報を公開しています。詳細な情報は下記ページをご参照ください。

- ◆ データ利活用関連

- [データ利活用の推進 | 社会・産業のデジタル変革 | IPA 独立行政法人 情報処理推進機構](#)



- ◆ Software-Defined Society関連

- [システム／ソフトウェア開発の革新 | 社会・産業のデジタル変革 | IPA 独立行政法人 情報処理推進機構](#)



# Software-Defined Society 解説コンテンツ



アジリティの実現に必要な考え方や基盤となる技術、そして「Software-Defined Society」について、わかりやすく紹介するコンテンツを公開しています。

## ドキュメント

あなたが創る、豊かな未来  
- Software-Defined Societyの実現に向かって -

IPA 独立行政法人  
情報処理推進機構

Copyright © 2026 Information-technology Promotion Agency, Japan (IPA)

## 動画

動画プレイヤーのスクリーンショット。検索バーには「検索」とあり、IPAのロゴが右下に表示されています。再生ボタン、音量、0:02 / 8:27、一時停止、字幕、設定、フルスクリーンなどのコントロールボタンが並んでいます。チャンネル登録ボタンと高評価ボタンも表示されています。

## チラシ

あなたが創る、  
豊かな未来

Software-Defined Societyの実現に向かって

変化が激しく将来の予測が困難な現代社会、このような社会では、詳細に計画を立てて完璧を目指すというアプローチは通用しなくなっています。そんな時代を生かすために重要なのが、状況変化に合わせて行動を変える柔軟性。

すなわち、アジリティです。

アジリティを実現するための3つの考え方

<b>変化を前提に</b> 将来的な変化を前提に設計します。	<b>まずは仮説ベースで</b> 完璧を目指すのではなく、まずは仮説ベースで実験します。	<b>対話による共創</b> 多様な関係者との対話を通してモノの価値を最大化させます。
-----------------------------------	---	--

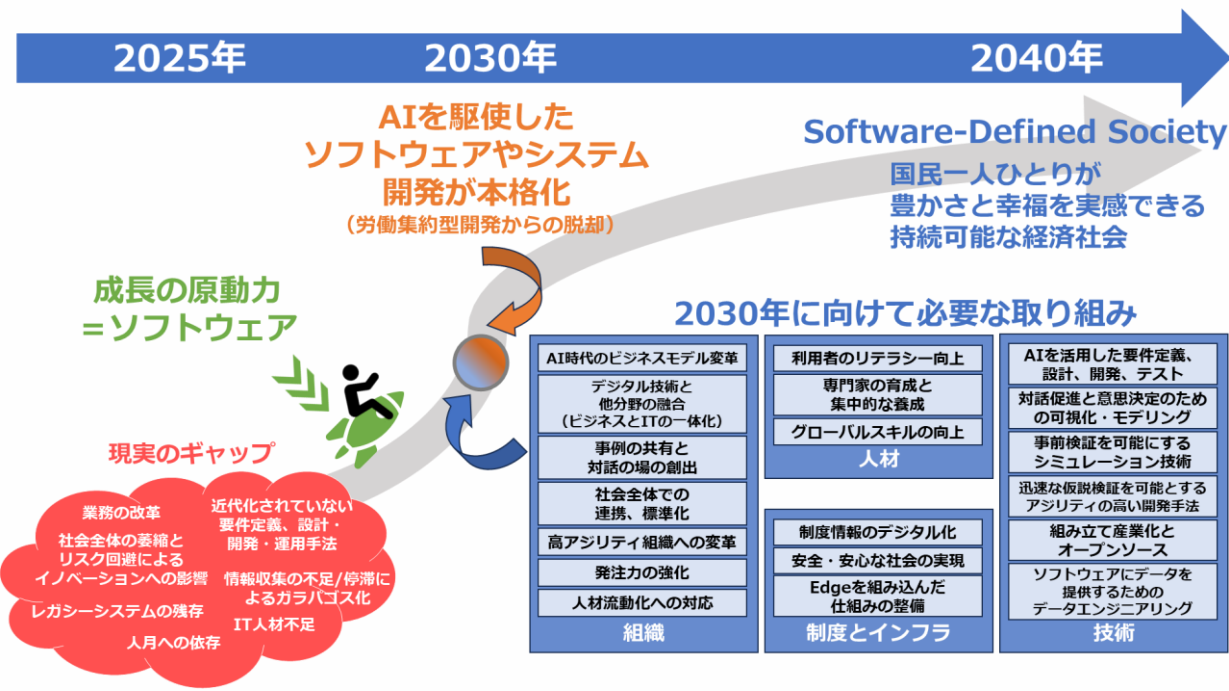


そして今、高いアジリティとソフトウェアによって生み出される  
Software-Defined Society  
という考え方が注目されています。

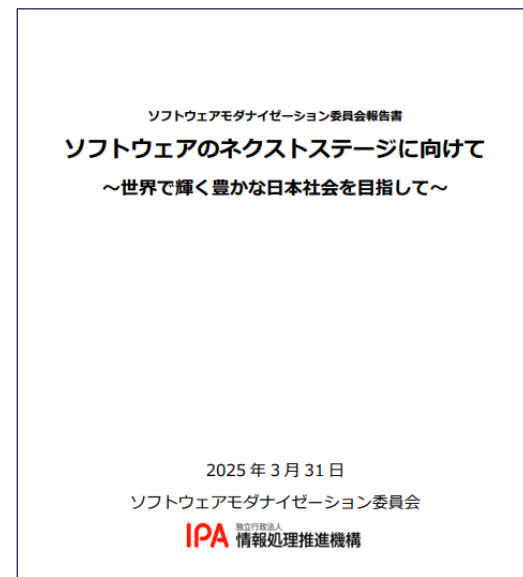
裏面へ  
続きます

# 「ソフトウェアモダナイゼーション委員会」報告書

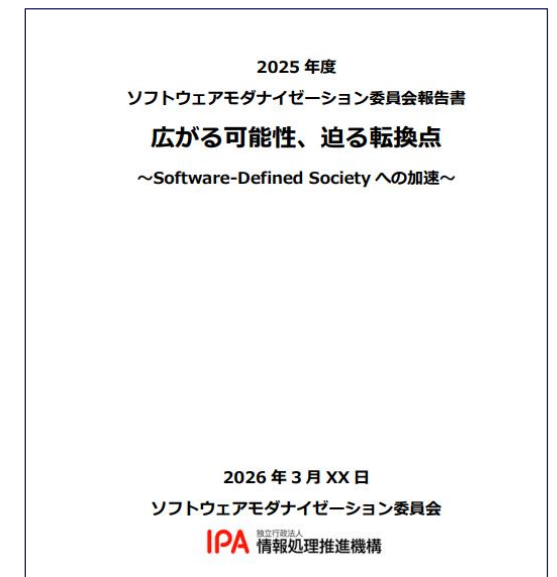
ソフトウェアがもたらす価値を最大化する観点から、社会や産業界が目指すべき姿と、それに向けて必要な取り組みについて検討を実施し、報告書として公開しています。



ロードマップ ～ソフトウェアのネクストステージへ～



【ソフトウェアモダナイゼーション委員会報告書】  
ソフトウェアのネクストステージに向けて  
(2025年3月公開)

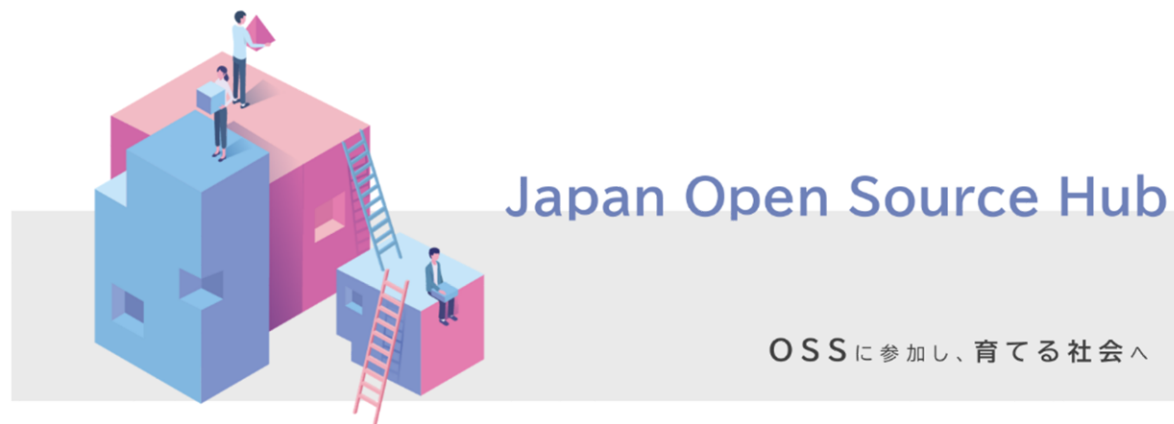


【ソフトウェアモダナイゼーション委員会報告書】  
広がる可能性、迫る転換点  
～Software-Defined Societyへの加速～  
(2026年3月公開)

# Japan Open Source Hub

Japan Open Source Hubは、オープンソースの価値をひらき、つながり、育てるための入り口です。

OSS（オープンソースソフトウェア）に触れるのが初めての方も、導入を進める方も、ここから次の一歩を見つけてください。



Japan Open Source Hubは、オープンソースの価値をひらき、つながり、育てるための入り口です。  
OSSに触れるのが初めての方も、導入を進める方も、ここから次の一歩を見つけてください。

Search for  
“IPA OSS”!

IPA