

業務システム構築基盤

SystemDirector Enterprise for Java (バッチ) ご紹介

日本電気株式会社

本資料ではSystemDirector Enterprise for Java (オンライン)をfor Java (オンライン)、SystemDirector Enterprise for Java (バッチ)をfor Java (バッチ)、SystemDirector Enterprise for Java (モダナイゼーション)をfor Java (モダナイゼーション) と略します

第 1 章 SystemDirector Enterprise とは

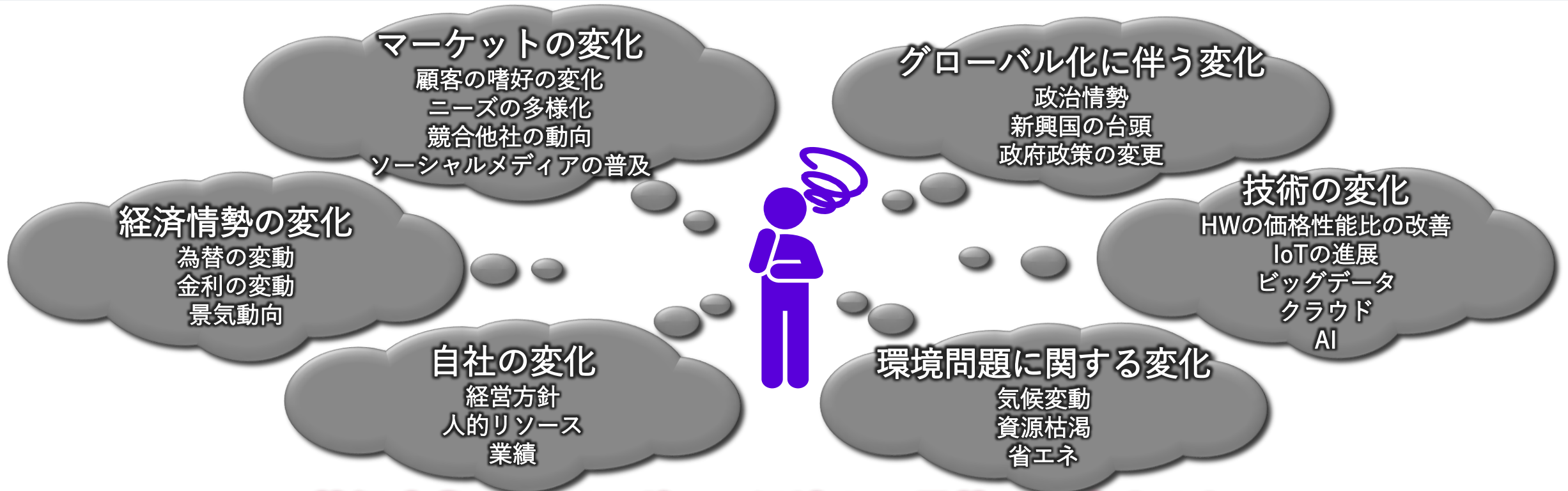
第 2 章 SystemDirector Enterprise for Java
(バッチ) の特長

第 3 章 ライセンス体系

第 4 章 関連情報

第1章 SystemDirector Enterprise とは

市場を取り巻く環境の変化や、デジタル技術・ICT技術の進化への対応力が、ビジネスの成否に大きな影響を及ぼす



状況変化のスピードに、迅速かつ柔軟に対応するために、
開発のプロセス・基盤の見直しが必要

探索的
アプローチ

継続的な仮説検証
サイクルの実施

リーンスタートアップ、アジャイル、
UX/デザイン思考、共創

新技術

先進の技術・
最新のサービス活用

IoT、AI、アナリティクス、
クラウド、コンテナ、
サイバーセキュリティ、OSS

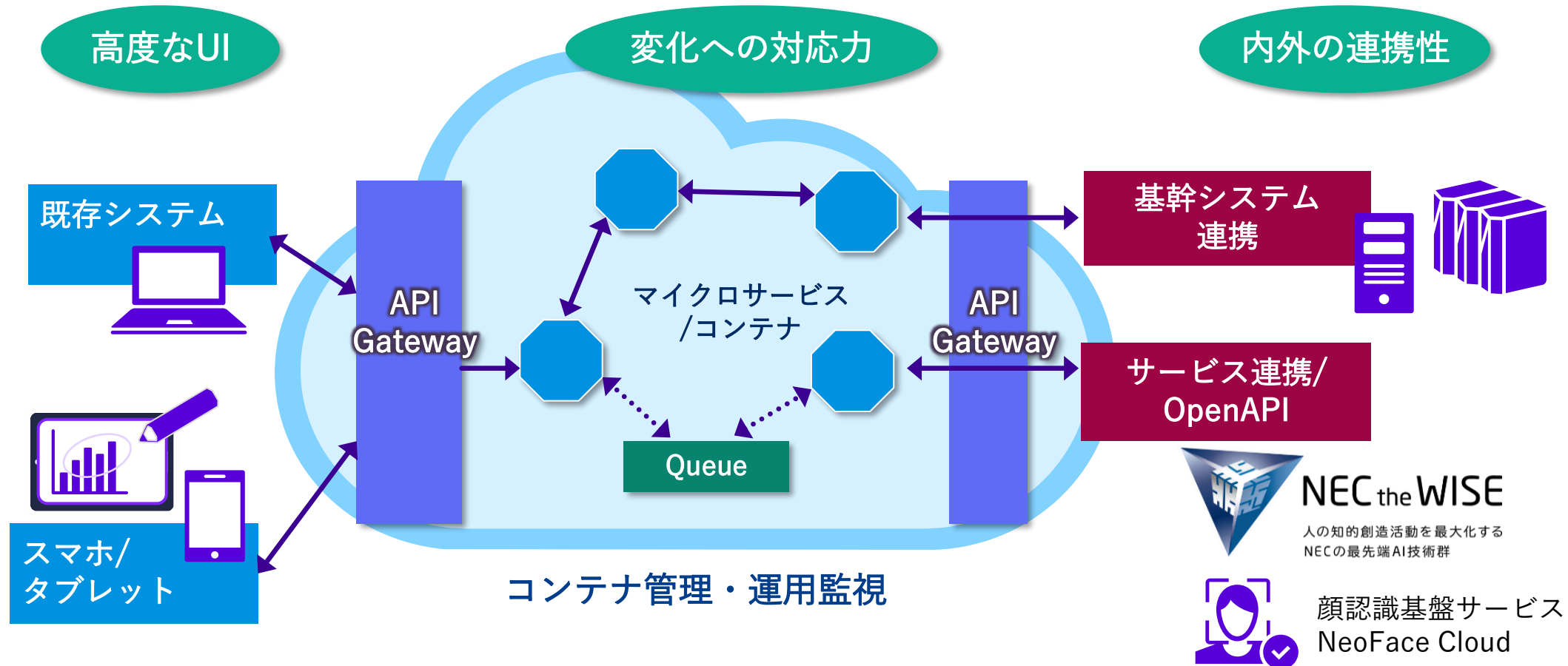
価値創造

企業内部と外部の
アイデアを有機的統合

オープンサービスイノベーション、
オープンAPI、
クラウドネイティブ

1.3 新しいAPアーキテクチャー

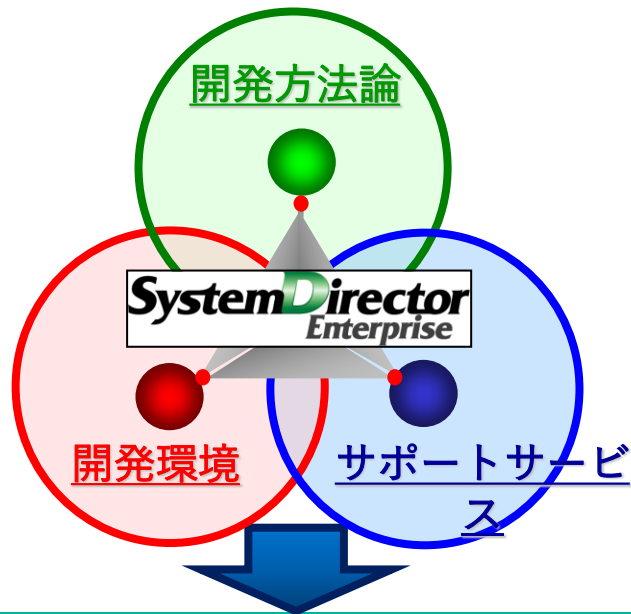
各機能が疎結合で構成され、柔軟性のあるアーキテクチャが主流となる



1.4 SystemDirector Enterpriseのコンセプト

NECグループ標準の業務システム向け統合開発環境

開発方法論、開発環境、サポートサービスにより、効率的なシステム構築をサポート



◇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

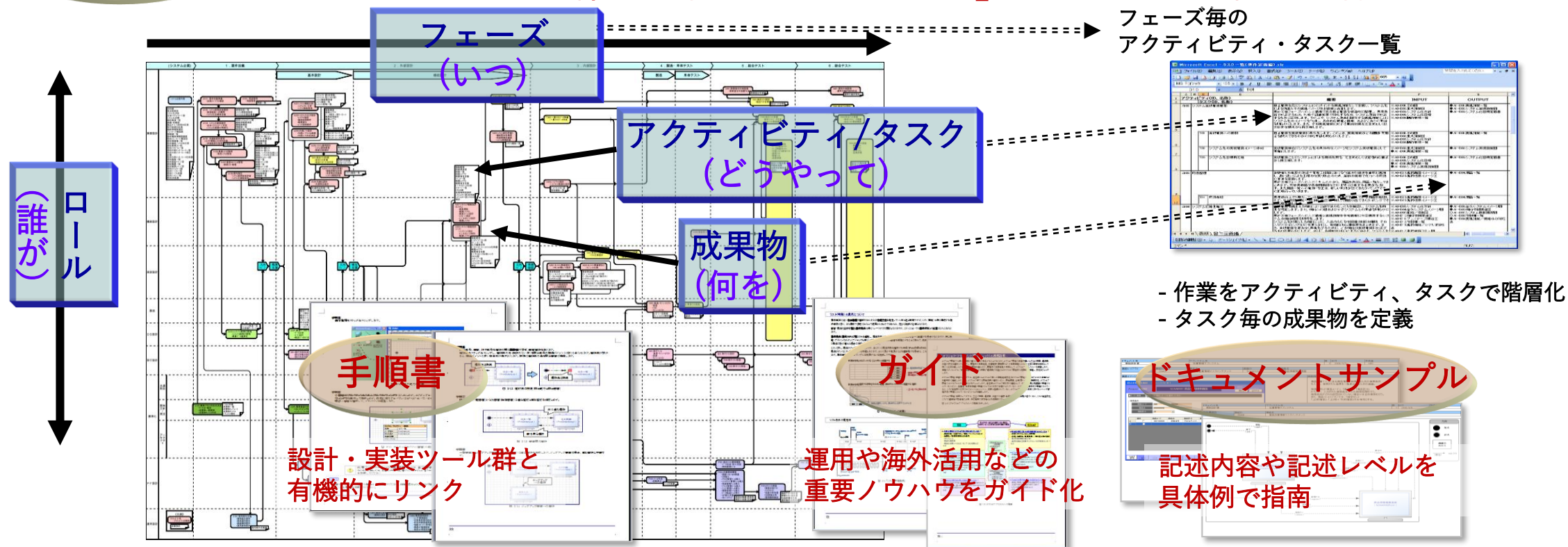
サポート提供

- ・ 専任の技術サポート部隊を用意
- ・ 長期サポート保証

1.5 SystemDirector Enterprise 開発方法論

SoR／SoE開発を標準化する開発プロセスを中心に
各種手順書、ガイド、ドキュメントサンプルなど豊富なコンテンツを提供

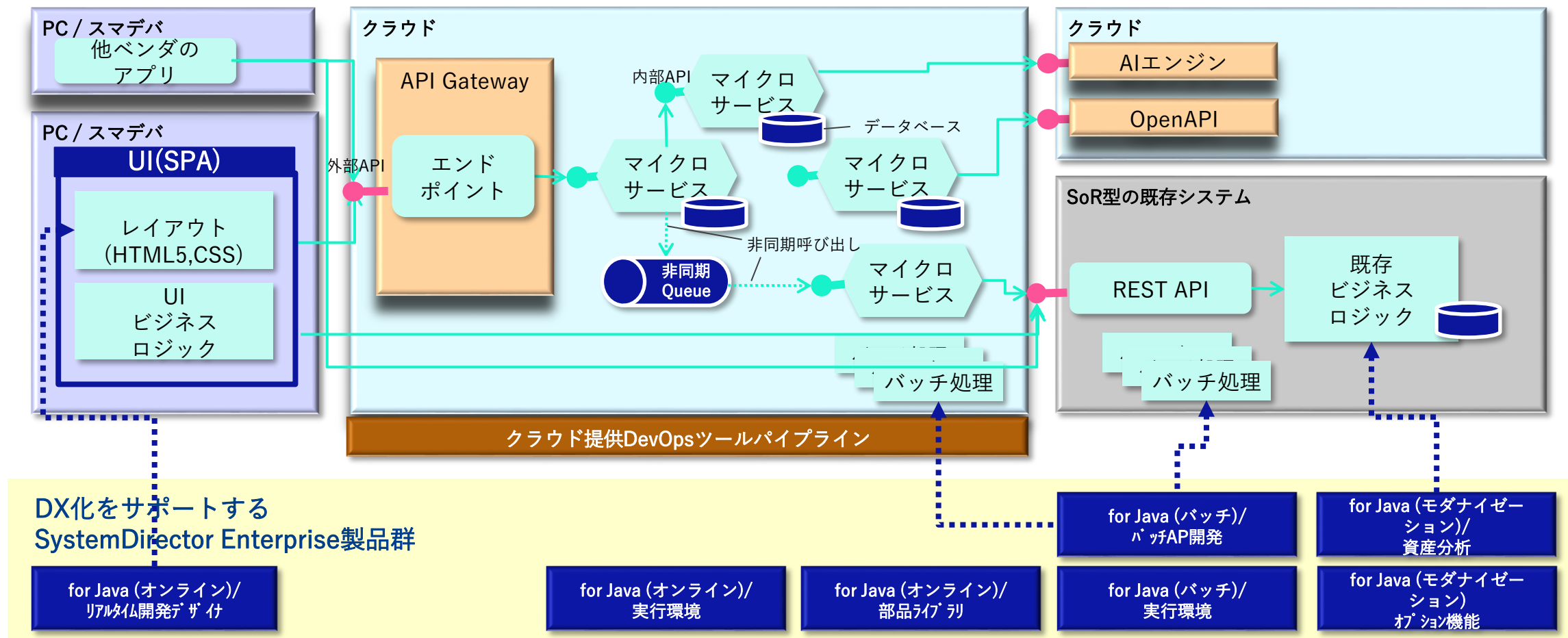
開発プロセス SI現場の成功・失敗ノウハウを実践的な開発プロセスに結集
SIに必要な「いつ・誰が・何を・どうやって」を漏れなく重複なく体系化



1.6 SystemDirector Enterprise 開発環境

BluStellar

モダナイゼーションのBeforeからAfterまで一貫したサポートを提供。さらに、クラウド提供CI/CDパイプラインやOSSとの連携を強化し、SoE開発をフルサポート



第2章 SystemDirector Enterprise for Java (バッチ)の特長



2.1 製品の狙い、メリット

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
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

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

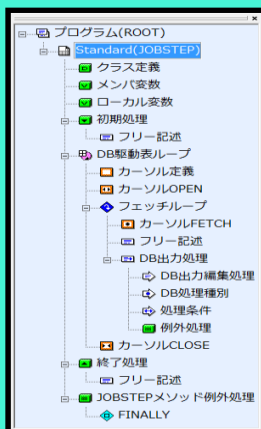
メインループ

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

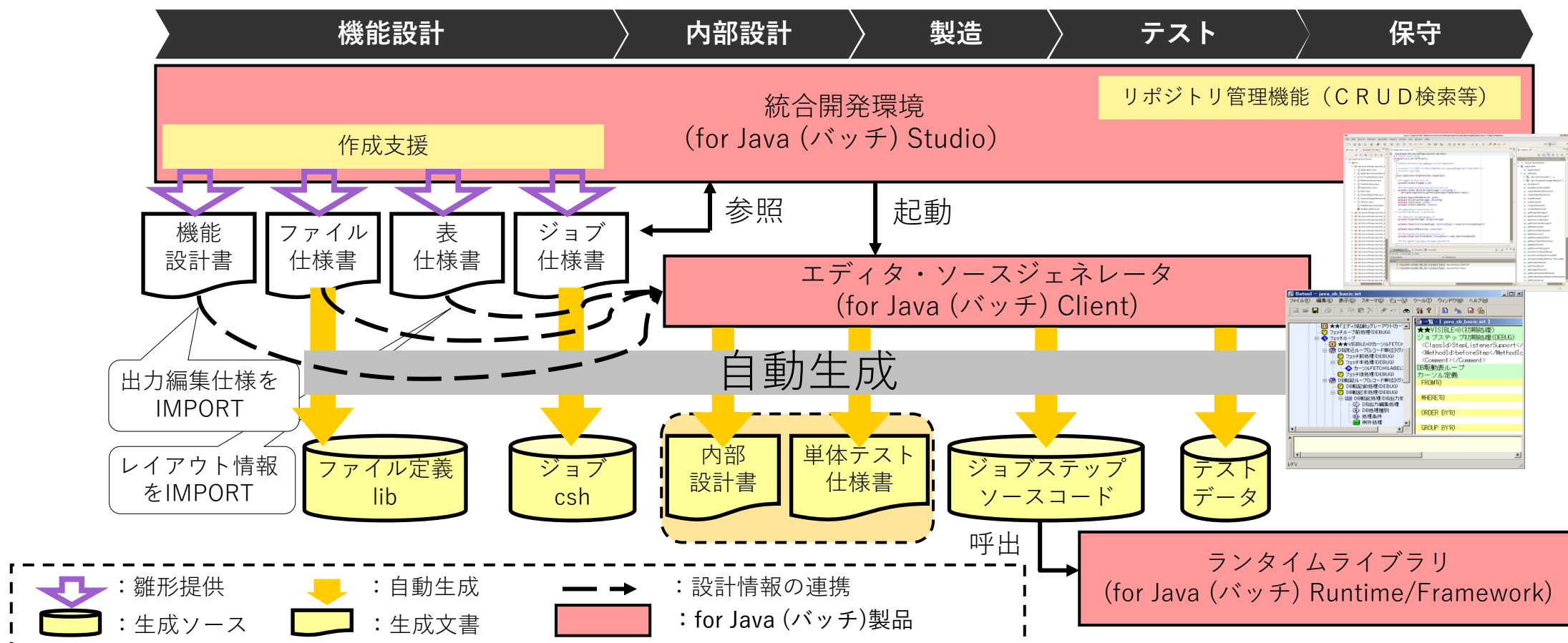
終了処理

例外処理

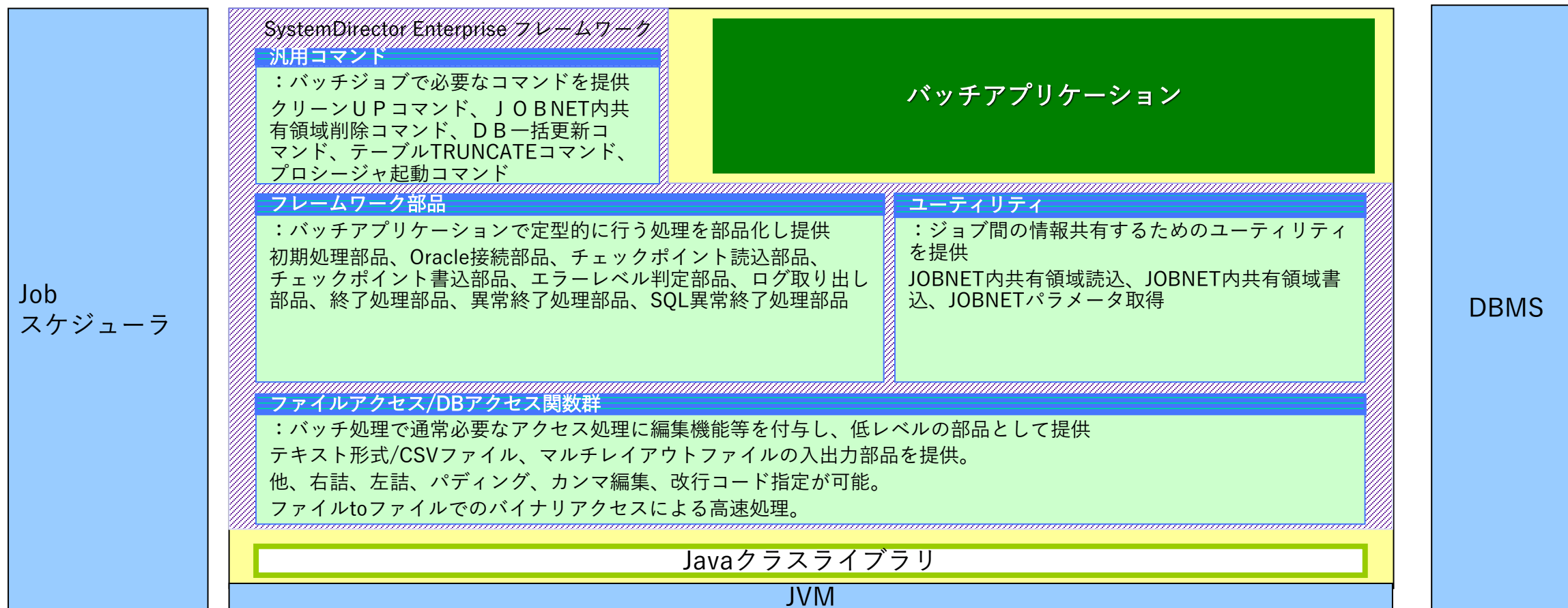


2.2 全体機能構成

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支援



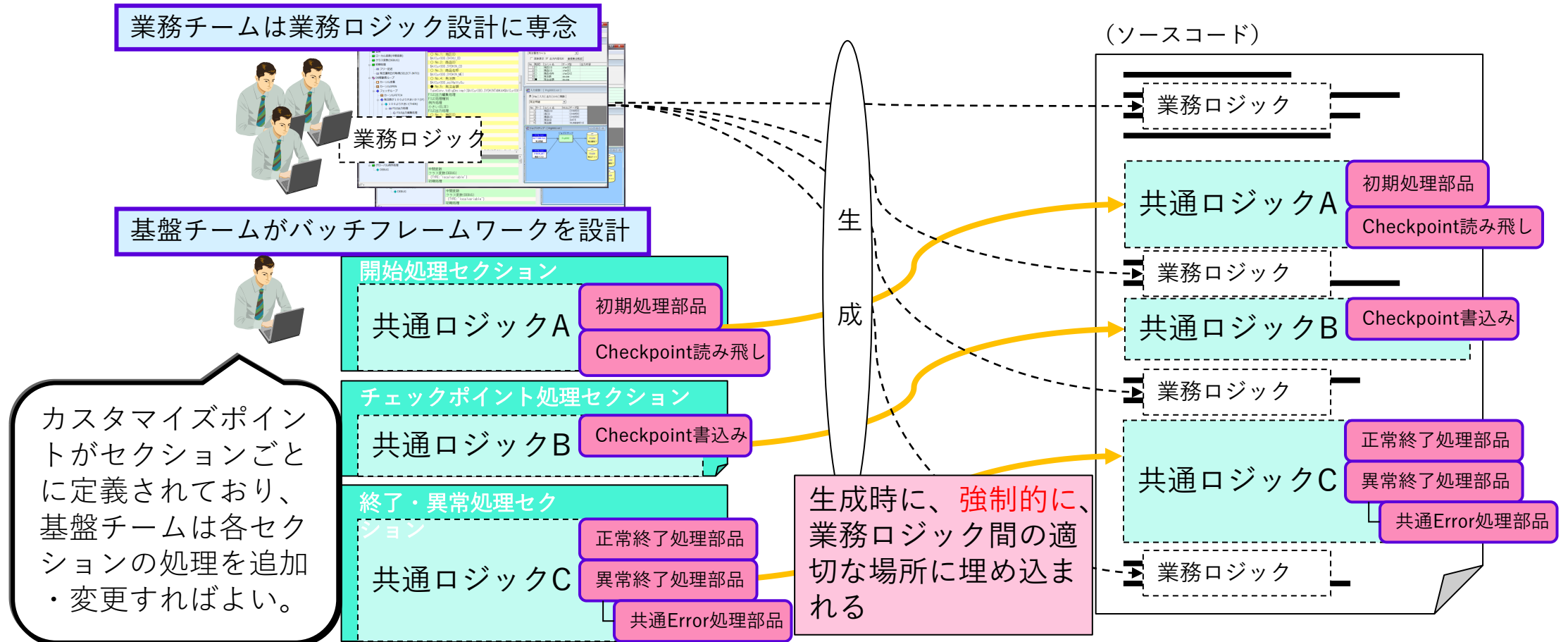
バッチジョブ/バッチアプリケーション構築に通常必要な
コマンド/APIを提供することで、業務SEは業務ロジックに注力可能



2.4 カスタマイズ可能なバッチフレームワーク

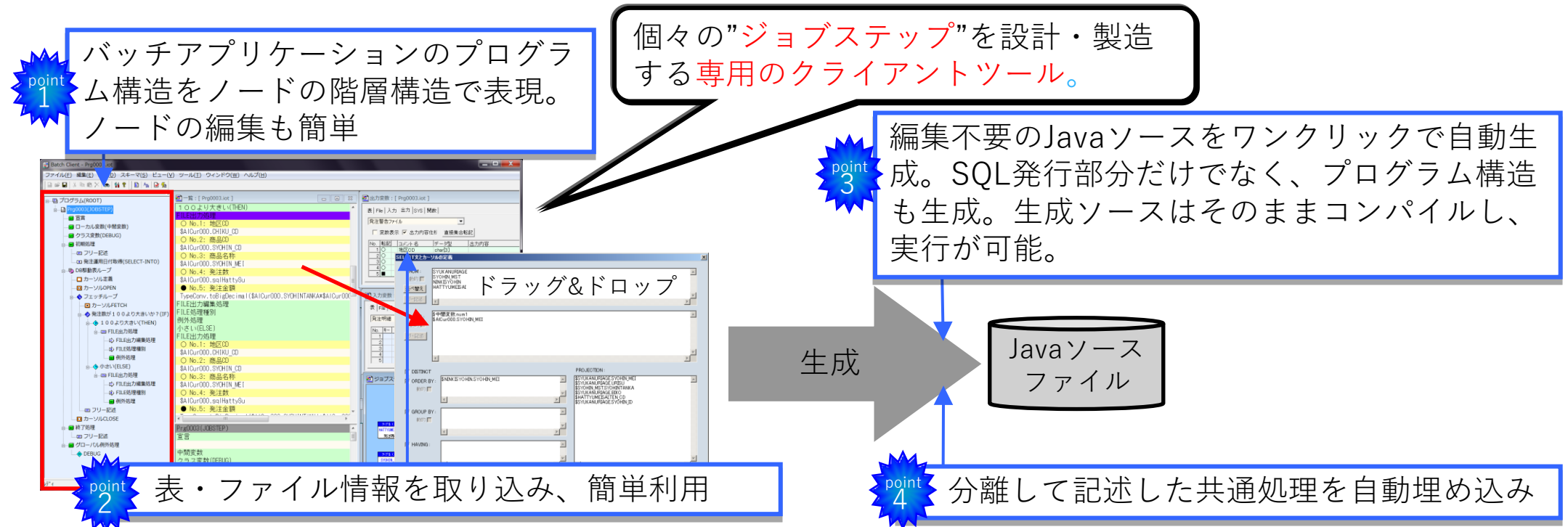
バッチフレームワークのカスタマイズが可能

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



2.5 操作性に優れたGUIと強力な生成機能

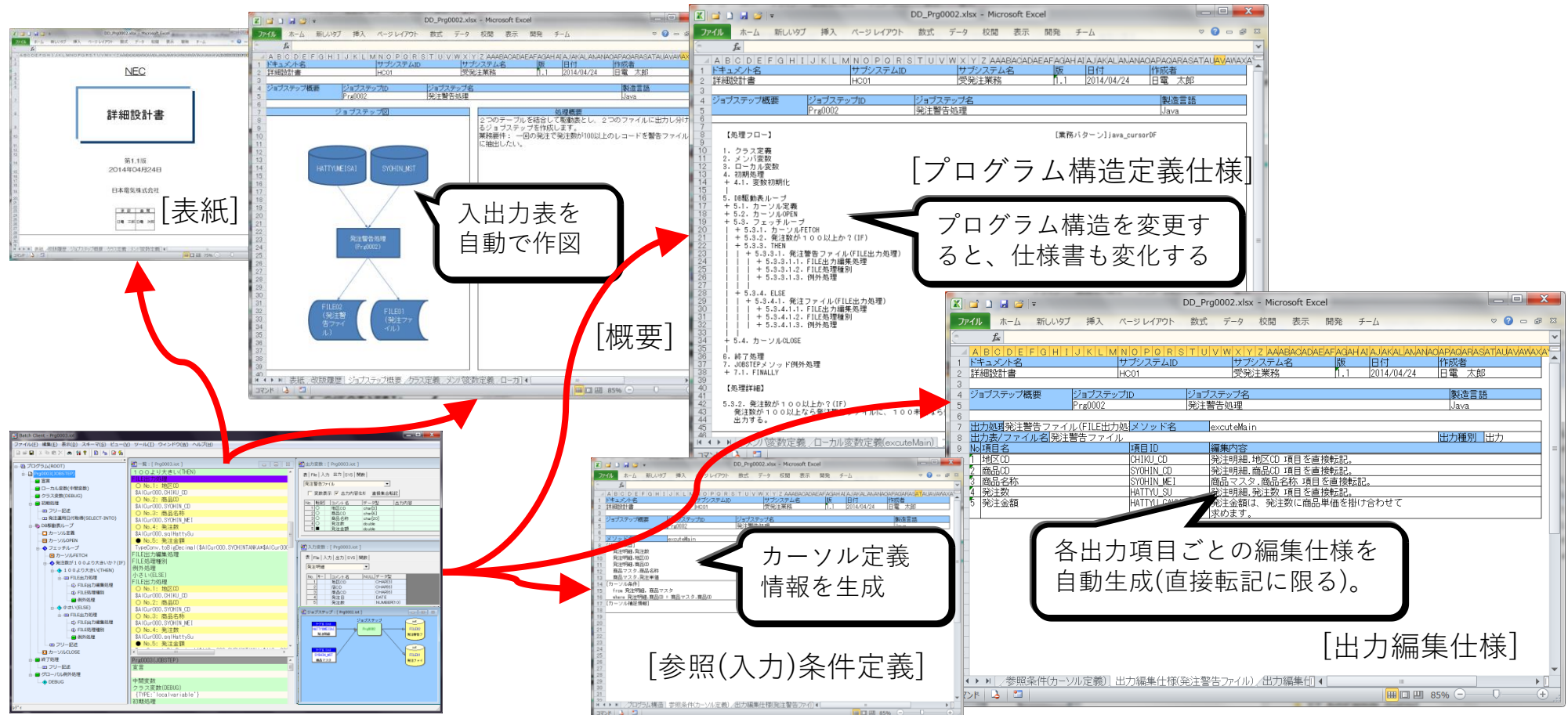
バッチアプリケーションの業務処理をGUIで実装し、ソースや設計書を自動生成
言語固有のルールをソースジェネレータが隠蔽し、開発者は業務処理(ビジネスロジック)の設計/実装に専念可能



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

2.6 ドキュメント自動生成

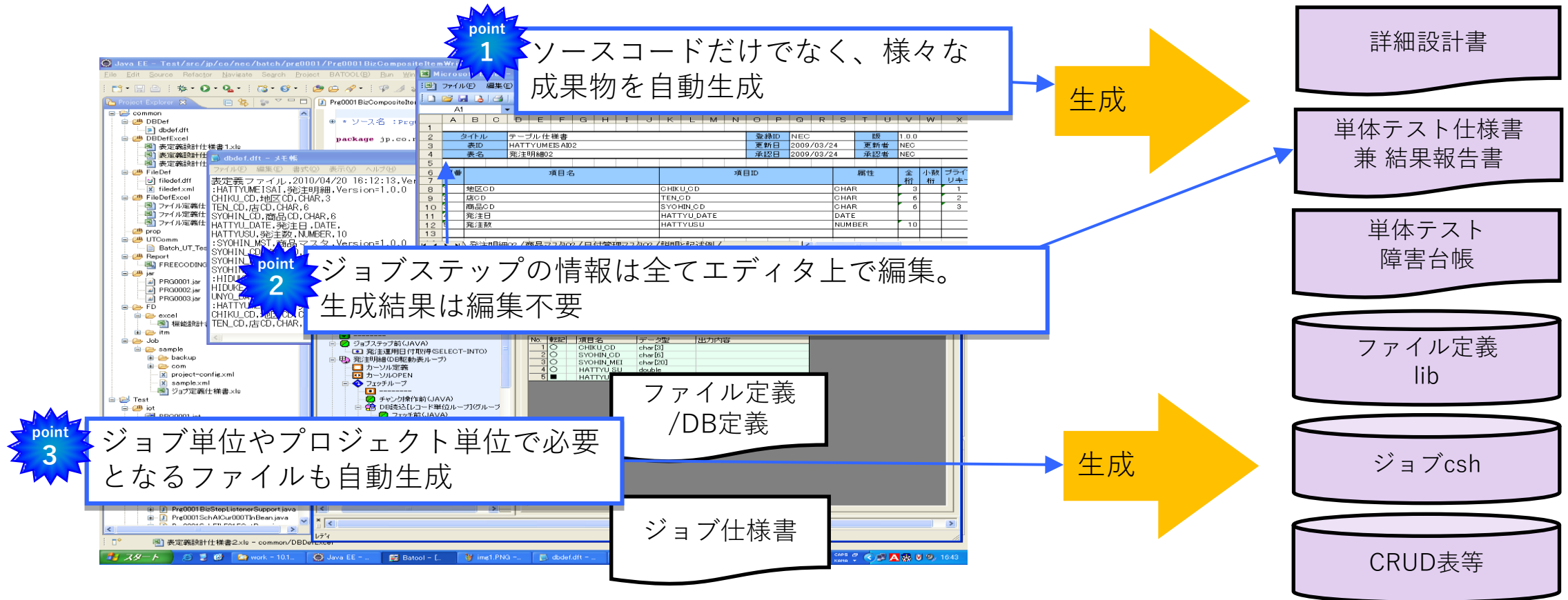
プログラム構造や出力編集仕様のドキュメントを、ソース生成時と同じ情報を元に生成することで、ソースと仕様書の乖離を防止



2.7 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

- 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
- 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



2.8 成果物の管理(2/2)

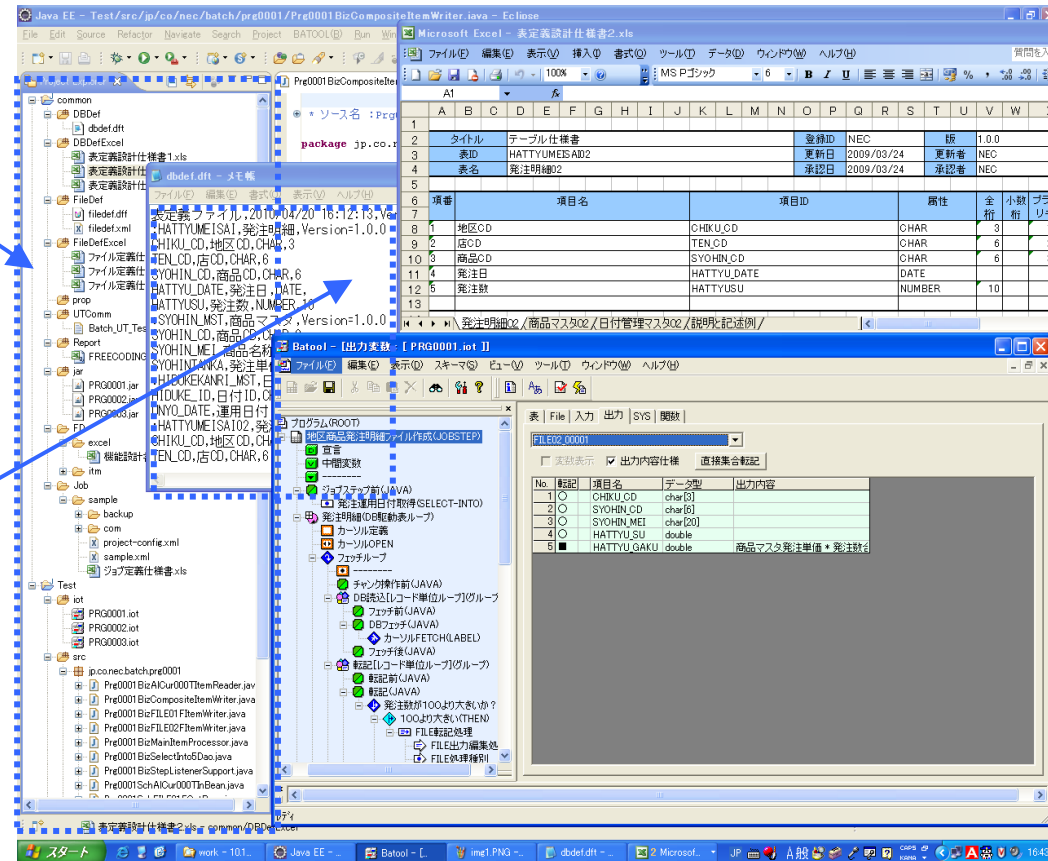
- 機能設計書、詳細設計書、ジョブステップソース、単体テスト仕様書、表定義/ファイル定義仕様書を一括で管理
- リポジトリ機能により複数の成果物から情報を抽出し、CRUD表生成、影響範囲検索、ソース規模/生成率測定。また、各種生成物を一括生成。

point
1

統合環境で成果物を管理。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。

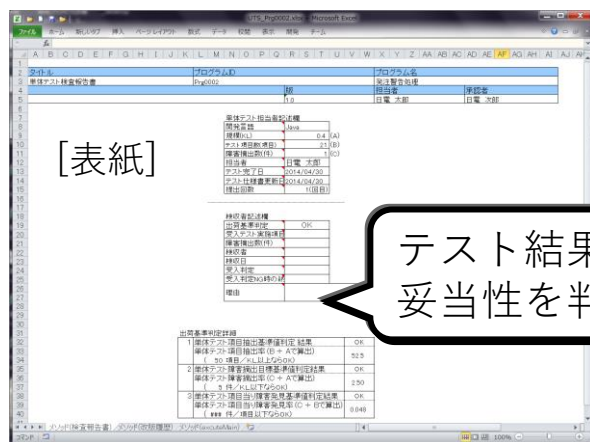


point
3

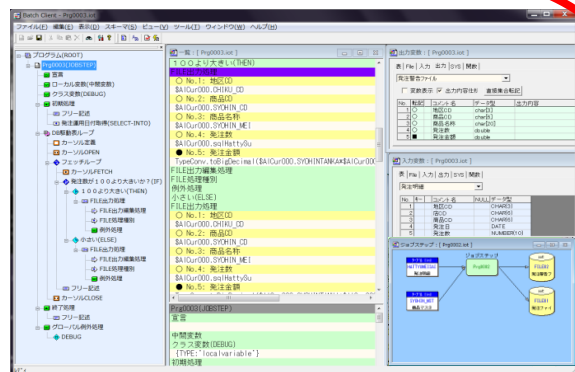
CRUD表生成など、複数のジョブステップの情報を集計可能。

2.9 テスト項目の自動生成

プログラム構造から一部テスト仕様を自動生成
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



テスト結果の
妥当性を判定



項目	テストレベル	テスト項目	想定される結果
1	1	変数初期化の確認	
2	2	文字列型変数の初期化 数値型変数の初期化	文字列型変数が""で初期化されていること。 数値型変数が0で初期化されていること。
3	3	DB駆動表ループの確認	
4	4	カーソル条件の確認	<p>【カーソル定義】</p> <p>[取得項目]</p> <ul style="list-style-type: none">発注明細.発注数発注明細.地区CD発注明細.商品CD商品マスタ.商品名称商品マスタ.発注単価 <p>[FROM]</p> <ul style="list-style-type: none">発注明細[HATTYU_MSA]商品マスタ[SYOHIN_MST] <p>[取得条件]</p> <p>発注明細.商品CD = 商品マスタ.商品CD</p> <p>[取得順序]</p> <p>指定無し</p> <p>[グルーピング]</p> <p>指定無し</p> <p>[グルーピング条件]</p> <p>指定無し</p> <p>条件式の処理がされる 左記の条件でTHENの処理がされる 左記の条件でTHENの処理がされる 左記の条件でELSEの処理がされる</p>
5	5	発注数が100以上かの確認	
6	6	発注数==100	
7	7	発注数>100	
8	8	発注数<100	
9	9	THENの確認	
10	10	発注警告ファイルの確認	発注警告ファイル処理がされる
11	11	ELSEの確認	発注ファイル処理がされる
12	12	出力編集項目の確認	
13	13	CHIKU_CD(地区CD)	発注明細.地区CDを直接転記する
14	14	SYOHIN_CD(商品CD)	発注明細.商品CDを直接転記する
15	15	SYOHIN_MEI(商品名称)	商品マスタ.商品名称を直接転記する
16	16	HATTYU_SUI(発注数)	発注明細.発注数.を直接転記する
17	17	HATTYU_GAKU(発注金額)	発注金額は、発注数に商品単価を掛け合わせて求めます。

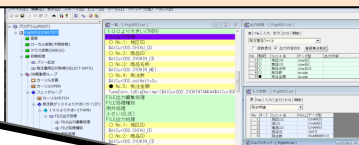
カーソル定義された内容
をテストする項目を
生成

IF文の条件を自動判定し、
テスト項目を
生成

各出力項目ごとの編集
仕様をテスト項目として
生成

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

```
import java.util.List;
import java.util.Map;

import org.springframework.batch.item.ItemProcessor;
import org.springframework.batch.item.ItemProcessor;

public class SampleBatchItemProcessor implements ItemProcessor<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



```
/*+ ordered USE_NL(TB_X
TB_Y)
INDEX(TB_X INDEXNAME)
INDEX(TB_Y INDEXNAME) */
```

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



```
import java.util.ArrayList;
import java.util.List;
import java.util.Map;

import org.springframework.batch.item.ItemProcessor;
import org.springframework.batch.item.ItemProcessor;

public class SampleBatchItemProcessor implements ItemProcessor<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 Java (バッチ) 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。

2.11 ランタイムライブラリ

- 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
- プロシージャ起動コマンド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を提供
- 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



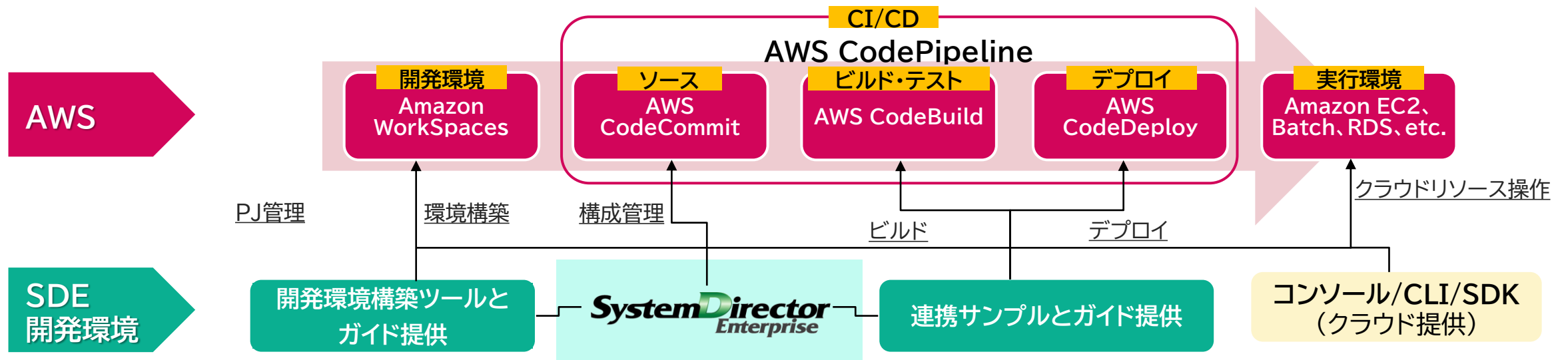
汎用的な部品を、プロジェクト毎に作るのではなく、既存部品を適用することで、品質向上と効率化が図れる。



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] ー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 ー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] ー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

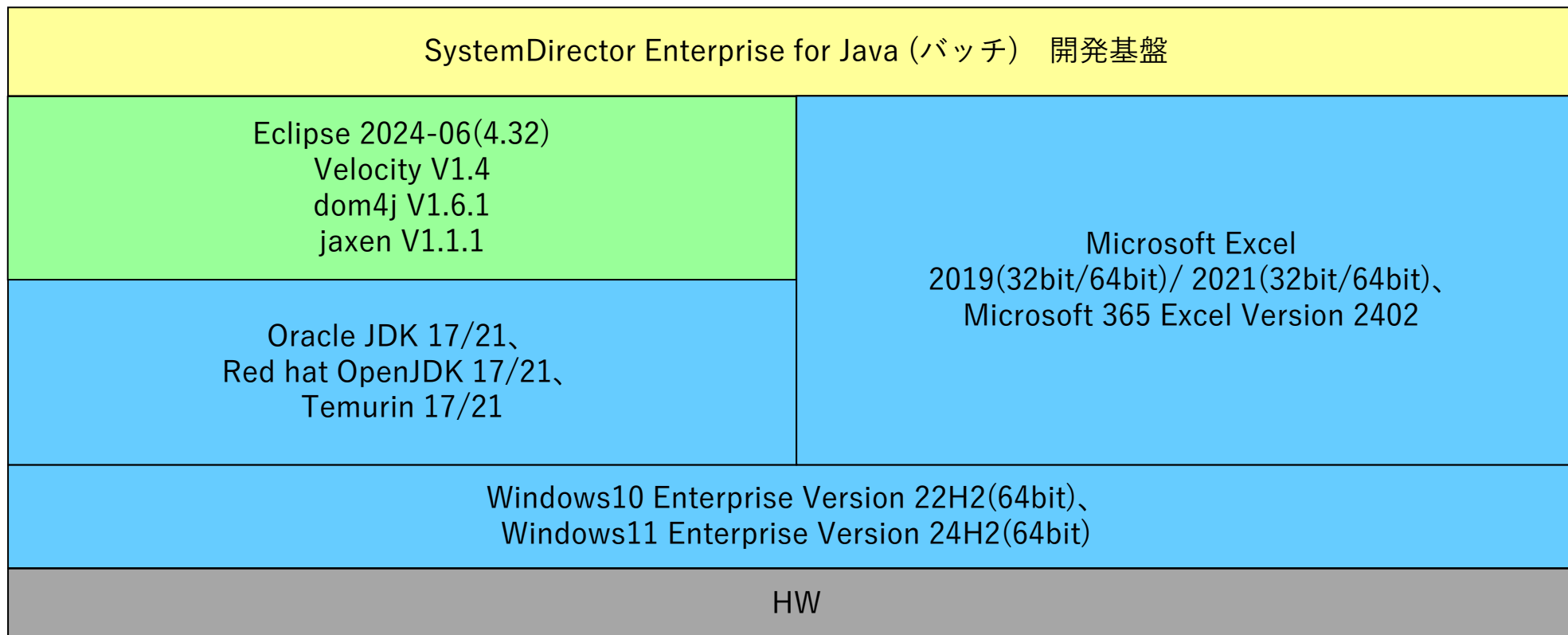
2.12 クラウド開発支援機能

- CI/CD連携を容易に実現できる、AWSのCI/CDパイプラインサービスの利用手順を示すガイドとサービス特有のビルド・デプロイの設定を含むサンプルが利用可能
- 関連するソフトウェアのインストールやセットアップを簡単に行うことができる開発環境構築ツールが利用可能



第3章 ライセンス体系

3.1 動作環境(開発環境)



凡例

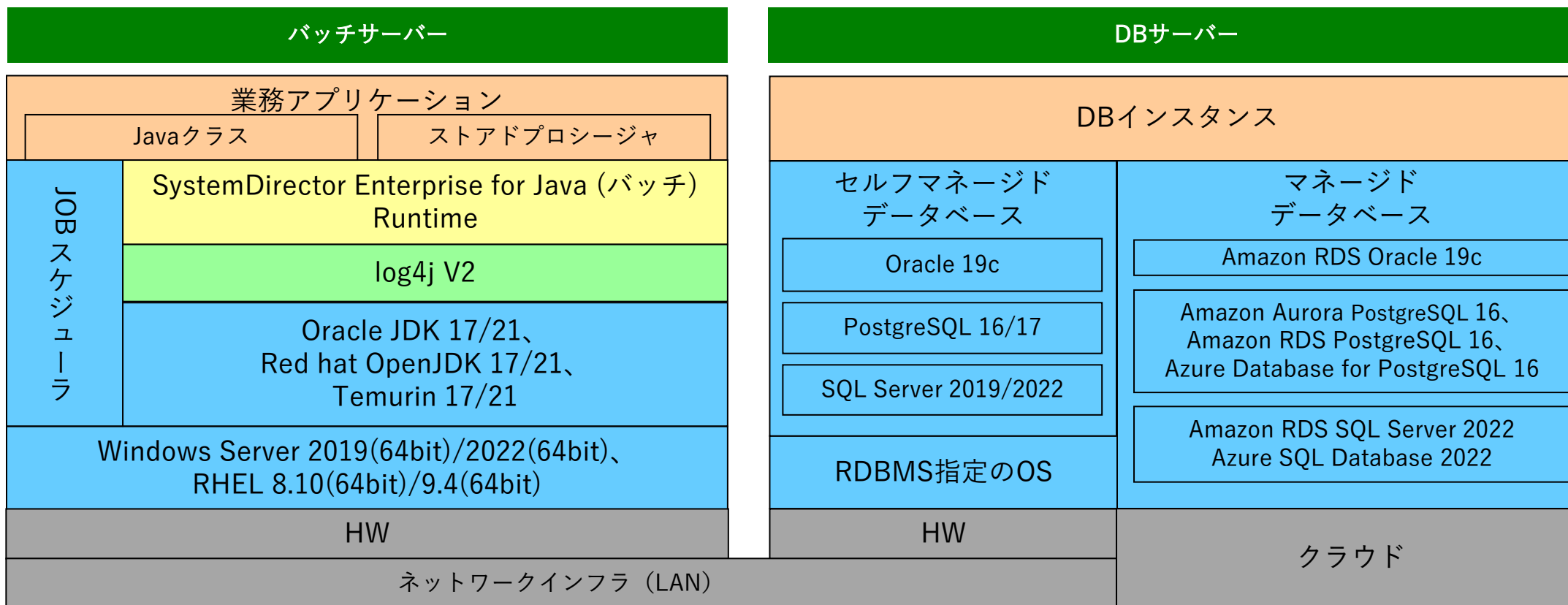
ハードウェア

調達品ソフト

SystemDirector Enterprise for Java (バッチ)
OSSツール

SystemDirector Enterprise for Java (バッチ)
自製ツール

3.1 動作環境(実行環境)



※ AWSおよびAzureでは、オンプレミス環境でサポートするすべてのOSおよびセルフマネージドデータベースをサポートします。

※ 実行環境サポートOSとDBサーバと文字コードの組み合わせは下記をご覧ください。

https://jpn.nec.com/SystemDirectorEnterprise/Batch/requirement_java.html

凡例

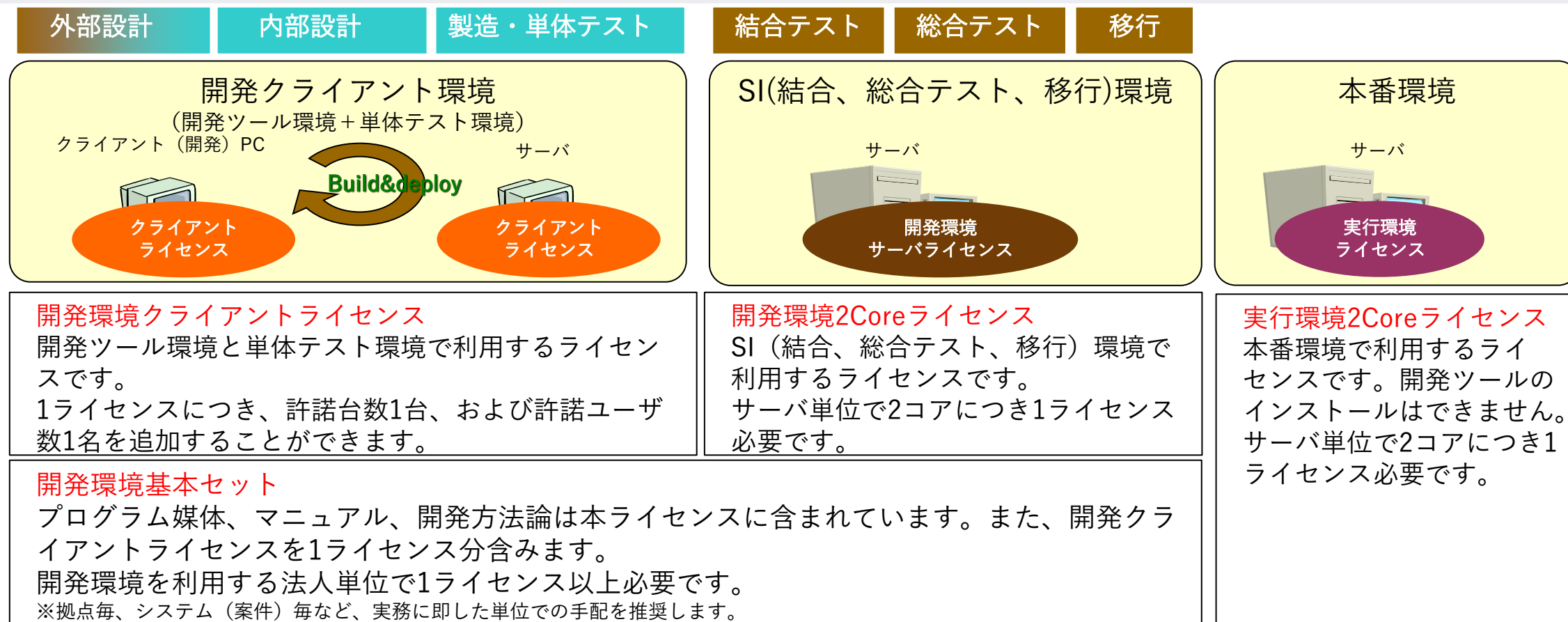
ハードウェア またはクラウド	調達品ソフト	SystemDirector Enterprise for Java (バッチ) OSSツール	SystemDirector Enterprise for Java (バッチ) 自製ツール	プロジェクトが 開発した資産
-------------------	--------	----------------------------------------------------	---------------------------------------------------	-------------------

3.1 動作環境(実行環境)

以下のコンテナ環境をサポートします。

コンテナオーケストレーション	ホストOS	ベースイメージ
Amazon ECS 1.86.3	Amazon Linux 2023	Red Hat Universal Base Image 8
Amazon EKS 1.30	Amazon Linux 2023	Red Hat Universal Base Image 8

SystemDirector Enterprise for Java (バッチ)を構成するライセンス



※システム移行時に並行稼働期間が3カ月を超える場合は開発環境ライセンスの購入が必要になります。

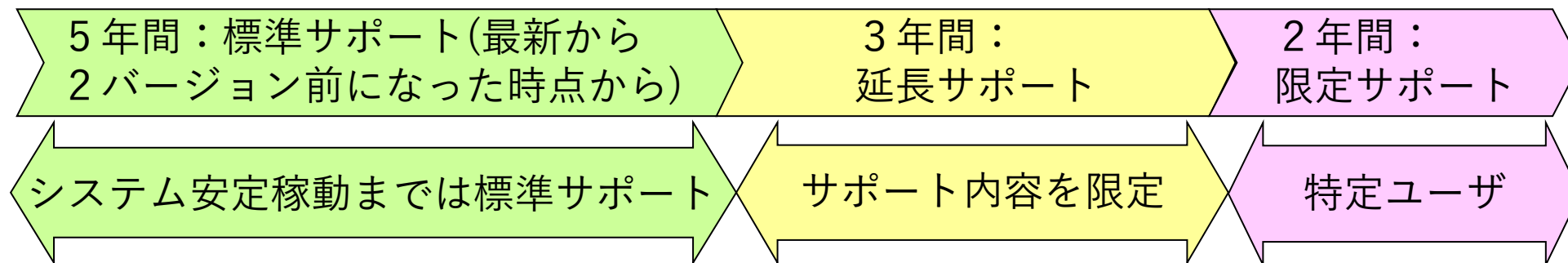
第4章 関連情報



様々な業種、様々な規模で導入実績あり

導入事例	開発本数
スーパーPOSシステム	約 7 5 0 本
地方銀行基幹系システム	約 2、 0 0 0 本
自動車リース系システム	約 3、 5 0 0 本
商社系EDIシステム	約 6 0 0 本
半導体グローバル生産管理システム	約 1、 0 0 0 本
コンビニ店舗本部システム	約 1、 5 0 0 本
外食商品管理システム	約 5 5 0 本
流通業基幹システム	約 1、 6 0 0 本

トータル10年間 + α のサポートを保証



レスポンスサービス	電子メールでの問い合わせ対応	電子メールでの問い合わせ対応	電子メールでの問い合わせ対応
バージョンアップサービス	バージョンアップ申請を受け付けての対応	バージョンアップ申請を受け付けての対応	バージョンアップ申請を受け付けての対応
リビジョンアップサービス（障害パッチ提供）	フル提供	障害は既存パッチのみ。クリティカル、セキュリティ関係は新規対応	障害は既存パッチのみ。新規パッチは別途有償



SystemDirector Enterprise サポート窓口では
業務システム構築基盤に関する様々なご質問やご相談に
お応えします。例えば・・・

新しい技術をどのように適用すれば良いか
業務アプリケーションの開発や保守を効率化するにはどうしたら良いか
既存資産をどう活用したら良いか



などお気軽にご相談ください

ご購入前のお問い合わせ

SystemDirector ご相談窓口

 Eメール contact@systemdirector.jp.nec.com

- SystemDirector Enterprise, NeoFace, WebOTX は日本電気株式会社の登録商標です。
- Windows, Windows Server, Excel, SQL Server, Visual Studio, .NET Framework, Internet Explorer, Microsoft Edge 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。
- Oracle, Java, JavaScript, JDK, WebLogicは、Oracle Corporationおよびその子会社、関連会社の米国およびその他の国における商標または登録商標です。
- 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 LLC の商標です。
- iPad, Mac OSは、米国および他の国々で登録されたApple Inc.の商標です。
- iOSは、米国Ciscoのライセンスに基づき使用されています。
- Tomcat は、Apache Software Foundation の登録商標または商標です。
- Docker は、Docker Inc. の米国およびその他の国における登録商標もしくは商標です。
- PostgreSQL はPostgreSQL の米国およびその他の国における商標です。
- Linux はLinus Torvalds 氏の米国およびその他の国における登録商標または商標です。
- Red Hat Enterprise Linuxは、米国およびその他の国における Red Hat, Inc. の登録商標です。
- 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。

- 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。

BluStellar