HITACHI
Inspire the Next

SPI Japan 2023

設計ドキュメントから抽出した品質確保観点の活用による ソフトウェア品質の向上

テキストマイニング技術を活用した設計ドキュメント分析

2023/10/11

株式会社 日立製作所 デジタルシステム＆サービス
デジタルプラットフォーム事業部 ミドルウェア本部 生産技術部

内田智也(発表者)、香西周作

HITACHI
Inspire the Next

自己紹介

内田　智也 （うちだ ともや）

- 所属：
  （株）日立製作所 デジタルプラットフォーム事業部
  ミドルウェア本部　生産技術部

  ※設立より一貫して社会インフラの基盤に使用される基幹ソフトウェア製品
  （メインフレーム用OS、DB、アプリケーションサーバ、運用管理製品等）
  を開発・保守している事業部

- 経歴：
  - 分散トランザクション基盤製品の開発・保守
  - Enterprise Service Bus製品の開発・保守
  - アプリケーションサーバ製品の開発・保守
  - ソフトウェア開発環境の整備
  - 基幹製品の生産性向上・品質向上の推進

製品開発の現場経験を生かして、
SEPGで活動中。

HITACHI
Inspire the Next

Contents

HITACHI
Inspire the Next

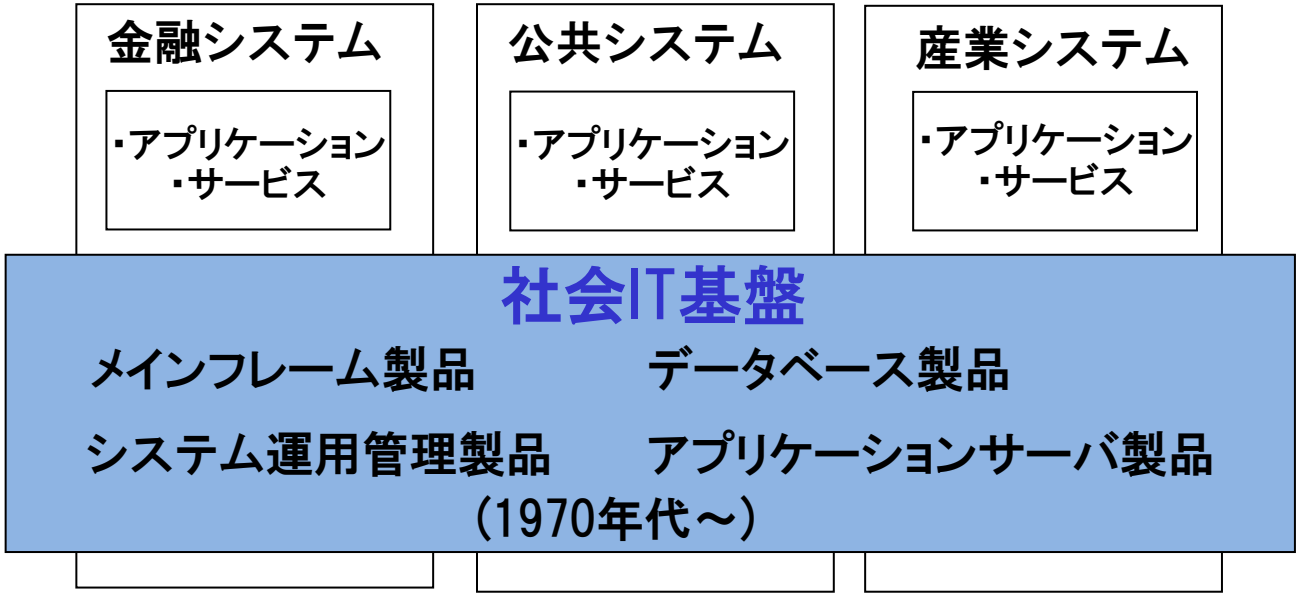
1. 課題と解決方針

1. 課題と解決方針(１)

HITACHI
Inspire the Next

背景

- 当事業部ではミッションクリティカルな社会IT基盤で使用されるソフトウェア製品の開発・保守を行っている。
- 古いもので1970年代から長年エンハンスをし続けてきている。
- 停止すると社会的にも大きな影響を与えることから、品質を維持することが重要。

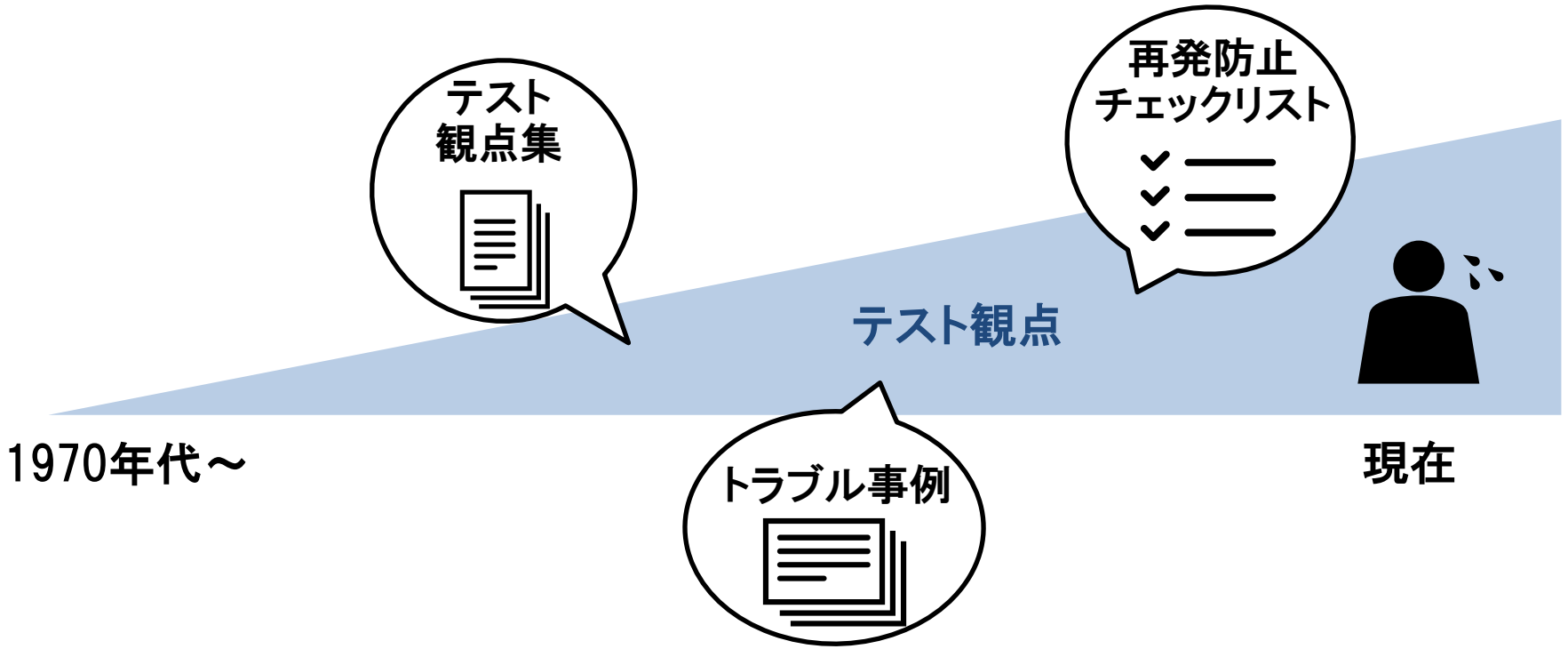


1. 課題と解決方針(２)

HITACHI
Inspire the Next

品質維持の課題

- 品質を維持するために重要なことは、過去のノウハウやトラブル事例から導き出したテスト観点を設定してテストを実施すること。
- テスト観点は熟練者や経験者の知見に依存する。
- テスト観点表は長年かけて蓄積されて、量が膨大で利用が難しい。



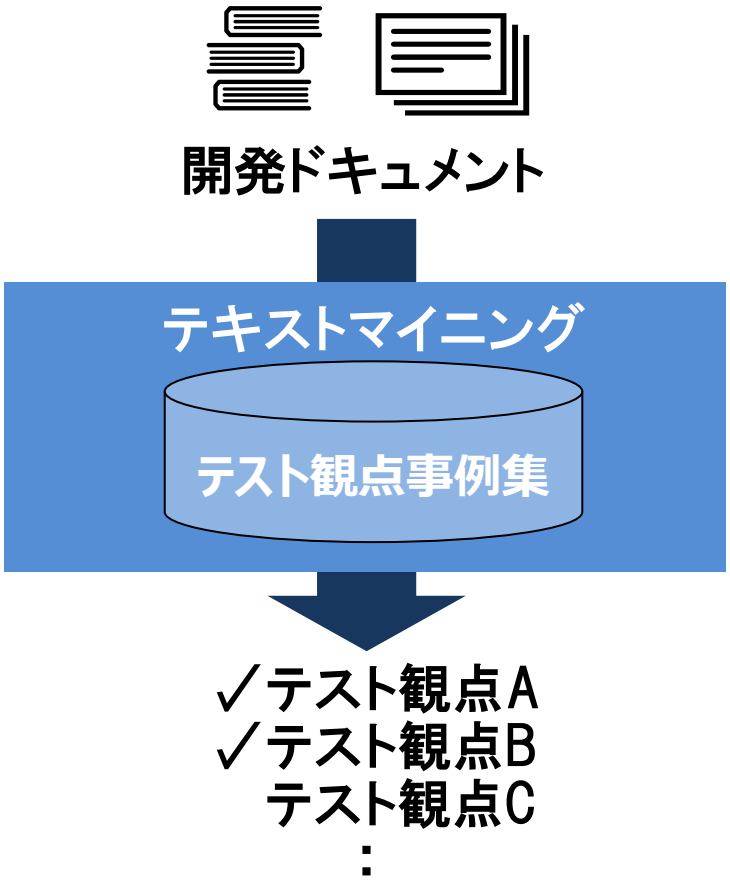
引継ぎがうまく行われず、テスト観点が漏れてしまう

1. 課題と解決方針(３)

HITACHI
Inspire the Next

課題の解決策

- テキストマイニング技術を使用して、開発時に作成されたドキュメントからテスト観点を機械的に抽出し、含まれているテスト観点をチェックする。
- 抽出する観点は事業部共通のテスト観点事例集を基に設定。



テキストマイニングとは、
文章に含まれる情報を取り出して分析する方法。

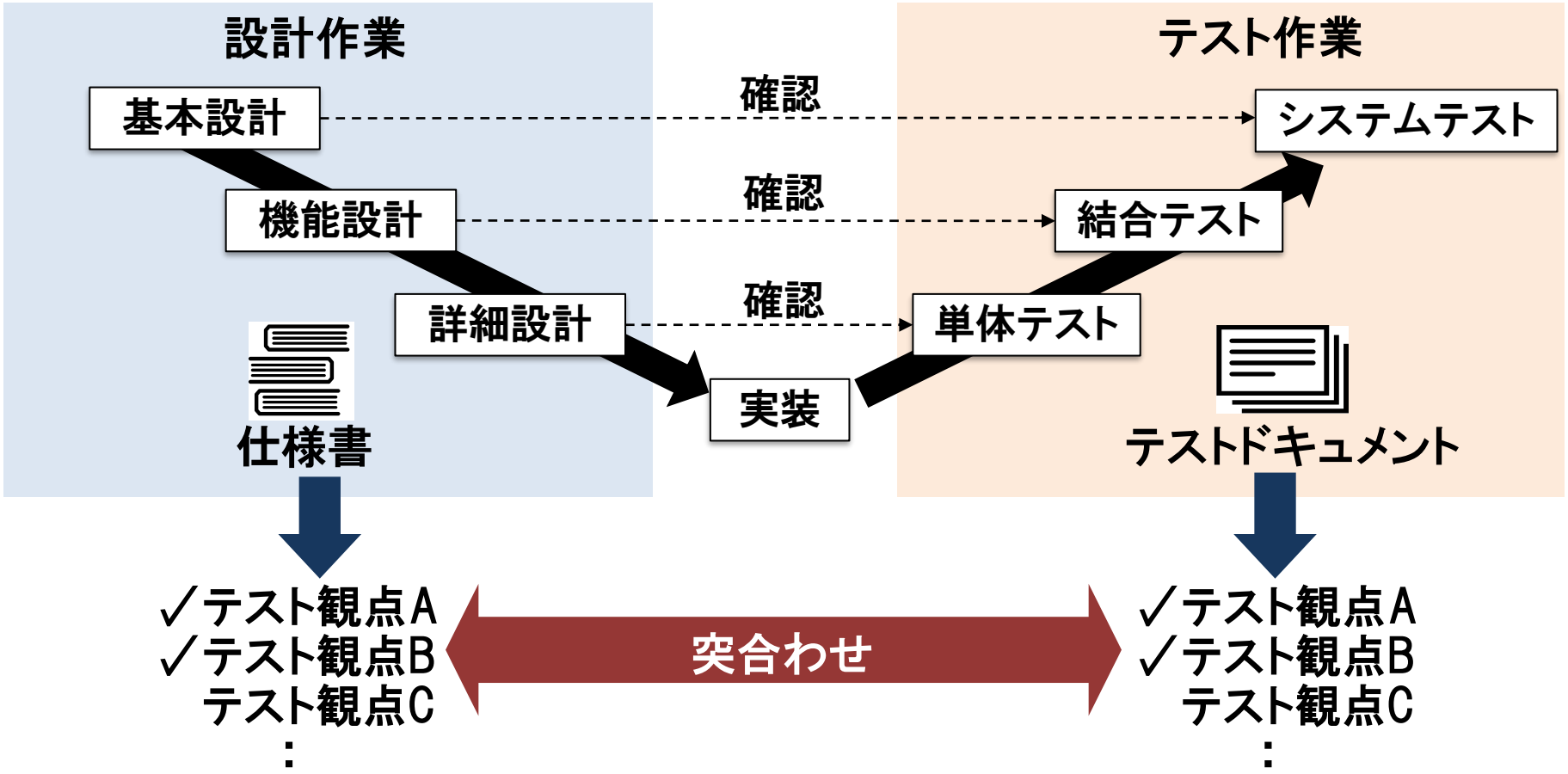
文章を用語や文節で区切り、それらの相関関係を確認したり、指定したルールに該当する内容の抽出が行える。

1. 課題と解決方針(４)

HITACHI
Inspire the Next

課題の解決策

- 仕様書およびテストドキュメントの双方から抽出したテスト観点を突き合わせることで、テスト観点の網羅性をチェック。



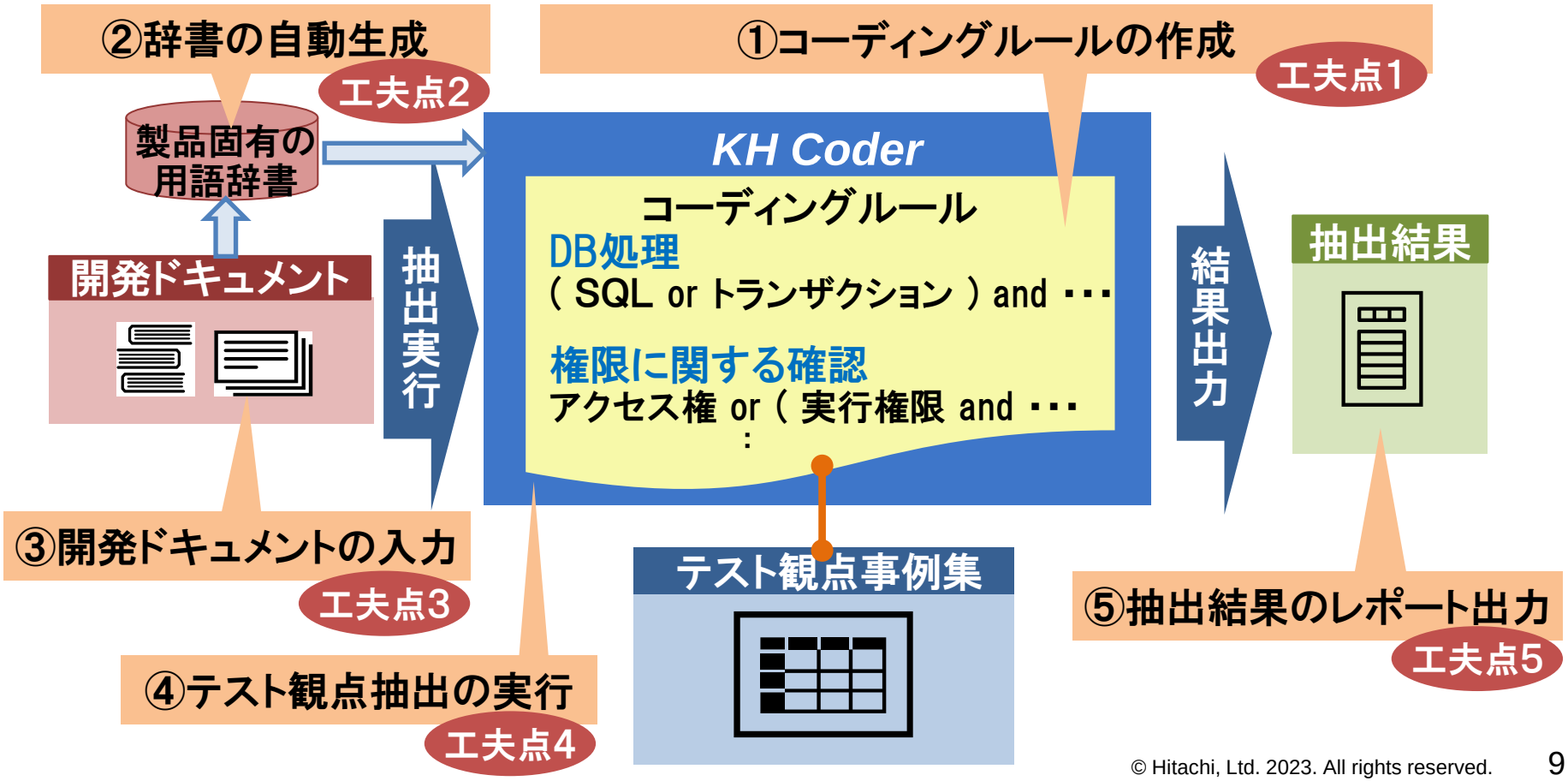
2. 施策の仕組みと工夫点

2. 施策の仕組みと工夫点(１)

HITACHI
Inspire the Next

施策の全体像

- テキストマイニングにはKH Coderというフリーソフトを使用。
- テスト観点の抽出を①～⑤の手順で実行。
- 各手順における工夫点の内容を以降のスライドで説明。



2. 施策の仕組みと工夫点(２)

HITACHI
Inspire the Next

KH Coderの概要

- テキスト型(文章型)データを統計的に分析するためのフリーソフト
- 「テキストマイニング」または「計量テキスト分析」と呼ばれる方法に対応
- URL：http://khcoder.net/

KH Coder

多変量解析による データ要約の機能

- 対応分析
- クラスター分析
- 自己組織化マップ
- 共企ネットワーク　等

作成した基準に従い 言葉を分類する機能

- コーディングルールを 扱う機能 等

データ検索の機能

- KWICコンコーダンス 等

※参考文献[2]を参考に記載

2. 施策の仕組みと工夫点(３)

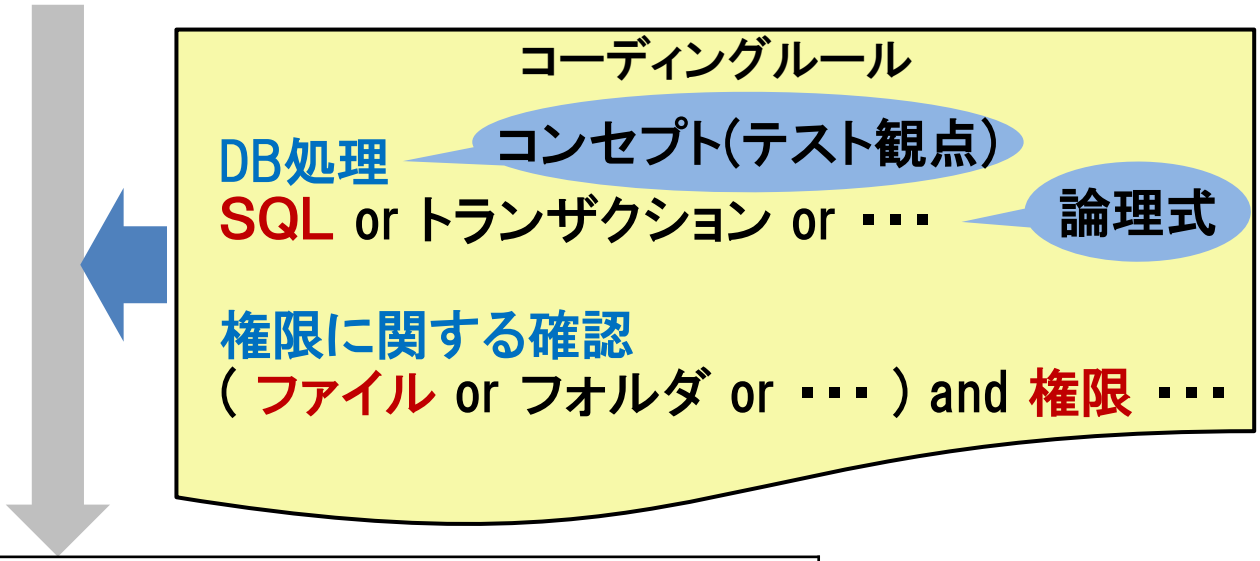
HITACHI
Inspire the Next

KH Coderのコーディングルール

- コーディングルールとは、注目したい独自のコンセプトを取り出すためのルール。
- コーディングルールを作成してコンセプト(テスト観点)を抽出する。
- テスト観点との関連が分かり易いテストドキュメントからのテスト観点抽出を通してルールを整備

文1:SQL/を/実行/して/結果/を/確認/する。
文2:ファイル/の/編集/には/管理者/の/権限/を/必要/とする。

形態素解析



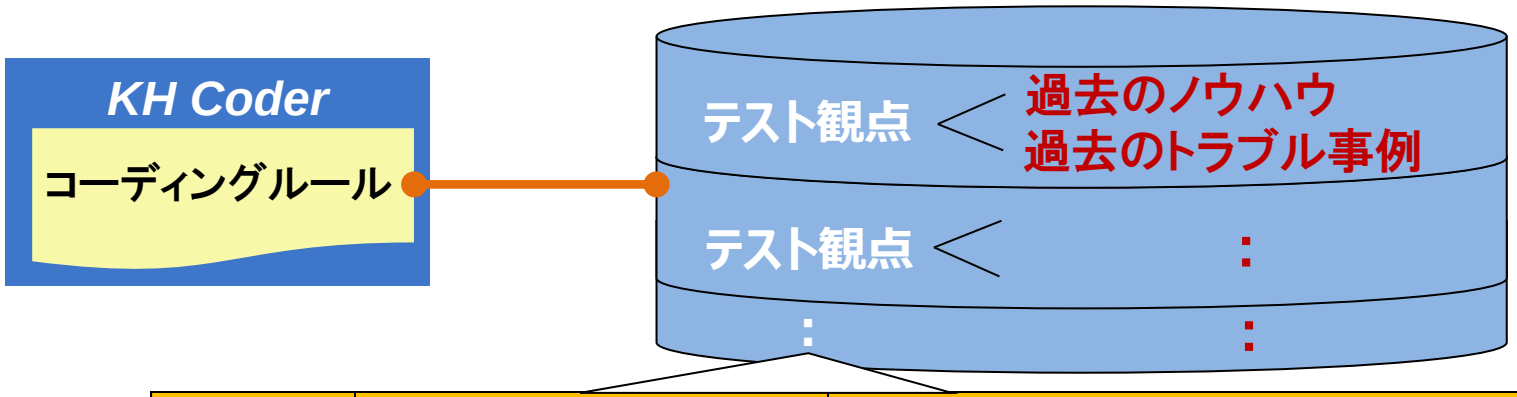
文1	DB処理のテスト観点あり
文2	権限に関する確認のテスト観点あり

2. 施策の仕組みと工夫点(４)

HITACHI
Inspire the Next

工夫点1　過去ノウハウ・トラブル事例も含めたコーディングルール作成

- 検査部門が纏めている**事業部共通のテスト観点事例集**を基にしてコーディングルールを作成。
- テスト観点だけでなく**過去ノウハウやトラブル事例**から抽出した用語もコーディングルールに含めることで、幅広い用語をカバー。



大分類	中分類	テスト観点
：	：	：
リソース管理	資源効率性(CPU・メモリ・ディスク等)	CPU利用率が妥当であることを確認すること
		無駄に領域を使用しすぎていないかを確認すること
		メモリ使用量が妥当であるかを確認すること
	資源のリーク	ファイル操作時に不当に資源が増加しないこと。
		OS資源を使用する場合は、生成・消滅のタイミングを十分に検証したか。
		処理の順番が保証されているかを確認すること
	自プログラム製品の資源利用状況確認	プロセスIDを保持して他製品のプロセスをkillしないことを確認すること
		ポート番号が重複しないことを確認すること
タイミング処理	局所的なタイミングの確認(システム全体)	システム全体での局所的なタイミングを考慮した項目を設定すること
		処理中にOSのシャットダウンやサービス停止要求がきても異常にならない
	局所的なタイミングの確認(GUI操作)	複数同時に画面描画を行うテストを実施すること。
	局所的なタイミングの確認(起動・停止)	GUI画面の起動・停止が重なる場合の項目を設定すること
：	：	：

2. 施策の仕組みと工夫点(５)

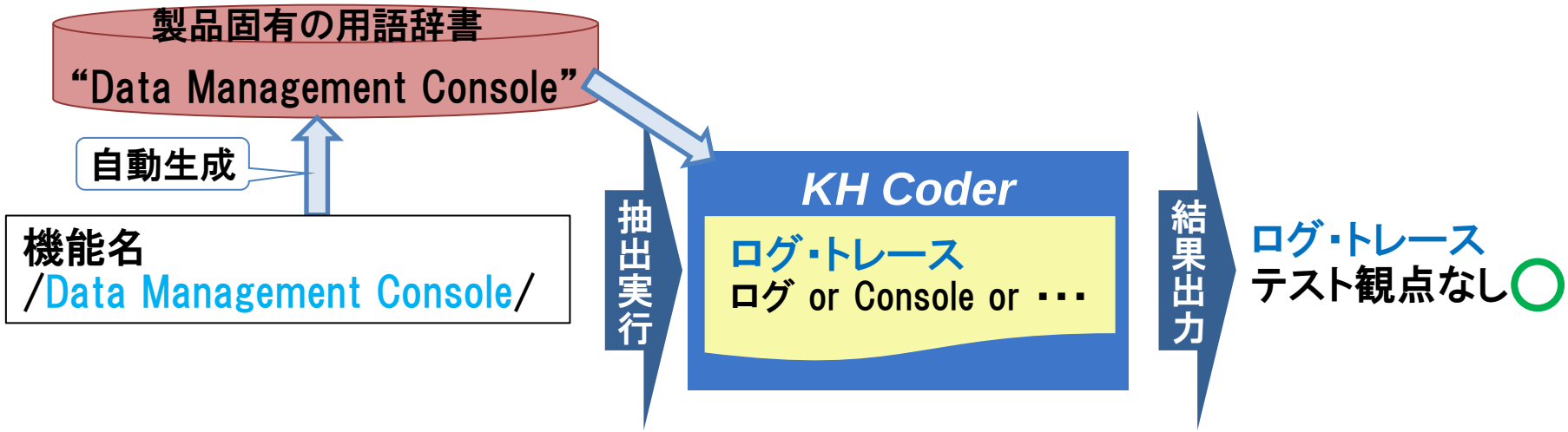
HITACHI
Inspire the Next

工夫点2 製品固有の用語辞書の自動生成

- 抽出対象の開発ドキュメントから製品固有の用語辞書を自動生成して KH Coderに登録し、誤検知を防止。

➢ 用語辞書なし

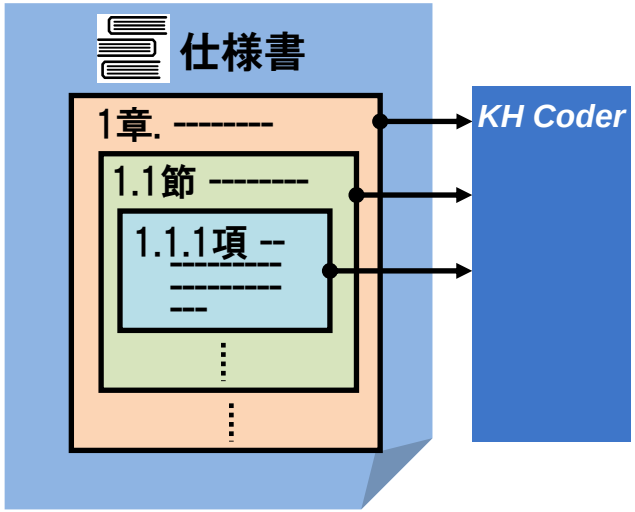
➢ 用語辞書あり

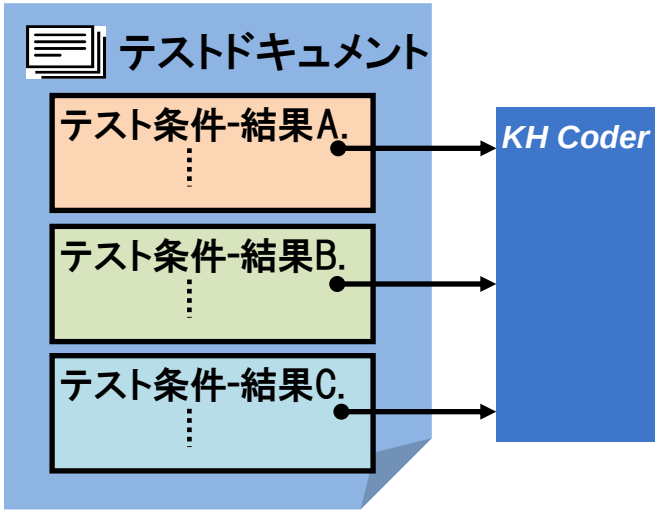


HITACHI
Inspire the Next

工夫点3 文書の細分化

- 観点の抽出元を特定できるように、抽出対象の開発ドキュメントの文章の細分化を行う。
- 仕様書は章/節/項の単位、テストドキュメントはテスト条件-結果の単位に、それぞれ分割してKH Coderに入力する。





2. 施策の仕組みと工夫点(7)

HITACHI
Inspire the Next

工夫点4 文書に合わせたコーディングルールの適用設定

- 誤った抽出を防ぐため、文書に合わせてコーディングルールの適用設定をする。
- 仕様書は文(文章を構成する各文)の単位、テストドキュメントはテスト条件-結果の単位に適用して、関係のある内容から正しく抽出されるようにする。

資源リーク
(メモリ and 解放)

資源状況確認
(メモリ and 所要量)

仕様書

1. ○○○機能
このプロセスはメモリの状態を管理する。・・・。ファイルの更新が完了すると排他制御によりロックを解放する。・・・

✕テスト観点あり(誤検知)

1. ○○○機能
このプロセスはメモリの状態を管理する。・・・。ファイルの更新が完了すると排他制御によりロックを解放する。・・・

○テスト観点なし

テストドキュメント

○○○機能
テスト条件:実行時のメモリの状態を表示
テスト結果:所要量はXバイト

✕テスト観点なし(未検知)

○○○機能
テスト条件:実行時のメモリの状態を表示
テスト結果:所要量はXバイト

○テスト観点あり

2. 施策の仕組みと工夫点(８)

HITACHI
Inspire the Next

工夫点5 抽出結果を製品開発者へ直接説明

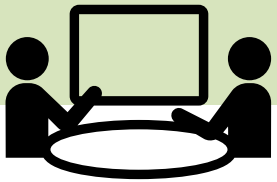
- 結果の有効活用・施策の定着化のため、抽出結果を製品開発者へ直接説明をする。
- 主な説明内容は、テスト観点の全体傾向、突き合わせの結果、不足している観点、前バージョンの抽出結果との比較など。

主な説明内容

製品A　テスト観点抽出結果

観点	開発項目
…	…
＊008-インタフェース	10.0%
＊009-開始__終了	0.0%
＊010-機能確認	80.0%
＊011-処理方式	0.0%
＊012-限界・境界条件	40.0%
＊013-異常処理	30.0%
＊014-適用範囲内の詳細確認	0.0%
＊015-DB処理	20.0%
＊016-権限に関する確認	40.0%
＊017-国際化プログラミング	0.0%
＊018-大量データの入出力	0.0%
…	…

1. テスト観点の全体傾向
2. 仕様書とテストドキュメントの観点突き合わせ結果
3. 不足している観点
4. 前回バージョンの抽出結果との比較



2. 施策の仕組みと工夫点(９)

HITACHI
Inspire the Next

テスト観点抽出結果のレポート出力例

テスト観点の項目

テスト観点の抽出結果

大分類	チェック観点(中分類)	開発項目 TID000	TID001	TID002	TID003	TID004	TID005	TID
前提条件	＊001-前提条件	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
	＊002-サポートソフトウエア	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
	＊003-サポートハードウエア	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
	＊004-顧客ニーズ	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
	＊005-顧客環境の考慮	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
マニュアル・ヘルプ	＊006-マニュアル	0.0%					0.0%	
	＊007-ガイド・ヘルプ	0.0%					0.0%	
基本機能	＊008-基本機能(インタフェース)	0.0%	0.0%	13.3%	2.8%	0.0%	0.0%	
	＊009-基本機能(開始_終了)	0.0%	0.0%	0.0%	1.4%	0.0%	0.0%	
	＊010-基本機能(機能確認)	0.0%	100.0%	20.0%	90.1%	30.0%	44.4%	
	＊011-基本機能(処理方式)	0.0%	0.0%	0.0%	2.8%	0.0%	0.0%	
	＊012-基本機能(限界・境界条件)	0.0%	0.0%	13.3%	2.8%	0.0%	0.0%	
	＊013-基本機能(異常処理)	4.8%	0.0%	6.7%	4.2%	20.0%	13.9%	
	＊014-基本機能(適用範囲内の詳細確認)	0.0%	0.0%	20.0%	2.8%	0.0%	0.0%	
	＊015-基本機能(DB処理)	0.0%	66.7%	13.3%	62.0%	30.0%	30.6%	
	＊016-基本機能(権限に関する確認)	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
	＊017-基本機能(国際化プログラミング)	0.0%				0.0%	0.0%	
	＊018-基本機能(大量データの入出力)	0.0%				0.0%	0.0%	
	＊019-基本機能(その他)	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
入出力機能	＊020-入出力機能	0.0%	0.0%	13.3%	4.2%	20.0%	52.8%	
	＊021-入力容易性	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
	＊022-メッセージ表示(正確性、分かり易さ等)	0.0%	0.0%	0.0%	0.0%	10.0%	38.9%	
	＊023-画面表示(理解容易性等)	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
	＊024-仕様・表示の統一性	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
	＊025-ログ・トレース	4.8%	0.0%	33.3%	0.0%	30.0%	11.1%	
障害	＊026-障害・異常(障害復旧時間)	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
	＊027-障害許容性(Fault_Tolerance)	4.8%	0.0%	0.0%	1.4%	0.0%	13.9%	
	＊028-部分的一時的障害への対応							

テスト観点あり

テスト観点なし

2. 施策の仕組みと工夫点(10)

HITACHI
Inspire the Next

観点の抽出精度

- 機械抽出を最初に実行して抽出される精度は90%以上であり、残りは対策を行い機械抽出を再実行して100%にする。
- 対策内容
  - 誤検知防止のため製品固有の用語辞書を追加
  - 必要に応じてコーディングルール自体の見直しも実施
    (誤検知の要因となる用語の削除、同義語や類義語等の追加、
    論理式の見直し等)

機械抽出の最初の実行結果

		仕様書	テストドキュメント
抽出する 全テスト観点数 76個	正しく抽出できた	71個 (93.4%)	69個 (90.8%)
	正しく抽出できなかった	5個 (6.6%)	7個 (9.2%)

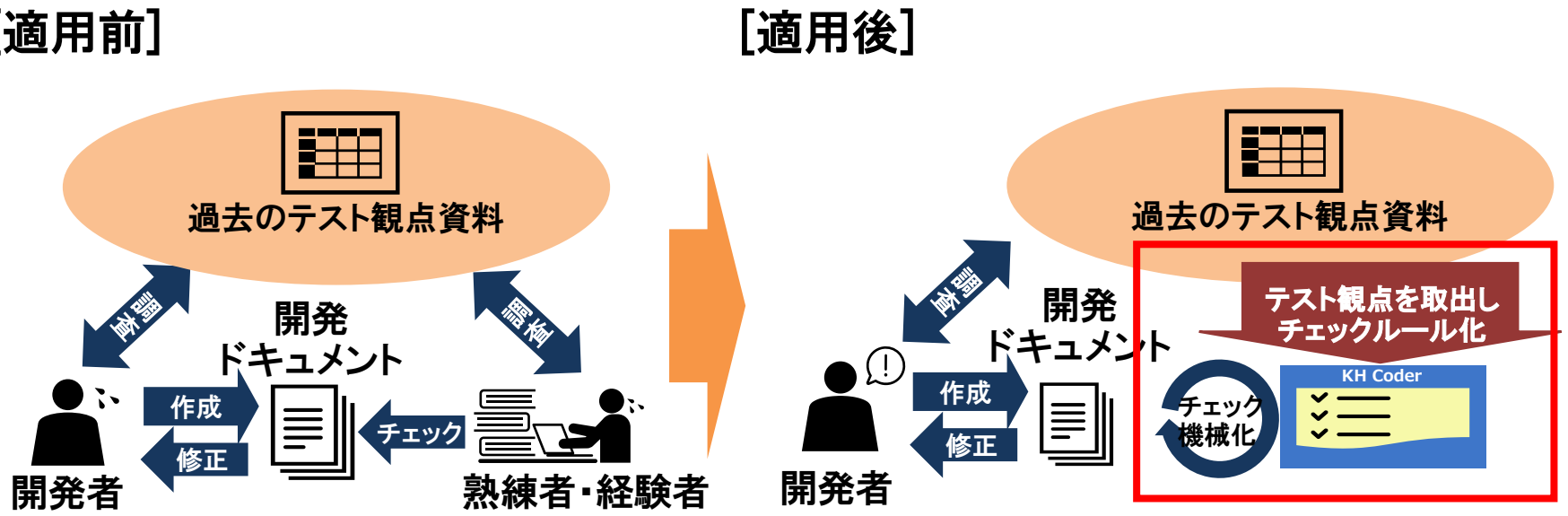
対策を行い、機械抽出を再実行して100%にする

3. 適用事例と効果

3. 適用事例と効果(１)

適用事例

- 事業部の主要製品に施策適用を実施。
- 適用前は、過去のテスト観点資料を参考にしつつ、開発者が開発ドキュメントを作成して、熟練者・経験者がチェック。
- 適用後は、作成した開発ドキュメントを機械的にチェックして、必要なテスト観点が網羅されているかを確認。



3. 適用事例と効果(２)

HITACHI
Inspire the Next

適用事例

- 適用した製品の開発ドキュメントに対して**テスト観点漏れのチェック**を実施。
- 仕様書とテストドキュメントのテスト観点抽出結果の**突合せでテスト観点漏れを発見**。

製品	アプリケーションサーバ製品A　バージョンX
開発項目	4個
開発ドキュメント	機能仕様書（Wordファイル：約70ページ）
	結合テスト項目（テスト条件-結果 約140個）
適用方法	・開発ドキュメントから、開発バージョンの内容だけを取り出し。 (前回バージョンとの差分を取得) ・取り出した開発バージョンの内容を、4つの開発項目に分類。

適用結果

開発ドキュメントとテストドキュメントのテスト観点の突合せを実施し、
開発項目の1つで、**2件のテスト観点漏れを発見**。

- **製品の一部で障害が発生した場合の対処に関する確認**
- **同一資源への操作を繰り返した場合の動作確認**

開発者にフィードバックして対策を実施。

3. 適用事例と効果(３)

HITACHI
Inspire the Next

施策の適用効果

’21年度/上期～’23年度/上期までに、施策適用した主要製品全体で、

テスト観点漏れ発見　1製品あたり平均2.3件

製品開発者からの声

開発者

作成した開発ドキュメントの内容がどういうテスト観点に分類されるか、**客観的に分かった**のが良かった。

具体的なテスト観点が整理されていて、テスト観点に**抜けがないか確認するのに有用**であると感じた。

開発者

3. 適用事例と効果(４)

HITACHI
Inspire the Next

開発者からの要望・意見

- 新たなテスト漏れ事例の検出の要望、および不足しているテスト観点の指摘に関する意見があった。

開発者

テスト漏れの具体事例があり、それを施策で検出できるようにしてほしい。

⇒具体事例を調査して必要な観点項目(コーディングルール)を追加。

開発者

事業部全体の基準に沿った画一的な指摘だけでなく、開発内容を加味したチェックをしてほしい

⇒開発ドキュメントを元に開発内容を考慮して必要なテスト観点を選択・提示するように改善。将来的に、この結果を蓄積して必要なテスト観点の選択・提示を自動化できるようにする

＜製品Xの抽出結果＞
「2038年問題対応」テスト観点B、テスト観点C

＜製品Yの抽出結果＞
「2038年問題対応」テスト観点A、テスト観点B

製品共通で必要なテスト観点を蓄積

「2038年問題対応」

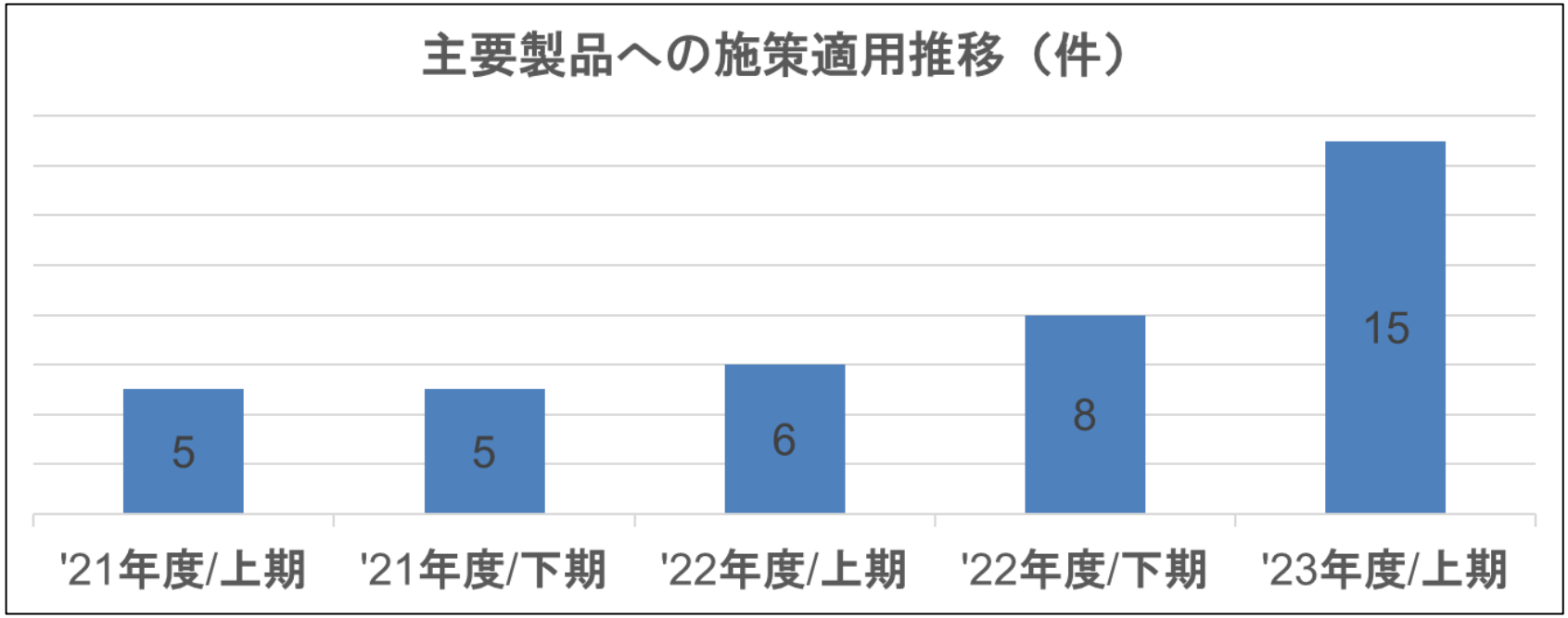
テスト観点	要否
A	不要
B	必要
C	不要

3. 適用事例と効果(５)

HITACHI
Inspire the Next

施策適用した製品件数

- 事業部の主要製品に'21年度/上期から施策適用を開始。
- 1度適用したプロジェクトの継続適用があり、新規適用するプロジェクトも増えている。
- 製品開発者と連携して施策内容を充実させて件数を増やしていく。



HITACHI
Inspire the Next

4. まとめ

4. まとめ

HITACHI
Inspire the Next

取り組みの成果

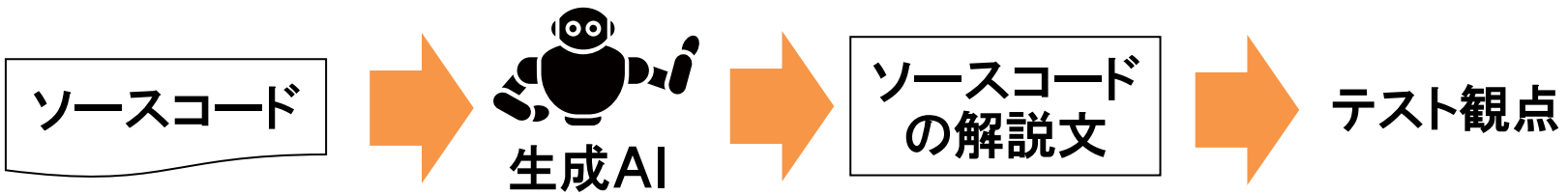
- 実際の製品開発に施策を適用して、課題に対する効果が得られた。
- 開発者からのフィードバックをもらい、今後の予定に繋がった。
- 適用製品の件数が継続・増加しており、ニーズが高いことが分かった。

今後の方針

今後も熟練者や経験者の退職・事業シフトが進むため、継続して引継ぎの課題に取り組んでいく必要がある。

今回の成果により、蓄積された過去の知見を抽出して機械的に活用していくことは有効であることが分かった。

今後は、カバーできていない実装作業について、生成AIにソースコードを入力して解説文を出力し、今回の仕組みで観点を抽出する。



参考文献

[1] 香西 周作、内田 智也、北川 福太郎 (2022). 設計資産分析による大規模ソフトウェアのエンハンス作業効率化～熟練者/前任者ナレッジの活用～ SPI Japan 2022
https://www.jaspic.org/event/2022/SPIJapan/session1B/1B2_004.pdf

[2] 樋口 耕一 (2020). 社会調査のための計量テキスト分析 第2版 ナカニシヤ出版

[3] 保田 勝道 (1995). ソフトウェア品質保証の考え方と実際　日科技連出版

HITACHI
Inspire the Next

END

設計ドキュメントから抽出した品質確保観点の活用による
ソフトウェア品質の向上

テキストマイニング技術を活用した設計ドキュメント分析

2023/10/11

株式会社 日立製作所 デジタルシステム＆サービス
デジタルプラットフォーム事業部 ミドルウェア本部 生産技術部

内田智也(発表者)、香西周作


Hitachi Social Innovation is
POWERING GOOD