



X-Analysis AI^{*}

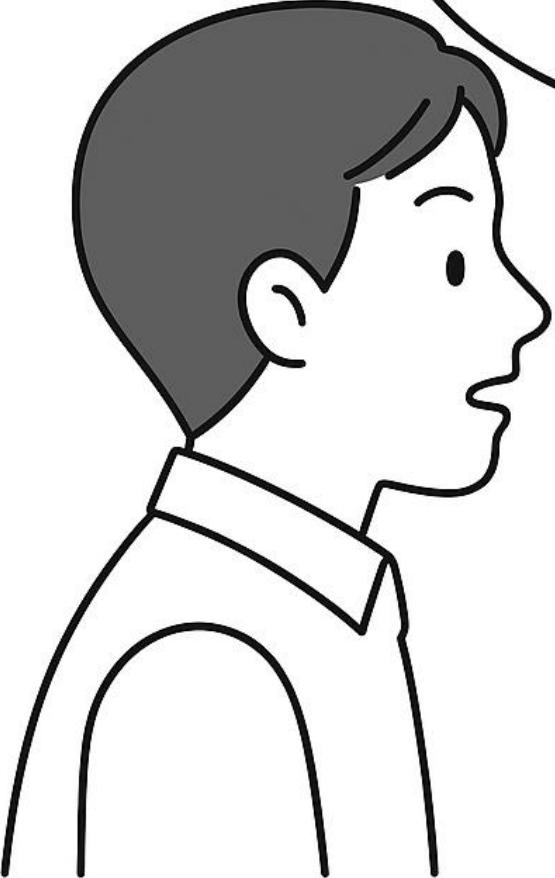
生成AIと解析技術で挑む IBM i 資産継承の最前線



2025/11/27

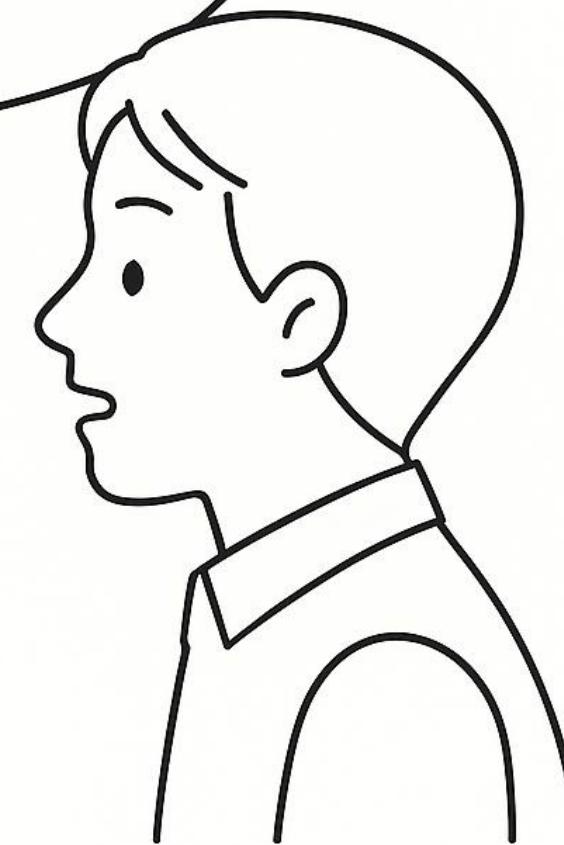


生成AIの登場で
こんなこと
期待しませんでした？



製品区分フィールドを
定数「B」と比較
しているのはどこ

この仕切価格を
更新している
プログラムはどれ?





この伝票ファイルと
参照や親子関係にあ
るファイルはどれ？



計算で桁落ちの
危険性があるソース
箇所全部知りたい

3日前に
誰かが修正した
箇所を知りたい



知ってますか？

これらは
生成AI誕生20年以上
前から実現できて
ています



実現しているのは？



X-Analysis

機能は既に100以上



可視化と洞察

Visualization & Insight

アプリケーションの中身をわかりやすいチャート、フローなどで表現するため、構成、関連、仕様の理解にかかる時間が短縮されます。



影響分析

Impact analysis

アプリケーション変更の影響箇所や詳細仕様に関わるロジック、数式、変数を特定します。正確、高速に行うので、調査効率が格段に上がります。



診断と棚卸

Diagnostics & Inventory

アプリケーションの問題点を特定し、プログラムやファイル複雑性などを点数付します。予防保守、制度の高い工数見積に用いることができます。

AI時代に足りないことは？



自由な探索 (Go Anywhere-from-Anywhere)

- どこからどこにでも移動可能（思考を止める必要がない）
 - 自発的にシステムを理解できる
 - 概要 ⇄ 詳細（コード）

すごく便利だけど自然言語操作できない

だから実装しちゃいました

X-Analysis AI[★]

AIと会話で課題解決

The screenshot shows the X-Analysis application interface. At the top, there's a navigation bar with tabs: Repositories, Metrics, Objects, Program Structure (which is selected), and Data Flow List. Below the navigation bar, the repository is set to XAN4CDXA, library to XAN4CDEM, and program to WWCUSTS. A search bar is present. The main area displays a table of program structures. On the left side of the interface, there are several callout boxes with icons and text, representing AI-generated assistance messages:

- WWCUSTSを頂点としたプログラム呼び出し構造を見せて**
- プログラムの呼び出し構造を表示しました。特定のプログラムについて詳しい情報（ソースコード、ビジネスロジック、データフローなど）が必要な場合は、お申し付けください。**
- CUSTMNT1のデータフローを見せて**
- プログラムCUSTMNT1のデータフローを表示しました。このデータフローについて特定の部分の詳細や、他のプログラムのデータフローも確認したい場合はお申し付けください。**
- CUSTFMNTを頂点としたプログラム呼び出し構造を見せて**
- 申し訳ありません。プログラム「CUSTFMNT」のライブラリ名が必要です。プログラムが存在するライブラリ名を教えていただけますでしょうか？**
- XAN4CDEMIPです**

At the bottom left, there's a "Chat" section with a text input field and a send button. The footer contains the text "X-Analysis Assistant can make mistakes. Check important info." and page navigation links.

特徴1

自然言語インターフェース

人間の言葉でアプリケーションに関する質問をするだけで、AIがリポジトリ情報に基づきわかりやすく回答。非エンジニアでも扱いやすい設計です。

課題解決の敷居が更に下がる



時間と効率化

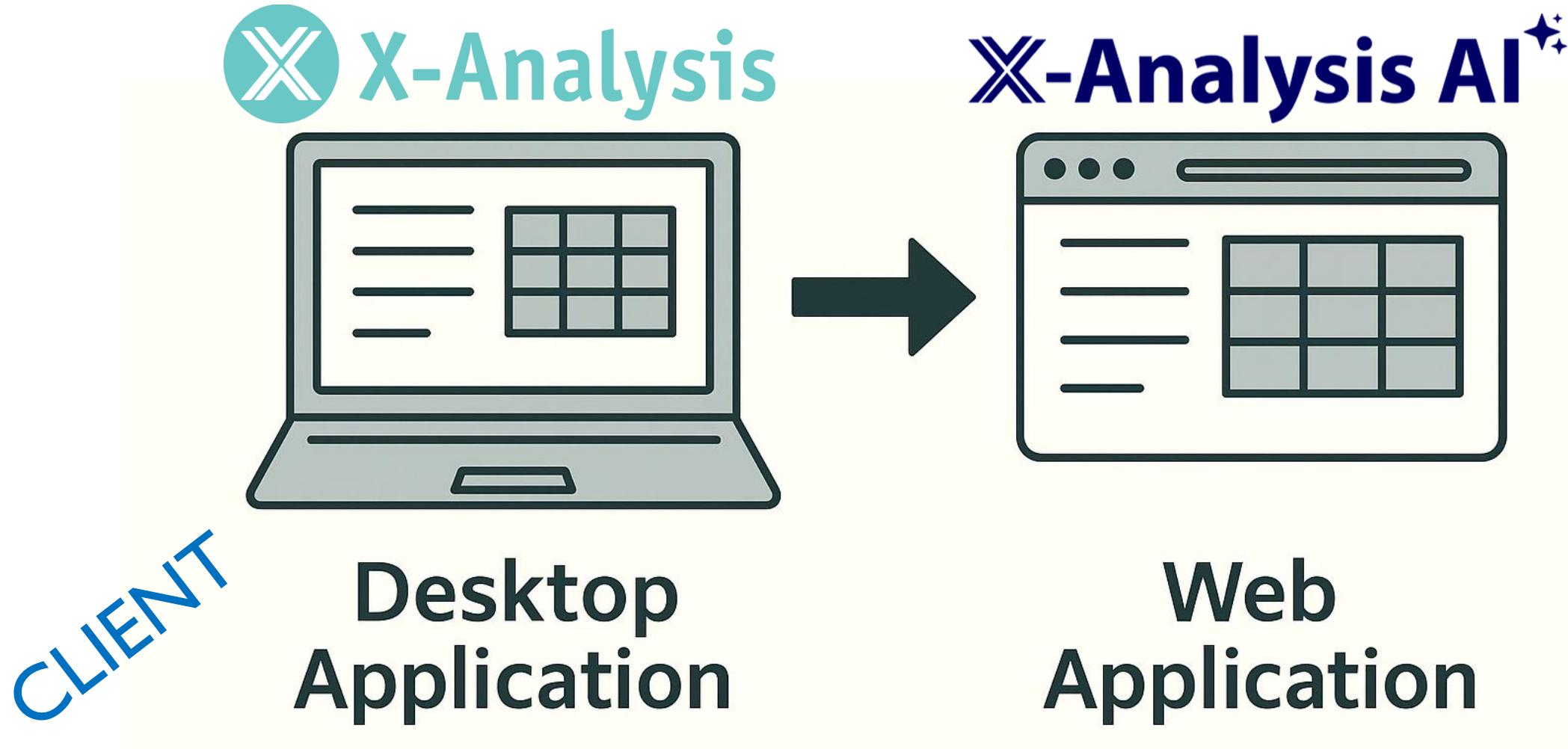
現行システムのメンテナンスに時間をかけず
に新しい取り組みの時間をどう作るか？



ノウハウの伝達

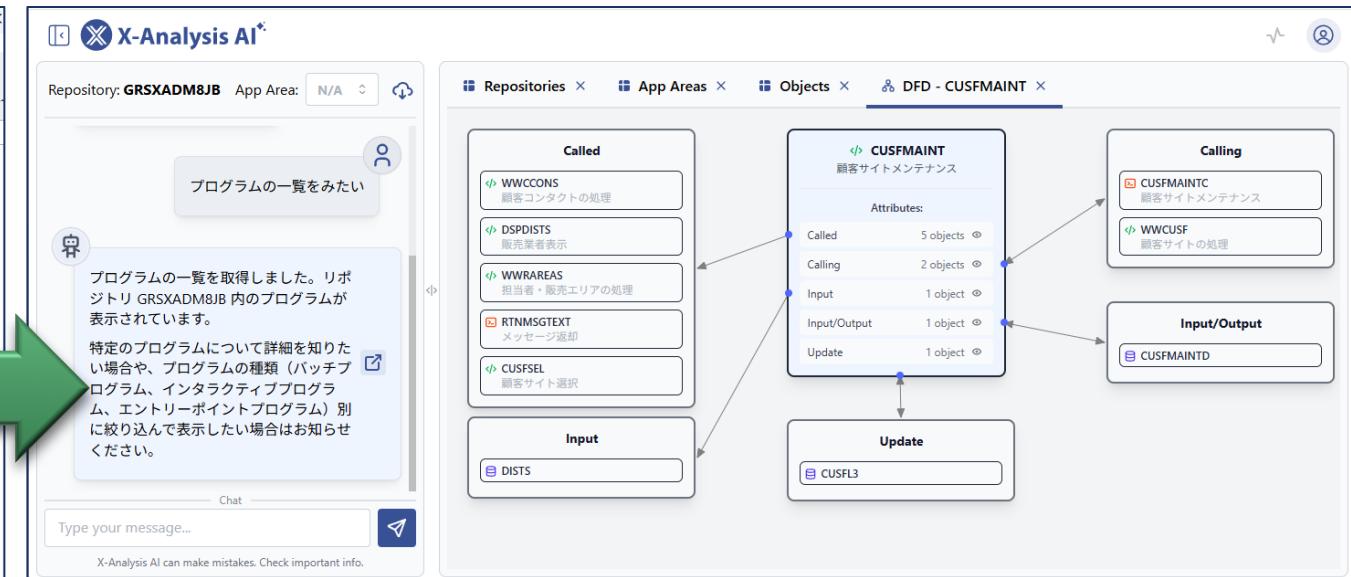
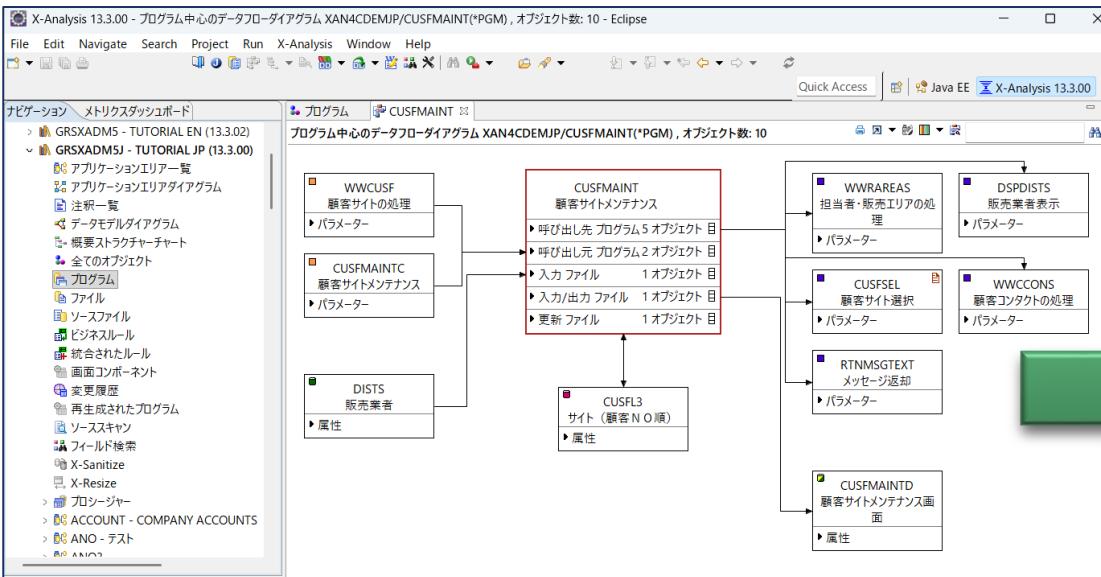
どのように若手エンジニアにベテランのノ
ウハウを伝えるか？

既存機能はどんどん統合 予定



既存機能はどんどん統合予定

例えばデータフローダイアグラム



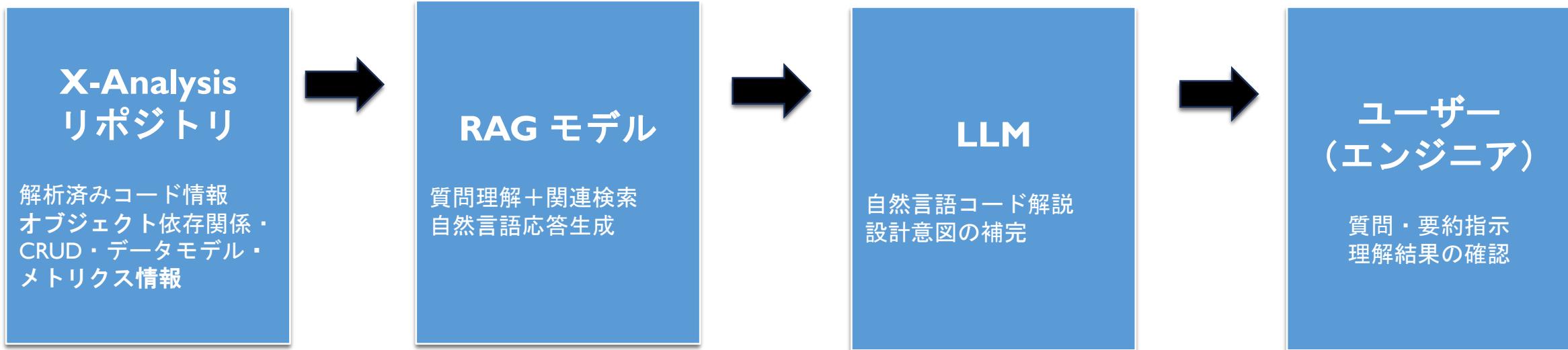
クライアントソフトウェア

Webアプリケーション

リポジトリ & LLMの強力タッグ



X-AnalysisとLLMの連携構造



ユーザーアプリケーション → X-Analysisリポジトリ → RAG検索 → LLM応答 → ユーザー提示

コードアシスト機能もプラス

The screenshot shows the X-Analysis AI interface with the following details:

- Top Bar:** リポジトリ: GRSXADM8JB, アプリエリア: 適当なし.
- Left Sidebar:** こんにちは、X-Analysis AIです。今日はどのようなご用件でしょうか? (Informal greeting message), リストが読み込まれました: アプリエリア (List loaded from application area).
- Right Content Area:**
 - Section 1: プログラムの説明と目的**

CUSMAINTは、顧客サイトメンテナンス(顧客サイトメンテナンス)プログラムであり、CUSFL3ファイル(顧客サイト(顧客NO順)・顧客番号別サイト)内の顧客記録のオンラインメンテナンスの主要な機能として機能します。このプログラムはフラットスクリーンインターフェイスを通じて顧客情報を表示し、ユーザーが顧客の詳細を更新できますようにします。DISTSファイルからデータを取得し、顧客サイト選択(CUSSEL)、販売業者表示(DSPOTS)、メッセージ送却(RTNMSGTEXT)、顧客コントクトの処理(WWCCONS)、担当者・販売エリアの処理(WWRAREAS)などの補助機能を呼び出すことができます。パラメーターは呼び出されたときにプログラムに渡され、顧客サイトメンテナンス(CUSMAINTC)および顧客サイトの処理(WWCUSH)によって呼び出することができます。
 - Section 2: ビジネスルールの概要**
 - 顧客名は必須であり、空白にすることはできません（インジケータ32で検証済み）
 - 電話番号とFAX番号は、数字とスペース（0～9とスペース）のみで構成されている必要があります（インジケータ34と35で検証済み）
 - ディストリビュータコードが提供されている場合、DISTSファイル内に存在している必要があります（インジケータ38で検証済み）
 - ステータスは、定義済みのコンパイル時配列の37個のステータスコードから有効な値である必要があります（インジケータ39で検証済み）
 - 連絡先名は必須であり、空白にすることはできません（インジケータ40で検証済み）
 - 敬称は、「Mr.」、「Mrs.」、「Ms.」、または「Dr.」（指定されている場合）のいずれかである必要があります（インジケータ41で検証済み）
 - 顧客番号は一度設定すると変更できないように保護されます（インジケータ31）
 - すべての検証エラーはレコードの更新を妨げ、RTNMSGTEXT経由で取得した適切なエラーメッセージを表示します。
 - Section 3: ロジックの適用**

プログラムは標準的な保守パトーンに従います。顧客番号で既存の顧客レコードを取得し、現在の値を画面に表示し、ユーザーによる変更を受け入れ、入力を検証し、有効な場合はファイルを更新します。メインのプロセスには、画面の値を設定し、月略語を使用して現在の日付をフォーマットし、顧客番号フィールドを保護し、CUSSELで顧客が指定されていない場合に顧客の選択を求める初期化ルーチン("INZSR")が含まれます。メインループは、終了(F3)またはキャンセル(F12)が押されるまで継続され、連絡機能(F8=連絡先、F9=販売店、F10=配達区域)のコマンドキー処理が行われます。フィールドの入力は、データベースフィールドを画面フィールドにマップするWRITESCNサブルーチンによって行われ、UPDATEFILEサブルーチンはデータベースレコードを更新する前に逆のマッピングを実行します。
 - Section 4: 主要なデータ処理手順**
 - レコード検索: 顧客番号をキーとしてCUSFL3へのCHAIN操作。レコードが見つからない場合はプログラムが終了します。
 - 画面への入力: "YMD"形式を使用して日付変換を行い、WRITESCNサブルーチンでデータベースフィールドを表示フィールドにマッピングします。
 - 日付処理: 現在の日付コンボネートを抽出し、月略語検索('JanFebMarAprMayJunJulAugSepOctNovDec')と部分文字列計算を使用して日付の表示形式を設定します。
 - 検証処理: VALIDATEサブルーチンで順次フィールド検証を行い、インジケータ("IN32~IN44")を使用して同時にエラーフラグを設定します。
 - 数値検証: 電話/FAX検証のための「0123456789」文字列に対するCHECK操作
 - ファイル検索: ディストリビューター検査のためのDISTSファイル(RPRODSレコード形式)に対するSETLL操作
 - 配列ルックアップ: LOOKUP操作を使用して、コンパイル時の37個の有効なステータスコードの配列に対するステータス検証
 - 更新操作: エラー処理を伴うRCUSUFレコード形式の更新操作。成功した場合は更新フラグを'Y'に設定します。
 - Section 5: スクリーンとデータの視覚化**

このプログラムは、レコード形式ZZFT01の表示ファイルCUSMAINTDを使用し、24x80の保守画面を1画面表示します。画面にはタイトルとして「Customer Site Maintenance」と、会社ロゴ「Databorough Ltd.」が青色で表示されます。ファンクションキーのオプションは青色で表示され、「F8=連絡先、F9=販売代理店、F10=配達エリア」に加え、標準のCA03(終了)とCA12(キャンセル)が存在します。フィールドには、顧客番号(インジケータ31)で保護され、強調表示、顧客名、住所(4行)、国、郵便番号、電話番号、FAX番号、メールアドレス、ウェブサイト、販売代理店/営業担当者コード、ステータス、連絡先情報(氏名、役職、職名)、ISO形式の連絡日が含まれます。エラーの強調表示には、検証インジケータ32~44によって制御される反転表示属性(DSPATR(RI))が使用されます。画面には、プログラム名、現在時刻、フォーマットされたメッセージを入力してください。

ソースコード概要説明
ソースコード詳細説明
テストケース生成
・・・・
その他
自然言語でコードの理解を加速させる

リポジトリ連携のメリット①

その他



特徴3

ハルシネーション(幻覚)対策

AIの誤回答を最小限に抑えるため、AIに参照させるコンテキストに信頼性の高いリポジトリデータを組み込むアプローチを採用しています。調査対象の情報をX-Analysisのリポジトリから検索し、その文脈を添えてLLMに問い合わせることで精度を向上させています。

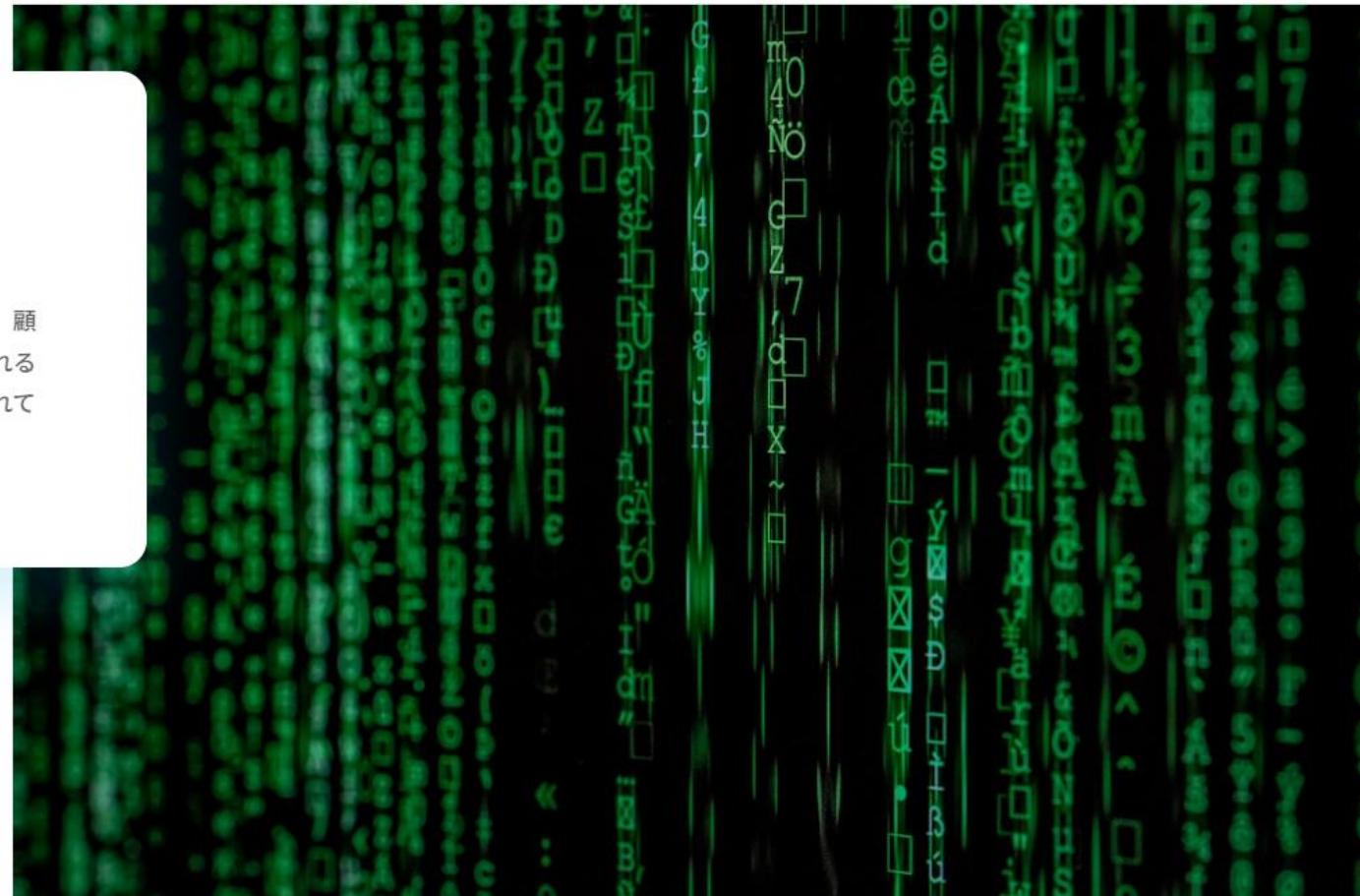
リポジトリ活用のメリット②

その他

特徴4

セキュアな環境と柔軟な設計

プログラム解析に必要な最小限の情報のみをLLMプロバイダーと共有し、顧客のデータベースファイルの内容そのものは一切送信しません。共有されるのはプログラム名やフィールド名、該当コード片のテキストなどに限られており、機密データは外部に出ません。



若手エンジニアのレガシー資産理解を加速する技術シナリオ

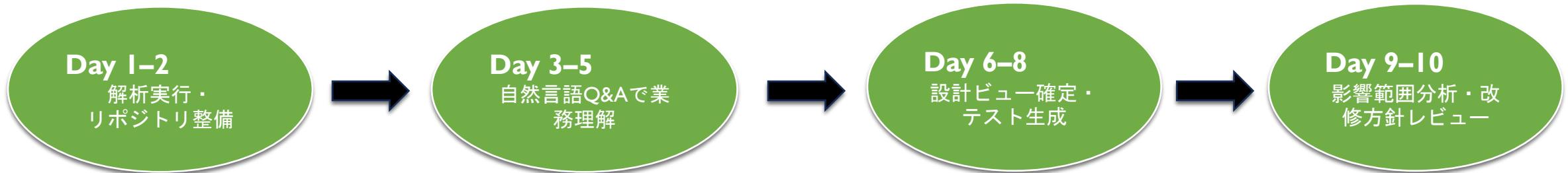
- ゴール
 - 自然言語によるコード解説・設計情報の自動抽出・テストパターン生成を組み合わせ、2~4週間で対象サブシステムの理解と安全な変更着手を可能にする
- 対象
 - IBM i (RPG/COBOL/CL、DB2 for i) 等で構成されたレガシー業務アプリケーション
- 全体像
 - ① X-Analysisリポジトリに資産を完全解析→依存・データフロー・画面/バッチ構造を可視化
 - ② RAG連携LLMにより自然言語Q&Aでコードを説明
 - ③ 設計情報（CRUD、エンティティ、入出力IF、制御パターン）を自動抽出し設計ビュー化
 - ④ 代表シナリオからテストデータ・テストパターンを自動生成、回帰影響範囲で優先度付け
- 具体フロー（1スプリント=2週間想定）
 - Day 1-2: 解析実行・リポジトリ整備／主要ユースケース選定
 - Day 3-5: 自然言語Q&Aで業務ルール・例外処理を把握→設計ビューへ確定
 - Day 6-8: CRUDとデータフローからテストパターン自動生成→不足分を追加入力
 - Day 9-10: テスト実行／影響範囲の変更方針レビュー→安全な改修計画を合意
- 生成物
 - 要約レポート（自然言語のコード解説/Q&Aログ）
 - 設計ビュー（モジュール構造、データモデル、IF仕様）
 - テスト資産（テストパターン・テストデータ・トレーサビリティ表）
- KPI/効果
 - 理解リードタイム ▲50-70%
 - 仕様漏れ/手戻り ▲30-50%
 - 回帰テスト工数 ▲30%（代表パターン化）

技術シナリオ図解：若手エンジニアの理解促進フロー



利用技術構成 : X-Analysis リポジトリ + LLM (RAG連携) + 自動設計抽出 + テスト生成エンジン

スプリント時系列図（2週間サイクル例）



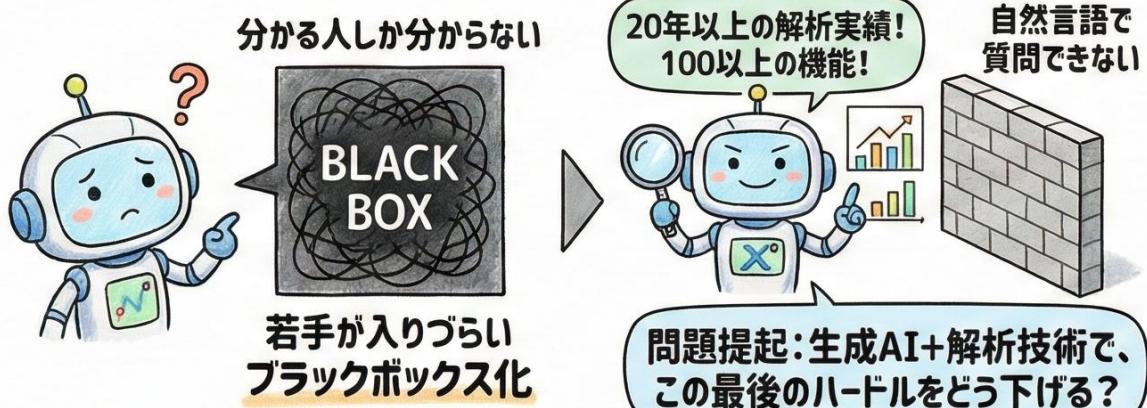
- ✓ Day 1-2: 解析実行・リポジトリ整備／主要ユースケース選定
- ✓ Day 3-5: 自然言語Q&Aで業務ルール・例外処理を把握→設計ビューへ確定
- ✓ Day 6-8: CRUDとデータフローからテストパターン自動生成→不足分を追加入力
- ✓ Day 9-10: テスト実行／影響範囲の変更方針レビュー→安全な改修計画を合意

まとめ

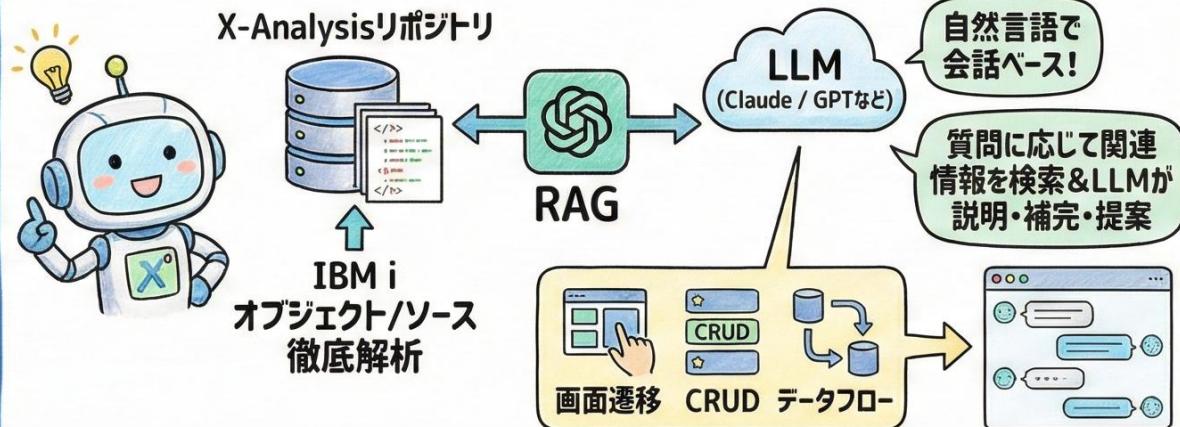
IBM i 資産継承を加速！X-Analysis と 生成AI の連携



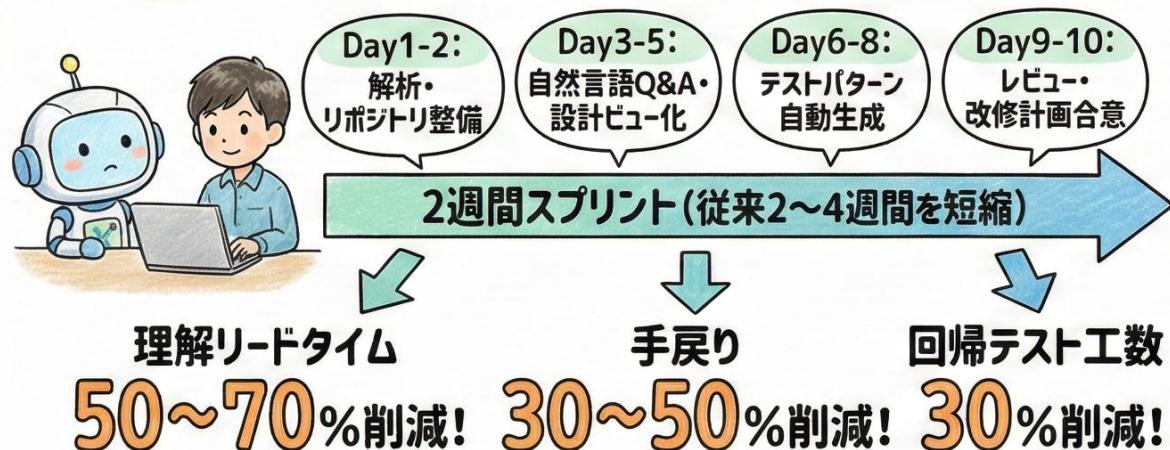
起：IBM i 資産継承の課題と X-Analysis



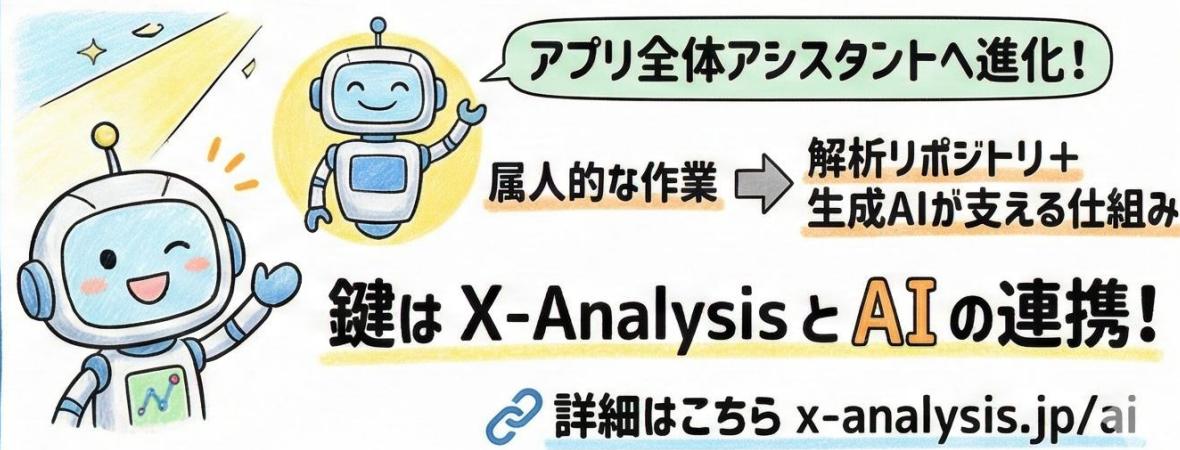
承：生成AIとRAG連携による新アシスタント構造



転：若手理解を加速する2週間スprint



結：AIが支える資産継承の未来



THANK YOU

<https://x-analysis.jp/ai>

<https://x-analysis.jp/tech/tag/AI>



FRESCHE
SOLUTIONS

