

ソフトウェア品質シンポジウム 2025 特別講演

社会を支える製品の設計・開発業務への生成AI活用

2025年9月26日

株式会社 東芝

代表取締役 社長執行役員 CEO

一般社団法人 量子技術による新産業創出協議会 (Q-STAR)

代表理事

島田 太郎

本日の講演内容

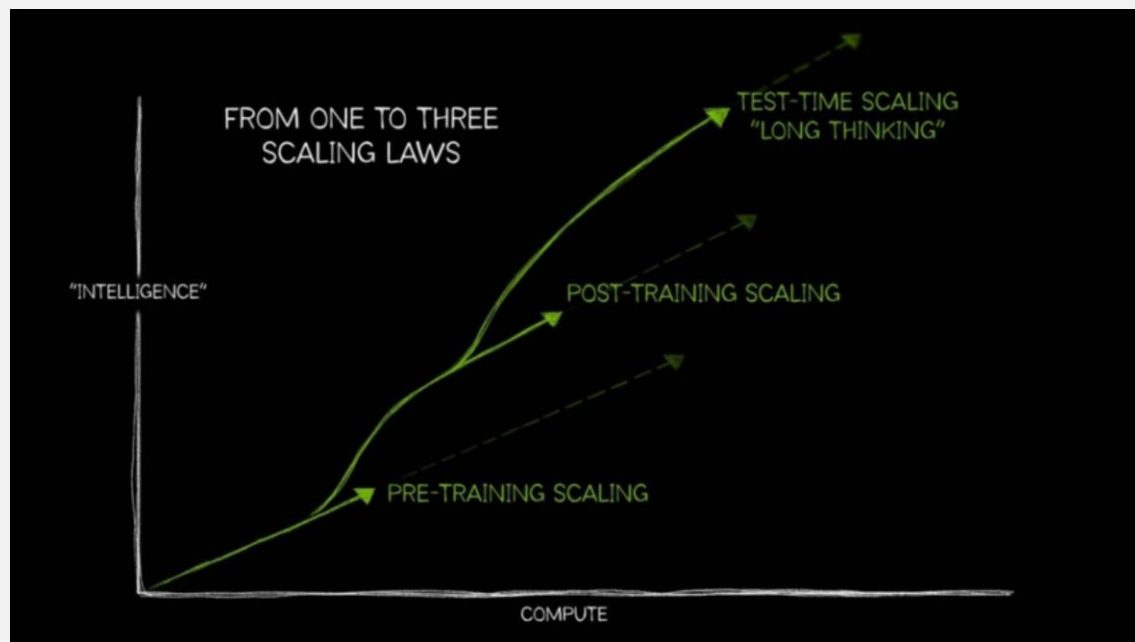
- 01 生成AI活用のトレンド
- 02 東芝の生成AI活用に向けた取り組み
- 03 生成AIによるソフトウェア品質への取り組み
- 04 クリティカルシステムにおける生成AI活用
- 05 生成AIを活用したソフトウェア開発の将来
- 06 結局重要なのはデータになる

01

生成AI活用のトレンド

進化の速度が過激かつ次々と新しいトピック

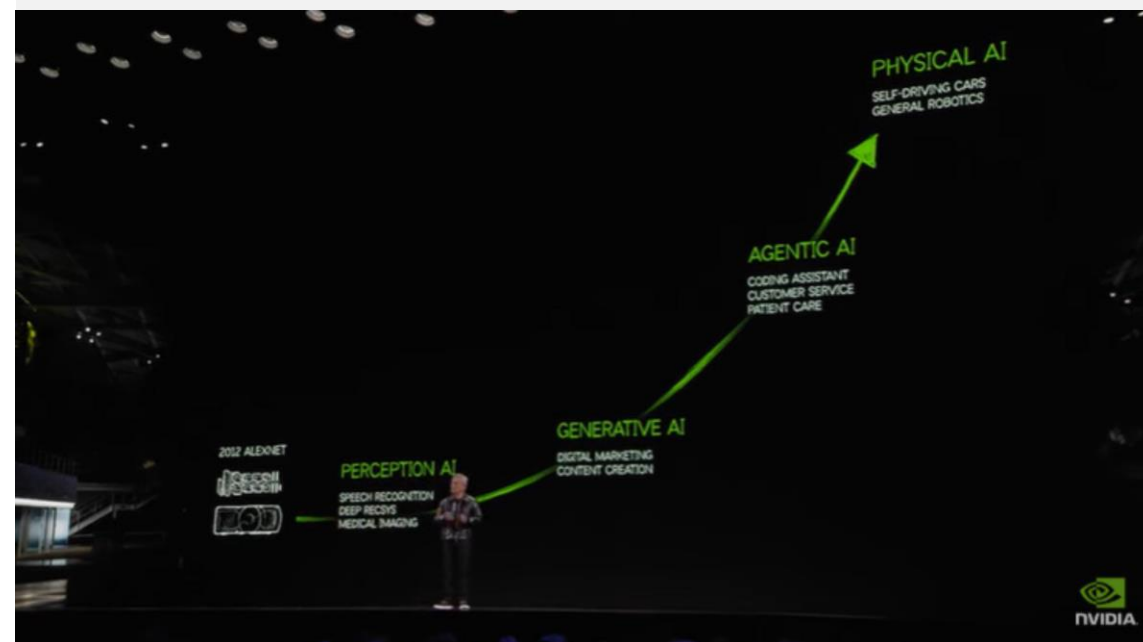
Pretraining→Post training→Reasoning



出典

<https://www.nvidia.com/en-us/glossary/cot-prompting/>

Generative→Agentic→Physical AI



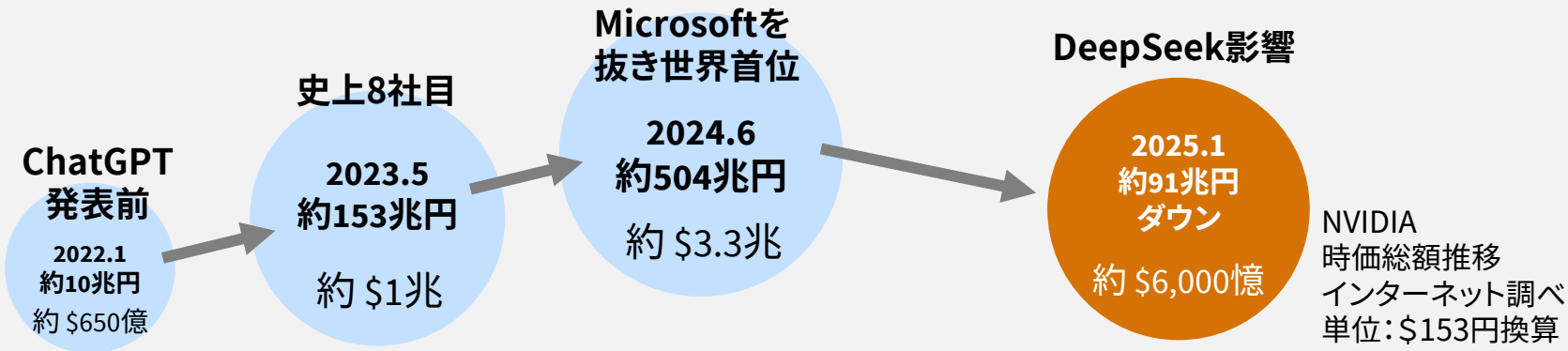
出典

<https://www.nvidia.com/ja-jp/events/ces/>

LLMの進展と企業価値の変化

乗り遅れた日本、LLM*を取り巻く世界の巨大経済圏の中心に食い込めず

米NVIDIA社
ChatGPTの頭脳を司る
「GPU」開発の立役者である
米NVIDIA社が世界を席卷



2017年

2017
論文発表 (Google)
『Attention Is All You Need』

2019年

2019
GPT-1発表

2022年

2022.11
GPT-3.5(ChatGPT)発表

2025年

2025.01
米国スターゲート・プロジェクト
発表
今後4年間で最大5,000億ドル (約
78兆円) の大投資計画

2025.01
DeepSeek (中国)
より安価で省エネ
AIモデルを発表

2017年の論文発表からわずか8年足らずで驚異的な経済規模へと成長

ChatGPTがもたらしたAI以外の領域における主な影響

- 世界中で半導体とデータセンターへの投資が激化 (2025年に投資総額20兆円超を予測)
- 電力不足がクローズアップされ発電設備企業の株高へとつながる

*ChatGPTはLLMを応用した対話型AI

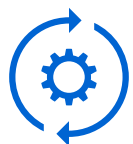
現在の生成AI活用で一番大切な事



今使っている技術が、すぐに陳腐化する事が分かっている事である



今の技術が古くなる事を前提に考えないといけない



昨日テストして使えなかったから、明日使えないとは限らない



AIを使ったシステム全体を、ラーニングシステム化しないといけない

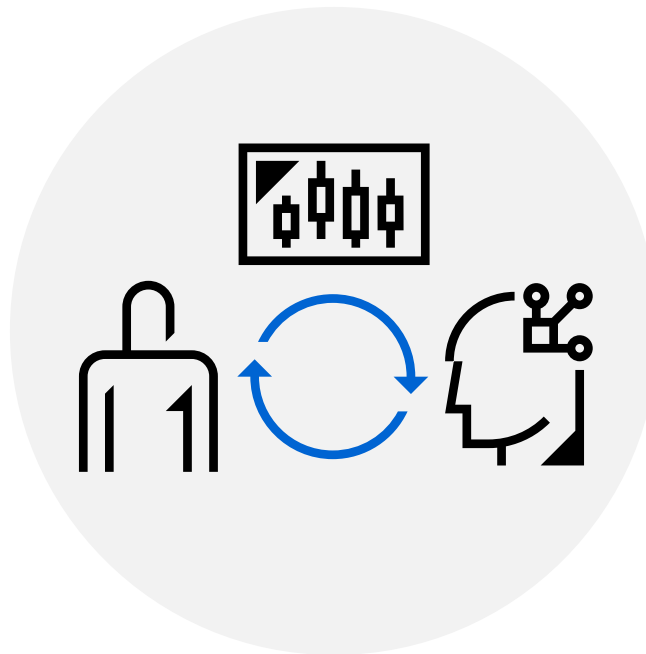


すなわち、品質を維持し、かつ向上できるように作らないといけない

ソフトウェア開発における生成AI活用の進化



生成AIが一問一答で
開発者を手助け



開発者と生成AIが
連携して問題に取り組む



生成AIが複雑なタスクを解決し
開発者が結果を確認する

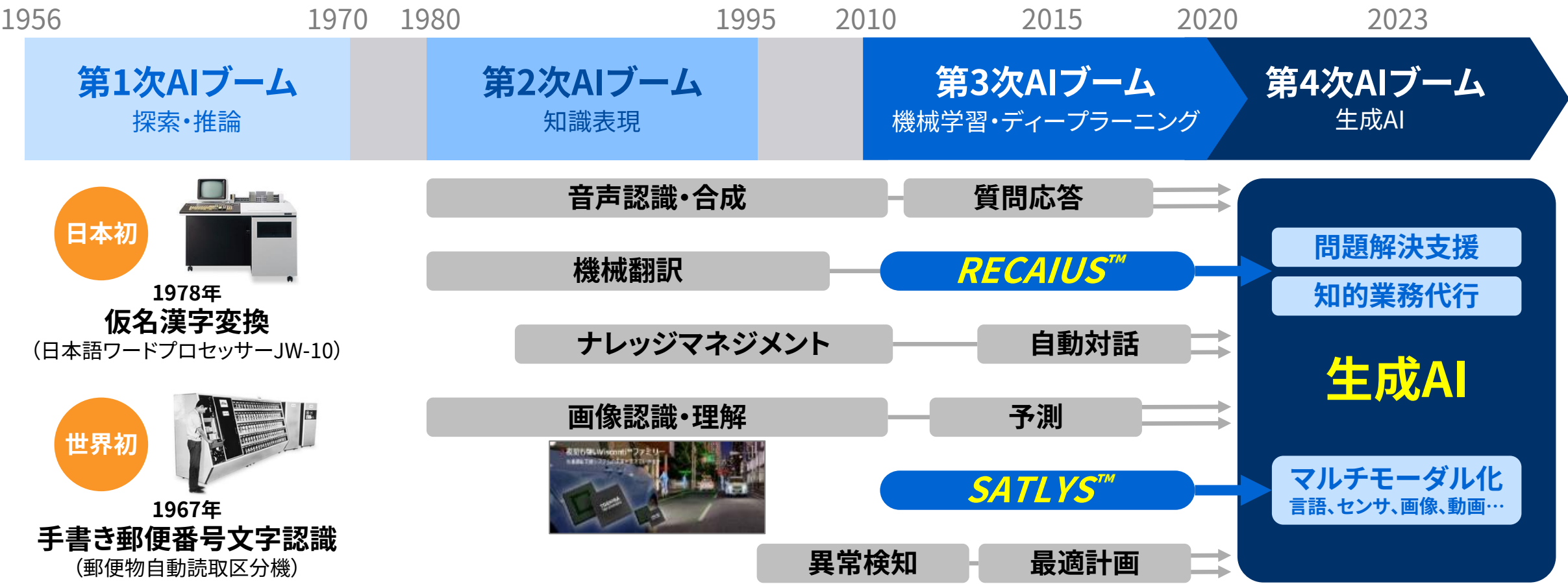
02

東芝の生成AI活用に向けた取り組み

東芝グループのAI技術の取り組み

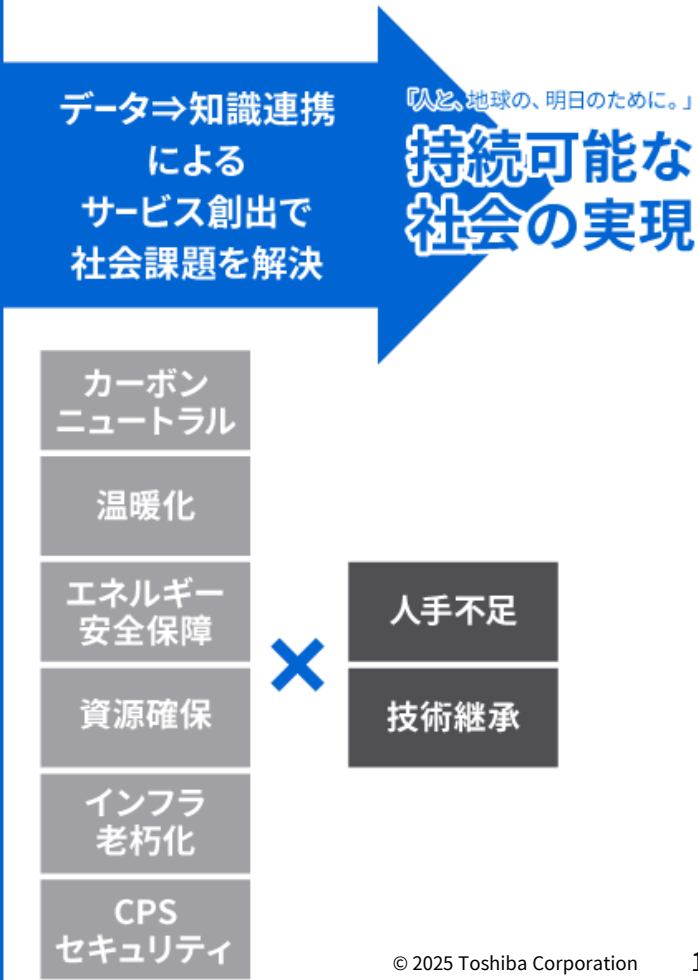
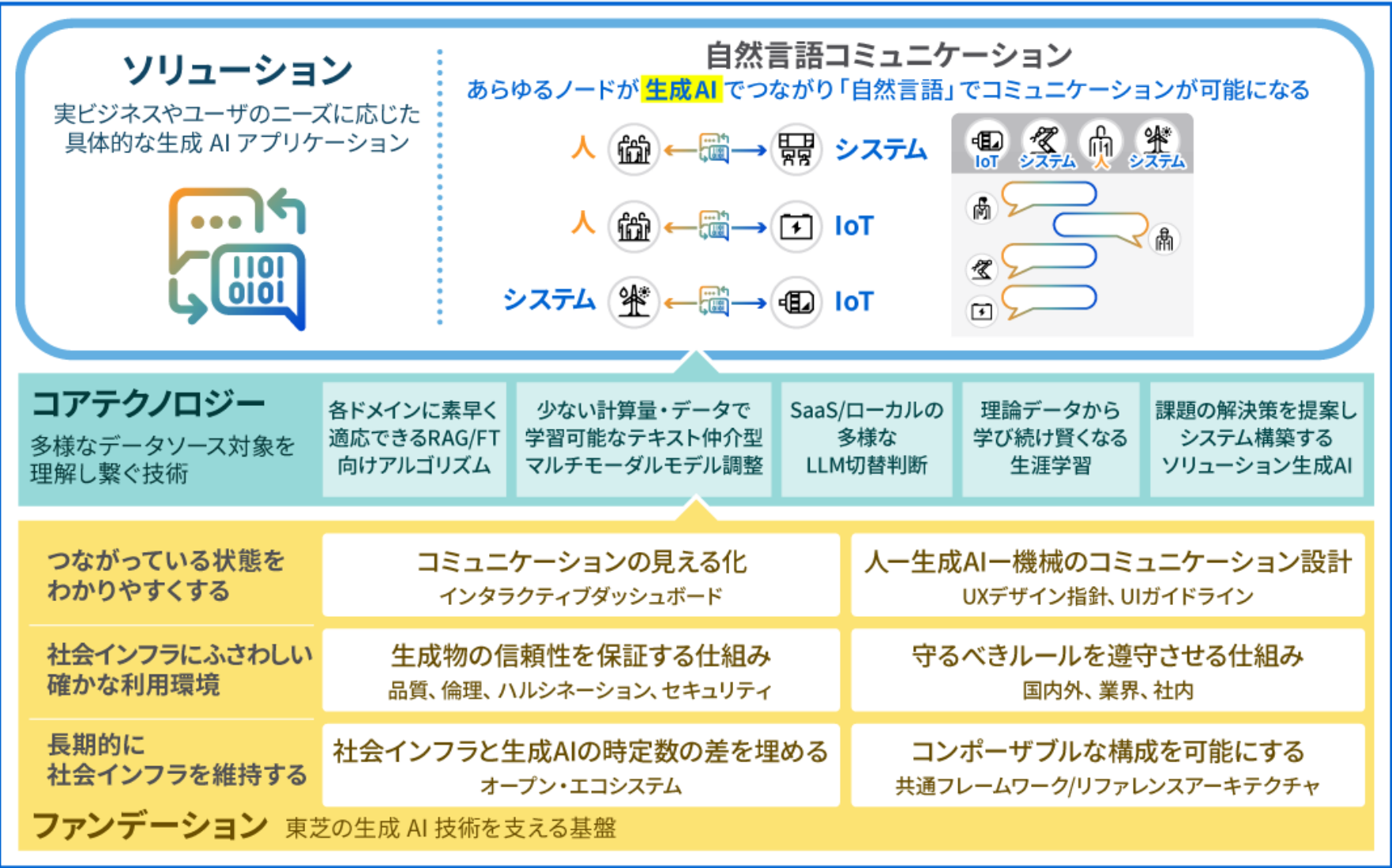
半世紀以上に渡るAI技術の蓄積とフィジカルデータでお客様の価値創出を支援

言語処理、音声・画像、分析分野で **50年以上の歴史** AI関連累計特許出願数 **世界3位/日本1位**
当社調査、2019年11月時点



東芝 生成AIのビジョン

生成AIによる多様なデータ・知識連携で、安全・安心で持続可能な社会インフラを支える



東芝 生成AIのビジョン:「人-生成AI-機械」のコミュニケーション設計

生成AIのリスク対策をユーザーインタフェース(UI)でも行う

背景:生成AIのリスク

東芝グループは、生成AIアプリ開発を推進する一方で、社会インフラ事業において生成AIのリスクが大きな問題につながることを認識し、慎重に取り組む必要があります。

■生成AIのリスクが問題(インシデント)につながる例

リスク

- 入力データのわずかなノイズに出力が影響される
- 正しく判断できていたデータに対して、追加学習後に誤った判断結果を出力する

インシデント

意思決定の誤り

例えば...
ポンプ設備の運用で生成AIを活用した場合

生成AI

20XX年に、判断を急いだ結果
○に繋がった事例があります。
また、○は正常に稼働しており
ポンプ稼働の基準には達していません。
ポンプ稼働はまだ早いです。

極端で不適切なアウトプット

ポンプ稼働の判断と
非常事態通知の遅れ

まだ稼働しなくても
よさそうだな

インフラシステム

大規模被害

生成AIは、曖昧な状況においても、都合の良い根拠やデータを抽出し、もっともらしい情報としてアウトプットする可能性があります。
非常時は、人が判断を迫られている状態で、精神的・時間的余裕が無いため、**判断を誤りやすい状態**にあります。

不適切に活用した結果起こりえる影響
大規模被害につながり経済活動が停滞／責任問題が問われる

リスク

- 他社の商標・登録商標・著作権・肖像権・権利を侵害する
- 著作権・肖像権・権利を侵害する
- 著作権・肖像権・権利を侵害する

インシデント

権利や利益の侵害、偽・誤情報の拡散

生成AI

他社の商標・登録商標・著作権・肖像権・権利を侵害する

著作権・肖像権・権利を侵害する

著作権・肖像権・権利を侵害する

リスク

- 人種・性別・年齢・性傾向などの属性に基づく偏見・差別を助長する
- 人種・性別・年齢・性傾向などの属性に基づく偏見・差別を助長する
- 人種・性別・年齢・性傾向などの属性に基づく偏見・差別を助長する

インシデント

不公平や偏見の強化

生成AI

人種・性別・年齢・性傾向などの属性に基づく偏見・差別を助長する

人種・性別・年齢・性傾向などの属性に基づく偏見・差別を助長する

人種・性別・年齢・性傾向などの属性に基づく偏見・差別を助長する

目的

- ・ リスクがインシデントに発展しないよう、ユーザーと生成AIアプリとの双方向のやりとり(インタラクション)の質を向上
- ・ ユーザーが直感的にリスクを回避、生成AIアプリを安全かつ便利に利用できるようにします

■対象者:

生成AIアプリの企画・開発関係者、
主にフロントエンド開発を行う設計者とデザイナー

■生成AIのリスク対策は『技術』と『ユーザーインタフェース』両方が必要

生成AIリスク対策	
技術	ユーザーインタフェース
- 安全対策(ガードレール) - 監視ツール(オペレーターバリエーション) - モデル・プロンプト品質向上	人と生成AIアプリのユーザーインタフェースの原則

■対象プロセス:

要件定義と設計・仕様、評価

内容

人と生成AIアプリのインタフェースの原則

生成AIアプリ開発における、ユーザーと生成AIアプリとのインタラクションを検討する際の注意点と検討アプローチ方法

■原則はアプリ利用のステップに合わせて大きく4つに分類

■SPOサイトのイメージ(原則3-2の例)

1. ユーザーが生成AIを適切に利用できるようにユーザーの知識と利用状況に適した機能を提供します
 - 利用前のユーザーとリテラシーと利用状況の把握
2. アプリの機能である生成AIを適切に利用できるようにします
 - 利用中、生成AIのアウトプット前を含むインタラクション
3. 生成AIのアウトプットを検証、補正、制御できるようにします
 - 利用中、生成AIのアウトプット後のインタラクション
4. 利用を繰り返すことで人のリテラシーとシステムの精度を向上できるようにします
 - 利用の繰り返しによる、ユーザーとアプリの成長

3-2. ユーザーが生成AIのアウトプットを検証できる情報や方法を提供します
参照元の確認・比較ができるように、ユーザーが元データにアクセスできるようにします。



※今後、利用環境Tとの協働で生成AI開発のための共通UI部品を検討予定

2023年

GIANT-LEAP:STEP: -1

生成AI活用の下準備 **見える化**



実態調査と先行利活用

見える化・試行

効果測定・試行する

(準備) 社内外調査

体制調査

利用シーン
使う情報の拡大

活用ステップ
活用範囲と成果の拡大

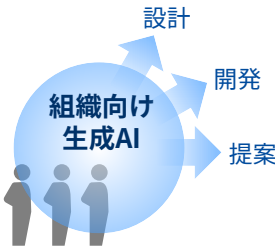
Generative AI Navigation HUB
「生成AI活用推進のためのコンサル/伴走、
周知/啓蒙、普及促進、利用ガイドライン、
運用を立案・実行の拡大」

2024年

誰でも

GIANT-LEAP:STEP: 0

生成AI活用の **民主化**



誰もが生成AIを活用している

定着

商用サービスを使う

社内文書
活用100%

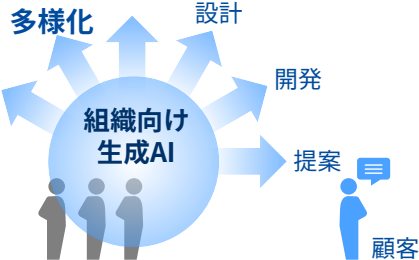
HUB体制スタート

2025年

いろいろ

GIANT-LEAP:STEP: 1

生成AI活用の **多様化**



いろんなシーンで使っている

進化

自組織情報を使う

マルチモーダル
応用100%

社内利活用常態化

2026年

その先へ

GIANT-LEAP:STEP: 10

生成AI活用による **業務改革**



顧客の課題解決に
生成AIで伴走している

改革

顧客情報を使う

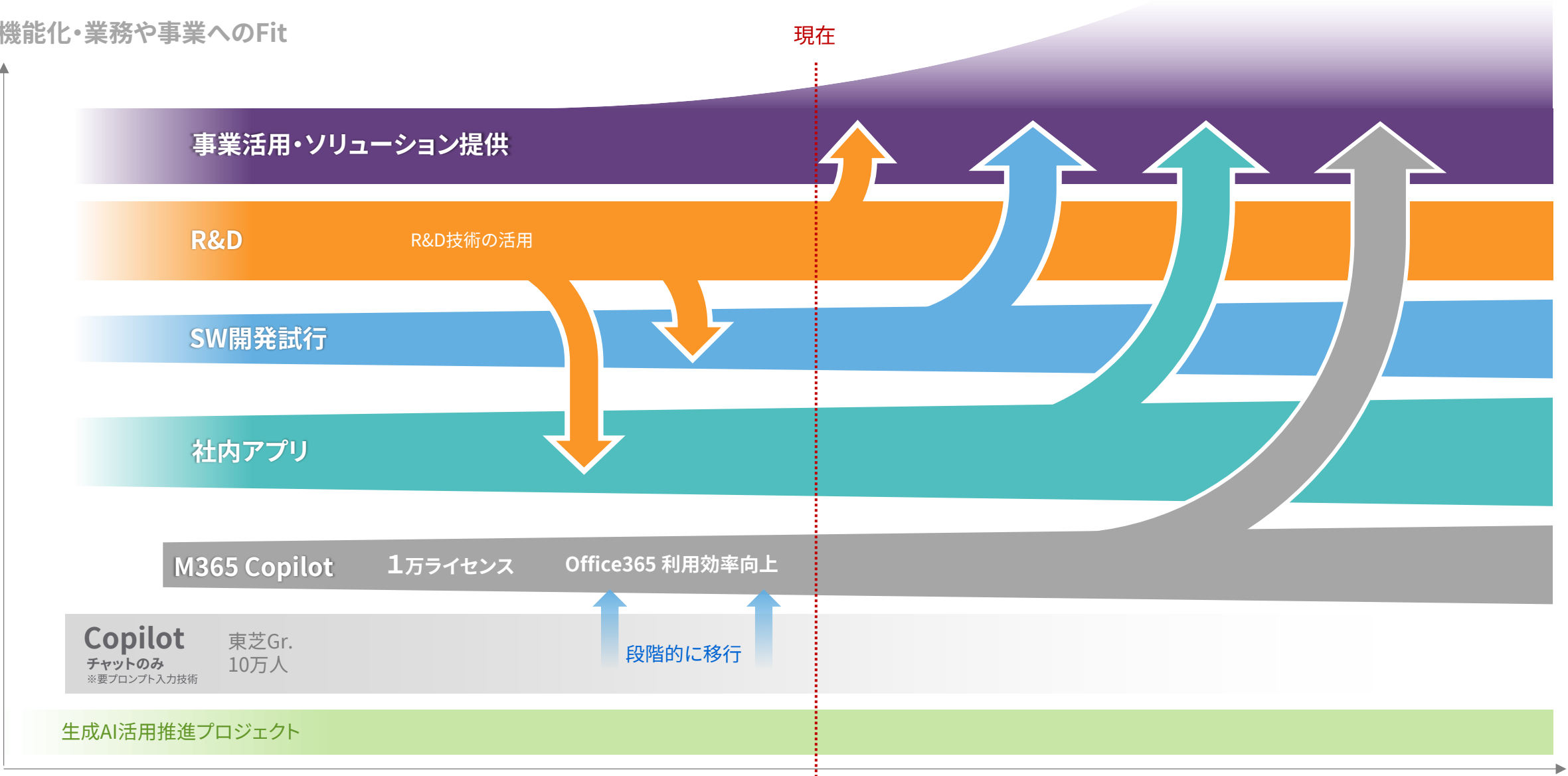
対象業務50%削減

事業活用常態化

全体構想Map

M365 Copilotをベースに、社内アプリ充実させ、事業活用へ展開させてゆく

高機能化・業務や事業へのFit



FY24 商用サービスを使う

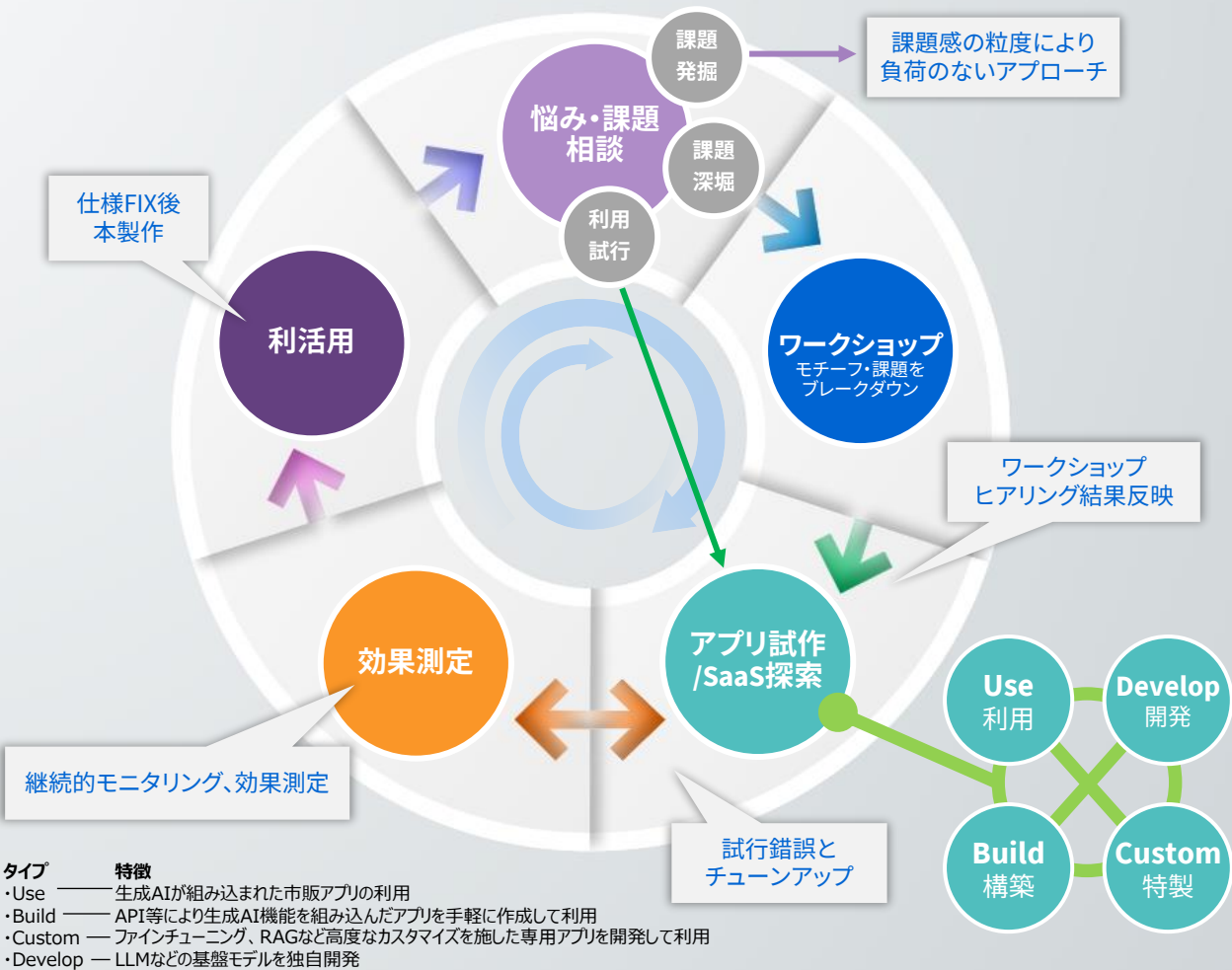
FY25 多様な情報に基づく活用

FY26 業務改革

生成AI活用推進の体制とミッション

生成AI利活用には試行錯誤と継続的なチューンアップが不可欠で、特にライフサイクル前半での伴走は必須

生成AI利活用のライフサイクル



チーム	ミッション
PMO (組織統制とPRを行う)	・PJ全体とりまとめ ・社内浸透広報活動(組織風土醸成)
コンサル/伴走 (実行を加速する)	・課題のブレークダウン、明確化 ・WS運営、解決策の可視化、明文化
周知/啓発 (情報・知識を広める)	・セミナーの開催、企画 ・教育の開催、企画
利用ガイドライン (ルールを定める)	・社内ルールの立案、浸透
運用 (円滑に利用する)	・利用環境マネージドサービス ・モニタリング、効果測定
利用環境 (道具・環境を用意する)	・試作アプリの開発 ・ユースケースの発信、促進

生成AIの社内活用による業務効率化 -試行から本活用へ-

トップダウンで改革を加速、1人あたり5.6時間/月の業務効率化余地を確認

ユースケースカタログを参照して業務効率化に取り組み、
1人あたり5.6時間/月の業務効率化余地を確認

利用申請者は、初期ターゲットを10,000名とし、現在**12,000名超**

利用したツール(複数回答あり)

	アンケート 回答結果	Teams	Out look	Chat	Word	PPT	Excel	One Note	White board
1人1ヶ月の工数削減数(想定)									
1.5時間未満	60	37	19	16	14	11	3	1	1
1.5-5時間	137	111	56	63	55	44	20	9	1
5-10時間	73	62	38	34	34	27	9	4	1
10-15時間	20	17	12	7	10	10	4	2	0
15-30時間	12	10	7	4	7	7	2	1	1
30時間より多い	5	4	4	1	3	1	1	1	1

平均値5.6時間/月

1人あたり:
5.6時間/月

コンサル/伴走
(実行を加速する)

利用環境
(道具・環境を用意する)

周知/啓発
(情報・知識を広める)

利用ガイドライン
(ルールを定める)

運用
(円滑に利用する)

※調査は生成AIによる業務改善トライアルに参加した400名を対象に実施

“稼ぐ力”を高めるために従業員 1 万人に Copilot を導入



マイクロソフト事例記事：

[「東芝再興計画」実現に向け、“稼ぐ力”を強化するために Microsoft 365 Copilot を従業員 1 万人に導入。業務プロセス全体の効率化へ活用シーンを拡大](#)

03

生成AIによるソフトウェア品質への取り組み

ソフトウェア品質における生成AIの実施例

ソフトウェア品質における生成AIの活用では、
既存の静的解析ツールや人間系のチェック/レビューと置き換えるのではなく、
技術により作業を補完/支援してより高い品質を達成するために生成AIを活用

生成AIの特徴



- 網羅的な結果生成に向かない
- 生成結果に確率で誤りが混入する



ソフトウェア品質においては、抜けが無く・正しい対応、が期待される

ソフトウェア開発工程と対応づけて整理した生成AI活用技術の例

ソフトウェア品質における生成AI活用例

A ドキュメントやソースコードのレビュー

B テストケースやテストコードの作成

C ドキュメントやソースコードの作成

要件定義	設計	開発	テスト	運用・保守	PJ管理
	設計書 生成・修正・補完	ソースコード 生成・修正・補完	テスト生成 (シナリオ, データ, コード)	マニュアル生成	議事録自動生成
A	C	マイグレーション	B		
	設計レビュー支援	コードレビュー 支援			
	文章チェック				
要求仕様分類			テスト実施 優先度提案		OSS違反検出 OSS提案
要求仕様書 理解支援	設計書 理解支援	ソースコード 理解支援			
ドキュメント検索		ソースコード 検索		異常検知	QA生成

A

ドキュメントやソースコードのレビュー

ドキュメントやソースコードのレビューにおける生成AIの活用



生成AIが誤りや問題点を指摘してくれ、
誰でも品質の高い成果物を作れるようにしたい



静的解析やルールベースと比較して指摘の網羅性や
指摘内容の妥当性に課題があり、実活用に繋がりづらい



既存のツールの代替ではなく補完として生成AIを使い、
これまで発見できなかった問題点を発見する

ドキュメントやソースコードのレビューにおける生成AIの活用



ドキュメント



ソースコード



静的解析
ルールベース

- 条件を明文化できる問題点の指摘に強い
- 問題の可能性のある箇所を網羅的に指摘できる



過去のレビュー記録



トラブル報告書



生成AI

- 過去のレビュー記録やトラブル報告から、言語化の難しい問題指摘を実現できる
- レビュー対象の記載に現れていない抜けや検討不足を指摘できる



ドキュメント



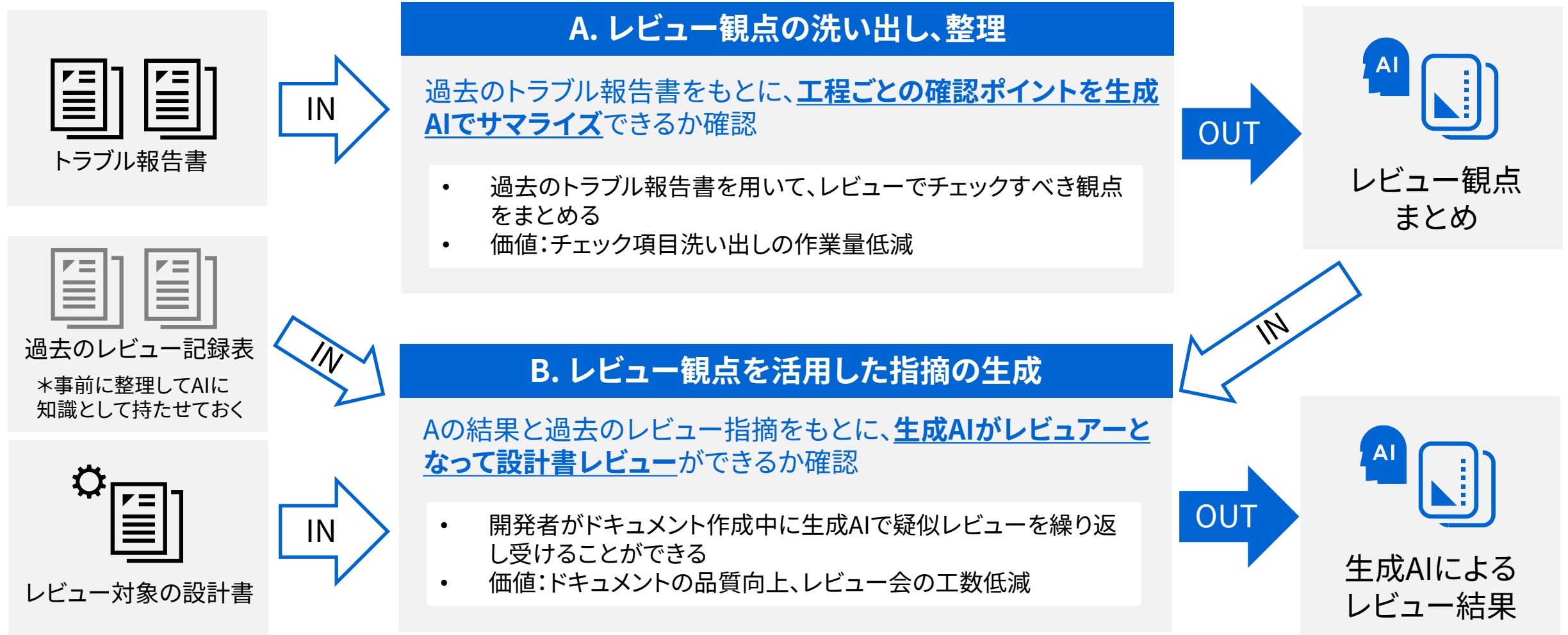
ソースコード



既存技術と生成AIの組合せで
網羅的・具体的な問題の解消と
言語化の難しいノウハウに
基づく問題の解消を実現

ドキュメントレビューの実施例

2つのステップで過去のトラブル報告やレビュー指摘を活かす



A. レビュー観点の洗い出し、整理：生成結果の構成

要件定義工程

レビュー観点をソフトウェア開発工程ごとにグループ化して生成し、対象工程の担当者に活用いただく

過去の発生トラブル

- トラブル番号：1429, 1837, 1865, 2098
- トラブル内容：
 - 画面サイズとフォントサイズの組み合わせに不足がありメッセージの表示崩れが発生(1429)
 - エラーメッセージと画面表示方法が定義されておらず意図しない表示不具合が発生(1837, 1865)
 - PC端末やブラウザの対象範囲や想定外ケースを正しく要件化できずエラーが発生(2098)
- 対応内容：
 - 画面レイアウトと画面方式にフォントサイズについての記載欄を追加(1429)
 - メッセージ追加時に画面レイアウトとメッセージ仕様書をペアで更新するよう作業手順を改定(1837, 1865)
 - システムの構成要素を分解して要件化しそれぞれに合意を取る(2098)

対象工程に関連する過去トラブルを整理して番号・内容・対応を生成し、レビュー観点の洗い出し結果の確認に活用いただく

レビューすべき内容

No.	確認内容	確認内容の具体例	関連トラブル番号
1	画面レイアウトと表示要素の確認	画面レイアウトごとに、画面項目・画面サイズ・フォントサイズ・表示方式を明記し、レビュー会で確認する	1429, 1837, 1865
2	異常系動作の確認	異常動作の要件化と、機能仕様・画面仕様を一覧化し、テストフェーズで反映されているかを確認	1837, 1865, 2098

過去トラブルに基づくレビューすべき内容を生成し、レビュー時のチェック項目として活用いただく

外部設計・内部設計工程

過去の発生トラブル

~~~~~省略~~~~~

## B. レビュー観点を活用した指摘の生成: 生成結果の構成

以下、「CS Tools V2.1 基本設計書」のレビュー結果を示します。

---

### ## 4. システム方式

#### ### 4.4. インフラストラクチャ方式の記載漏れ

##### \*\*該当箇所\*\*

4.4.2 保守対応方式

4.4.4 接続／通信方式

→「対象外」と記載されている

##### \*\*指摘内容\*\*

「保守対応方式」および「接続／通信方式」は、設計書に必ず記載すべき内容です。過去のレビューでは「詳細設計書の記載内容への参照を記載すること」と指摘があります。

---

レビュー対象ドキュメントの章節ごとに、レビュー該当箇所と指摘内容を生成し、ドキュメントの修正作業で活用いただく

#### ### 5.1. アプリケーションアーキテクチャ

~~~~~省略~~~~~

レビュー会で想定される質問を生成し、想定回答の検討で活用いただく

2. 想定質問

1. **メール送信サービスの文面はどこに記載されていますか？**
2. **「対象外」となっている方式の詳細は、どの資料に記載されていますか？**
3. **「サービスレベル定義」はどこで参照できますか？**

3. その他・補足

- 誤字・脱字は特に見当たりませんでした。
- 全体的に機能仕様は充実していますが、非機能要件や外部連携仕様の明記が不足しています。
- 添付資料の参照方法や要約を本文に記載することを推奨します。

ドキュメント全体に関する指摘・評価を生成し、ドキュメントの品質向上に活用いただく

以上、ご確認をお願いいたします。

B

テストケースやテストコードの作成

テストケースやテストコードの作成における生成AIの活用



生成AIが仕様や設計の観点からテストを生成し、ソフトウェアに問題が無いことを確認してくれる

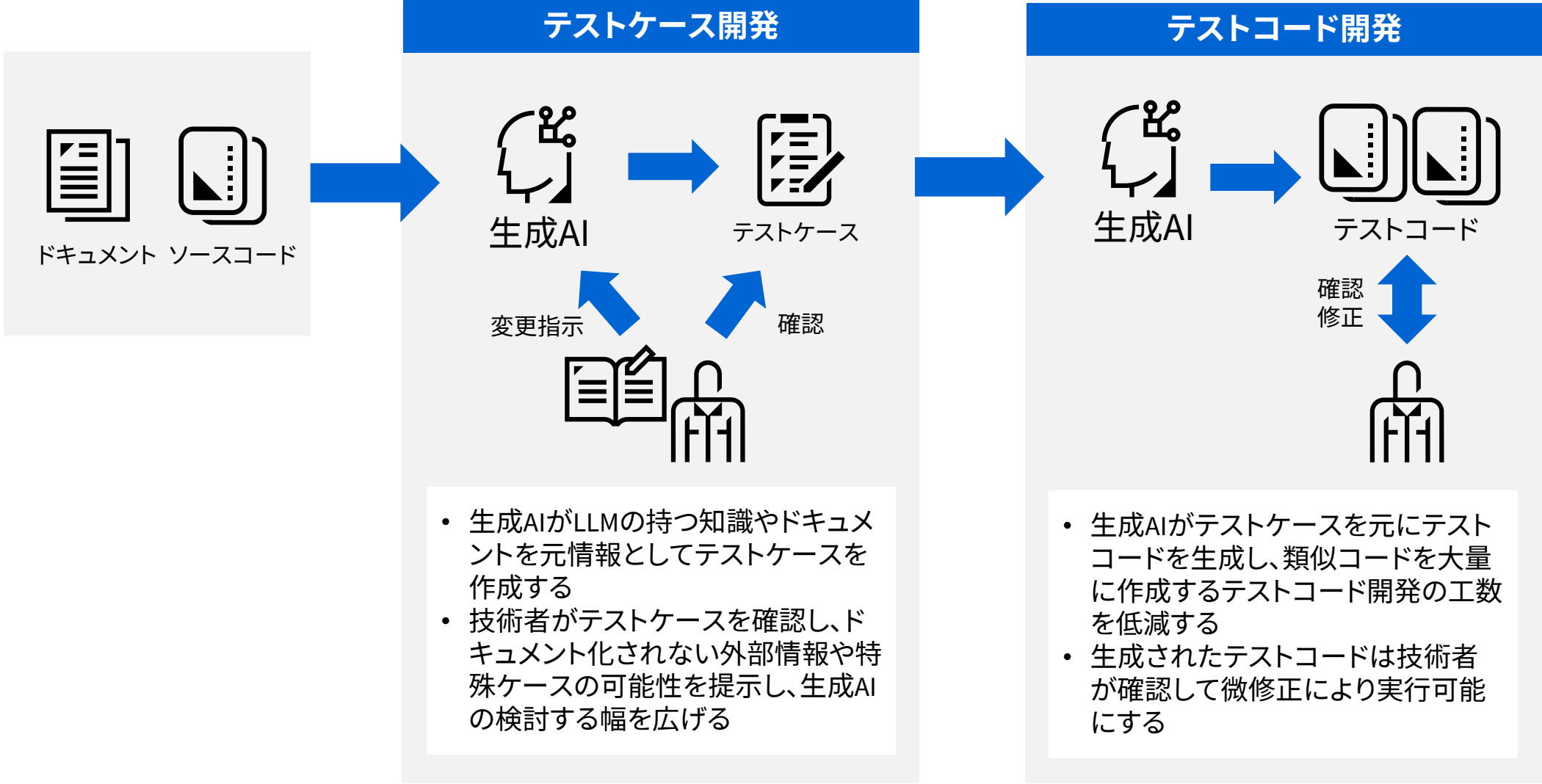


ありきたりなテストや類似ケースが多く生成され、コーナーケースや仕様想定外の動作に対するテスト作成に向かない



テストケース開発とテストコード開発のそれぞれで、技術者と生成AIが連携してテスト品質を高める

テストケースやテストコードの作成における生成AIの活用



テストコード生成の例

テストコードの生成実験では、生成されたテストコードの分岐網羅率は平均91%、
人手でのテストコード作成時間と比べて66.7%の作業時間削減

テストケース

1 参照 使用済み >

以下に、指定された制約条件と出力形式に基づいて、`check_data_validity` 関数のテストケースを提案します。

テストケース1: 正常系

事前条件

- `data_path` が有効なパスである
- `data_path` が存在する
- `data_path` に `images.zip` と `annotation.csv` が存在する
- `images.zip` には有効な画像ファイルが含まれている
- `annotation.csv` には、存在する画像ファイル名と0-9の範囲のカテゴリが含まれている
- 画像のピクセル値が0-255の範囲内である

テストの確認内容

- `check_data_validity` 関数がエラーなく実行される
- `check_data_validity_results.json`

期待値

- `check_data_validity_results.json` の `result` が `pass` である
- `not_exist_images`, `failed_to_load_images`, `out_of_range_images`, `out_of_range_category` のリストが全て空である

テストケース2: 異常系 - `data_path`

事前条件

- `data_path` が存在しない

テストの確認内容

- `check_data_validity` 関数がエラーをスローする

期待値

- `data_path {data_path} does not exist.` というエラーメッセージがスローされる

事前条件は、正常系の場合のif文の条件を満たすものが全て出力された。

期待値も、正常系の場合の最終的な結果が具体的な値とともに出力された。

異常系として、事前条件を満たさない場合のテストケースが一通り出力された。

テストコード

1 参照 使用済み >

以下に、指定されたテストケースに基づいて、テストコードを生成します。

```
import pytest
import os
import shutil
import pandas as pd
import cv2
import numpy
import json
from check_data_val

def setup_function():
    if not os.path.exists('test_data'):
        os.makedirs('test_data')

def teardown_function():
    if os.path.exists('test_data'):
        shutil.rmtree('test_data')

def create_test_data(images_exist=True, annotation_exist=True, invalid_image=False, invalid_category=False):
    if images_exist:
        cv2.imwrite('test_data/images.zip', np.random.randint(0, 256, (100, 100, 3), dtype=np.uint8))
    if annotation_exist:
        df = pd.DataFrame({
            'file_name': ['images.zip'] if not invalid_image else ['invalid.jpg'],
            'category': [5] if not invalid_category else [10]
        })
        df.to_csv('test_data/annotation.csv', index=False)

def test_check_data_validity_case1():
    create_test_data()
    check_data_validity('test_data')
    with open('check_data_validity_results.json', 'r') as f:
        results = json.load(f)
    assert results['results'][0]['result'] == 'pass'
    assert not results['results'][0]['not_exist_images']
    assert not results['results'][0]['failed_to_load_images']
    assert not results['results'][0]['out_of_range_images']
    assert not results['results'][0]['out_of_range_category']

def test_check_data_validity_case2():
    with pytest.raises(Exception, match=r"data_path test_data does not exist."):
        check_data_validity('test_data')

def test_check_data_validity_case3():
    create_test_data(images_exist=False)
    with pytest.raises(Exception, match=r"data_path test_data does not contain images."):
        check_data_validity('test_data')
```

- 生成されたテストコードは、テスト対象を呼び出す部分の修正のみで実行できた。
- テストはソースコードとしては全て成功するはずのもので、テスト結果はPassed3件、Failed6件。
- テストコードの修正でPassed8件、Failed1件となった。Failedの1件はテストデータの準備が困難なもの

C

ドキュメントやソースコードの作成

ドキュメントやソースコードの作成における生成AIの活用



生成AIへ要求を伝えればソフトウェアが自動で開発され、非技術者でもソフトウェアを開発できるようにしたい



生成結果には確率的に誤りが混入するため大きなソフトウェアを生成すると確認が困難、また生成AIが要求の行間を読んで補完し想定しない機能を生成することがある



生成AIによる成果物生成は有識者の作業支援をメインとし、有識者による確認を必須という前提のもと、あえて途中成果物を生成することで人間系での確認を容易にする

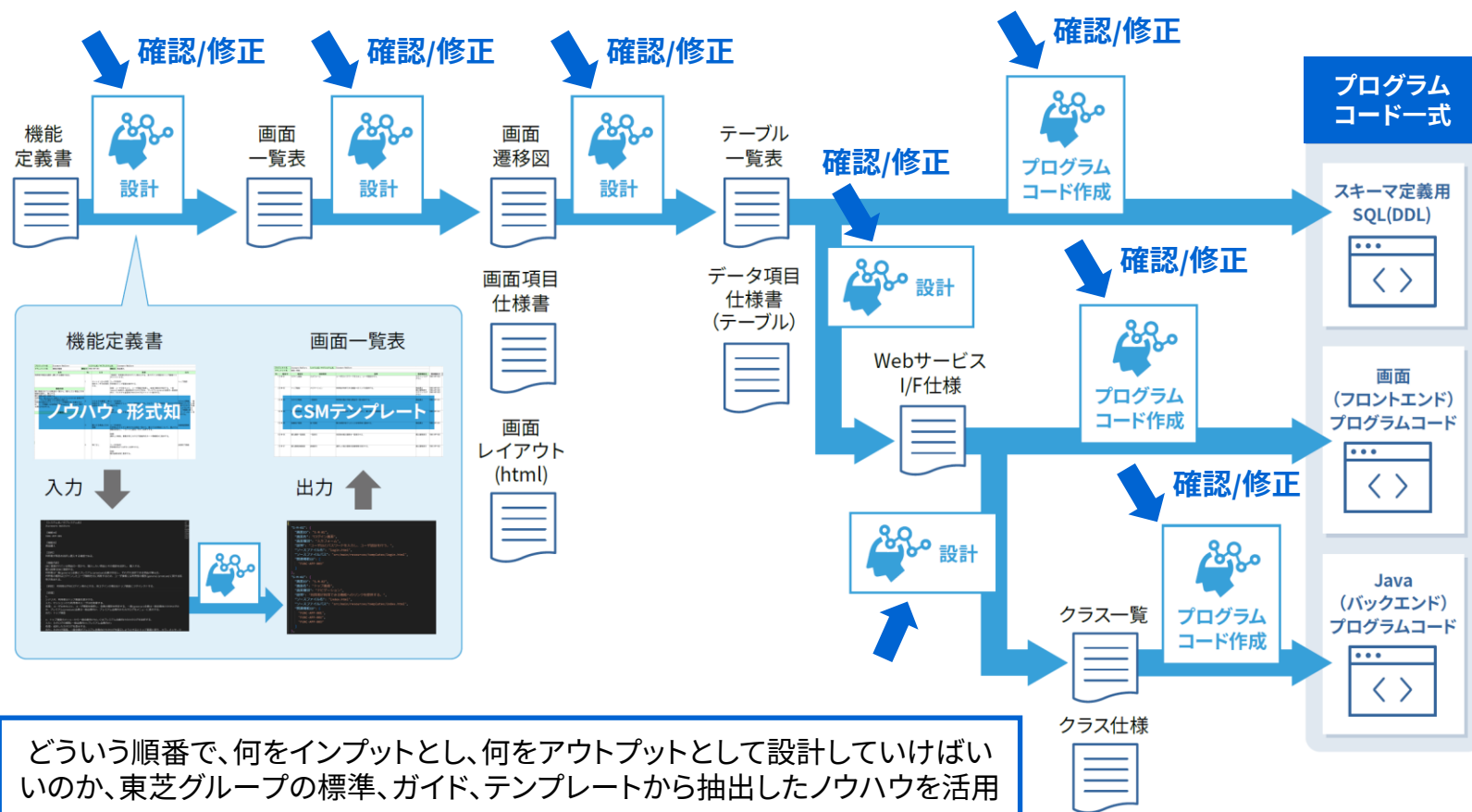
設計書からのソースコード生成

システム開発のノウハウを基に、設計を段階的に詳細化してソースコード一式を生成



ソースコードまで一度で生成すると確認が難しい

- ・仕様との不一致
- ・動作パターンの網羅
- ・エラー処理の不足



CSM: CommonStyle Methodology, I/F: interface, SQL: Structured Query Language, DDL: Data Definition Language

生成結果の評価

生成したプログラムコードについては一定量の手直し、コード追加が必要ではあるが、大部分はそのまま利用可能

本検証ではECサイトのアプリケーション開発をモチーフとし、「商品購入」、「購入履歴表示」の機能定義書を入力として利用

生成したアプリケーションの動作イメージ



生成/手直したソースコードの規模

| No | 対象 | 生成したコード規模 | | | 生成後に人手で手直したコード規模
(コード行数のみをカウント) | | | |
|----|--------------------|-----------|------------------------|-------|------------------------------------|------|------|--------------------|
| | | ファイル数 | 総行数
(コメント/
空行含む) | コード行数 | 削除行数 | 修正行数 | 追加行数 | 手直し後の
コード
行数 |
| 1 | スキーマ定義用SQL(DDL) | 1 | 48 | 44 | 9 | 11 | 27 | 62 |
| 2 | 画面(フロントエンド)ソースコード | 7 | 296 | 294 | 12 | 10 | 93 | 375 |
| 3 | Java(バックエンド)ソースコード | 19 | 1,365 | 497 | 78 | 26 | 207 | 626 |
| 4 | 合計 | 27 | 1,709 | 835 | 99 | 47 | 327 | 1,063 |

生成した実装コードの**17%**は**削除、修正**が必要であったが、**残りの83%**はそのまま**利用可能**

複雑な処理は**未実装部分も多く**、
生成後にある程度**コード追加が必要**

17%を削除、修正し、残りの83%はそのまま利用

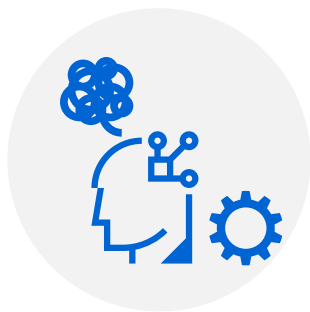
手直しやコード追加が必要だった例

- ・セッション管理や、複雑なトランザクション処理(商品購入)が**未実装**で、コード追加を実施
- ・フロントエンド-バックエンド間、バックエンド-DB間の**データ項目の整合が取れておらず**、修正を実施

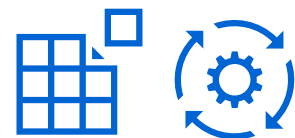
04

クリティカルシステムにおける生成AI活用

クリティカルなシステムに求められるソフトウェア開発の品質と生成AIの活用



クリティカルなシステムにおける生成AI活用の難しい点

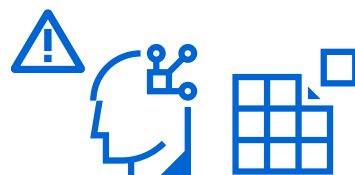


システム全体の挙動とソフトウェアのロジックを併せて検討し、
高い品質を作りこむ必要性

クリティカルなシステムに求められるソフトウェア開発の品質と生成AIの活用



社会インフラのソースコードには、厳密に動作ロジックやタイミングが規定され、製品個体に合わせて調整しなければならない部分がある



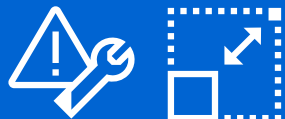
生成AIの活用では、生成結果の確認により高いレベルの注意が必要：
コードレベルで詳細に挙動を把握しロジックの確認や調整をする



生成AI活用を求める品質の違い



- 一般的な製品では、ドキュメント作成やソースコード開発に全般的に生成AIを活用してドラフトを作成、それを開発者が確認・修正することで開発工数低減を実現できる
- クリティカルな製品では、システム構想段階からシステム全体にわたる品質作りこみを計画し、厳密に製品を作りこむ文化
- 生成AIの特徴による、確率で誤りが混入する挙動や、生成するごとに結果にブレが出る点は文化にそぐわない



技術的な解消、適用範囲の検討が求められる

クリティカルなシステムに求められるソフトウェア開発の品質と生成AIの活用



技術による解消

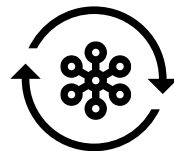


現在



人間系による確認・修正しテストで結果を確認することから、統計的手法など他手法を組み合わせで品質を担保する

将来



Agentの組み合わせや連携により品質を担保する仕組みを構築する

Fly By Wire 航空機のデジタルにおける多重化

品質を多重化で担保するような仕組みが必要になる、信頼性

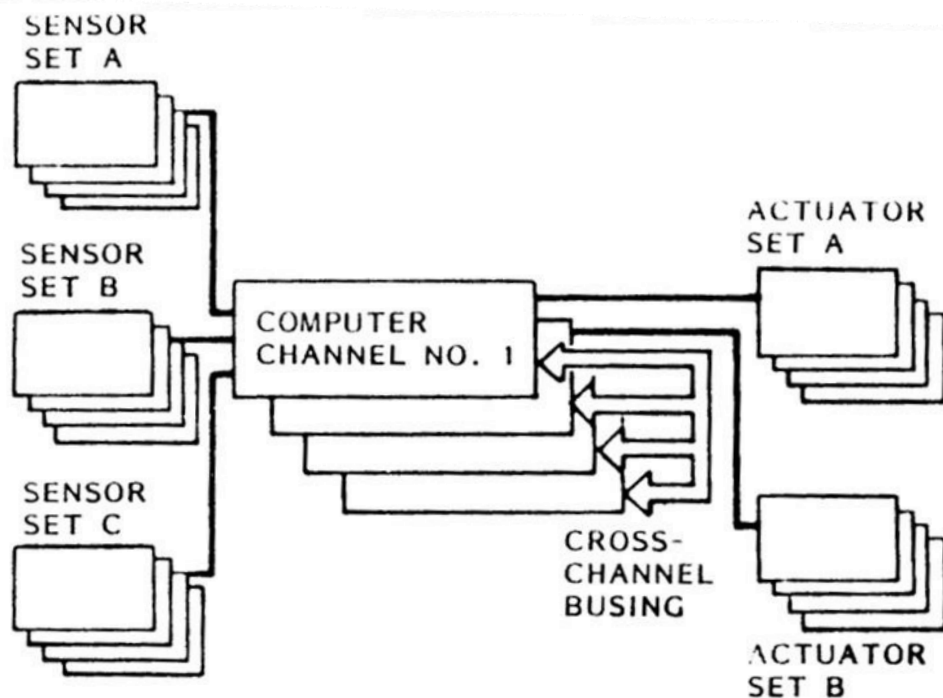


Figure E-1. Quadruplex DFCS Architecture

出典

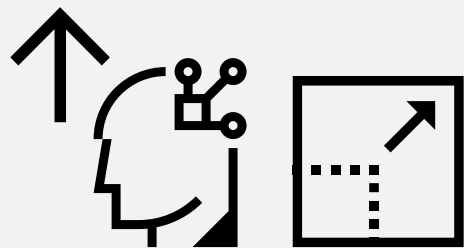
<https://ntrs.nasa.gov/api/citations/19890000687/downloads/19890000687.pdf> p11(vii)



05

生成AIを活用したソフトウェア開発の将来

AIを使いこなす人材、その育成がコアコンピタンスとなる



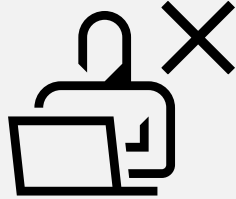
生成AIの技術は急速に向上し、活用できる対象業務や問題の複雑さは驚異的に拡大し続けている



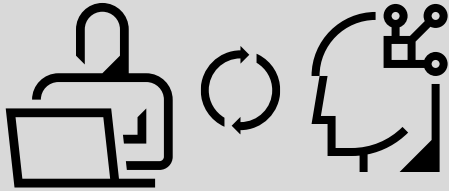
将来には、少数の生成AIを使いこなす技術者が大量の業務をこなすことができるようになる

AIを使いこなす人材、その育成がコアコンピタンスとなる

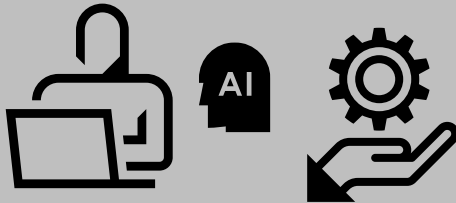
NO



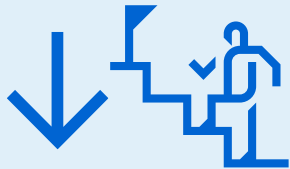
生成AIの進化により、ソフトウェア開発者は不要になるのか？



生成AIに指示する・結果を確認するといった使いこなす技術者、自身の本質的な要求を認識していないユーザと生成AIの間を取り持つ技術者、が必要



AIを使いこなせる人材、AIを舐めて結果を活用できる人材の必要性が高まる

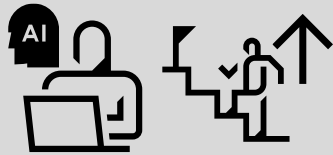


しかし人材は育てなければ増えない、生成AI活用の拡大によりソフトウェア技術者が経験を積む機会は減少していく

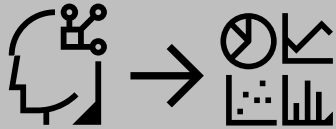
AIを使いこなす人材、その育成がコアコンピタンスとなる



将来にむけて企業が取り組むべきことは？



AIを使いこなせる技術者の育成スキームの構築と、
経験を積む機会の創出



AIの成果物を客観的に計測・評価する方法の構築



これまでの、後輩を教育し、技術を継承してきた
現在は、AIに対して良い成果物(ドキュメント、ソースコード)を作れるように、
AIを教育している



将来は、AIを教育しつつ合わせて人も育て、
AIとの連携で技術を継承する人材を育成していく

06

結局重要なのはデータになる

スマートマニュファクチャリングの実現を支援する製造業向けソリューション

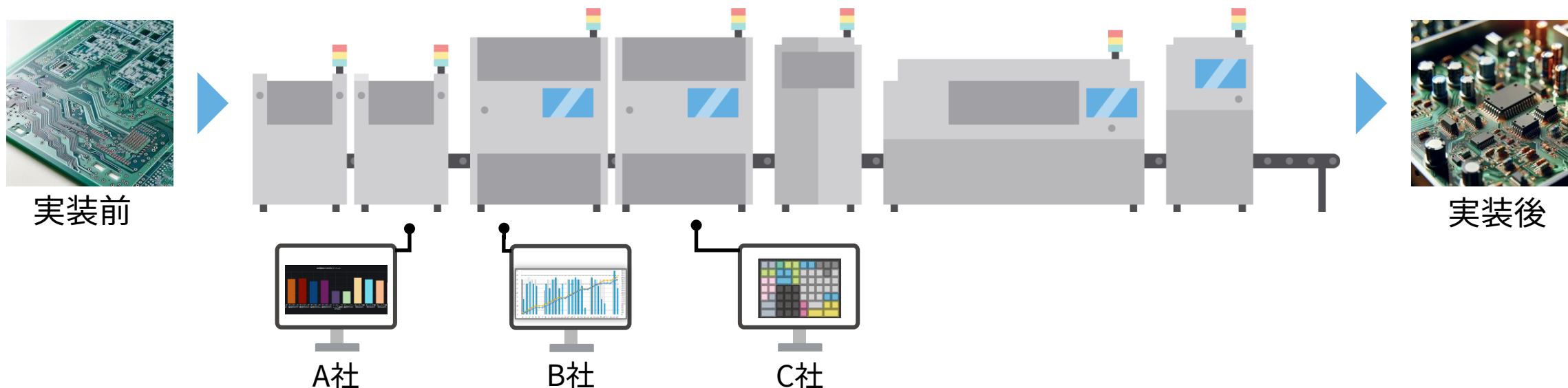
Information Technology

Operational Technology

| 設計 | 調達 | 生産 | 運用・保守 |
|---|---|--|---|
| <p>分散連成シミュレーション</p> <p>VenetDCP®</p> <p>異なるツールを接続しデジタル試作を実現</p> <p>オープンアーキテクチャPLM</p> <p>Aras Innovator</p> <p>製品ライフサイクルをカバーするPLM</p> | <p>サプライヤコミュニケーション基盤</p> <p>戦略調達ソリューション Meister SRM™</p> <p>バイヤ-サプライヤ間の様々な情報共有を支援</p> <p>サプライチェーンプラットフォーム Meister SRM ポータル™</p> <p>サプライチェーン全体構造とリスクを可視化・分析</p> | <p>製造データ収集・活用基盤</p> <p>ものづくりIoTソリューション Meister Factory シリーズ</p> <p>製造工程における計画・実績データ統合基盤</p> <p>製造データ基盤クラウドサービス Meister Manufact X™</p> <p>製造実行ソリューション</p> <p>半導体製造業向け製造実行システム Meister MES™</p> <p>半導体製造向け製造実行システム</p> <p>製造実行システム Meister MES NEO™</p> <p>多種多様な生産形態に対応した製造実行システム</p> <p>ヒトの活動見える化</p> <p>Meister Apps® 現場作業見える化パッケージ</p> <p>現場作業員活動の見える化と改善推進</p> <p>旧型設備の活用</p> <p>Meister Apps® 設備あやつり制御パッケージ</p> <p>RPA技術による設備の自動化/遠隔操作</p> <p>外観検査自動化</p> <p>Meister Apps® AI画像自動検査パッケージ</p> <p>東芝独自の良品学習による検査の自動化</p> <p>3D データ活用</p> <p>Meister Apps® 製造現場の3D CADデータ活用ソリューション Meister MR Link™</p> <p>MR技術による立体物の検査支援</p> | <p>工程改善支援</p> <p>Meister Apps® 工程改善アシストパッケージ for SMTライン</p> <p>製造ラインの設備データで工程改善を推進</p> <p>アセットIoTデータ基盤</p> <p>設備・機器メーカー向けアセットIoTクラウドサービス Meister Remote X™</p> <p>設備・機器メーカーのリモートメンテナンス支援</p> <p>工場・プラント向けアセットIoTクラウドサービス Meister Operate X™</p> <p>工場やプラントのオペレーションサポート</p> <p>現場業務のデジタル化ソリューション Meister AR Suite™</p> <p>AR技術による設備運転/保全支援</p> |

| | | |
|--|---|---|
| <p>はか</p> <p>電磁流量計
マイクロ波濃度計</p> <p>鉄鋼圧延ライン用計測器 (T-CAT)</p> <p>リストバンド型センサ</p> | <p>あやつる
みまもる</p> <p>計装クラウドサービス</p> <p>産業用コンピュータ</p> <p>産業用コントローラ</p> <p>計装コンポーネント仮想化プラットフォーム Meister Controller Cloud™ PLCパッケージ typeN1</p> <p>リモートで製造ラインの制御やメンテナンスを実現</p> <p>計装クラウドエンジニアリングサービス Meister nV-Tools Cloud™</p> <p>クラウド上でコントローラの開発環境を管理</p> | |
| <p>見積授受</p> <p>各種調査依頼</p> <p>BCP管理</p> <p>サプライヤ</p> | <p>試作</p> <p>計画</p> <p>製造装置</p> <p>作業指示</p> <p>検査</p> <p>作業者</p> <p>稼働監視</p> <p>エネマネ</p> <p>稼働履歴</p> <p>自社工場</p> | <p>遠隔監視</p> <p>故障予兆</p> <p>保守履歴</p> <p>顧客</p> |

様々な設備データの一元管理や、人手に頼った分析改善作業に課題



設備データの自動収集・蓄積・活用

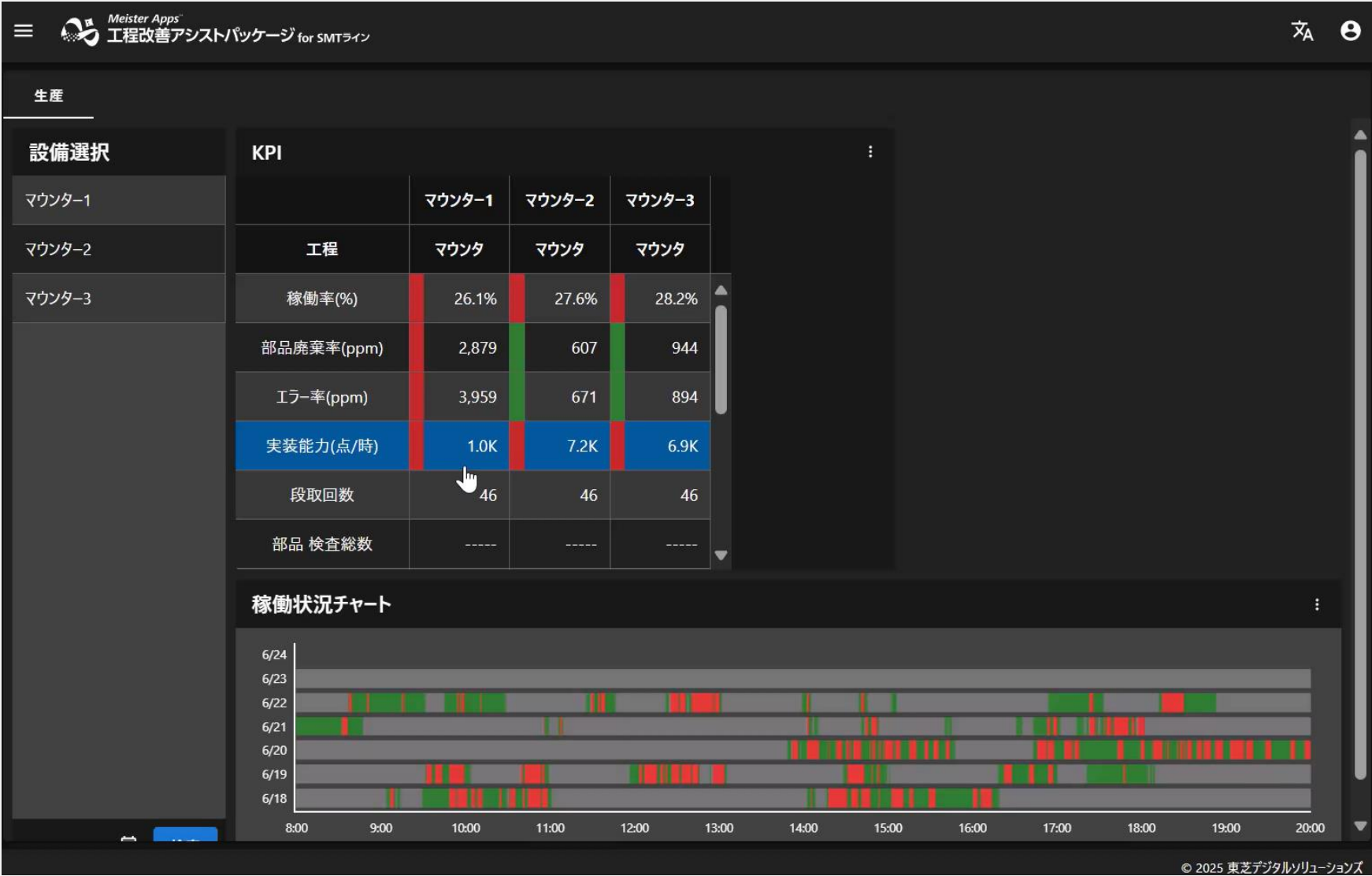
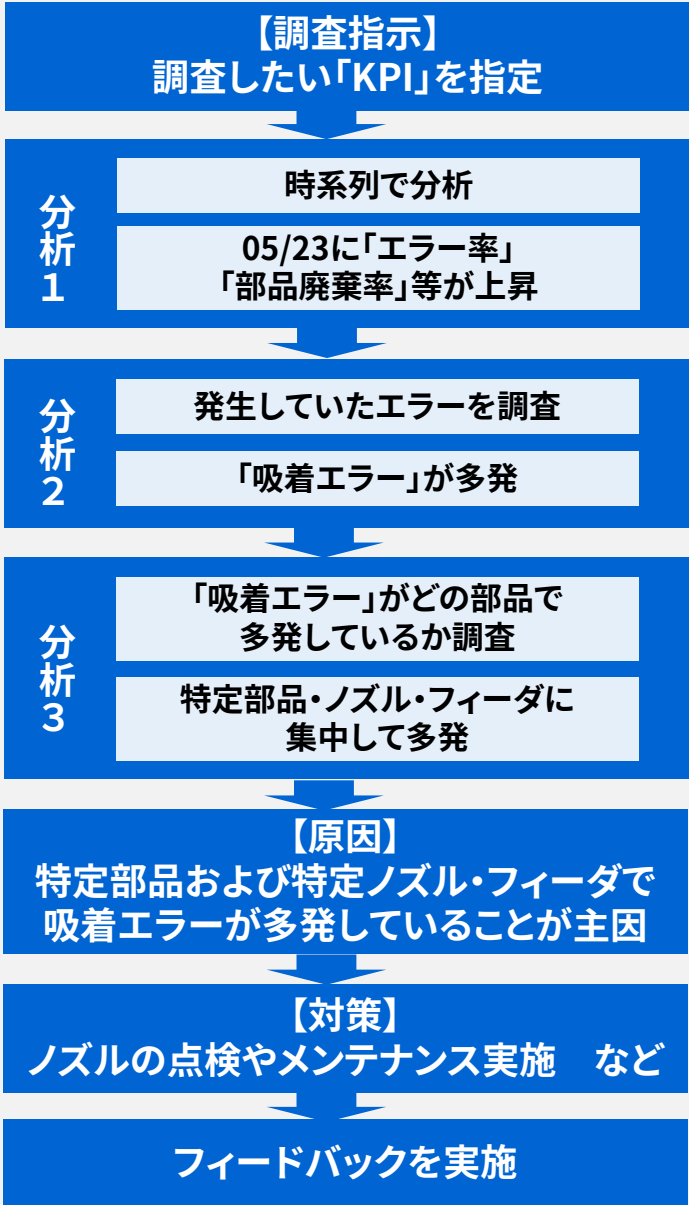
Meister Apps™
工程改善アシストパッケージ for SMTライン

熟練者に頼らない工程改善

AIエージェントによる分析・改善提案
※開発中

SMTラインの
工程改善を加速

プロトタイプの動作例



本プロトタイプの3つの特徴

1 1クリックで使える

- ダッシュボード画面のKPIをクリックするだけで、内部的にプロンプトが生成され、改善提案が提示される
- ユーザーはキーボード入力不要

プロンプト自動生成例

| | 装置A | 装置B | 装置C |
|------|-----|-----|-----|
| KPI① | *** | *** | *** |
| KPI② | *** | *** | *** |
| KPI③ | *** | *** | *** |

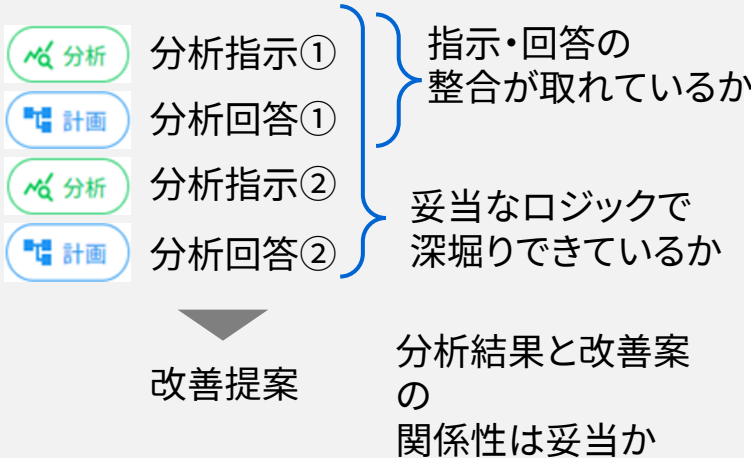
- ①ダッシュボードの任意の箇所をクリック
- ②「{装置名}の{KPI名}を調査してください」

エージェントオーケストレーターに自動入力

2 AIの判断根拠を可視化

- エージェント間の会話をチャット形式で表示し、分析過程を可視化
- AIの回答の妥当性判断に有用

妥当性判断例



3 東芝の製造ノウハウを活用した改善提案

- SMTラインに特化した製造ノウハウを参照し、一般的なLLMよりも具体性や、精度を向上

一般的なLLMの回答例

- 「日次、週次、月次でのメンテナンスをしてください」
⇒具体性が無く、実際にどのような作業をすればよいか分からない。

本デモの回答例

- 「日次で〇〇を確認してください。確認方法は、まず装置の××を開け・・・」
⇒ノウハウ集や装置マニュアルを参照し具体的な指示に落とし込み



「次戦に備えて日本が採るべき戦略とは」

デジタル庁創設で出遅れたデジタル化を急ぐ日本政府、DXを果敢に叫び、コロナ禍を生き延びようとする日本企業。

日本が抱えてきた課題を一気に顕在化せしめた新型コロナは、容赦ない変革を日本企業に突きつけています。一方、GAFAと呼ばれる米大手IT企業は、コロナ禍を追い風に変え、いっそう成長を加速させています。

GAFAに覇権を握られ、生きる道を失ったかに見える日本企業。一時は事業の「選択と集中」を徹底できない日本が、叩かれていた時代もありました。しかし、日本には技術や人の多様性が残りました。これこそが今から始まる二回戦で大きな武器と変わるのです。

GAFAが寡占したかに見えるデータは実は一部。活用されていないデータの多くが、現実世界に眠っています。そしてハードウェアに強い日本こそ、この貴重なデータを握っているのです。あとはどう活用するか次第。

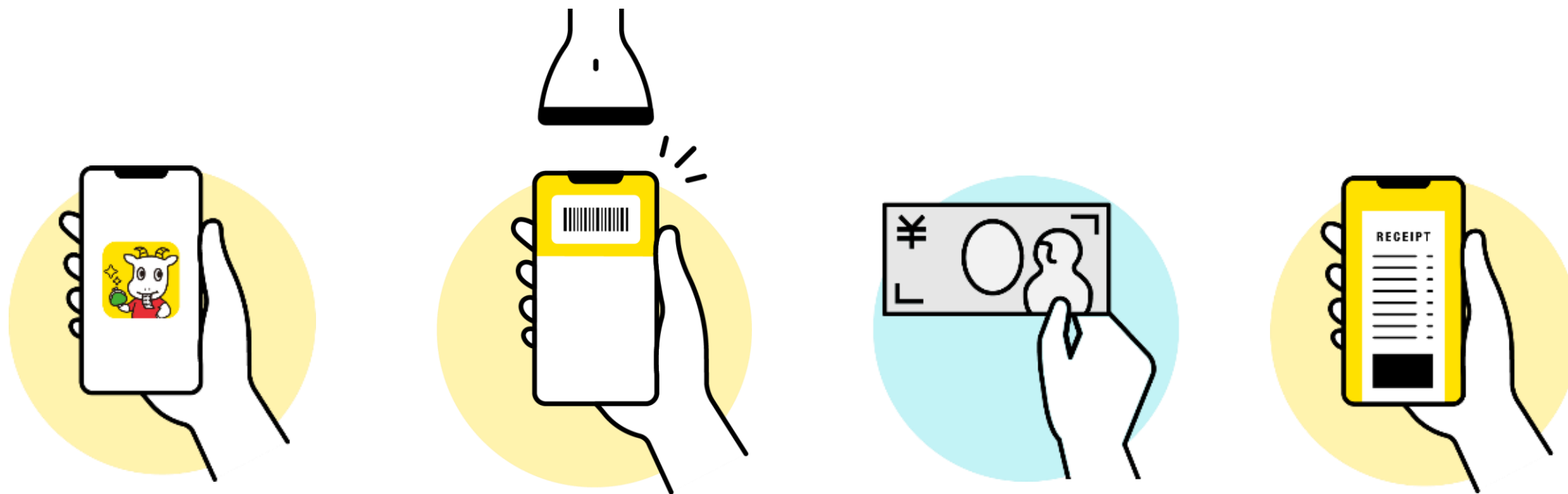
本書で紹介する「スケールフリーネットワーク」は、20世紀後半に発見された新しいネットワーク理論です。この理論をビジネスに応用することで、イノベーションを起こす土壌を作り出すことができます。

著者は東芝執行役上席常務・最高デジタル責任者の島田太郎氏、『アフターデジタル』や『DeepTech(ディープテック)』などのベストセラー本を通じて日本が進むべき道を照らし続けるフューチャリストの尾原和啓氏。

スケールフリーネットワークのビジネスへの応用を丁寧に解説したビジネス書です。

電子レシートがスマートフォンに届くレシート管理アプリ

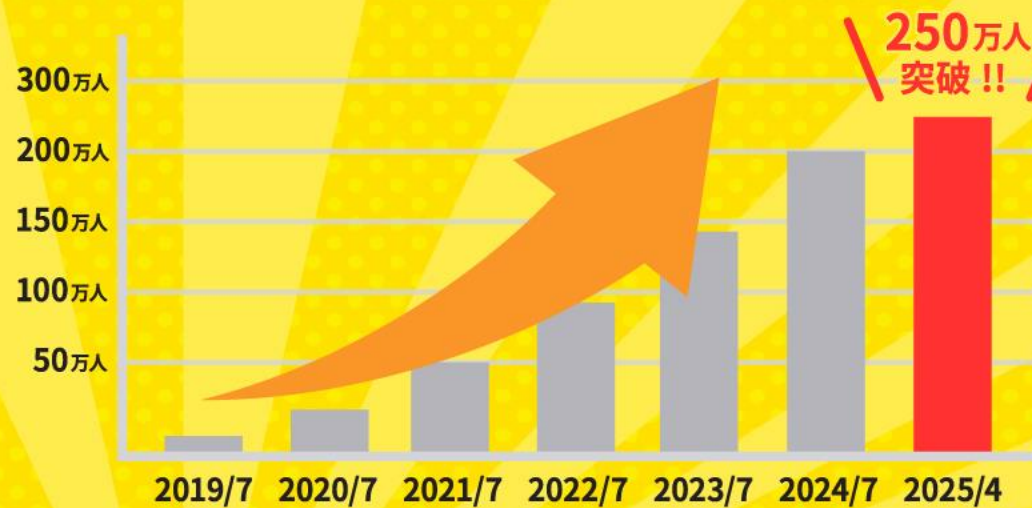
お客さま アプリをダウンロード
して会員登録 ▶ お客さま アプリを立ち上げ
バーコードを提示 ▶ お店 いつもどおりに
お会計 ▶ お客さま スマートフォンで
レシート確認



2025年4月6日に会員数250万人を突破！

スマートレシート

会員数
250万人
突破！



2024/10/1～マーケティングソリューション領域で協業を開始



スマートレシートの電子レシートに基づいた購買情報を活用することで、
ドコモのさまざまなマーケティングソリューションを通じたより高度なマーケティングが可能に
1億を超える「dポイントクラブ」会員のお客さまへ最適な情報を提供

レシートインフォマティクス

効率の良いレシート特化の独自AI技術で付加価値の高い購買データを提供

高齢化/人口減少

エネルギー/環境問題

人々の新しい気づきと行動を生み出し
持続可能な未来を、ともに創り続ける

地域社会の希薄化

貧困や教育格差

260万人の購買ビッグデータ

購買トレンドの発見



クラスタリングAI



大規模言語AI

カスタムできる
分析者視点で自動分類

レシートインフォマティクス技術

3つのAIで購買ビッグデータをグリーンに分析

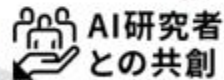


グラフ解析AI

LightGMAIC™

世の中の
先端AI技術

製造分野で培った
東芝のAI技術



AI研究者
との共創

省エネ&高効率化のための
ビッグデータ処理技術

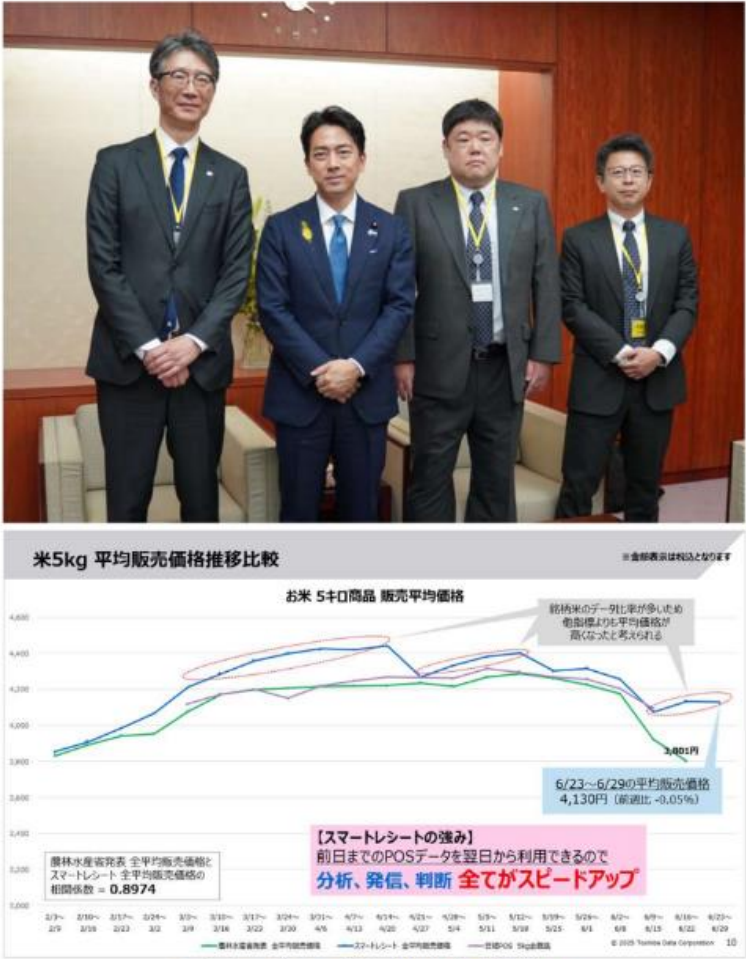
ビッグデータ プラットフォーム
「TDC DATA LAB」

農水大臣にスマートレシートから見えるコメの販売値段を説明

今後政策への反映の為に協業を進める

【小泉進次郎 農林水産大臣 Xより】

| | |
|-----------------|---|
| 日時
場所
訪問者 | 2025年7月1日 16時00分～16時45分
農林水産省 3階 大臣室
島田CEO、(TDC) 北川P、篠田EX |
| 目的 | 昨今の米価格上昇に関して、
データから見えることについてのディスカッション |
| サマリー | <ul style="list-style-type: none">米の販売価格と上昇の仕方には地域差がある備蓄米販売により価格は安定、
購買量も昨年並みまで回復米高騰において、短期間での米離れは見られない |
| PRポイント | <ul style="list-style-type: none">データをリアルタイムに地域別に見ることができる消費者起点であるため、個人の購買行動の分析に活用できる農水省の戦略立案に向けたデータとして活用できる今回の米高騰以外（野菜、魚、肉、卵など）でも活用できる |



スマートレシートの強みである、リアルタイム性をPR

スマートレシートデータを使ったAIエージェントを作成



shimada taro(島田 太郎) さんが作成

Copilot へメッセージを送る



 近くのお得情報を教えて
現在地からおすすめの店舗やサービス
を教えてください。

 普段買うものから提案して
私の買い物履歴をもとにおすすめ
商品を提案してください。

 東芝製品のおすすめは？
東芝関連の商品やサービスを優先
して提案してください。

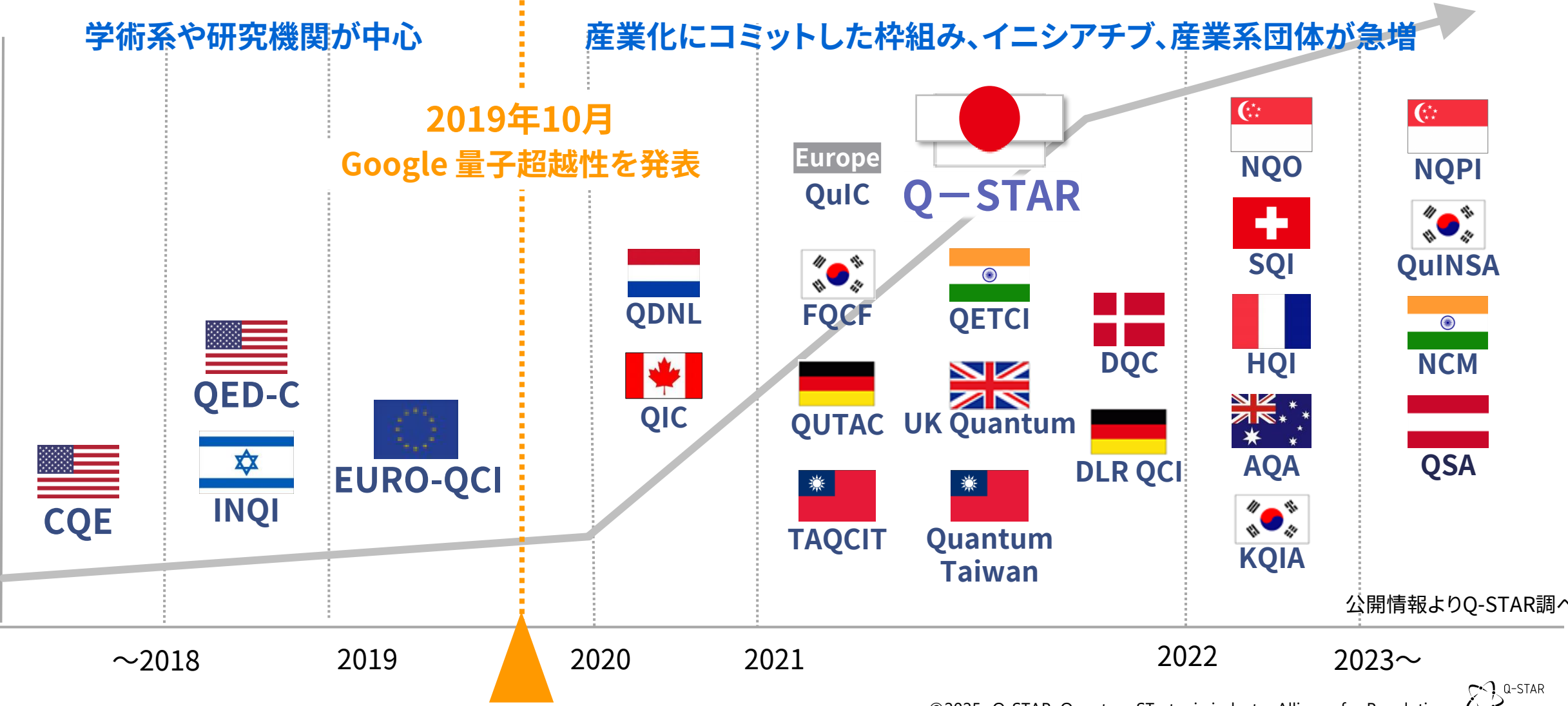
誰でもプログラムできる時代

アーキテクチャー開発か、データを取得するか、
素晴らしいアイデアを出すか



世界で増加する量子関連団体・枠組み

2019年Googleによる量子超越性の発表を契機に世界各地で量子の産業化やエコシステム強化を目指す団体や枠組みが誕生。日本は2021年にQ-STARを設立し産業化競争に参入。



グローバル標準及びグローバルサプライチェーンの主導権確立へ向けて

Q-STARが掲げる3本の柱

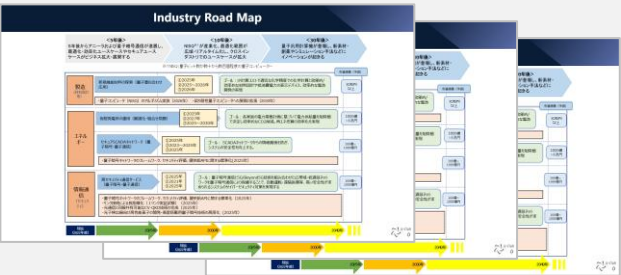
ユースケース

各部会においてベンダー企業とユーザー企業が一体となったユースケース議論と検証

多数のユースケース議論



議論の中から厳選し産業ロードマップ化



テストベッド

G-QuATとのテストベッド連携による実機環境でのユースケース実証で早期社会実装に向け取り組む



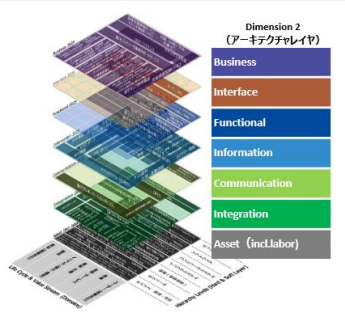
出典: <https://unit.aist.go.jp/g-quat/index.html>

標準化

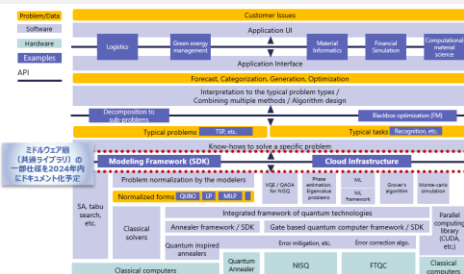
量子技術をビジネスへとつなげるフレームワーク

QRAMI

社会実装へ向けた全体像を俯瞰 関係者間での認識の共有に寄与



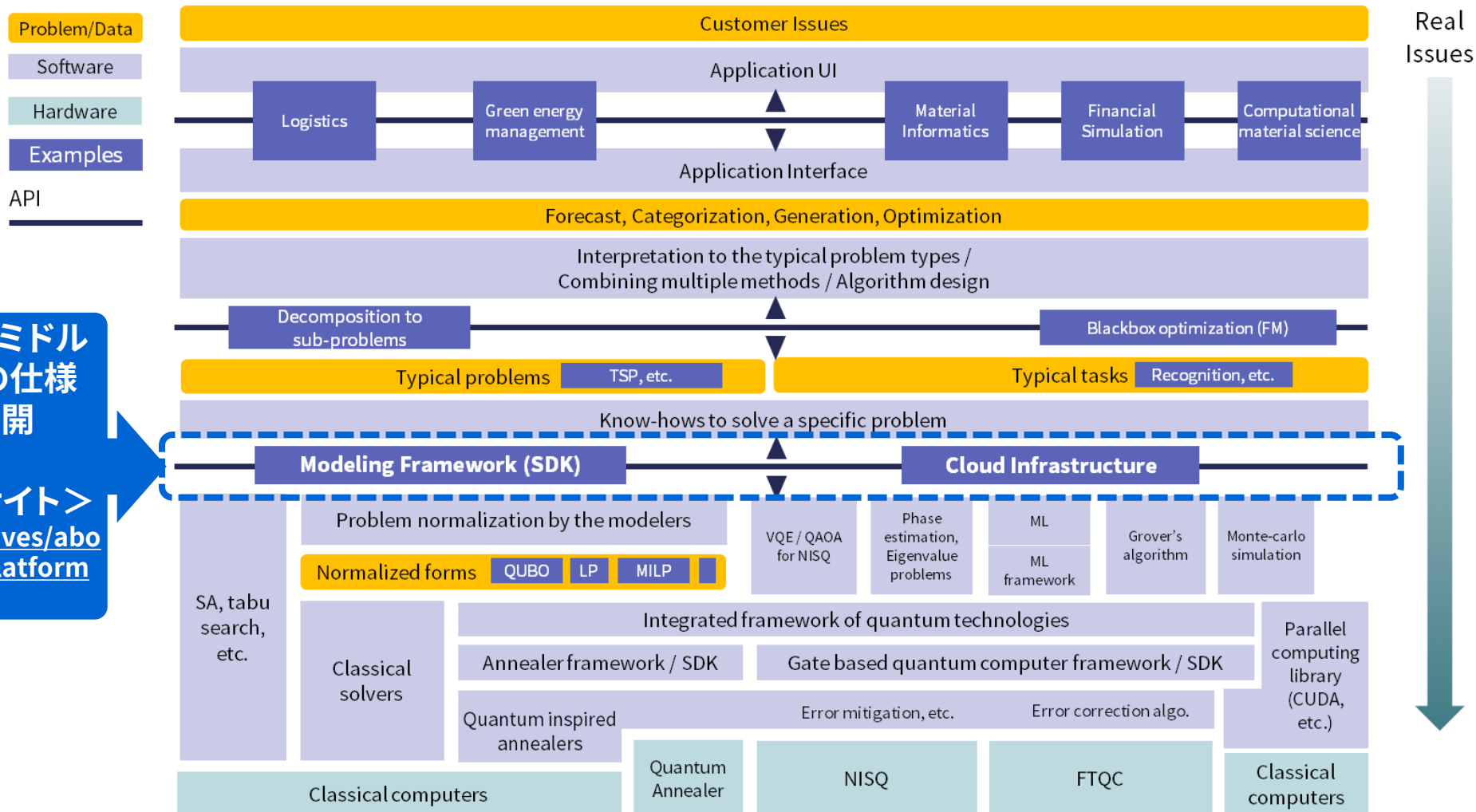
ソフトウェアスタック図



社会課題を解決するための量子技術活用をシステムティックに検討するツール
ユースケース探索等に寄与

量子コンピューティング領域のソフトウェアスタック図

社会課題を解決するための、量子技術活用をシステムティックに検討
(量子インスパイアード技術の社会実装を根子に、グローバルでのエコシステムの醸成に貢献します)



最適化問題を解くミドルウェア (QODE*) の仕様書をグローバル公開

<国内向け公開サイト>
https://qstar.jp/archives/about_news/software_platform_20250529

量子時代に大切なのは、
専門家ではなくエンジニア

TOSHIBA

私たちの存在意義

新しい未来を始動させる。

We turn on the promise of a new day.