

# 業務システム構築基盤のご紹介 SystemDirector Enterprise for Batch

日本電気株式会社

本資料では SystemDirector Enterprise for Java Batch / for C Batchを、「for Batch」と略記することがあります。

# Orchestrating a brighter world

未来に向かい、人が生きる、豊かに生きるために欠かせないもの。  
それは「安全」「安心」「効率」「公平」という価値が実現された社会です。

NECは、ネットワーク技術とコンピューティング技術をあわせ持つ  
類のないインテグレーターとしてリーダーシップを発揮し、  
卓越した技術とさまざまな知見やアイデアを融合することで、  
世界の国々や地域の人々と協奏しながら、  
明るく希望に満ちた暮らしと社会を実現し、未来につなげていきます。

# 目次

## 第 1 章 SystemDirector Enterprise とは

- 1.1. SystemDirector Enterprise の狙い
- 1.2. SystemDirector Enterprise の体系
- 1.3. SystemDirector Enterprise の構成要素
- 1.4. SystemDirector Enterprise の製品ラインナップ

## 第 2 章 開発方法論

- 2.1. 概要
- 2.2. 一般的な開発方法論との違い
- 2.3. 共通フレーム対応
- 2.4. フロントローディング

## 第 3 章 開発環境

- 3.1. 提供機能のポイント
- 3.2. アーキテクチャモデル
- 3.3. フレームワーク
- 3.4. 開発ツール
- 3.5. 動作環境

## 第 4 章 サポートサービス

- 4.1. サポートメニュー
- 4.2. 教育メニュー
- 4.3. 有償サービス（バッチ標準化支援）
- 4.4. お問い合わせ先

## 付録 関連製品

# 第1章 SystemDirector Enterprise とは

## 1.1.SystemDirector Enterprise の狙い

### 環境認識

技術の高度化、  
新技術の台頭

短納期・  
スピード重視

長期間の  
保守メンテナンス

### ニーズ

「信頼できるSI」

先端技術の活用  
迅速な対応

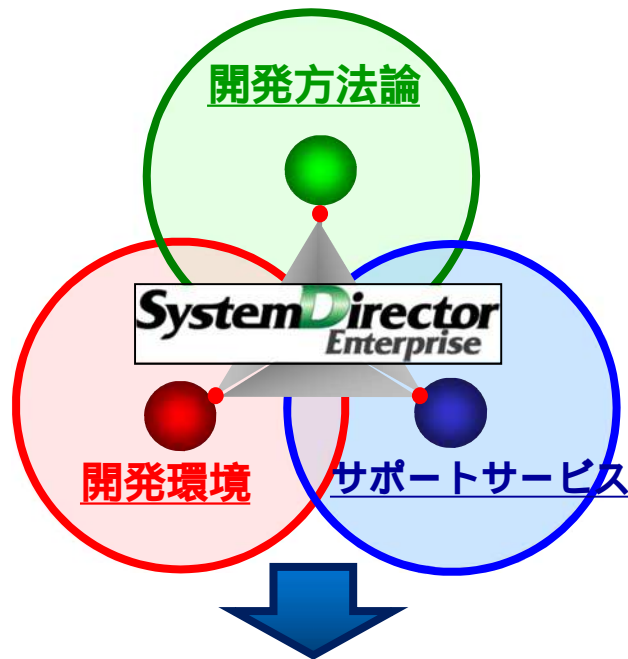
生産性の向上

十分な期間の  
サポート提供

これらを実践するシステム構築環境を整備  
今までのNECグループのシステム構築技術を結集  
SystemDirector Enterprise の開発へ

## 1.2.SystemDirector Enterprise の体系

NECグループ標準の業務システム構築基盤です。開発方法論、開発環境、サポートサービスによる効率的なシステム構築を支援します。



### 開発方法論（誰が、いつ、何をするのか）

- 概説書、手順書、ガイド、ドキュメントサンプル集
- 各工程の作業手順をサポート

### 開発環境（何を作るか、どうやって作るのか）

- 業界標準アーキテクチャを採用したフレームワーク
- 開発効率を向上する開発ツールの提供

### サポートサービス（どのように利用するか）

- システム構築をさまざまな形でサポート
- トータル10年間のソフトウェア製品サポートを提供

### 先端技術の活用

- ・ 開発環境で取り込みガイドも添えて提供

### 生産性の向上

- ・ 開発方法論で標準作業の効率化
- ・ 開発環境で生産性向上

### サポート提供

- ・ 専任の技術サポート部隊を用意
- ・ 長期サポート保証

## 1.3.SystemDirector Enterprise の構成要素

業務システムを構築するために必要十分な支援内容を提供します。



誰が、いつ、  
何をするのか

開発方法論

SystemDirector  
Enterprise

開発環境

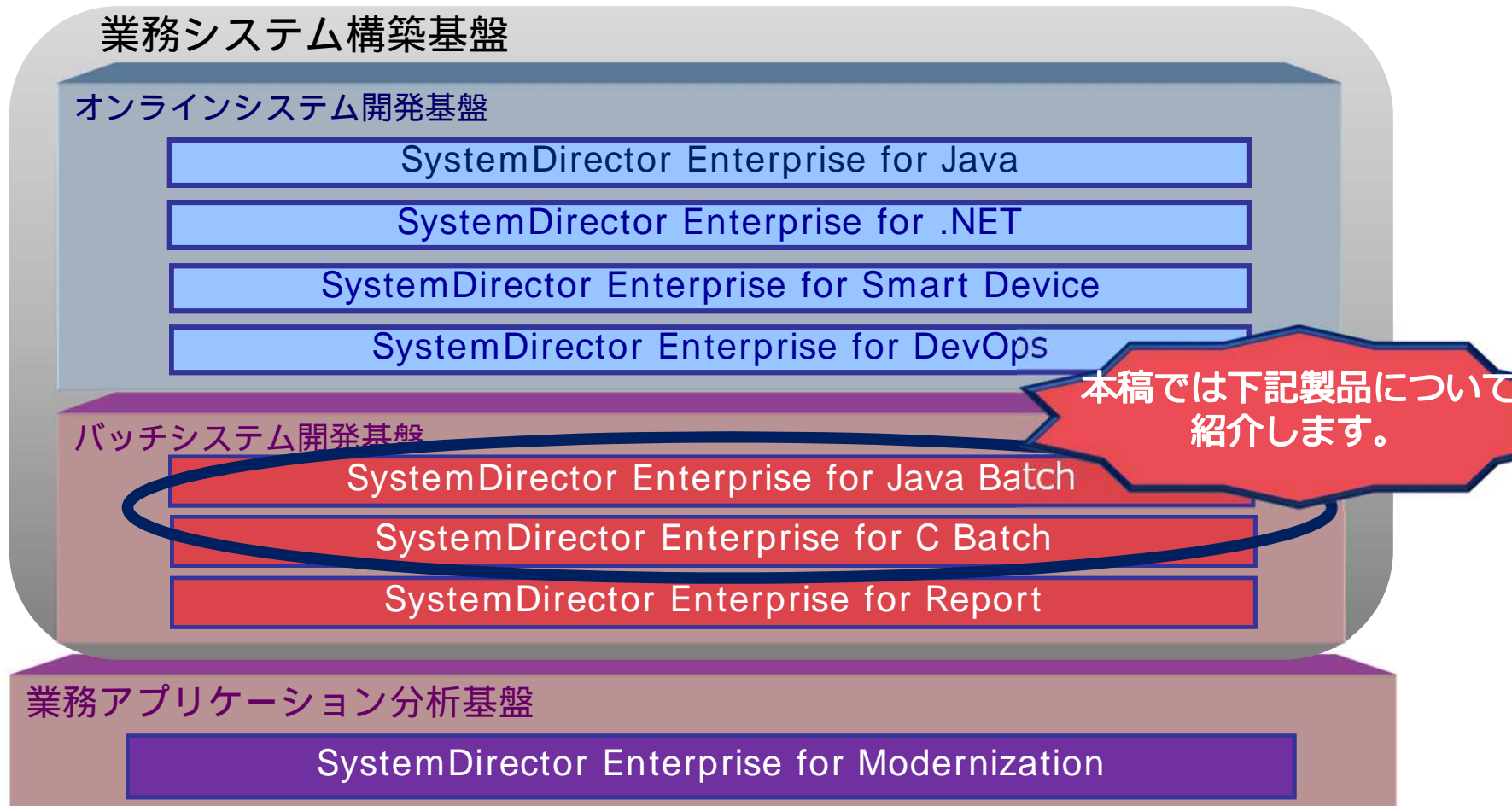
サポートサー  
ビス

何を作るか  
どうやって  
作るのか

どのように  
利用するか

## 1.4.SystemDirector Enterprise の製品ラインナップ

- 要件定義～テストをサポートする業務システム構築基盤と、保守～再構築をサポートする業務アプリケーション分析基盤を提供
- ソフトウェアライフサイクル全般をサポートする製品群として継続強化中





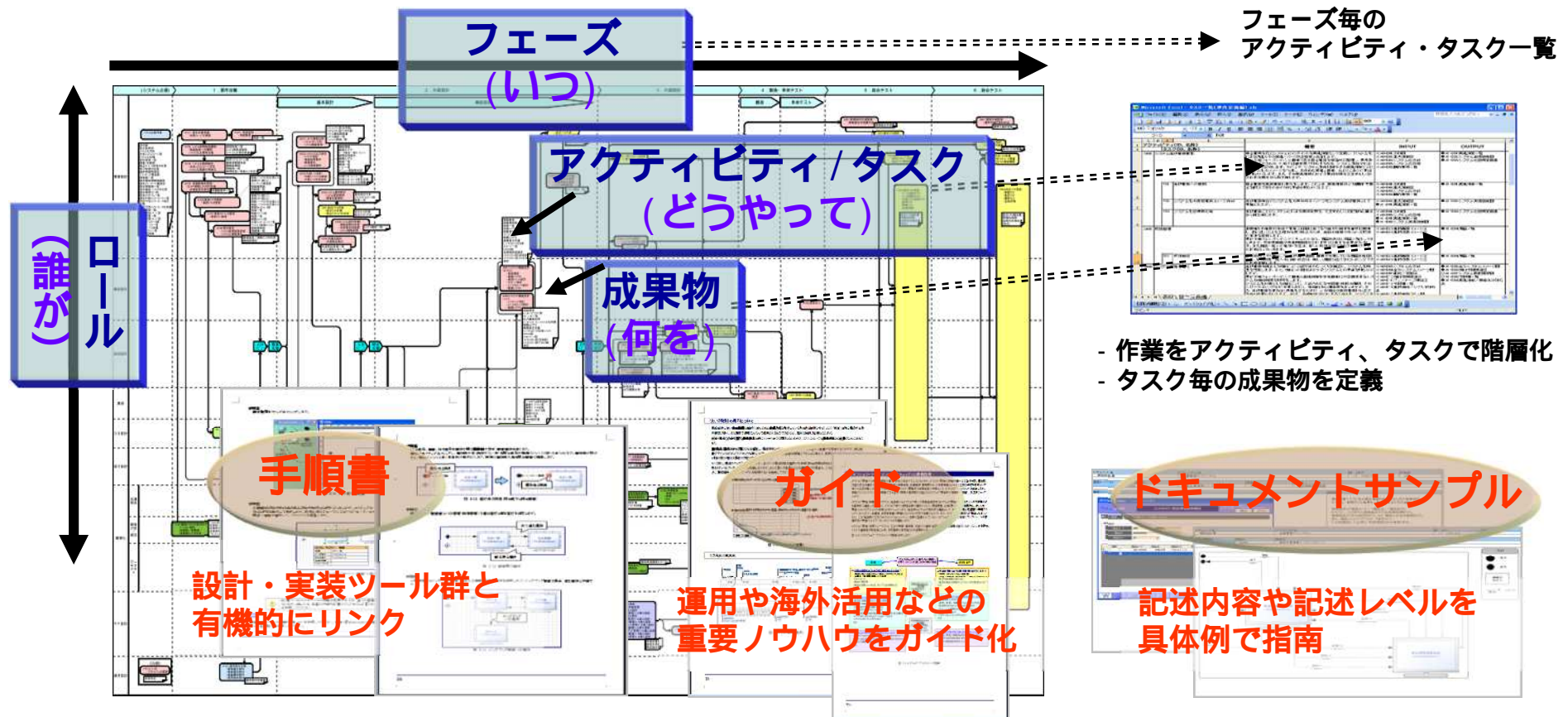
## 第 2 章 開発方法論

## 2.1.概要

開発プロセスに従った手順書、ドキュメントサンプルなどを提供します。

### 開発プロセス

SI現場の成功・失敗ノウハウを実践的な開発プロセスに結集  
SIに必要な「いつ・誰が・何を・どうやって」を漏れなく重複なく体系化



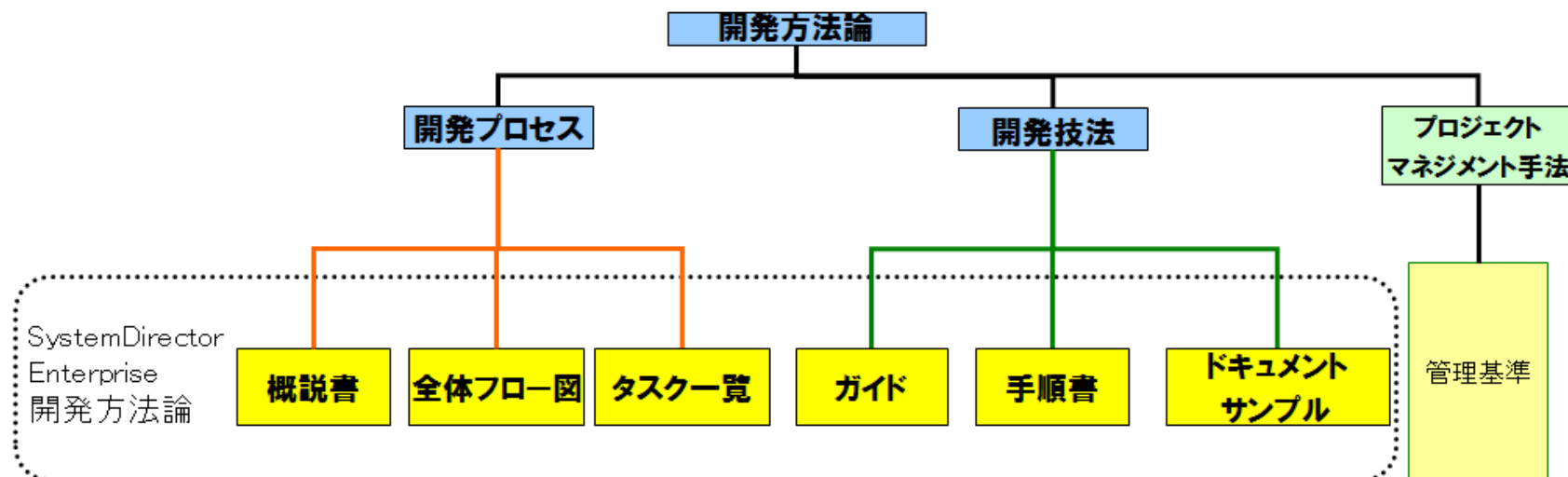
## 2.2.一般的な開発方法論との違い

役割を意識した、手順、成果物、規約などを体系的に定義します。

一般的に開発方法論は、以下の3つで構成されます。

- オブジェクト指向やデータ中心アプローチなどの具体的な開発手法やツール類を定めた「開発技法」
- 開発をどのような手順で進めるかを定めた「開発プロセス」
- プロジェクトマネジメント手法

SystemDirector Enterprise開発方法論は、オープン環境における業務システム開発をターゲットとして、ソフトウェアライフサイクルプロセス（SLCP）における「要件定義」「外部設計」「内部設計」「製造・単体テスト」「結合テスト」「総合テスト」「移行」の一連の開発プロセスを中心に、役割（ロール）ごとの作業手順、成果物（ドキュメント）、作業規約を体系的に定めた開発方法論です。



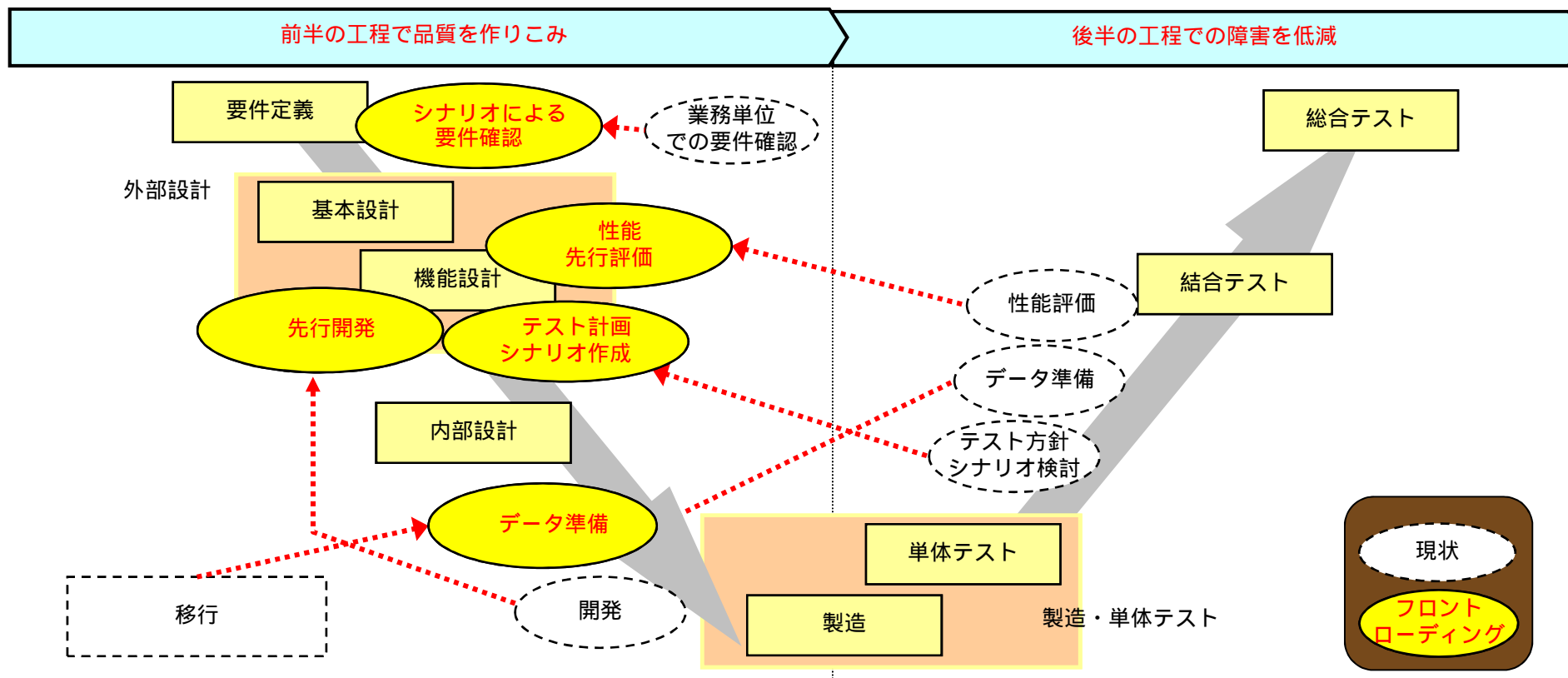


## 2.4.フロントローディング

開発の前半で品質を作りこむ手法を取り入れています。

「フロントローディング」とは？

開発の後半で発生する問題を未然に防ぐことを目的に、開発初期でやるべきことを正しく行い品質を作りこむ手法



## 第3章 開発環境



## 3.1.提供機能のポイント

バッチアプリケーションの開発を支援する各種機能を提供します。

Point  
1

### アーキテクチャモデル

- これまで弊社で構築した数多くのバッチシステムで採用された実績あるバッチフレームワークを集約し、SystemDirector Enterprise推奨のアーキテクチャモデルとして提供することで、アーキテクチャ選定の検討工数と構築工数を削減。

Point  
2

### フレームワーク

- バッチアプリケーションの枠組みのデフォルトを提供、また、プロジェクト毎にカスタマイズ可能。定義された枠組みは全バッチアプリケーションに強制的に適用されるが業務ロジックとは独立、業務SEは業務ロジックに注力可能。

Point  
3

### 開発ツール

- 実績のある SystemDirector Enterprise開発基盤のノウハウを生かし、開発方法論をベースに バッチアプリケーションの作成を支援するジェネレータ（設計情報からのソースコード生成やドキュメント生成）等の開発環境を提供

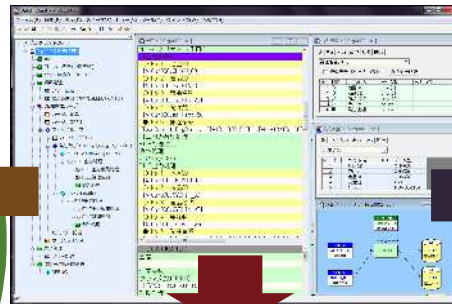
## 3.2.アーキテクチャモデル

適用範囲の広いバッチアプリケーション・テンプレートから、ソースコードを自動生成します。

バッチ開発のノウハウを集約した3系統のアプリケーション・テンプレートを提供します。

### 入力駆動表ループ系

DB表やファイルのレコードを順々に読み込み、1レコードずつ処理。  
【特長】  
1レコード毎の処理を細かく設計。



コントロールブレーク系  
ソートされたレコードを順々に処理しつつ、キー項目の値が変わった（ブレークした）時に追加で処理をできるように制御。  
【特長】  
大量の帳票用データ作成に有効。

### 一括処理系

ループ処理なしに任意の処理を実行。  
【特長】  
自由度が高く、1命令を処理するだけのものから、ループ処理を駆使した複雑な処理構造の構築が可能。

この基本構造を元に、業務処理に応じてプログラム構造を柔軟に拡張可能。  
すぐに『動くソース』を生成し、生成率は7割以上。

### 基本構造

初期処理

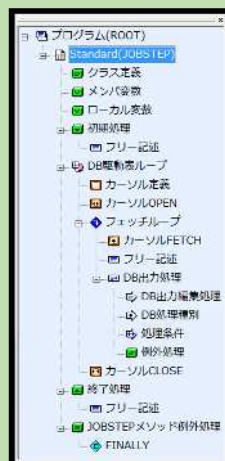
メインループ

入力処理

出力処理

終了処理

例外処理



### 拡張後

初期処理

LookUp処理

DB更新処理

メインループ

入力処理

条件判定

LookUp処理

INSERT処理

条件判定

終了処理

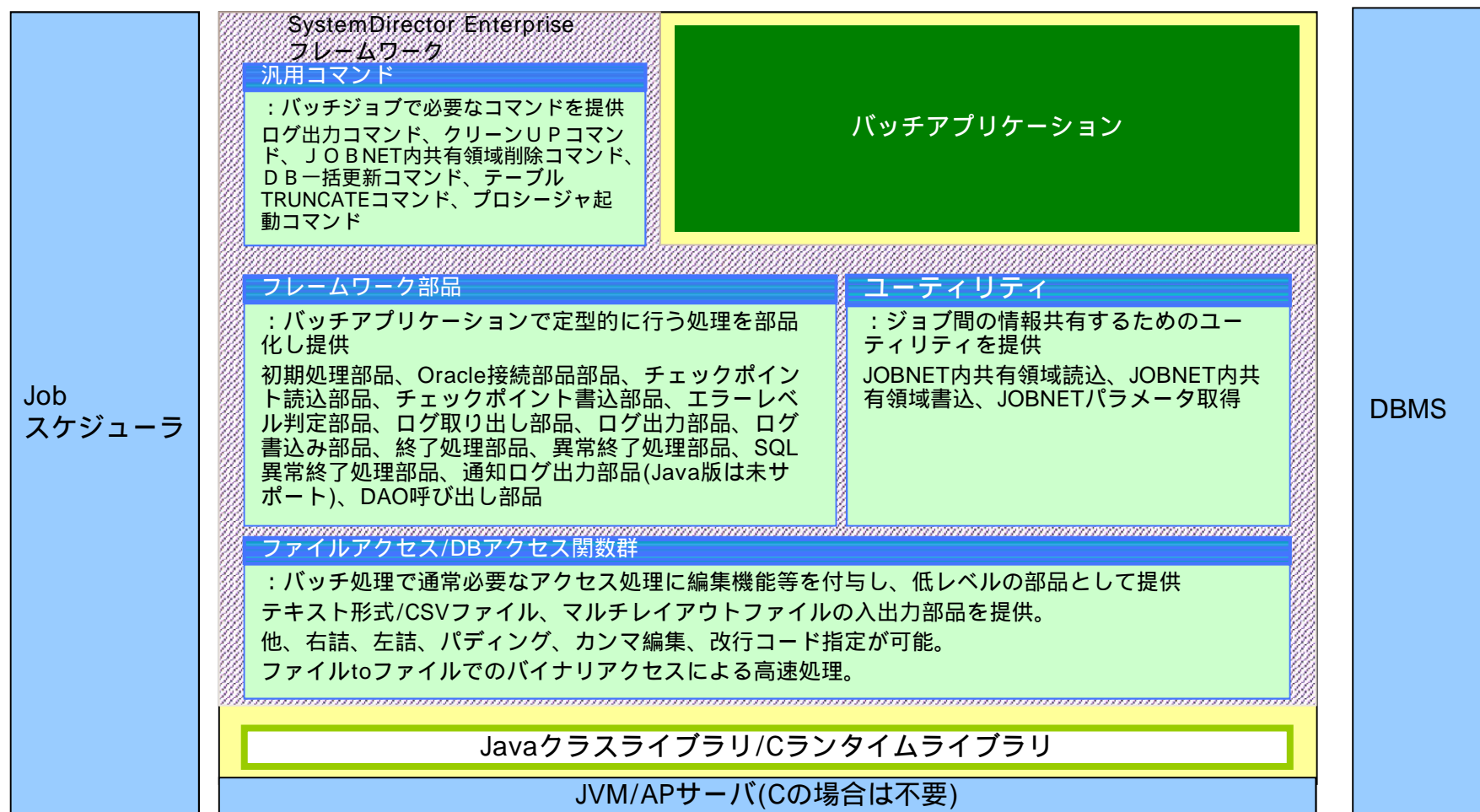
例外処理





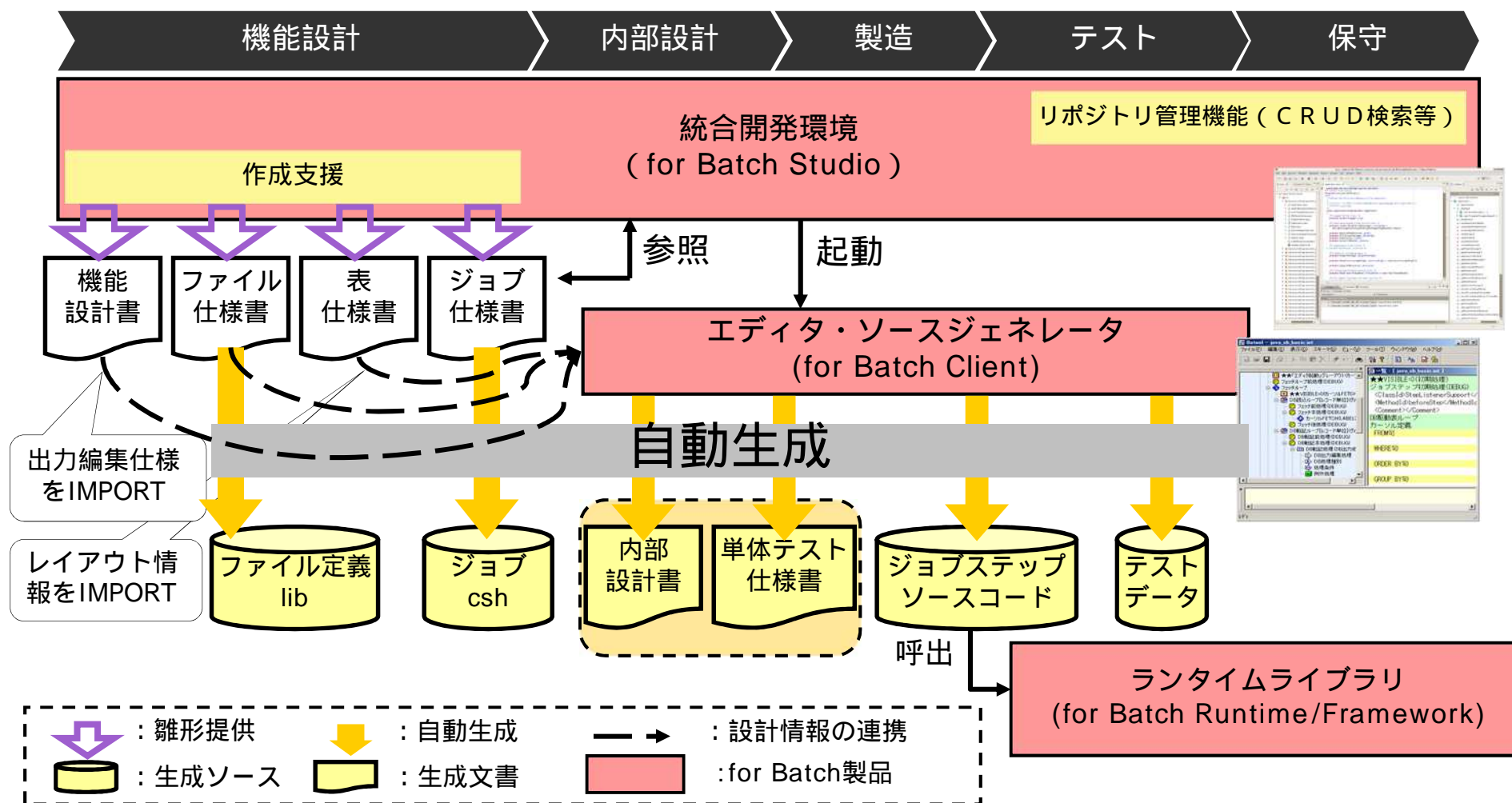
### 3.3.フレームワーク

バッチジョブ/バッチアプリケーション構築に通常必要なコマンド/APIを提供することで、業務SEは業務ロジックに注力可能です。



### 3.4.開発ツール : 開発ツールと開発工程

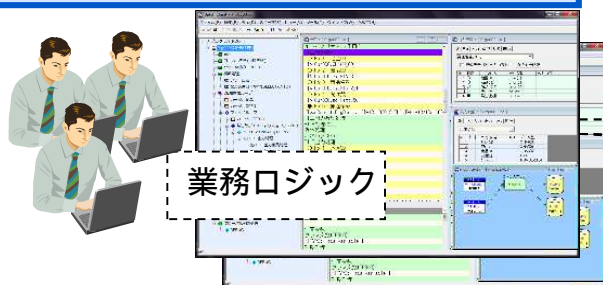
開発手順をベースに、各開発工程で開発環境（各種ツール群）がバッチシステム開発をサポートします。



### 3.4.開発ツール :カスタマイズ可能なバッチフレームワーク

バッチフレームワークのカスタマイズが可能、かつ、ソースコード生成時に強制的にバッチフレームワークの処理を埋め込むことでソースコードの品質を均一化する仕組みを提供します。

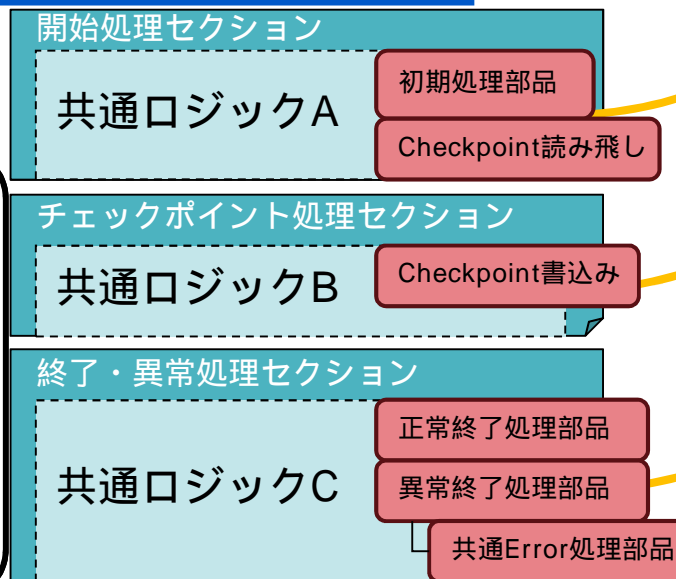
業務チームは業務ロジック設計に専念



基盤チームがバッチフレームワークを設計



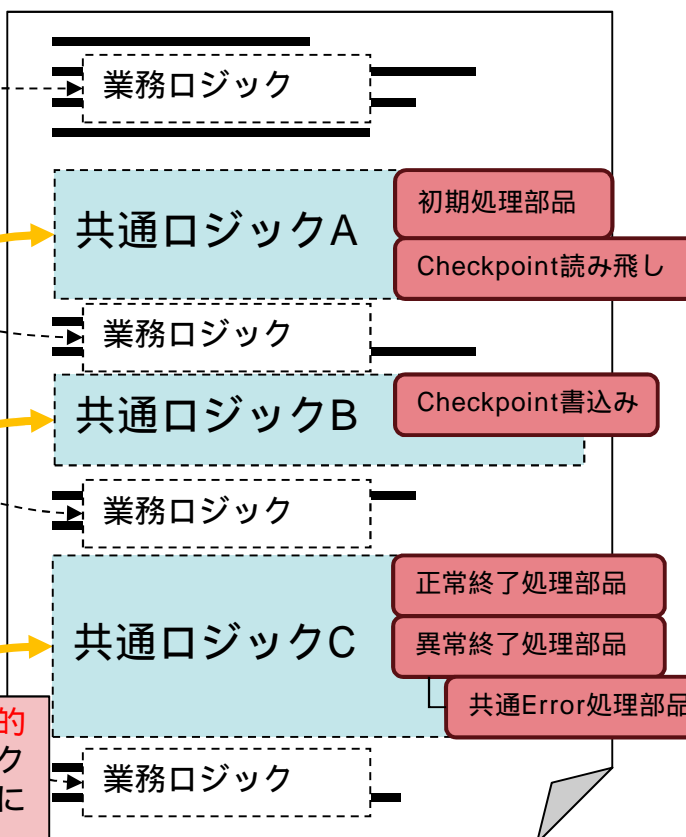
カスタマイズポイントがセクションごとに定義されており、基盤チームは各セクションの処理を追加・変更すればよい。



生成

生成時に、強制的に、業務ロジック間の適切な場所に埋め込まれる

(ソースコード)



## 3.4.開発ツール：開発環境の概要

バッチ系アプリケーション開発を効率化する各種エディタ群、ツール群、ランタイム群を統合化し、開発工程間の作業をシームレスに連携した開発環境です。過去10年以上に渡り、自治体、製造業、流通業などのお客様に利用されており、大規模案件の事例も多数あります。

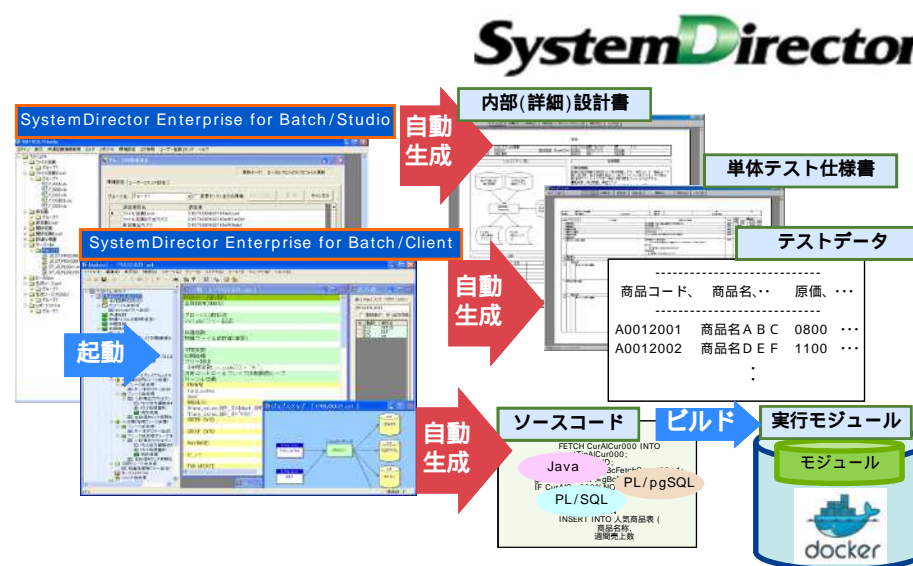
### 主な機能と導入効果

#### 1. バッチ系システム開発(SI)の効率化

- ビジュアルなアプリケーション設計を可能とし、簡単な操作で**ソースコードを自動生成**。
- 生成対象言語は、**Java**、**Pro\*C/C++**、**PL/SQL**、**PL/pgSQL**、**ECPG**をサポート。
- **jBatch(JSR352)**に対応、**WebOTX Application Server**や**GlassFish**や**WildFly**上でバッチが稼働
- Oracle/PostgreSQL/MySQL/SQL Server/RDS Aurora
- 実行環境として**Docker**コンテナをサポート、**AWS**活用と合わせて運用コストを削減。

#### 2. バッチAPの品質・実行性能の向上

- **フレームワークに依存しない柔軟な設計**ができ、PJ毎のフレームワークカスタマイズも可能。
- Javaバッチに特化した**高速DB/ファイル処理**ライブラリを提供、**JavaとCで同等の実行性能**。



#### 3. バッチAPの保守性の向上

- **バッチ詳細/テスト仕様書を生成し**、ソースと仕様書の一貫性を維持。
- プロセス指向の処理方式を採用、**現行資産移行も容易**。
- Cバッチ資産を**Javaバッチに移行**する支援機能を提供。

## 3.4.開発ツール：操作性に優れたGUIと強力な生成機能

バッチアプリケーションの業務処理をGUIで実装し、ソースや設計書を自動生成します。

言語固有のルールをソースジェネレータが隠蔽し、開発者は業務処理(ビジネスロジック)の設計/実装に専念できます。

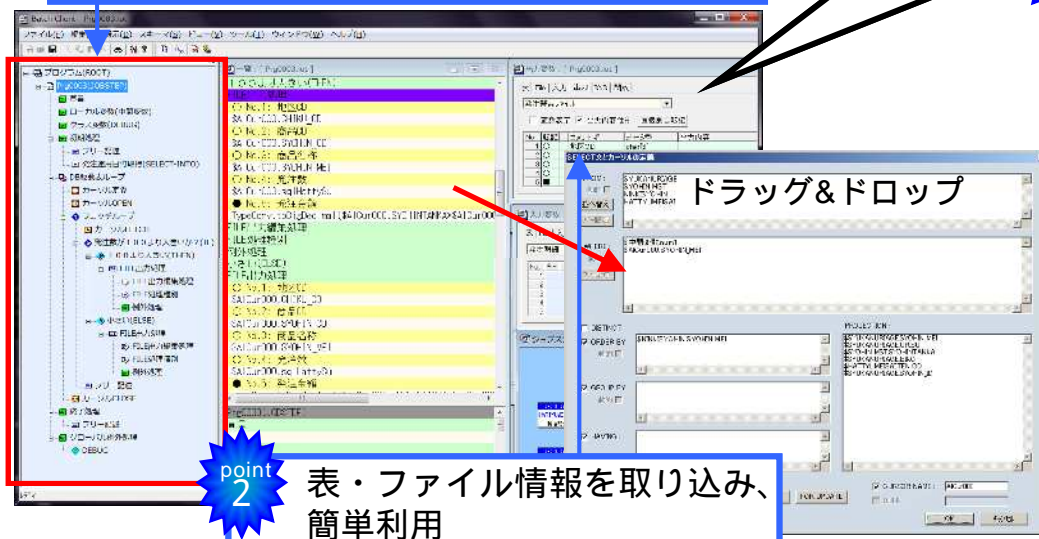
point 1

バッチアプリケーションのプログラム構造をノードの階層構造で表現。ノードの編集も簡単

個々の「ジョブステップ」を設計・製造する専用のクライアントツール。

point 3

編集不要のC/Javaソースをワンクリックで自動生成。  
SQL発行部分だけでなく、プログラム構造も生成。生成ソースはそのままコンパイルし、実行が可能。



生成

Javaソース  
ファイル

Cソース  
ファイル

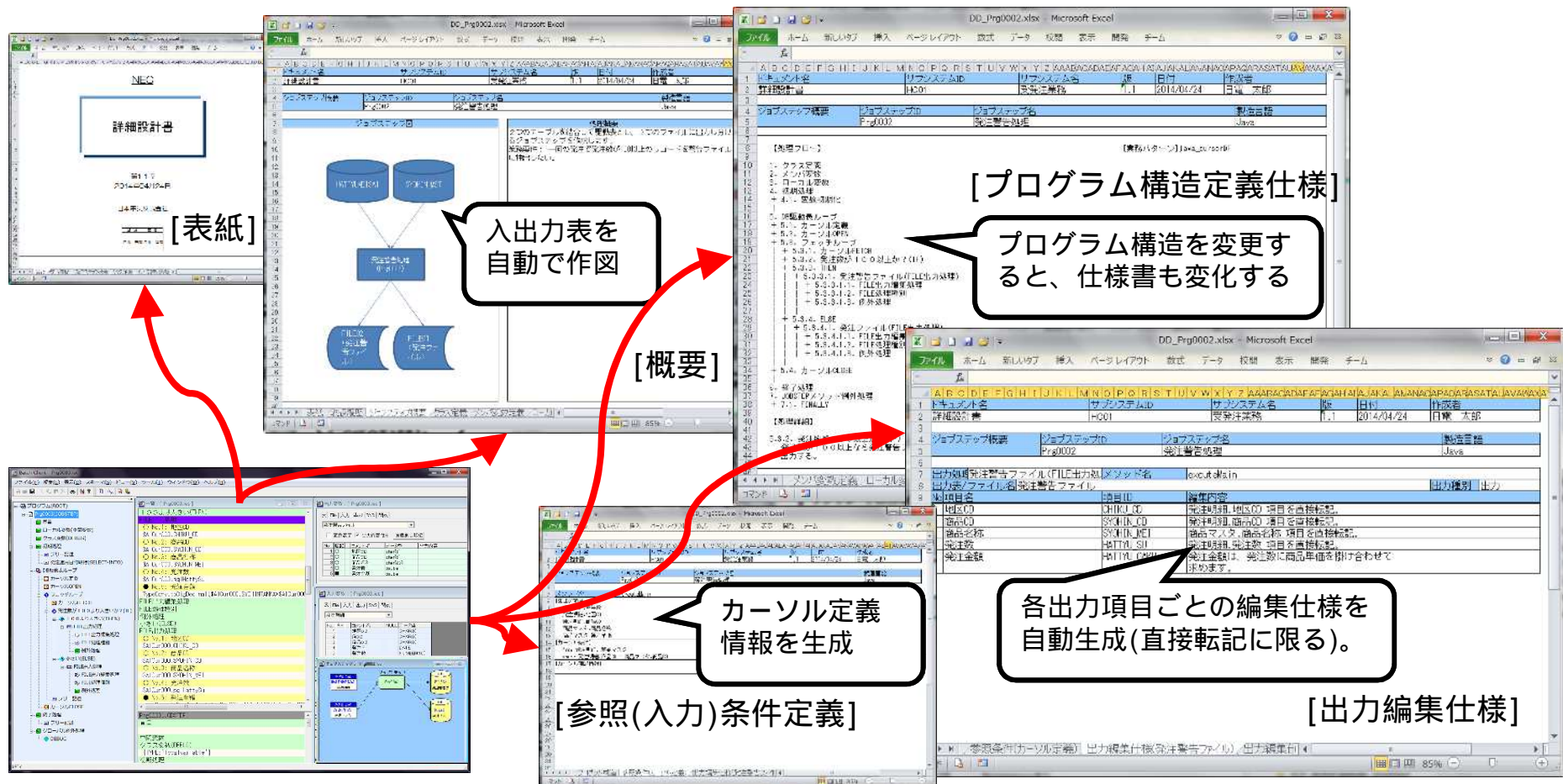
成果物の品質・書式の一貫性が得られるため、  
特に大規模プロジェクトで効果を発揮



## 3.4.開発ツール：ドキュメント自動生成

ソースだけではなく、詳細設計書を生成し、成果物の整合を維持します。

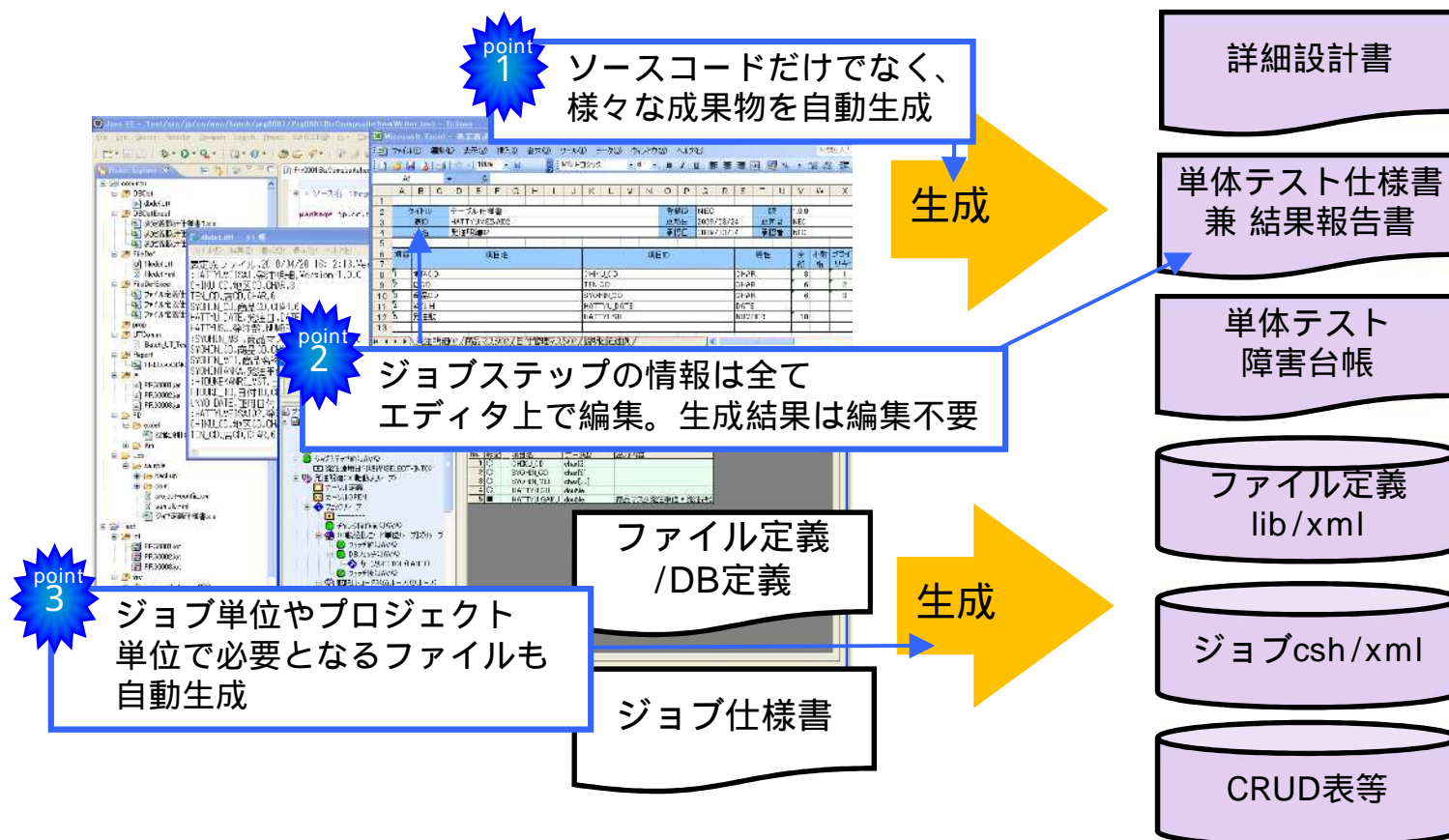
プログラム構造や出力編集仕様のドキュメントを、ソース生成時と同じ情報を元に生成することで、ソースと仕様書の乖離を防ぎます。



## 3.4.開発ツール : 成果物の管理(1/2)

成果物の情報の連携・整合維持ができ、保守性を向上します。

ジョブステップの情報は1つのファイルで一元管理。成果物間の情報の乖離を防止  
詳細設計書、単体テスト仕様書等も一括で自動生成。手入力や二度打ちを排除  
一元管理した情報から、レイアウト変更時の影響特定やCRUD表生成等、保守情報の活用が可能



## 3.4.開発ツール : 成果物の管理(2/2)

### 各種成果物を一括で管理

機能設計書、詳細設計書、ジョブステップソース、XMLファイル、単体テスト仕様書、表定義/ファイル定義仕様書。

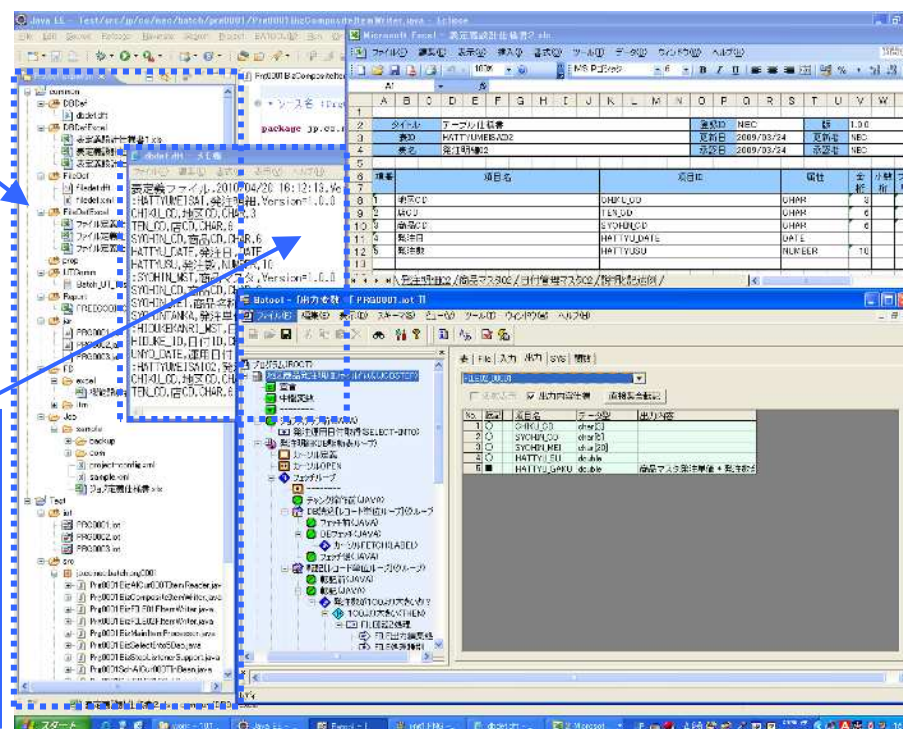
リポジトリ機能により複数の成果物から情報を抽出し、CRUD表生成、影響範囲検索、ソース規模/生成率測定。また、各種生成物を、一括生成できる。

point  
1

統合環境で成果物を管理。  
Java版の生成コードは、  
Eclipseの機能でデバッグ  
やコード検証が可能。

point  
2

表定義/ファイル定義仕様  
書から情報を抜き出し、  
機能設計/詳細設計書に情  
報を引き継ぎ。  
更に、生成コードにまで  
反映。



point  
3

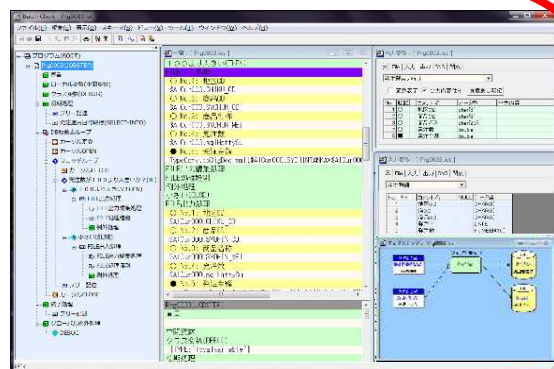
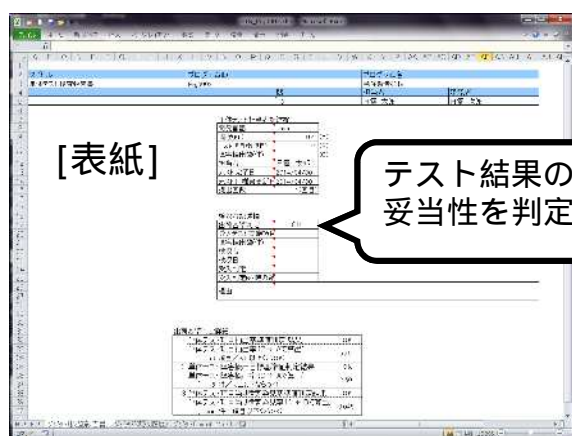
CRUD表生成など、複数の  
ジョブステップの情報  
を集計可能。



## 3.4.開発ツール : テスト項目の自動生成

加えて、単体テスト仕様書も生成し、テスト項目を自動で生成。

プログラム構造から一部テスト仕様を自動生成。加えて、開発者が自分で設計したテスト項目を合わせ、ひとつのテスト仕様書に生成。ひとつのバッチ処理の設計情報を一元管理。

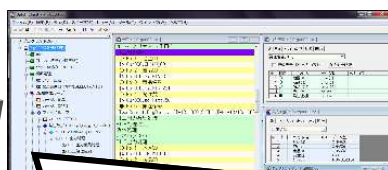


項目	テストレベル	テスト項目	期待される結果	実行日時	担当者	確認結果
1	1	■実行処理の確認				
2	1	■文字列型変数の初期化	文字列型変数が""で初期化されていること。	4月1日	OK	
3	1	■数値型変数の初期化	数値型変数が0で初期化されていること。	4月1日	OK	
4	1	■ループ処理の確認				
5	1	■ループ条件の確認				
6	1	■ループ条件の確認				
7	1	■ループ条件の確認				
8	1	■ループ条件の確認				
9	1	■ループ条件の確認				
10	1	■ループ条件の確認				
11	1	■ループ条件の確認				
12	1	■ループ条件の確認				
13	1	■ループ条件の確認				
14	1	■ループ条件の確認				
15	1	■ループ条件の確認				
16	1	■ループ条件の確認				
17	1	■ループ条件の確認				
18	1	■ループ条件の確認				
19	1	■ループ条件の確認				
20	1	■ループ条件の確認				
21	1	■ループ条件の確認				
22	1	■ループ条件の確認				
23	1	■ループ条件の確認				
24	1	■ループ条件の確認				
25	1	■ループ条件の確認				
26	1	■ループ条件の確認				
27	1	■ループ条件の確認				
28	1	■ループ条件の確認				
29	1	■ループ条件の確認				
30	1	■ループ条件の確認				
31	1	■ループ条件の確認				
32	1	■ループ条件の確認				
33	1	■ループ条件の確認				
34	1	■ループ条件の確認				
35	1	■ループ条件の確認				
36	1	■ループ条件の確認				
37	1	■ループ条件の確認				
38	1	■ループ条件の確認				
39	1	■ループ条件の確認				
40	1	■ループ条件の確認				
41	1	■ループ条件の確認				
42	1	■ループ条件の確認				
43	1	■ループ条件の確認				
44	1	■ループ条件の確認				
45	1	■ループ条件の確認				
46	1	■ループ条件の確認				
47	1	■ループ条件の確認				
48	1	■ループ条件の確認				

## 3.4.開発ツール : バッチアプリケーションの高速化

バッチアプリケーションの実行性能を向上するための様々な機能を提供。

RDBMSアクセスにて複数レコードを同時に処理するコードを自動生成



生成

```
package sample;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import org.springframework.batch.item.ItemProcessor;

public class SampleItemProcessor<T> implements ItemProcessor<T, T> {
    private int count = 1;
    private List<Object> outList01 = new ArrayList<Object>();
    private List<Object> outList02 = new ArrayList<Object>();
    private Map<String, Object> parameter;

    public Map<String, Object> getParameter() {
        return parameter;
    }

    public void setParameter(Map<String, Object> parameter) {
        this.parameter = parameter;
    }
}
```

業務処理を設計し、ソースを生成すると、入力・出力レコードを数千件のレコード単位で分けて一括で処理するコード(バルク処理)を生成し、高速実行。

RDBMSチューニングのための細かな設定が可能



ヒント句の記述、SELECT時のテーブル記載順の変更、バルク処理のレコード単位を個々に指定できるなど、性能向上のための細かな設定が可能。

ファイルからファイルに情報を転送する処理で、文字コードを意識しないバイナリーコピーを行うライブラリを提供



```
package sample;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import org.springframework.batch.item.ItemProcessor;

public class SampleItemProcessor<T> implements ItemProcessor<T, T> {
    private int count = 1;
    private List<Object> outList01 = new ArrayList<Object>();
    private List<Object> outList02 = new ArrayList<Object>();
    private Map<String, Object> parameter;

    public Map<String, Object> getParameter() {
        return parameter;
    }

    public void setParameter(Map<String, Object> parameter) {
        this.parameter = parameter;
    }
}
```

リンク

ランタイムライブラリ  
(for Batch Runtime)

項目の転記において「バイナリーモード」を選択すると、バイナリーコピーを行うライブラリとリンクし、実行時に文字コード変換を行わずに高速実行。(Java版)

## 3.4.開発ツール :ランタイムライブラリ

汎用的な処理をライブラリ化し、品質を向上します。

ファイルアクセスにおいて一般的な文字列編集(CSVやマルチレイアウト等)を行うメソッドを準備し、生成コードから自動呼出し

ログ出力やチェックポイント処理など、汎用的なフレームワーク部品(関数/コマンド)を提供、更にこれら部品を生成コードに自動展開し、フレームワークを各バッチアプリケーションに反映

Javaでは、高速ファイルアクセスライブラリや専用JDBCライブラリを提供



汎用的な部品を、プロジェクト毎に作るのではなく、既存部品を適用することで、品質向上と効率化が図れる。

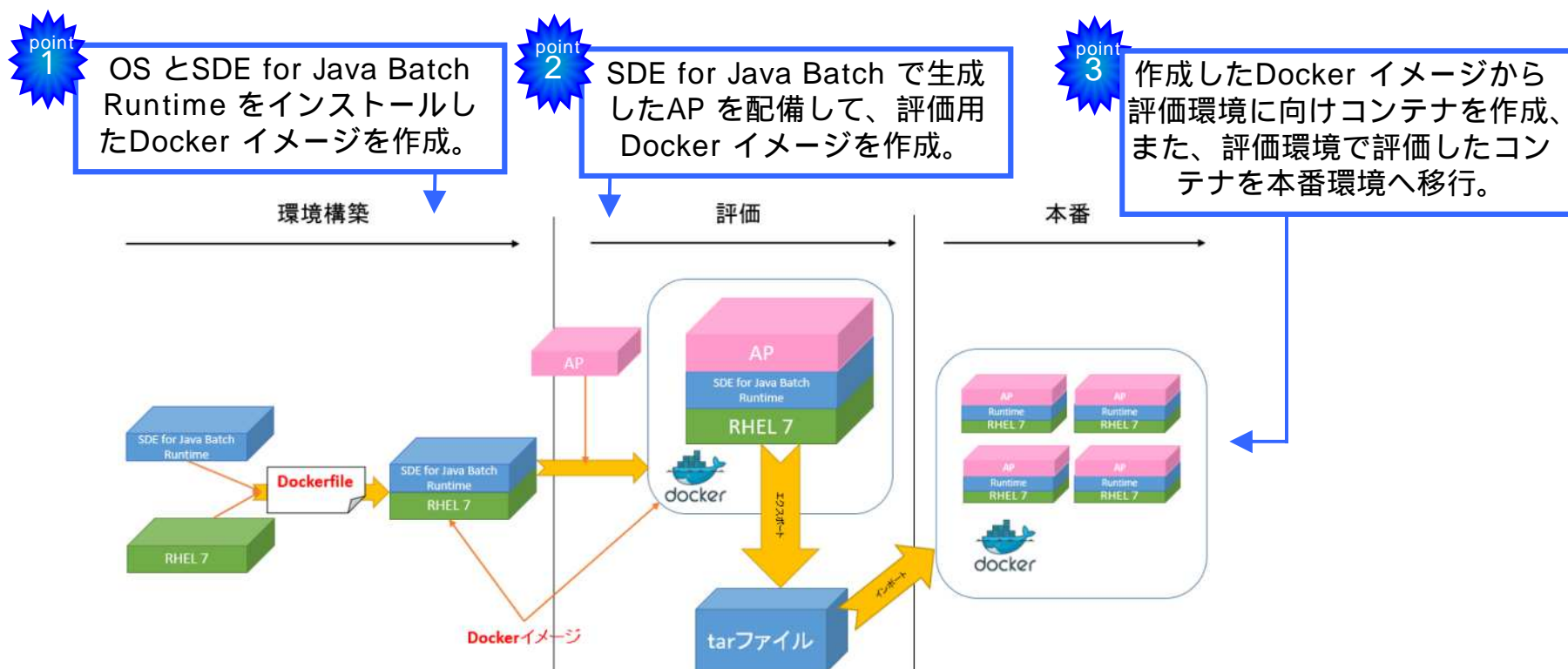


種別	機能一覧
ファイル書式編集関数群	テキスト形式/CSV形式ファイル、マルチレイアウトファイルを簡易に扱えます。 他、右詰/左詰、パディング、カンマ編集、改行コード指定ができます。 ファイルアクセスやDBアクセス(Java版)の専用ライブラリによる高速処理を実現します。
汎用関数及び汎用コマンド	[関数群] - エラーレベル判定部品、JOBNETパラメータ取得部品、クリーンUP部品、ログ出力部品 - チェックポイント処理部品群、ジョブ内共有領域部品
	[コマンド群] - ログ出力コマンド、DB一括更新コマンド、テーブルTRUNCATEコマンド、 プロシージャ起動コマンド、過去File削除コマンド、不要リソース削除コマンド

## 3.4.開発ツール : Docker実行環境

バッチアプリケーションをDocker上で実行できます。

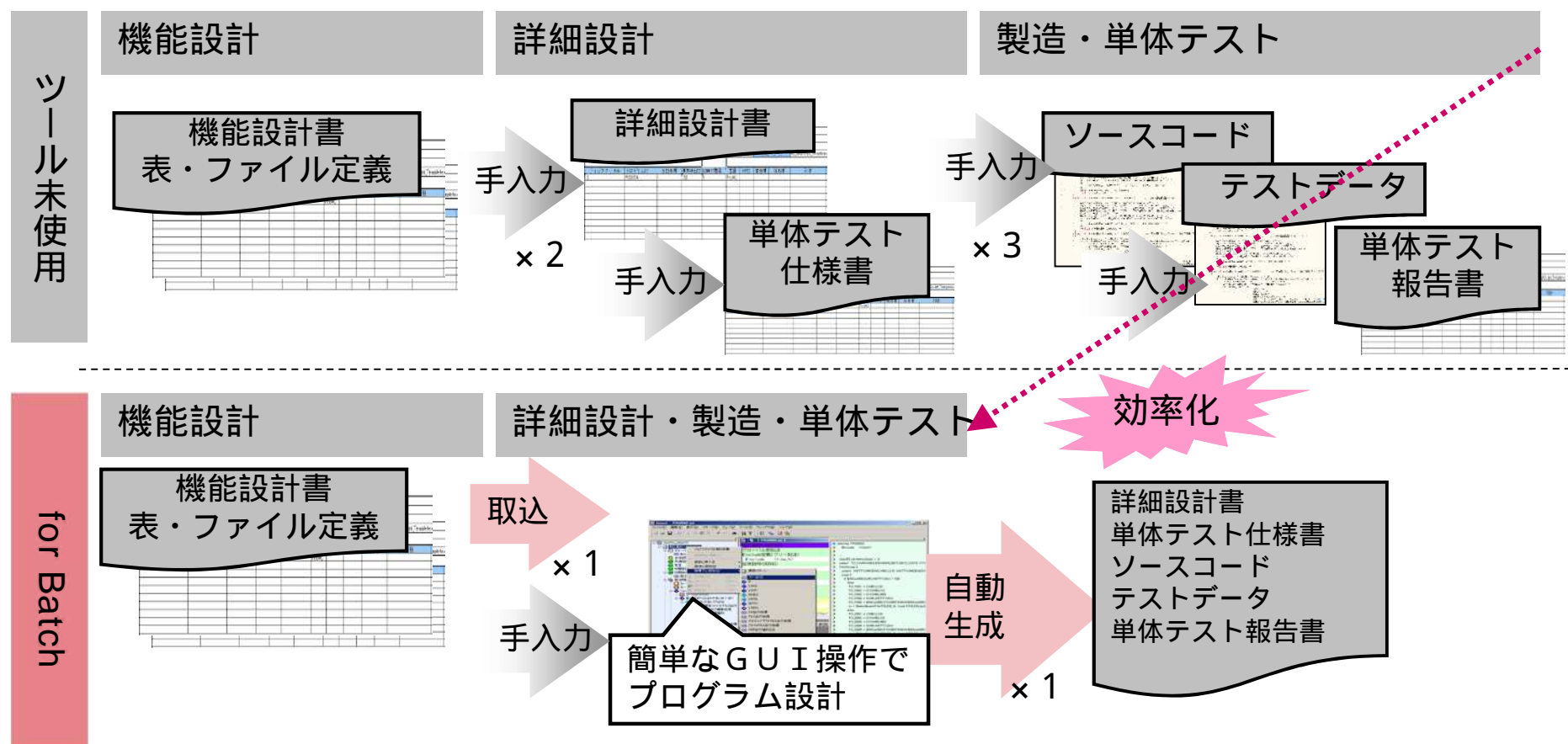
SDE for Java Batchで生成したバッチアプリケーションをDockerコンテナ上で実行することで、運用コストを削減。また、環境構築のための「Docker 導入ガイド」も提供し、導入を容易にします。



### 3.4.開発ツール :適用効果のまとめ ~作業効率化~

SystemDirector Enterprise for Batchは、余計な手間を省きます。

二度打ちや手入力を削減できます。





### 3.4.開発ツール :適用効果のまとめ ~ 品質の向上 ~

SystemDirector Enterprise for Batchは、品質を高める機能を提供します。

バッチ系APで典型的な処理構造を予め準備し、ソースを生成します。

バッチフレームワーク系及び業務系の共通部品を提供します。

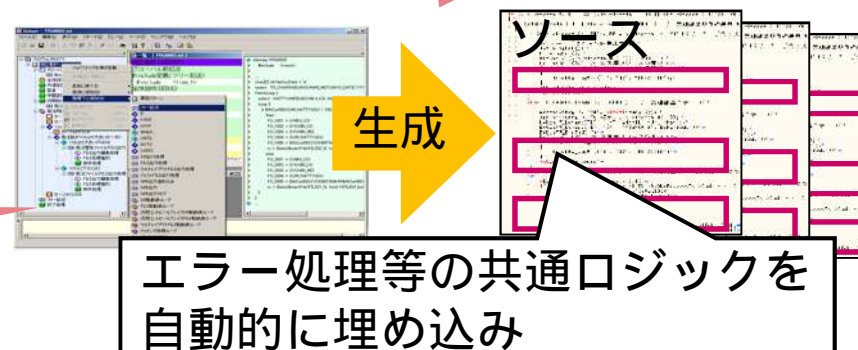
共通処理を自動的に埋め込みます。

業務ロジックに専念可能  
共通処理の事後の  
差し替えも容易



	Pro * C部品	PL / SQL部品	共通部品 (コマンド)
業務	日付操作		過去ファイル削除
	文字列操作		
基盤	ログ出力・取得		
	チェックポイントリラン制御		
	Oracle接続		
	エラー制御		

100本以上の  
部品を提供！

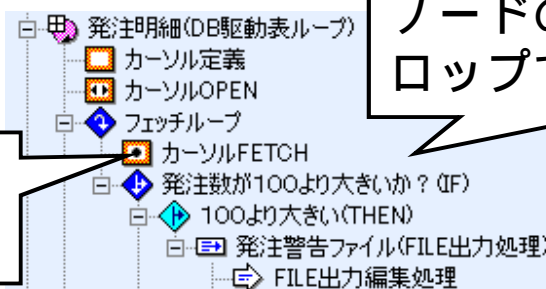


### 3.4.開発ツール :適用効果のまとめ ~ 保守性の向上 ~

SystemDirector Enterprise for Batchを使えば、保守工程における仕様変更に対応、保守性を向上します。

プログラムを簡単に変更できます。

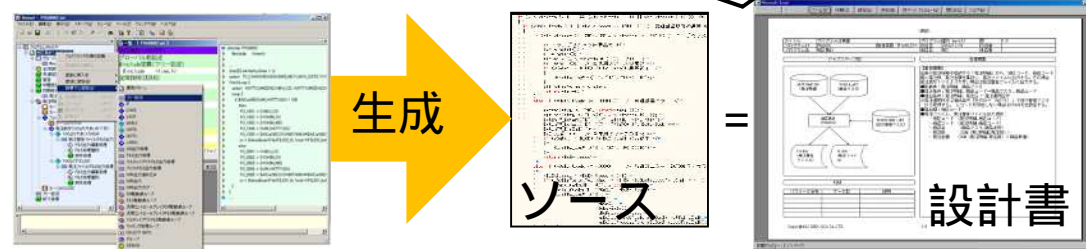
プログラムの処理フローを階層化されたノードで表現



ノードのドラッグ&ドロップで簡単に変更

ソースと設計書の整合性を保ちます。

ソースも設計書も同じファイルから自動生成



リポジトリ内を検索できます。

- CRUD検索
- バージョン差分検索
- フリー検索

影響範囲を容易に特定

## 3.5.動作環境(SystemDirector Enterprise for C Batch)

Pro\*C、PL/SQL、ECPG、PL/pgSQL版の場合

SWスタック

環境

【凡例】

for Batch製品

OS/ミドル

個別PJ資産

APサーバ

DBサーバ

業務AP

SystemDirector Enterprise  
for Batch Runtime

オプション  
(業務要件に応じて選択)

SystemDirector Enterprise  
for Batch Framework

JOBスケジューラ

Windows/Linux

DBインスタンス

Oracle/PostgreSQL

Windows/Linux

[開発環境] . . . . .

- ・クライアントOS  
Windows7(32bit/64bit)/  
Windows10(64bit) Version 1703
- ・コンパイラ  
ANSI C/C++ \*1、  
Microsoft Visual C/C++ 2015/2017 \*1
- ・for Batch開発環境サーバを利用する場合  
Microsoft EXCEL 2013/2016 \*1、  
.NET Framework4.7.0 \*1  
バージョン管理ソフトと連携する場合は、  
更にCVSまたはSVN \*1が必要です。

[実行環境] . . . . .

- ・サーバOS  
Windows Server 2012 R2(64bit)、  
Windows Server 2016(64bit)、  
Linux Linux6系(64bit)
- ・DB  
Oracle11g/12c/12cR2系 \*1、  
PostgreSQL 9.2系 \*1

\*1 調達品ソフト



## 3.5.動作環境(SystemDirector Enterprise for Java Batch)

### Java版(PureJava / jBatch連携)の場合

SWスタック

環境

【凡例】

for Batch製品

OS/ミドル

個別PJ資産

APサーバ

DBサーバ

業務AP

DBインスタンス

ジョブ  
スケ  
ジュ  
ー  
ラ

SystemDirector Enterprise  
for Batch Runtime

Java標準クラスライブラリ

Java VM/APサーバ

Windows/Linux

Oracle/PostgreSQL  
/MySQL/SQL Server  
/Aurora PostgreSQL  
/Aurora MySQL

Windows/Linux

[開発環境]

- ・クライアントOS  
Windows10(64bit) Version 1803
- ・ for Batch開発環境サーバを利用する場合  
Microsoft EXCEL 2016(64bit)\*1、  
Eclipse4.10\*1、VelocityV1.4、dom4jV1.6.1、  
jaxenV1.1.1、  
バージョン管理はEclipseの機能を利用

[実行環境]

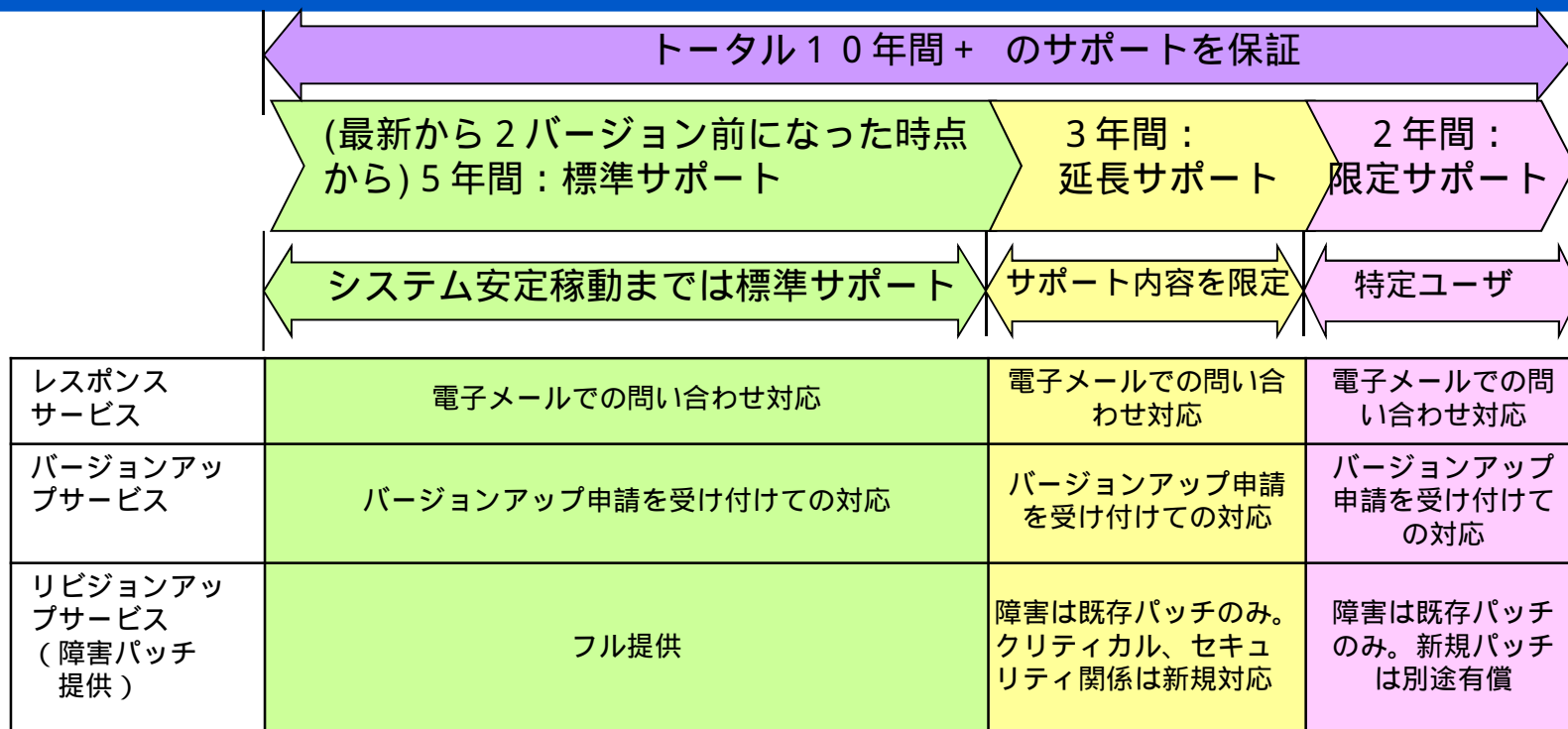
- ・サーバOS  
Oracle JDK 8/11\*1、  
RedHat OpenJDK 8/11\*1、log4jV2\*1
- ・ APサーバ  
WebOTX Application Server V10.2\*1、  
GlassFish V4.1.2\*1、JBoss WildFly V8.1\*1
- ・ DB  
Oracle11g/12c/12cR2/18c\*1、  
PostgreSQL 9.6\*1、MySQL 5.6\*1、  
SQL Server2016\*1、Aurora PostgreSQL 9.6  
Aurora MySQL 5.6

\*1 調達品ソフト

## 第4章 サポートサービス

## 4.1. サポートメニュー

サービスメニューとして、以下のサービスを提供します。  
お客様のご要望に応じて内容を組み立てさせていただくことも可能です。



■ 上記サービスに含まれないものでも、ご要望に応じて調整可能です。

- 例) SystemDirector Enterpriseを新規に導入する場合の環境構築 / プロジェクト立ち上げ教育といった導入支援
- 例) お客様専用窓口の設置によるホットラインサポート (メールに加えて、電話・現地での問い合わせ対応なども実施)

## 4.2.教育メニュー

きめ細やかなサポートで、バッチ開発の早期立上げを支援します。

### SystemDirector Enterprise for Batch教育

for Batchを使用した開発の進め方を、座学と演習を交えて開発者に教育します。教育内容は、実機を使っての for Batchによるバッチアプリケーションの製造から実行まで行う1日コースです。

【前提】 RDBMSの知識、及びプログラミング言語の知識を要します。

【注意】 会場や演習用PCはご準備願います。

なお、SystemDirector Enterpriseのベースとなっている一般知識が前提となります。前提知識を習得するための教育コースもございますので、合わせてご検討ください。

### NECマネジメントパートナー ラーニング事業サイト

<https://www.neclearning.jp/>

	Java言語	
前提コース	Javaによるデータベースアクセス	Javaプログラミング基礎 (入門、基礎1、基礎2)
	C言語	
前提コース	SQL (入門、基礎)	Cプログラミング (基礎1、基礎2)

## 4.3. 有償サービス（バッチ標準化支援）

### バッチシステム構築の標準化を支援します

SystemDirector Enterprise for Batchを利用したバッチ系業務アプリケーション開発の標準化作業を、経験豊かな要員が支援するサービスです。for Batchを初めて導入するプロジェクトは是非ご活用ください。

サービス内容は、プロジェクトの希望に応じて、個々に設定できます。以下にその例を示します。

なお、NEC-GによるSI体制の下での支援が前提となります。

メニュー	成果物
標準化支援	<ul style="list-style-type: none"><li>・バッチ方式設計書</li><li>・バッチ系業務AP開発フローガイド</li><li>・バッチ系業務A P ログ出力標準</li><li>・バッチ実装ガイド</li><li>・バッチ業務A P 物件チェックリスト</li><li>・共通単体テスト仕様書</li></ul>
ドキュメントテンプレート カスタマイズ	<ul style="list-style-type: none"><li>・機能設計書テンプレート</li><li>・詳細設計書テンプレート</li><li>・単体テスト仕様書テンプレート</li><li>・ファイルレイアウト定義設計書テンプレート</li><li>・テーブルレイアウト定義設計書テンプレート</li><li>・JOB定義生成ツールテンプレート</li></ul>
標準化実装支援	<ul style="list-style-type: none"><li>・共通部品の開発</li><li>・業務パターン及びプロパティファイル設計</li></ul>

## 4.4.お問い合わせ先

### ご購入前のお問い合わせ

NEC SystemDirector Enterprise お問い合わせ窓口

 Web

<http://jpn.nec.com/SystemDirectorEnterprise/contact.html>

記載されている会社名、製品名は、各社の登録商標または商標です。

# 付録 関連製品

# 大量データ高速ソート・マージユーティリティ SORTKIT

## < 製品概要 >

大量データを高速処理するためのソート/マージ専用ユーティリティ。多様なデータフォーマットやレコードフォーマットに対応し、大量データを処理するバッチ処理や業務アプリケーションのソート処理を高速化。

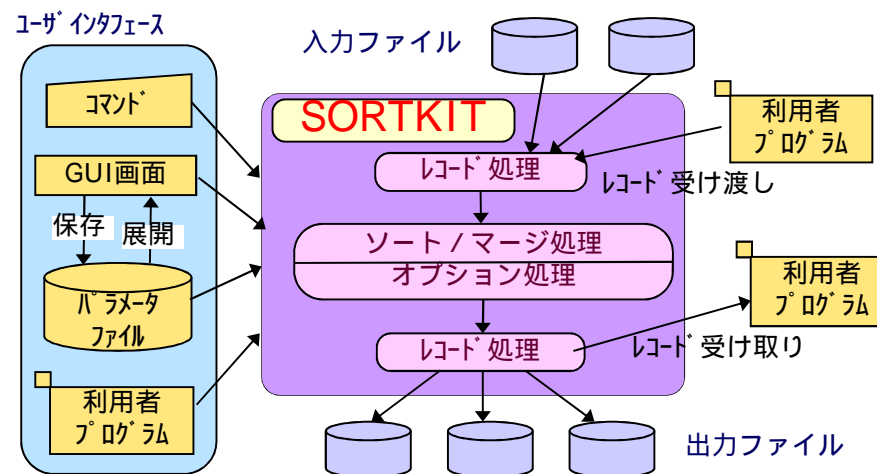
## < 導入効果と機能 >

大量データの処理に適した高速ソートアルゴリズムの採用によるソート処理時間を短縮。

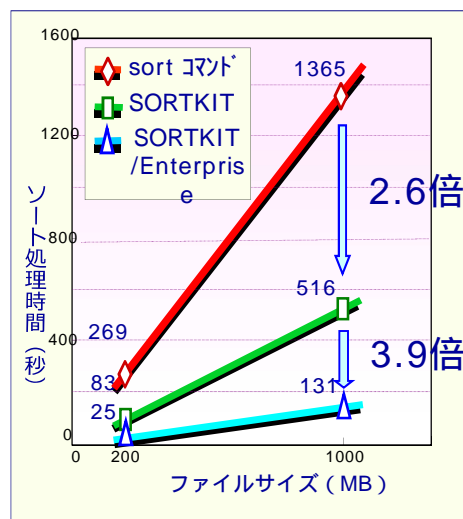
既存のバッチプログラムや業務アプリケーションと連携し、大量データ処理の高速化を実現

- ・充実したインターフェースで各種の実行形態に対応。
  - バッチコマンドによるバッチ処理
  - GUIによる対話的なオプション設定や実行
  - NEC COBOLの整列併合機能との連携
  - C言語で作成されたアプリケーションからAPI利用
  - MS Visual BasicなどのアプリケーションからActiveXコントロールを利用
- ・ビジネス用途に必要な各ファイルタイプをサポート。
  - テキストファイル（可変長レコード・可変長フィールド形式）
  - NEC COBOL形式、MicroFocusCOBOL形式
  - NEC版C-ISAM形式、IFASPRO 形式、RAF 形式
- ・ビジネス業務に適した各データタイプをサポート。
  - ASCII / 1バイトJIS
  - EBCDIC / EBCDICカタカナ
  - 漢字コード（シフトJIS / JIPS / EUC / UCS-2）

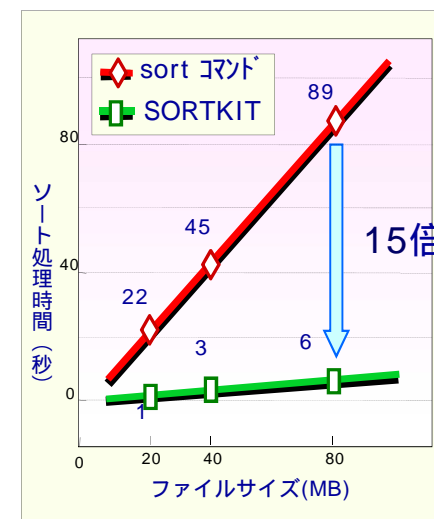
## イメージ図



Windows sort コマンドとの比較



UNIX sort コマンドとの比較





# 業務処理を自動化し、実行状況の一元監視を実現するジョブ管理製品 WebSAM JobCenter

## <製品概要>

複雑な業務を自動化し、お客様の業務を飛躍的に効率化・最適化するジョブ管理製品。さらに、業務の実行状況を一元監視し、異常検知から、自動障害回避・対処まで行え、迅速な障害対応が可能。

## <導入効果と機能>

### 自動化による業務運用コスト削減

- 業務をフロー化することで、正確で効率の良い業務運用を実現。
- 日次/月次や営業日を意識して実行するなど、業務に合わせたスケジュールが容易に作成可能。

### 業務の実行状況をわかりやすく「一元監視」

- ジョブ実行状況の一元監視により、実行エラーの早期発見、早期対応が可能。

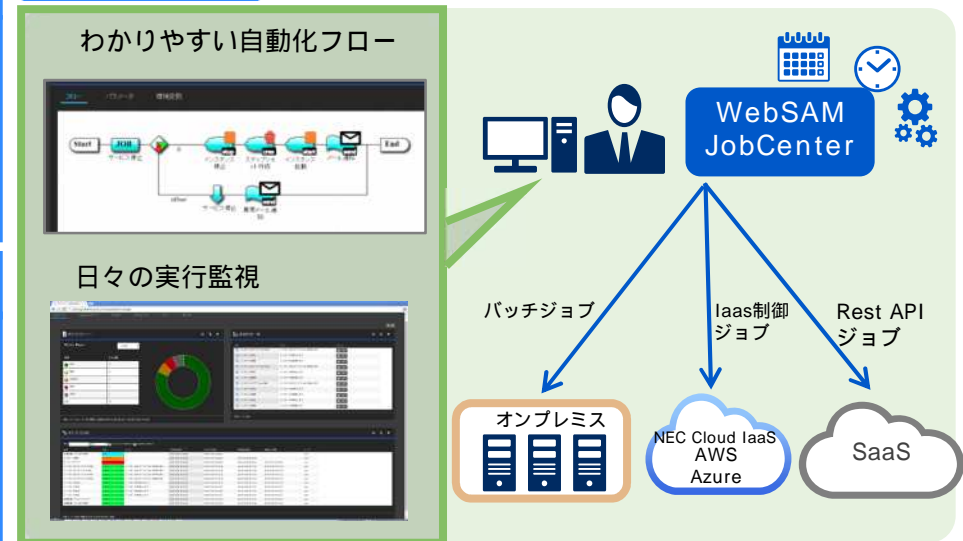
### ジョブ実行状況の分析をサポート

- ジョブの実行実績を一括収集しExcelやCSV形式で出力可能。ピーク時間や遅延傾向、実行スケジュールの多重度などの分析に活用可能。

### ハイブリッドクラウド連携

- IaaSの制御部品とRest APIが実行可能な部品によりクラウドサービスと連携する処理を自動化。

## イメージ図



## NEC優位性

- 機能、実績が豊富、かつ、他社より安価。
- IaaS(AWS、NEC Cloud IaaS、Azure)とSaaSを含めたハイブリッドクラウド環境の業務運用の自動化
- 直観的に操作でき、ジョブ実行状況の確認やオペレーションがシンプル。 AWS : Amazon Web Services

動作環境（詳細は以下のURLをご参照ください）：

Windows Server 2016, Windows Server 2012 R2 他

Red Hat Enterprise Linux, HP-UX, AIX に対応

製品URL（社外）： <http://jpn.nec.com/websam/jobcenter/>

価格（最小構成）300,000円～

- SystemDirector Enterprise, InfoFrame は日本電気株式会社の登録商標です。
- SVF, SVFX-Designerは、ウイングアーク 1 s t 株式会社の登録商標です。
- Windows, Office, Excel, Visual Studio, .NET Framework はMicrosoft Corporationの米国およびその他の国における登録商標です。 また、Windows の正式名称は Microsoft Windows Operating System です。
- Java, GlassFishはOracle Corporation の米国およびその他の国における登録商標です。
- Amazon Web Services, 'Powered by Amazon Web Services'ロゴ、その他のAWS商標はAmazon.com, Inc.の米国およびその他の国における登録商標です。
- Log4jはThe Apache Software Foundationの登録商標です。
- Eclipseは Eclipse Foundation, Inc. の米国およびその他の国における登録商標です。
- AndroidはGoogle Inc. の米国およびその他の国における登録商標です。
- iOS はCisco Systems G.K.の米国およびその他の国における登録商標です。
  
- その他、記載されている会社名および製品名は、各社の商標または登録商標です。

 **Orchestrating** a brighter world

**NEC**