

USDM による要件抽出漏れゼロへの挑戦

～ USDM で乗り越える ユースケース記述の限界 ～

【発表内容】

- カーナビのソフトウェア
- 現状の要件定義
- ユースケース記述
- 問題点と原因の分析
- ユースケース記述の限界と課題
- USDMの導入
- 今後の取り組み

(株)デンソー ITS開発部
矢野 恵生

YOSHIO_YANO@denso.co.jp

自動車事業



インジェクタ

プラグ



ETC



エンジンECU



メーター



発電機



非自動車事業

環境機器・産業機器・情報機器



赤外線温熱機



自然冷媒
ヒートポンプ式給湯機



産業用ロボット



QRリーダー

開発対象製品



カーナビゲーション

カーナビゲーションのソフトウェアを開発

■ ソフトウェア規模の増加

ベースソフトの規模は
モデルごとに増加

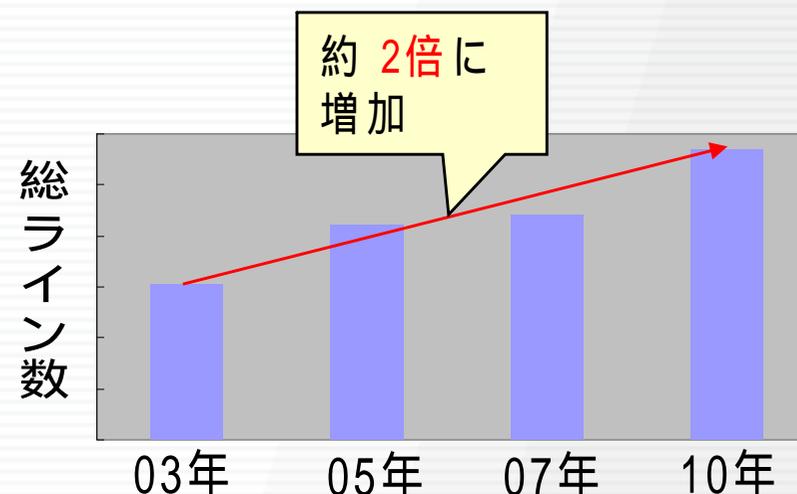
■ 要求機能の複雑化

- 複数機器の連携
- 複数モジュールの連携

[要求機能]	検索機能
[必要機器]	タッチパネル、マイク、 HDD、SDカード、...
[必要モジュール]	検索、音声認識、...

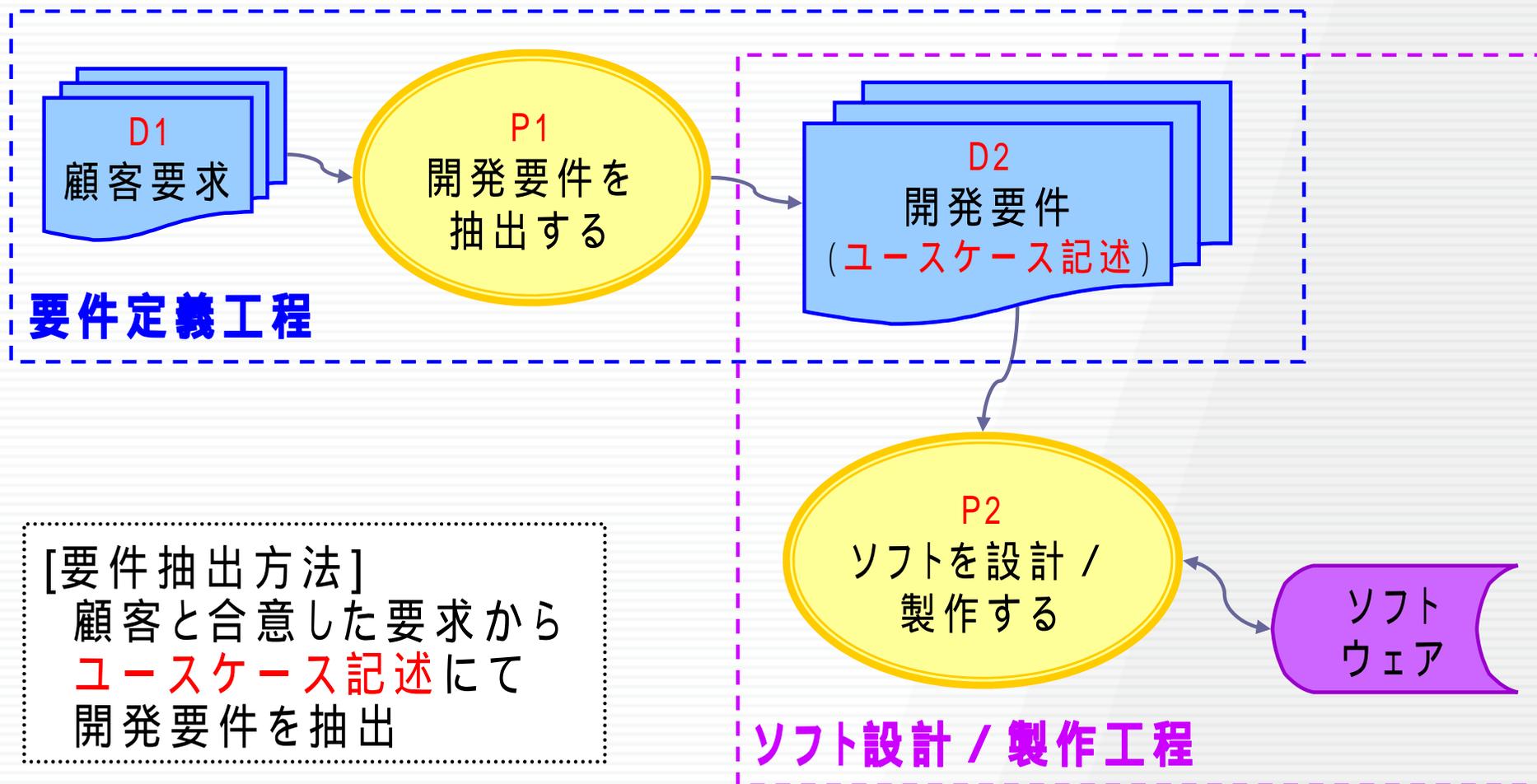
⇒ 機能の要件抽出にて考慮すべき対象が増大

開発要件の正確な抽出がますます困難に



■ 要件定義工程

顧客の要求から開発要件を抽出



UML2.0 をベースとしたフォーマット

ユースケース番号	AA-11	
ユースケース名	を する	
概要	の を、××に する。	
事前条件	なし	
事後条件	が××に されている。	
関連文書	ABCDE.doc	
フロー	STEP	イベント
基本フロー：B-1	1	主アクタが を要求した場合、本ユースケースは起動される。
	2	主アクタは、 の を、××に する。(A-1)
	3	本ユースケースを終了する。
代替フロー：A-1	1	が存在しない場合、本フローは起動される。
	2	システムはコーションを表示する。
	3	本ユースケースを終了する。
例外フロー：E-1	1	なし

[基本フロー] 最も**基本的**なユーザとシステムのやり取り

[代替フロー] 基本フローの**一部の処理を変更**する場合のフロー

[例外フロー] 処理の途中で**異常やエラー**が発生した場合のフロー

■ ユースケース記述とは

- ユーザ (関係者 / 関係システム) から見た
システムの外部機能を表現したもの
- 一連のバリエーションからなるシナリオ群をユーザの観点で
1つにまとめたもの

[シナリオ] ユーザがシステムとやり取りする一通りの筋書き

■ 記述例 (Web上でホテルを予約する)

主アクター : Webユーザ

事前条件 : なし

主成功シナリオ :

1. アクターは、宿泊したい最寄の駅を選択する。
2. システムは、選択された駅の周辺にあるホテルを表示する。
3. アクターはホテルを選ぶ。
4. …

拡張 :

- 2a. 選択された駅の周辺にホテルが存在しない :
 - 2a1. システムは、周辺にホテルが存在しないことを示すメッセージを表示する。
 - 2a2. …

■ 世の中に導入された経緯

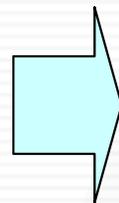
Ivar Jacobsonが1992年に、OOSE (Object Oriented Software Engineering) の中で仕様の記述法として導入

- ◆ プロジェクトにおいて道を踏み外さないためのガイドライン
- ◆ テスト計画のベースやオブジェクト指向分析 / 設計のインプット

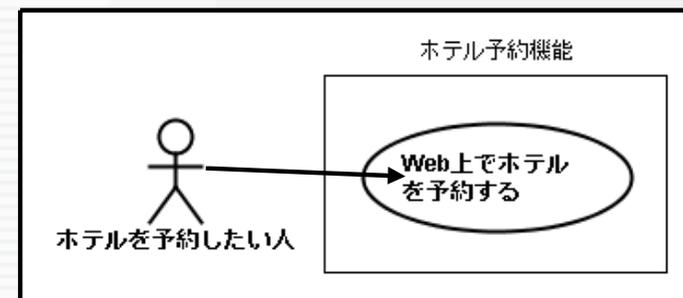
■ ユースケース図との関係

ユースケース図は、ユースケース記述をより直感的に理解するために導入されたもの

主アクター : Webユーザ
事前条件 : なし
主成功シナリオ :
1. アクターは、宿泊したい最寄の駅を選択する。
2. システムは、選択された駅の周辺にあるホテルを表示する。
3. アクターはホテルを選ぶ。
4. ...
拡張 :
2a. 選択された駅の周辺にホテルが存在しない :
2a1. システムは、周辺にホテルが存在しないことを...
2a2. ...



直感的に
表現



■ 要件抽出漏れの発生

基本機能の開発にて要件抽出漏れによる不具合 (2.3%) が発生

【不具合の発生した機能】

目的地を検索する機能

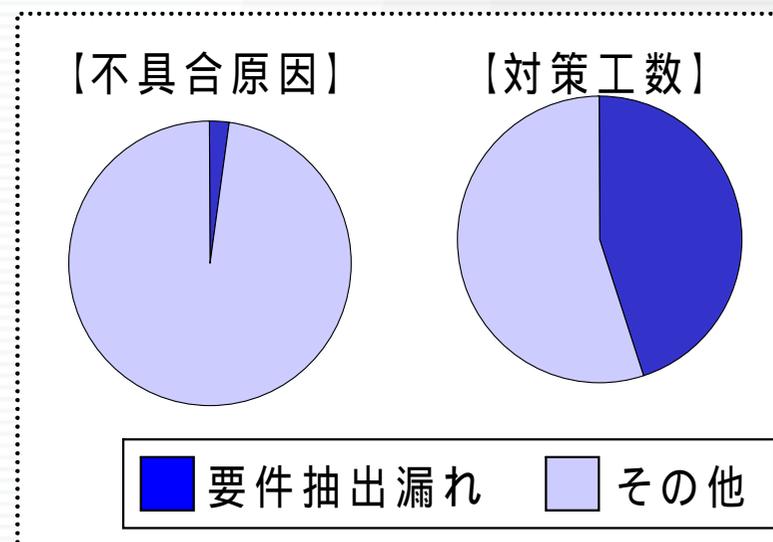
メモリ地点を管理する機能

■ 開発工数への影響

外部のテストにて発覚したため
手戻り工数大

- ◆ 全体設計のやり直し
- ◆ 影響範囲の再検討 etc

全対策工数の約45%を要した



要件漏れの原因解明 / 対策が必要

■ 漏れた要件の特徴

要件抽出の**対象ではない機能**から**影響を受ける振舞い**

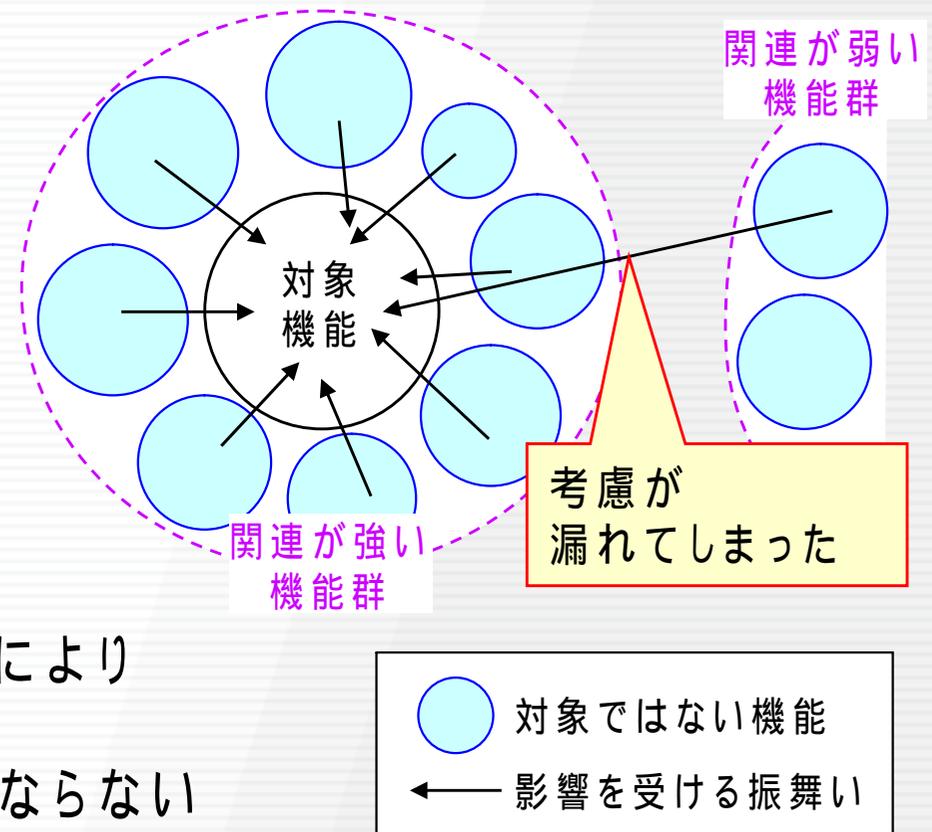
■ 要件例 (Req1)

● 対象機能

ユーザが入力した国名から、ナビ対象の国を切り替える機能

● 抽出が漏れた要件

システムの言語を切り替える機能により言語が切り替わった場合は、入力された国名を消去しなければならない

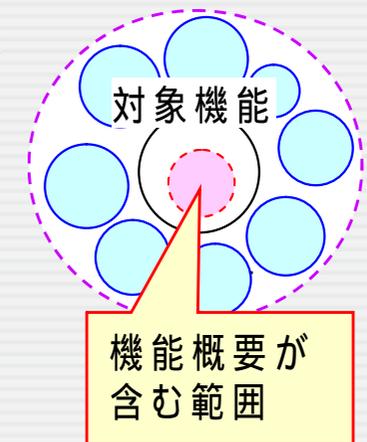


■ 問題の原因

- ユースケース記述では**機能概要から直接要件を抽出**

概要		の を、××に する。	機能の概要
事前条件		なし	
事後条件		が××に されている。	直接抽出
フロー	STEP	イベント	
基本フロー：B-1	1	主アクタが を要求した場合、本ユースケースは起動さ	
	2	・・・	
代替フロー：A-1	1	が存在しない場合、本フローは起動される。	
	2	・・・	
例外フロー：E-1	1	なし	

- 機能概要**には**対象機能の振舞いの一部**のみを記述
 - 対象ではない機能から受ける影響も含めた全ての振舞いを、1つの文章で記述するのは困難
 - 漏れた要件に繋がる記述は存在しない



■ 分析結果

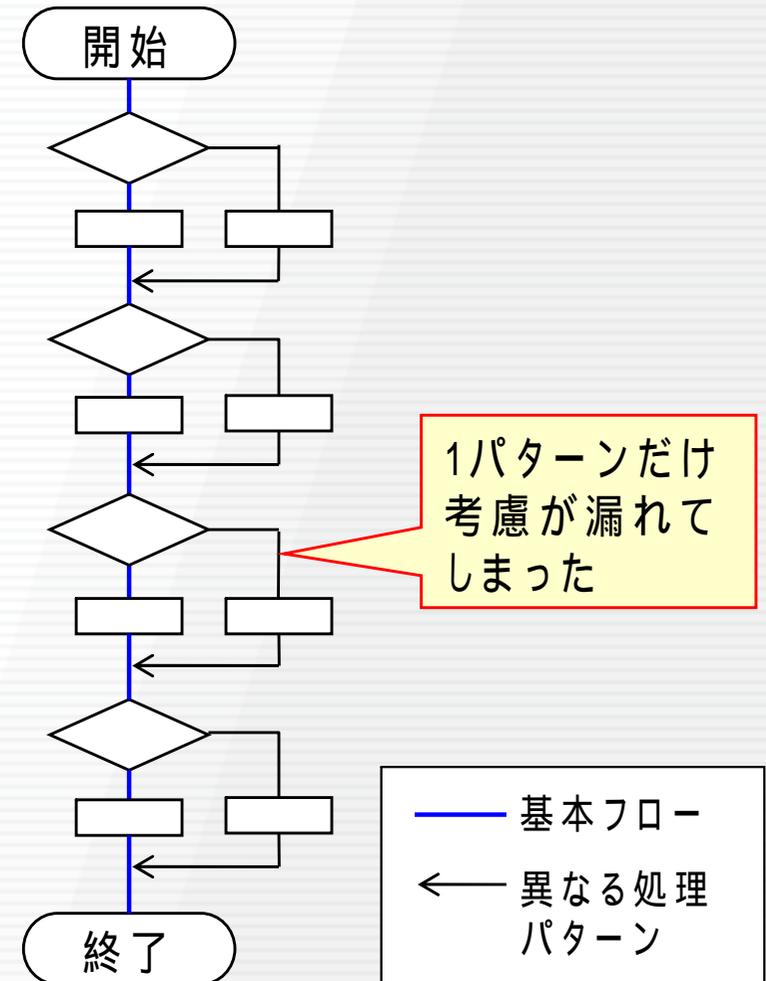
機能概要に記述されていない振舞いは漏れることが多い

■ 漏れた要件の特徴

基本フローとは異なる処理パターンにおける振舞い

■ 要件例 (Req2)

- 対象機能
入力された住所部品文字列
(国 / ストリート / シティ / ハウスNo)
から地点を検索する機能
- 基本フロー
国 / ストリート / シティ / ハウスNoが
全て入力された場合の振舞い
- 抽出が漏れた要件
シティが入力されない場合の振舞い

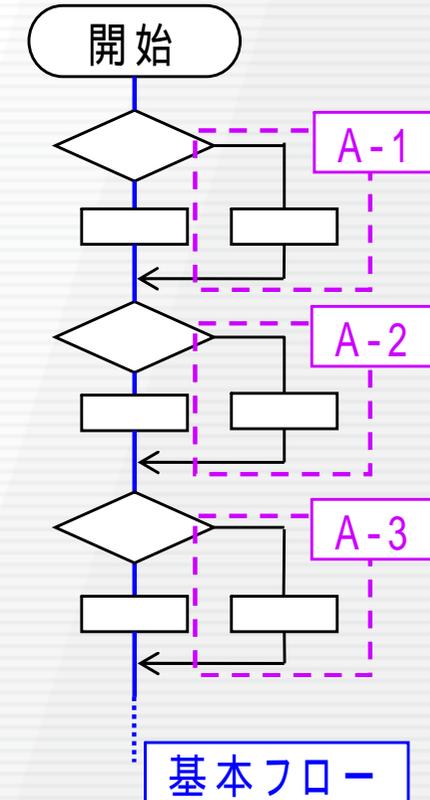


■ 問題の原因

ユースケースでは**全ての要件を並列 (同じ階層) に記述**

概要	の を、××に する。	
フロー	STEP	イベント
基本フロー : B-1	1	主アクタが を要求した場合、本ユースケースは起動さ
	2	...
代替フロー : A-1	1	が存在しない場合、本フローは起動される。
	2	...
代替フロー : A-2	1	が既に+++に存在する場合、本フローは起動される。
	2	...
代替フロー : A-3	1	が既に***に存在する場合、本フローは起動される。
	2	...
例外フロー : E-1	1	なし

- ➡ 結果、**網羅性の確認が困難**となりやすい
- 要件同士の関連が分からない
同じ観点で抽出したものなのか？
 - 同じ観点で抽出された要件がまとまっていない



■ 分析結果

基本フローと異なる処理が複数あると漏れる可能性がある

■ ユースケース記述の限界

- (1) **機能概要に記述されていない振舞い**を書ける場所が無い
機能概要に記述されていない振舞いは漏れやすい
- (2) 要件の**区分 / 整理**ができない
基本フローと異なる処理が複数あると漏れる可能性がある

■ 課題

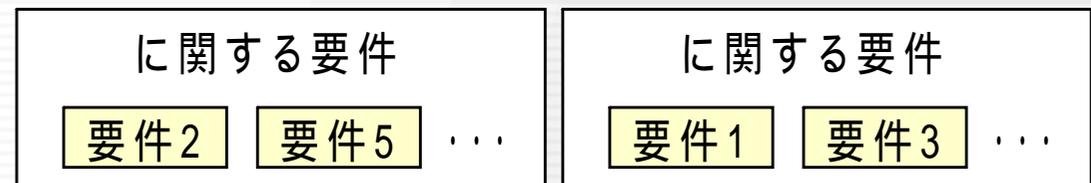
以下を満たす要件定義書の導入

- (1) 機能の**全ての振舞いを網羅**できる
- (2) 要件を**区分して整理 / 抽出**できる

【ユースケース記述】



【ありたい姿】



■ USDM (Universal Specification Describing Manner)

(株)システムクリエイツの清水吉男氏が提案した表記法

- ◆ 仕様の抽出を容易に、かつ抽出漏れを少なくすることを狙っている
- ◆ XDDPの中で作成する要求仕様書の表記法としても使用

■ 主な特徴

- (1) 要求と仕様を階層的に表現
- (2) 要求の理由を記述

本資料では
 「要件」と「仕様」は同じ概念を指す
 (弊社内部の用語としては「要件」
 が用いられているが、
 ここから先はUSDMにおける表現に
 合わせて「仕様」を用いる)

機能USDM		
要求	AA-11	
	理由	
	説明	
	要求	AA-11-1
	理由	
	説明	
	要求	AA-11-1-3
	理由	
	説明	
	要求	AA-11-2
	理由	
	説明	
		< * * * >
		AA-11-2-1
		AA-11-2-2
		< * * * >
		AA-11-2-11

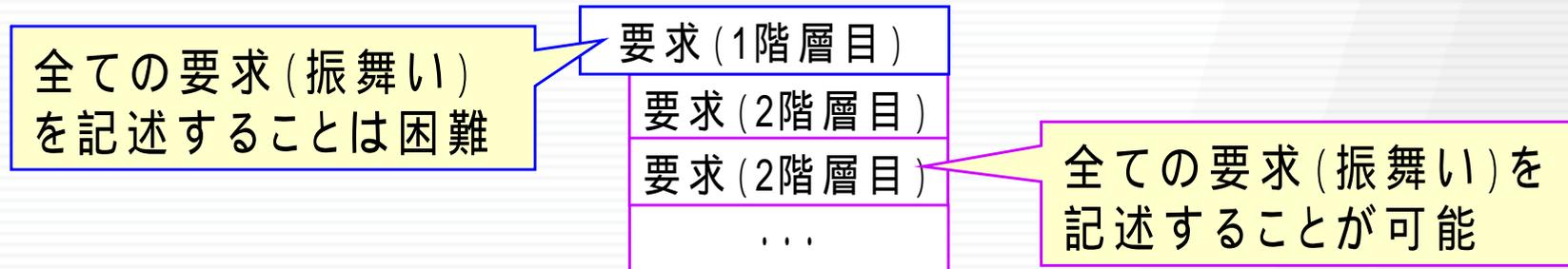
要求の背景である理由を記述

要求を階層的に表現

仕様を要求の下に記述

■ 要求と仕様を階層的に表現

- 2階層目の要求により、要求の全体を明確にすることが可能



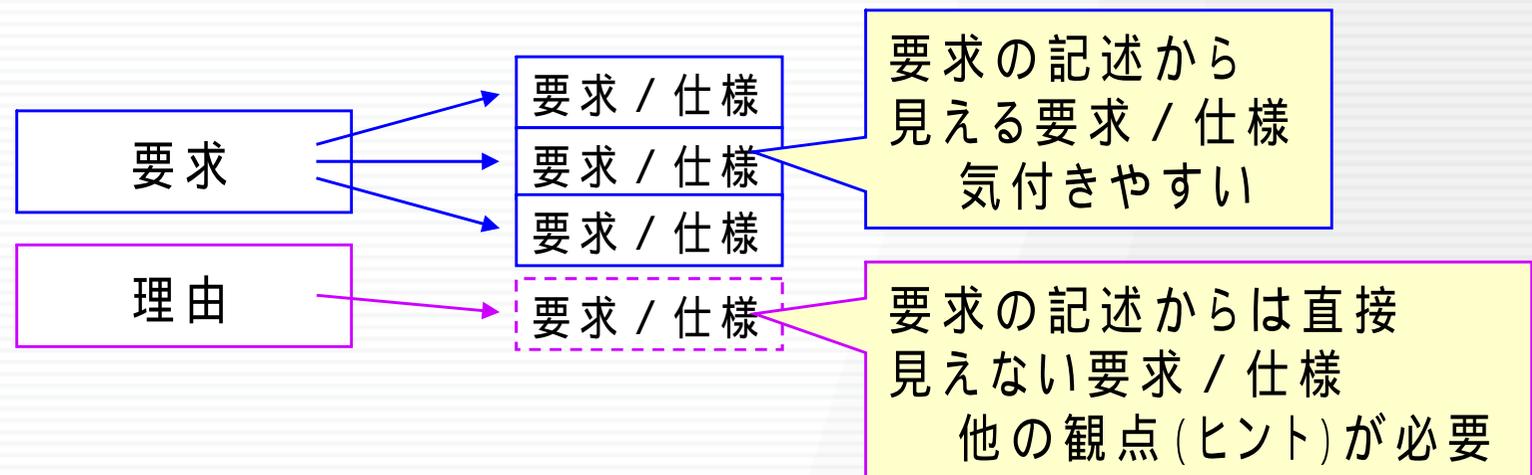
- 階層的に表現されることで、問題範囲の限定が可能
 - ◆ 仕様策定者による漏れを防止
 - ◆ DRによる漏れを防止

⇒ 階層化の有効性はマッキンゼー社のMECEにて証明済み

MECE : Mutually = 要素が互いに、Exclusive = 重複がなく、
Collectively = 集めると、Exhaustive = 全体を尽くす

■ 要求の理由を記述

- 下位層の要求 / 仕様の漏れに気付ける可能性が高まる
理由は要求の存在理由であり背景
⇒ 要求の記述には含まれていない観点が見えることがある



- 不要な要求の除去が可能
- 必要かつ有効な仕様の抽出が可能

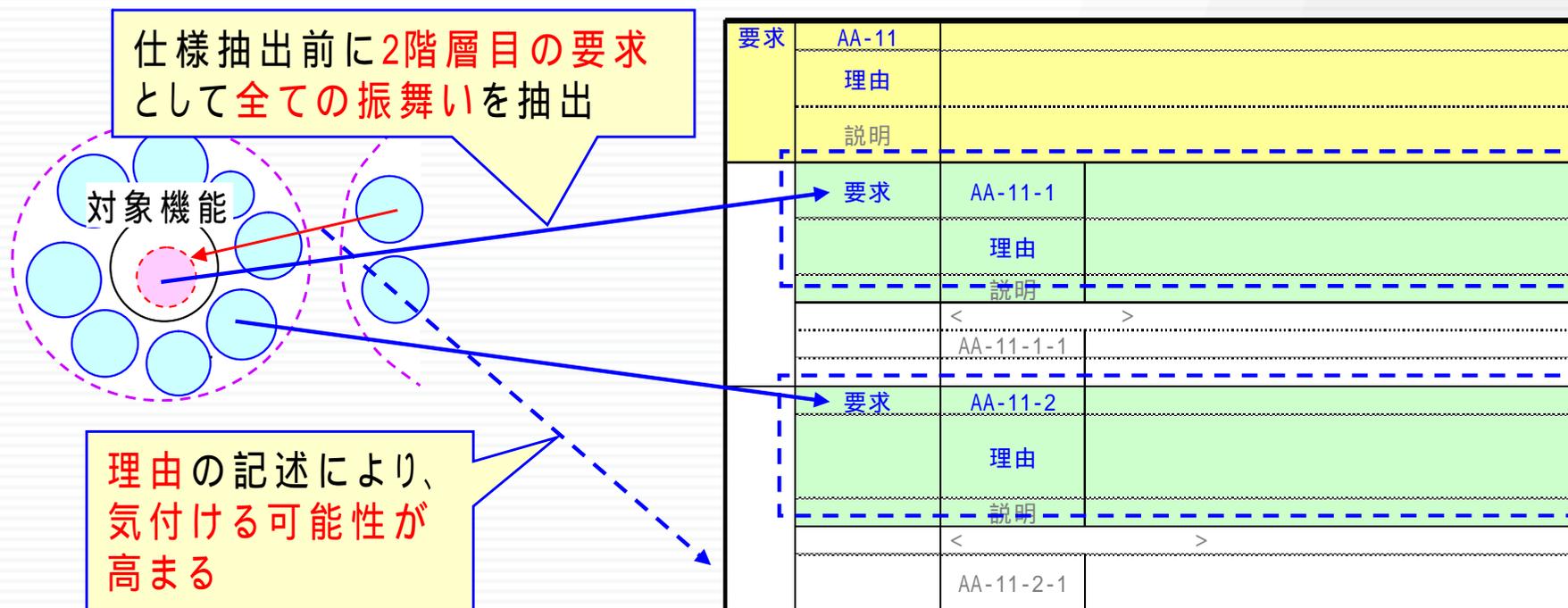
■ USDMによる課題解決

(1) 機能の**全ての振舞いを網羅**できるのか？

1階層目の要求 + 理由を元に、

2階層目の要求で全ての振舞いを明確にすることにより解決

- 作業者の目が機能全体に向きやすい
- 理由の記述が、関連の弱い機能からの影響に気付くきっかけとなる
- 文書として明示されていることで、DRでの気付きにつながる



■ 不具合への適用 - Req1 への対応 -

[対象機能]

ユーザが入力した国名から、
ナビ対象の国を切り替える機能

[抽出が漏れた要件]

システムの言語が切り替わった場合は、
入力された国名を消去しなければならない

● 第1階層目の要求 / 理由を記述

国切り替え機能USDМ		
要求	AA-11	ユーザが入力した国名から、 国に関するナビ対象の情報を切り替えて欲
	理由	国が変われば使用する言語が変わるので、 国に関する情報を入れ替える必要があるため。
	説明	
	要求	AA-11-1

仕様策定者(顧客)から
確認した**要求の背景**を
記述

⇒ 「システムの言語」、「国の情報」といった機能概要からは
見えなかった**観点**が理由の欄に記述できる

- 第2階層目の要求 / 理由を記述

国切り替え機能USDM			
要求	AA-11	ユーザが入力した国名から、 国に関するナビ対象の情報を切り替えて欲しい	
	理由	・国が変われば使用する言語が変わるので、 国に関する情報を入れ替える必要があるため。	
	説明		
要求	AA-11-1	ユーザが国名を にしてほしい。	より入力できるよう
	理由	・・・	
	説明	・・・	
		・・・	
要求	AA-11-2	AA-11-1 で入力した国名より、 ナビ対象の情報を切り替えてほしい。	
	理由	・・・	
	説明	・・・	
		・・・	
	(略)		
要求	AA-11-5	機能によりシステムの言語が変更された場合は、 AA-11-1 で入力した国名を消去してほしい。	
	理由	・・・	
	説明	・・・	
		・・・	

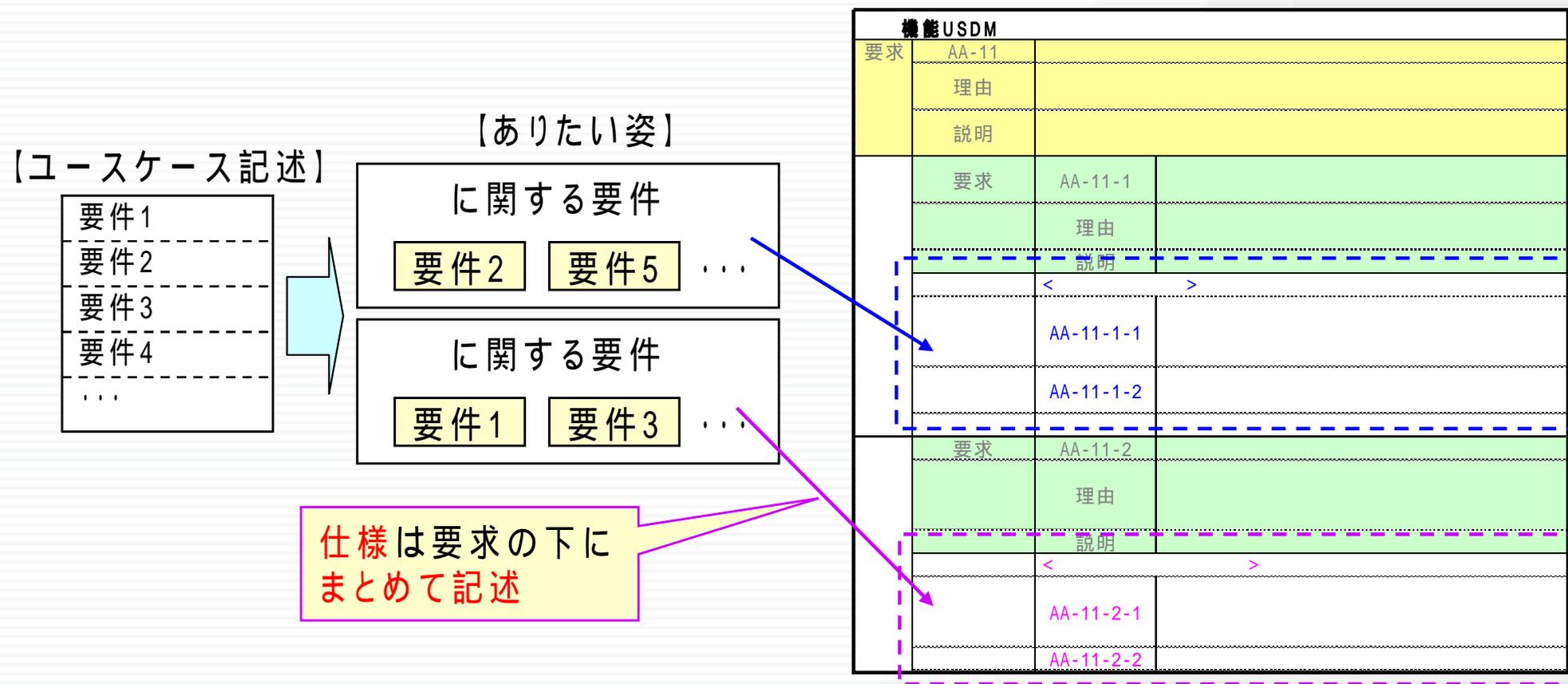
2階層目の要求として
全ての振舞いを明確化

1階層目の理由に
記述された
「システムの言語」という
観点から、
抽出が漏れた仕様を
発見するための振舞いに
気付くことができる

■ USDMによる課題解決

(2) 要件を区分して整理 / 抽出できるのか？

仕様が2階層目の要求ごとに記述されるため
自動的に満たされる



■ 不具合への適用 - Req2 への対応 -

[対象機能]

入力された住所部品文字列から
地点を検索する機能

[抽出が漏れた要件]

シティが入力されない場合の振舞い

- 第1、2階層目の要求 / 理由を記述

住所部品文字列からの地点検索機能USDM			
要求	BB-11	国、ストリート、シティ、ハウスNoを入力し、 地点検索してほしい。	
	理由 説明	...	
	要求	BB-11-1	ユーザが住所部品を より入力できるようにしてほしい。
	理由 説明	...	
	要求	BB-11-5	ユーザが住所部品の入力をキャンセルした場合は、 してほしい。
	理由 説明	...	

2階層目の要求として
全ての振舞いを明確化

- 要求の下の仕様を抽出

住所部品文字列からの地点検索機能USDM			
要求	BB-11	国、ストリート、シティ、ハウスNoを入力し、地点検索してほしい。	
	理由説明	...	
要求	BB-11-1	ユーザが住所部品をより入力できるようにしてほしい。	
	理由説明	...	
要求	BB-11-5	ユーザが住所部品の入力をキャンセルした場合は、 してほしい。	
	理由説明	...	
要求	BB-11-5-1	ユーザが国入力をキャンセルした場合は、 してほしい。	
	理由説明	...	
要求	BB-11-5-2	ユーザがシティ入力をキャンセルした場合は、 してほしい。	
	理由説明	...	

住所部品が
入力されない場合に
限定して仕様を抽出
することで、
全ての仕様を抽出
することができる

■ その他の不具合の検討

(1) 機能の全ての振舞いを網羅できることにより対応可能

不具合内容	適用結果
実行できないはずのシーンで実行できてしまう	2階層目の要求として、「各シーンでの実行可否」に関する振舞いを挙げることで抽出漏れが発生しないことを確認
特定の状態において、出力結果がおかしい	2階層目の要求として、「現在の状態」に関する振舞いを挙げることで抽出漏れが発生しないことを確認

(2) 要件を区分して整理 / 抽出できることにより対応可能

不具合内容	適用結果
特定のエラー発生時に、エラー画面に遷移しない	2階層目の要求として「エラー発生」に関する振舞いを挙げることで、全エラーについて仕様が抽出できていることを確認

過去の不具合について有効性を確認

■ 今回の発表のまとめ

- ユースケース記述の構成上の問題について検討
 - ◆ 機能概要に記述されていない振舞いは漏れやすい
 - ◆ 基本フローと異なる処理が複数あると漏れる可能性がある
- 別手法としてUSDMを採用
- 過去の不具合についてUSDMの有効性を確認

■ 現行開発への展開状況

- 一部の開発にて試行中
 - ◆ 次世代ナビの操作画面開発
 - ◆ 社内のWebシステム開発

■ 定量データの測定

現在以下のデータを測定中

抽出漏れ要件の発生件数

要件抽出工程の必要工数

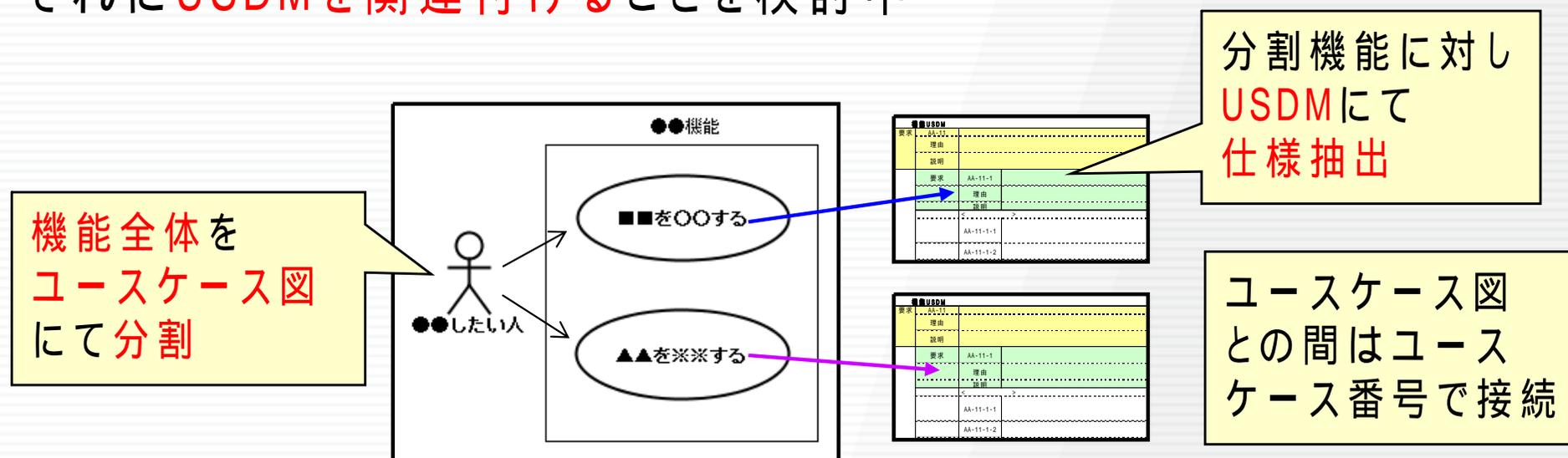
抽出可能な要件数 (要件の展開力)

全工程の開発工数

■ ユースケース図との統合

全体の機能分割をユースケース図で実施、

それにUSDMを関連付けることを検討中



派生開発プロセス (XDDP) と要求仕様の書き方 (USDM)
((株)デンソー 研修資料、作 : 清水吉男)

「派生開発」を成功させるプロセス改善の技術と極意
(著 : 清水吉男)

ユースケース実践ガイド - 効果的なユースケースの書き方 -
(著 : アリスター・コーバーン)

組込みソフトウェア開発のためのオブジェクト指向モデリング
(著 : SESSME WG2)

連載 : 【改訂版】初歩のUML 第8回
<http://www.atmarkit.co.jp/im/carc/serial/renew_uml08/renew_uml08.html>
(株式会社豆蔵)

USDMを活用した要件定義の改善 ~ ユースケース記述からUSDMへ ~
(SPI Japan2010、作 : 矢野恵生)