MITSUBISHI SPACE SOFTWARE

ソフトウェア要求分析から詳細設計まで シームレスにつなぐ開発手法

2013年9月20日

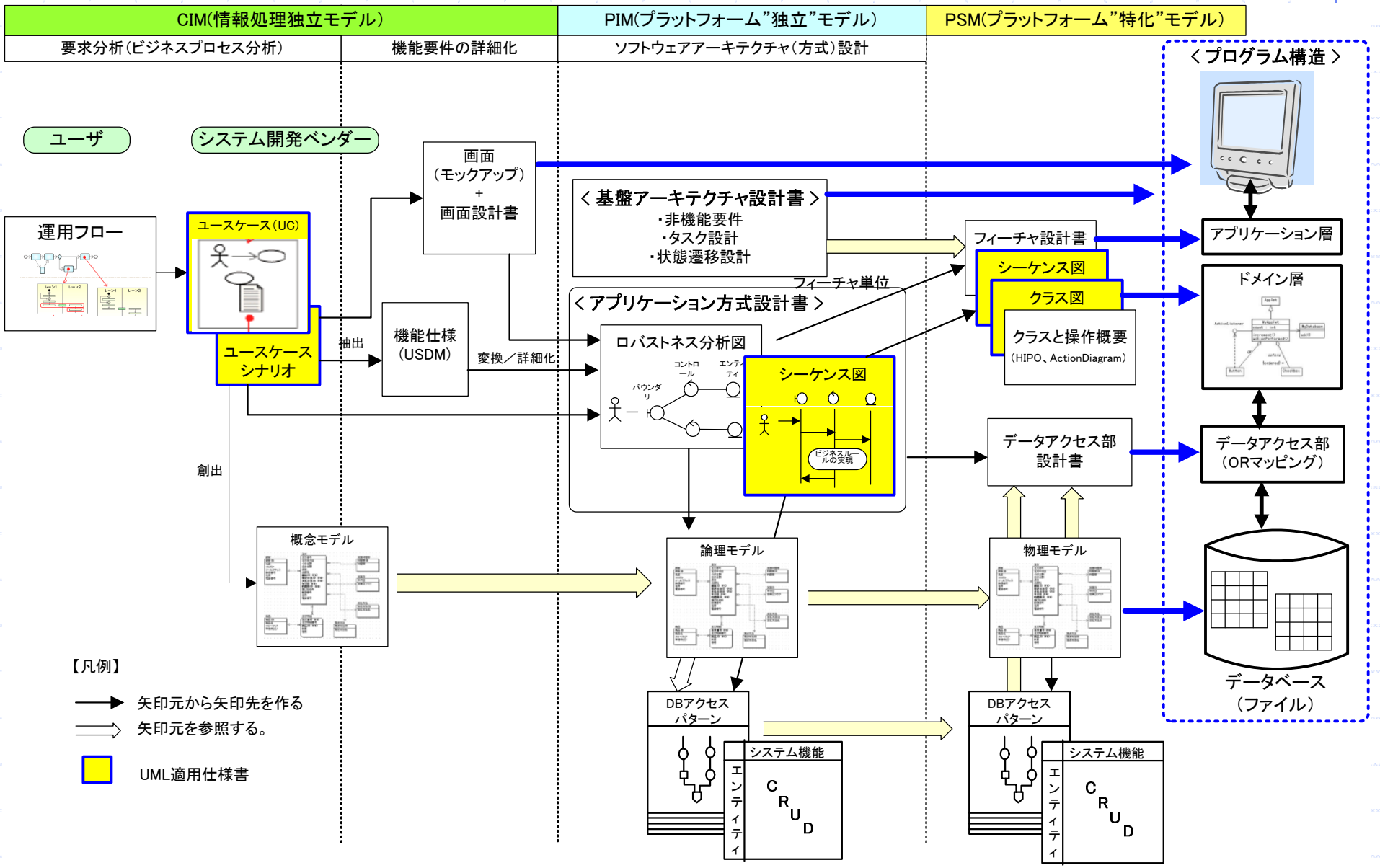
三菱スペース・ソフトウエア株式会社

目次

ソフトウェア設計手順の概要

MITSUBISHI SPACE SOFTWARE

情報系・組込共通に利用できる、汎用的な設計方法論（ユースケース駆動）に基づいた開発手順書を実行している。



＜開発手順作成のポイント＞

①開発フェーズ間の成果物トレーサビリティ

②開発フェーズ内の設計粒度

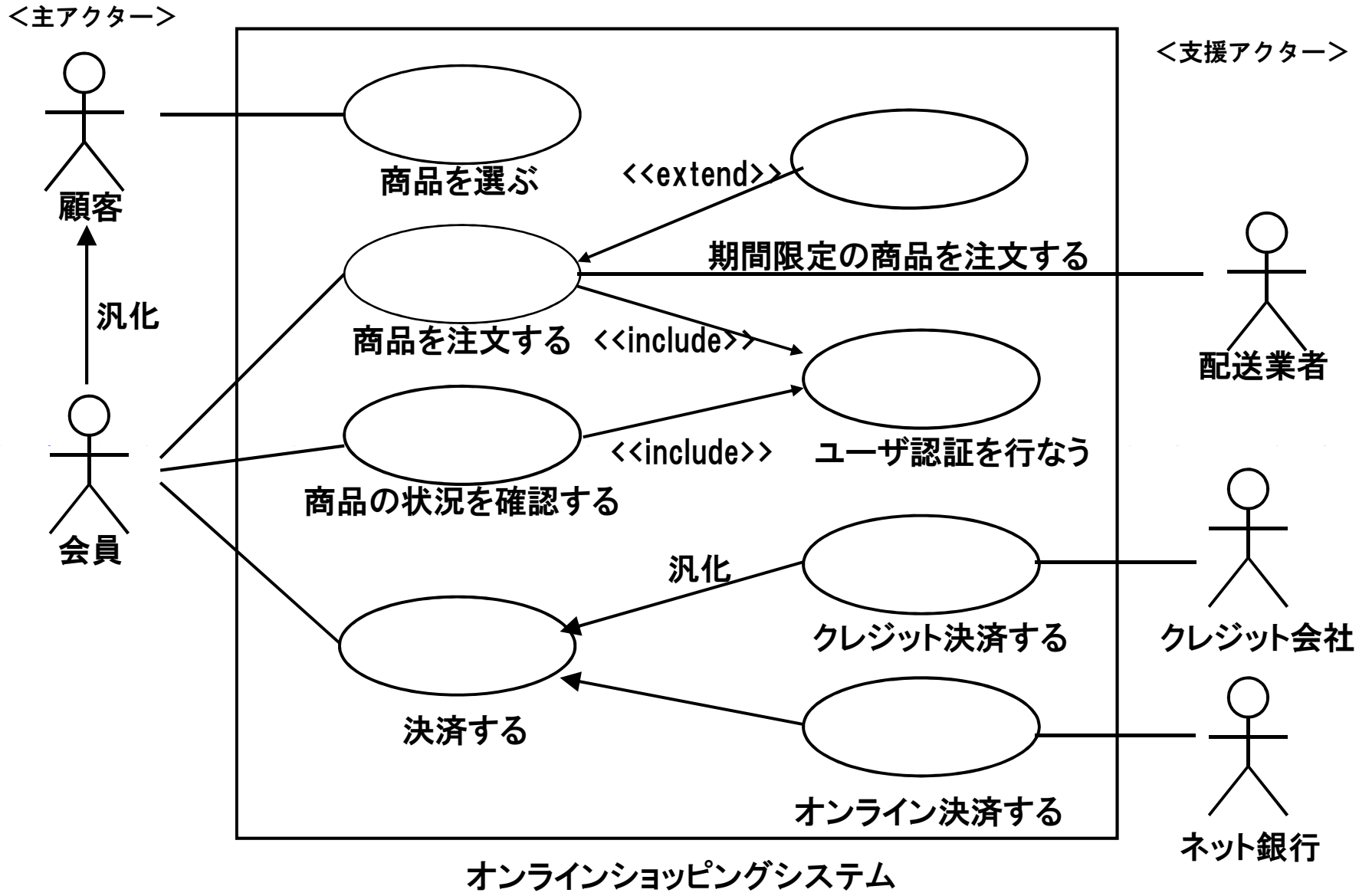
③エンティティの意味の明確化

開発（設計）手順に適切なツールを選択する。
（ツールに合わせて運用を変えない）

例）EA（Enterprise Architect）＋トレーサビリティ

システム運用分析～　要求仕様

ユースケース（例）オンラインショッピング



ユースケース記述

MITSUBISHI SPACE SOFTWARE

番号	1.1	名称	オンラインS/Wを起動する　（UC名称は、「xxをxxする」（名詞＋動詞）の表現にする）
概要		xxx。	
優先度	必須	根拠	（優先度の根拠を記載する）
事前条件		1.　xxxであること。	
成功事後条件		1.　xxxであること。 （基本系列、およびどの代替系列が終了しても成り立つ事後条件）	
必須事後条件		1.　PWR LEDが緑点灯していること。 （いかなる系列が終了しても成り立つ事後条件）	
主アクタ		ブートローダ （UCを起動するアクタを記載する）	
起動イベント		オンラインS/Wのエントリポイント呼び出し	
非機能要件		「xxx要求仕様書　4.2.2 性能目標」参照 （UCに固有の非機能要件は、本欄に記載する）	
基本系列		1.　ブートローダは、システム（オンラインS/W）のエントリポイントを呼び出す。 2.　システムは、表1.1-1の条件に基づきxxxを判定する。 （図表は外出しにしてよい） 3.　《xxxの場合》システムは、xxxを行う。 （条件と、それに伴う振る舞いは代替系列で表現しても良いし、ディシジョンマトリクスで記載しても良い。分岐が複雑な場合はアクティビティ図やフローチャート、状態遷移表で表現しても良い） 4.　システムは、xxx通知をxxxに送信する。	
代替系列3A		3a.《xxxでない場合》 3a1. システムはxxxを行う。	
例外系列3B		3b1. xxxに失敗した場合、xxxは、xxxを行いUCを終了する。 （代替系列、例外系列は、それが終わったら元の系列に戻るのが標準の動作であることに注意。処理を終える場合は、この例のように「UCを終了する。」と明確に記載すること） （以下のように書いても良い。 3b.《xxxに失敗した場合》 3b1. xxxは、xxxを行いUCを終了する。	
例外系列3A1A		3a1a. xxxに失敗した場合、システムはxxxを行う。	
備考		（備考欄には、システムの振る舞いに関係ない参考情報のみを記載することを原則とする。システムの振る舞いに関係する事項を記載する場合は、必ず系列から参照する形にする）	

MITSUBISHI SPACE SOFTWARE

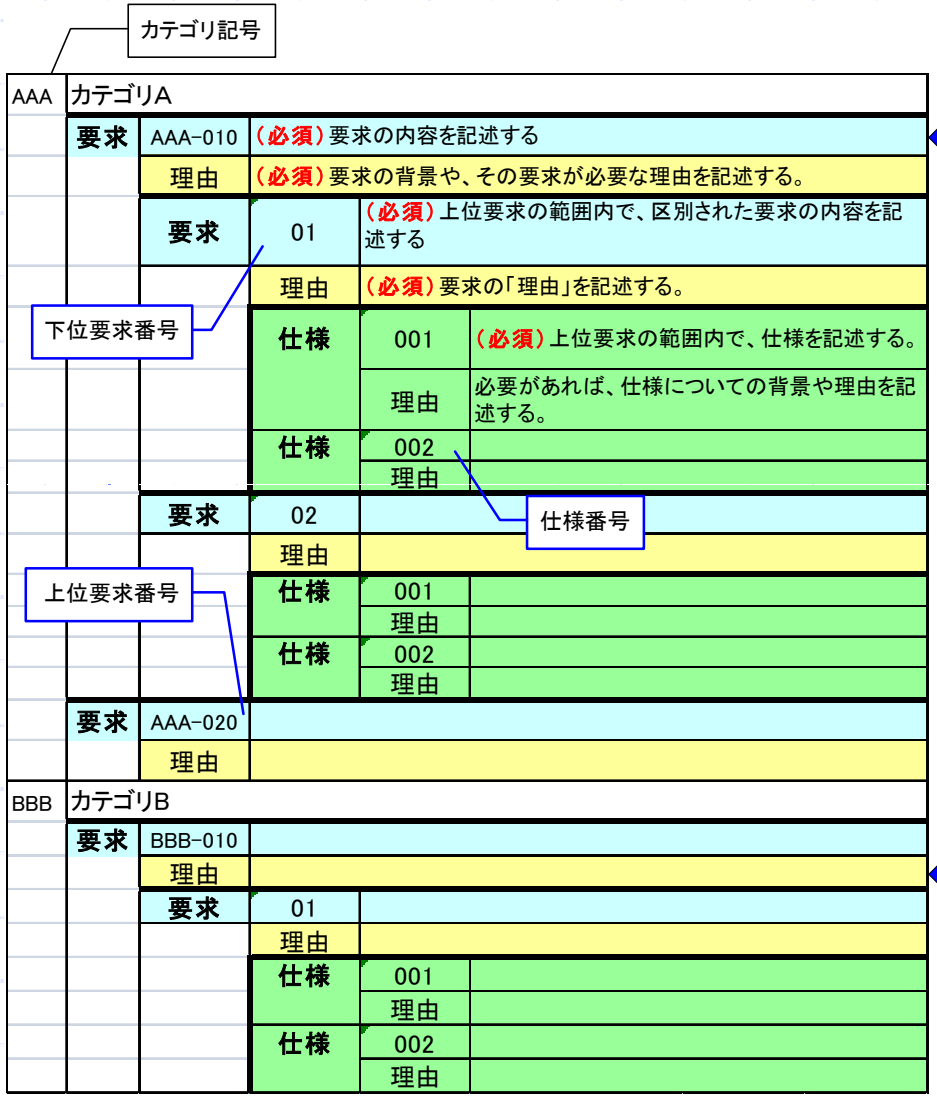
ユースケース記述だけでは仕様を拾いきれない！！

要求定義 （視点： システム外部）	◆「～がしたい」 利用者の希望（Require）。 ◆事業運用をビジネスレベルで考え、それを実現するコンピュータシステムへの要求。 機能要求は「システムの外側からみた振る舞い」で記述できる ＜ ユースケース記述は、要求を合理表現した要件と考える ＞	◆本を検索したい
仕様定義 （視点： システム内部）	◆「～が必要」 要求の目的を満足する機能や条件。 ◆システムが要求を満たすべき“具体的な振る舞い”を記述したもの。 ◆機能の程度やDB・通信などの利用方法。 仕様は要求に含まれる“動詞”と “目的語”である。 ＜ USDM と考えて良い ＞	◆素早く探す ◆あるテーマに関する本をできるだけ多く探す

USDM： 要件を仕様に展開する方法

MITSUBISHI SPACE SOFTWARE

「要求」は曖昧さを含んでおり、「要件」は形式的、合理的な表現になったものと定義する。要件は、ユースケース記述で表現できているとして、「要件」から始まり「階層構造」で「仕様」を記述していく。



ユースケース記述のタイトル（要求）を最上位要求として記述する。

■要求と仕様を区別すると、次の利点が得られる。

①客先から要求として出てきている内容が、「手段」であって、本来やりたいこと（＝要求）を必ずしも表現していないと分かることがある。

②仕様が曖昧な場合、”何故、そのような仕様が出てきているのか（＝理由）”を知りたくなる。
そして、理由を突き詰めると、その仕様では、要求を満足しなかったり、要求そのものが出し切れていないと分かることがある。

要求分析時の何故なぜ分析

＜重要＞要求には、「理由」がある。
その理由が記述できない時は、往々にして、要求と称して、仕様（手段）が書かれていたりする。この「理由」が、要求と仕様を仲介する役割を担う。

【注】USDM(Universal Specification Describing Manner)は、「清水吉男」が提唱した仕様書の表記法です。

ユースケース記述とUSDMの関係

- ユースケース全体をUSDMの**上位要求**に展開
- ユースケースの各ステップはUSDMの**下位要求**あるいはグループに展開
- ユースケースの各ステップの実現方式をUSDMの**仕様**に展開

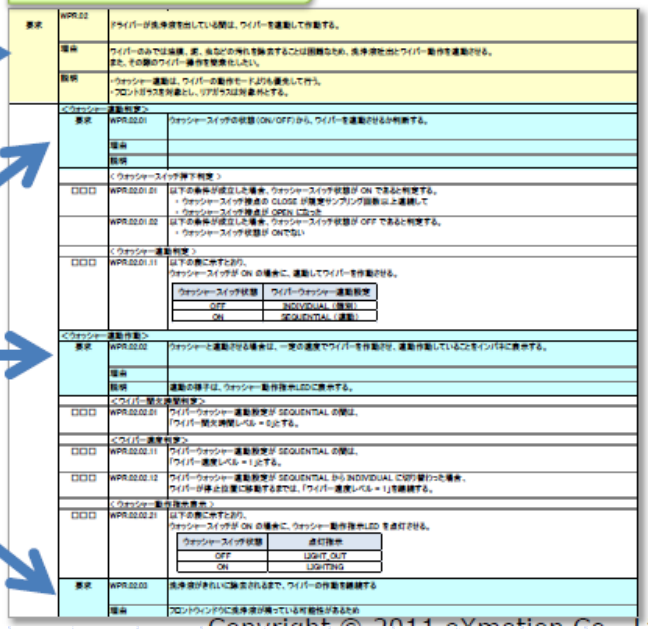
システムの外側 ⇐ ⇒ システムの内側

ユースケース記述

ユースケース番号		UC-WPR-02
ユースケース名		ウォッシャーと連動して、ワイパーを作動させる
概要		ドライバがウォッシャーを使ってフロントウィンドウを洗浄している場合、 洗浄液が吐出している間は、システムはワイパーを作動させる。 油膜、泥、虫を洗浄除去する
主アクタ		ドライバ
副アクタ		・ワイパーアクチュエータ ・洗浄液吐出ポンプモータ ・ウォッシャーSW（洗浄液吐出スイッチ、PushSW2） ・洗浄液吐出ポンプモータの動作を模擬するためのLED(LED1)
事前条件		ウォッシャーSW を操作
事後条件		ワイパーの作動が停止している
フロー	STEP	ステップ
基本フロー(B)	1	ドライバは、ウォッシャーSW を操作して、フロントウィンドウに洗浄液を吐出する。
	2	システムは、洗浄液が吐出している間、ワイパーを作動させる。 <A1><E1><*2>
	3	ドライバは、ウォッシャーSWの操作を止め、洗浄液の吐出を止める。
	4	システムは、洗浄液がきれいに除去されるまでワイパーの作動を継続する。
	5	システムは、ワイパーの作動を停止させる。 停止位置まで動作させ停止、または停止位置まで動作させUC01に復帰 （ユースケースの終了） <A2>

ユースケースは、”システムの外側”から見える振る舞いをアクターの視点で表したものである。

USDM



アクターは、〇〇を行う →

＜ 機能要求 ＞

← システムは、▲▲を行う

機能仕様 ＝ システム内部の振る舞い

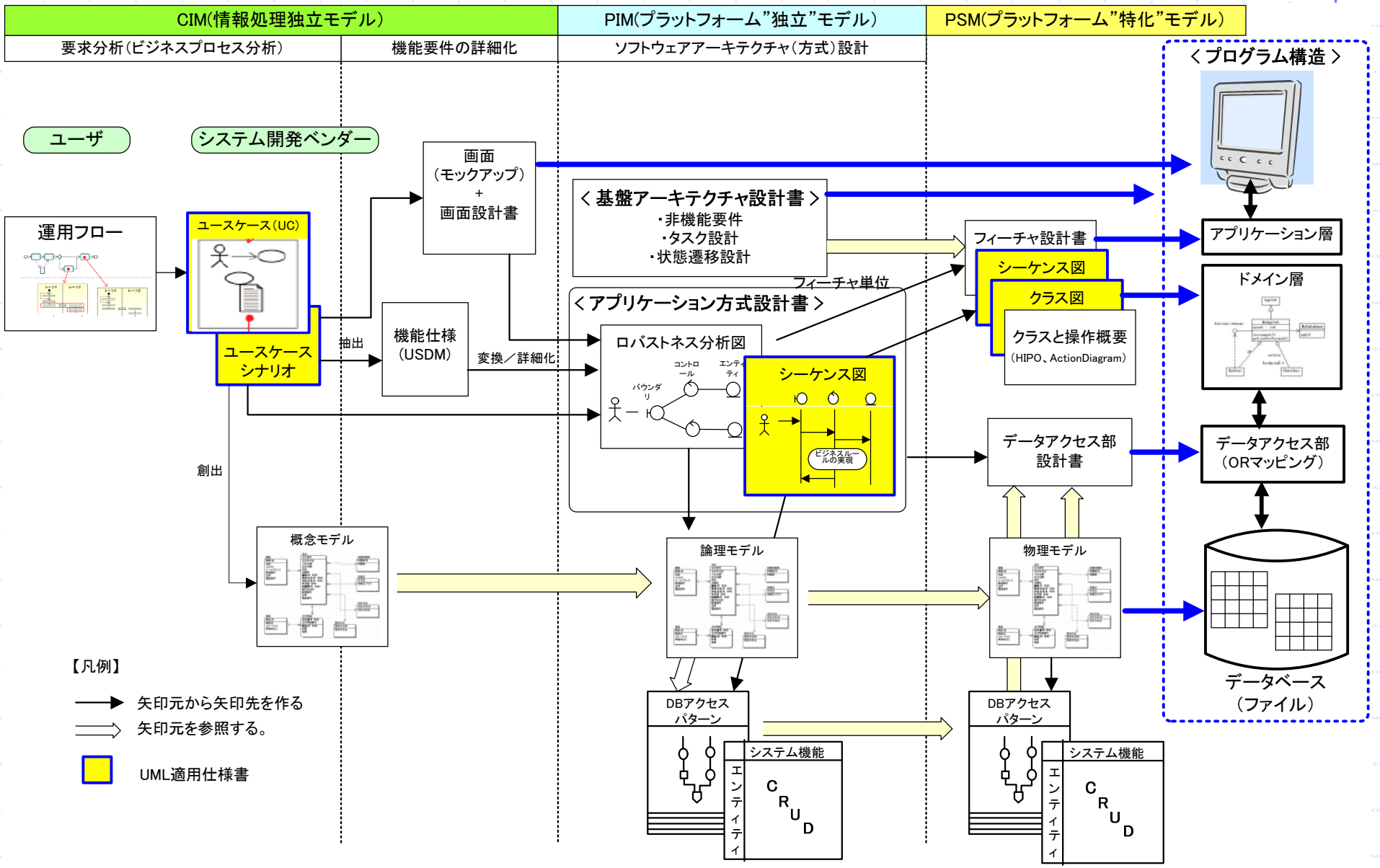
【注】(株)エクスモーションで実践している手法

ソフトウェア方式設計

ソフトウェア設計手順の概要

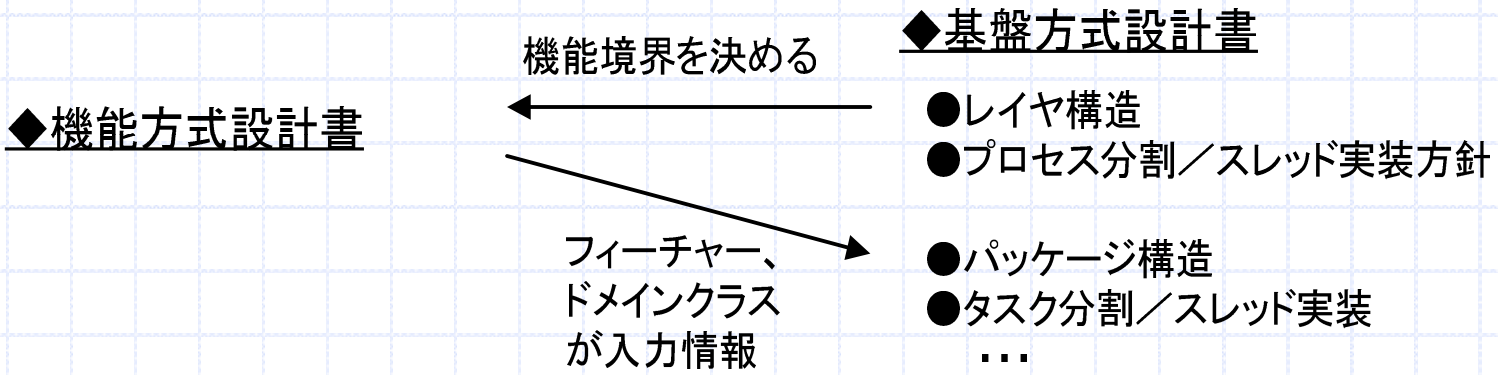
MITSUBISHI SPACE SOFTWARE

情報系・組込共通に利用できる、汎用的な設計方法論（ユースケース駆動）に基づいた開発手順書を作成した。



基盤方式設計と機能方式設計の関係

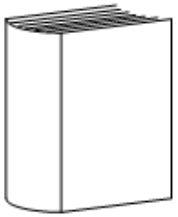
- ソフトウェア方式設計書は、大きく、
  『基盤方式設計』
  『機能方式設計』
  の２つに分ける。
- 『基盤方式設計』とは、S/W要求分析における「非機能要件」（性能、品質）を満たす観点から見たS/W構造を設計することと考えればよい。
- 『機能方式式設計』とは、S/W要求分析における「機能要件」を満たす構造を設計することと考えればよい。



基盤方式設計とは

MITSUBISHI SPACE SOFTWARE

基盤方式設計書



特に強い影響を受ける

要件

非機能要件　　機能要件

＋客観的な尺度

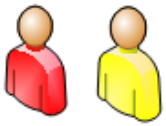
品質属性

可用性
変更容易性
性能
使いやすさ
テスト容易性
セキュリティ

ノウハウが蓄積された設計カタログ

もしくは

有識者

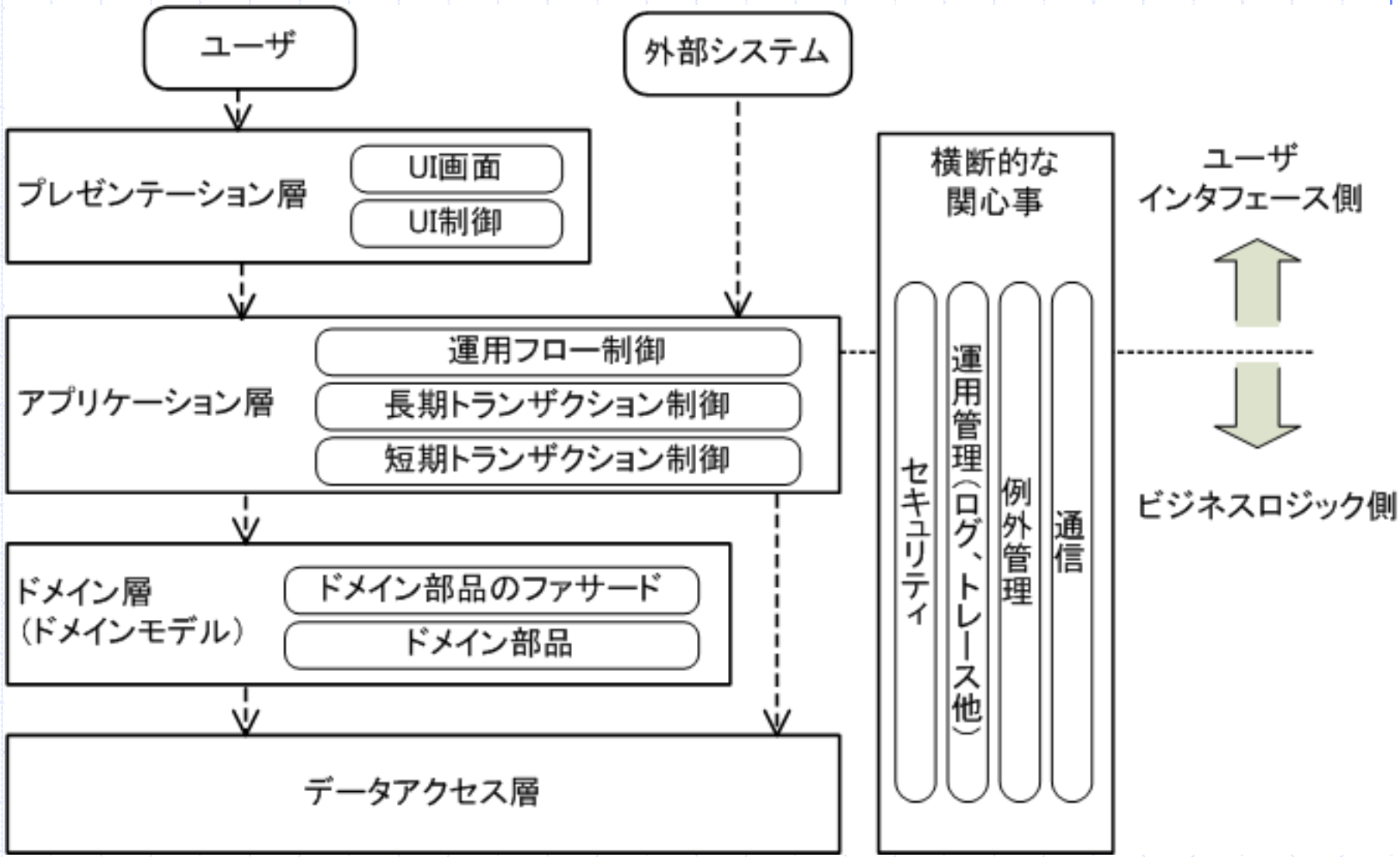


アーキテクチャ設計に要求される特性分類	基盤アーキテクチャの記述内容	要求(品質特性)					
		可用性	変更容易性	性能	セキュリティ	試験可能性	使いやすさ
●モジュール・ビュー（構造的特性） ソフトウェアの構造	システム構成(HW,SW構成)		●				
	レイヤ構造		●				
	レイヤ間インタフェース		●				
	利用フレームワーク		●				
	論理パッケージ構成		●				
●コンポーネント・コネクタービュー（動的特性） ソフトウェアが動作する時の仕組み	プロセス管理			●			
	メモリ管理			●			
	データ管理		●	●			
	状態遷移設計および実装		●	●			
	ネットワーク管理	●	●	●			
	トランザクション管理	●		●			
	国際化		●				●
	例外処理	●					
	障害管理	●					
	セキュリティ管理				●		
	ログ管理					●	
●割り当て(配置) どの部分が誰によって作られ、どのコンピュータに配置されるか	共通ライブラリ		●				
	設計上の留意点	●	●	●	●	●	●

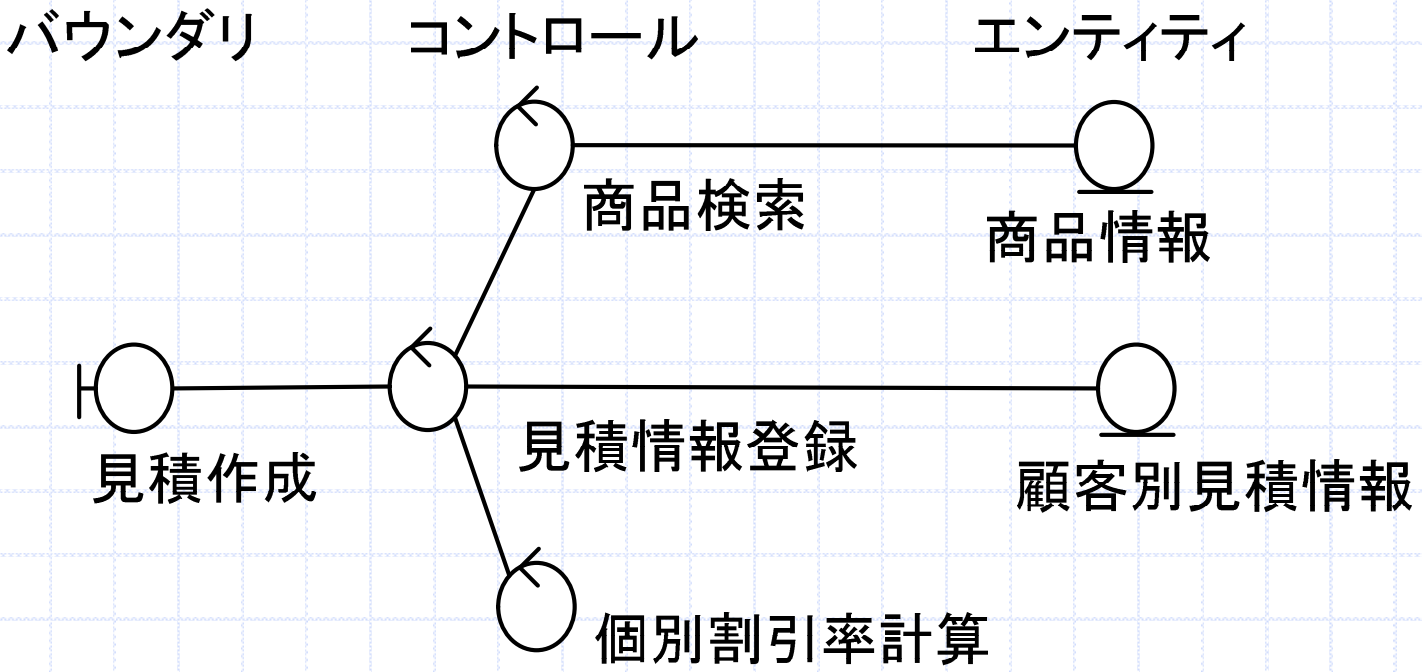
機能方式設計

MITSUBISHI SPACE SOFTWARE

基盤方式設計で決めたレイヤ構造ごとのクラス定義を実施するのが、機能方式設計である。
再利用性を高めるために、ドメイン層にドメインモデルを採用する。



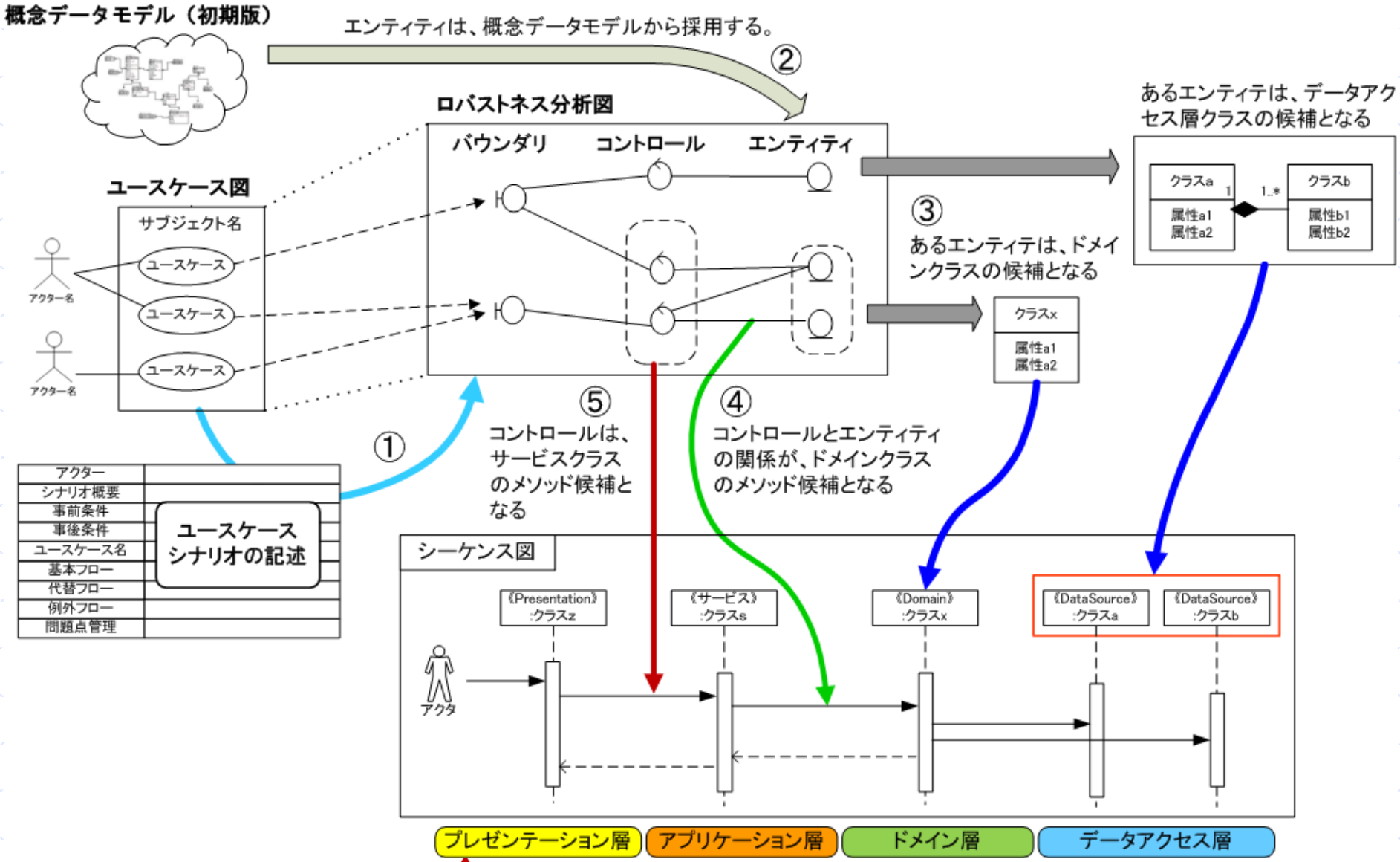
ロバストネス図はアプリケーションに要求される機能を、図形を使用して視覚的に表現するものである。ロバストネス図では、アプリケーションを、バウンダリ（画面、印刷）、コントロール（機能）、エンティティ（データ、ファイル）に分類し、それぞれの関係、振る舞い表現する。



ユースケース/ロバストネス/分析シーケンス図の関係

MITSUBISHI SPACE SOFTWARE

機能方式設計に「ロバストネス分析」を介在させることで、UMLだけは困難であった設計のトレーサビリティが取り易くなり、**設計効率が上がる。**



画面（操作）とサービス（フィーチャー）の関係

MITSUBISHI SPACE SOFTWARE

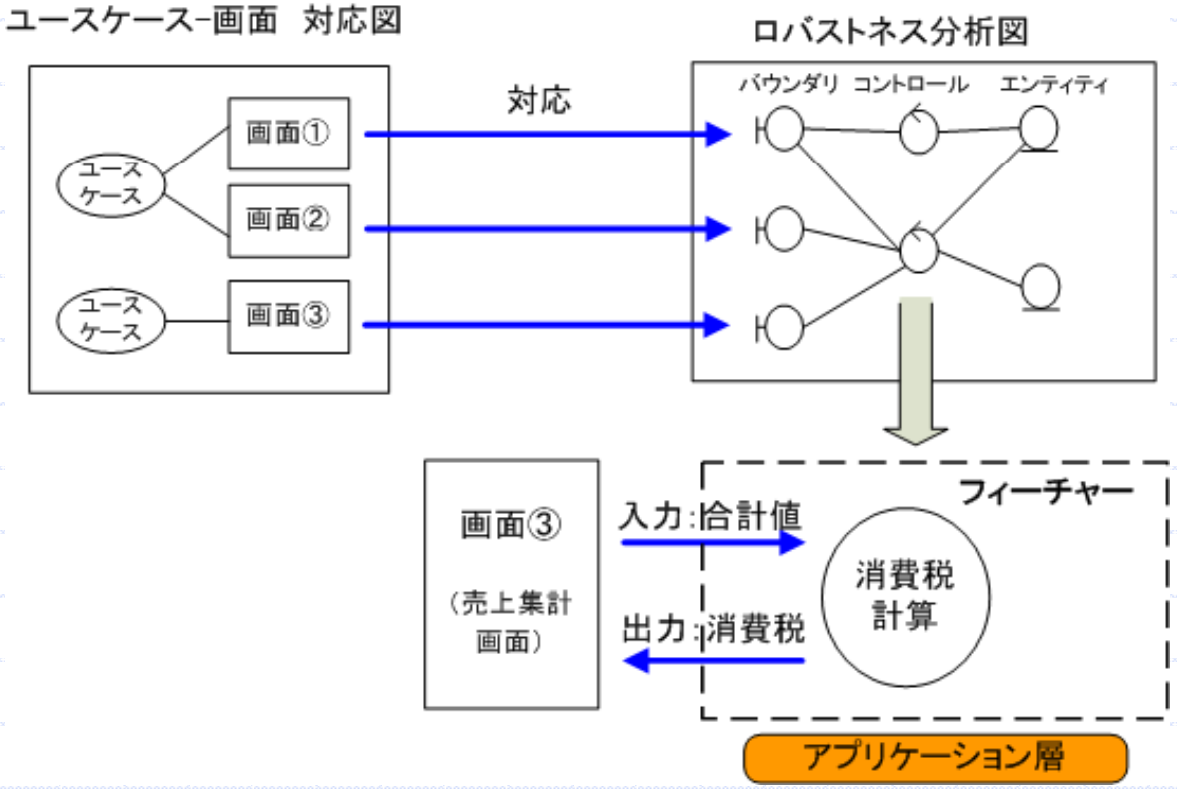
FDD(Feature Driven Development)におけるFeature（ユーザ機能）とは、「顧客にとって価値ある機能（＝サービス）」である（オリジナル定義）。

■フィーチャーの表現形式

＜オブジェクト＞＜結果＞＜アクション＞
販売　　合計　　計算

例

“販売”の“合計”を“計算”する
- 販売員の成果を査定する。
- スケジュールされた自動車整備を実施する。
- カード所有者のクレジットカード取引を認証する
- 消費税を計算する。

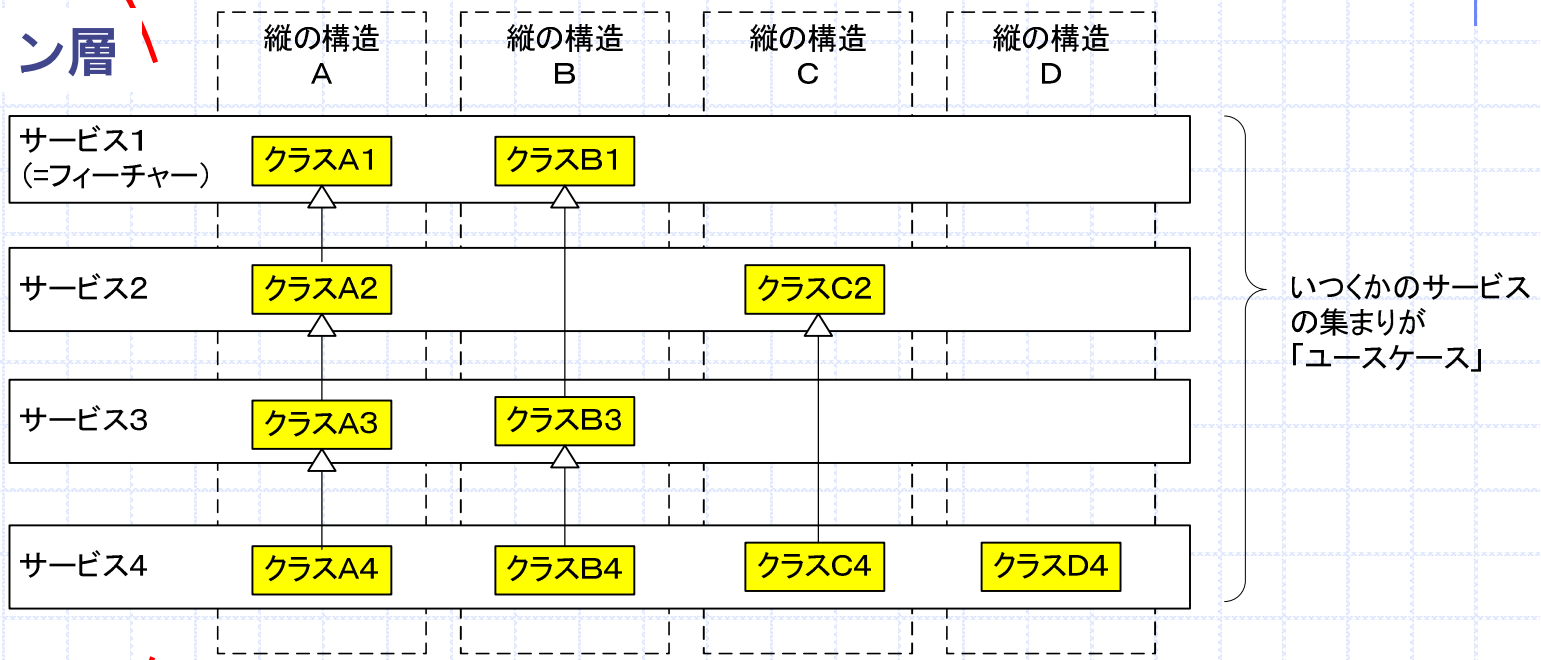


MITSUBISHI SPACE SOFTWARE

ドメイン層：オブジェクト指向

アプリケーション層

手続き型



ユースケース/USDM/ロバストネス分析図の関係

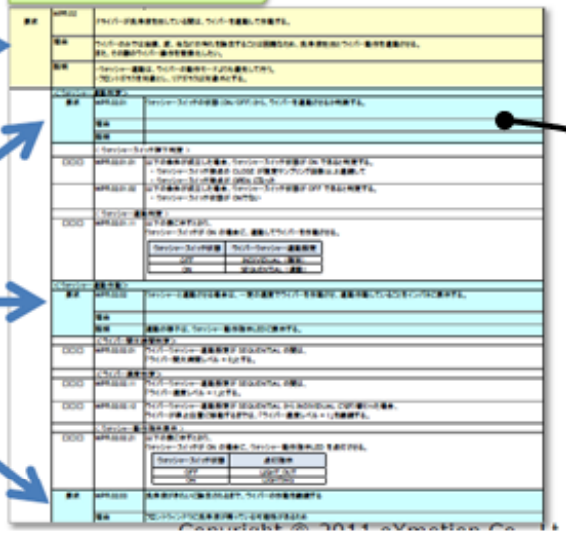
MITSUBISHI SPACE SOFTWARE

＜ユーザ要求仕様＞

ユースケース記述

＜ソフトウェア要求仕様＞

USDM

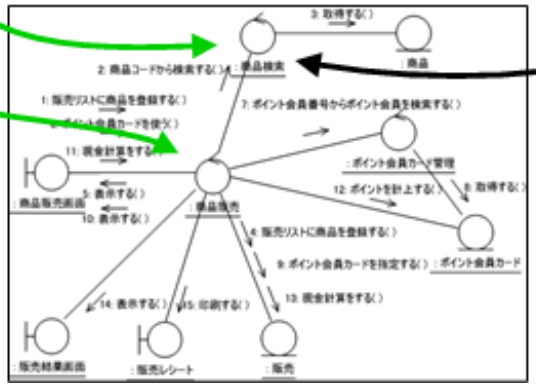


機能要求は、ロバストネス分析のオブジェクト(B/C/E)の仕様と考える。

ユースケースは、”システムの外側”から見える振る舞いをユーザの視点で表したものである。

機能要求 ＝ 「システムの振る舞い」部分

＜アプリケーション方式設計＞

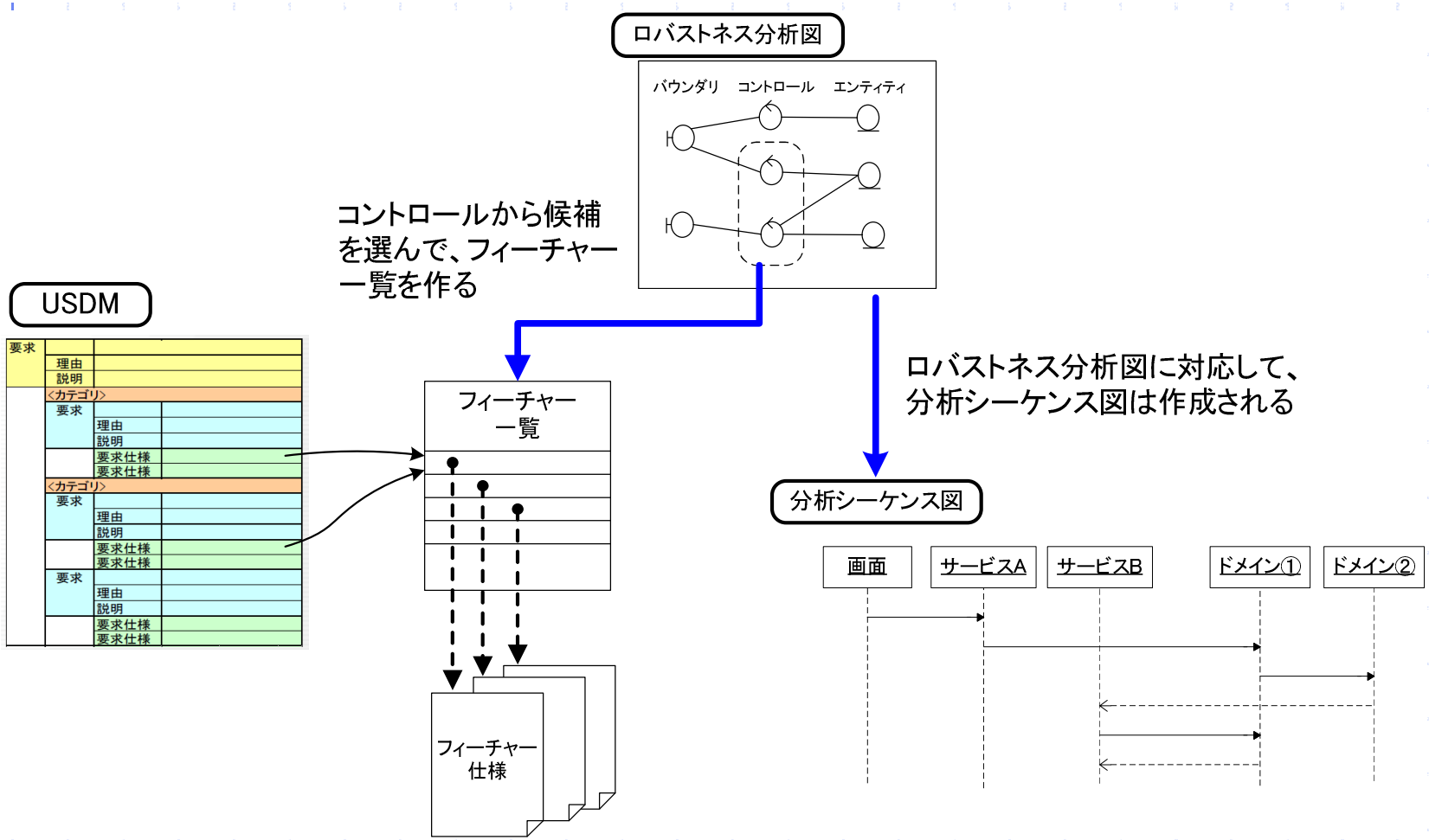


ソフトウェア詳細設計へ

要求仕様と方式設計（外部仕様）の記述のダブりを排除できている

仕様変更の影響箇所を特定する

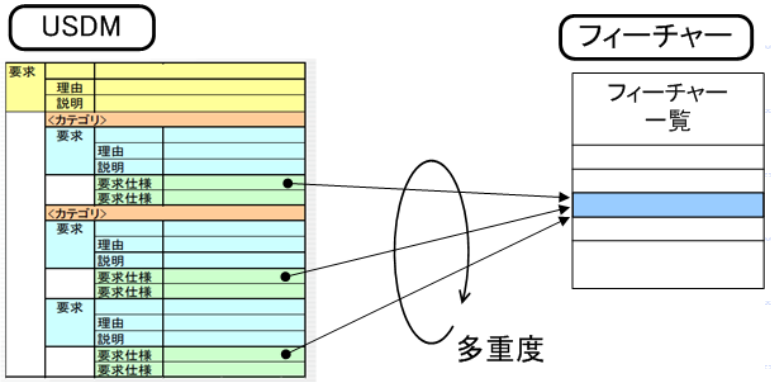
①USDMとフィーチャー間のトレーサビリティを利用して、仕様変更に対応するフィーチャー仕様の内容を確認する。
②フィーチャーの変更だけで対応できない場合、分析シーケンス図から、そのフィーチャーが利用しているドメインクラスの変更を考える。この時、変更を考えるドメインクラスを利用している他のフィーチャーをEAを利用して全て抽出し、影響が無いことを確認する。



設計検証：設計書間の多重度を活用して検証する

MITSUBISHI SPACE SOFTWARE

ソフトウェア開発において、上位から下位に詳細分解される成果物（例：設計書）間の対応関係を表現するのが、トレーサビリティ・マトリクス（TM）である。このTMの中で、多重度という指標を定義して設計検証に使う。ここで、多重度とは、ある成果物の項目が、他の成果物の項目といくつの関連を持つかを表現するものである。



ドメインと設計プロセスが同じであれば、多重度はある範囲に収まることが予想される。これを定量的に数値表現する「成果物間の対応比率定義表」を使って、TMに表現された内容から、その設計粒度を機械的にチェックすることで設計検証に使うことを考える。

◆成果物間の対応比率定義表（＝多重度）

	USDM仕様項目（数）			フィーチャー（数）			クラス（数）			プログラム規模（KL）		
	最小	最大	単位	最小	最大	単位	最小	最大	単位	最小	最大	単位
ユースケース（UC）	5.0	15.0	仕様/UC	3.0	6.0	FT/UC	8.0	20.0	クラス/UC	1.0	1.8	KL/UC
USDM仕様項目				3.0	5.6	仕様/FT	3.8	5.9	仕様/クラス	14.0	22.0	仕様/KL
フィーチャー（FT）							1.0	3.0	クラス/FT	88.0	178.0	Line/FT
クラス										10.0	18.0	クラス/KL
メソッド										15.0	25.0	Line/メソッド

設計検証の例

MITSUBISHI SPACE SOFTWARE

システム概要：　・クライアント／サーバ（画面数：95）
・データベースあり（ORマッピングツール利用）

項目	実績値	指標値
システム規模	117 KL	
業務フロー	8 数	
ユースケース	152 U.C	0．8 KL/U.C
		19 U.C/業務フロー
仕様(USDM)	1349 項目	11．5 仕様/KL
クラス	1334 クラス数	11．4 クラス/KL
フィーチャー数	387 クラス数	3．5 仕様/クラス
メソッド	8411 メソッド数	13．9 Line/メソッド

ご静聴ありがとうございました。