

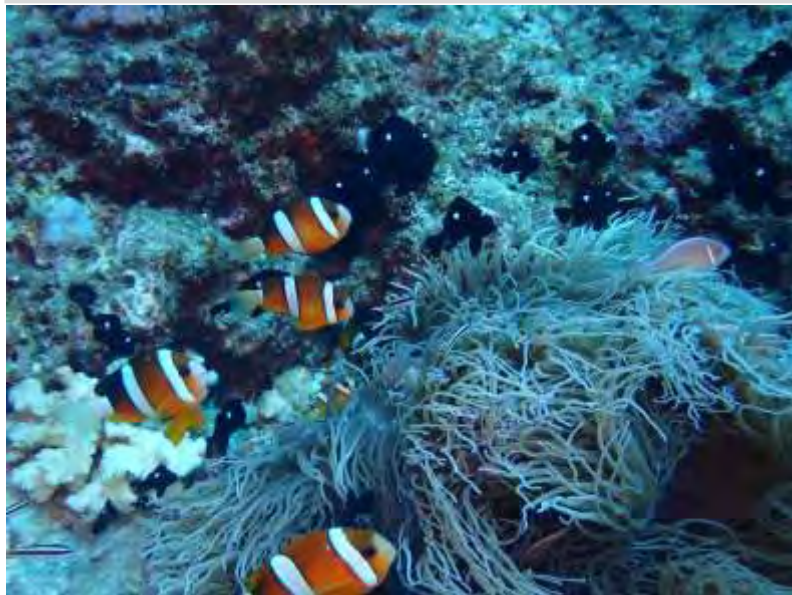
# 勘や経験やgrepはもう不要

## 影響調査はピンポイントで的確に！

---

株式会社アシスト  
社長室 新規事業推進担当  
永見 絢子

# 自己紹介



**社長室 新規事業推進担当**

**永見 絢子**

**アシスト入社後、BI製品のフィールド技術、プリセールス、研修講師を担当。**

**2016年より社長室で新規取扱製品の立ち上げを担当。**

**趣味は読書とダイビング。**

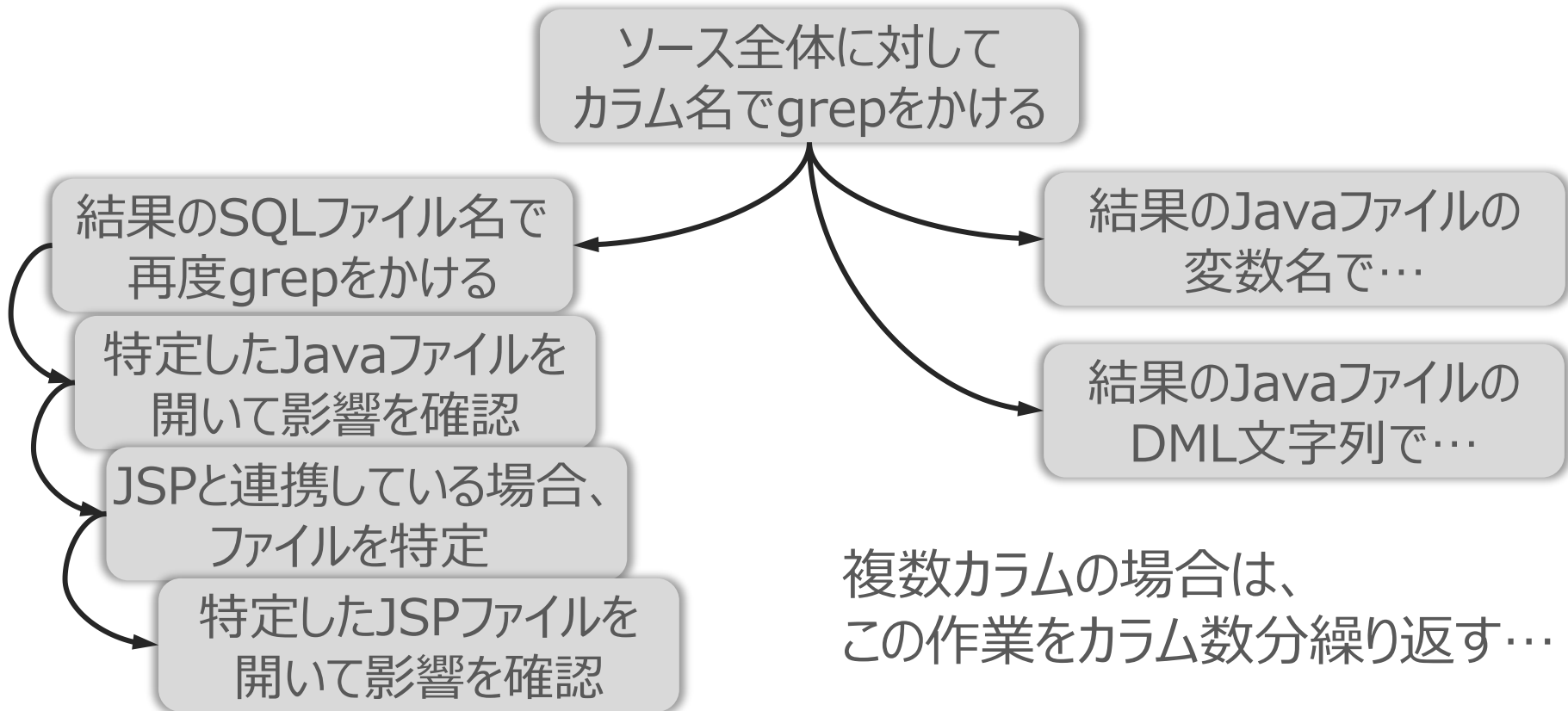
まずは質問です

---

影響調査

どうしてですか？

# 特定カラムの変更 : grepの場合



# なぜ手間と時間をかけざるを得ないのか

- 開発当時のメンバーがいない
- ドキュメントがメンテナンスされていない
- 継続的に機能追加や仕様変更を繰り返してきた



**誰もシステム全体を把握していない…  
地道に調査するしかない…**

# 影響調査に工数・時間がかかるだけでなく…

---

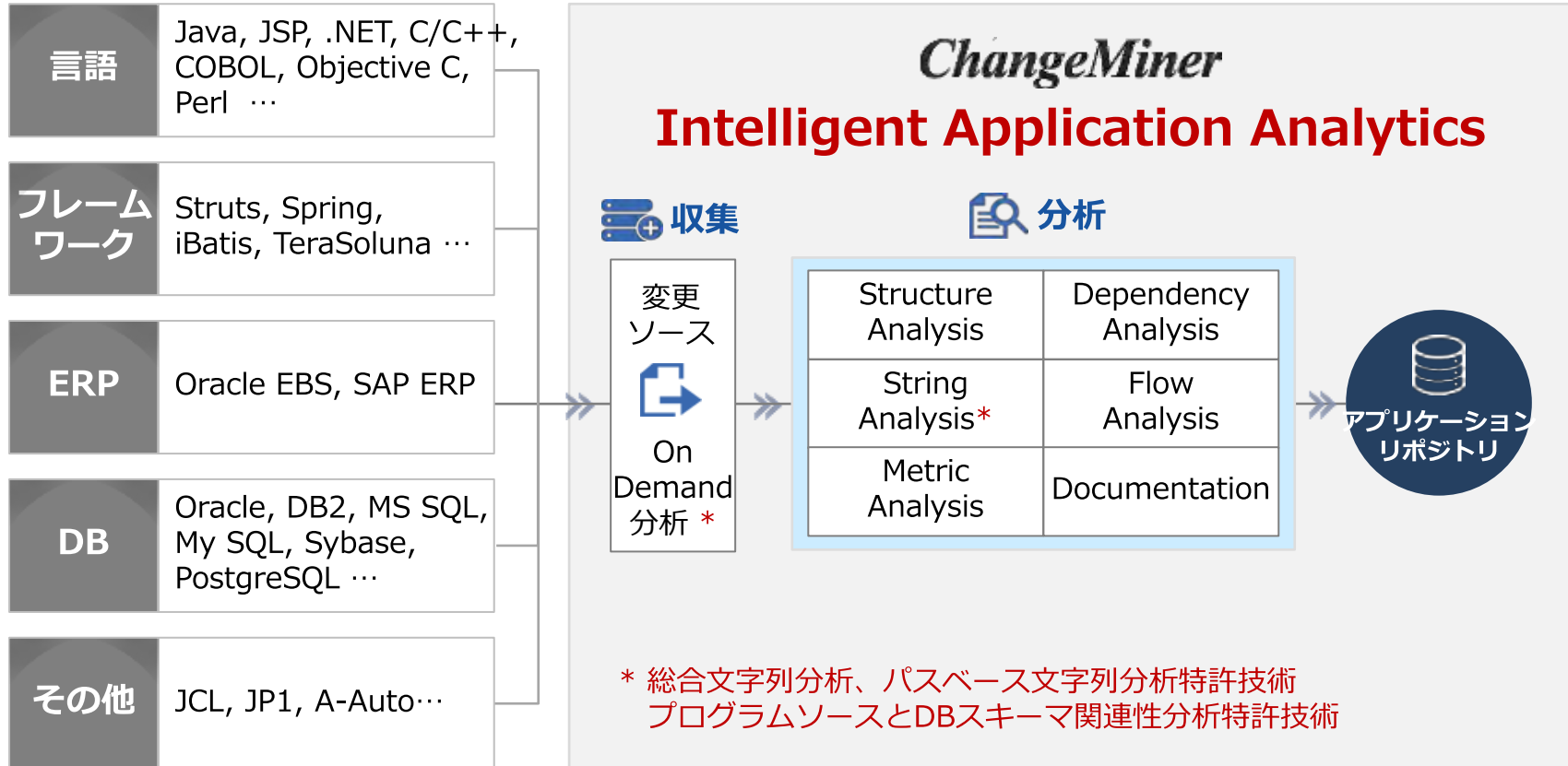
- 調査漏れによる障害が発生してしまう  
思いがけない箇所に影響が出てしまい、本番運用時に障害発生、業務を止めてしまった。
- テスト工数がかかってしまう  
障害を防ごうと、網羅テストを実施している。

そこで

---

# ChangeMinerで システムを可視化

# ChangeMiner : システムの可視化・影響分析ソリューション





# 主要機能要約

## ダッシュボード と統計

- プログラム/DB 資産と変更状況
- 非呼出プログラム/テーブル、非存在オブジェクト、呼出頻度
- 国際標準メトリックを用いたソースコードの規模と複雑度

## 検索

- ファイル名やキーワードを使った高速な検索
- ソースとDBオブジェクト全体を対象とした検索
- 複数条件を組み合わせた絞り込み検索

## 構造分析

- 業務/プログラム/DB構成と要約情報を提供
- ソースコード内部構造情報を可視化
- DBスキーマ内部構造情報を可視化

## 関連分析

- 画面プログラムからDBまでの関連を可視化
- プログラムとDBのCRUDマトリックスを自動生成
- オブジェクトレベルの関連を可視化

## フロー分析

- 各種ダイアグラムを自動生成
  - フローダイアグラム、クラスダイアグラム、シーケンスダイアグラム、SQLダイアグラム
- データ（変数）値のフローを追跡（Java、PL/SQL、COBOL）

## 変更履歴 管理

- 指定期間内のソースコードの変更状況
- オブジェクトレベルの新規/修正/削除などの詳細内容を追跡して比較

## レポート

- 分析結果をExcelやPDFなどに出力
- 多彩な標準レポートを提供
- 任意のデザインのレポートを作成できるデザイナー \*オプション

## 管理機能

- 分析設定を簡単指定できるプリセット機能
- 収集や分析作業の自動化
- サーバ資源の使用状況と実行作業をモニタリング

# 主要機能要約

## ダッシュボード と統計

- プログラム/DB 資産と変更状況
- 非実行プログラム（ニール）非存在プログラム

## 検索

## 構造分析

- 業務/プログラム/DB構成と要約情報を提供
- ソースコード内部構造情報を可視化
- DBスキーマ内部構造情報を可視化

## 関連分析

- 画面プログラムからDBまでの関連を可視化
- プログラムとDBのCRUDマトリックスを自動生成
- オブジェクトレベルの関連を可視化

- 各種ダイアグラムを自動生成
- フローダイアグラム、クラスダイアグラム、シーケンス

## レポート

- 分析結果をExcelやPDFなどに出力
- 多彩な標準レポートを提供
- 任意のデザインのレポートを作成できるデザイナー \*オプション

## 管理機能

- 分析設定を簡単指定できるプリセット機能
- 収集や分析作業の自動化
- サーバ資源の使用状況と実行作業をモニタリング

特に評価の高い機能を  
3つご紹介

# 機能 1 : オブジェクトをまたがる検索

- ソースコードとDBスキーマ全体を対象に検索
  - どこを検索するか、意識する必要なし
  - 完全一致や正規表現、ファイル形式指定などの条件指定が可能

**キーワード検索**

SampleObject

☐ 大/小文字区別 ☐ ソースコードの情報を表示

文字列

検索対象

業務 DEMO

ファイル形式 Java Class File,DB2 File,Function,HDB DBD GEN File...

検索

**絞り込み検索**

検索対象

業務 DEMO

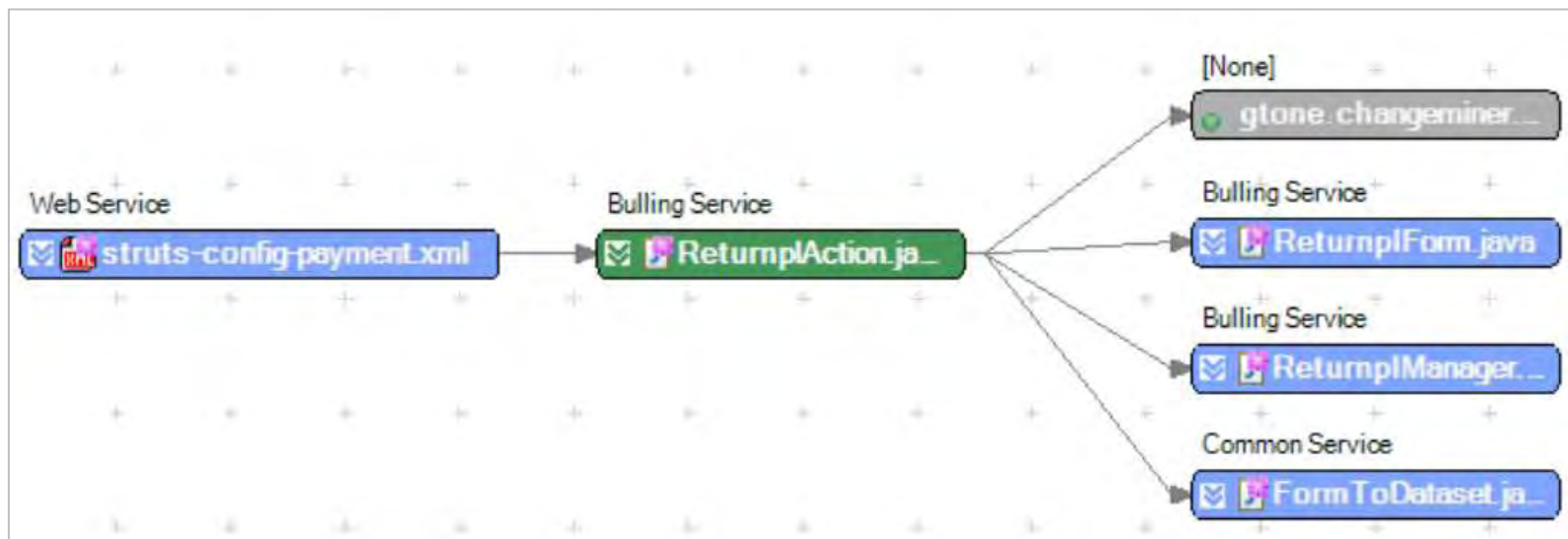
条件の追加

☒ And ☐ Or ☐ ソースコードの情報を表示 検索

検索項目	検索語	大/小文字区別	
キーワード	Sample	<input type="checkbox"/>	削除
オブジェクト名	Sample_Object	<input type="checkbox"/>	削除

## 機能 2 : 関連マップ

- プログラムからDB項目までの関連を可視化
  - 修正の影響範囲がどこまで及ぶのかを視覚的に把握
  - 仕様変更やトラブル箇所の説明などのコミュニケーションに活用



## 機能 2 : 関連マップ : デモ

- ソースコードの該当行までクリック操作でたどれる

The screenshot illustrates the '関連マップ' (Related Map) feature. It shows a table of related links and a corresponding source code editor.

**関連リンク (Related Links) Table:**

呼出元のオブジェクト	行番号	関連タイプ	呼出先のオブジェクトの...	呼出先のオブジェクト	CRUD	正確性	区分
ReturnplAction.java	45	Use Local Associat...	ReturnplForm.java	ReturnplForm			精度分析

**ReturnplAction.java Source Code:**

```
38 try
39 {
40     Logger.debug.println("***** PolicyreturnAction Start
41
42     DataSet input = new DataSet();
43     DataSet inputCount = new DataSet();
44
45     ReturnplForm returnplForm = (ReturnplForm)form;
46
47     FormToDataset.copyProperties(input, returnplForm);
```

Arrows indicate the navigation path from the table row (line 45) to the corresponding line in the source code editor.

# 機能 3 : CRUDマトリックス

- プログラムとDB間のCRUD関係を可視化
  - テーブルやカラム変更時に影響を受けるプログラムを把握
  - 関連があるプログラム数がわかるので、工数の概算見積に活用

CRUD-(demo java)

C R U D W Q 180

	プログラム	AGENCY				ALMAILM1				ALMAILM12				LINK...	LINK...
		AGENCYCODE	AGENCYNAME	AGENCYPHONENUMBER	LP_ID	MAILFROM	MAILTO	SEQIDX	TITLE	MAILFROM	MAILTO	SEQIDX	TITLE	COLUMN_ID	COLUMN_ID
▶ C:\COM_DEMO\java\	KosisSearch1Bean.java	_R_	_R_	_R_	_W_	_R_	_R_	_R_W_	_R_					_R_	
	KosisSearch1Bean2.java	_R_	_R_	_R_	_W_					_R_	_R_	_R_W_	_R_		_R_

# 機能 3 : CRUDマトリックス : デモ

## ● SQL文の実行コードを確認できる

The screenshot displays a demo application window titled "CRUD-(demo java)". It features a table with columns for "AGENCY" (AGENCYCODE, AGENCYNAME, AGENCYPHONENUMBER) and "ALMAJMI" (MAILFROM, MAILTO, SEQIDX, TITLE). Below the table, there is a section titled "CRUDマトリックス - KosisSearch1Bean2.java" which lists permissions for various actions (C, R, U, D, W, Q) for different entities. A red box highlights the "行番号" (Line Number) field with the value "355". An arrow points from this box to a corresponding line in the code editor, which is also highlighted with a red box. The code editor shows the following SQL query preparation:

```

344
345     m_sql.append("SELECT A.AGENCYCODE,B.COLUMN_ID,A.AGENCYNAME,A.AGENCYPHONENUMBER  \n");
346     m_sql.append("FROM AGENCY A, LINKTABLEJ B\n");
347     m_sql.append("WHERE A.LP_ID='" + table_id + "' ORDER BY A.LPCODE");
348
349
350
351
352     if(KosisInfo.PRINT_QUERY) {
353         System.out.println("KosisSearch1Bean.getClassInfo = [" + m_sql.toString() + "]");
354     }
355     m_pstat = m_con.prepareStatement( m_sql.toString() );
  
```

# ChangeMinerで的確な影響調査を実現！

- 全体の影響範囲を視覚的に把握！
- 修正対象のプログラムを迅速に把握！

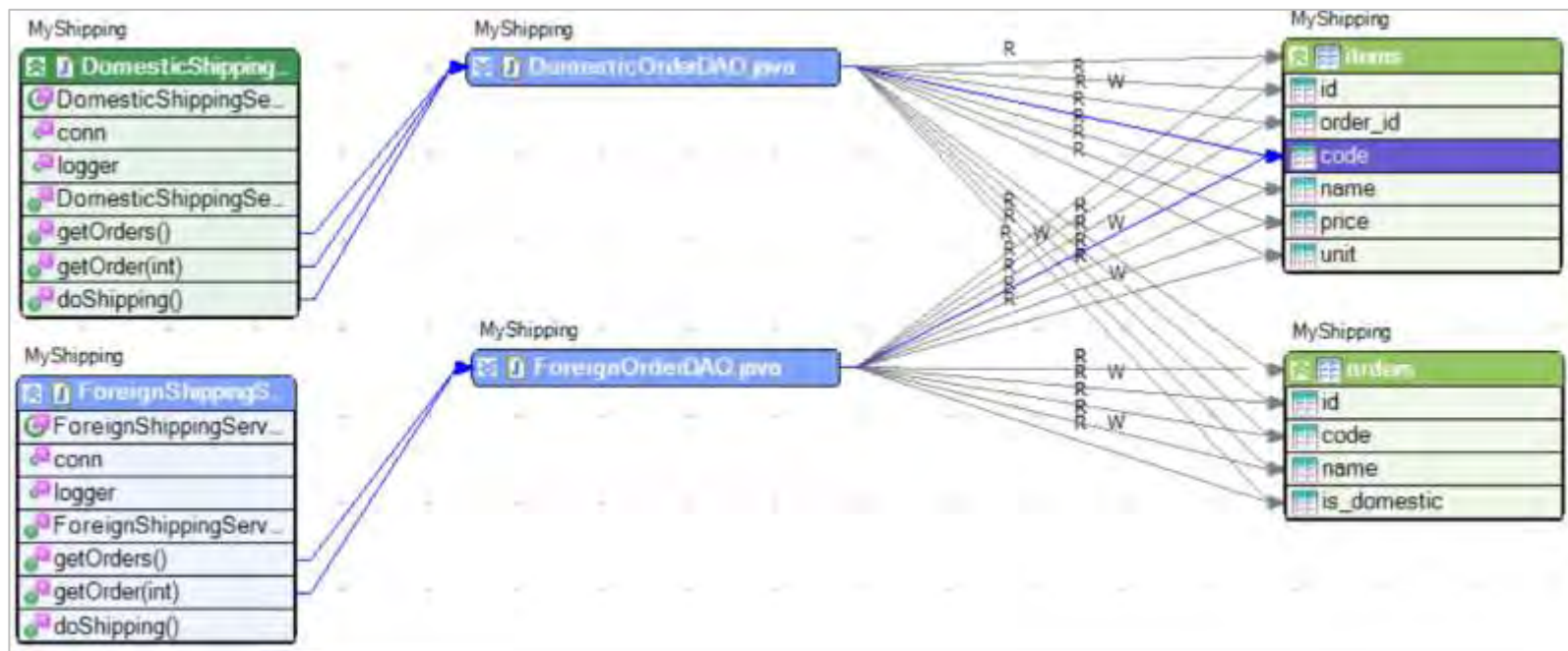
某カード会社では  
基幹システムの影響調査作業時間を  
grepと比較して**平均46.3%削減**



# ChangeMinerの具体的な活用例

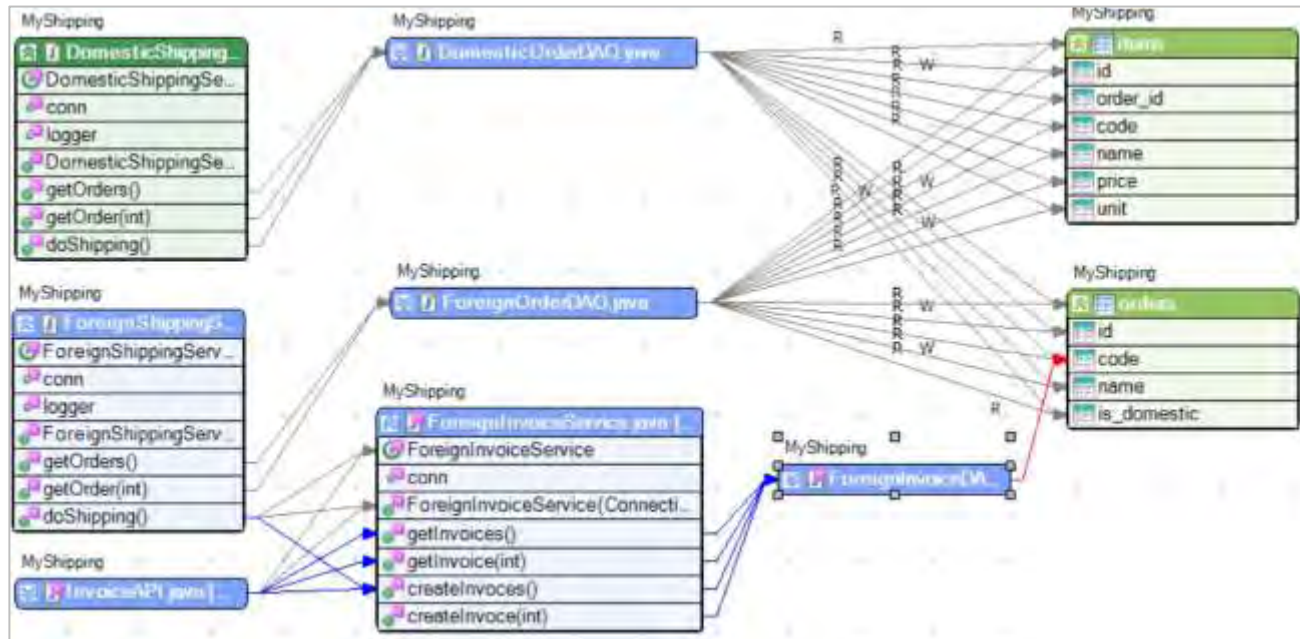
# ケース 1 : 国内/海外出荷プログラムの修正

- 業務拡大で商品コードの桁数を増やすこととなり、国内出荷/海外出荷プログラムの変更が発生。



# ケース 1 : 国内/海外出荷プログラムの修正

- ChangeMinerで商品コード項目を起点として確認すると、インボイス処理も修正が必要なことが判明。



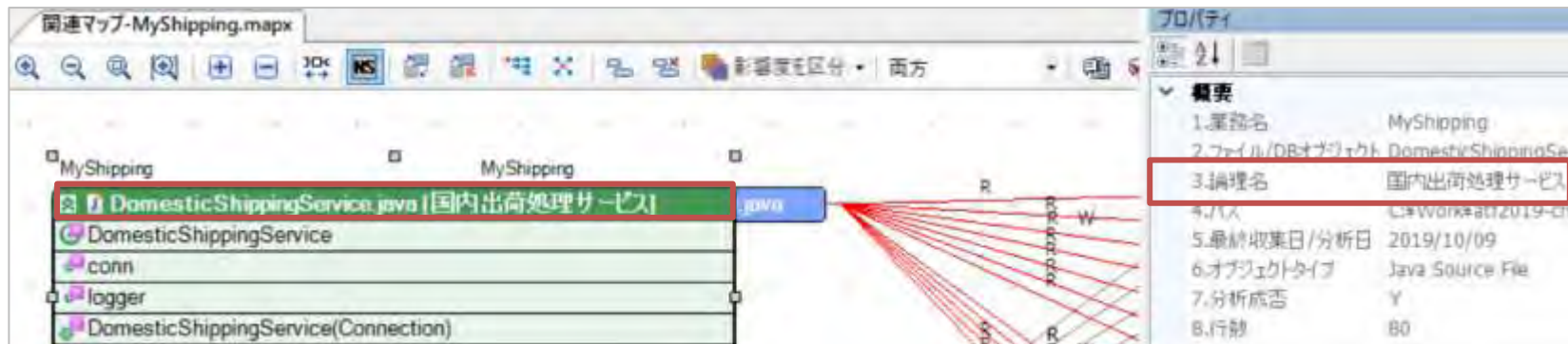
# ケース1：国内/海外出荷プログラムの修正

---

**DB項目を起点として関連を可視化し、  
修正対象プログラムの調査漏れを防止**

## ケース2：工数見積のエビデンスとして活用

- 関連マップに物理名だけでなく、論理名も表示。



# ケース2：工数見積のエビデンスとして活用

- グリッド表示にして、そのまま見積に添付する資料として活用

The screenshot shows a software development environment with a grid view of a project structure. The grid is organized into four levels:

Level 1	Level 2	Level 3	Level 4
InvoiceAPI.java [海外出荷インボイス用API] (MyShip...	ForeignInvoiceService.java [海外...	ForeignInvoiceDAO.java [海外出...	( R ) orders (MyShipping)
ForeignShippingService.java [海外出荷処理サー...	ForeignOrderDAO.java [海外出荷...	( R W ) orders (MyShipping)	
		( R W ) items (MyShipping)	
DomesticShippingService.java [国内出荷処理サー...	DomesticOrderDAO.java [国内...		

An arrow points from this grid to an Excel spreadsheet titled 'export\_list.xlsx'. The spreadsheet contains the same grid data, showing the export of the project structure for documentation purposes.

## ケース2：工数見積のエビデンスとして活用

**資料作成工数を削減し、  
ユーザと画面名や機能名での会話ができる**

# ChangeMinerの活用により

---

- **影響調査にかける時間と工数を短縮**  
クリック操作で関連する画面やプログラム、テーブル/カラムを把握できる。
- **調査漏れによる障害を防ぐ**  
万一障害が発生しても、どのテーブルや業務に影響が及ぶのかを素早く把握し、迅速に対応できる。
- **適切なテスト範囲を見極められる**  
修正範囲が明確になるので、テスト範囲を限定できる。



# ChangeMinerの検討ステップ°

# ChangeMinerを徹底体験

---

## 1 製品詳細の紹介とヒアリング

ChangeMinerの詳細機能のご紹介と併せて、お客様環境と分析要求をヒアリングし、ご要望への対応可否の認識合わせをします。

## 2 PoC（有償）の実施

お客様のシステムや分析要求に合わせてチューニングやカスタマイズを行い、ChangeMinerがお客様のシステムで効果を発揮するかを見極めます。

# ご清聴ありがとうございました

本資料に関するお問い合わせ

株式会社アシスト 社長室

gtone@ashisuto.co.jp

超	サ	ポ					
愉	快	カ	ン	パ	ニ	ー	

# アシスト

※本資料に記載している情報は、2019年10月17日現在のものです。

※本資料の内容は、今後予告なく変更されることがあります。

※文中の社名、商品名等は各社の商標または登録商標である場合があります。