

Orchestrating a brighter world



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

日本電気株式会社

本資料では SystemDirector Enterprise for .NETを、「for .NET」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  
補足.エディションの違い

## 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

# 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

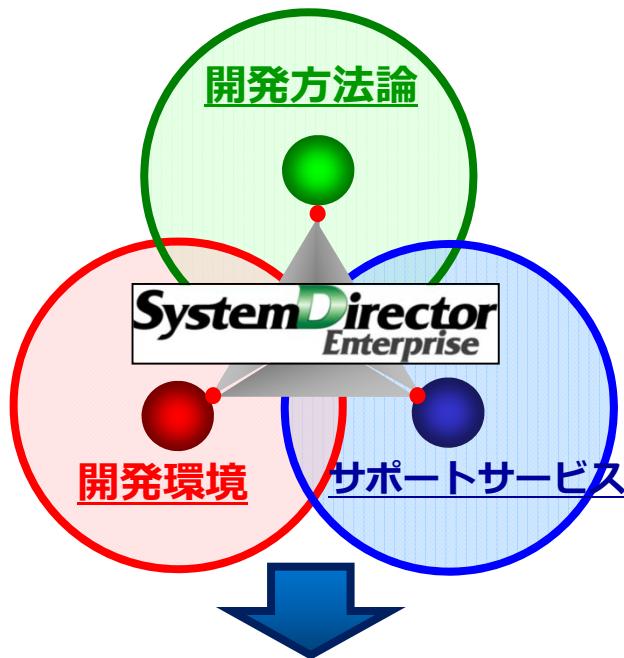
生産性の向上

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

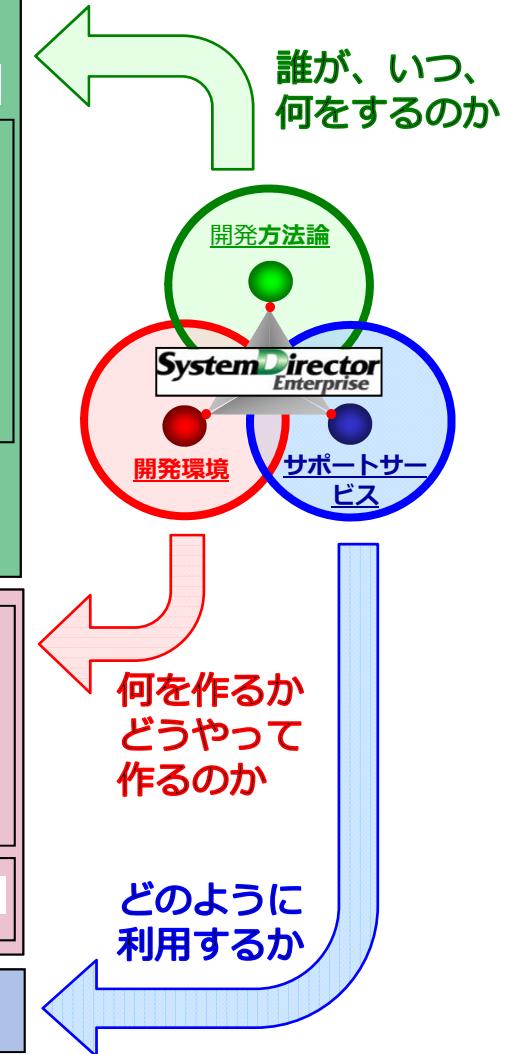
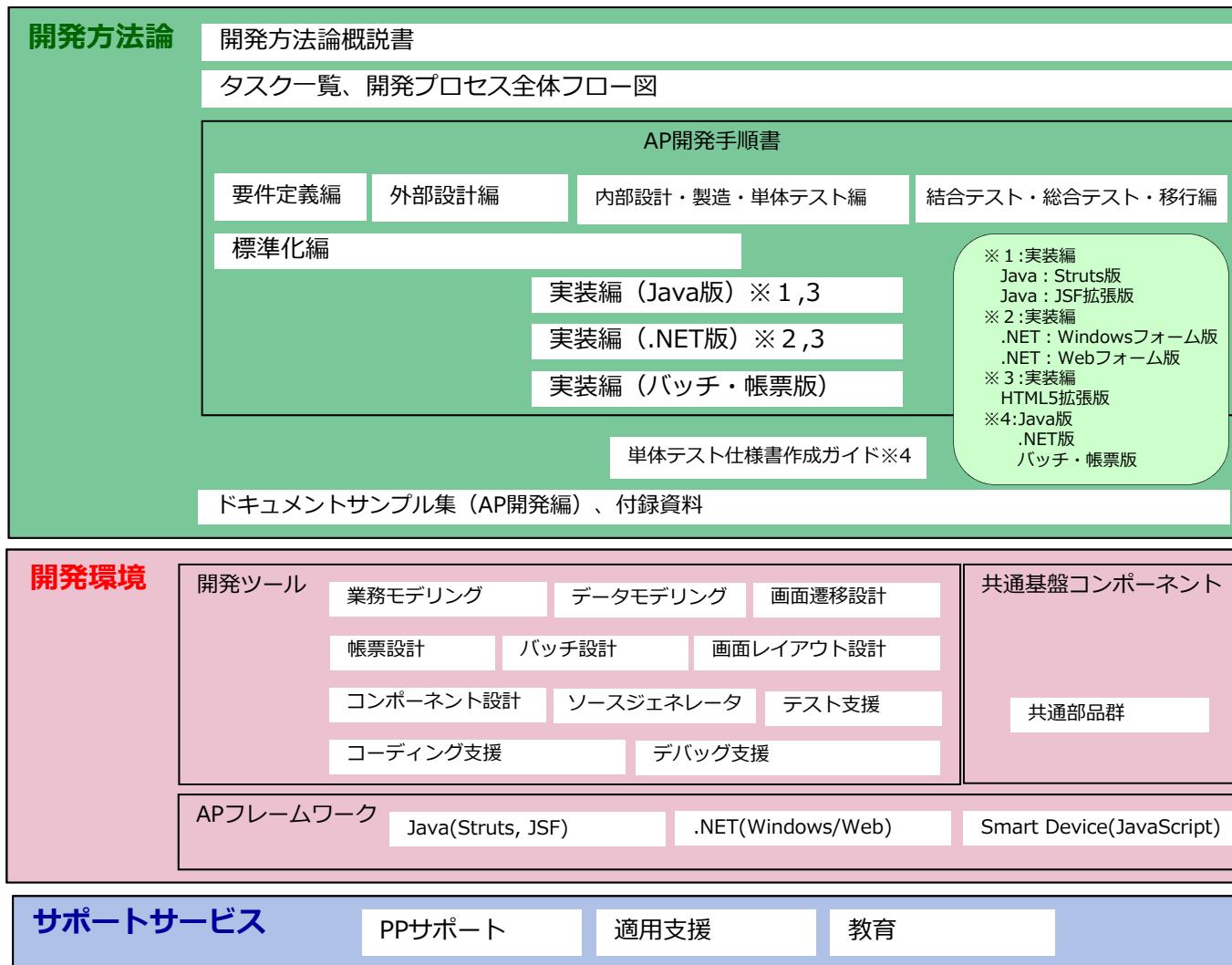
- ・開発方法論で標準作業の  
効率化
- ・開発環境で生産性向上

### 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

「オンラインシステム開発基盤」「バッチシステム開発基盤」の  
製品カテゴリーで、お客様のシステム開発を支援します。

## オンラインシステム開発基盤

SystemDirector Enterprise Suite Package

SystemDirector Enterprise

SystemDirector Enterprise for .NET

SystemDirector Enterprise for Smart Device

本稿では下記製品について  
紹介します。

## バッチシステム開発基盤

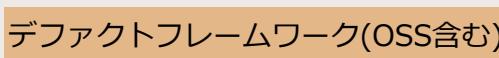
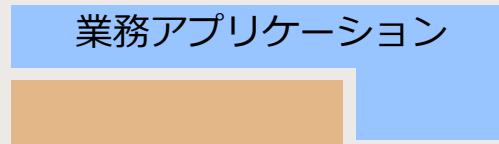
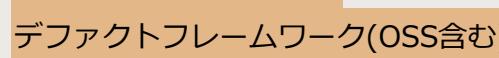
SystemDirector Enterprise for Java Batch

SystemDirector Enterprise for C Batch

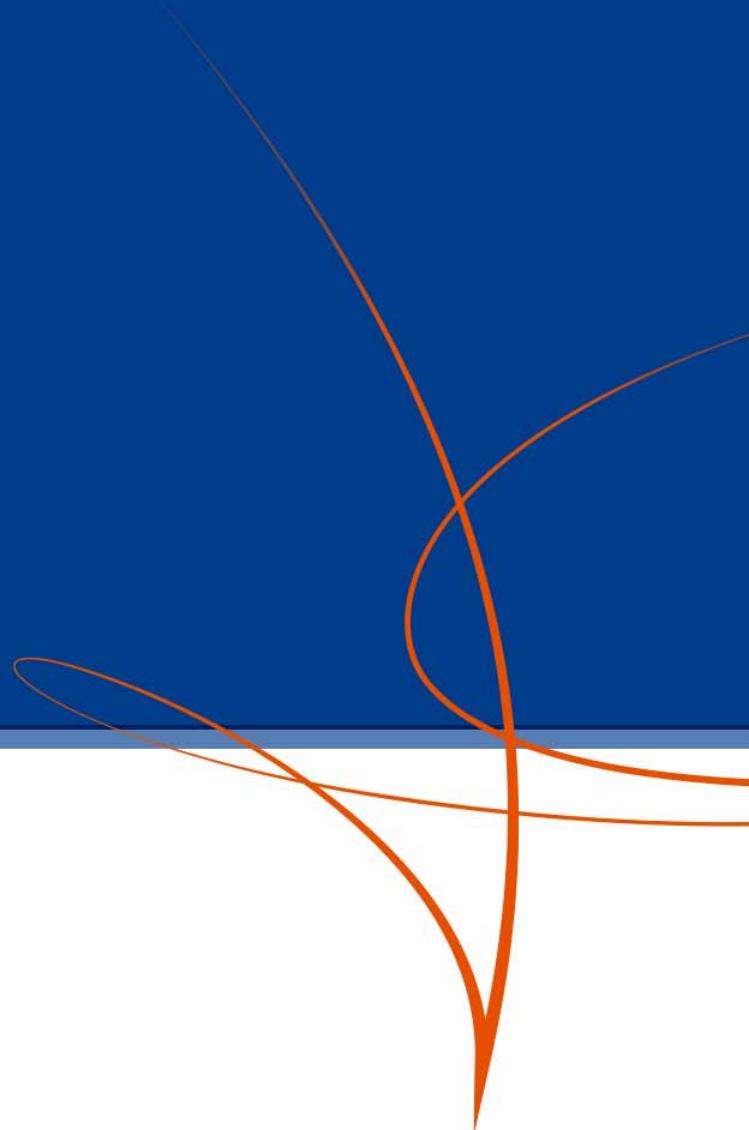
SystemDirector Enterprise for Report

# 補足.エディションの違い

SystemDirector Enterpriseでは、生産性重視のProfessional Editionと適用柔軟性の高いExpress Editionの2つを提供しており、要望に合わせて選択可能です。

	Professional Edition	Express Edition
製品コンセプト	生産性重視の開発環境 生産性を高める独自のフレームワーク提供	適用しやすい開発環境 導入コストを下げるためOSS等標準技術のみを利用
独自フレームワーク	独自拡張機能を有したフレームワークを提供   	独自実装を排除し、ピュアなソースコードで開発   
メリット	<ul style="list-style-type: none"><li>独自ツール、フレームワークにより作業効率化を実現</li><li>フレームワークによる標準化で工数を削減</li></ul>	<ul style="list-style-type: none"><li>Eclipse、Visual Studioのみの開発より高い生産性</li><li>部分的な切り離しが容易</li></ul>
デメリット	<ul style="list-style-type: none"><li>独自フレームワークによる一気通貫な機能提供となるため、部分的に切り離しにくい</li></ul>	<ul style="list-style-type: none"><li>開発の自由度が大きいため、標準化工数がかかる</li><li>実行環境に独自性がないため、サポート提供がない</li></ul>

## 第2章 開発方法論

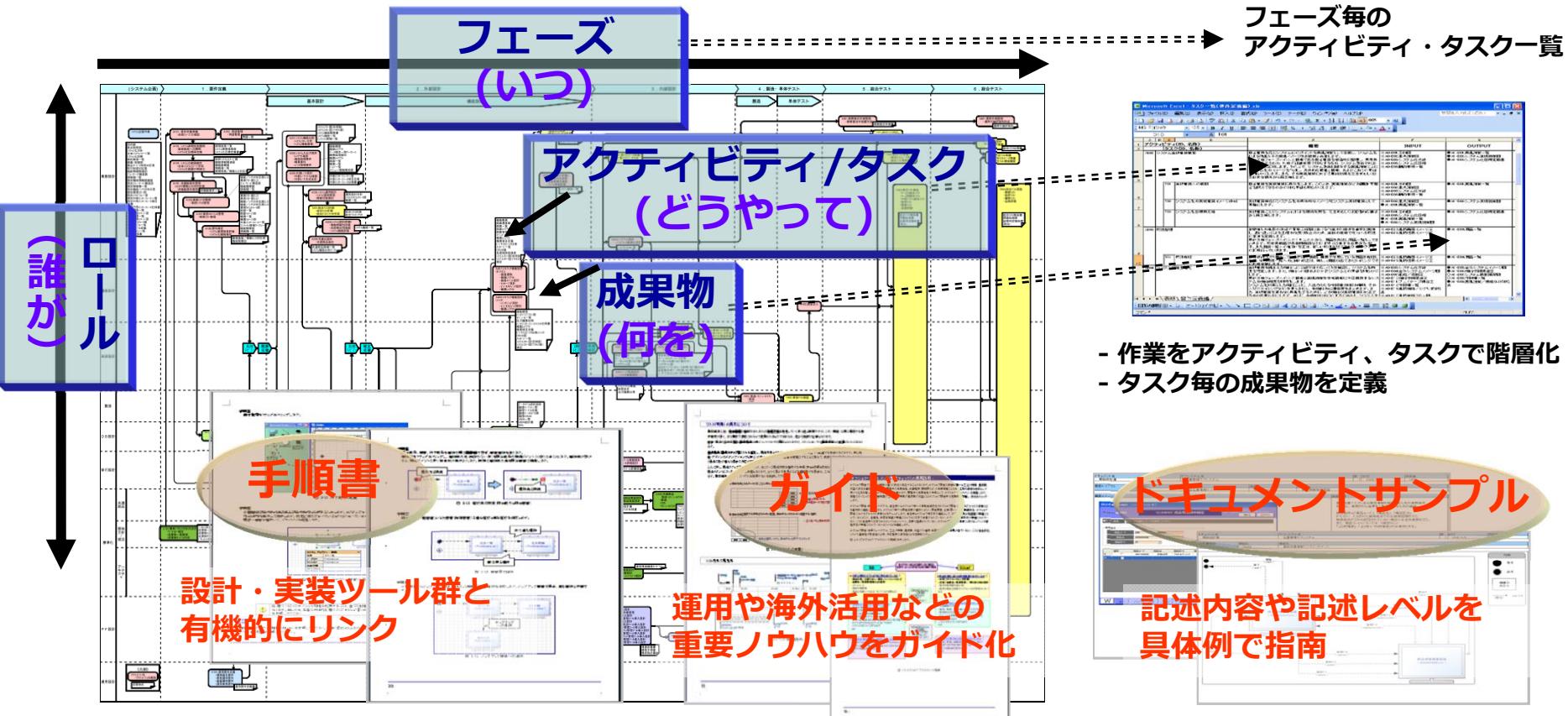


## 2.1.概要

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



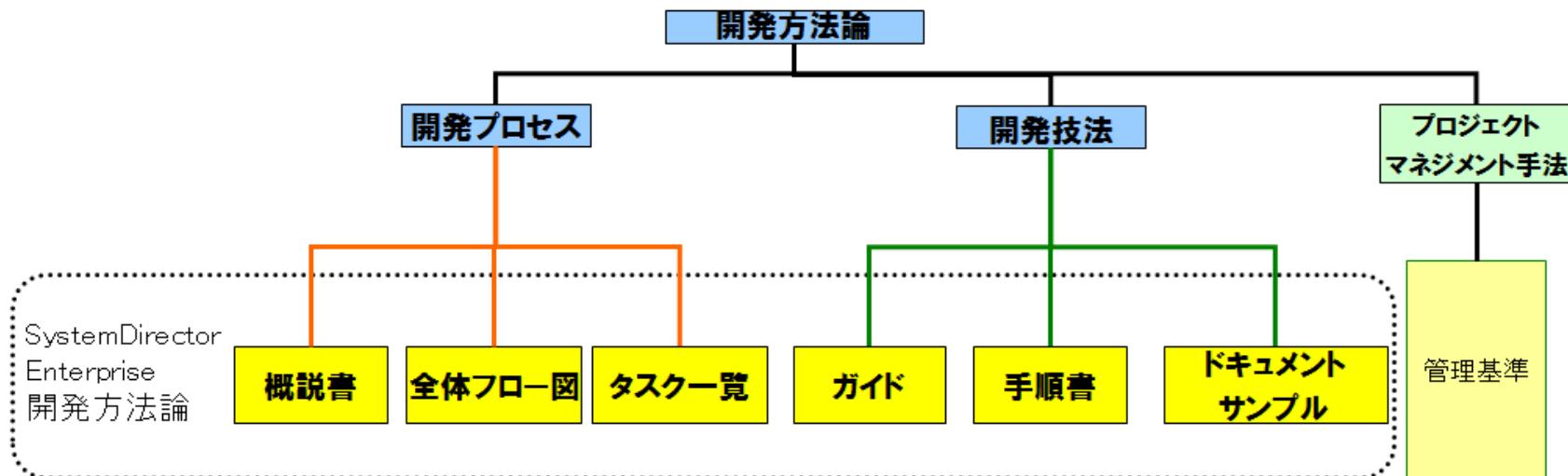
## 2.2.一般的な開発方法論との違い

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。

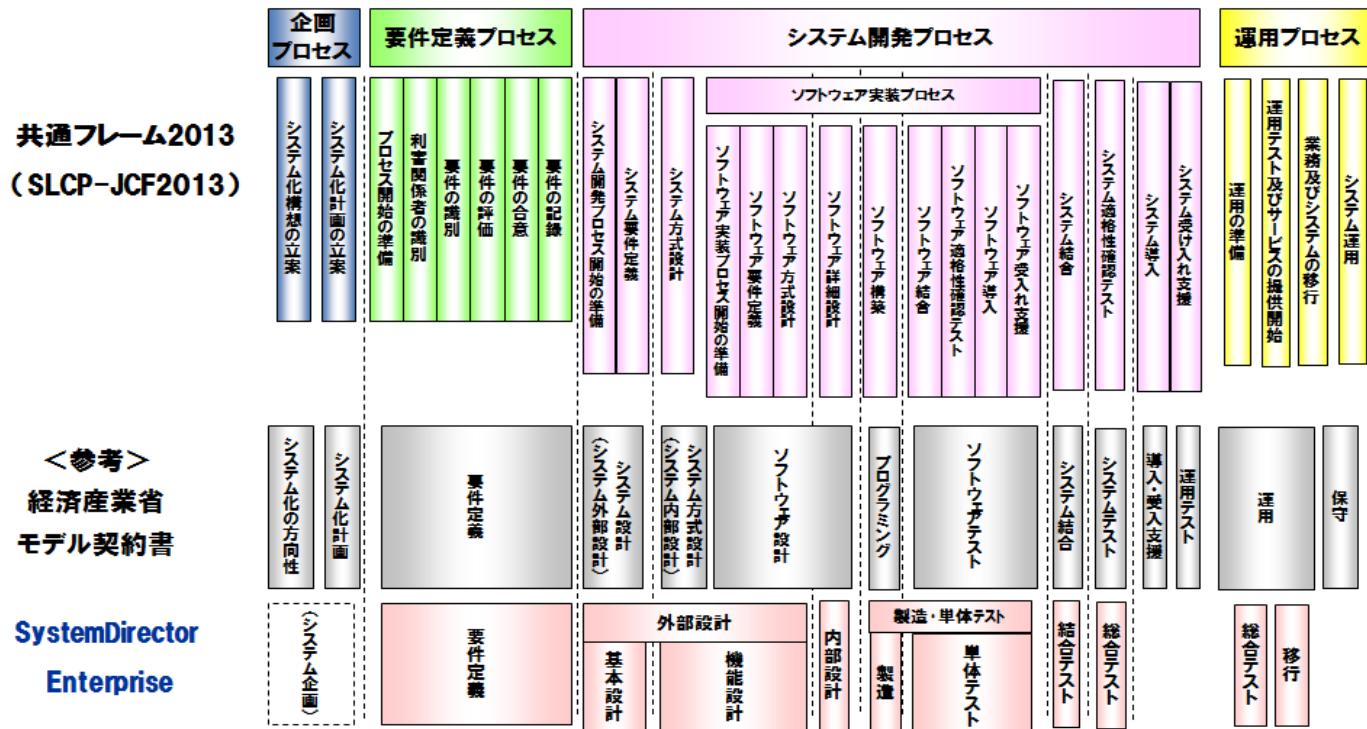


図1 プロセスレベルの対応関係

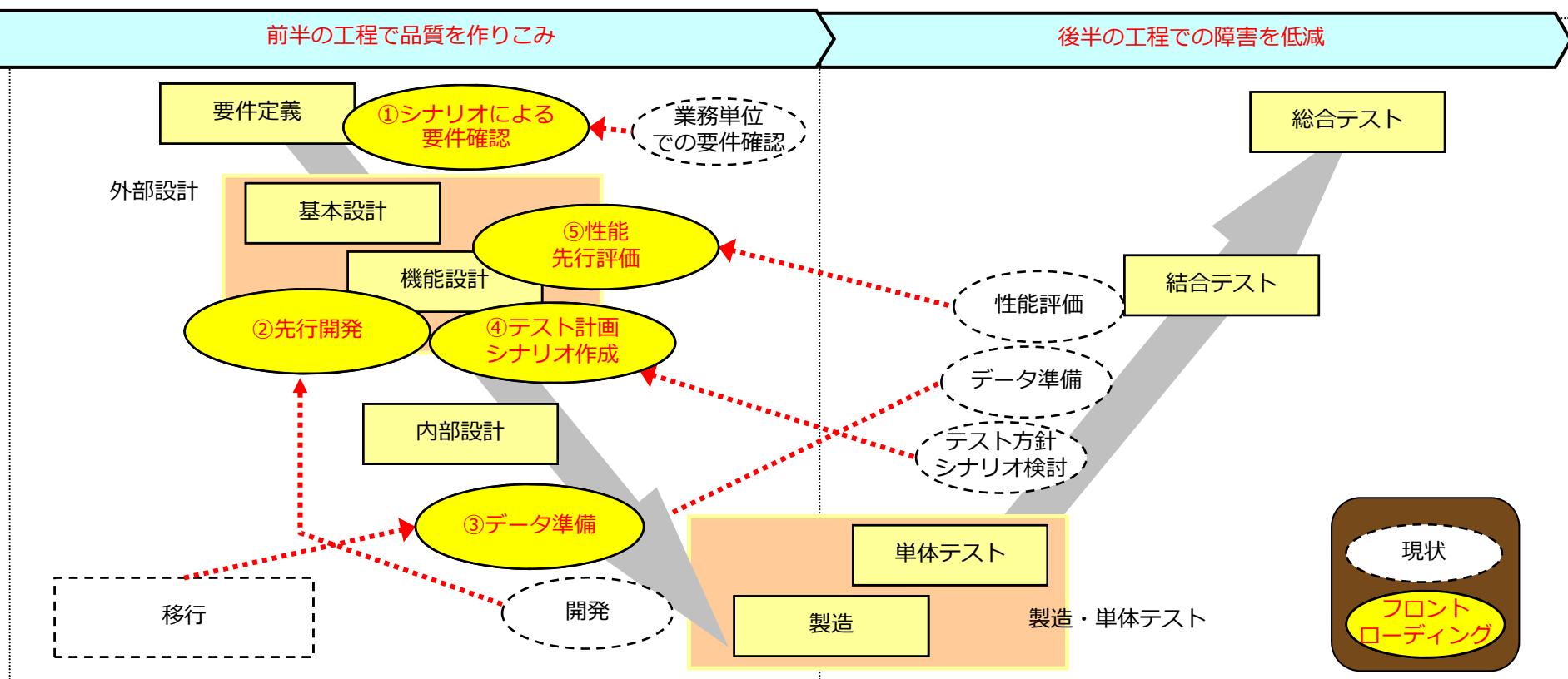
※ 共通フレーム2013との関係はウォータフォールモデルをベースとし、フェーズ、作業順序を考慮して整理しています。

## 2.4. フロントローディング

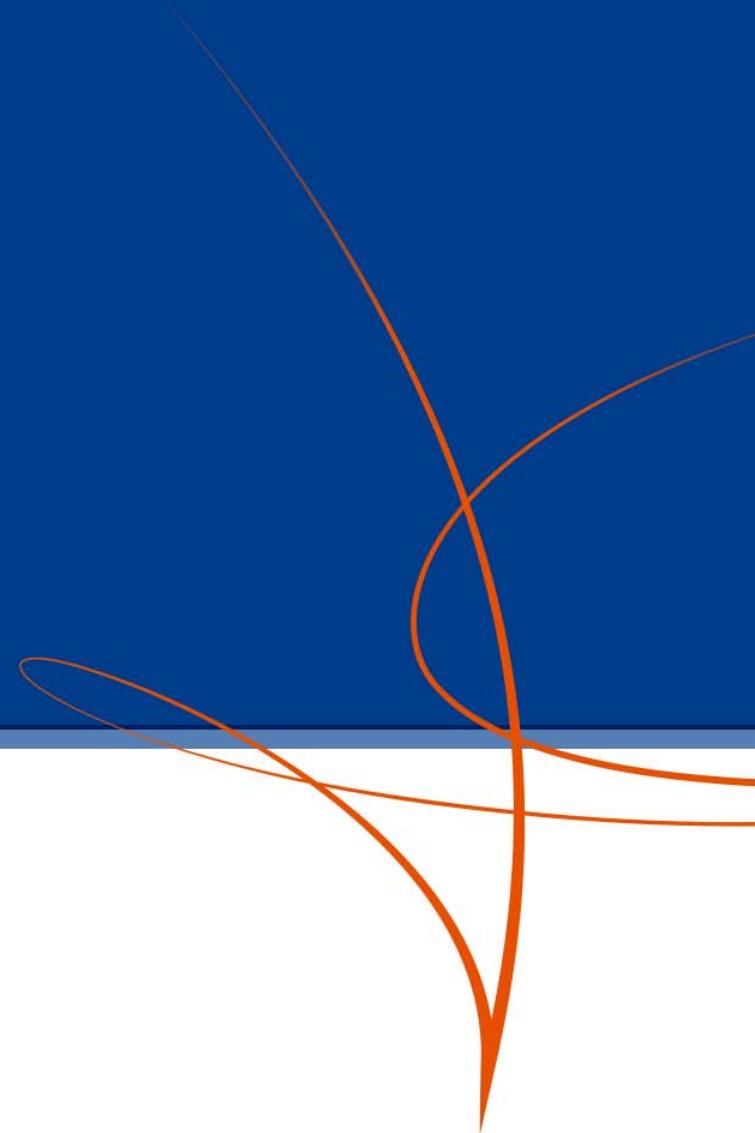
開発の前半で品質を作りこむ手法を取り入れています。

「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

Windowsフォーム、ASP.NET Webフォーム/ASP.NET MVCを用いた業務システムの構築を総合的にサポートいたします。

Point  
1

#### アーキテクチャモデル

- Microsoft社ガイドライン「Application Architecture Guide 2.0」に準拠した推奨アーキテクチャを整備。
- スムーズな最新技術対応や、アーキテクチャ選定工数を削減。

Point  
2

#### フレームワーク

- Windowsフォーム、ASP.NET Webフォーム/ASP.NET MVCを用いた業務システムの構築において、不可欠となる共通機能を提供。

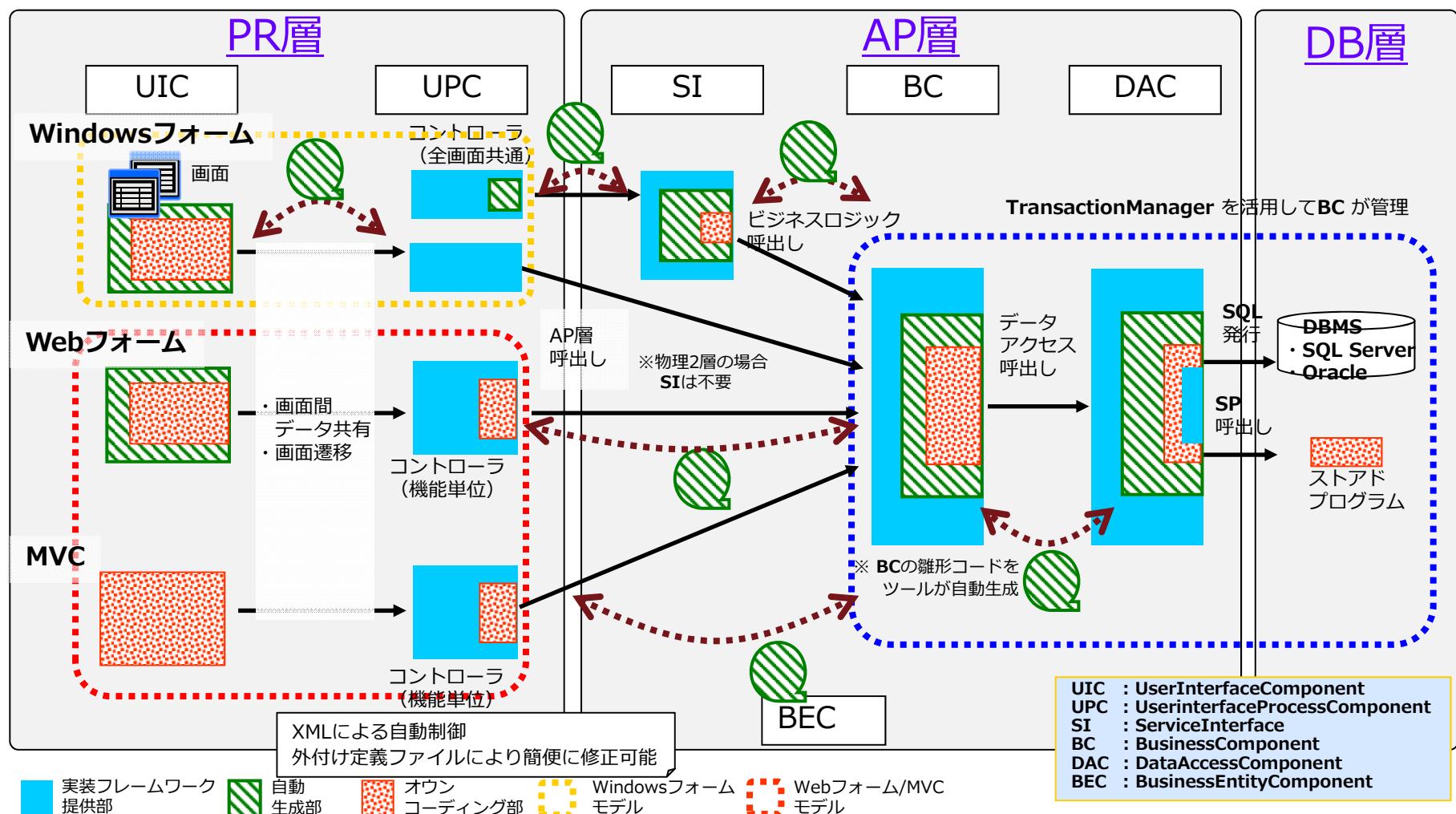
Point  
3

#### 開発ツール

- Visual Studio + .NET Frameworkによる一般的な開発スタイルをそのままに開発を効率化するテンプレートやジェネレータを整備。

### 3.2.アーキテクチャモデル

Microsoft社ガイドライン「Application Architecture Guide 2.0」に準拠  
マルチレイヤ構成を採用。均一な品質のアプリケーション開発が可能。



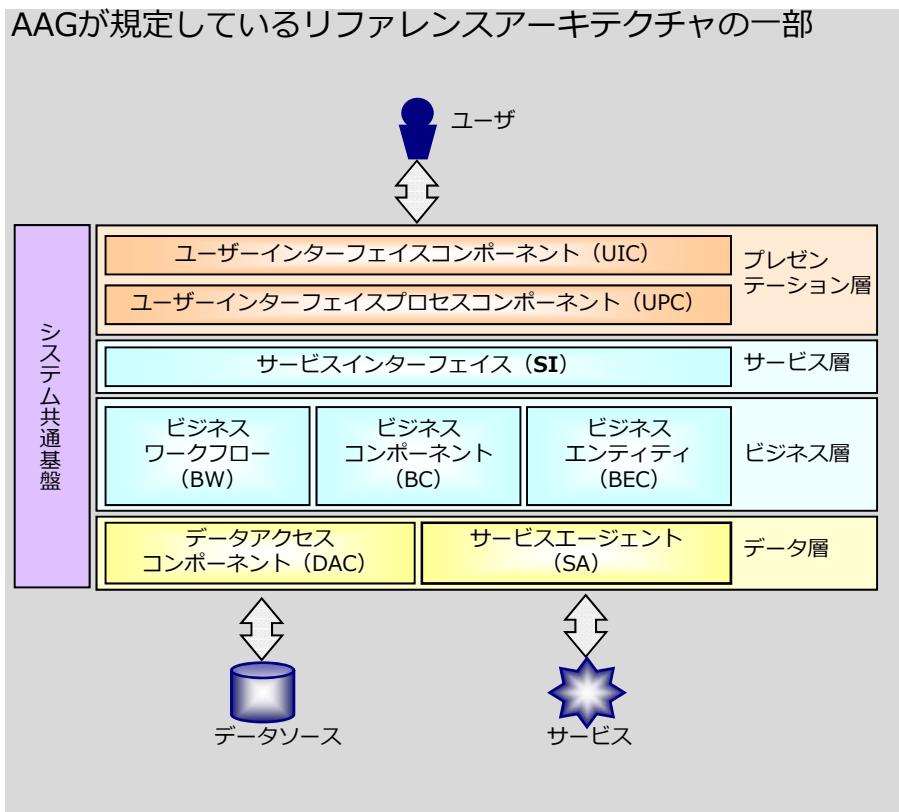
# 補足. 「AAG(Application Architecture Guide)」とは

Microsoft社が提供する.NETアプリケーションのアーキテクチャ策定と設計のためのガイダンス

アプリケーションの論理階層の分け方、各階層で実現すべき機能を解説

複数のハードウェアシステムに分散させやすいアプリケーションを構築可能

AAGが規定しているリファレンスアーキテクチャの一部



	コンポーネント	主な役割と位置付け
PR層	UIC	ユーザとの対話、データの表示とそれに関連するロジック、入力受付、データ入力チェック、フォーマット機能、Undo、クリップボードサポート、表示キャッシュ、ページング制御など
	UPC	画面遷移制御、入力仕振りデータの保持、一時保存機能など
AP層	SI	BCに対する通信ラッパ機能、データ形式変換、通信制御、認証など
	BW	ビジネス・プロセスの定義
データ層	BC	ビジネスロジックやトランザクション制御
	BEC	レイヤ間で受け渡すビジネスデータ
DB層	DAC	データソースに対するアクセス制御
	SA	他システムとの連携
システム共通基盤		セキュリティ、運用管理、通信などのシステム共通の機能

### 3.2.1. クラス構造の統一化

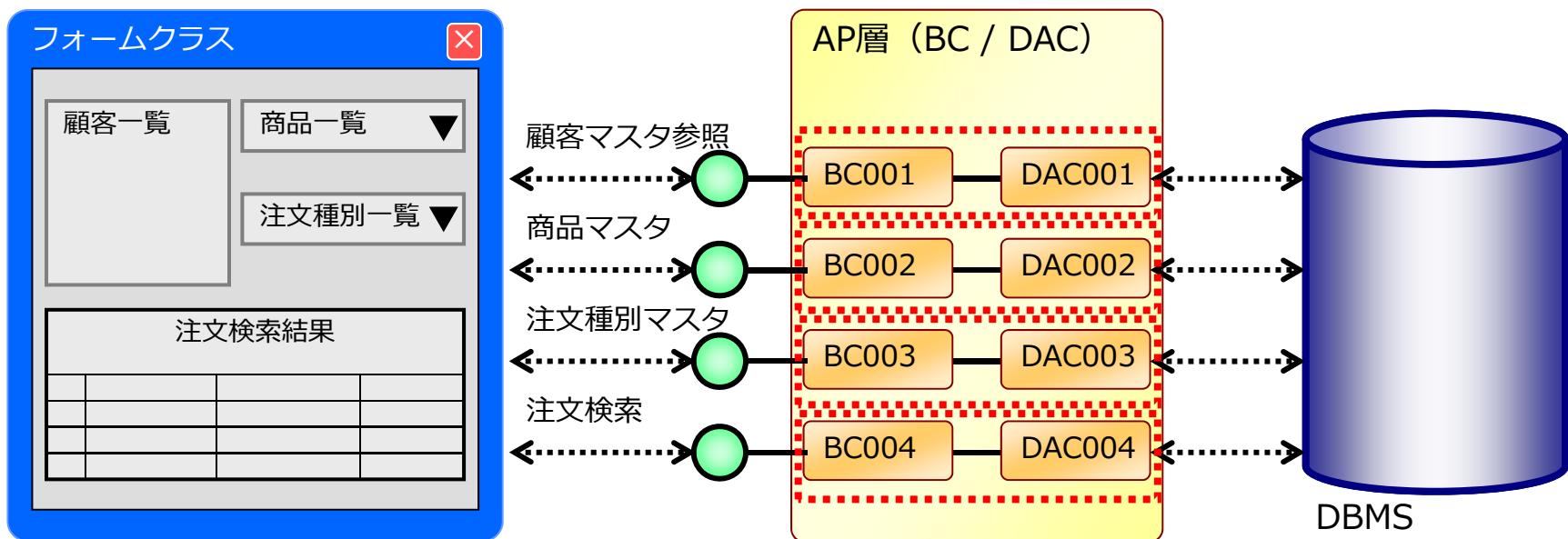
BCクラスの構造をあらかじめ規定し、成果物の品質を均一化

BCクラスが持つメソッドは、Executeの1つのみ

BC のメソッドは、1回の呼び出しで処理が完結するサービスとして設計

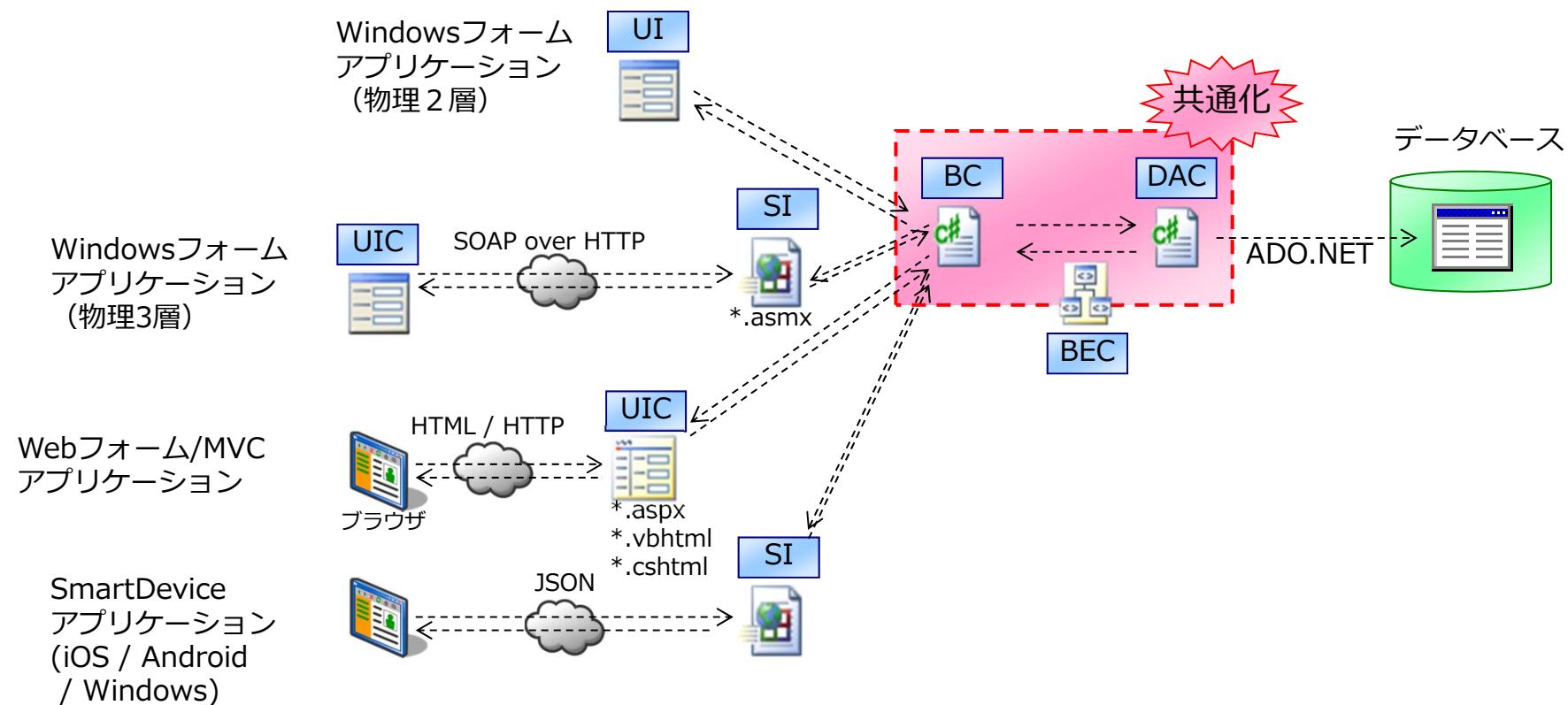
トランザクション制御コードの記述は不要（BusinessComponentBase利用時）

クラス設計とトランザクション設計が直結し、レビュー・保守効率が向上



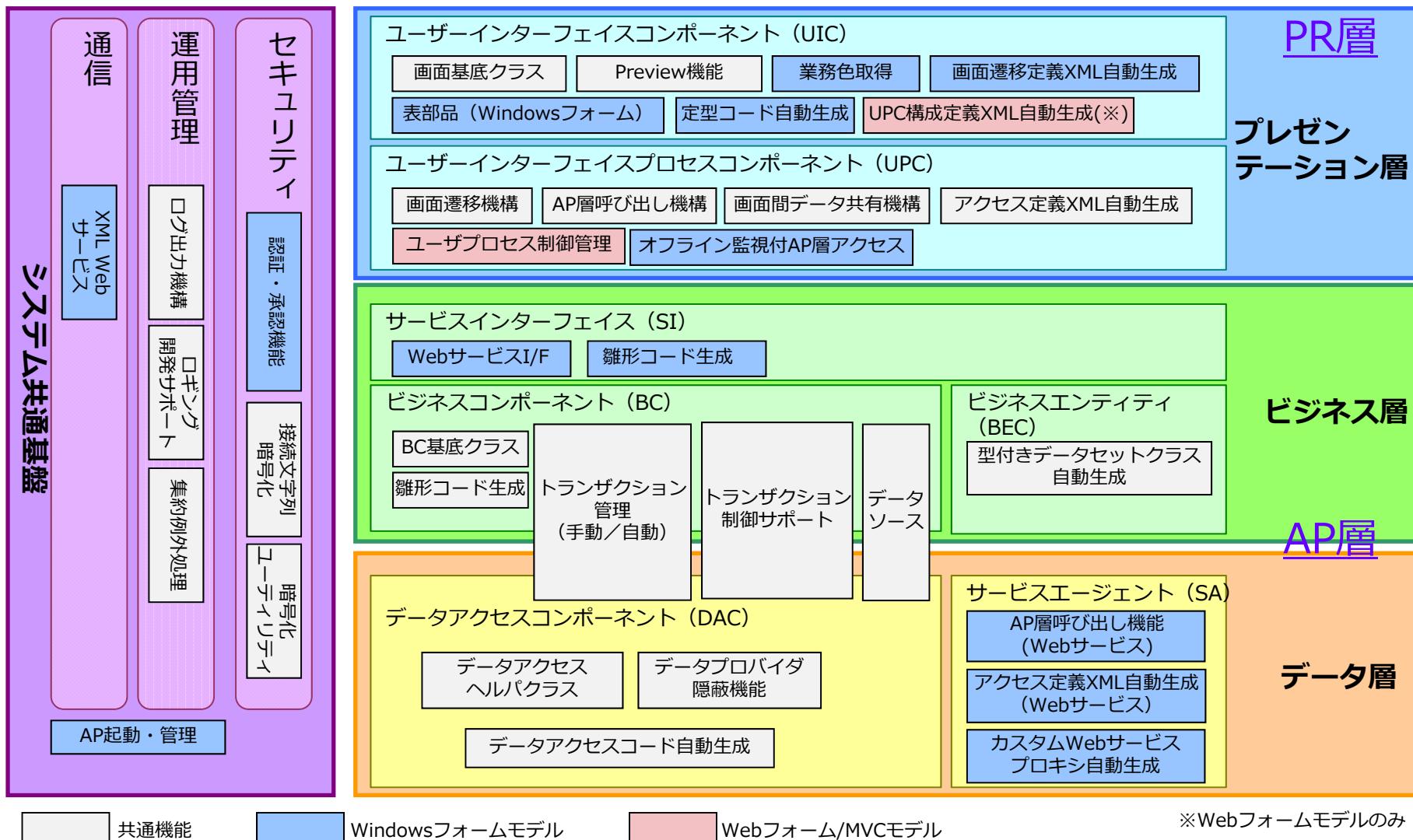
### 3.2.2. さまざまなクライアント形態に対応

- AP層はクライアント形態にかかわらず共通のアーキテクチャ
- AP層のコンポーネントは、独立性が高く、様々な形態のクライアントに対応
- 設定情報の変更により、クライアントアーキテクチャをシームレスに移行可能



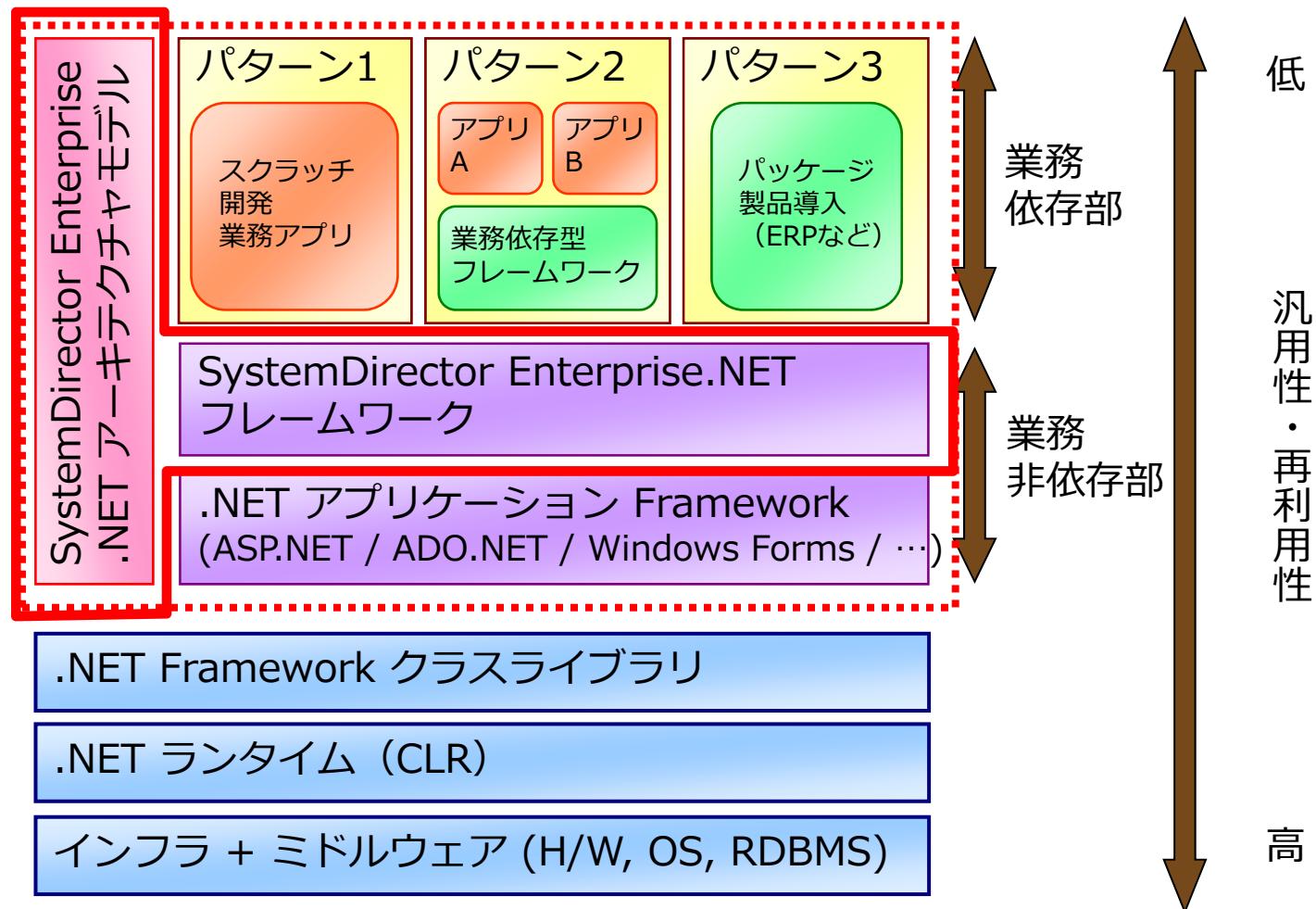
### 3.3.フレームワーク

アーキテクチャモデルに合わせ、実装効率を高めるための機能を提供



### 3.3.1.NET Frameworkとの関係

.NET Frameworkをベースとし、NEC独自の強化を行った  
業務非依存の「アーキテクチャモデル」と「フレームワーク」を提供

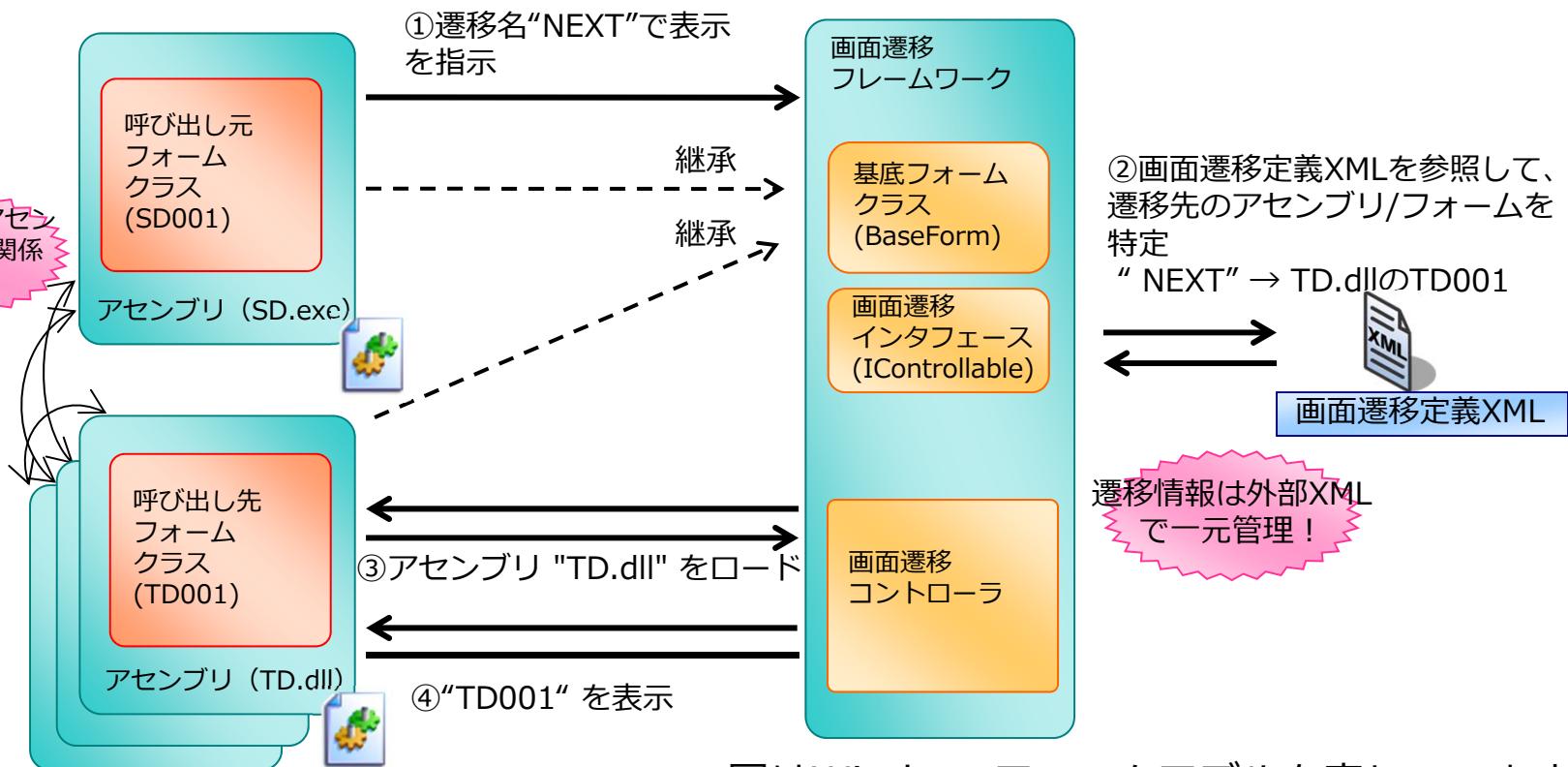


### 3.3.2.画面遷移機構

各モデルの特性に合わせた画面遷移オペレーションの管理機能を提供

遷移情報は、外部のXMLファイルで一元管理し、高い保守性を実現

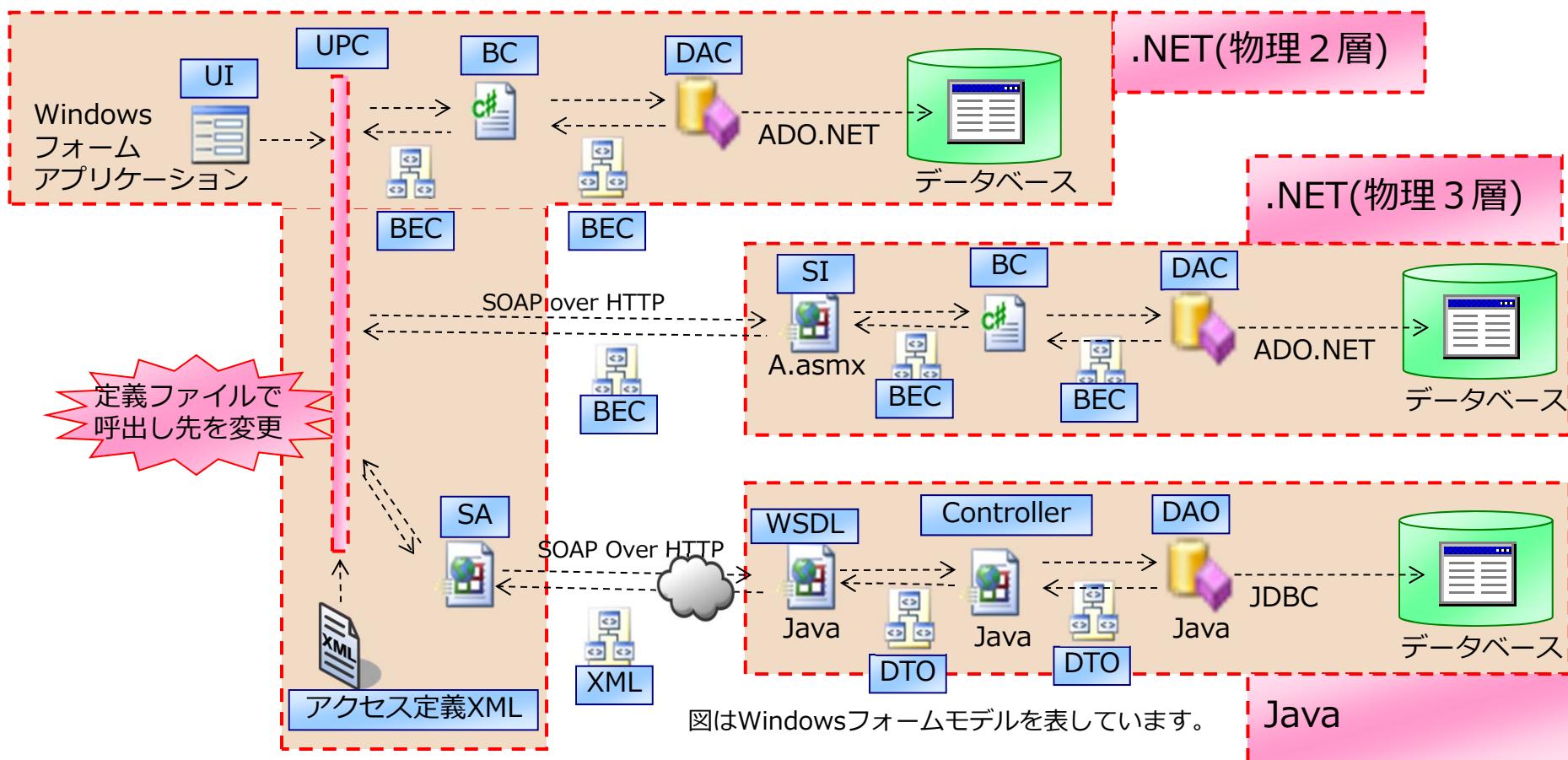
画面間の依存関係が切れるため、呼び出し先フォームを含むDLLを参照設定する必要がなく、分離並行開発が可能



図はWindowsフォームモデルを表しています。

### 3.3.3. A P層呼び出し機構

- PR層／AP層間を疎結合にするとともに通信手段を隠蔽  
PR層はAP層の配置先を意識することなく単一のAPIで呼び出すことが可能  
外部ファイル化したアクセス定義XMLを修正するだけで呼び出し先を変更可能



### 3.3.4. トランザクション管理機能

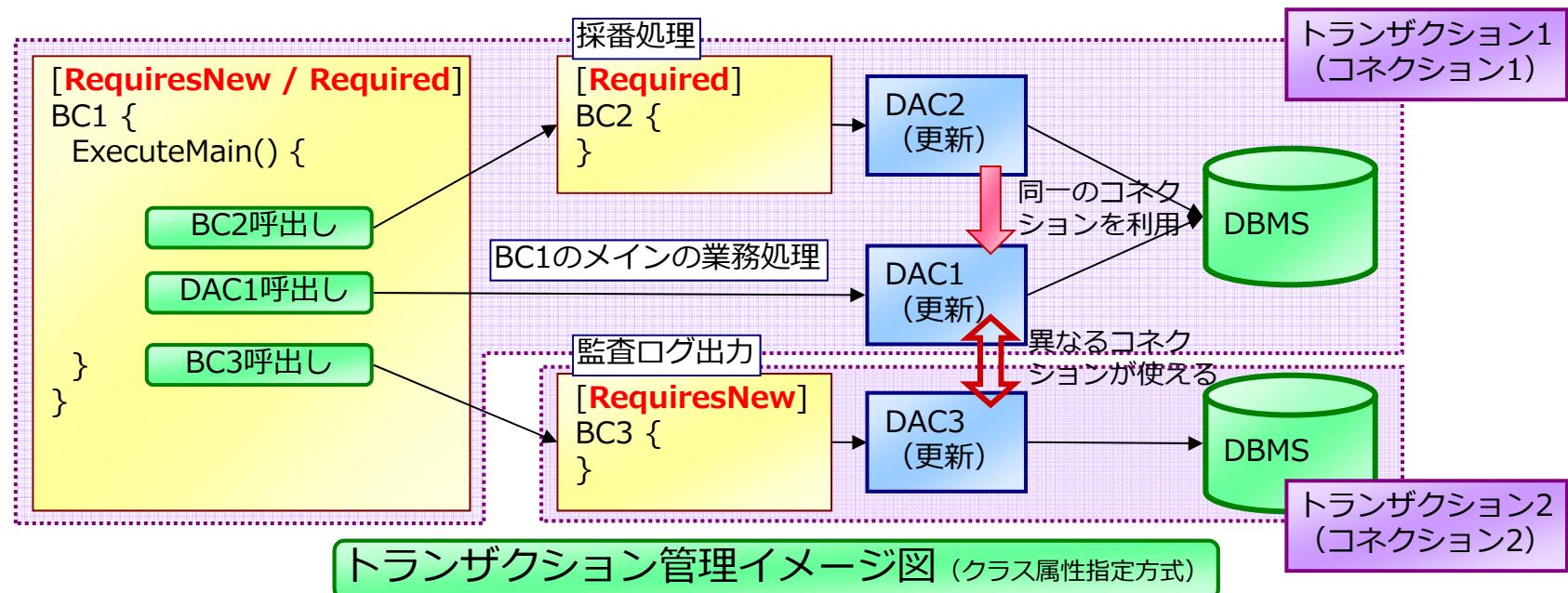
トランザクション管理に係わる共通処理をフレームワークで提供

トランザクションを管理するための共通機能を提供します

データベース接続・コミット・ロールバック・データベース切断

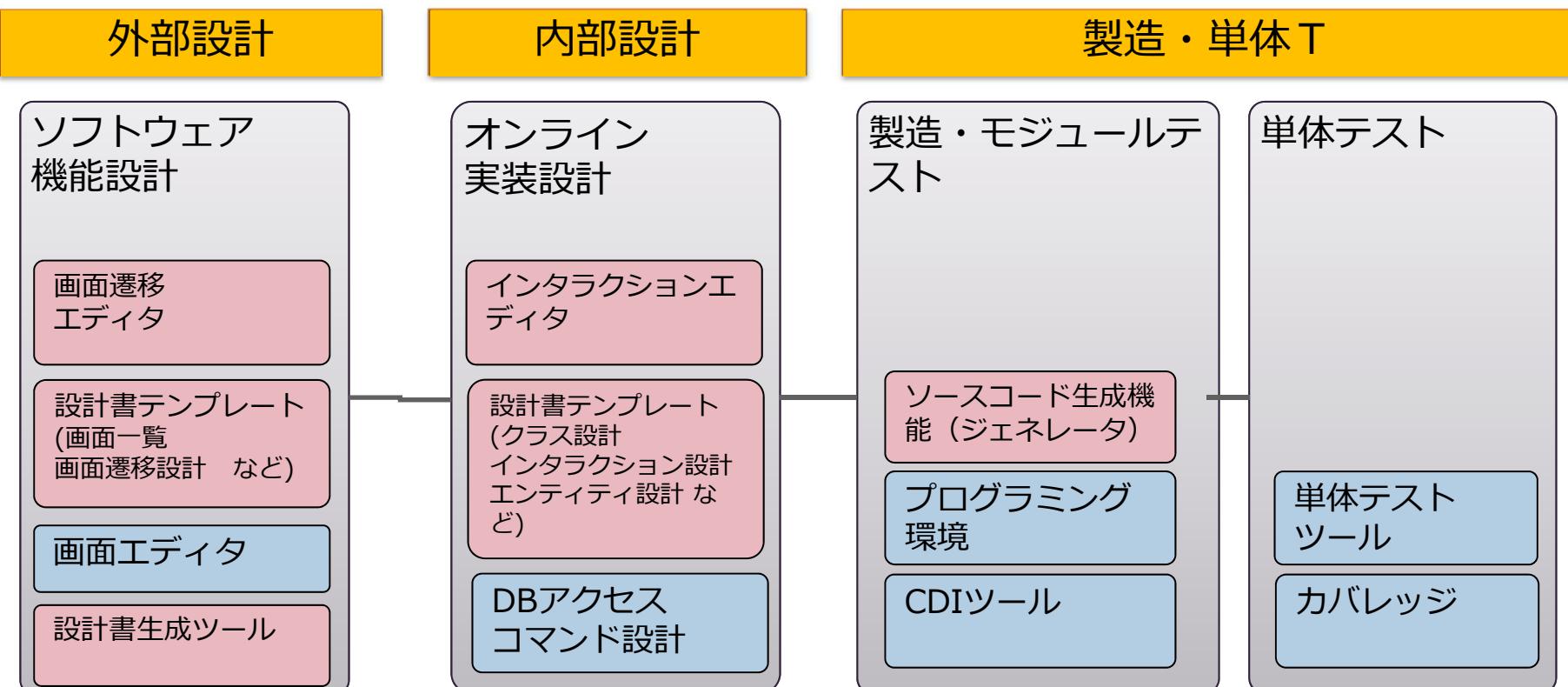
データソース管理機構やデータアクセスヘルパークラスとシームレスに連携

コネクションのクローズ漏れなどを防ぎ、実装にかかるコストを低減



### 3.4.開発ツール

Visual Studioをベースに、開発プロセスの各工程を効率化するテンプレートやジェネレータを整備



凡例： SystemDirector Enterpriseで提供

Visual Studioを利用

### 3.4.1.開発ツールの特徴（1／2）

Visual Studio + .NET Frameworkによる一般的な開発スタイルをそのままに、開発効率化するための設計書テンプレートとジェネレータ群を提供

Excelベースの設計書テンプレートを提供。

設計情報をインプットに、推奨アーキテクチャとなるソースを生成

ソース生成後はVisual Studio標準の開発方式でコーディング。

Visual Studio標準のCDIツールやテストツールも制約なく利用可能

設計書テンプレート  
をベースに設計  
整合性チェック機能

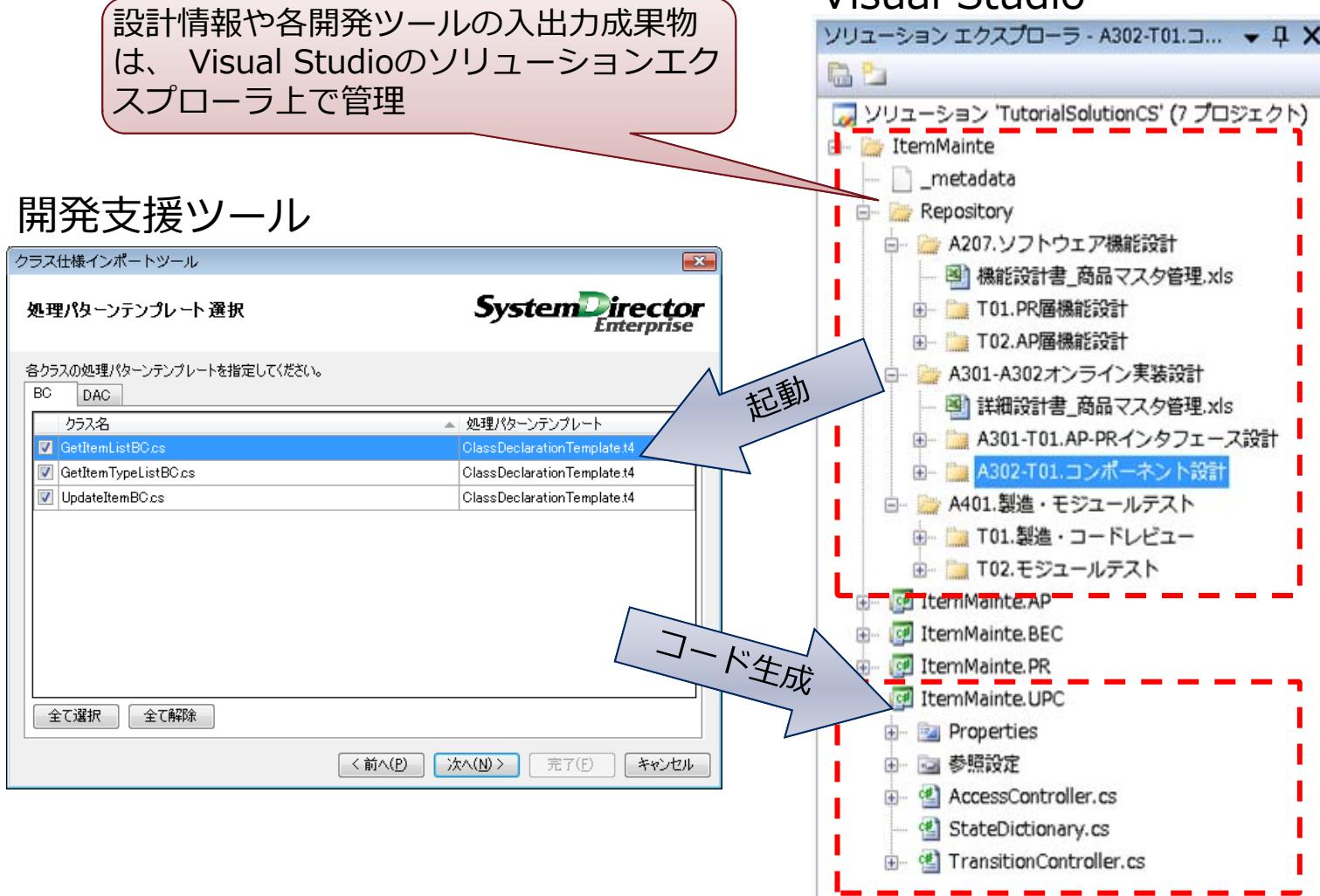
ジェネレータによる  
ソース生成

オウンコーディング  
Visual Studio標準の  
CDIツール、テスト  
ツールを活用

ガイドによる開発支援

### 3.4.1.開発支援ツールの特徴（2／2）

Visual Studio上で、各工程の成果物の管理を実現



### 3.4.3.開発ツール一覧

開発工程は「SystemDirector Enterprise開発方法論」に準拠。おもに、機能設計～製造・単体テストまでを開発ツールでサポート

ツール／テンプレート	概要
設計書テンプレート	画面遷移設計、クラス設計、インターラクション設計 他
	画面遷移エディタ
	インターラクションエディタ
ジェネレータ	画面一覧インポート
	画面遷移一覧インポート
	クラス仕様インポート
	インターラクション仕様インポート
	データセット仕様インポート
	ViewModel仕様インポート
	データコントラクト仕様インポート
	DACビルダ
	デザインリフレクタ
その他	キーマネージャ
	データソースXML生成ツール

### 3.4.4.画面遷移エディタ

Win

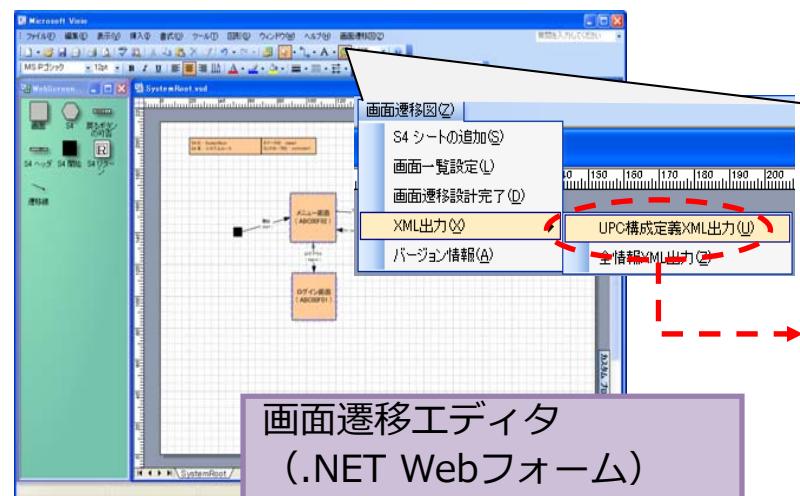
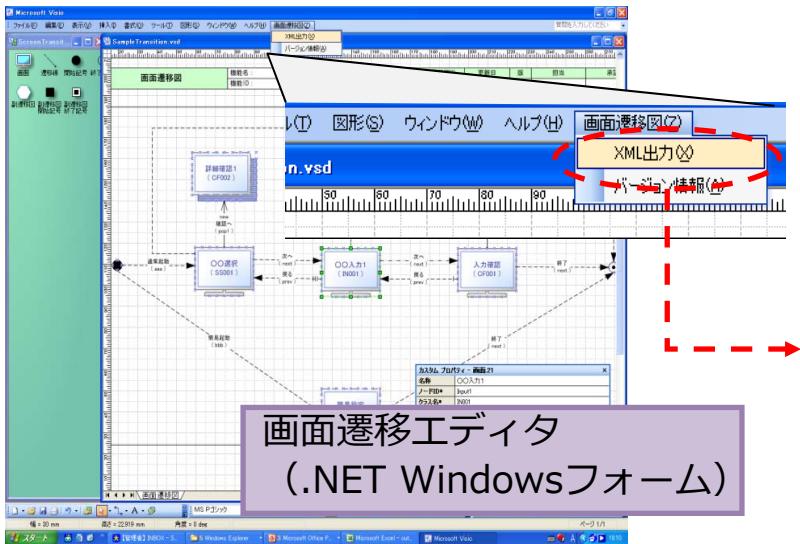
Web

外部設計

内部設計

製造UT

画面遷移の妥当性を設計時点で検証。遷移情報はXMLファイルとして出力



XML

遷移情報

XML

- 不正な設計情報を整合チェックにより検出することで、設計時の品質向上が図れます。
- 設計情報をxmlファイルに出力し、後工程に引き継ぐことが可能です。
- Excel帳票で遷移を一覧情報として設計し、遷移図自体は任意のツールで作成することも可能です。

画面一覧インポート  
画面遷移一覧インポート

開発支援ツール

画面遷移機構

SystemDirector Enterprise  
.NETフレームワーク

※ MVCモデルは未対応

### 3.4.5.画面一覧インポートツール

Win

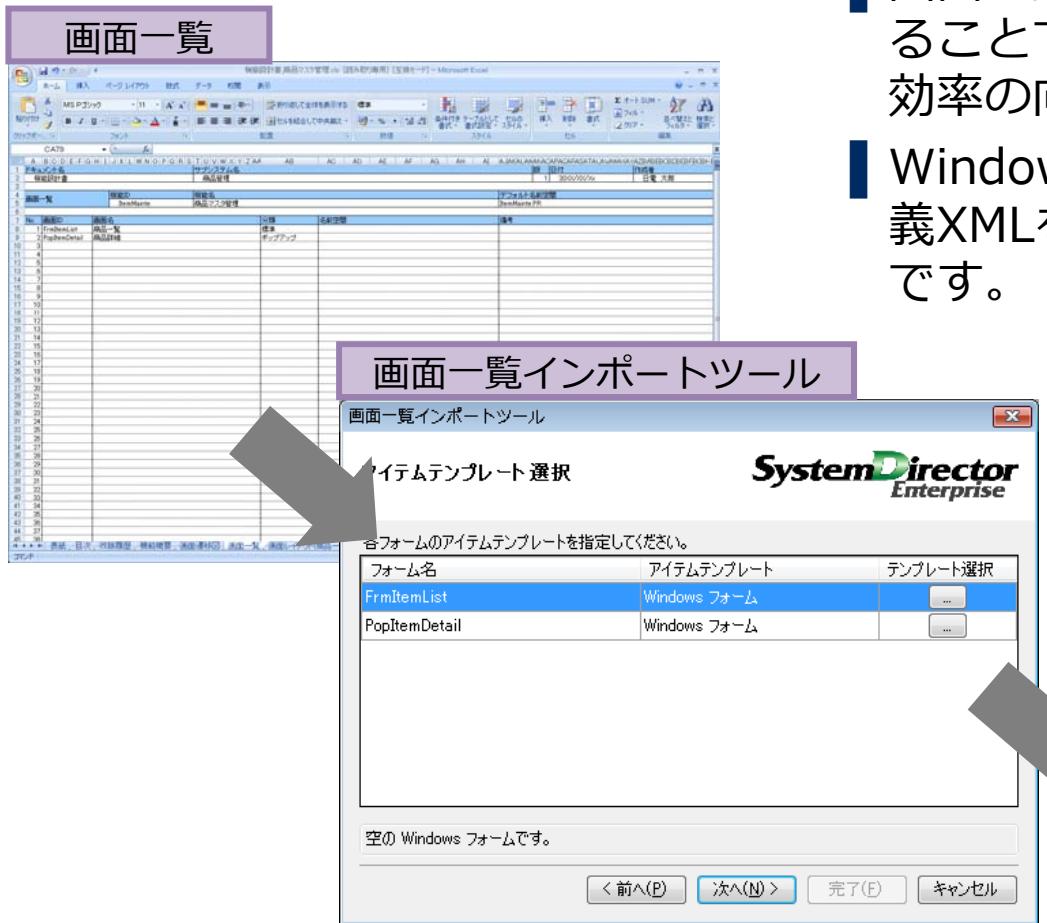
Web

外部設計

内部設計

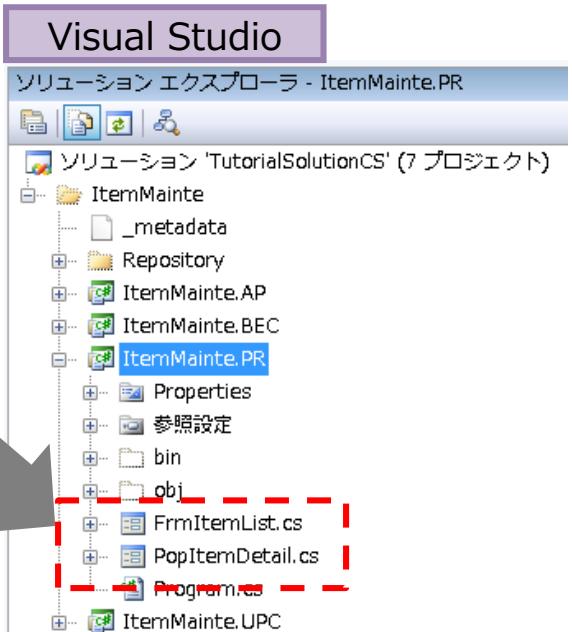
製造UT

#### 画面一覧(Excel)からFormを一括生成



画面一覧(Excel)からFormを一括生成することで手作業によるミスを防ぎ、作業効率の向上が図れます。

Windowsフォームモデルでは画面遷移定義XMLを入力として生成することも可能です。



※ MVCモデルは未対応

### 3.4.6. プレビュー機能

Win

Web

外部設計

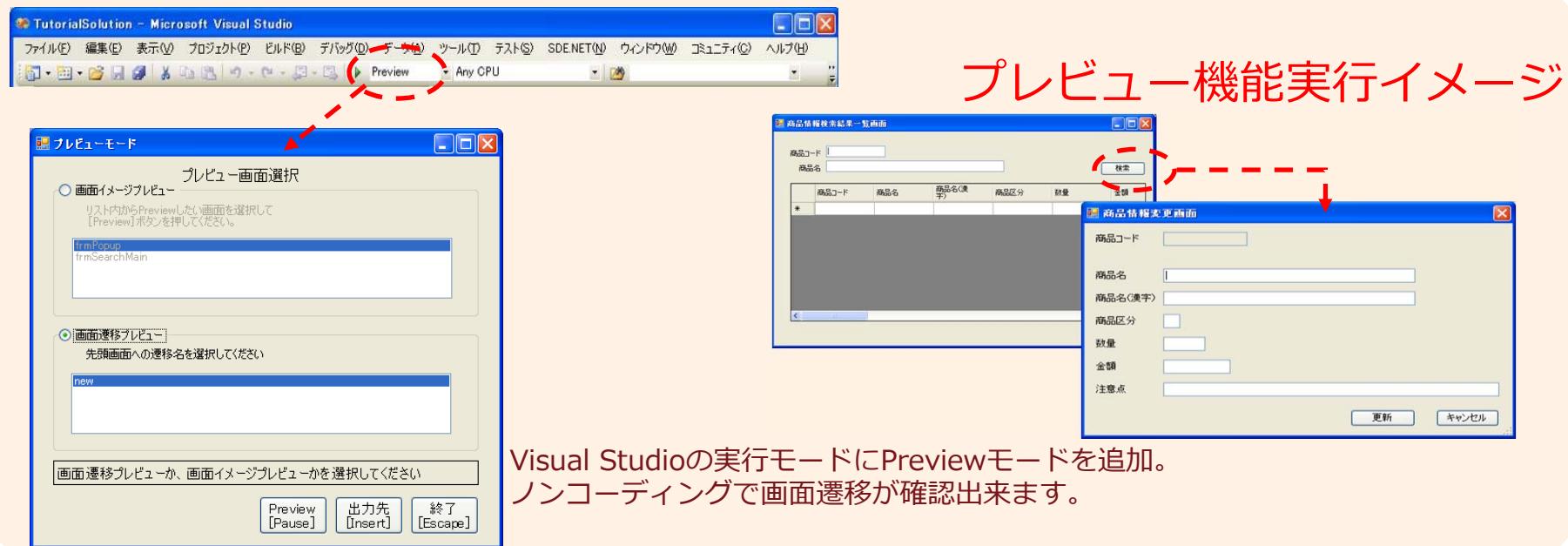
内部設計

製造UT

画面デザイン設計時にレイアウトや画面遷移を確認

業務ロジックの実装が行われていない状態でも画面遷移の確認が可能です。

デザイン設計段階で実行イメージや操作感を確認することで、仕様の定義モレや仕様の食い違いを早期発見し、下流工程での手戻りを未然に防ぎます。



### 3.4.7. デザインリフレクタ

Win

Web

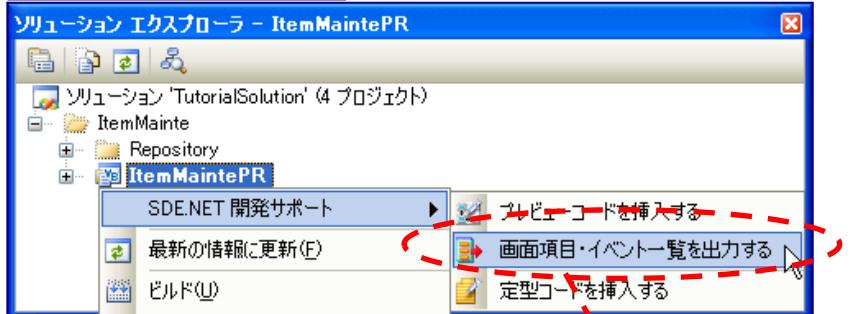
外部設計

内部設計

製造UT

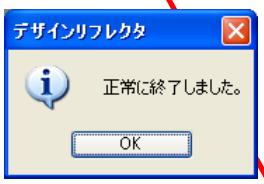
画面レイアウトから、画面とイベントの設計情報をExcelファイルに出力

Visual Studio



画面項目情報とイベント情報を取得し、Excelファイルとして出力します。

出力項目/ExcelフォーマットはPJ毎にカスタマイズが可能です。



設計情報

Microsoft Excel - 画面項目定義_frmSe.xls											
No.	項目名	オブジェクト名	種類	I/O	行数	属性	必須	初期値	チェック	フォーマット	テーブル名
8	txtPRODUCT_CD	TextBox		10	0						
9	txtPRODUCT_NN	TextBox		30	1						
10	btnSEARCH	Button			2			検索			
11	btnUPDATE	Button			3			更新			
12	Label1	Label			5			商品コード			
13	Label2	Label			6			商品名			
14	grdPRODUCT	DataGridView			7						
15											

### 3.4.8. インタラクションエディタ

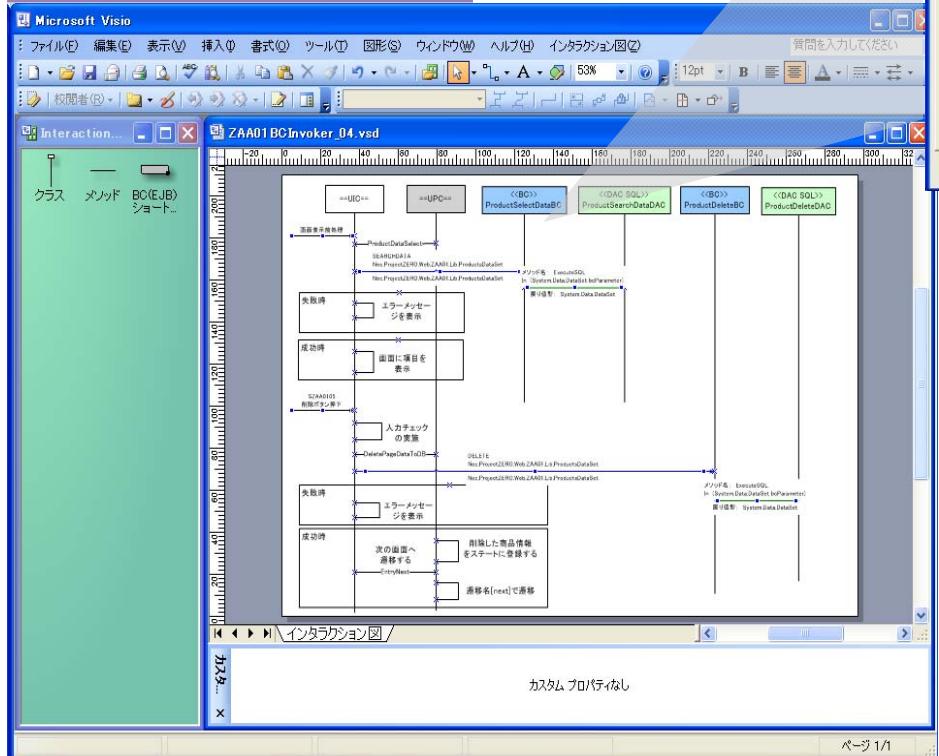
Win  
Web

外部設計

内部設計  
製造UT

AP層のコンポーネント構成をビジュアルなモデルとして設計  
ソースコードの雛形やAP層アクセスに必要なXML情報を自動生成

#### インタラクション図



Visioのテンプレートとしてインタラクションエディタを提供。PR層とAP層のインタラクション図が作成できます。

記述したインタラクション図に基づいて「AP層コードの雛形」、「アクセス定義 XML」、「仮想アクセス定義 XML」を生成します。

Excelの仕様書を用いて設計することも可能です。

### 3.4.9.ジェネレータ

Win

Web

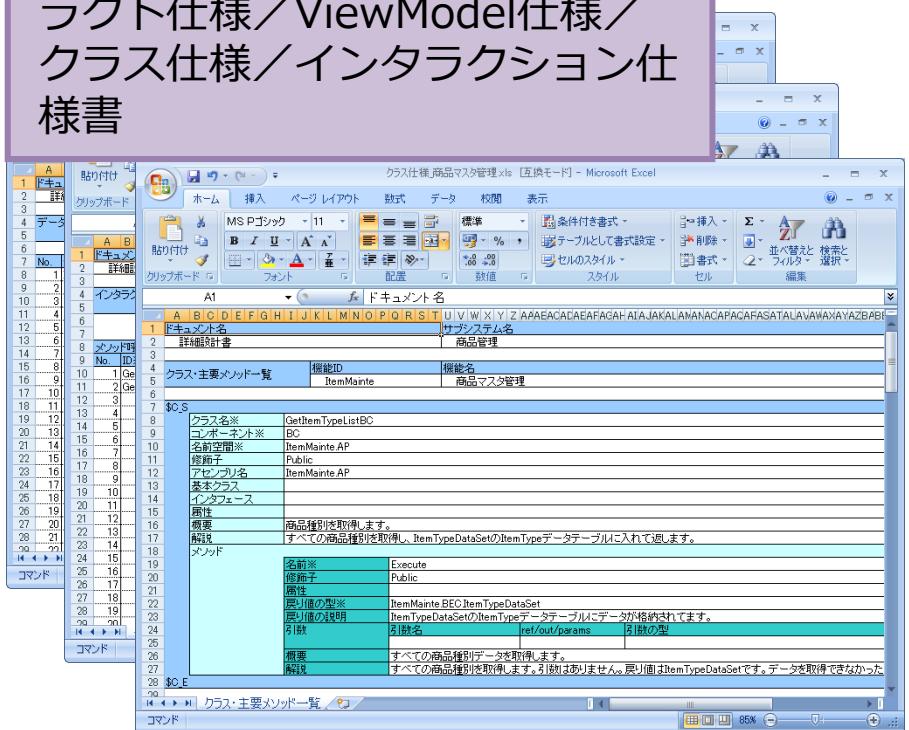
外部設計

内部設計

製造UT

内部設計で作成した仕様書(\*.xls)からエンティティクラスやBC/DACクラスのスケルトンを一括生成

データセット仕様／データコントラクト仕様／ViewModel仕様／クラス仕様／インターラクション仕様書



データセット仕様／データコントラクト仕様／ViewModel仕様／クラス仕様／インターラクション仕様書をExcel帳票ベースで作成します。

設計書フォーマットはPJ毎にカスタマイズが可能です。

作成した設計書の内容を元に、コードを自動生成することで、手作業によるミスの軽減と作業効率を向上します。

(仕様書毎の個別生成も可能)



BC/DACクラス

エンティティクラス

画面遷移コード

AP層呼び出しコード

## 3.4.10.DACビルダ

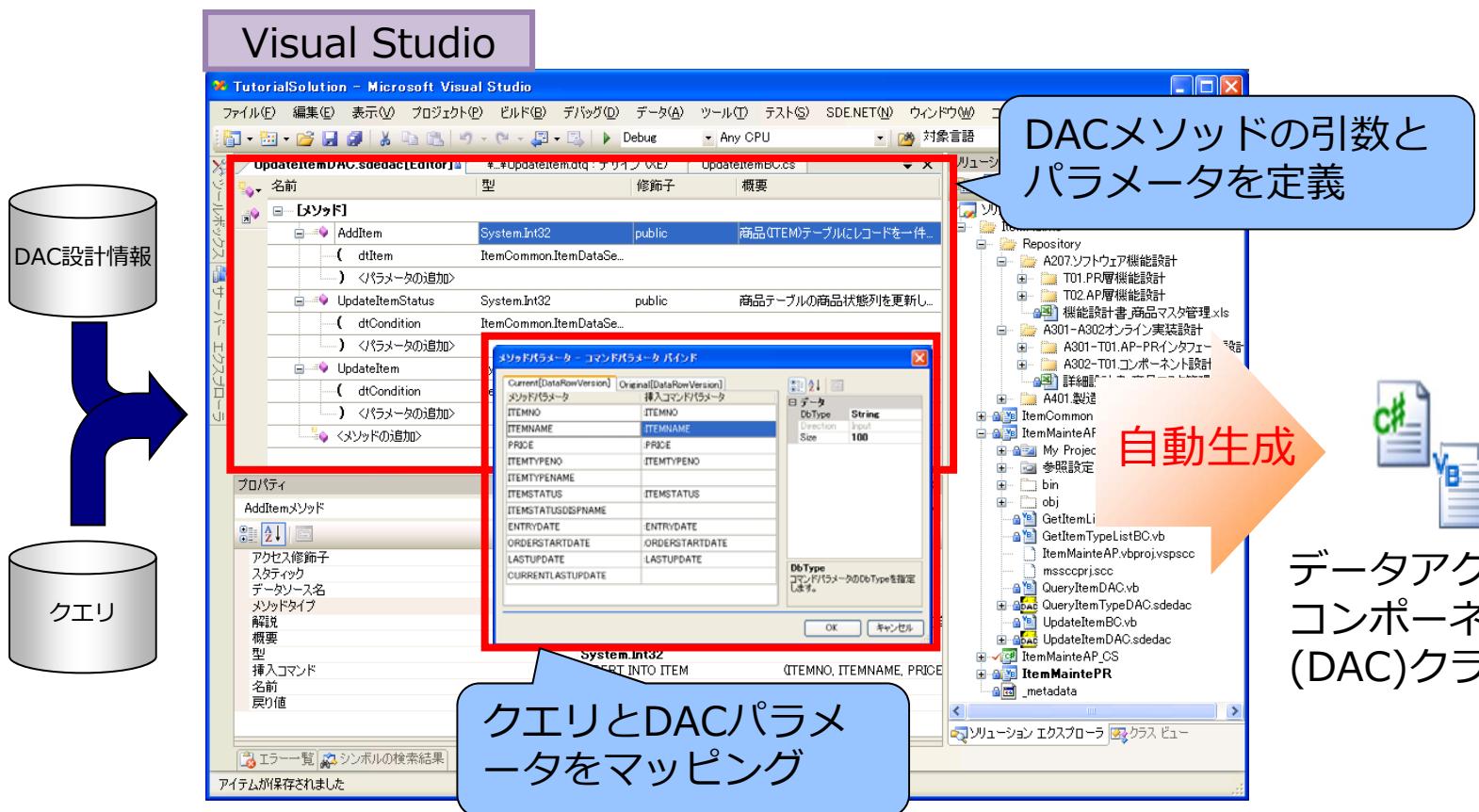
Win  
Web

外部設計

内部設計  
製造UT

Visual Studioデザイナ上で、クエリとDAC設計情報をマッピングして、ソースコードを自動生成

- データアクセスコンポーネントは、生成後必要に応じて自由にカスタマイズ可能。
- 設計情報の変更による再生成も、それまでに施したカスタマイズの内容を保護。



### 3.4.11. テスト機能連携

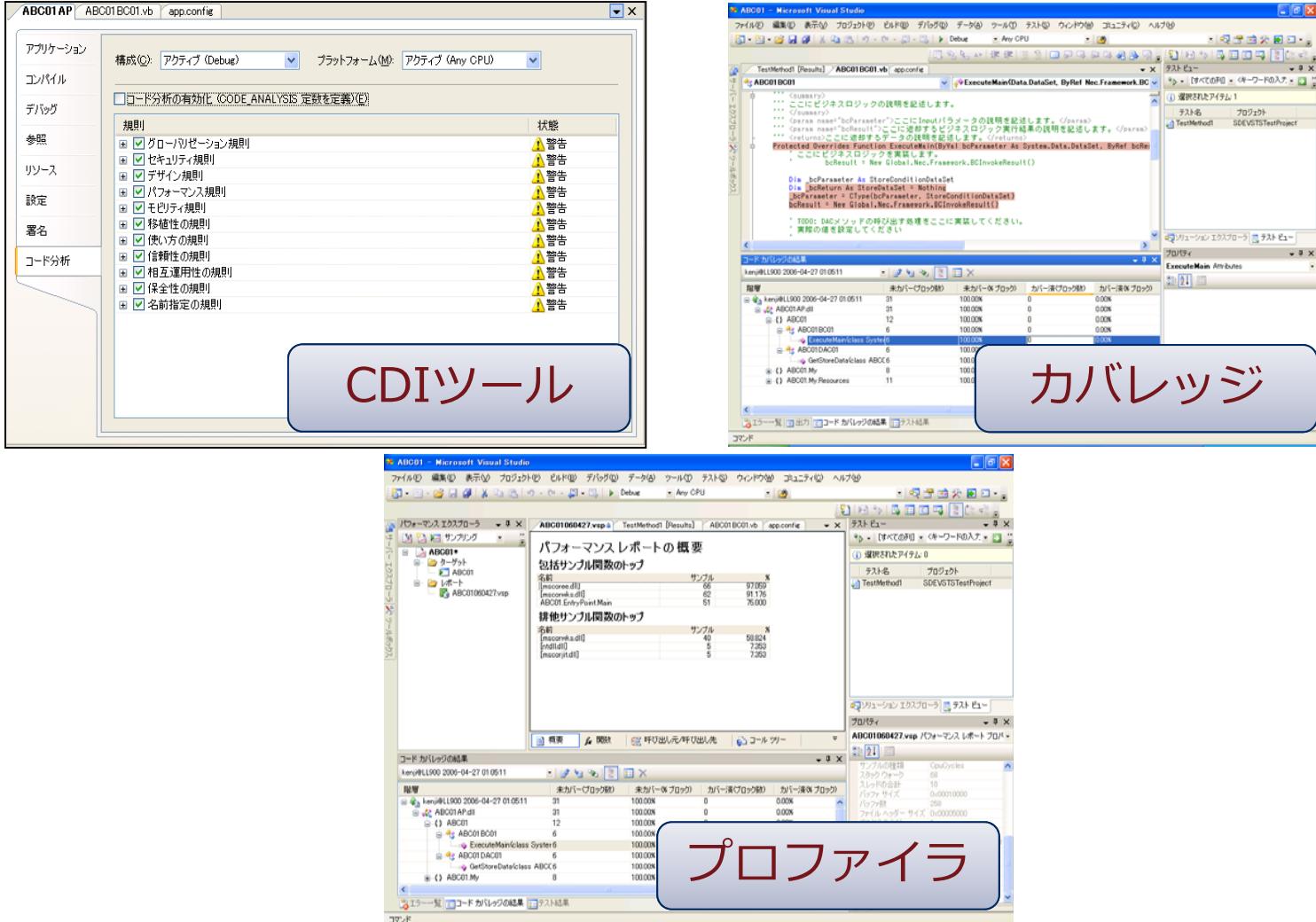
Win  
Web

外部設計

内部設計

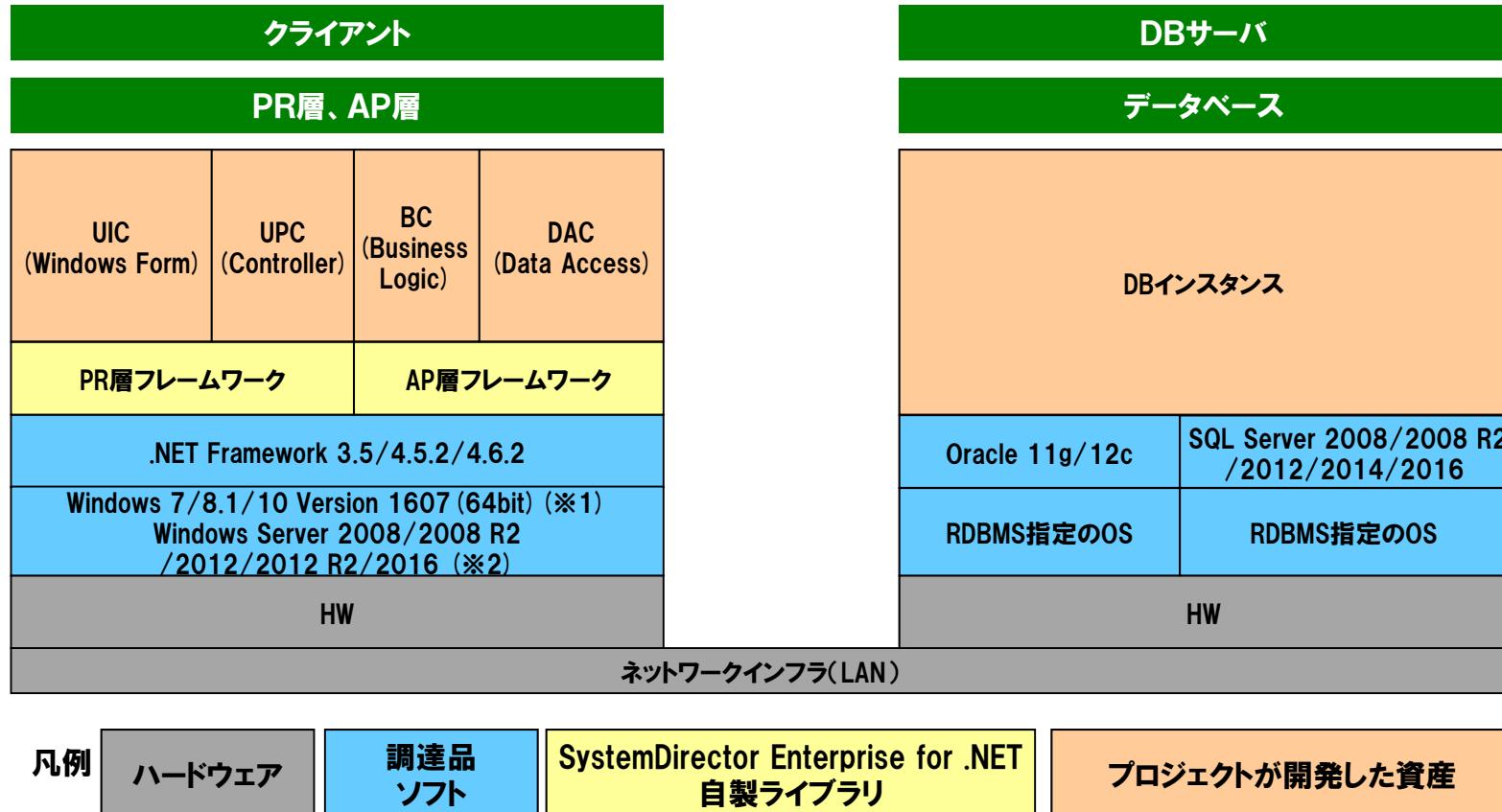
製造UT

Visual Studioのテスト機能と連携し、テスト効率と品質をさらに向上



### 3.5.動作環境

#### Windowsフォームモデル（2層）

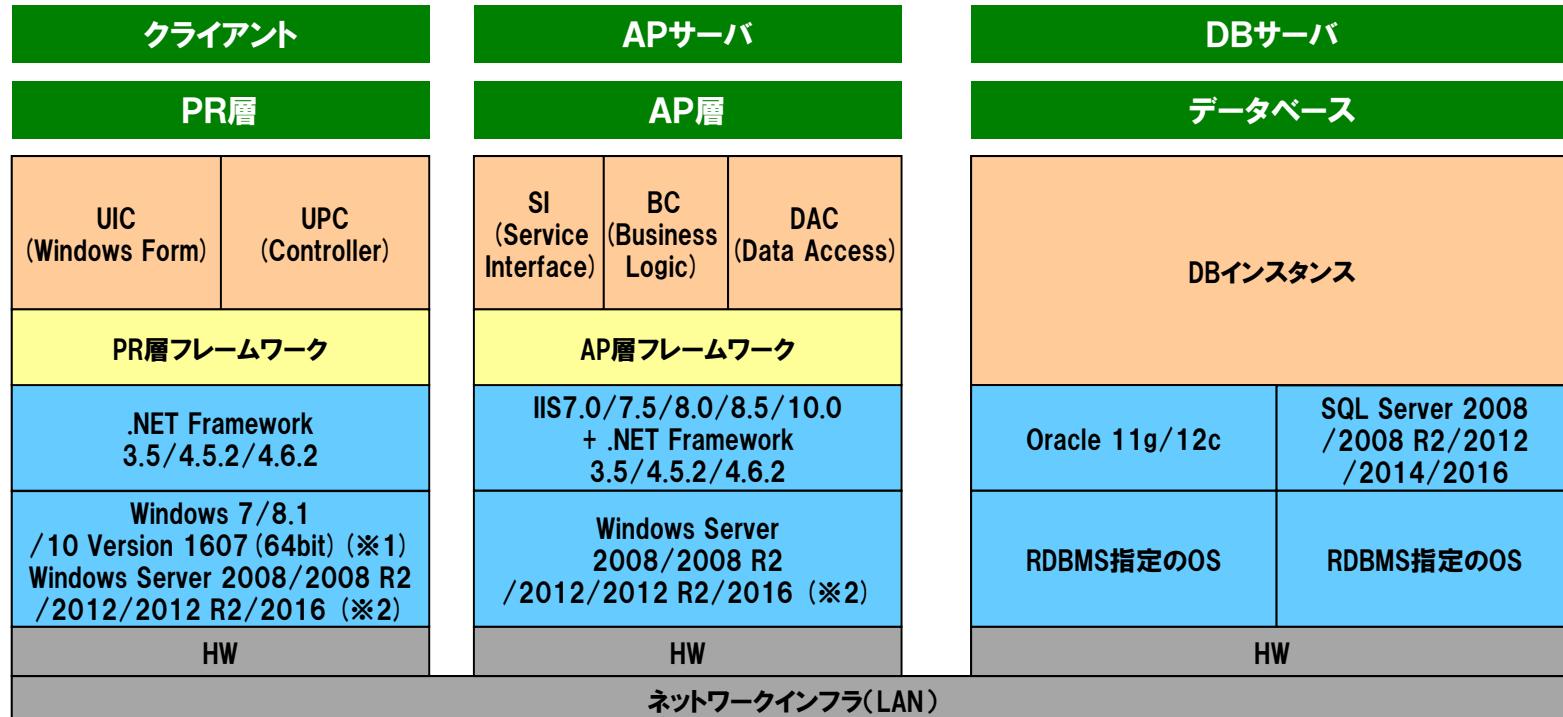


※1: 特に指定がない場合、32bit/64bitに対応

※2: Windows Server 2008は32bit/64bitに対応、2008 R2以降は64bitに対応

### 3.5.動作環境

#### Windowsフォームモデル（3層）



凡例

ハードウェア

調達品  
ソフト

SystemDirector Enterprise for .NET  
自製ライブラリ

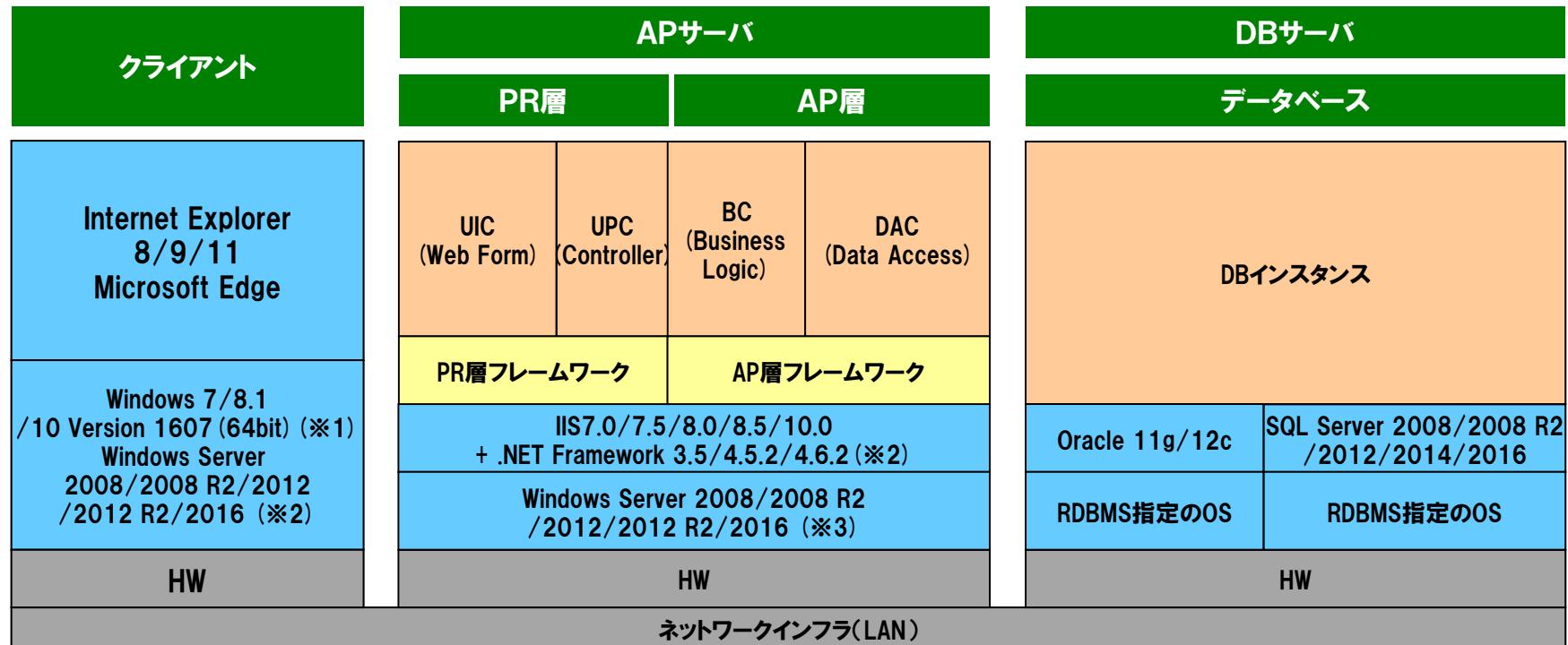
プロジェクトが開発した資産

※1: 特に指定がない場合、32bit/64bitに対応

※2: Windows Server 2008は32bit/64bitに対応、2008 R2以降は64bitに対応

## 3.5.動作環境

### Webフォーム/MVCモデル（3層）

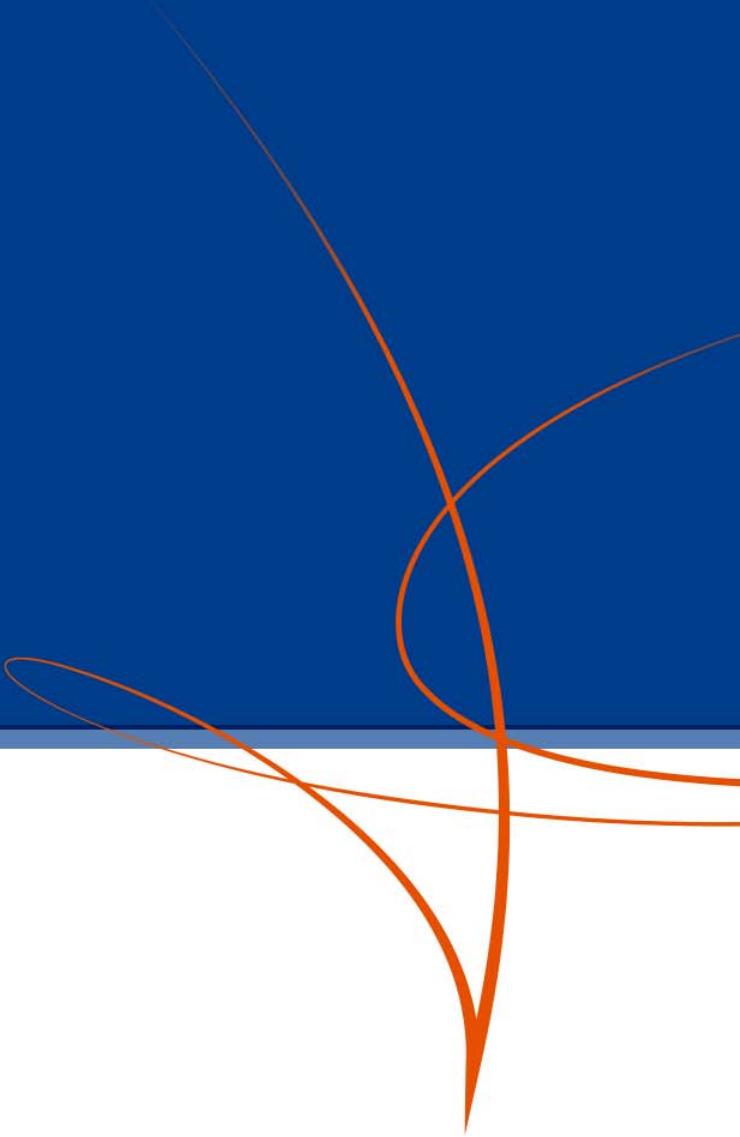


※1: 特に指定がない場合、32bit/64bitに対応

※2: MVCは.NET Framework 4.5.2/4.6.2対応

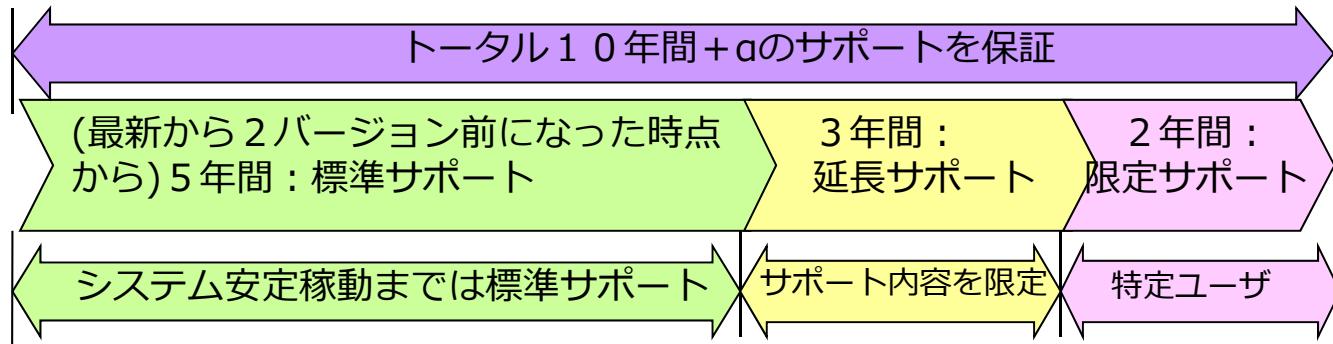
※3: Windows Server 2008は32bit/64bitに対応、2008 R2以降は64bitに対応

## 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	JavaによるWebアプリケーション開発1 -サーブレット・JSP-	Visual Studio による Windowsプログラミング	HTML5/CSS3基礎
	Javaによるデータベースアクセス	Webアプリケーション開発入門 (ASP.NET編)	JavaScriptプログラミング
	Javaプログラミング基礎 (入門、基礎1、基礎2)		

## 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>

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。

- 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**