(お客様用)

Webアプリケーション時代の開発標準

ComponentAA開発標準
ご紹介資料

2004年 5月20日

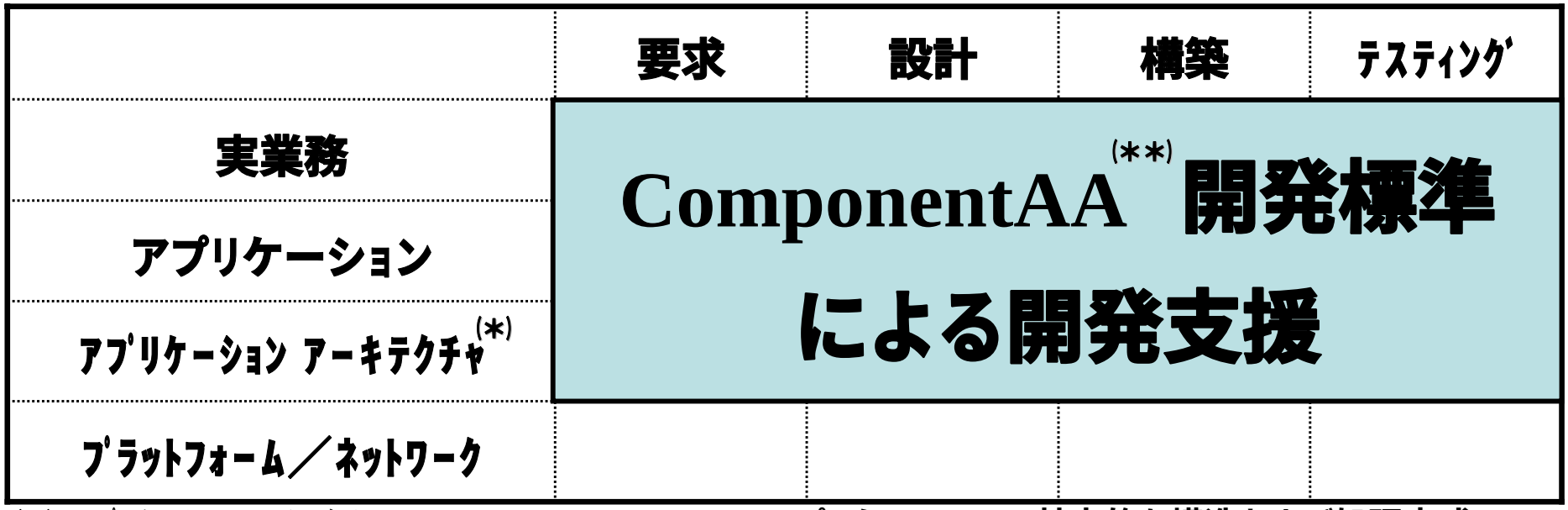
富士通株式会社

目　　次

1. ComponentAA開発標準とは

お客様に、短納期・高品質のシステムインテグレーションを提供するためのアプリケーション開発標準。

- ドキュメント標準
- 開発手順
- 設計／テストなどの技法／ガイド

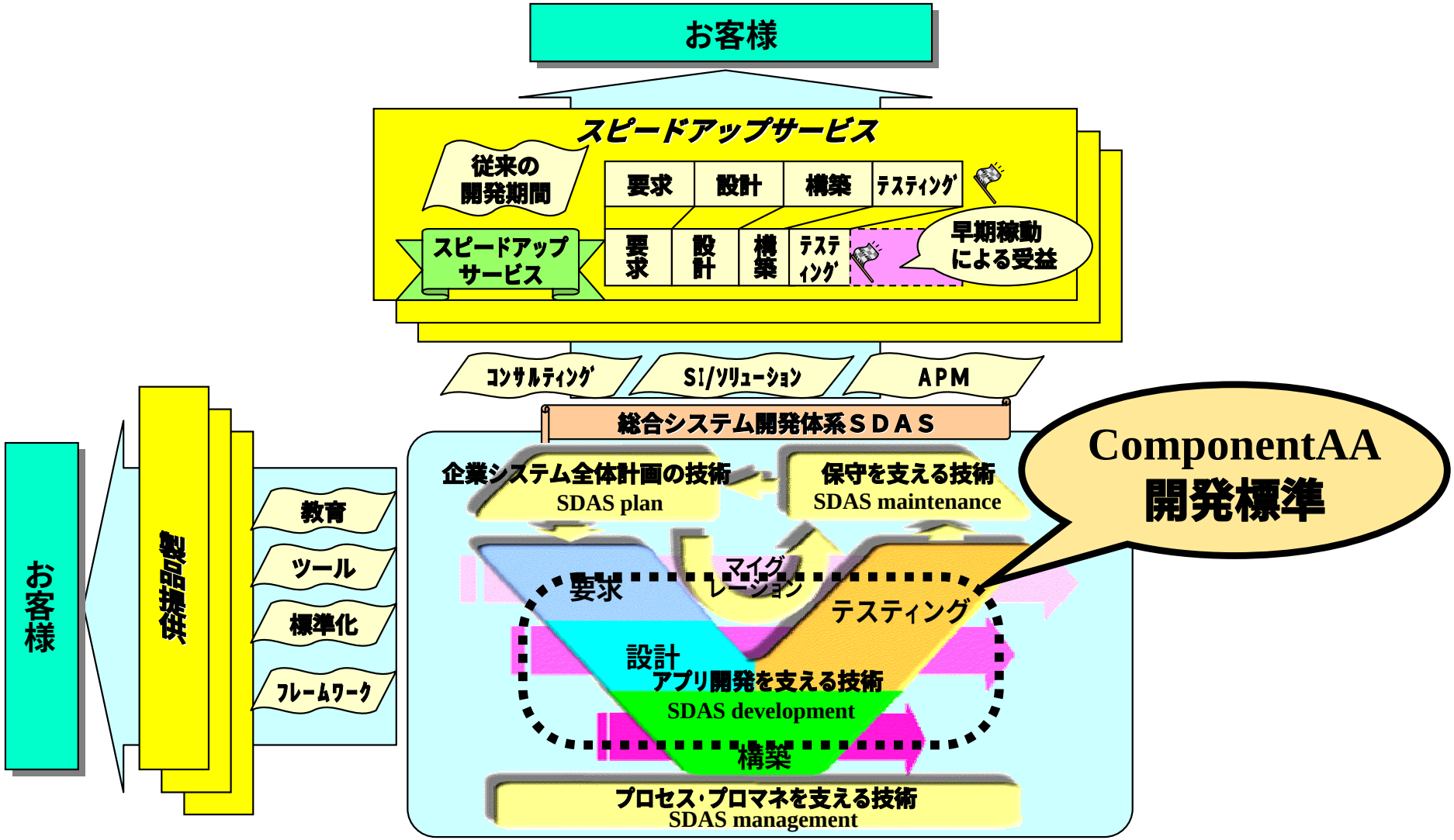


(*) ｱﾌﾟﾘｹｰｼｮﾝｱｰｷﾃｸﾁｬ(Application Architecture)：アプリケーションの基本的な構造および処理方式。

(**) ComponentAA：富士通のコンポーネントベースのアプリケーションアーキテクチャ。

1.1 総合システム開発体系SDASでの位置づけ

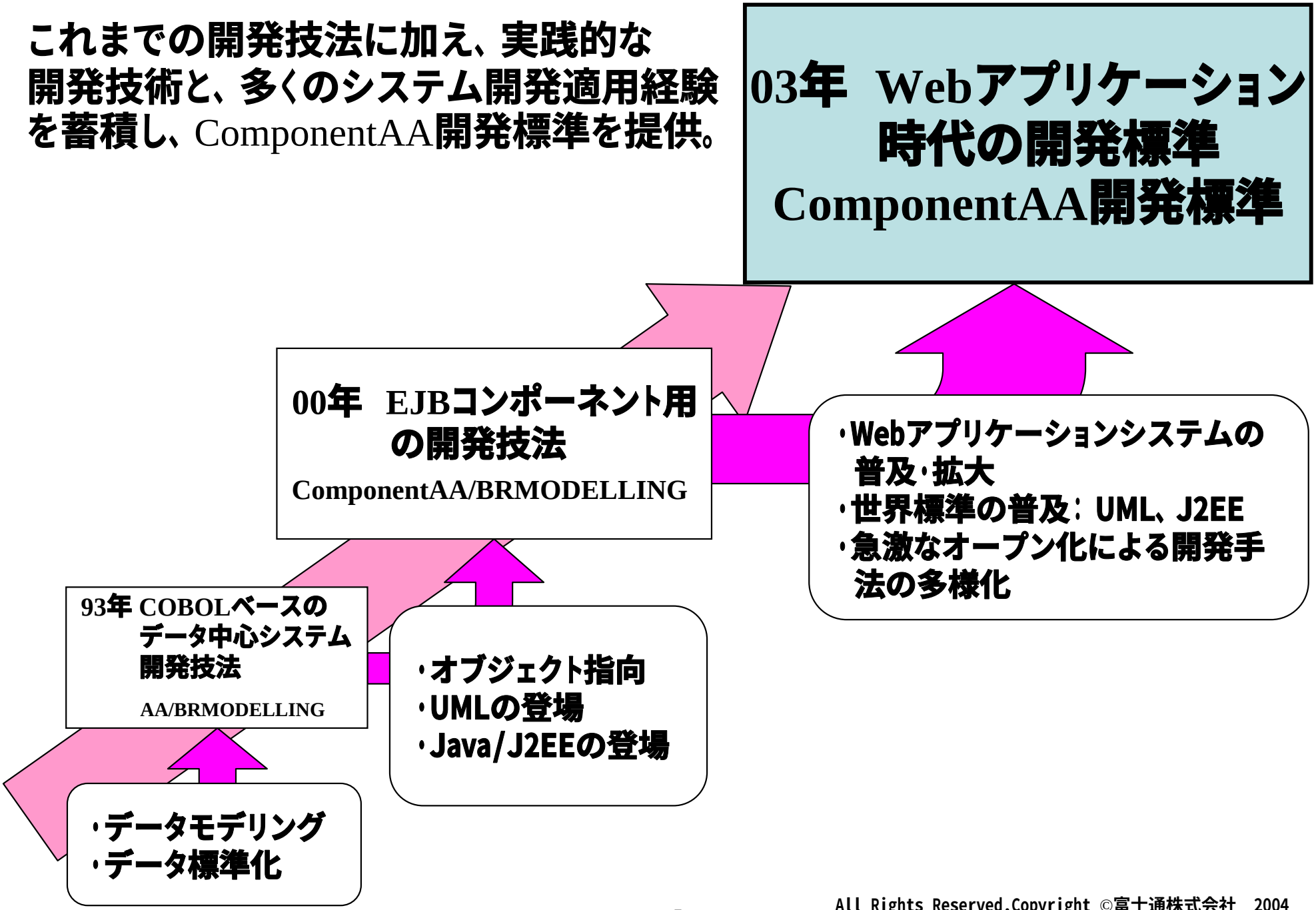
ComponentAA開発標準は、SDASのアプリケーション開発（SDAS development）の中核となる技術。



～お客様に高品質なシステムを短期に提供するための富士通の総合システム開発体系「ＳＤＡＳ」～

1.2 ComponentAA開発標準の生い立ち

これまでの開発技法に加え、実践的な開発技術と、多くのシステム開発適用経験を蓄積し、ComponentAA開発標準を提供。



2. 適用対象

ComponentAA開発標準は様々なシステムの開発に適用可能。

アプリケーション形態：
- オンライン（3層／2層）
- バッチ

開発言語と適用フレームワーク：
- Java／COBOL (*)
- 複数のフレームワークに対応

システム規模：
- 中小規模から大規模まで適用可

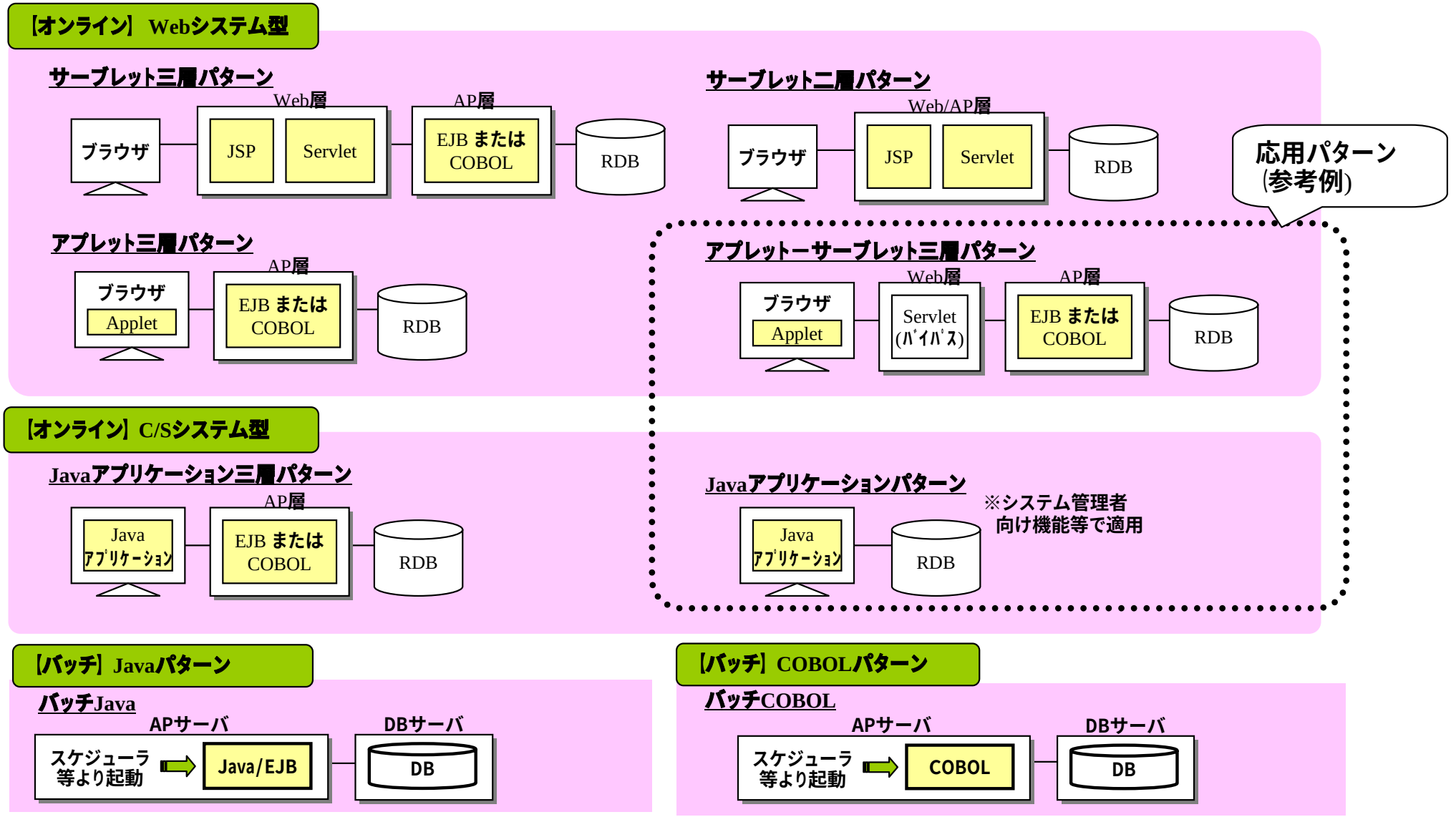
(*) ドキュメント標準の上流工程（SA～UI工程）は、他の開発言語でも適用可能。

2.1 アプリケーション形態

以下のアプリケーション形態に適用可能。

・プレゼン層はJava(Servlet/Applet/Javaアプリ)、アプリ層はJava(EJB)またはCOBOL。
二層システムに も対応。

・バッチはJavaまたはCOBOL。



※AP/DBサーバを同一サーバとする場合もあり

2.2 開発言語と適用フレームワーク

◆ 開発言語

Java

COBOL (APサーバのみ)

※その他の言語 (C,C++など)についても、多言語間連携の方式を提示予定。

◆ 適用フレームワーク

Java：

B².Sframeworkの以下のフレームワークに対応。

サーブレット： **Interstage Application Framework Suite (Apcoordinator)**
Struts等への応用展開も可

アプレット： **Client J Framework**

ＥＪＢ： **Interstage Application Framework Suite**

COBOL:

B².Sframeworkの以下のフレームワークに対応。

Interstage Application Framework Suite + e.Frame

ハードウェア、OS、ミドルウェア、RDBMSは、開発標準としては特に製品を限定しない。

2.3 システム規模

中小規模から大規模までの開発に適用可能。

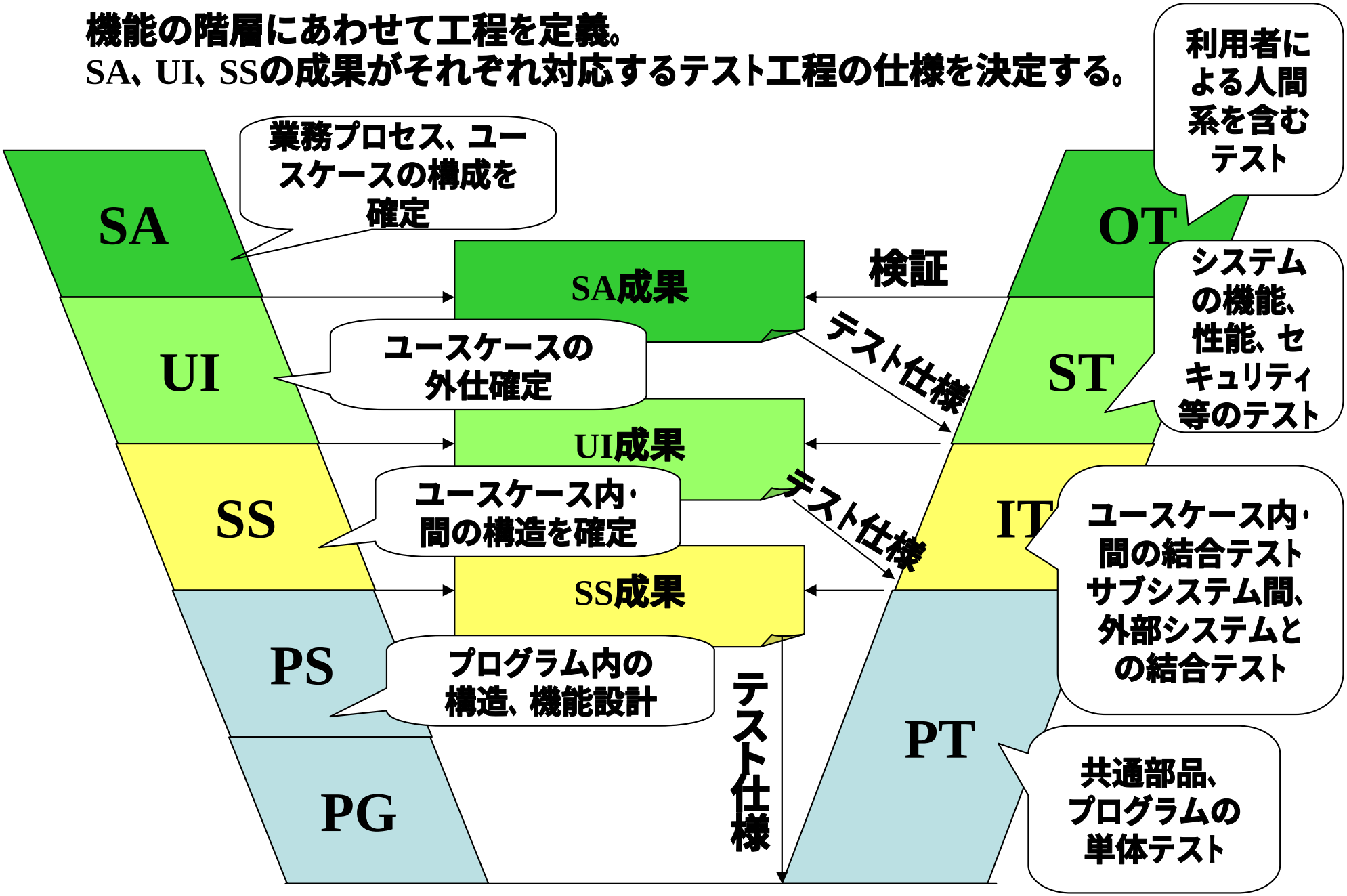
ドキュメント標準のカスタマイズ例を示し、中小規模システムへの適用も容易にしている。

ファイル名	No	工程名	ドキュメントコード	大規模Prjでのドキュメント標準	中規模Prjでのｶｽﾀﾏｲｽﾞ例		小規模Prjでのｶｽﾀﾏｲｽﾞ例	
					適用	備考	適用	備考
システム要件定義書（業務）	1	SA	SA01	業務要件定義	○		○	
	2		SA02	業務機能概要定義	○		○	
	3		SA03	業務運用要件	△	SA02に記述。	△	パッケージ開発のため運用要件は少ない。
	4		SA04	ｼｽﾃﾑ間ｲﾝﾀﾌｪｰｽ概要	×	一部記述。 FIXはUIで実施。	×	システム間連携なし。
	5		SA05	概念ER図	☆	UI工程で作成。	×	UI工程でテーブル関連図を作成。
	6		SA06	ﾕｰｽｹｰｽ・ｴﾝﾃｨﾃｨﾏﾄﾘｯｸｽ	☆	SAで省略。 UI工程で実施。	×	UI工程で作成。
	7		SA07	業務用語	○		△	定義数は少ない。

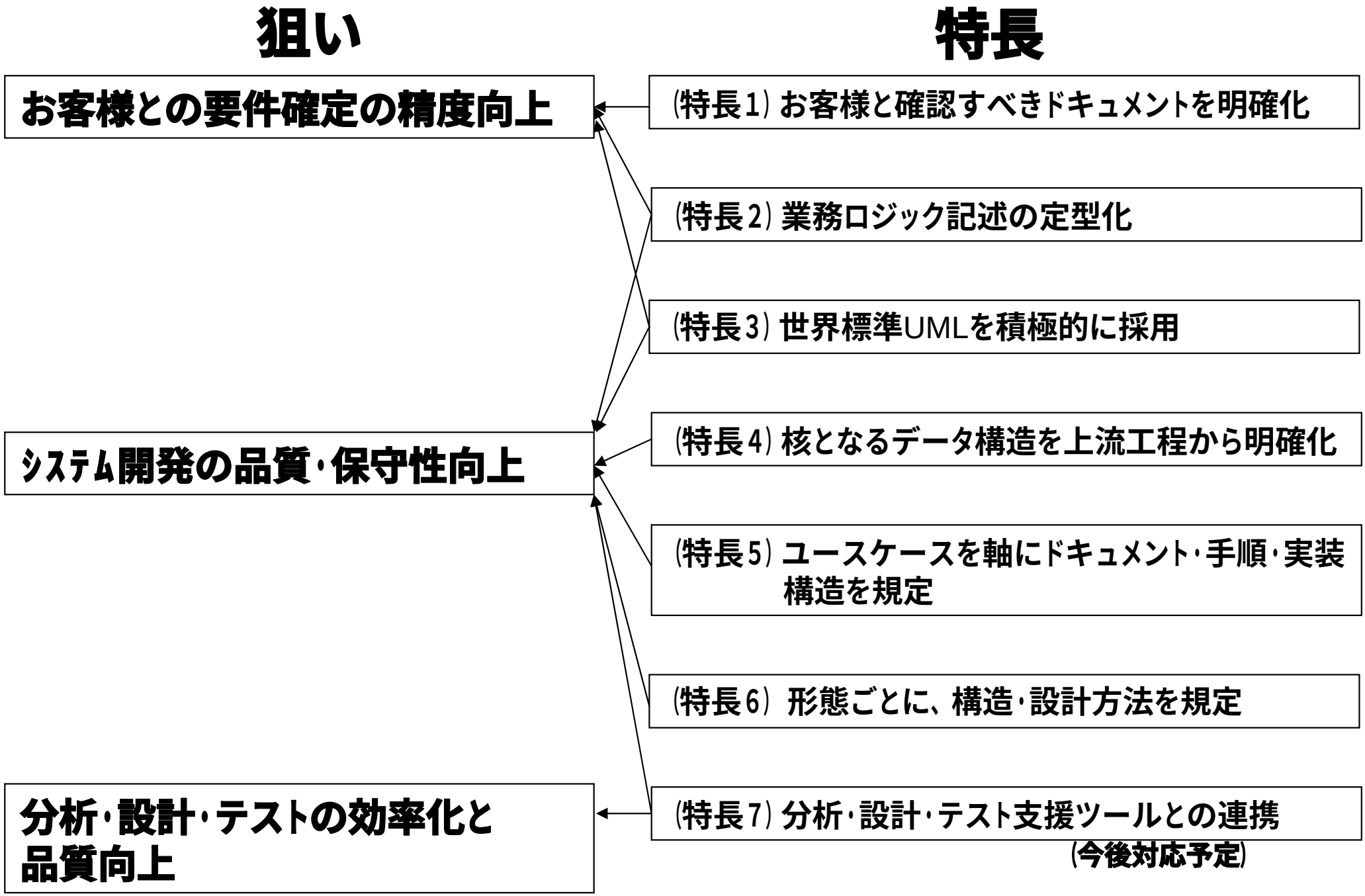
適用欄の意味：○→使用、△→他ドキュメントで代用、☆→工程変更（後工程で実施）、×→不必要

3. 開発工程

機能の階層にあわせて工程を定義。
SA、UI、SSの成果がそれぞれ対応するテスト工程の仕様を決定する。



4. ComponentAA開発標準の狙いと特長



4.1 お客様と確認すべきドキュメントを明確化

業務要件確定の早期化と精度向上の為、お客様と確認すべき、
必要最小限のドキュメントと、開発用ドキュメントを工程別に明確化。

工程	ドキュメント名	確認内容	顧客承認	
			情報システム部門	ｴﾝﾄﾞﾕｰｻﾞ
SA	業務要件定義	システム化要件と実現手段		○
	業務機能概要定義	「システム化範囲と対象業務」および「ユースケース機能概要」		○
	業務運用要件	業務運用要件		○
	業務用語定義	業務用語の定義		○
	システム間インタフェース概要	システム間インタフェース概要	○	
	概念ER図	(開発用ドキュメント)		
	ユースケース・エンティティマトリクス	(開発用ドキュメント)		
UI	ユースケース一覧	システム化業務フローとユースケースの整合性		○
	ユースケース機能記述	ユースケース毎の業務処理機能の詳細		○
	画面一覧	システムで使用する画面の種類および機能概要、利用部門		○
	帳票一覧	システムで使用する帳票の種類および機能概要、作成タイミング、出力場所等		○
	業務メッセージ定義	業務メッセージ仕様	○	
	テーブル／ファイル定義	キー項目、業務で使用する項目	○	
	データタイプ定義	データタイプの型、桁数、値の範囲	○	
	コード定義	名称、使用目的と概要説明、コード体系、値と範囲等	○	
	画面設計規約	画面設計についての規約	○	
	帳票設計規約	帳票設計につての規約	○	
	メッセージ規約	業務、システムのメッセージの規約	○	
	システム間インタフェース一覧	システム間、サブシステム間インタフェースの基本仕様	○	
	システム間インタフェース定義	システム間、サブシステム間インタフェースの詳細仕様	○	
	テーブル関連図	(開発用ドキュメント)		
	テーブル一覧	(開発用ドキュメント)		
	テーブル状態遷移	(開発用ドキュメント)		
	ユースケース・テーブルマトリクス	(開発用ドキュメント)		
	シナリオ記述	(開発用ドキュメント)		
	業務サービス仕様	(開発用ドキュメント)		

SA工程
開発用ドキュメント

UI工程
開発用ドキュメント

○：SA工程
顧客確認
ドキュメント

○：UI工程
顧客確認
ドキュメント

4.2 業務ロジック記述の定型化

ユースケースの業務ロジックの記述を、定義単位ごとに、図・表形式の定型記述とすることで、定義漏れを防止し、定義精度を向上。

ユースケースの確定

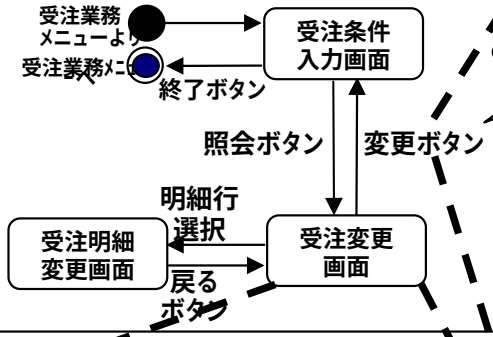
ユースケース一覧

	ユースケース名	
	XX処理	

ユースケース単位に機能概要設計

オンライン機能概要-画面遷移図

ユースケース名	XX処理
機能概要	・・・



画面項目定義

画面レイアウト定義

帳票項目定義

帳票レイアウト定義

アクション：
画面遷移を伴う利用者からシステムへの処理要求動作。

アクション単位に業務ロジック設計

アクション総括記述

遷移先画面表示定義

遷移元画面チェック定義

アクション

ユースケース名	XX処理
画面名	受注変更画面
アクション	変更

	項目名	チェック仕様およびエラー処理仕様	

入力データ処理定義

出力データ処理定義

ユースケース名	XX処理
画面名	受注変更画面
アクション	変更
出力テーブル名	受注テーブル

	項目名	出力データ処理仕様	

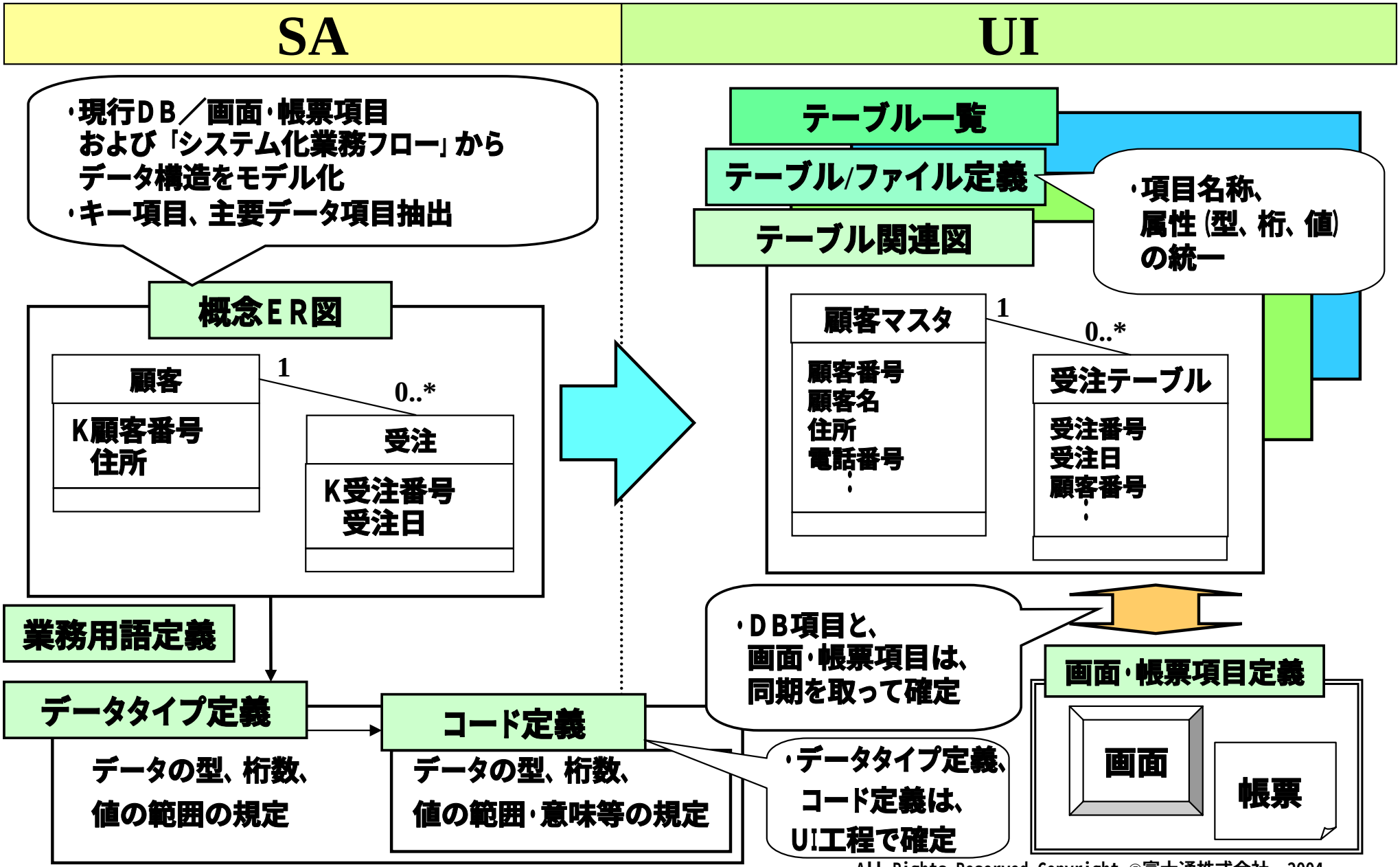
4.3 国際標準のＵＭＬを積極的に採用

実プロジェクト適用での評価を基に、国際標準の表記法を積極的に採用。

UML対応ドキュメント名	採用したUMLダイアグラム名	備　考
SA02 業務機能概要定義	アクティビティ図	表形式の書式と併用
SA05 概念ＥＲ図	クラス図	
UI02 ユースケース機能記述	アクティビティ図	
UI05 テーブル関連図	クラス図	
UI09 テーブル状態遷移	ステートチャート図	表形式の書式と併用
SS02 画面遷移定義	アクティビティ図	

4.4 核となるデータ構造を上流工程から明確化

核となるデータ構造およびデータ定義を上流工程から明確化。
⇒ システム開発の品質・保守性を向上。



4.5 ユースケースを軸にドキュメント・手順・実装構造を規定

ユースケースを軸とした機能単位で、仕様体系と実装構造を規定。
⇒システム開発の品質・保守性を向上。

機能構造の単位

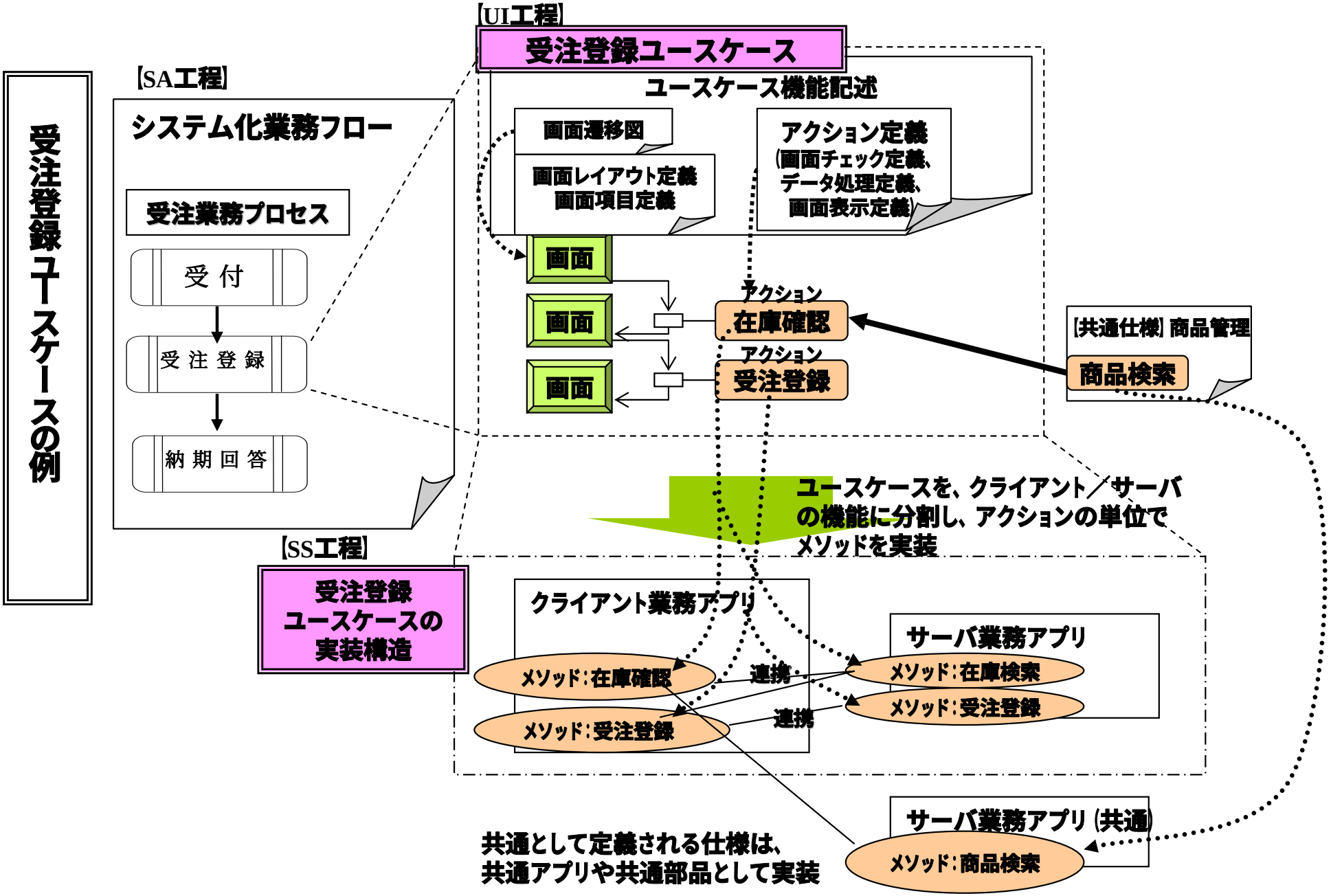
ユースケース：
利用者が開始し、その利用者にとって業務的に意味のある目的を完了するまでのプロセス。

オンラインでは、複数画面による一連の会話を通じて完結するプロセス。
（例）・受注1件のデータをDBに登録
・ATMでの普通預金引出し
・窓口で列車の座席を予約
バッチでは、業務的意味のあるくくりでの、データ作成・出力のプロセス
（例）・定期的に実行する売上実績データの集計
・メニュー起動による受注一覧表の印刷

アクション：
オンライン処理の会話で発生する、画面遷移を伴う、利用者からシステムへの処理要求動作。
（例）メニュー選択、登録ボタン押下、キャンセルボタン押下

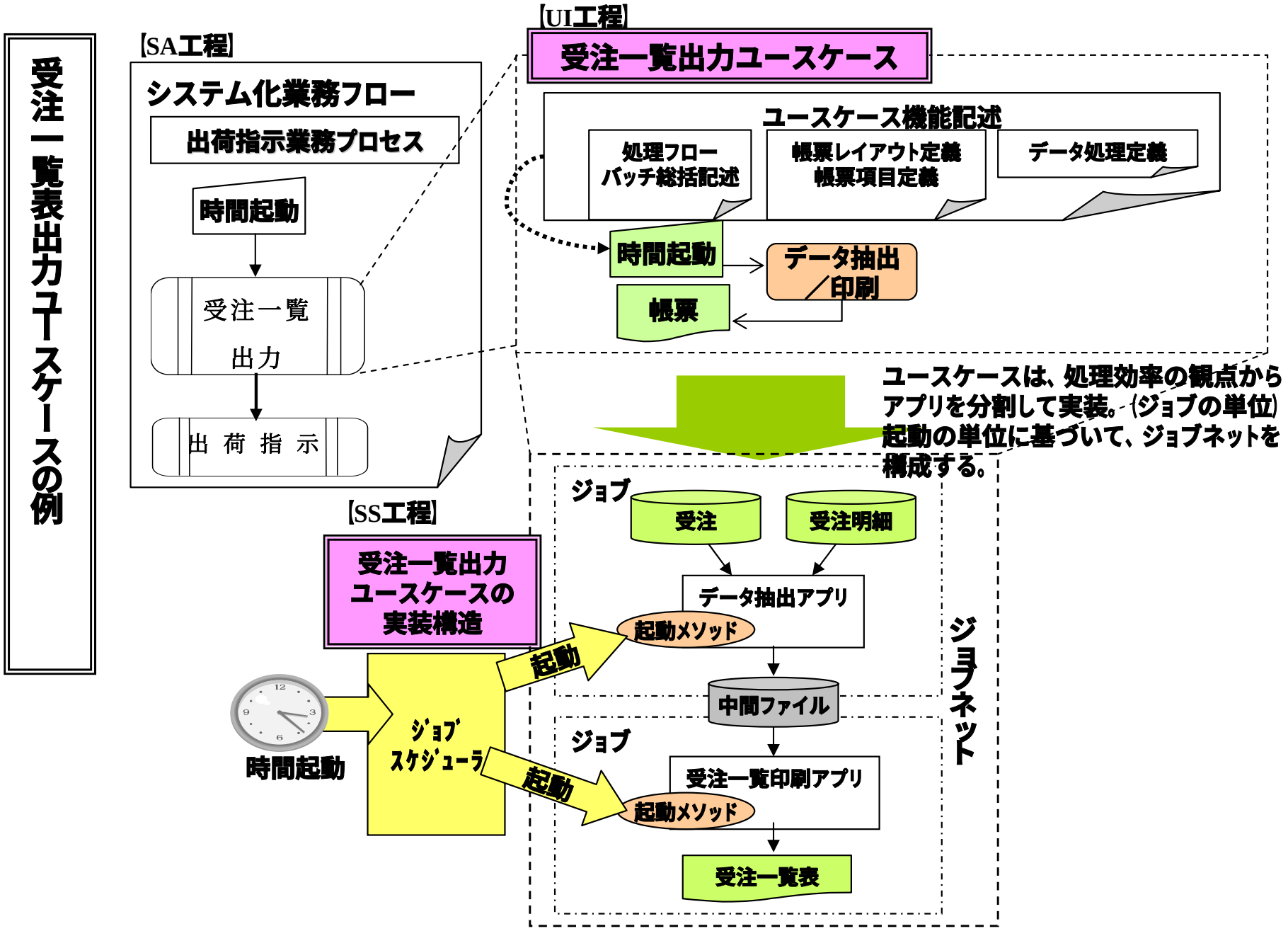
(1) オンライン処理の機能構造

ユースケースを軸とした機能単位で構造を規定。



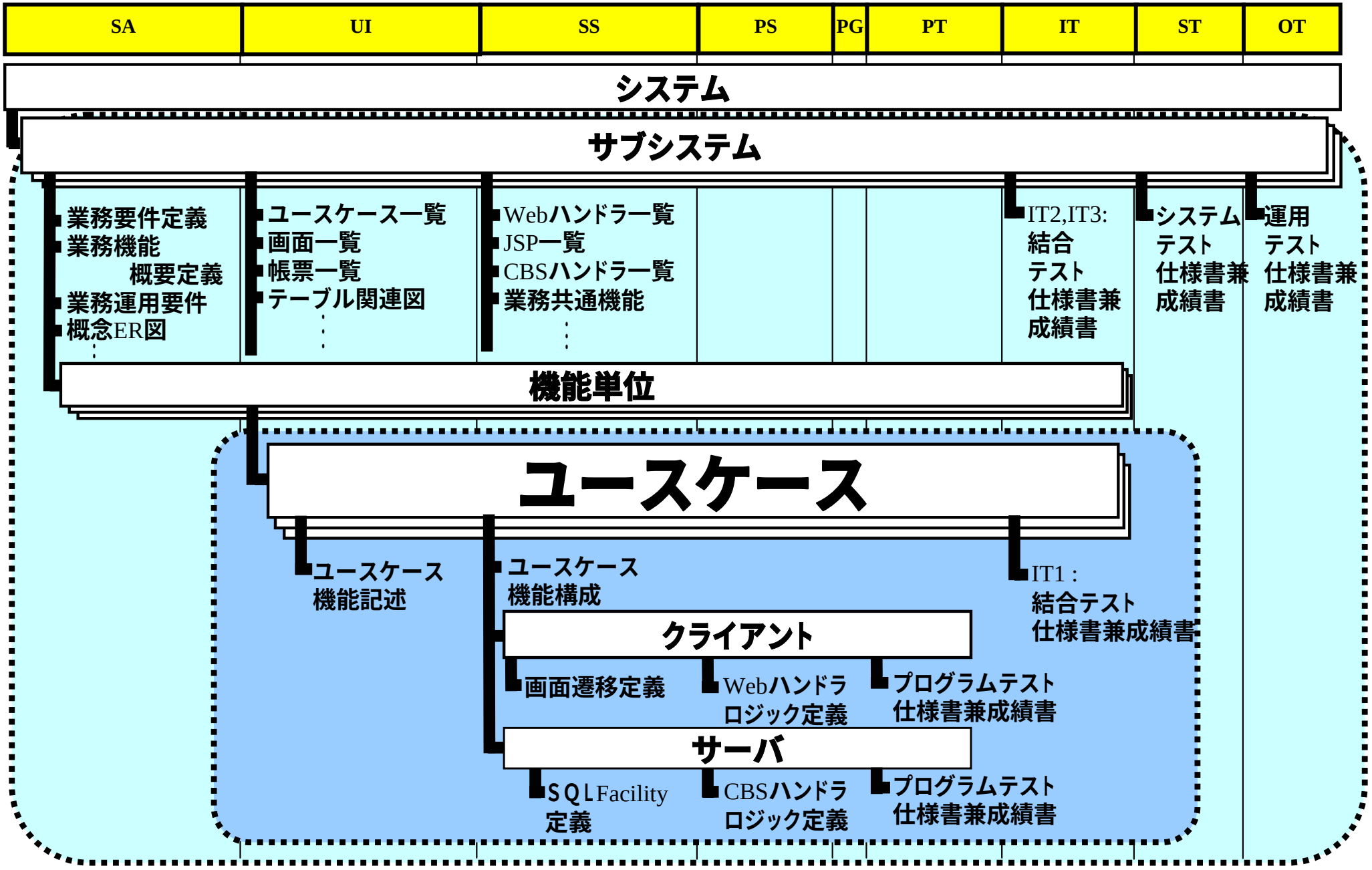
(2) バッチ処理の機能構造

ユースケースを軸とした機能単位で構造を規定。



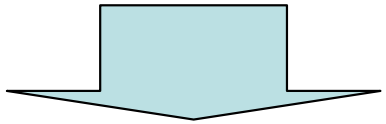
(3) ドキュメント管理体系

ユースケースを軸としたドキュメント管理体系により、進捗・品質の開発管理単位を明確化。



4.6 形態ごとに構造・設計方法を規定

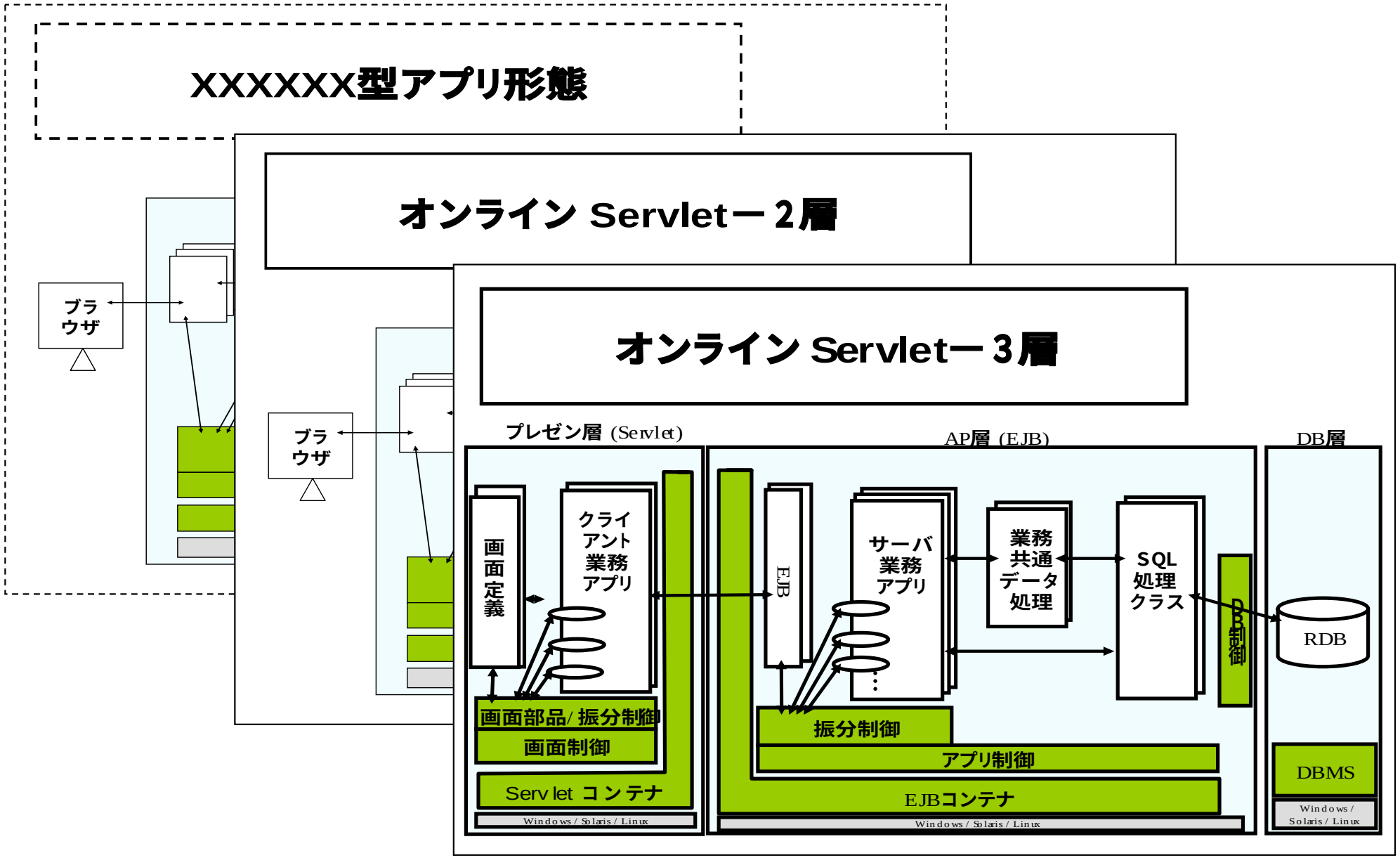
- アプリケーション形態ごとにアプリケーションの構造を規定
- 設計方法のパターン化
  アプリケーション内のコンポーネントの役割分担の指針、
  DBアクセス部品の設計指針など



システム開発の品質・保守性を向上

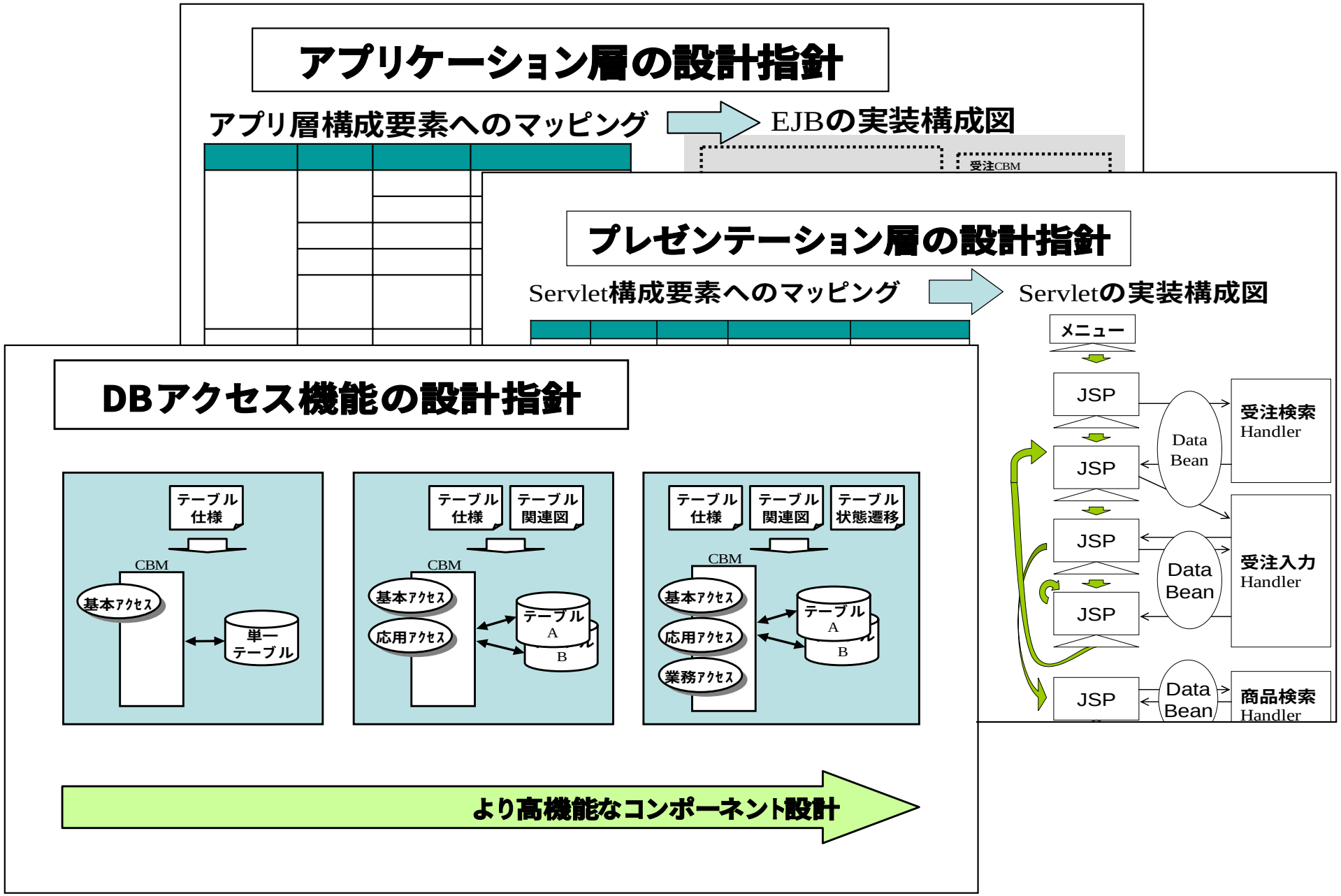
(1) アプリケーション形態ごとにアプリケーション構造を規定

アプリケーション形態ごとのアプリケーションアーキテクチャを提供。今後さらに拡充予定。



(2) 設計方法のパターン化

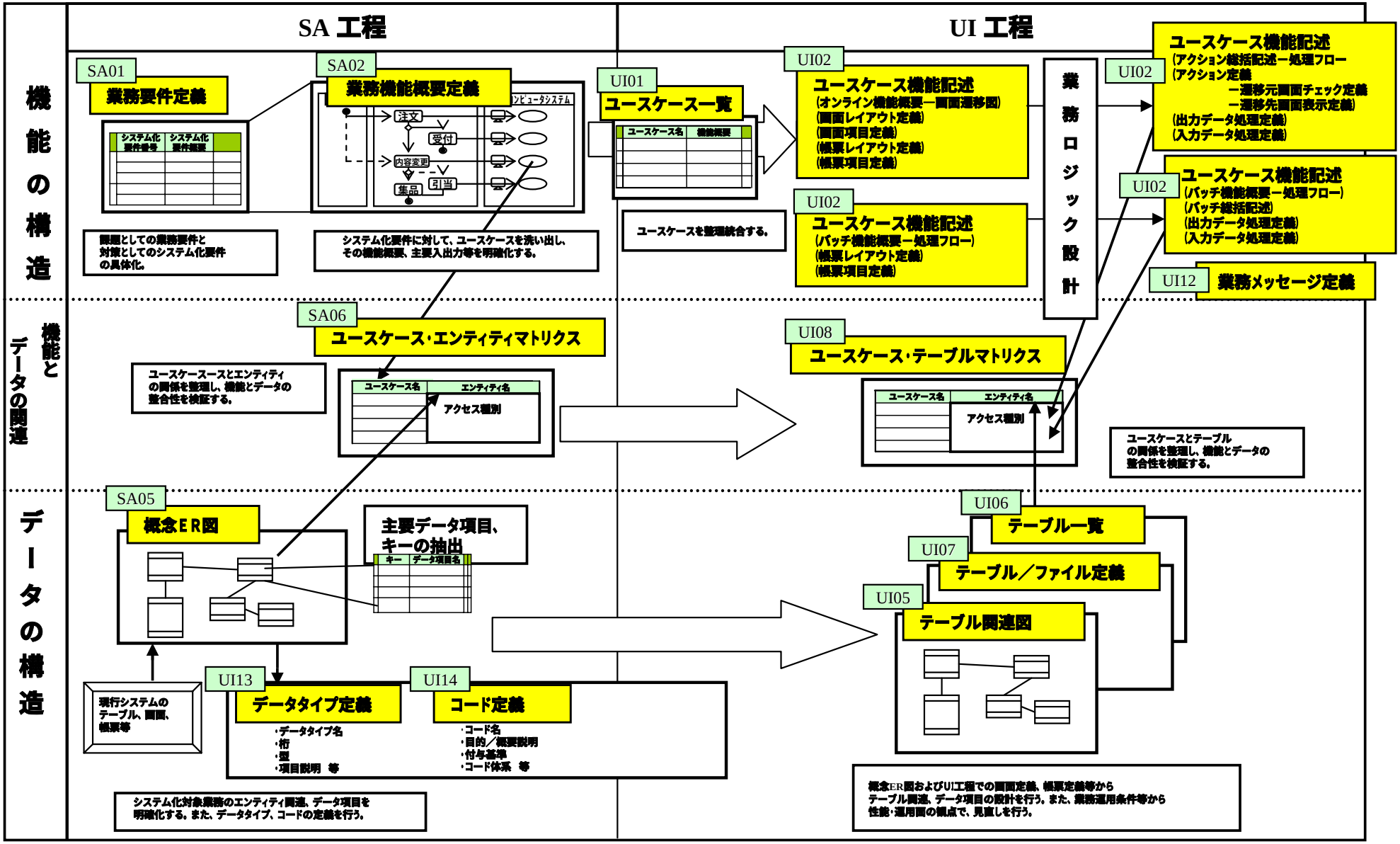
アプリケーション方式設計の方法をパターン化して提供。



ご参考１：主要な開発の流れ

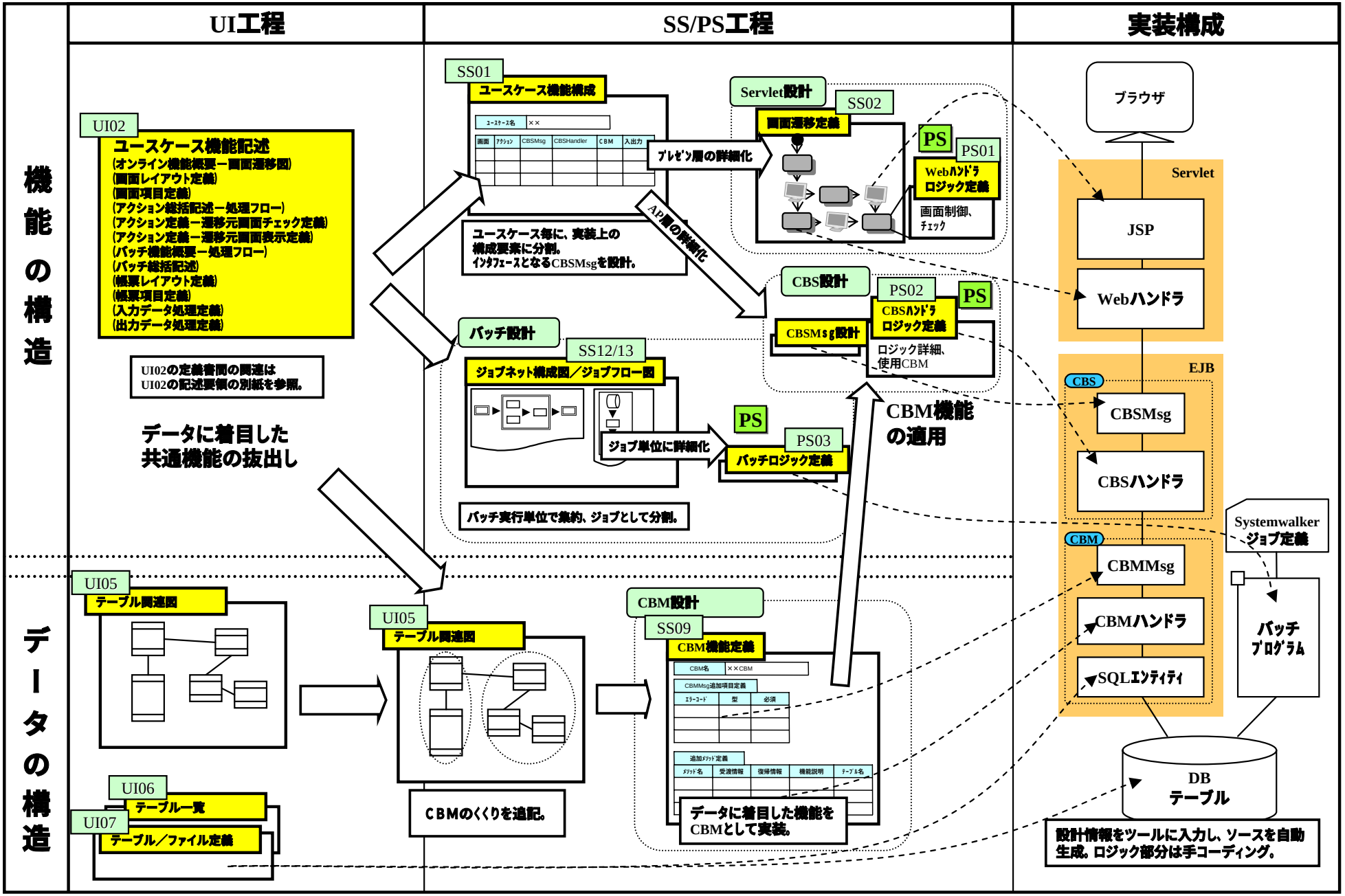
(1) ＳＡ、ＵＩ工程

機能とデータを、それぞれ相互の関係を検証しながら、ユースケースを軸とした設計を進める。



(2) ＳＳ～実装工程

ユースケースを軸として、サーバー、クライアントのアプリ構造設計を進める。



ご参考2：推奨ソフトウェア構成

層				製品名
クライアントPC	OS			Windows
	Servlet／Appletの場合		ブラウザ	Internet Explorer, Netscape Navigator
	Appletの場合		Javaプラグイン、JavaVM	ComponentAA/Client J (JBKプラグイン) を推奨
			Appletフレームワーク	Client J Frameworkを推奨
	Javaｱﾌﾟﾘｹｰｼｮﾝの場合		JavaVM	JDK1.3以降
			Javaフレームワーク	Client J Frameworkを推奨
Webサーバ	OS			Windows, Solaris, Linux
	WWWサーバ			Interstage Application Server, Apactch など
	Servletの場合		Servlertコンテナ	Interstage Application Server
			Servletフレームワーク	Interstage Application Framework Suite (Apcoordinator)を推奨
APサーバ	OS			Windows, Solaris, Linux
	オンライン	EJBの場合	EJBコンテナ	Interstage Application Serverを推奨
			EJBフレームワーク	Interstage Application Framework Suite を推奨
		COBOLの場合	OLTP	Interstage Application Server
			COBOLフレームワーク	Interstage Application Framework Suite ＋ e.Frame
	バッチ	ジョブスケジューラ		Systemwalker Operation Managerを推奨
		COBOLの場合	COBOLランタイム	NetCOBOL
			COBOLフレームワーク	e.Frame
DBサーバ	OS			Windows, Solaris, Linux
	RDBMS			Symfoware, Oracleなど

ご参考3：総合システム開発体系SDASのサイト

以下にて最新情報をご提供しておりますので、ご参照ください。

- ComponentAA開発標準のドキュメント標準
- 製品・サービスのカタログとご紹介資料
- 解説記事
- 導入事例

http://segroup.fujitsu.com/sdas/index.html