

SPI Japan 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～**

2022/10/6

株式会社 日立製作所 サービスプラットフォーム事業本部
デジタルエンジニアリング事業部
ソフトウェア生産技術部

香西 周作(発表者)、内田 智也、北川 福太郎

自己紹介

香西 周作 (こうざい しゅうさく)

・所属:

(株)日立製作所 サービスプラットフォーム事業本部
ミドルウェア本部 ソフトウェア生産技術部

※1969年の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・保守しつづけている事業部

・経歴:

- グループウェア製品の基盤ライブラリ **エンハンス・保守**
 - 基幹ソフトウェア製品の共通インストーラ **エンハンス・保守**
 - デジタルテレビ・DVDレコーダー **エンハンス効率化支援**
 - ストレージ、サーバ製品 **エンハンス効率化支援**
 - 基幹ソフトウェア製品 **エンハンス・保守効率化施策推進**
- 自分自身が
エンハンス担当
- エンハンス作業の
効率化担当

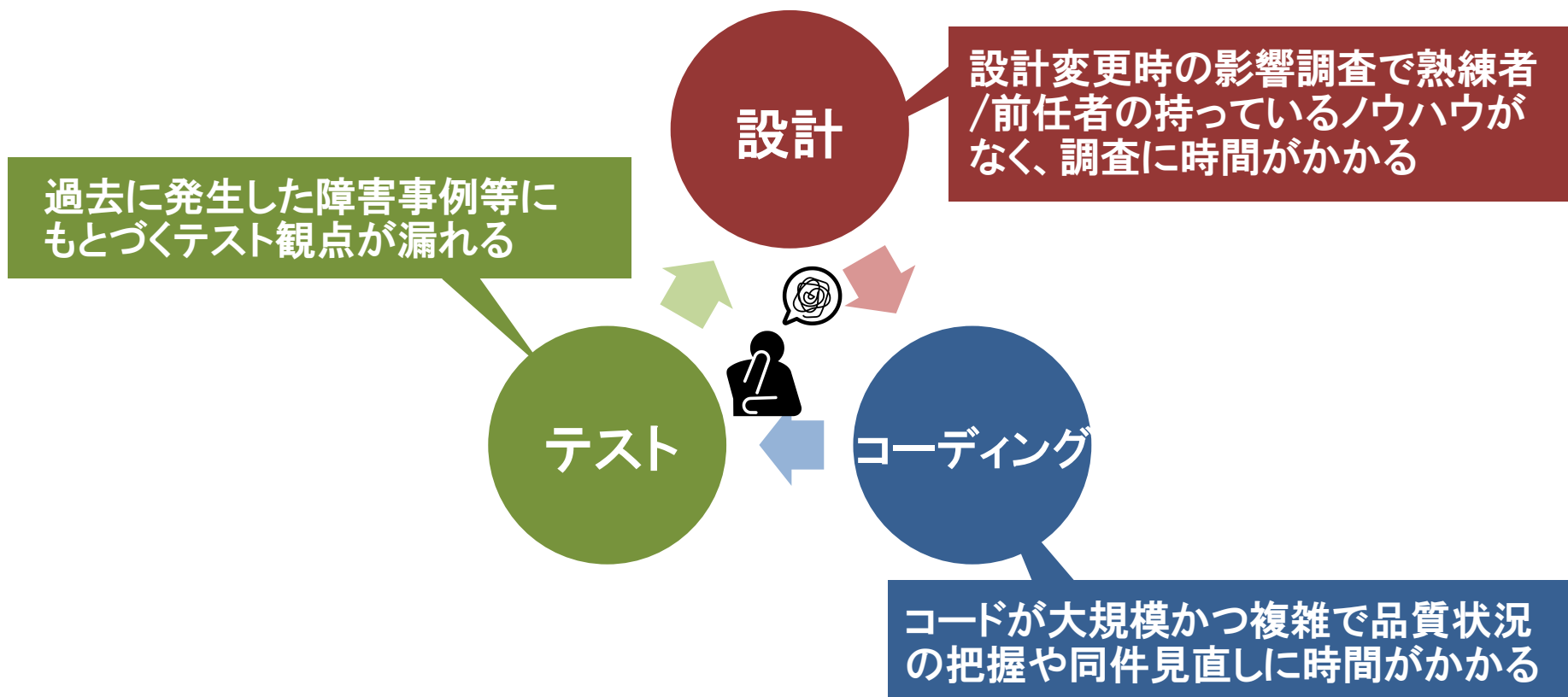
Contents

1. 大規模ソフトウェアのエンハンスにおける課題
2. ドキュメント関連度分析による設計作業効率化
3. 適用事例: マニュアル開発への適用
4. まとめ

1. 大規模ソフトウェアのエンハンスにおける課題

背景

- 前任者不在の状況で長年開発してきたソフトウェアのエンハンスを行う場面が増加(熟練者の退職、事業シフトにともなう異動等)
- 引き継いだ担当者は前提知識やノウハウが不足しており作業効率の悪化や、品質の低下を招く恐れがある



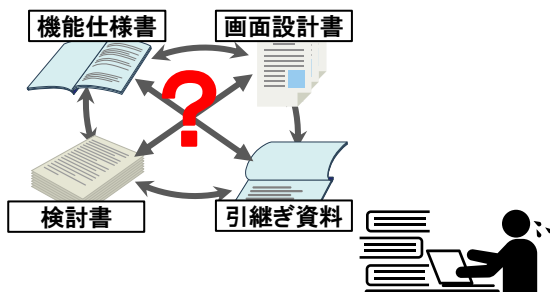
エンハンス担当者の課題・悩み

- 過去の開発で作成してきた大量の設計資料やコードを**ノウハウ不足**の状態
で調査する必要がある
 - ▶ **作業効率の悪化要因**
- 作業の前提となる知識を把握しきれず**必要な情報を見落とす**
 - ▶ **品質の低下要因**



設計

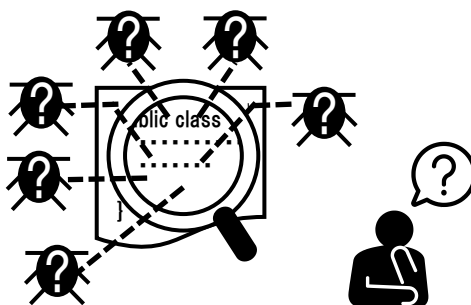
長年の開発で作成してきた設計資料を元に仕様を調査



悩み: 設計資料の量が多く、調査に時間がかかる

コーディング

品質状況把握のためコードチェックツール実行



悩み: ツールの指摘が膨大で重要な指摘が埋もれる

テスト

テスト観点集や障害事例等にもとづいてテスト項目を作成



悩み: 前提知識が多く、必要なテスト項目が漏れる

解決方針

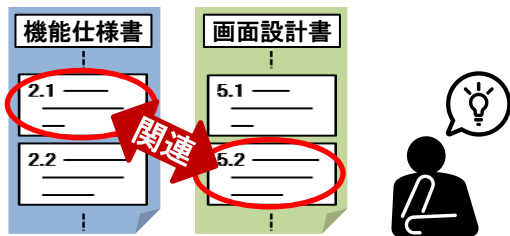
課題の本質的な要因は、引き継いだ担当者が「**前提知識、ノウハウの不足した状態で大量の設計資料やコードを対象に作業している**」ことにある

前任者のノウハウを過去の設計資産から抽出し機械的に活用できる仕掛けを整備する

設計

長年の開発で作成してきた設計資料を元に仕様を調査

設計資料間の関連度を分析して提示

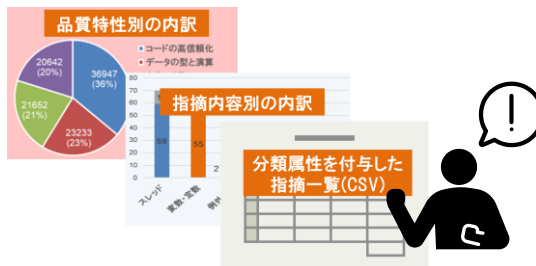


(調査作業の効率化)

コーディング

品質状況把握のためコードチェックツール実行

コードチェックツールの結果を品質視点で分類

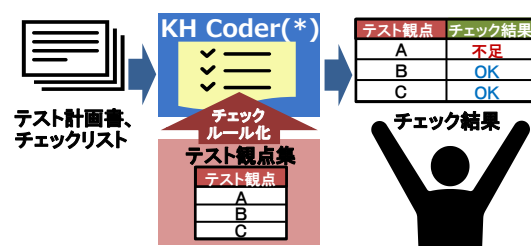


(指摘絞り込み負担軽減)

テスト

テスト観点集や障害事例等にもとづいてテスト項目を作成

作成したテスト項目の観点の網羅性を確認



(テスト観点漏れ防止)

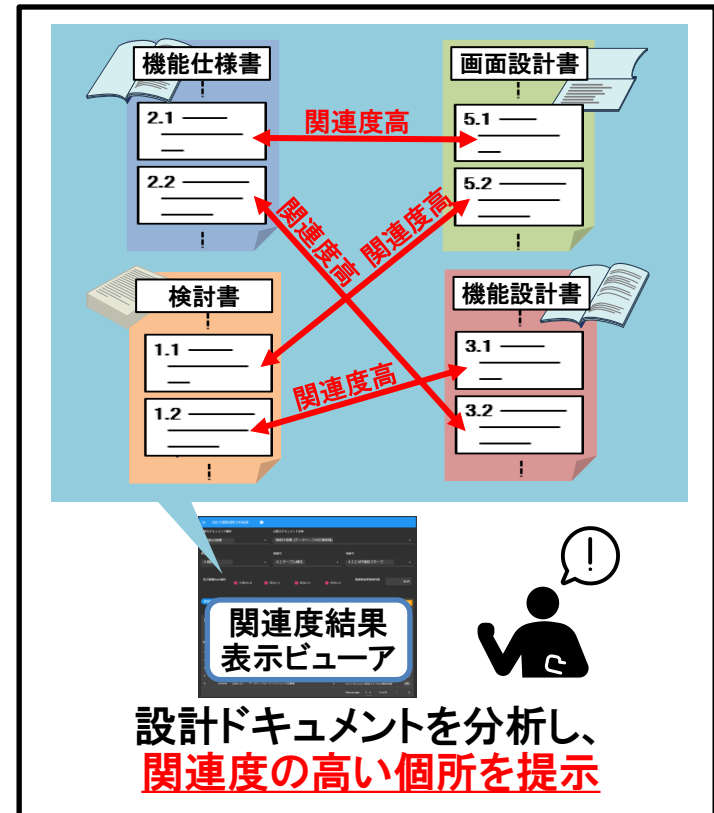
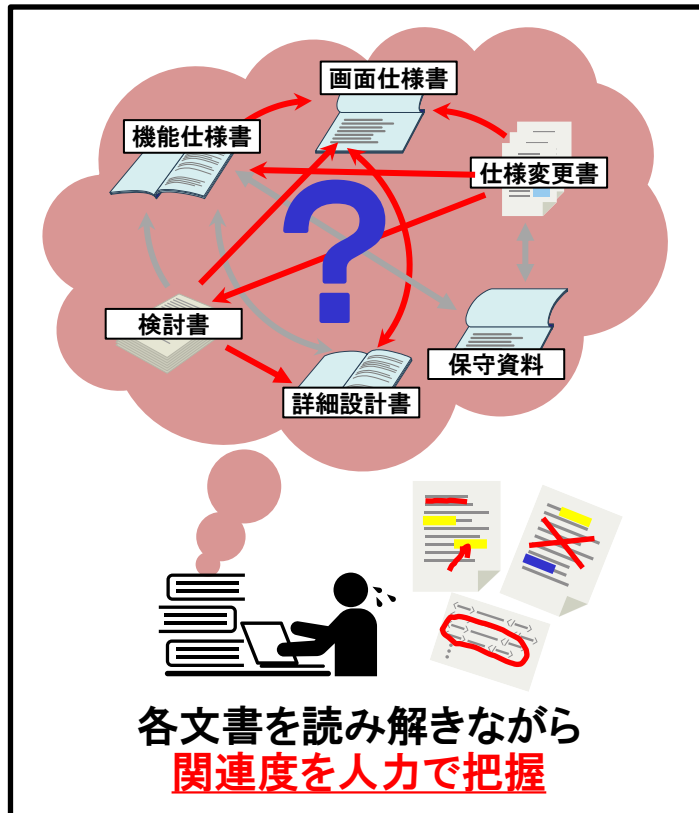
本日の発表内容



2. ドキュメント関連度分析による設計作業効率化

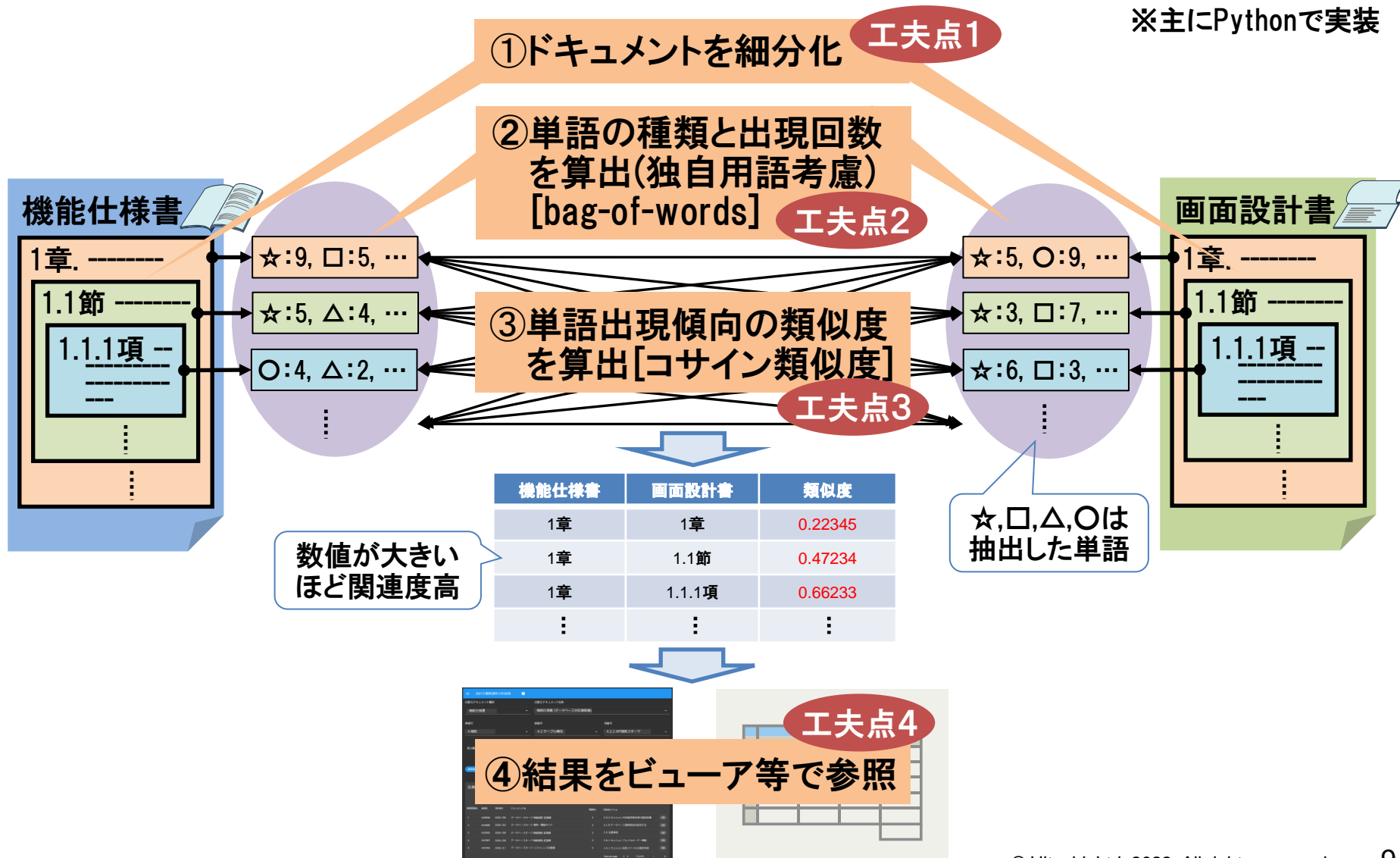
概要

- 設計ドキュメント間の関連度を分析し、関連度の高い個所を順に提示
▶ 関連度の高い個所から参照することで仕様・設計調査時間を短縮
- 分析対象ドキュメントを変えることで、様々な用途に活用可能
(機能仕様書 - 画面設計書、機能仕様書 - マニュアル等)



ドキュメント関連度分析の仕組み

※主にPythonで実装



bag-of-words

- 各文書に出現する単語の種類と出現回数をベクトルとして表現する手法

機能仕様書

この機能は他の機能と同期をとって通信を...

画面設計書

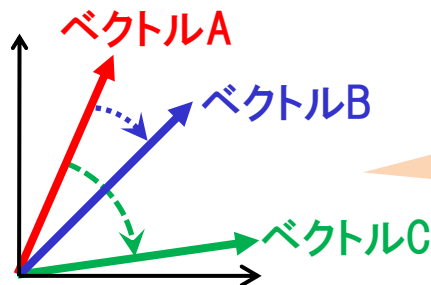
ここで通信に関する操作を...



| 文書名 | 出現単語の種類と出現回数 | | | | |
|-------|--------------|----|----|----|-----|
| | 機能 | 同期 | 通信 | 操作 | ... |
| 機能仕様書 | 2 | 1 | 1 | 0 | ... |
| 画面設計書 | 0 | 0 | 1 | 1 | ... |

コサイン類似度

- 2つのベクトルがどの程度似ているか(同じ方向を向いているか)を示す尺度(1に近い→類似している、0に近い→似ていない)



ベクトルBとベクトルCでは、ベクトルBの方がベクトルAに似ている

ドキュメント関連度分析結果ビューア

- 分析対象を絞り込んで、関連度分析結果を参照するためのビューア

① 設計文書選択

② 結果参照部位
(章節項)選択

③ 結果表示の
絞り込み指定

④ 結果表示ボタン

⑤ 結果表示

設計文書関連度分析結果

比較元ドキュメント種別: 機能仕様書 | 比較元ドキュメント名称: 機能仕様書 (データベース対応機能編)

章番号: 4.機能 | 節番号: 4.2 テーブル構成 | 項番号: 4.2.2 API機能スキーマ

表示階層No.の選択: 文書(No.0) 章(No.1) 節(No.2) 項(No.3) | 関連度結果取得件数: 20件

関連度判定結果表示 | 選択条件クリア | マニュアルサイト検索 | SEARCH

比較元ドキュメント名称: 機能仕様書 (データベース対応機能編)

関連度の高い順に結果を表示

| 関連度順位 | 関連度 | 資料番号 | ドキュメント名 | 階層No | 章節項タイトル | |
|-------|---------|----------|--------------------|------|-------------------------|---------------------|
| 1 | 0.08544 | 3000-J08 | データベースサーバ機能解説 拡張編 | 3 | 5.9.3 セッション引き継ぎ発生時の認証処理 | リンク |
| 2 | 0.03685 | 3000-J02 | データベースサーバ 運用・構築ガイド | 3 | 4.1.8 データベース接続環境を設定する | リンク |
| 3 | 0.02925 | 3000-J08 | データベースサーバ機能解説 拡張編 | 2 | 5.9 注意事項 | リンク |
| 4 | 0.02507 | 3000-J08 | データベースサーバ機能解説 拡張編 | 3 | 5.6.1 セッションフェイルオーバー機能 | リンク |
| 5 | 0.01816 | 3000-J17 | データベースサーバリファレンス定義編 | 3 | 3.4.1 セッション属性ファイルの指定内容 | リンク |

Rows per page: 5 | 1-5 of 20

工夫点1:ドキュメントを章/節/項に細分化する

- ドキュメント全体の関連度分析結果だけでは以下のような課題がある
 - ・ ドキュメント全体の関連度だけを見ても、関連する内容が含まれているかの判断ができない
 - ・ 結局、ユーザーは文書全体を読んだり、検索したりすることになるため、調査の手間がそれほど減らない
- ➡ 章/節/項に細分化した関連度を提示し、参照すべき個所を絞り込めるようにする(関連しそうかの判別も容易になる)

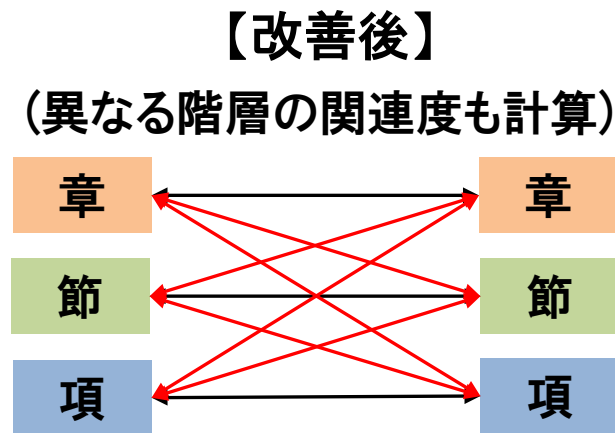
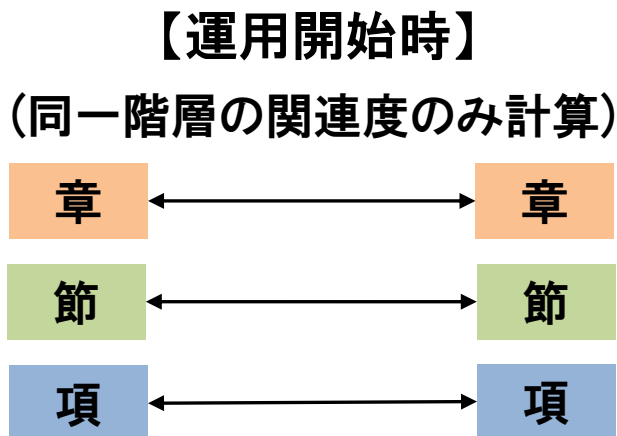
工夫点2:単語抽出時にプロジェクトの独自用語を識別する

- 設計書には一般的な単語を組み合わせて作られたプロジェクト独自用語が多く記載されている(機能名、コンポーネント名など)
例:「同時リセット機能」、「待機サーバ」等
- ➡ ドキュメントからプロジェクト独自用語を抽出し、一般的な単語と区別して識別することで、関連度の精度を向上

工夫点3:章/節/項全ての組合せで関連度を分析する

- 運用開始当初は、章は章、節は節のように同一階層でのみ関連度を分析（章と項のように異なる階層では精度が下がると推測）
- ユーザーから「異なる階層の関連度も分析して欲しい」という要望を受け、試行したところ、同一階層よりも**関連度の高い部位が検出された**

▶ ドキュメントによって章/節/項それぞれの記述量(含まれる単語数)は異なるため、階層による制限は設けずに関連度を分析するよう改善

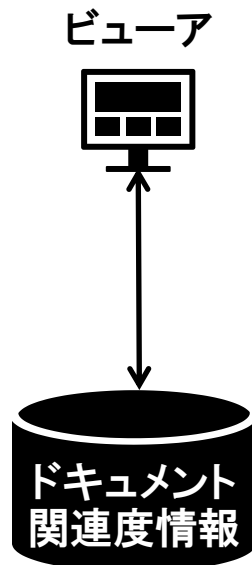


工夫点4: APIサーバを設置する

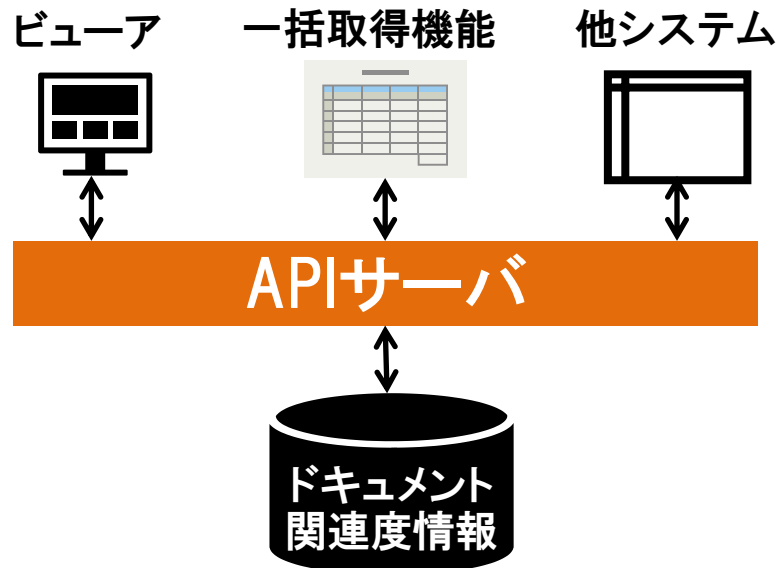
- 関連性分析の結果は専用のビューアを提供し参照
- ユーザーから「複数部位の結果を一括で取得したい」という要望を受けた(ビューアでは1件ずつの結果参照にとどまる)

▶ 今後も様々な要望(他システム連携等)が出てくると考え、「APIサーバ」を設置し、結果を様々な形態で利用できるように改善

【運用開始時】



【改善後】



分析結果の精度評価

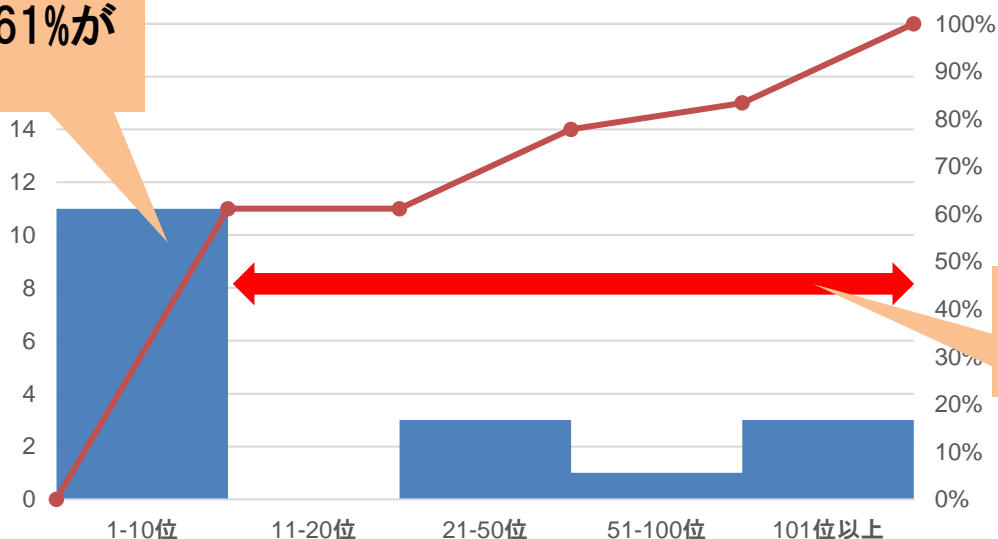
- 関連が明確なドキュメントを使って関連性分析結果の精度評価を実施

- ① 分析結果の上位10以内に入っている確率が61%
- ② 10位以内に入らない場合は、順位が極端に低くなる(100位など)

※上位10位までの結果を確認して見つからない場合は、調査方法を切り換えてもらうことでユーザーとは合意

関連度分析結果のパレート図

① 調査件数の61%が10位以内



② 10位より下は順位が極端に低くなる

3. 適用事例：マニュアル開発への適用

マニュアル開発の流れと課題

- 「マニュアル設計工程」では設計資料などの入力情報をマニュアルのどこにどのように載せるかを検討する



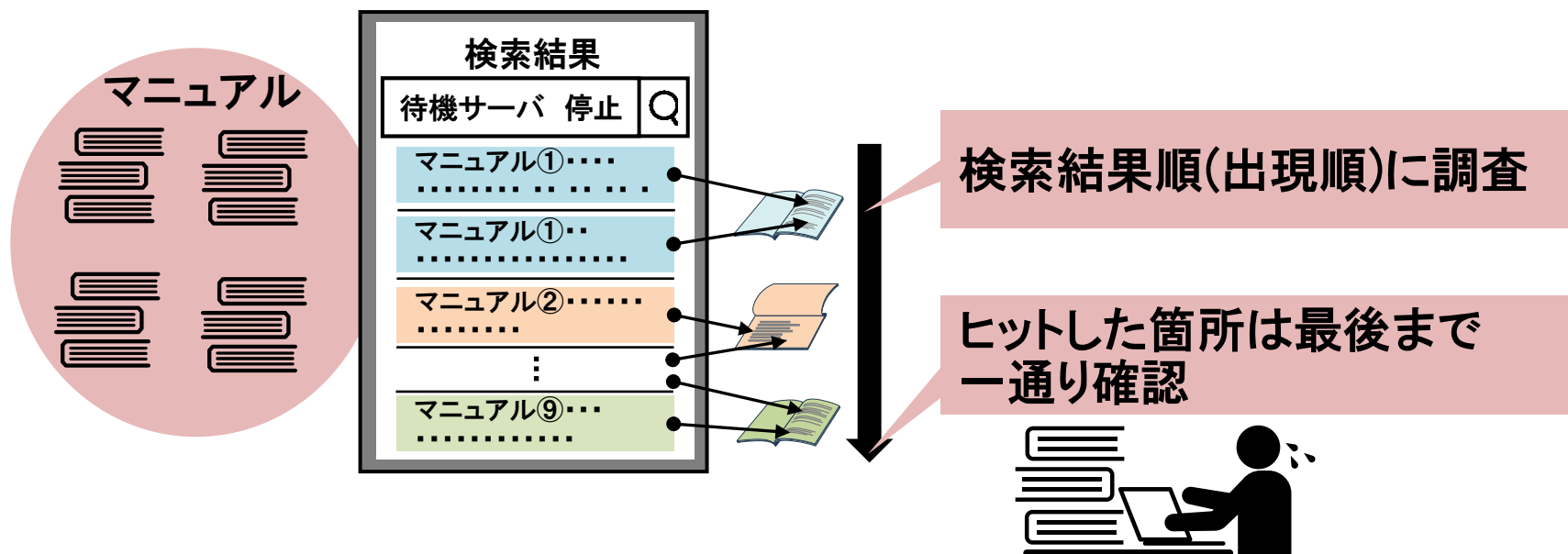
課題:設計資料に対応するマニュアルの追記・変更個所の特定が困難

- 設計資料、マニュアルともに冊数、ページ数が多い
(特に長年の開発の積み重ねで、設計資料の冊数が膨大)
- 設計資料とマニュアルの対応関係は複雑(複数冊 対 複数冊の対応)
- 記載個所の特定は「熟練マニュアル作成者のノウハウ」になっている



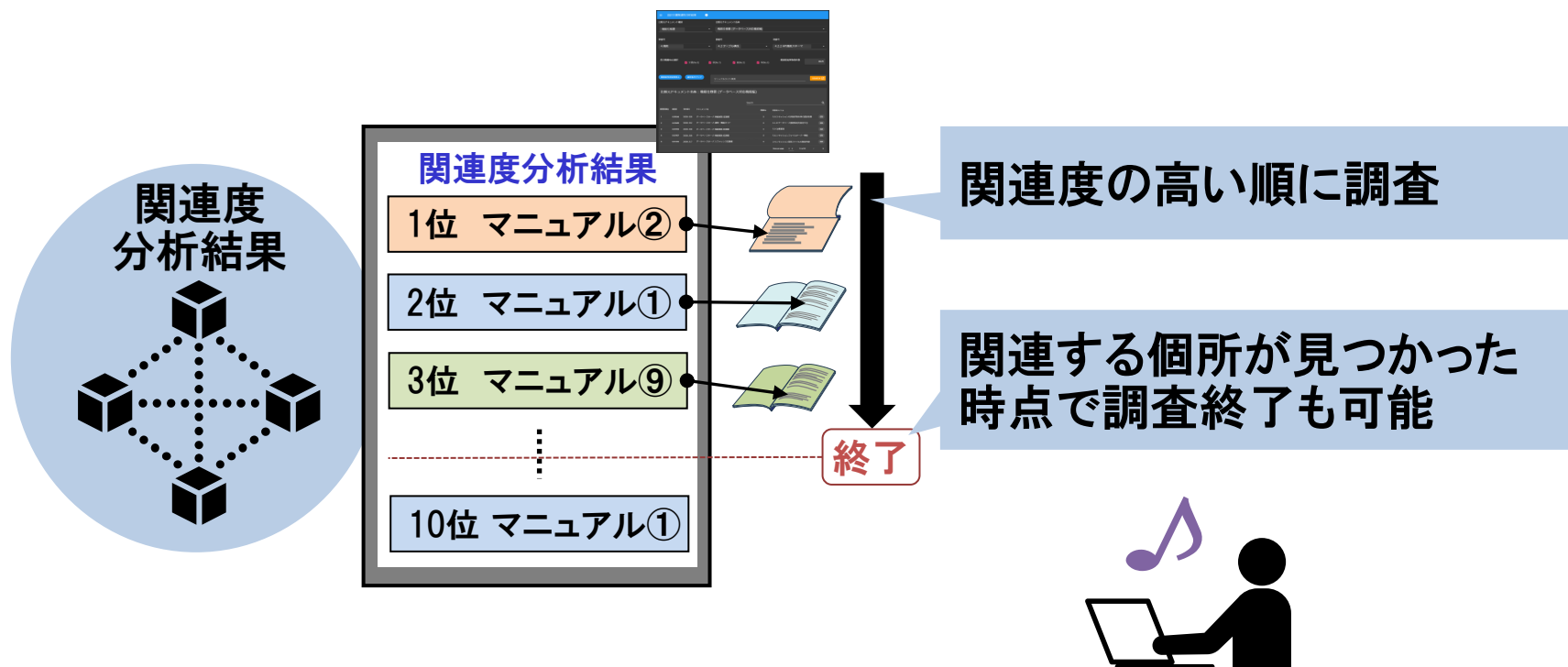
従来のマニュアル設計工程作業(ドキュメント関連度分析なし)

- 設計資料から抽出した複数のキーワードでマニュアルを検索
(検索結果は出現順に並んでいる)
- ヒットしたドキュメントの**該当箇所を通読**して関連しているかを調査
- ヒットした箇所の全てを一通り確認する(探している個所が検索結果の最後に来ることもあるため)
- 調査時間:15~20分/件



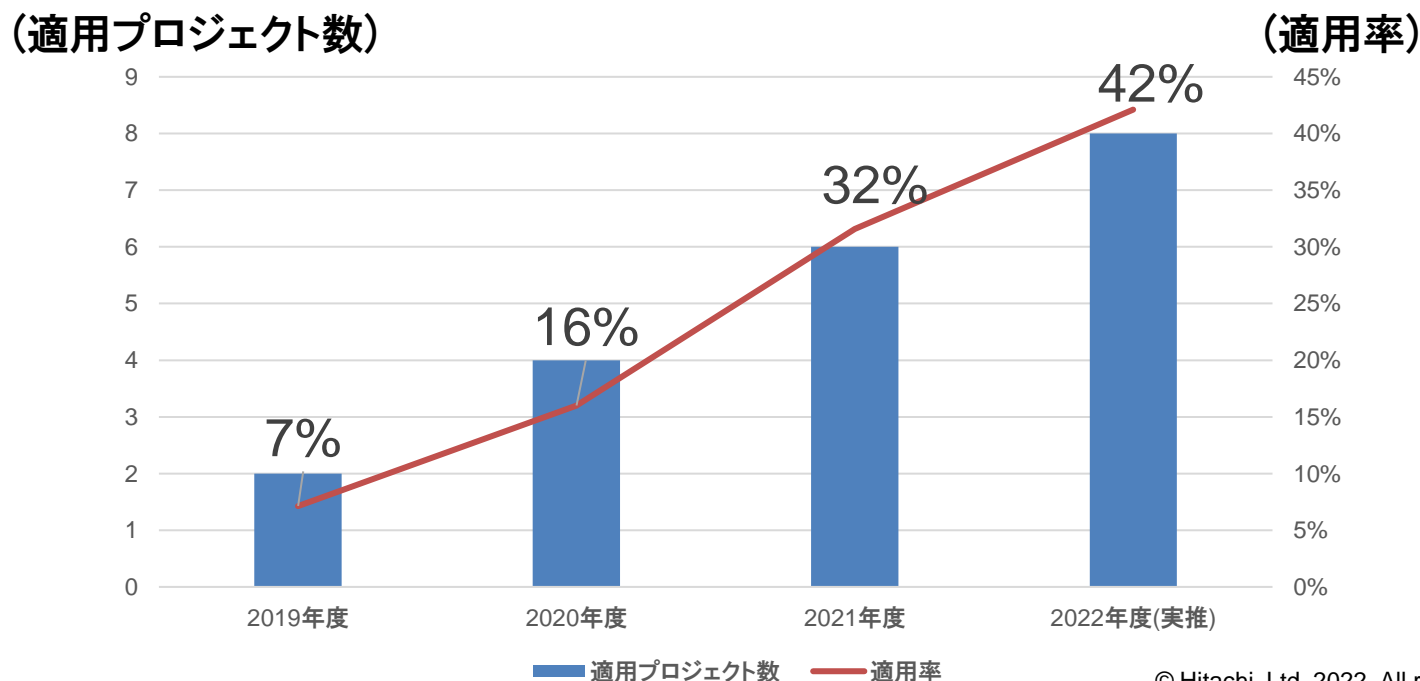
改善後のマニュアル設計工程作業(ドキュメント関連度分析適用)

- 調査対象記載部位(章/節/項)の関連度分析結果を表示
- 関連度の順位が高い順に記載内容を確認
- 上位で関連する個所が見つければ、その時点で調査を終了することも可能
- 調査時間:10分/件



適用状況(マニュアル開発プロジェクト)

- マニュアル開発プロジェクトでの活用は年々増加・定着化傾向
 - ・ 一度適用したプロジェクトでは、活用が定着化し毎バージョンで利用継続
 - ・ 新規適用プロジェクトも年々増加
 - ・ 適用プロジェクトとは定期的にフィードバック会を実施し、要望を反映
(例) ビューアの操作性改善、分析結果の一括取得など



4. まとめ

成果と今後の方針

- 各施策とも実プロジェクトにおいて、エンハンス作業の効率向上、品質向上に寄与する成果をあげている

(適用効果の例)

| 施策 | 適用効果 |
|-----------------|----------------------------|
| 設計工程向け施策(関連度分析) | マニュアル設計工程の作業時間短縮(効率向上) |
| コーディング工程向け施策 | 既存ソースの品質状況把握で活用多数(品質向上) |
| テスト工程向け施策 | 適用プロジェクトでテスト項目の漏れを検出(品質向上) |

- 結果説明会やフィードバック会等を実施することで、ユーザーとの間で良いフィードバックサイクルを構築できている

本施策の技術・成果を、他組織が推進している施策と連携させ、現場担当者の悩みに応える新たな施策を立ち上げる

- [1] Willi Richert, Luis Pedro Coelho. (2013). Building Machine Learning Systems with Python. Packet Publishing. (ウィリ・リチャート, ルイス・ペドロ・コエーリョ, 斎藤 康毅(訳)(2014). 実践 機械学習システム. オライリージャパン)
- [2] 独立行政法人情報処理推進機構 技術本部 ソフトウェア高信頼化センター編 (2018). コーディング作法ガイド[C言語版] ESCR Ver.3.0 独立行政法人情報処理推進機構
<https://www.ipa.go.jp/files/000064005.pdf>
- [3] 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

END

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～

HITACHI
Inspire the Next 