

Orchestrating a brighter world



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

日本電気株式会社

本資料では SystemDirector Enterprise for Javaを、「for Java」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. お問い合わせ先

付録 リアルタイム開発

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

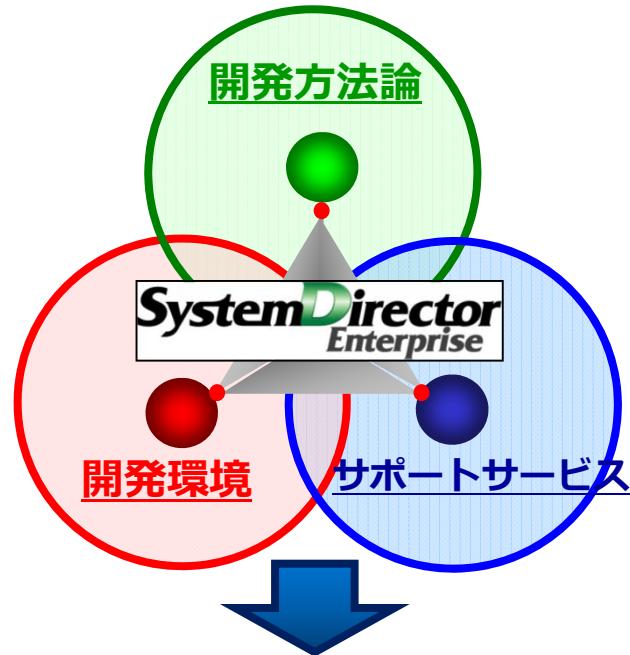
生産性の向上

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

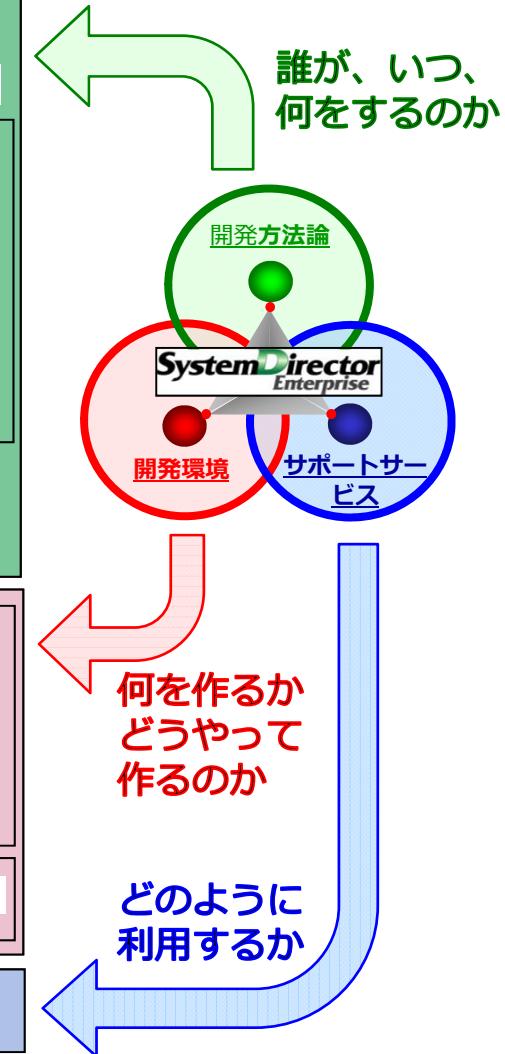
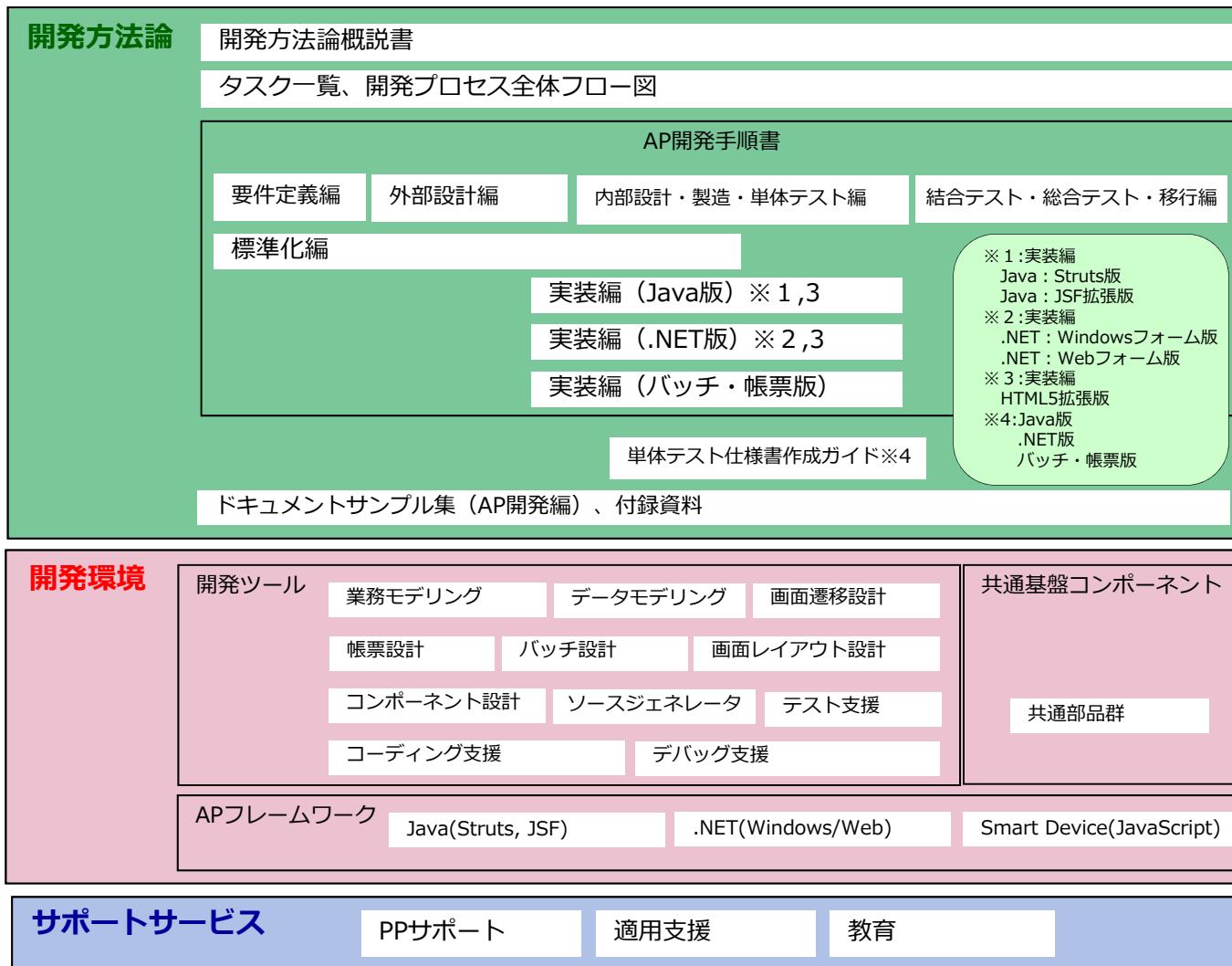
- ・開発方法論で標準作業の
効率化
- ・開発環境で生産性向上

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。

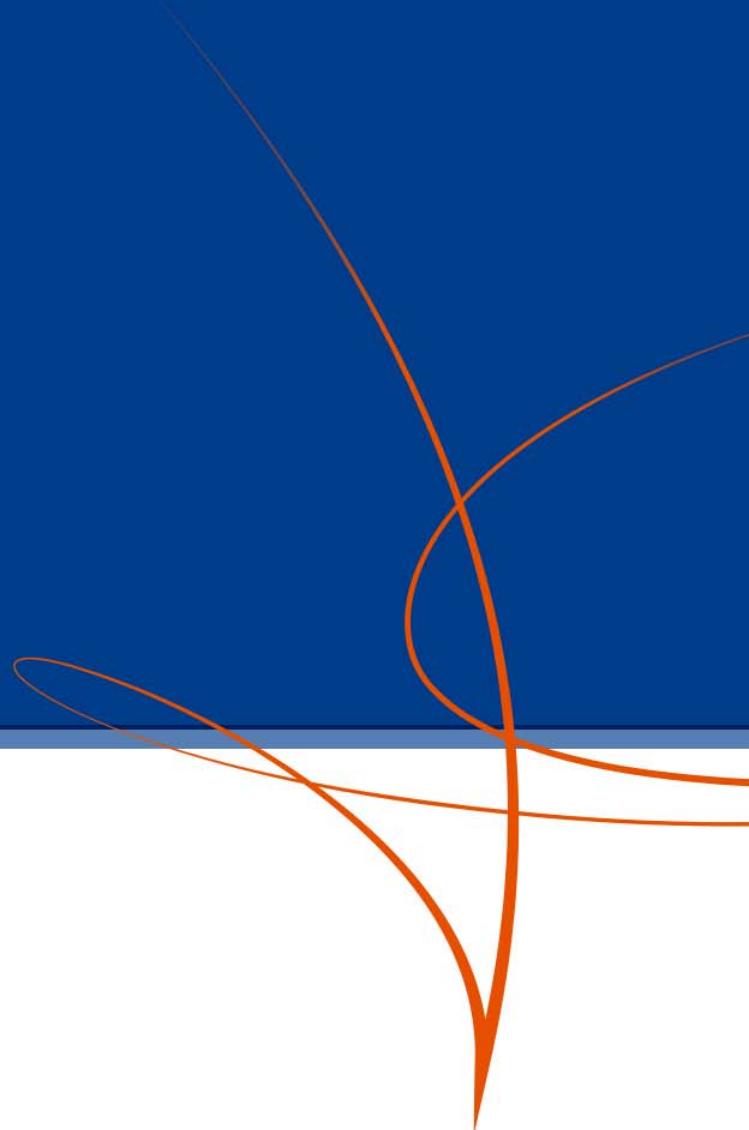


1.4.SystemDirector Enterprise の製品ラインナップ

- 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

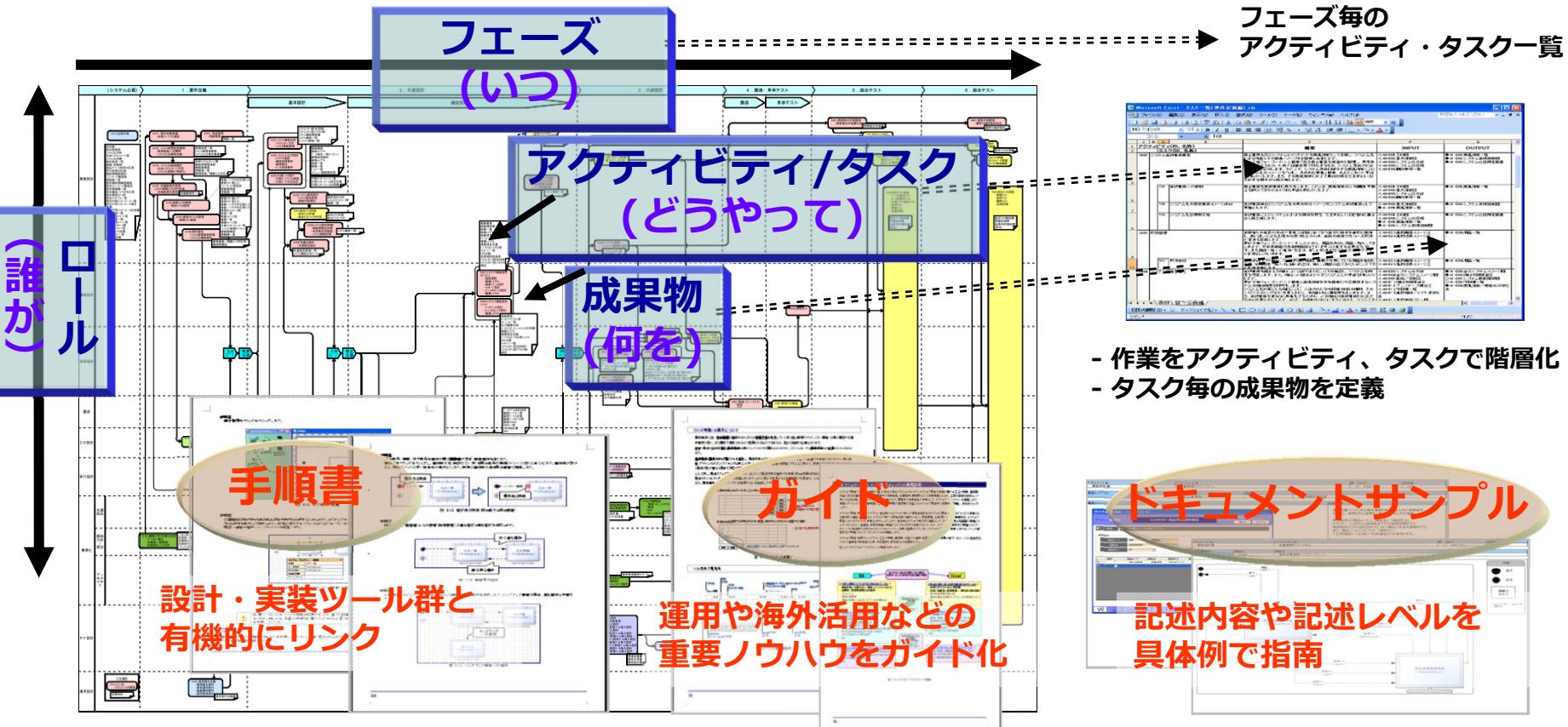


2.1.概要

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



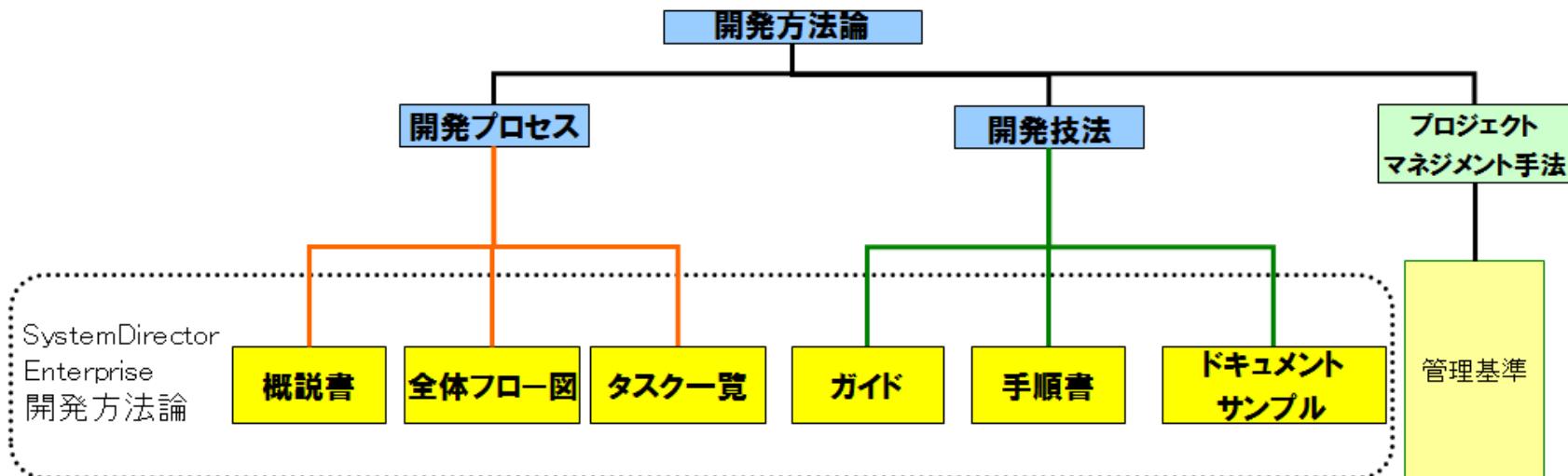
2.2.一般的な開発方法論との違い

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。



2.3.共通フレーム対応

業界標準に準拠した体系をサポートします。

本開発方法論で定義しているフェーズは、業界標準（共通フレーム2013：SLCP-JCF2013）に沿った体系となっています。本開発方法論は、共通フレーム2013の要件定義・開発プロセスをサポートしております。

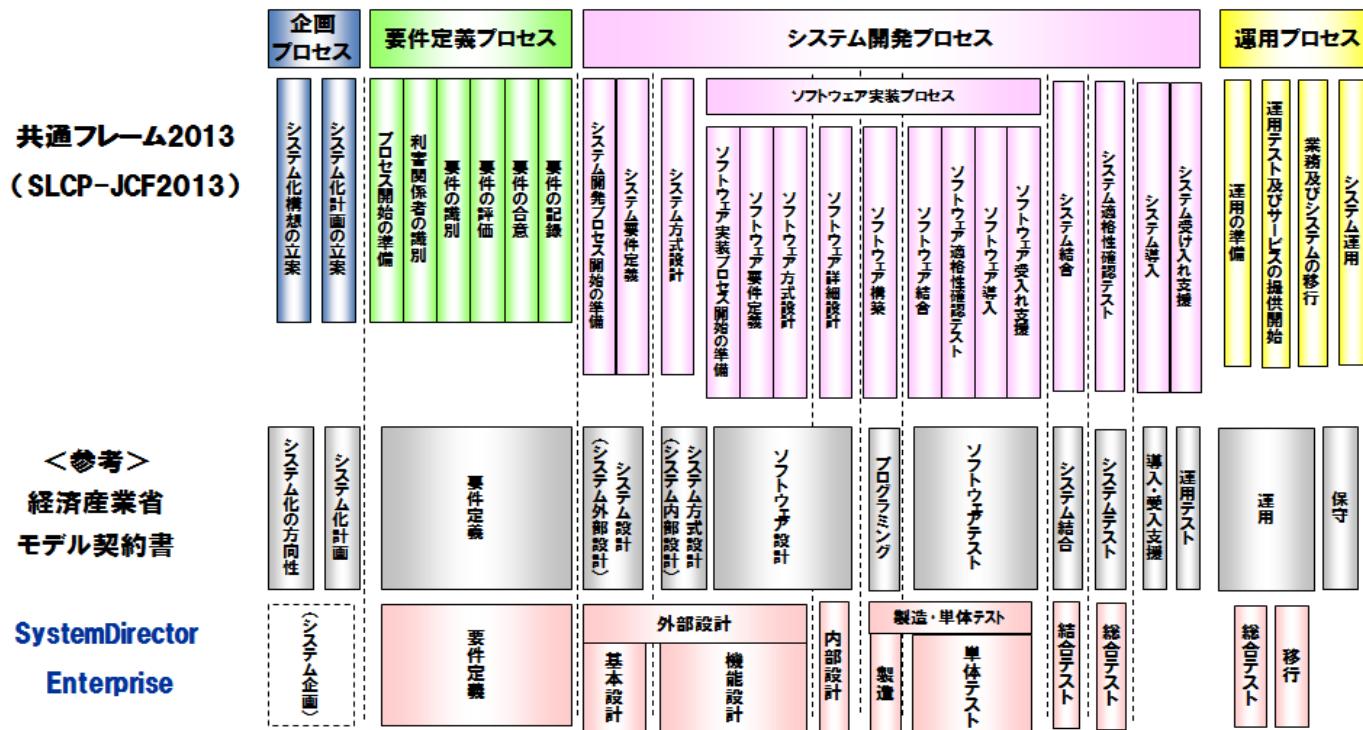


図1 プロセスレベルの対応関係

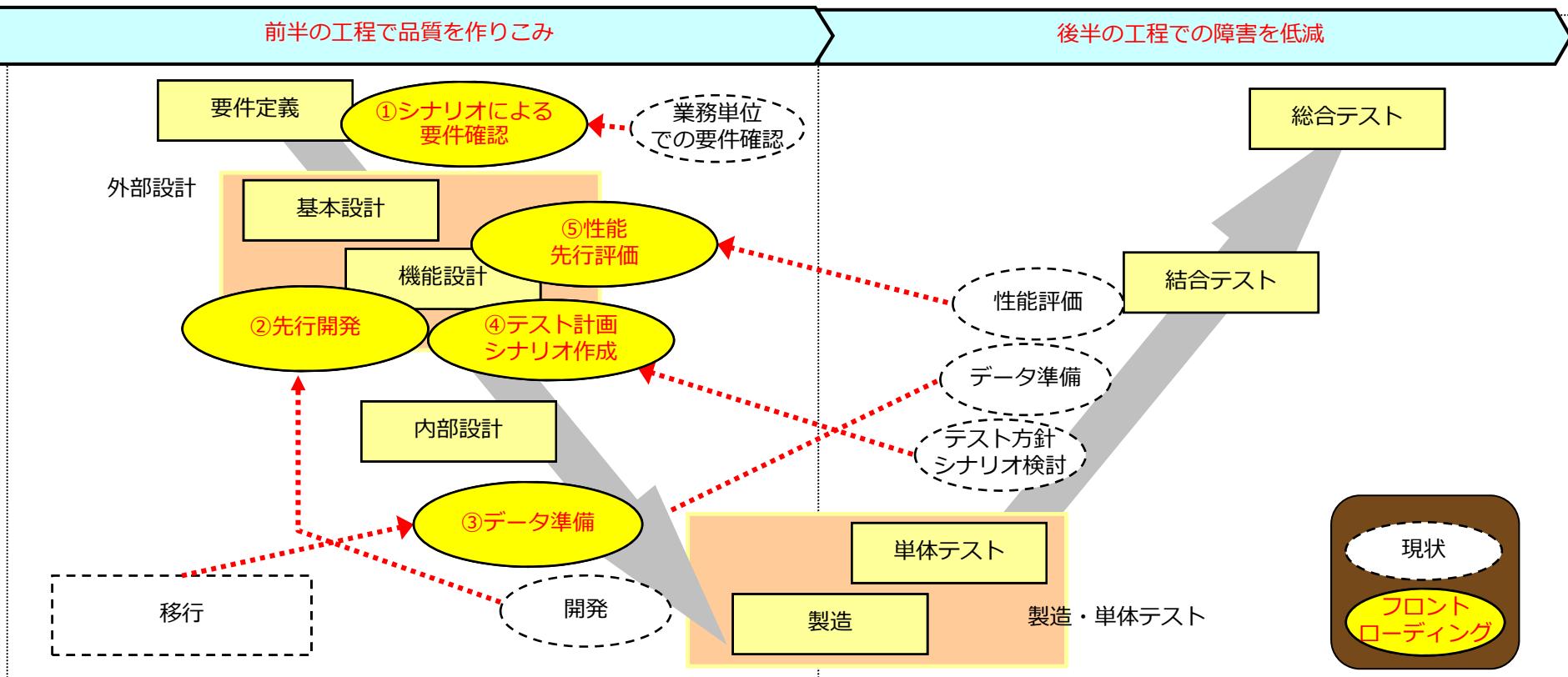
※ 共通フレーム2013との関係はウォータフォールモデルをベースとし、フェーズ、作業順序を考慮して整理しています。

2.4. フロントローディング

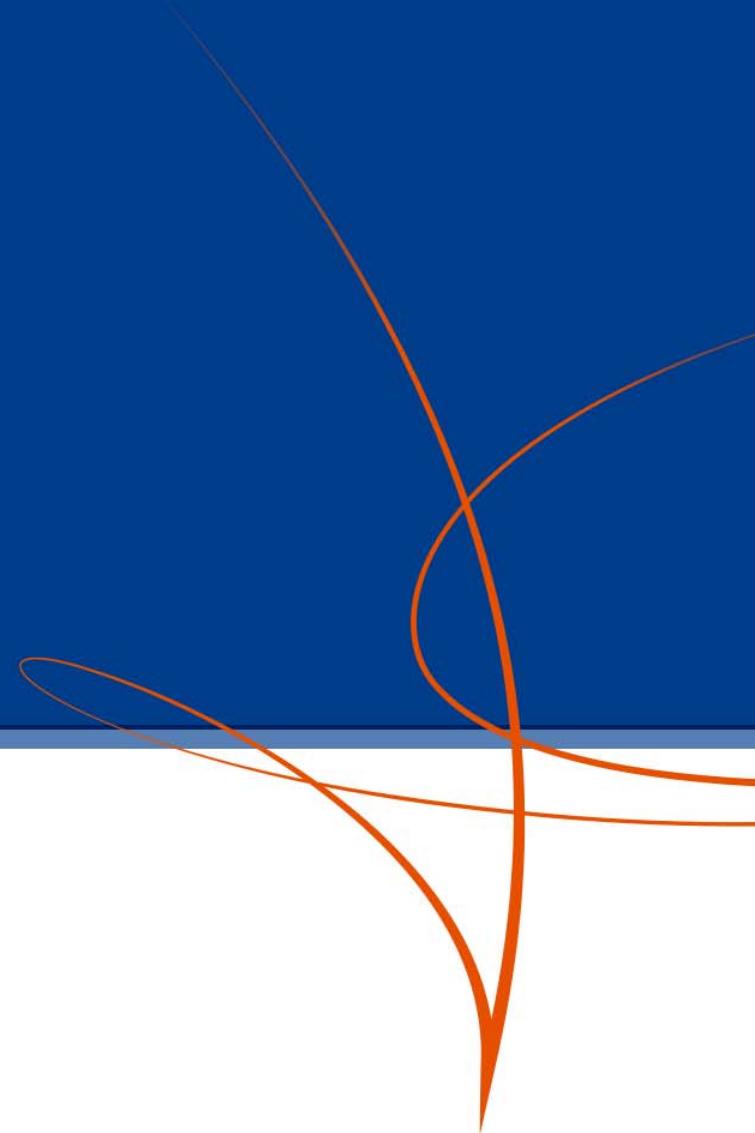
開発の前半で品質を作りこむ手法を取り入れています。

「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

Web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

- デファクトスタンダードなフレームワークをベースとしたSystemDirector Enterprise推奨のアーキテクチャモデルを提供することで、スムーズな最新技術への対応やアーキテクチャ選定の検討工数を削減

Point
2

フレームワーク

- Web 3層システムにおける業務システム構築に必要なフレームワークを提供することで、業務SEは業務ロジックに注力可能。

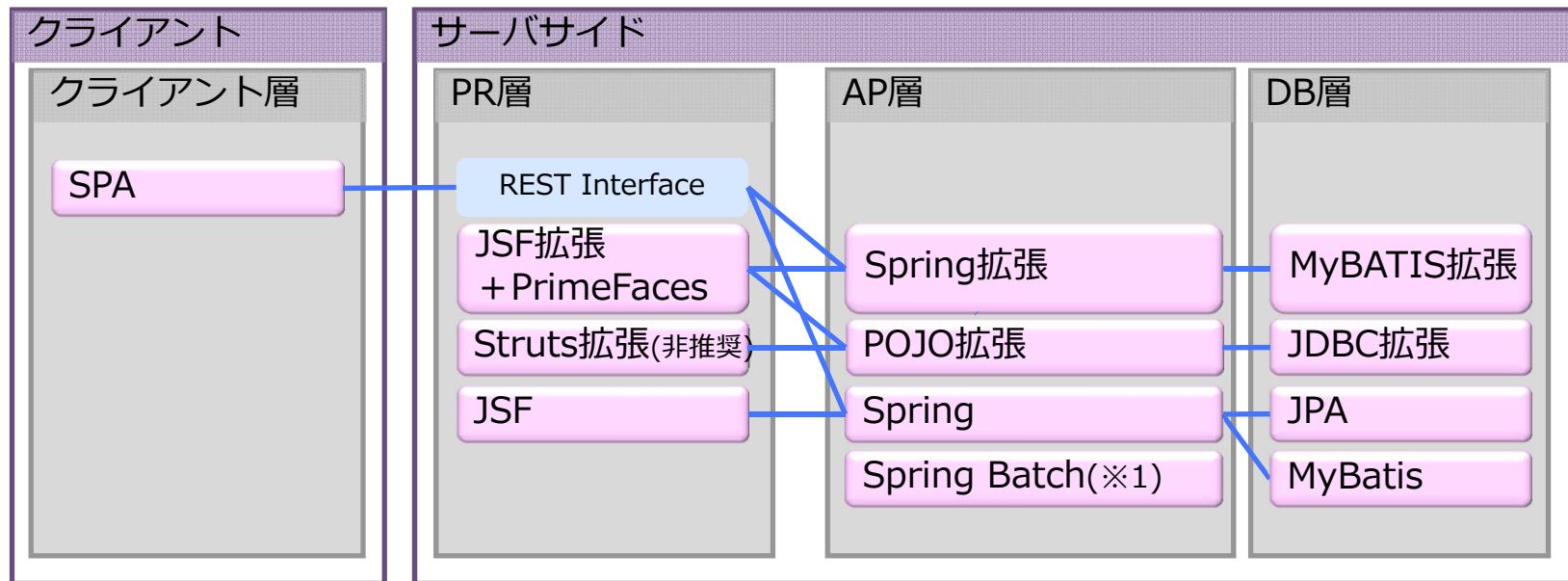
Point
3

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 Web 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）等の開発環境を提供

3.2.アーキテクチャモデル

デファクトスタンダードなJavaフレームワークをベースに独自拡張や推奨の組み合わせを定義したアーキテクチャモデルを提供します。



※各線の接続はツールかガイドで提供
※1 小規模簡易バッチAP開発向け

3.2.1. HTML5拡張モデル(SPA)の新設

サーバと疎結合で保守性の良いSPAアプリケーションを自動生成



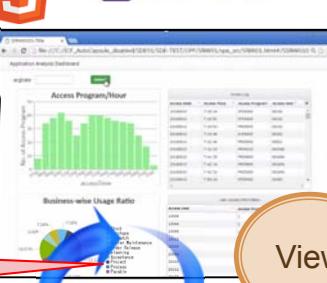
- HTML5、BootStrapによる表現豊かなユーザインターフェースを提供
- レスポンシブウェブデザインにより、マルチデバイス要件にも対応可能
- Angular、MVVMアーキテクチャによる保守性の高いソースコード
- オフライン要件にも対応可能

クライアント

HTML/CSS
A HTML5
B Bootstrap



B Bootstrap



画面サイズに合わせて
画面を自動切り替え



ANDROID

JavaScript

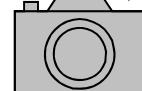
A ANGULARJS
by Google

View
Model

Controller

Model

Service



ローカル
デバイス



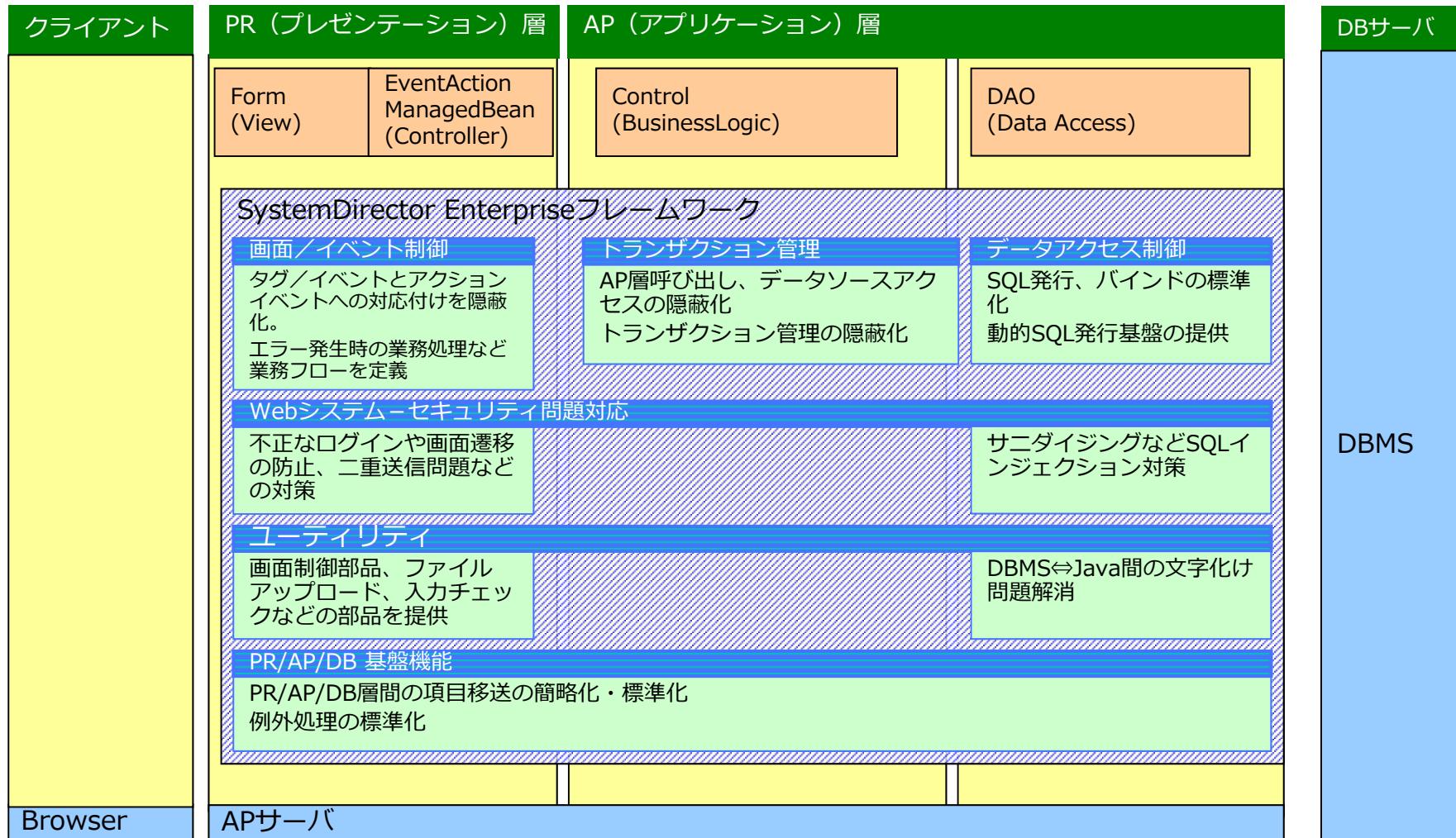
ローカル
ストレージ

REST/
JSON

サービス

3.3.フレームワーク

Web 3層システムにおける、業務システム構築に必要なフレームワークを提供することで、業務SEは業務ロジックに注力可能です。



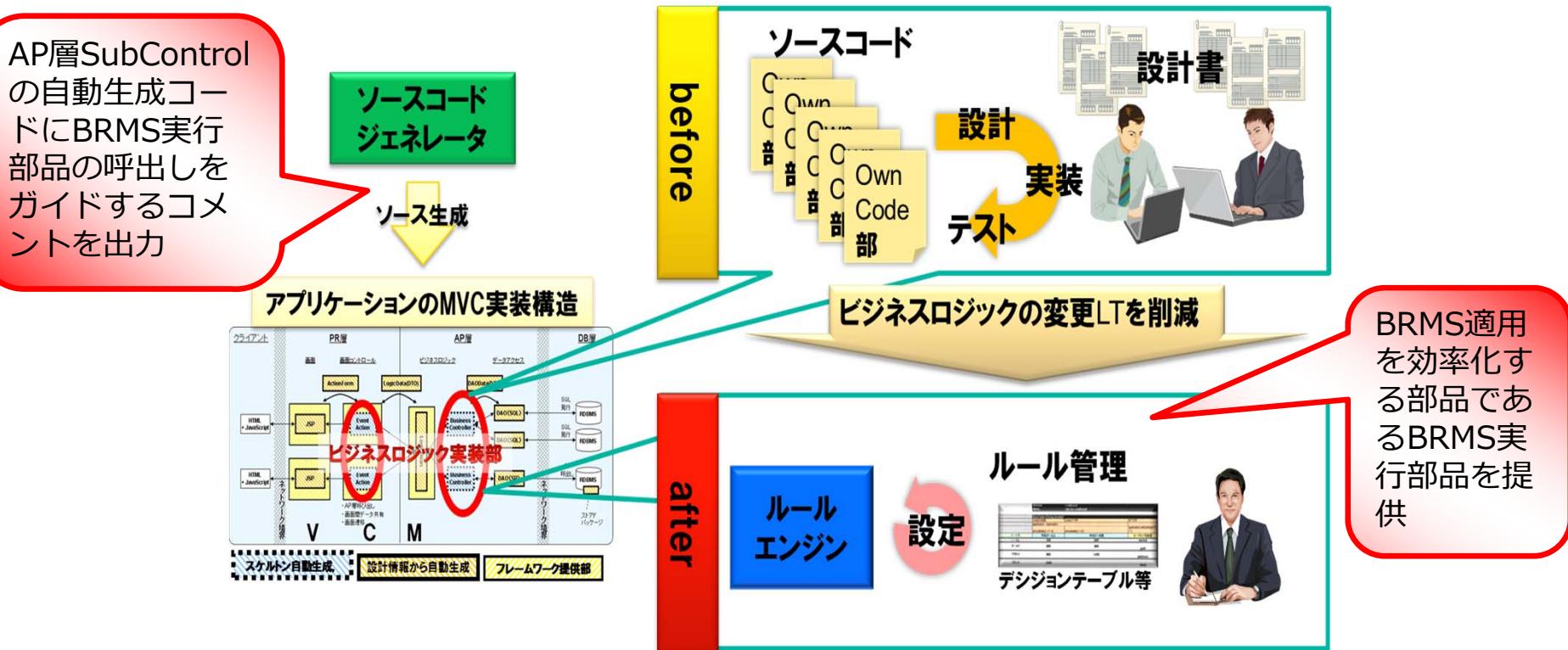
3.3.1. ユーティリティ（共通部品）

部品の機能一覧

- 算術関数機能
- 統計関数機能
- 文字列操作機能
- SQL文エスケープ機能
- 日付・時刻フォーマット機能
- 数値フォーマット機能
- サニタイジング機能
- 文字列比較機能、文字列変換機能
- 文字種検査・サロゲートペア検査機能
- エンコーディング機能
- メール送信機能
- CSV機能、ファイル操作機能
- GZip圧縮／展開機能、Zip圧縮／展開機能
- 外部プログラム実行機能
- ストップウォッチ機能、タイマー実行機能
- DNS問い合わせ機能
- クライアント情報取得機能
- 非永続マップ機能
- メッセージダイジェスト算出機能、対称鍵暗号機能、バイナリ符号化機能

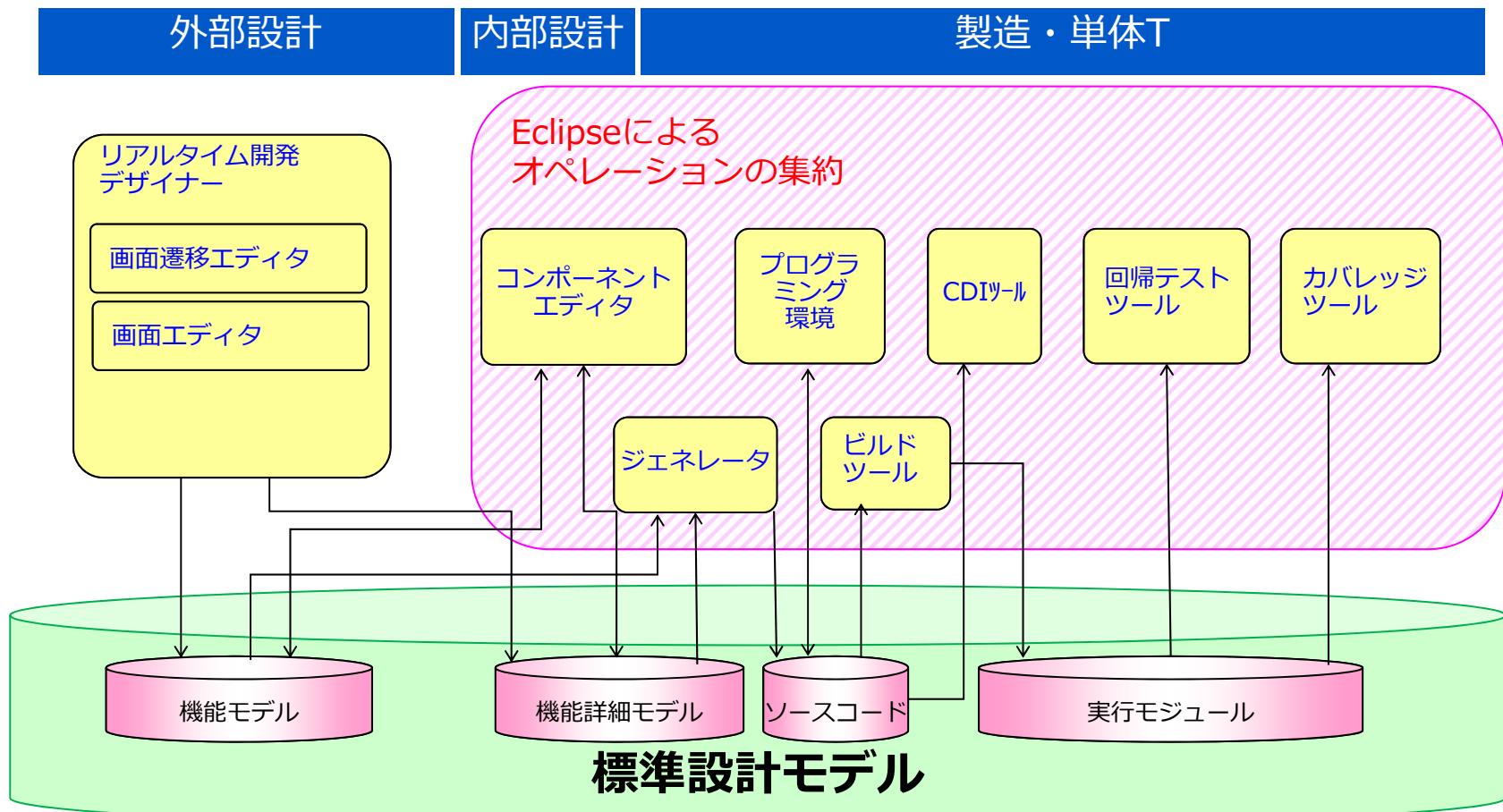
3.3.2.BRMS実行部品

BRMS適用を効率化する部品であるBRMS実行部品を提供します。BRMSを利用したシステムでは、業務ロジックをデシジョンテーブルとして表現し、ソースコードと分離する事により保守性向上や工数削減の効果が期待できます。



3.4.開発ツール

外部設計から製造・単体テストまでを対象に開発ツールを提供します。
各フェーズの成果物は標準設計モデル(リポジトリ)として格納され
次工程と連携します。



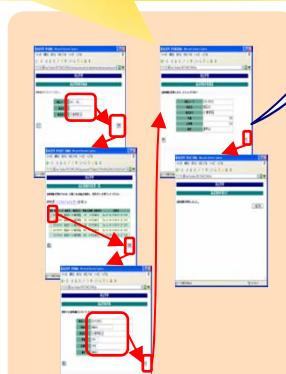
3.4.1.リアルタイム開発デザイナー

要件定義のその場でAPの作成を支援するツールです。テンプレートを活用しドラック&ドロップで開発でき、開発期間・開発コストを削減します。

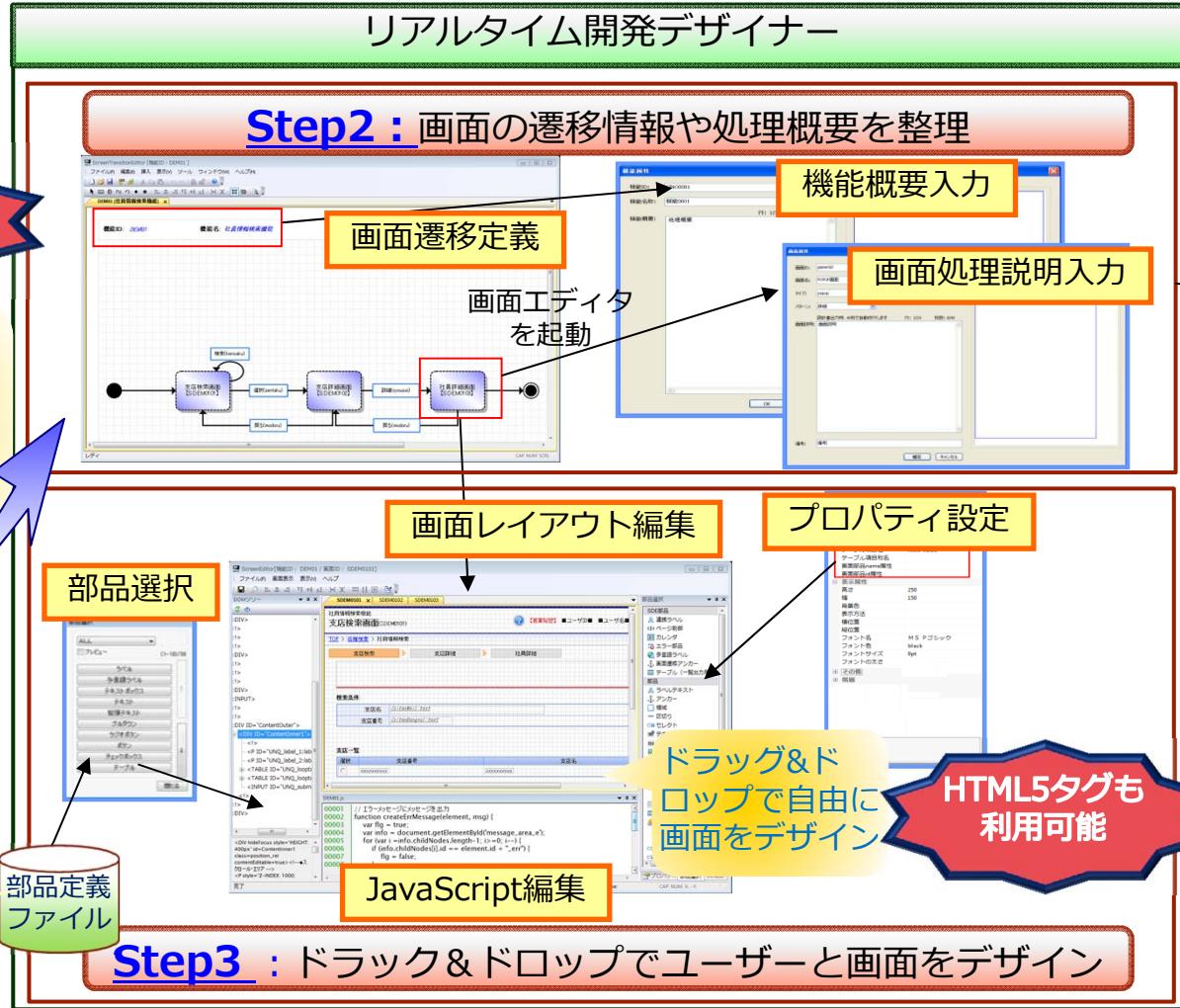
Step1:事前準備
テンプレート作成

レイアウト共通化も
実現可能

- ・画面、画面遷移、業務ロジックの自動生成のためのプロジェクト用ひな形整備



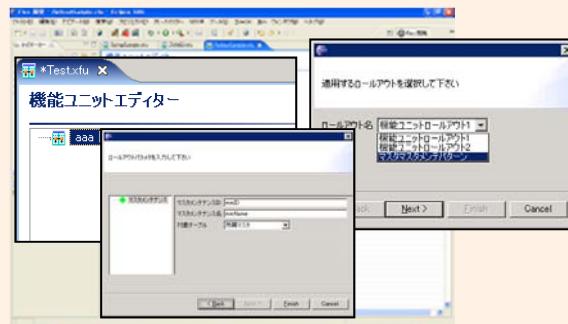
プロジェクト
カスタマイズ可能



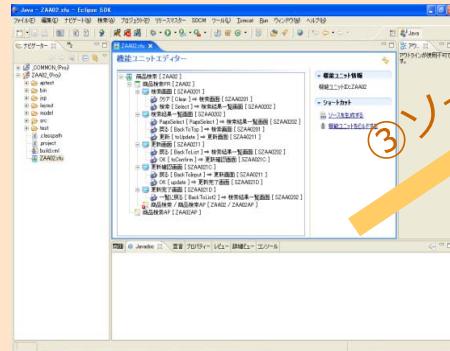
3.4.2.ロールアウト機能

あらかじめ業務画面のパターンをテンプレート化しておくことで、画面レイアウト、画面遷移制御、入力コントロール、DBアクセス処理、RESTサービスなどを一気に自動生成できます。

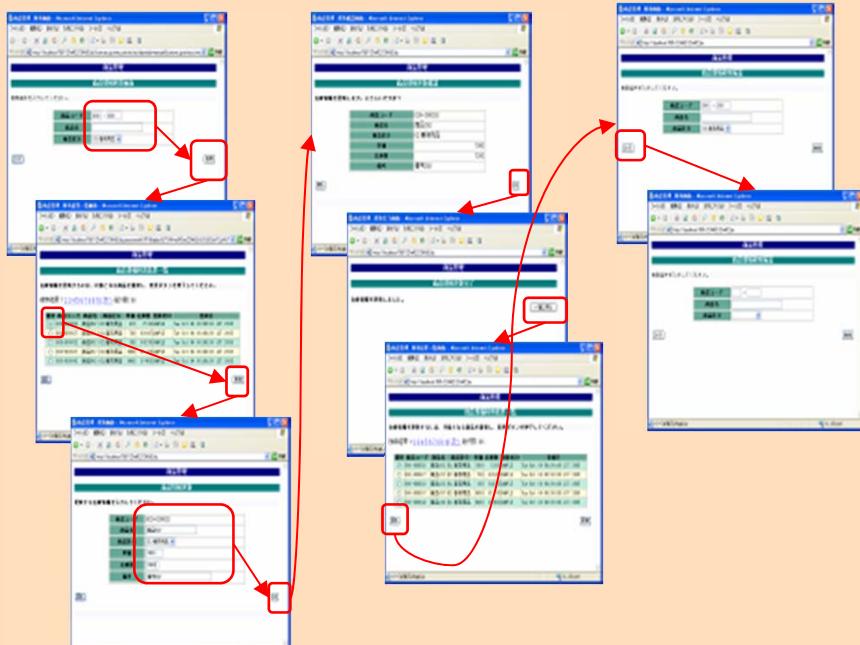
①パターンを選択しパラメータ入力



②内部設計データを一括生成

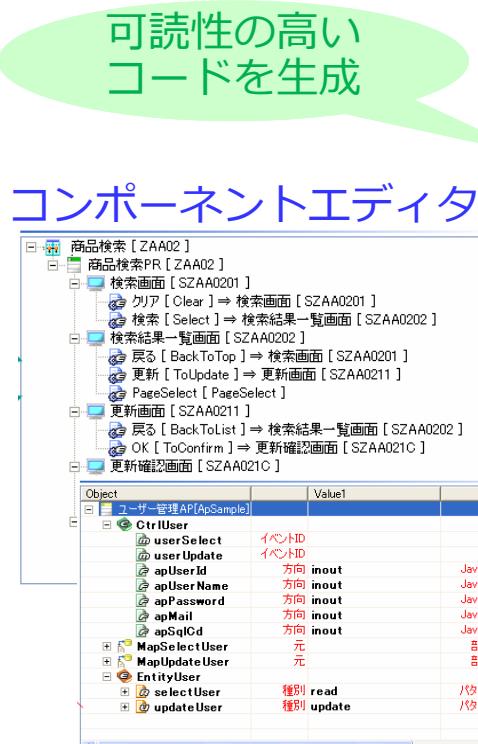


ビジネスロジックの追加のみで
アプリケーション実行可能



3.4.3.コンポーネントエディタ・ジェネレータ

主に詳細設計作業を支援する設計エディタ（コンポーネントエディタ）を提供し、データ定義やマッピング設計などを支援します。
設計情報をもとにソースコードを自動生成することで、品質・生産性向上を実現します（全コードの65%～85%程度を自動生成します）



雛形ファイル

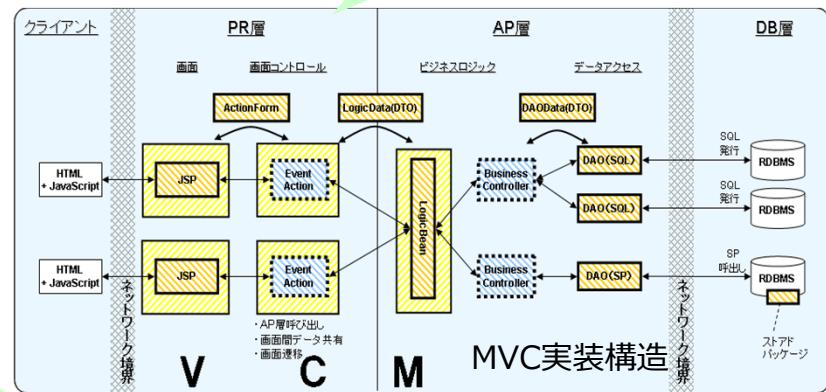
```
package ${packageName} ;  
public class ${name}Action {  
    public void execute() {  
    }  
}
```

ソースコード生成

ソースコード群

The diagram illustrates the MVC implementation structure. It shows the flow from the View layer (HTML+JavaScript) through JSP pages and Event Actions to the Model layer (Business Controller). The Model layer interacts with the Persistence Layer (DAO) and the Persistence Layer interacts with the Database (DB). Annotations like 'ネットワーク接続' (Network connection) and 'AP層呼び出し' (AP layer call) are present.

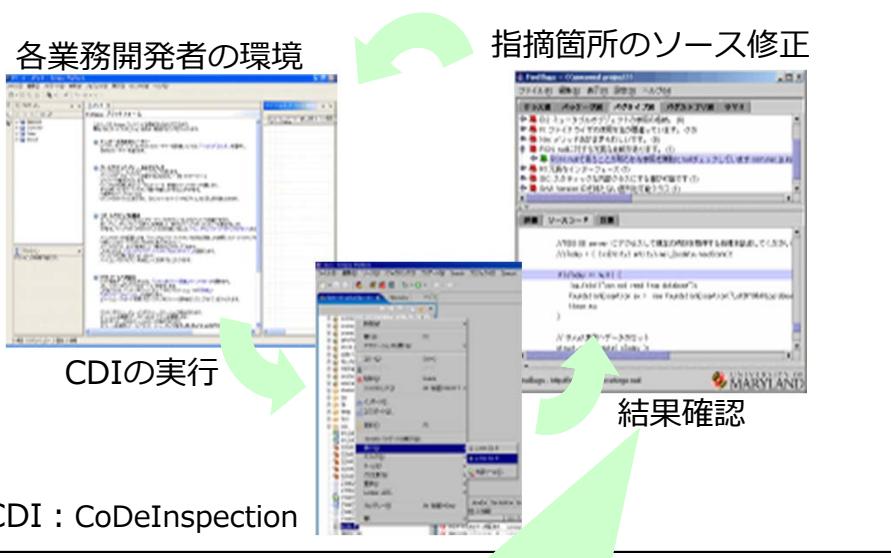
追記した業務ロジックは保護され
繰り返し自動生成が可能



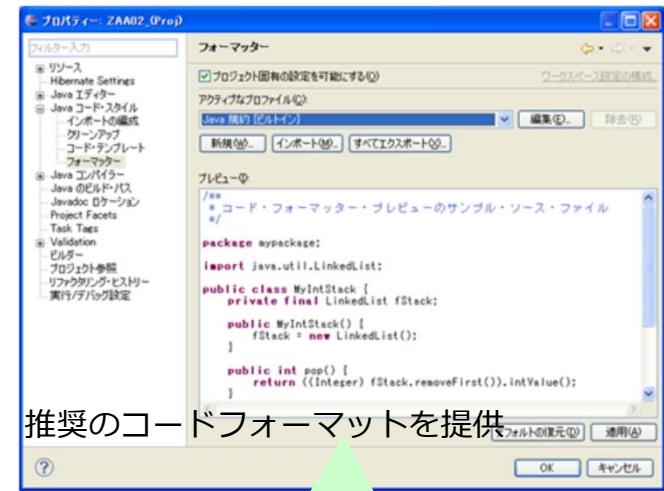
3.4.4. 製造支援ツール

CDIツール、コードフォーマッターによりオウンコーディング部分のソースコード品質の向上を支援します。

CDIツール (FindBugs)



コードフォーマッター (Eclipse)



コーディング規約を自動チェックして
レビューを効率化

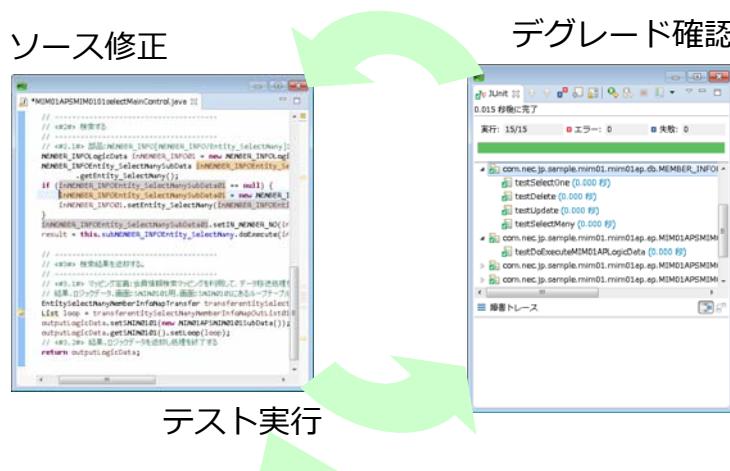
コードフォーマットを自動整形して
ソースを均一化

3.4.5. テスト支援ツール

モジュールテスト自動化ツール、カバレッジツールによりテスト品質の向上を支援します。

回帰テストツール (JUnit)

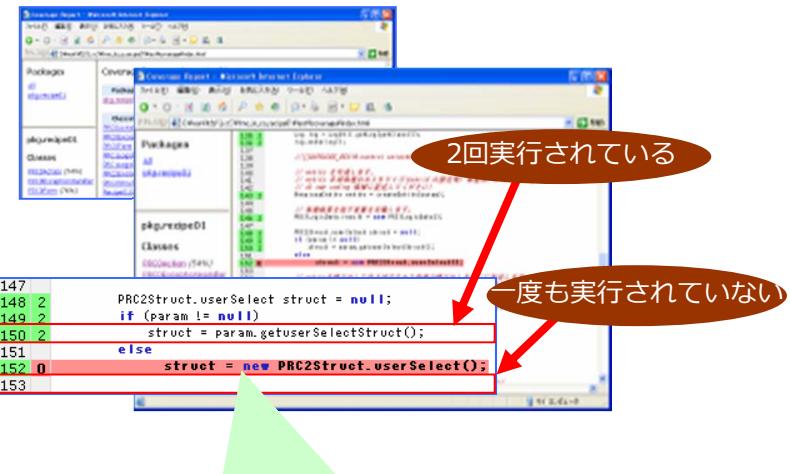
モジュールのテスト実行を自動化することで、デグレード確認のための回帰テストを効率的に実施できます。



オウンコーディング部分の
テスト駆動開発も可能

カバレッジツール (JaCoCo)

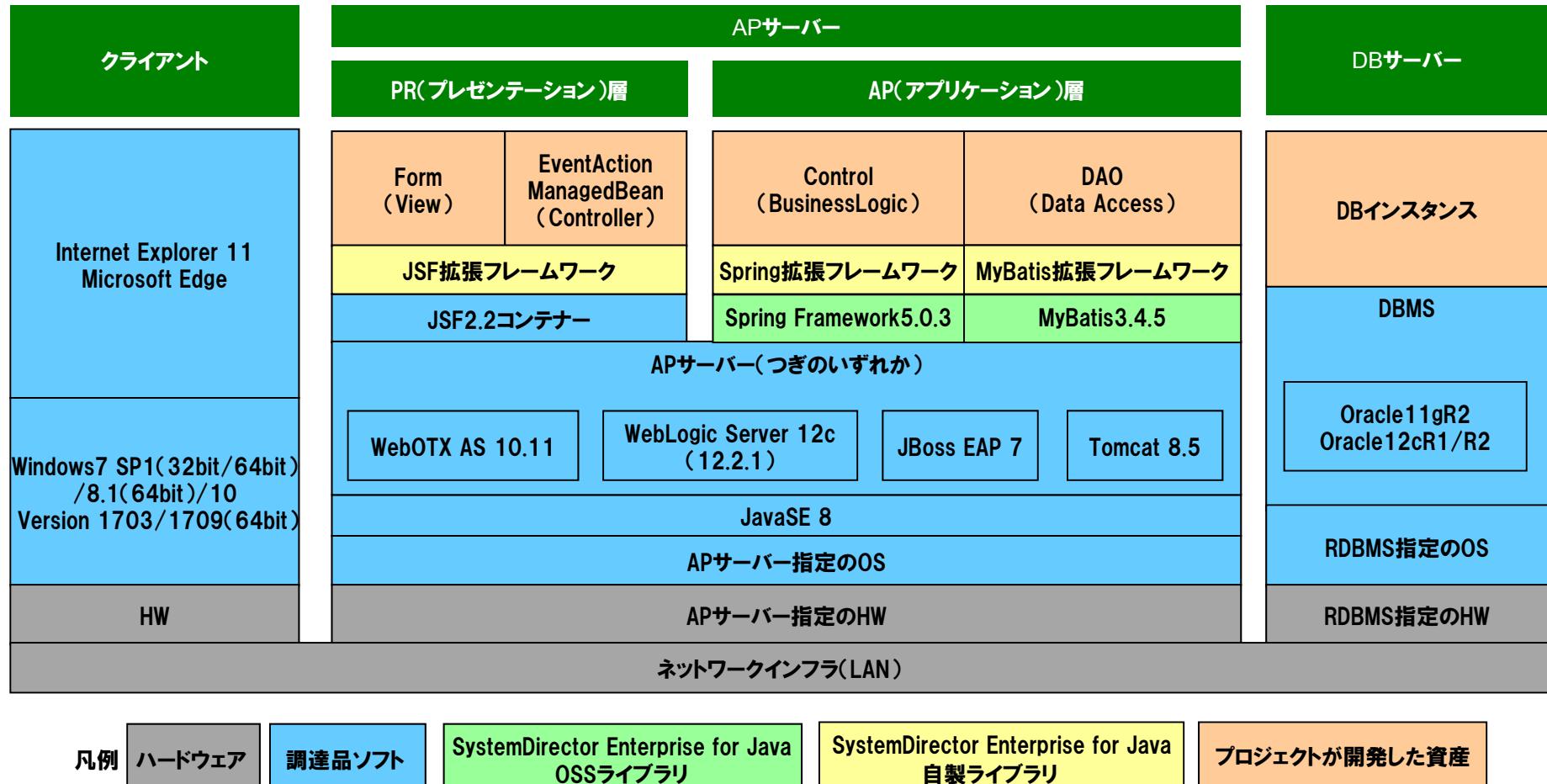
テストカバー率測定確認ツールを利用し、アプリケーションにおけるテスト網羅率を把握することができます。



網羅性を保証し、
確証としても活用可能

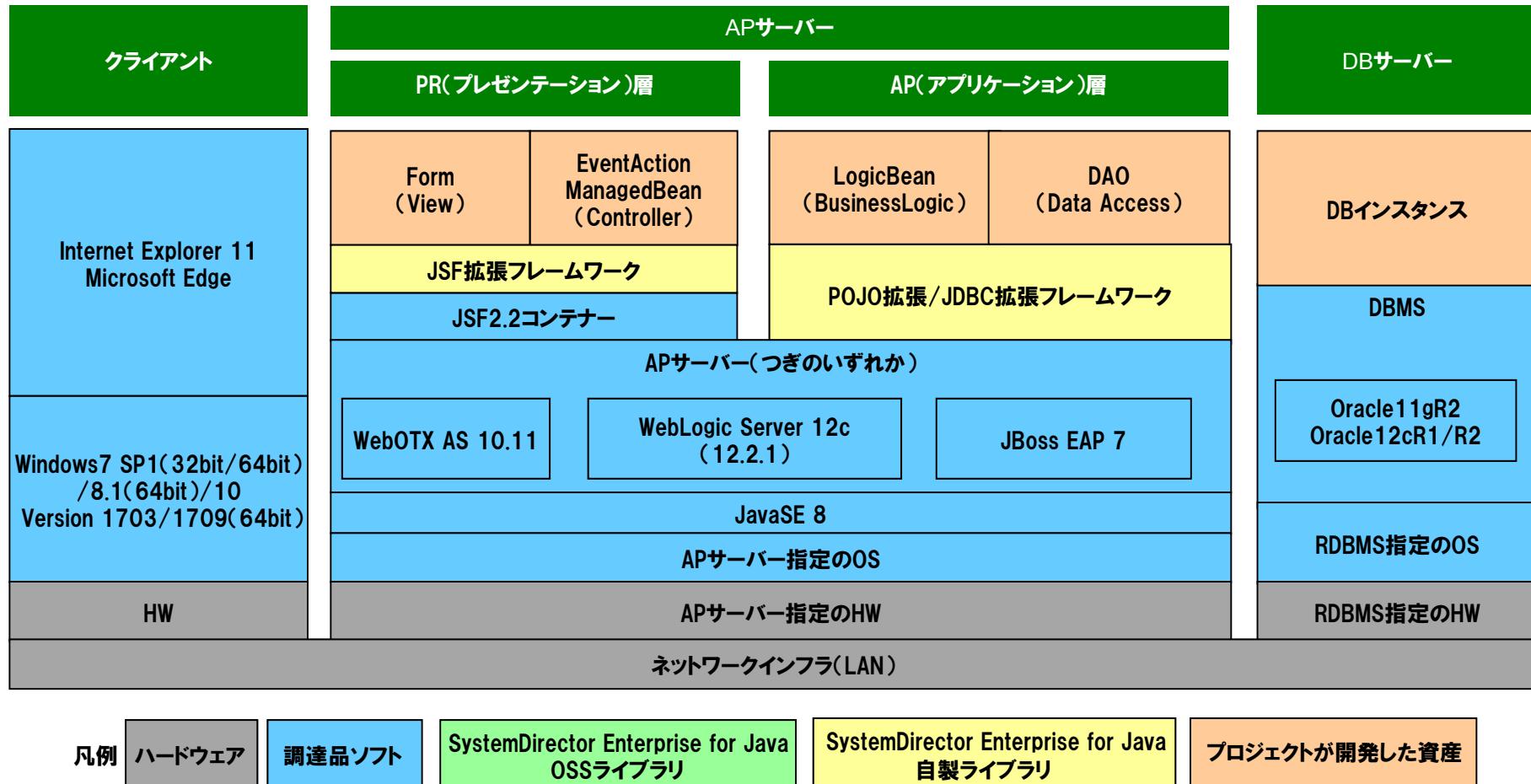
3.5.動作環境【実行環境】

JSF拡張/Spring拡張/MyBatis拡張モデル



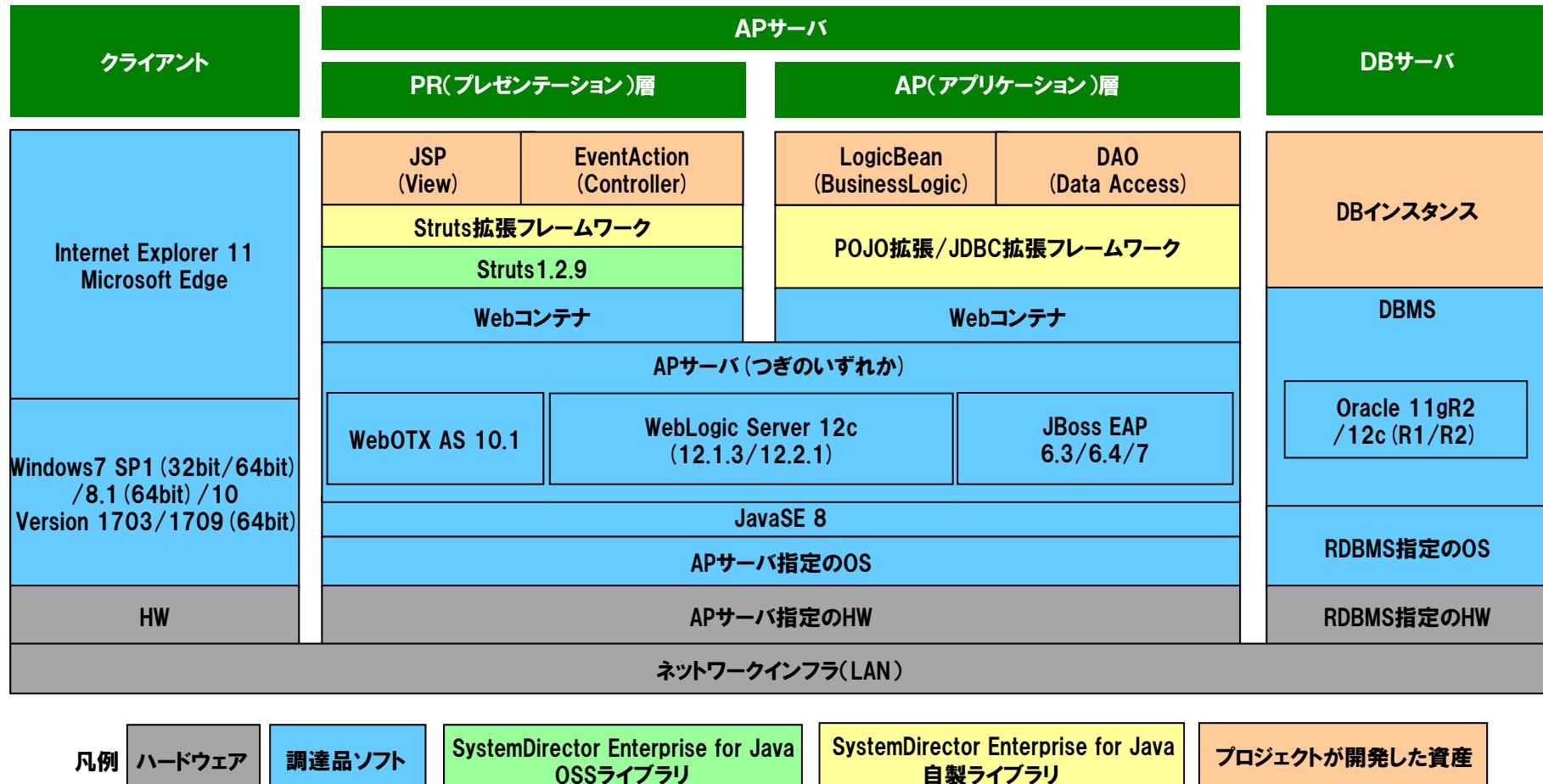
3.5.動作環境【実行環境】

JSF拡張/POJO拡張/JDBC拡張モデル



3.5.動作環境【実行環境】

Struts拡張/POJO拡張/JDBC拡張モデル



3.5.動作環境 【開発環境】

JSF拡張モデル、Struts拡張モデル、JSFモデル共通
開発環境サーバは、実行環境サーバと同一

開発環境クライアント

SystemDirector Enterprise for Java 開発基盤

Eclipse 4.7.2
FindBugs3.0.0
PMD5.2.2
EclEmma (JaCoCo) 0.7.6

JavaSE 8

Excel2016 (32bit) (※1)

Visio2016 (32bit) (※1 ※2)

Dreamweaver CS6 (※2)

Windows7 SP1 (32bit/64bit)
/8.1 (64bit)
/10 Version 1703/1709 (64bit)

HW

※1:2010/2013も利用可

※2:Struts拡張モデルのみ対応の推奨ツール

凡例

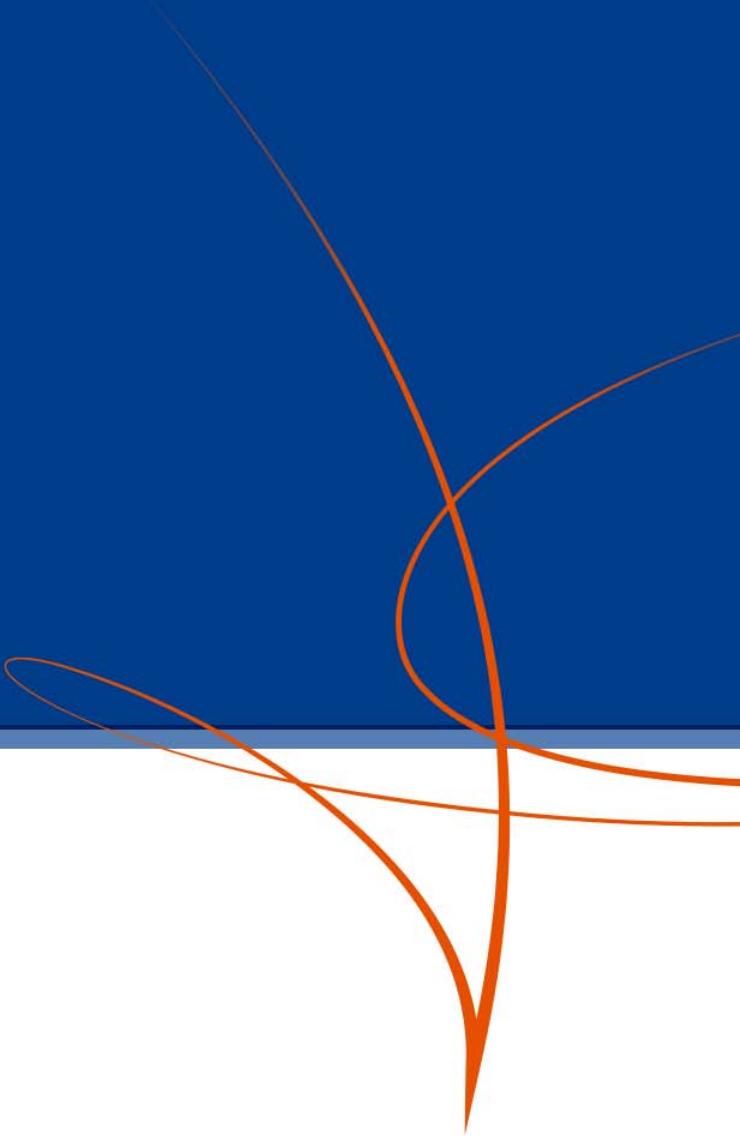
ハードウェア

調達品ソフト

SystemDirector Enterprise for Java
OSSツール

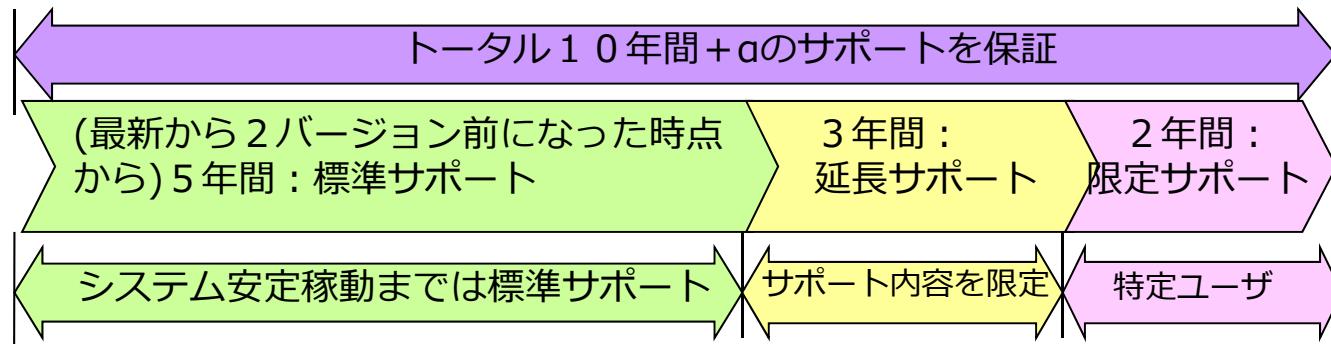
SystemDirector Enterprise for Java
自製ツール

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。



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

■ 基本サービスに含まれないものでも、ご要望があれば調整可能です。

- 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

- SystemDirector Enterpriseの全体像が把握できる研修メニューを、NECマネジメントパートナーのラーニング事業と連携して提供します。ニーズに合わせたカスタマイズメイド教育として実施します。
- 現在、定期開催されているコースはございませんが、ご要望いただければ時期・内容・場所など含めて開催調整させていただきます。お気軽にご相談ください。
- なお、SystemDirector Enterpriseのベースとなっている一般知識は前提となります。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	.NET	Smart Device
前提コース	<p>JavaによるWebアプリケーション開発1 -サーブレット・JSP-</p> <p>Javaによるデータベースアクセス</p> <p>Javaプログラミング基礎 (入門、基礎1、基礎2)</p>	<p>Visual Studio による Windowsプログラミング</p> <p>Webアプリケーション開発入門 (ASP.NET編)</p>	<p>HTML5/CSS3基礎</p> <p>JavaScriptプログラミング</p>

4.3.お問い合わせ先

● ご購入前のお問い合わせ

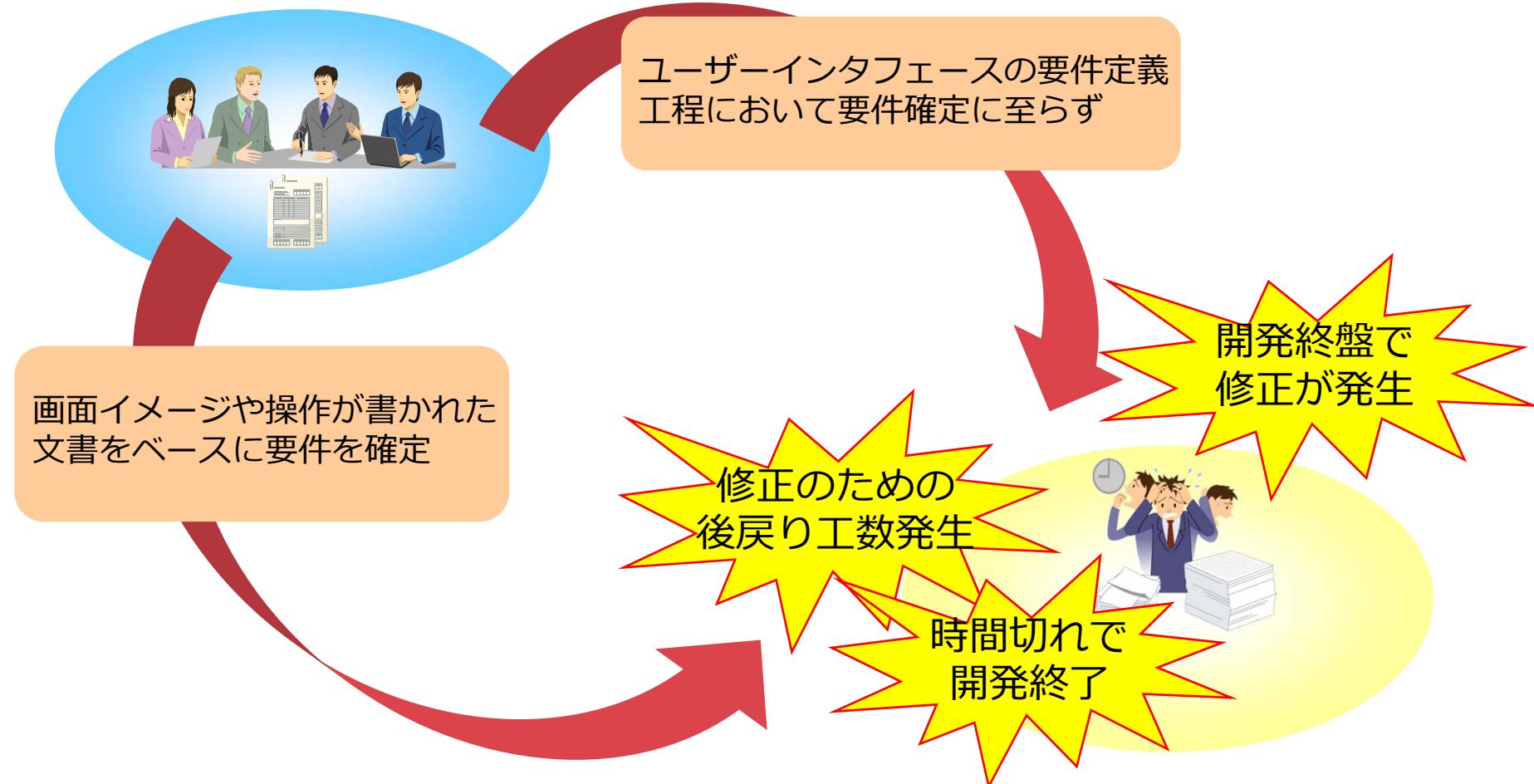
NEC SystemDirector Enterprise お問い合わせ窓口

- Web

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

付録 リアルタイム開発

要件定義工程の時点では仕様が曖昧で、開発後半に認識の相違やイメージが異なっていることが判明し無駄な工数、費用が発生



UIの要件確定を上流の早い段階で短期に実施

発注者、利用者の
CSが向上します

利用者



発注者
(レビュー)



開発者



- ・実際に動くAPにデータを表示して合意
- ・設計書も出力してレビュー
- ・微調整はその場で修正・再作成して確認

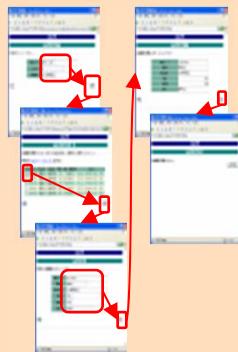
・動くAPを利用者に
渡して要件確認

リアルタイム開発の流れ

事前準備

Step0

共通的に利用できる
AP雛型の作成



画面、画面遷移、
業務ロジックの
自動生成のため
のプロジェクト
用雛型を整備し
ておく。

この整備には、
ロールアウト機
能を活用する。

画面工ディタ（リアルタイム開発デザイナー）

Step1

画面の遷移情報や処
理概要を整理



雛型の
活用



画面遷移パター
ンが決定される。



設計書情報とし
て、機能概要や
画面処理説明を
与える。

Step2

ドラッグ&ドロップで
ユーザと画面をデザイン

適宜部品を選択
しながら画面を
レイアウト。



部品定義
ファイル

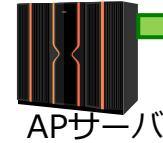
プロジェクト
カスタマイズ可能



プロパティの設
定やJavaScript
編集も実施する。

Step3

ワンクリックで設計書
とソースを自動生成



APサーバ

デ
プ
ロ
イ

Step4

実際に動かしながらユ
ーザと要件を最終確認



ブラウザから起
動して、画面遷
移やデータ表示
イメージなどを
動的に確認。



Step1. テンプレートによる画面遷移・画面パターンの選択・決定

既存システムを分析し、代表的な画面遷移パターンと画面レイアウトパターンをあらかじめテンプレート化（パラメータ化しており自由に変更可能）。テンプレートをベースに、画面デザインを行うことで画面仕様決定の迅速化、標準化を促進

テンプレートでは埋められない業務固有項目や、
レイアウトや遷移の微修正は、画面デザイナから
追加定義可能。
複雑な業務要求、画面構成にも対応可能

画面デザイナーからテンプレートの選択・読み込みが
可能。
GUIベースでイメージが見られるため、ユーザー
との認識齟齬を大幅削減。

代表画面テンプレートから選択して画面定義の
ベースを決定。
画面検討を
検討型 ⇒ 選択型へ

ユーザー様

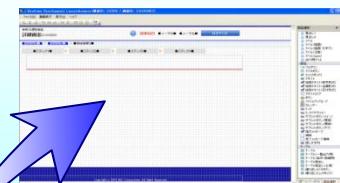


S E



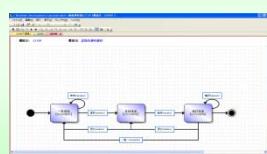
業務固有項目は
追加でカスタマイズ

リアルタイム開発デザイナー



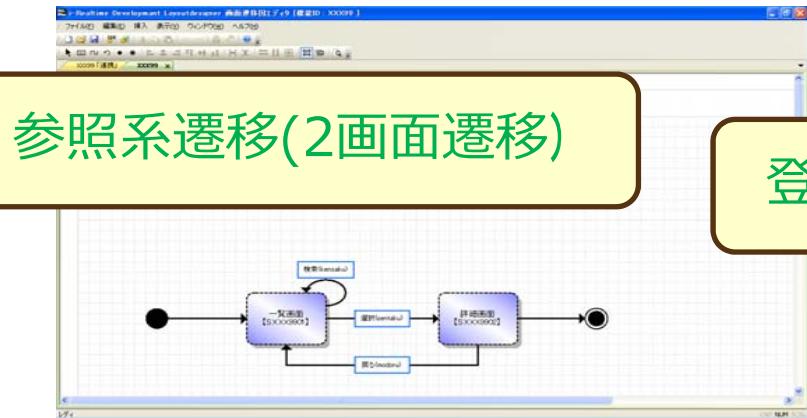
選択

代表画面遷移・デザインテンプレート

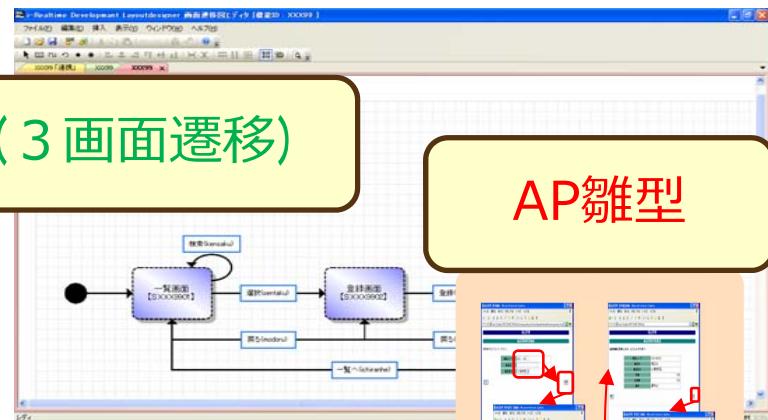


Step1. 事前準備～AP雛型の作成

対象システムの要件を分析し、画面、画面遷移、データファイル読込を実装したAP雛型を作成



参照系遷移(2画面遷移)



登録系遷移 (3画面遷移)

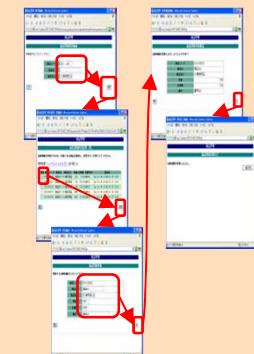
AP雛型



一覧画面テンプレート

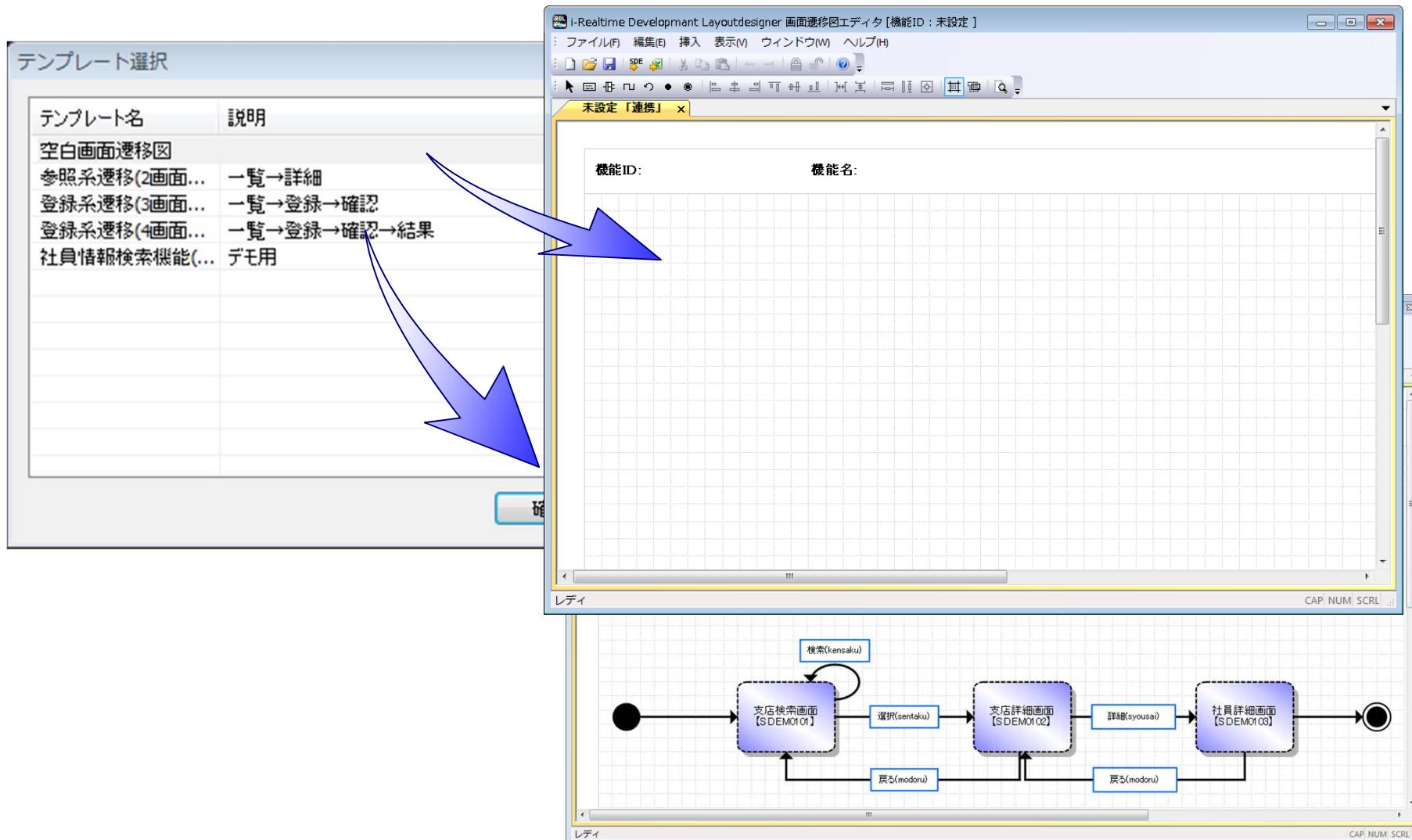
詳細画面テンプレート

登録画面テンプレート



Step2. 要件確認～画面遷移設計

事前に作成した画面遷移パターンから適切なものを選択



Step3. 要件確認～画面デザインのプレビュー実行

事前に作成した画面パターンから適切なものを選択

レイアウト変更(ボタンを移動、色を変更)、コンポーネントをドラッグ&ドロップで配置



Step3. 要件確認～画面デザインのプレビュー実行

画面デザイナーからプレビュー実行してレイアウト・画面遷移・JavaScript実行結果を確認

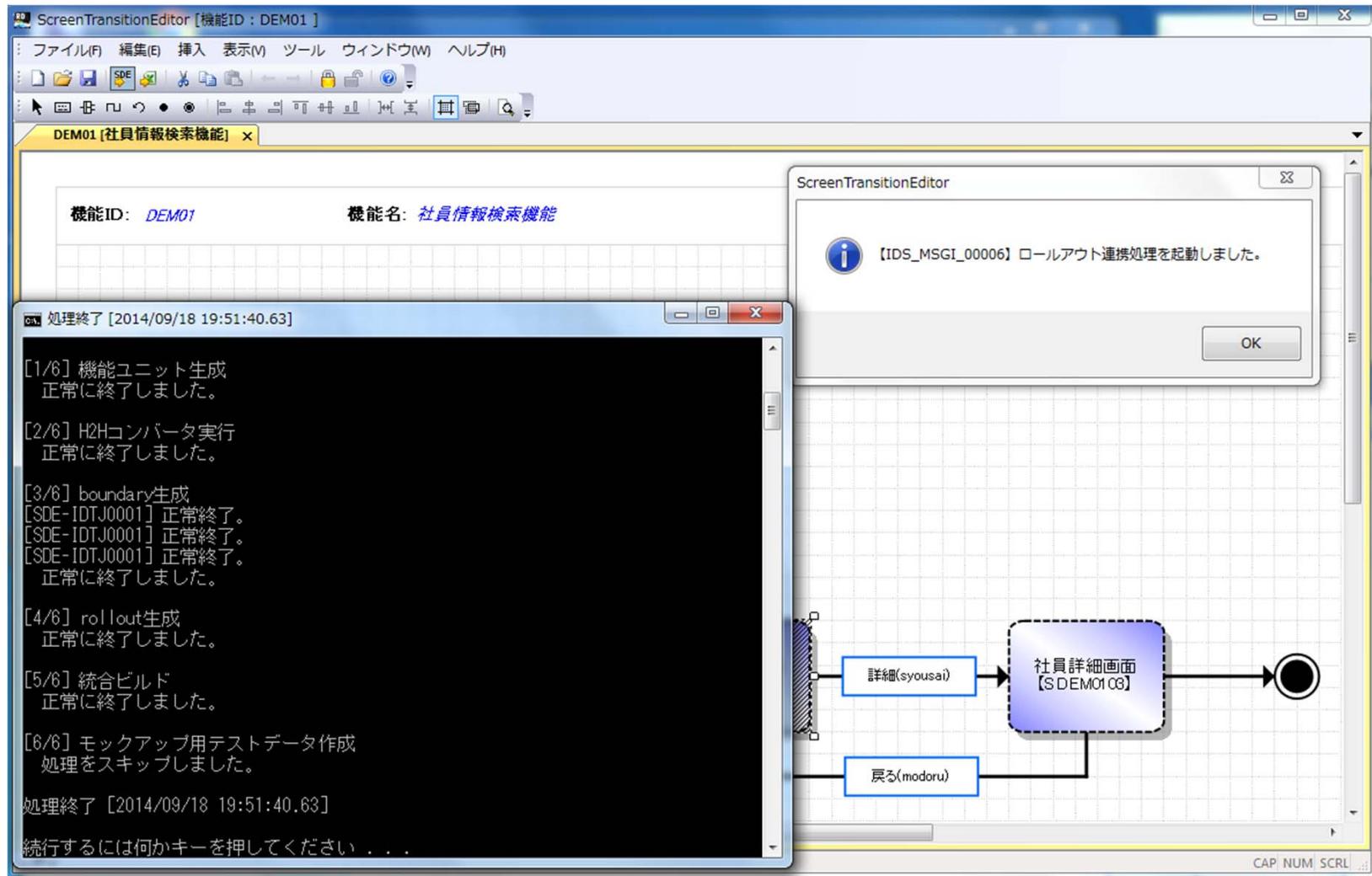
The screenshot shows the NEC ScreenEditor interface. On the left, the DOM tree and code editor for 'DEM01.js' are visible. The code editor contains the following JavaScript function:

```
00001 // エラーメッセージにメッセージを出力
00002 function createErrMsg(element) {
00003     var flag = true;
00004     var info = document.getElementById(element.id);
00005     for (var i = info.childNodes.length - 1; i >= 0; i--) {
00006         if (info.childNodes[i].id == element.id + "_err") {
00007             flag = false;
00008         }
00009     }
00010 }
```

The main preview window displays the 'Branch Search Screen (SDEM0101)' with a blue arrow pointing to the search form. The search form includes fields for 'Branch Name' and 'Branch Number'. Below the form is a table titled 'Branch List' with one row containing the value 'xxxxxx'. At the bottom right of the preview window is a 'Branch Details' button.

Step4. ソースコードの自動生成（ジェネレーター）

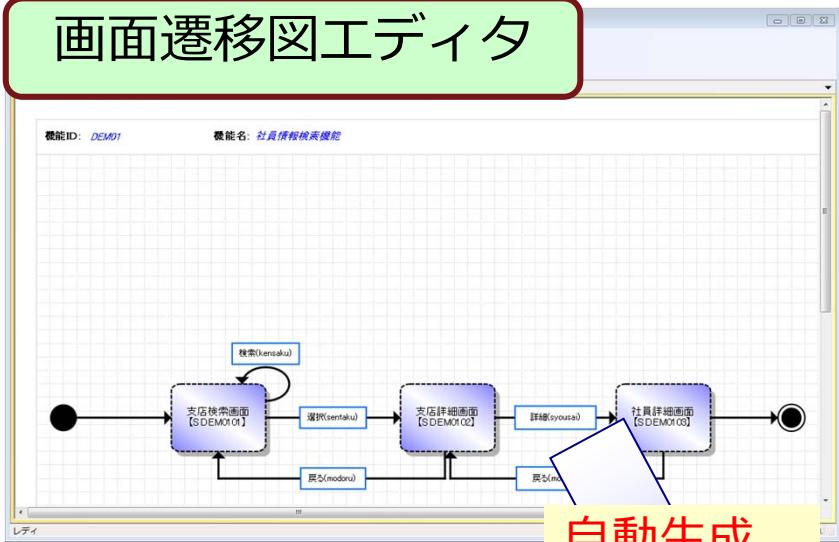
リアルタイム開発から実装コードを自動生成



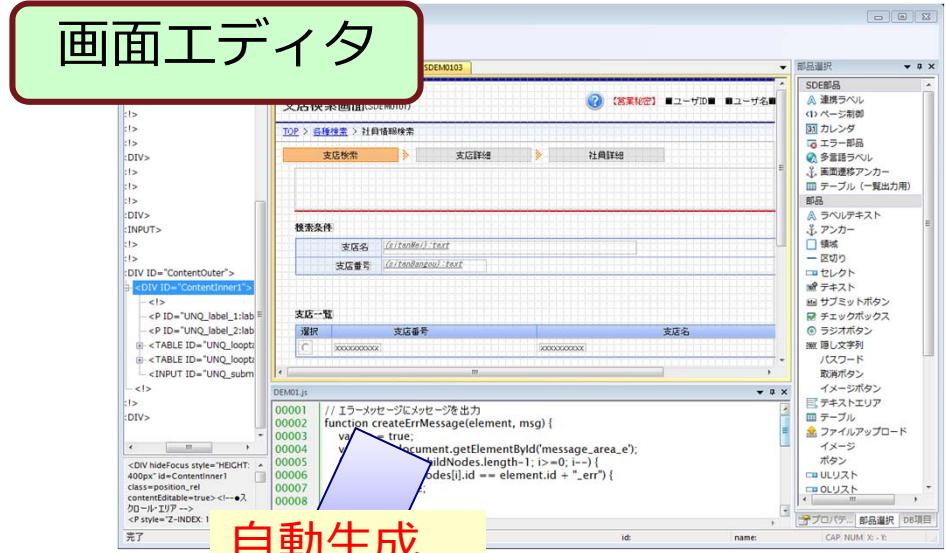
Step4. 設計書の自動生成

要件確定後、画面遷移図・レイアウト・項目定義など入力した内容を元に、機能設計書を自動生成

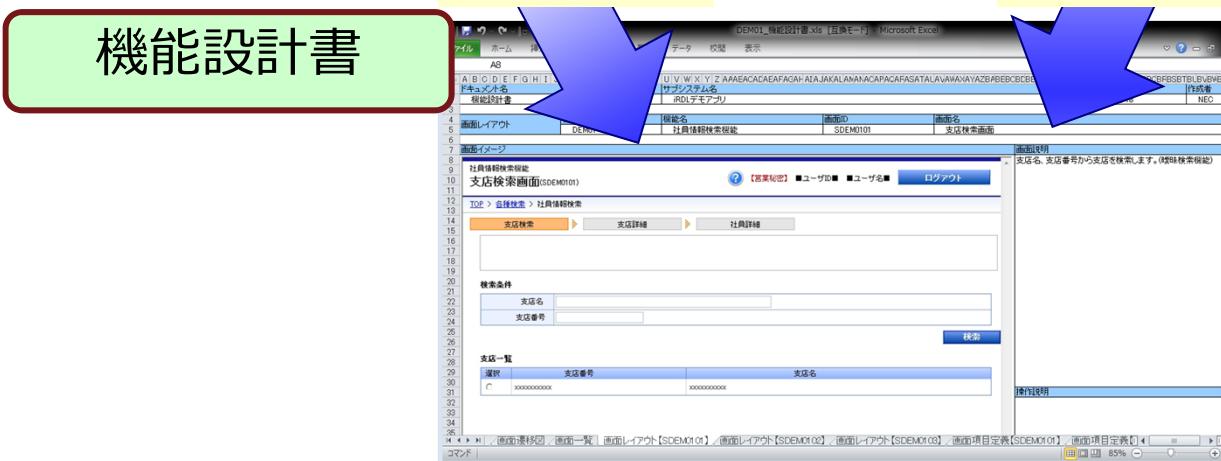
画面遷移図エディタ



画面工エディタ



機能設計書



Step5. AP実行～フォーマット、データ表示を確認

The screenshot shows a web-based application interface for store search. At the top, there's a browser header with the URL <http://localhost/RDD>, a search bar, and a tab labeled "支店検索画面 - iRD...". Below the header, the page title is "支店検索機能" and the sub-page title is "支店検索画面(SDEM0101)". There are buttons for "ログアウト" and "TOP > 各種検索 > 社員情報検索". A navigation bar below includes "支店検索", "支店詳細", and "社員詳細".

A green callout box with a red border and a red arrow points from the left side of the grid to the "支店番号" column header. The text inside the box says: "レイアウトを確認し、選択・支店番号の表示幅を調整".

The main content area displays a grid of store information:

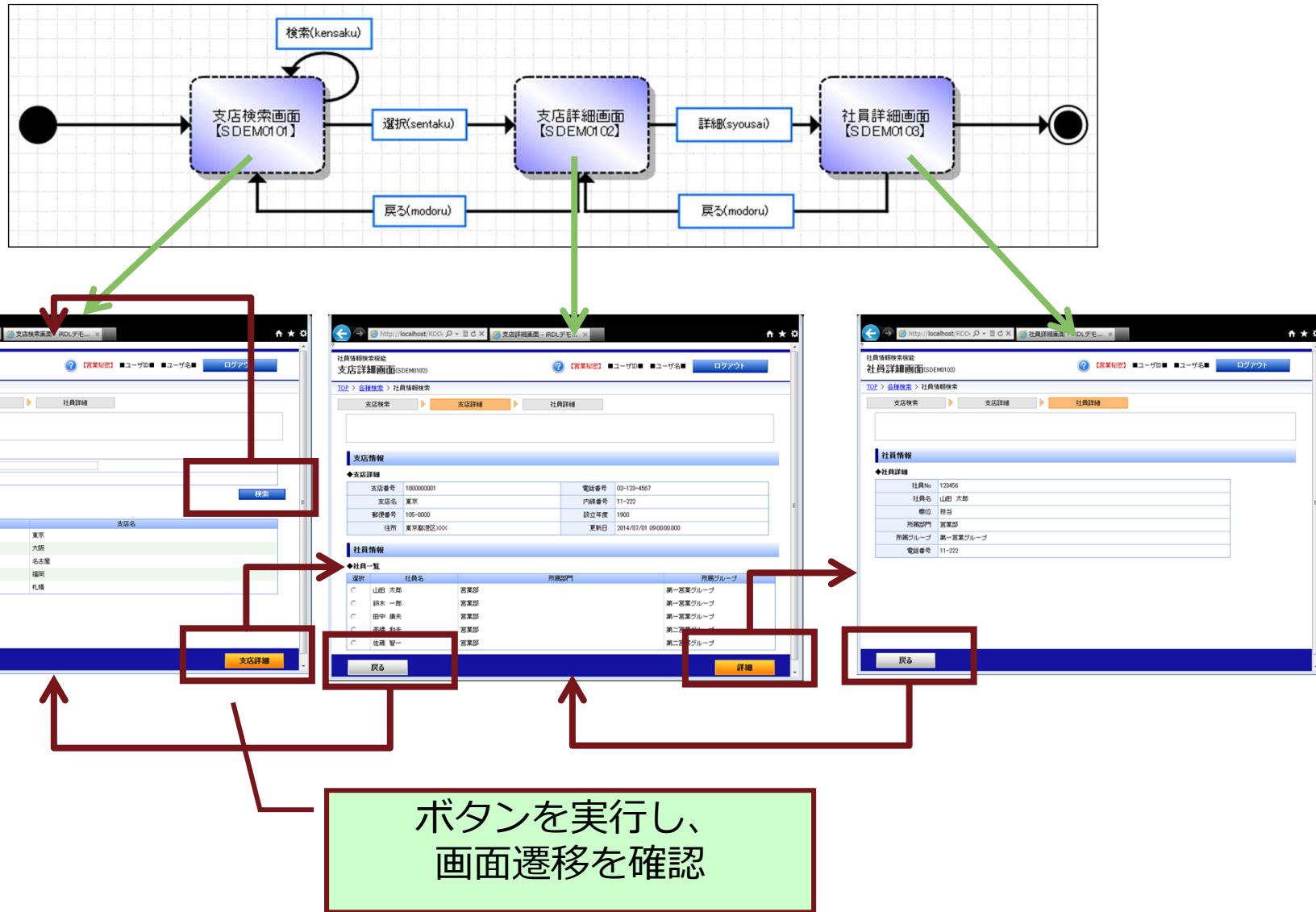
選択	支店番号	支店名
<input type="radio"/>	0000000001	東京
<input type="radio"/>	0000000002	大阪
<input type="radio"/>	0000000003	名古屋
<input type="radio"/>	0000000004	福岡
<input type="radio"/>	0000000005	札幌

A blue arrow points from the bottom right of the grid to a separate Excel spreadsheet window titled "DEM01APS...". The Excel window shows the same data in a table format:

	A	B
1	SITEN_CD	SITEN_NM
2	0000000001	東京
3	0000000002	大阪
4	0000000003	名古屋
5	0000000004	福岡
6	0000000005	札幌

A green callout box with a red border and a red arrow points from the bottom right of the grid to the Excel window. The text inside the box says: "モック用のExcelデータを画面に表示".

Step5. AP実行～画面遷移を確認



- SystemDirector Enterprise は日本電気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