

そうだったのか！ USDMM

AFFORDD カンファレンス2015
チュートリアル資料

梶本 和博

派生開発推進協議会

URL=<http://www.xddp.jp>

(株) エクスモーション

URL:<http://www.exmotion.co.jp>

kazuhiro.kajimoto@exmotion.co.jp

- USDMって知ってますよね、使っていますか？
- 上手く表現できていますか。
- 要求は、動作がイメージできる振る舞いで書けていますか
- 理由は、要求の裏付けとなっていますか
- 仕様は、要求を満足させることが出来る内容ですか
- 理由・要求・仕様は、適切に書けていますか。

梶本 和博（かじもと かずひろ）

<所属> : 派生開発推進協議会

株式会社エクスマーション

<業務> : ソフトウェアプロセス改善コンサルタント

<略歴> :

- ▶ 1975: 制御系ソフト設計の世界へ
オンライン銀行ターミナル/オフスコンピュータ
- ▶ 1984: 組み込みソフト設計に
ページプリンタ/スケーラブルフォント
- ▶ 1989: プリンタ商品企画に
オンデマンドパブリッシング向けプリンタ
- ▶ 1995: ソフトウェア品質/生産技術の世界へ
USD M / X D D P / P F D / S I O 9 0 0 0 / C M M / C M M I
- ▶ 2010: 派生開発協議会
- ▶ 現在: (株)エクスマーション
USD M / X D D P / P F D

1. USDMのおさらい
2. 上手く表現するために
3. こんな場合には
4. この表現は如何ですか

1. USDMのおさらい
2. 上手く表現するために
3. こんな場合には
4. この表現は如何ですか

1. 1 USDMとは

1. 2 USDMの特徴

➤ Universal Specification Describing Manner

“万人に通じる仕様を描写する作法”

❖ 根幹となる考え方

- ❖ 「要求」には「仕様」を導き出す重要な役割がある
- ❖ 仕様は要求に含まれる“動詞”及びその“目的語”にある
- ❖ 考案した人
 - ❖ (株)システムクリエイツ代表取締役・派生開発推進協議会代表

清水吉男氏

➤ USDMは表現に軸足を置いた表記作法

❖ 要求開発には3つの要素がある

USDMのカバー範囲



- ❖ 最近の要求開発の研究は、「要求の発見・抽出」に力点が置かれている
 - 「要求の発見」と「仕様の発見」を区別していない？
- ❖ USDMは「表現」に重点を置き、そこから他の要素をカバーする表記作法
 - せっかく発見された要求も、適切に表現されなければ仕様が漏れる
 - 良い表現は思考を助ける

➤ 「要求」と「仕様」を区別しセットで記述する

❖ 一般には、「要求」と「仕様」（「要件」）は文書が分けられている

❖ USDMでは「要求」と「仕様」を表現上で使い分ける

❖ 要求:「実現してほしいこと」であり、ゴール(・目的)に相当

❖ 機能に関する「要求」は“振る舞い”で表現し、「仕様」を引き出す役割を担う

❖ 具体的なものから、抽象度の高いものまでさまざま

❖ 仕様:「作るべきもの」に対する具体的な記述(具体的な動きや制限)

❖ 「仕様」は要求の中で存在するもので、すべて最終的にソースコード(実行文/定義文)に変換されるもの

❖ 具体的な処理フローや入出力の制限や動作に関わる条件など

❖ 仕様は、関係者間で“Specify(特定)”できるレベルにまで詳細化されていなければならない

◆ 依頼者は「ソフトウェアの出来上がりのイメージ」を特定できる

◆ 設計者は「何を作るべきか」を特定できる

◆ 検証者は「何をどう確認すべきか」を特定できる

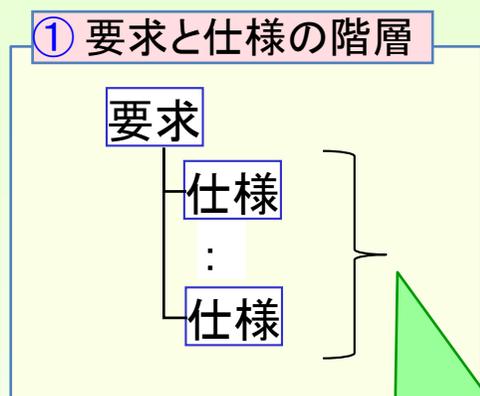
1.2 USDMの特徴

➤ 要求と仕様を階層構造で表現する

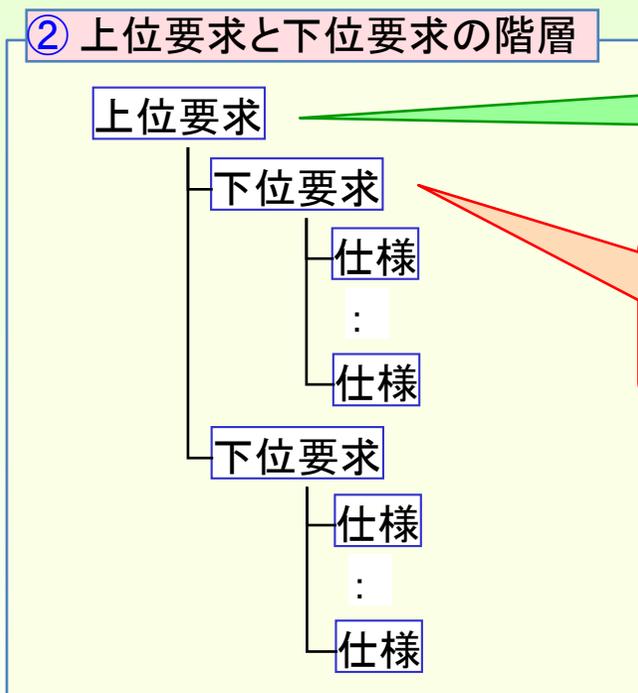
❖ USDMでは、「**範囲**」や「**階層化**」という考え方を持ち込む

❖ 要求と仕様を**2種類の階層構造**の中で捉える

❖ 仕様への論理展開がわかりやすくなる



下位の仕様を満たすことで要求を満たしたことになる



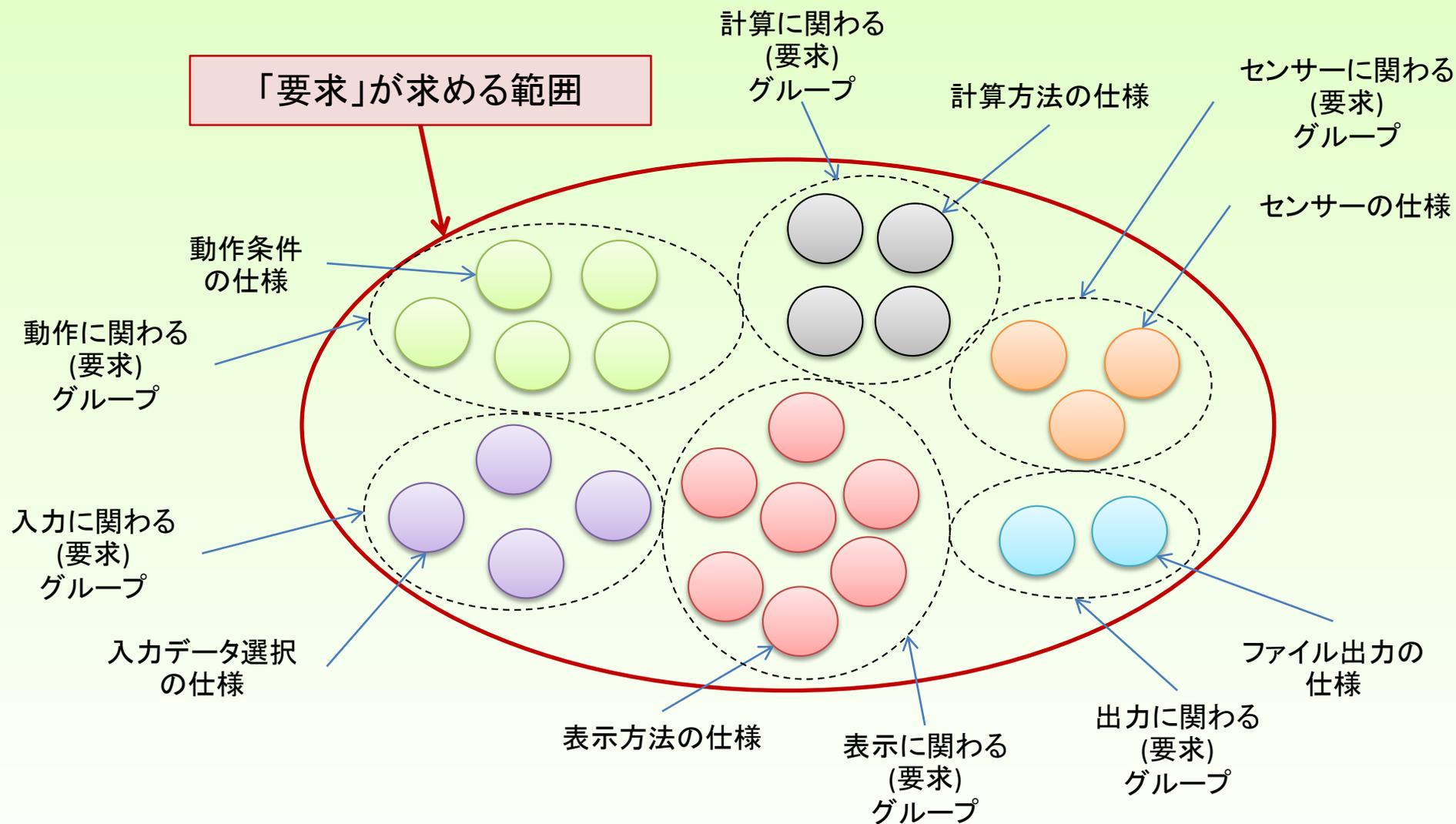
要求の範囲が広いときの対応

上位の要求は「範囲」と「振る舞いの全貌」を示す

要求の範囲を狭めることで、そこに含まれる「動詞」をすべて表現することを目指す

階層構造 = 仕様がモレにくい

➤ 要求と仕様の関係



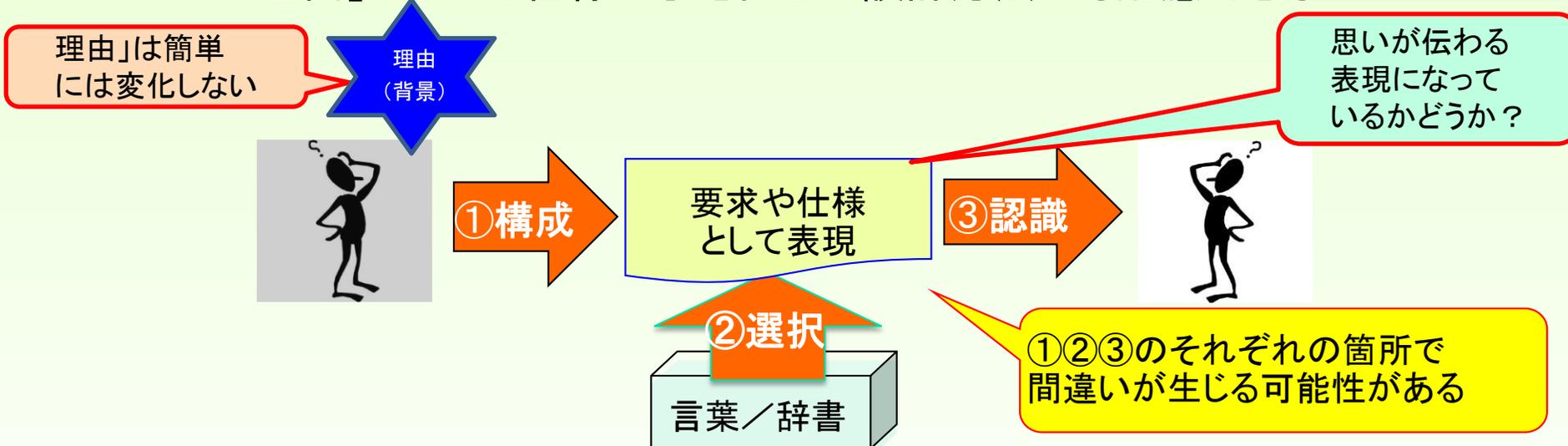
1.2 USDMの特徴

➤ 要求には必ず「理由」を付けて要求を安定させる

❖ 自分の思い(要求)が相手に正しく伝わるとは限らない

この状態を解消する方法として、USDMでは要求には必ず「理由」を付ける

- ❖ 「理由」があることで、「要求」の意味を理解しやすい
- ❖ 「理由」を書くことで、その要求に根拠がないことに気付くこともある
- ❖ 「理由」によって仕様の引き出しや設計方法にも配慮できる



「理由」の方が安定している
「理由」を捉えていれば要求は変化しにくい

▶ 「仕様」は要求の中の「動詞」にあるという考え

- ❖ 実現してほしい機能に関する「要求」を「振る舞い」として記述する

“仕様”は、要求の中の「動詞」および「目的語」に存在する

- ❖ 振る舞いの表現で使用される動詞と目的語に着目し、要求・仕様を正確に表現する
 - ❖ 全ての動詞（および目的語）を引き出す
 - ❖ 「動詞」に対して<仕様グループ>を立てる
 - ❖ 仕様は、動詞を「プログラムコード」に変換するための記述

USDMのこの構造が
仕様モレを軽減する

1.2 USDMの特徴

▶ 効果を引き出すUSDMの記載フォームと記載内容

上位要求

下位要求

仕様

下位要求

仕様

会員座席予約システム要求仕様			関連仕様	画面名:画面番号
項目	内容			
① 顧客による座席予約 (RSV)	要求 RSV.01	事前登録された会員を対象に、インターネット	① キーワードラベル	“機能を示す用語”や別名を添えて、要求や仕様の一覧性を良くする。 □をつけると、これ以上仕様化しない「認定仕様」となる
	理由 ③	お客さんが直接操作できるようにすることで納得の上で選んでもらうことで満足感を与え		
	説明 ④	会員に限定することで、支払い操作はすべて席は、指定席と自由席がある。また、ここで端末はすでにネットワークに接続して、複数の端末から同時に同じ講演の席を予約		
① 会員認証	⑥ <講演の予約可能確認>		② 要求	実現してほしい振る舞いを表現し、要求の範囲(ゴール)を明確に記述する。 要求には固有IDを付加する(例:RSV.01、RSV.01.01:番号も階層化する)。
	要求 RSV.01.01	② 会員番号と暗証番号の入		
	理由 ③	③ 精算をクレジットに限定す	③ 理由	理解のズレを抑えるために、その要求が必要な理由や背景を記述する。
	説明			
	⑦ <ログイン>		④ 説明	動きの事例、前提事項、用語定義など、要求の理解を助ける内容を記述する(ソースコードにはならない)。 場合によっては(必要ならば)仕様にも説明を付加し仕様の理解を助ける。
	□□ RSV.01.01.01	「初期画面」から[ログイン]		
	□□ RSV.01.01.02	「座席予約画面」から会員		
	□□ RSV.01.01.03	入力された会員番号が登		
	□□ RSV.01.01.04	会員登録されていない場	⑤ 仕様	仕様にも、要求と同様に固有IDを付加する(例:WPR.01.01.01:番号は要求からの階層でつける)
	□□ RSV.01.01.05	合		
□□ RSV.01.01.06	合			
□□ RSV.01.01.07	合			
□□ RSV.01.01.08	暗証番号が不一致のとき	⑥ グループ	要求や仕様の数が多い場合には、小さな集合に分割して範囲を限定する。	
□□ RSV.01.01.09	暗証番号の再入力に3度			
□□ RSV.01.01.10	る			
□□ RSV.01.01.11	暗証番号の再入力に3度			
□□ RSV.01.01.12	る	⑦ 仕様ラベル	仕様には□をつけ、要求と明確に分離する。 □は「レビュー済み/設計済/実装済」などのステータス管理に用いることができる(□□□→■□□や☑☑□)。	
□□ RSV.01.01.13	暗証番号の再入力に3度			
□□ RSV.01.01.14	る			
□□ RSV.01.01.15	暗証番号の再入力に3度			
□□ RSV.01.01.16	る			
① 要求 RSV.01.02	講演指定画面から講演名		⑤ 仕様	仕様にも、要求と同様に固有IDを付加する(例:WPR.01.01.01:番号は要求からの階層でつける)
理由	予約を急ぐかどうかの判			
説明 ④	④ 予約の対象とする講演名 注意		⑥ グループ	要求や仕様の数が多い場合には、小さな集合に分割して範囲を限定する。
<一覧表示の対象範囲>				
□□ RSV.01.02.01	講演一覧画面に表示する		⑦ 仕様ラベル	仕様には□をつけ、要求と明確に分離する。 □は「レビュー済み/設計済/実装済」などのステータス管理に用いることができる(□□□→■□□や☑☑□)。
□□ RSV.01.02.02	に登録されている講演			
□□ RSV.01.02.03	【説明】深夜に操作した場			
□□ RSV.01.02.04	また、この状況での			
□□ RSV.01.02.05	講演一覧の表示			
□□ RSV.01.02.06	「講演指定画面」の「一覧			
□□ RSV.01.02.07	表示する項目は、日時(開			
□□ RSV.01.02.08	講義ファイルを開いて、講			
□□ RSV.01.02.09	表示画面の見出しボタン			
□□ RSV.01.02.10	1) 主演者			
□□ RSV.01.02.11	2) 場所(会場)			
□□ RSV.01.02.12	3) 日付(初日~最終日)			
□□ RSV.01.02.13	4) 演出者			
□□ RSV.01.02.14	並び替えの指示は、表示			
□□ RSV.01.02.15	なう			

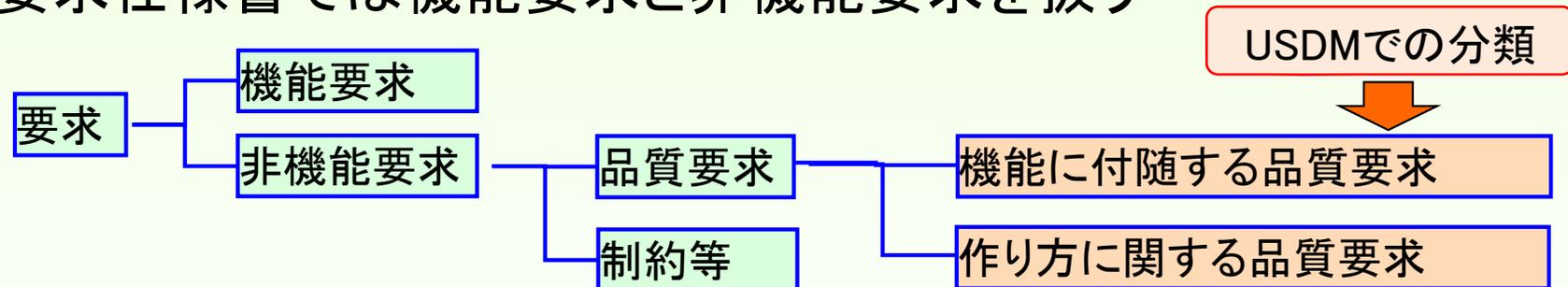
➤ USDMにおける要求仕様書の定義

- ❖ 「USDM」では要求仕様書を“作るための文書”と考える
 - ❖ 作業者に対して「作って欲しいこと」が書かれた文書
 - ❖ 「要件定義書」に近い

しかし、「Difinition」では、「Specify」することは難しい

“USDM”では、「要求仕様書とは、今回のプロジェクトで実現して欲しいこと (Requirements) について、“作ることの関係者”が実現内容についての認識を特定 (Specify) できている文書」と定義

❖ 要求仕様書では機能要求と非機能要求を扱う



1.2 USDMの特徴

➤ 要求仕様書と機能仕様書

❖ 「USDM」では、「要求仕様書」と「機能仕様書」を区別し使い分ける

❖ 要求仕様書・・・今回“**作るため**”の文書

❖ 機能仕様書・・・現状の製品やシステムの機能を“**説明している**”文書

	要求仕様書	機能仕様書(等)
目的	作るためのもの	機能を説明するもの
関係者	計画書で 特定されている	特定されていない
表現	関係者が Specify できる状態	一般的な記述
納期やコスト	背後に 背負っている	意識されない
バージョン管理	今回だけの文書	ソースコードと対で 管理される

- ❖ 機能仕様書は、ユーザーにとって「どのようなことができるのか」が記述されているが、その機能の**根源**となっている「**要求**」は記述されていない
- ❖ 要求仕様は、ユーザーが**求めていること**を「**要求**」とし、それに関連した「**仕様**」が書かれた文書
- ❖ 設計で必要となるのは「**仕様**」であり、「**要求**」は「**仕様**」を引き出す元となる重要な役割を果たす
- ❖ 機能仕様書は、保守性などの**作り方に関する要求**は扱われない

1. USDMのおさらい
2. 上手く表現するために
3. こんな場合には
4. この表現は如何ですか

- 2. 1 要求を上手く表現するには
- 2. 2 要求を上手く仕様化するには
- 2. 3 画面仕様を上手く表現するには
- 2. 4 非機能要求を上手く表現するには

2. 1 要求を上手く表現するには

2. 2 要求を上手く仕様化するには

2. 3 画面仕様を上手く表現するには

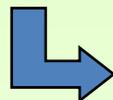
2. 4 非機能要求を上手く表現するには

- ▶ 多くの問題は、「要求」が適切に表現されていないことに起因している
 - ❖ 「機能名」の下に、いきなり「仕様」が書かれている

2.1 要求を上手く表現するには

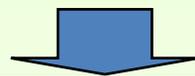
➤ 要求とは何か？

❖ システムに求められる**機能**や**性能**、作り方に対する**品質**など、実現したいことを「要求」として**抽象的に表現**したもの



作業者に対する要求

- ❖ システムや製品に求められる**振る舞いとしての機能**
- ❖ パフォーマンスなどの**機能に付随する品質**
- ❖ 保守性などの**作り方に対する品質**



実現したいゴール

機能仕様の表現でも、
作業者の方で、
“・ように作って欲しい”
と「要求」の形に読み
替えてくれる
が
意識して要求の表現
“…してほしい”
と表現することで要求の表現として
合わないものを見つけられる

要求	MAL01	受信した電子メールをキーワードで検索してメーラーで再利用 する
要求	BYY03	ネット販売の購入の最終確認操作では誤操作が入る余地を 排除する
要求	FLE10	今日の売れ筋商品のトップ10について1週間の販売動向を同時に 表示する

2.1 要求を上手く表現するには

➤ 要求”を引き出す

❖ 要求を発見する研究は世界中で進められている

❖ 顧客の求めるものを整理し、そこから「要求」の形にまとめる

❖ 『実践ソフトウェアエンジニアリング』(ロジャー・プレスマン著)では、**要求エンジニアリング**を以下のステップで説明している

①	開始	ニーズや新しいサービスの必要性が生じたとき
②	要求獲得	ここで要求を発見する
③	推敲	獲得した要求を表現しながら調整し洗練する
④	交渉	要求間の調整、組織内部での要求の調整など
⑤	仕様化	要求が安定したところで一気に仕様化する
⑥	検証	レビューを繰り返して精度を高める → ベースライン設定へ
⑦	要求マネジメント	その後の変更管理

繰り返す

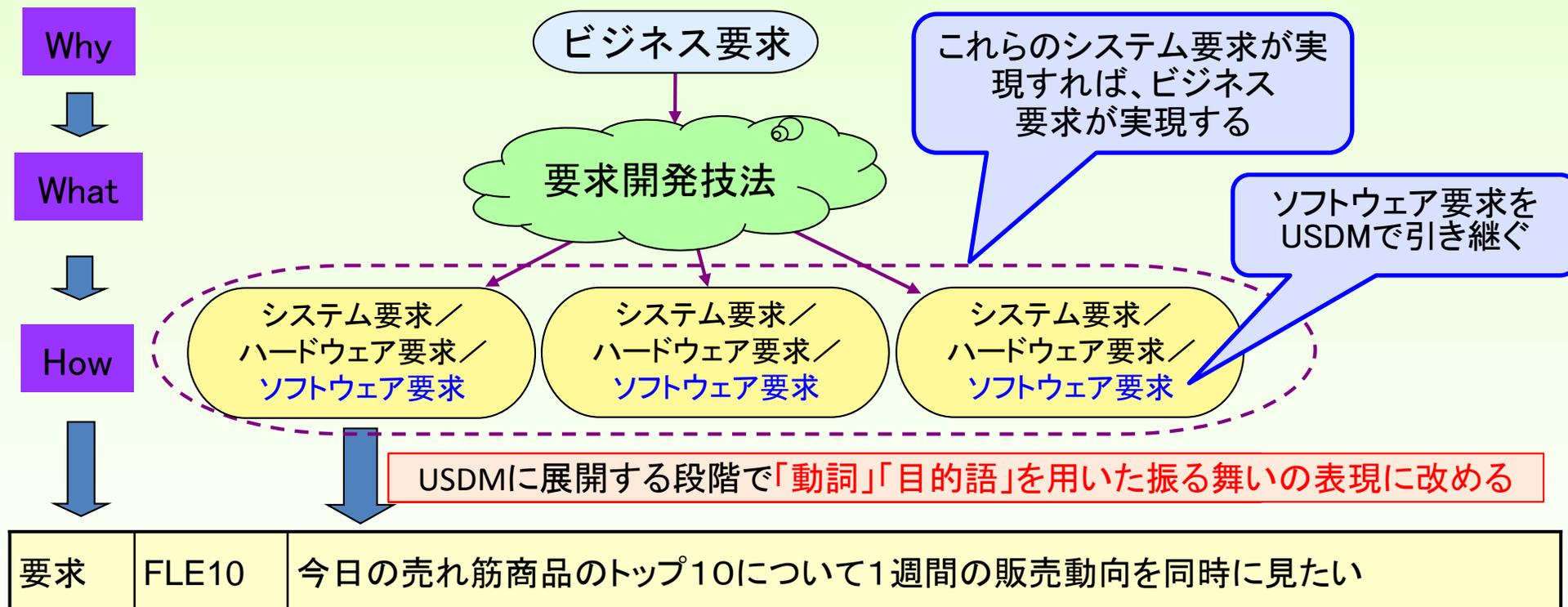
❖ 上記の②～④の中で要求が引き出される

❖ 一般には、⑤の段階で別の文書が作られるが、「USDM」では**一体で扱う**

2.1 要求を上手く表現するには

➤ ビジネス要求からシステム要求を引き出す

- ❖ 要求開発技法によって、**ビジネス要求**からシステム要求を引き出す
 - ❖ ビジネス上で解決したいことから、「システム要求」を引き出す
 - ❖ システム要求は表現は多様なため、「USDM」では扱わない(扱いにくい)



2.1 要求を上手く表現するには

- ▶ ビジネス要求からソフトウェア要求までは階層を成している



要求	FLE10	今日の売れ筋商品のトップ10について1週間の販売動向を同時に見たい
----	-------	-----------------------------------

2.1 要求を上手く表現するには

各レベルのUSDM適用とUSDMの関係

システムレベルのUSDMと、サブシステムレベルのUSDM、ソフトウェアレベルのUSDMに分けて記述する

システム要求仕様		
要求	AAA	
	理由	
	説明	
	仕様	AAA01
		理由
		説明

サブシステム要求仕様		
要求	AAA	
	理由	
	説明	
		<グループ>
要求	AAA01	
	理由	
	説明	
		<.....>
		□□□ AAA01.01
		□□□ AAA01.02
		□□□ AAA01.03
		<グループ>
要求	BBB01	
	理由	
	説明	
		<.....>
		□□□ BBB01.01
		□□□ BBB01.02
要求	BBB02	
	理由	
	説明	
		<.....>
		□□□ BBB02.01
		□□□ BBB02.02
		□□□ BBB02.03
		<.....>
		□□□ BBB02.04
		<グループ>
要求	BBB03	
	理由	
	説明	
		<.....>
		□□□ BBB03.01

システムソフトウェア仕様は、ソフトウェア要求になる

ソフトウェア要求仕様		
要求	SO01	
	理由	
	説明	
		SO01.01
		理由
		説明
		<.....>
		□□□ SO01.01.01
		□□□ SO01.01.02
		□□□ SO01.01.03
		□□□ SO01.01.04
		<.....>
		□□□ SO01.01.05
		<.....>
要求	SO01.02	
	理由	
	説明	
		<.....>
		□□□ SO01.02.01
		□□□ SO01.02.02
		□□□ SO01.02.03
		□□□ SO01.02.04

システム要求に対する
実現手段はシステム仕様

システム仕様は、
サブシステムの要求になる

コンポーネント
担当部署が仕様を定義

➤ USDM表記におけるキーワード

❖ 捉えた要求をモレなく、正確に仕様化するためには、以下のキーワードが重要

1. 「動詞」に着目する
2. 「範囲」を明確にする
3. 「理由」は必ず明記する
4. (振る舞いを) 詳細化するために「階層化」する
5. (振る舞いの) 範囲を狭めるために「分割」する
6. 「グループ」を用いて範囲を限定する
グループにより要求の粒度・深さを調整して要求を洗練させる
下位の要求や仕様を引き出す効果もある

2.1 要求を上手く表現するには

▶ 「動詞」に着目する

- ❖ 機能要求は“振る舞い”で表現される
- ❖ 「動詞」とその「目的語」を上手に使うって“振る舞い”を表現する
- ❖ タイトルや簡単な動作 (“名詞+する”のような)だけでは、要求の役割を果たさない



	要求	DA07	計測データの表示
--	----	------	----------



	要求	DA07	計測データをリアルタイムに 表示する
--	----	------	---------------------------



- 機能名では、伝わった気になるだけで、何を作るのかを特定できない
- 要求の表現が悪いと、そこから適切な仕様が引き出せない



計測データ表示	要求	ALM.03	計測データを 受信 し、平均値を 算出 しながらリアルタイムに 表示 したい その際、異常値が 検出 されたときは警告を 表示 したい
---------	----	--------	---

- ❖ 要求(振る舞い)に含まれる「動詞」をすべて表現することを目指す
- ❖ この要求の「赤字」の部分について仕様を展開することになる
 - ❖ 「動詞」を引き出してしまえば仕様は漏れない

2.1 要求を上手く表現するには

➤ 「範囲」を明確にする

「範囲」の表し方で見えるものが変わる

❖ 今回の要求の「範囲」は何処までかはっきりさせる

❖ 「範囲」の表現が不明確だと、その後の仕様の抽出でモレやすくなる

	要求	ALM03	予想との乖離が一定以上開いたときは通知する
--	----	-------	-----------------------



	要求	ALM03	商品毎に 設定 されている当日の売り上げ数量の予測に対して、実売データとの間に大きな 開きがある ときは警告メッセージをマネージャーのPC 画面に出す
--	----	-------	--

❖ 「動詞」が見えることで仕様を抽出しやすくなる

“要求を洗練する”

❖ 振る舞いの「範囲」をイメージできるように**要求の表現を調整**する

	要求	MAL01	受信および送信した電子メールを キーワードで検索して再利用 したい
--	----	-------	--

- 電子メールのファイルのすべてを対象にしない方が良さそうだ
- 再利用する時は、電子メールのソフトに直接接続できた方が良いだろう



❖ 実現したい「範囲」を表現することで、関係者の認識のズレを矯正する

	要求	MAL01	事前に 指定 された受信および送信した電子メールを キーワードで検索し、選択 した電子メールをメーカーに 繋いで再利用 したい
--	----	-------	--

2.1 要求を上手く表現するには

▶ 「理由」は必ず明記する（要求とセットで表現する）

❖ 要求には、それが必要な理由（存在理由）や「背景」がある

❖ その要求が無いと困ることは？

❖ その要求があることで恩恵を受けることは？

・「目的」は扱わない
・「理由」と重なるケースは「理由」として記述する

	要求	SAL.01	商品毎に設定されている当日の売り上げ数量の予測に対して、実売データとの間に大きな開きがあるときは警告メッセージをマネージャーのPC画面に出してほしい
	理由		すぐに原因を調べて、陳列方法の変更など適切な対応策を講じる必要がある

❖ 要求にその理由や背景を併記することで、要求が安定する

❖ 「理由」によって要求に対する関係者の認識のズレを調整できる

❖ 要求全体に対する理由の他に、特定の「動詞」に理由が存在することもある

❖ 必要かつ有効な仕様が引き出せる

❖ 設計上で工夫されることがある

❖ システムの非機能要求が理由になることもある

❖ 理由があることで要求の意義が明確になり、要求が洗練される

❖ 思い込みや勘違いを防ぐことができる

❖ 不純な要求を排除できる

2.1 要求を上手く表現するには

➤ (振る舞いを) 詳細化するために「階層化」する

❖ 抽象度が高い要求は、階層化し抽象度を下げる

❖ 隠れていた動詞を探し下位層の要求として表現する

❖ 下位層にも理由が必要

		要求仕様書	
メール検索機能	要求	MAL01	電子メールのグループを指定して、そこからキーワードで検索してメーラー上で再利用したい
		理由	メールが多くて、関連するメールを探すのに手間取る
		説明	
	要求	MAL01.01	表示された検索グループの中から一つを指定する
		理由	検索グループが複数あるから
		説明	
	要求	MAL01.02	いくつかのキーワードの入力を受け、それらを組み合わせて検索する
		理由	可能性のあるキーワード(複数)で探したい
		説明	
	要求	MAL01.03	検索結果を表示し、見つかったときは「subject」などの情報を一覧で表示する
		理由	Subjectと日付などから目的のメールを探すことになる
		説明	
	要求	MAL01.04	一覧の中から選択されたメールを開く
		理由	中身を見ないと分からないから
		説明	
	要求	MAL01.05	一つのメールを開いた状態でメーラーに繋いで編集できるようにする
		理由	コピーの手間を省いて編集の操作に入りたい
		説明	
要求	MAL02		

一般には、上位要求の下に「時系列」で配置する

2.1 要求を上手く表現するには

➤ (振る舞いの) 範囲を狭めるために「分割」する

- ❖ 要求の範囲が広すぎると下位要求(・仕様)が発散してモレが生じやすいため、明確な基準を用いて分割(・階層化)する

USDMの「4つの分割基準」

分割方法	内容と例
時系列分割	<p>時間軸に着目して動作・処理を分割する方法</p> <p>(例) メールを検索する → 検索グループから対象を指定する → キーワードを入力する</p>
構成分割	<p>時間的な順序を持たない機能や構成で分割する方法</p> <p>(例) メールシステム → メールを受信 → メールを送信 →メールの作成</p>
状態分割	<p>状態という概念に従って分割する方法</p> <p>(例) メールする → OFF 中 → 送受信待機中 → データ処理中</p>
共通分割	<p>複数の要求の共通する機能部分を切り出しまとめ独立させる方法</p>

構造化設計の「モジュールの分割基準」

入力・変換・出力分割

トランザクション分割

データ構造分割

共通分割

要求(下位要求)や仕様のモレを防ぐには、書き手と読み手が**分割基準を共有**する必要がある

2.1 要求を上手く表現するには

➤ 「グループ」を用いて範囲を限定する

- ❖ 要求の粒度(詳細化のレベル)によっては、分割した下位要求(・仕様)の数が多くなり、モレに気づきにくくなる
 - ❖ グループにより要求の粒度・深さを調整して要求を洗練させる
- ❖ 下位要求(・仕様)を共通項で「グループ化」し範囲を限定する
 - ❖ グループ名も動詞(・目的語)を意識する
 - ❖ グループ分けの基準にも、要求の分割基準を用いる
 - ❖ グループを設定してから要求(・仕様)を分割しても構わない

下位の要求(や仕様)を引き出す効果もある

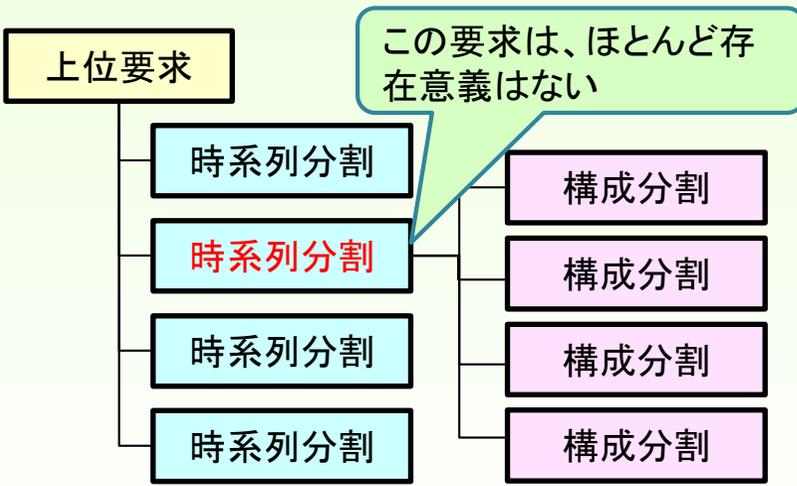
要求仕様書			
アラーム機能	要求	ALM.01	設定した時刻になったら、好みのアラーム音を鳴らしてほしい
		理由	決めた時刻に快適に起きたい
	説明		
		<時刻の設定>	
	要求	ALM.01.01	アラームを鳴動させる時間を設定できるようにする
	理由		好きな時間をセットしたいので
	説明		
		<アラーム音の設定>	
	要求	ALM.01.02	アラーム音をすでに持っている数種類のものから選択できるようにしてほしい
	理由		好みの音を鳴らしたいので
	説明		
		<アラームON/OFFの設定>	
	要求	ALM.01.03	アラームの鳴動をON/OFFできるようにしてほしい
	理由		アラームが不要なときアラームが鳴らないようにしておきたい
	説明		
		<アラームの停止>	
	要求	ALM.01.04	鳴動しているアラームの+音を停止させたい
	理由		鳴り続けているとうるさいので
	説明		

グループは<「名詞」>の形式で記述する

2.1 要求を上手く表現するには

- 要求の階層はむやみに深くしない
 - ❖ 部分的に要求の階層が深くなることは避ける
 - ❖ 下位の要求をまとめるだけの無意味な要求を作らない
 - ❖ 階層が深くなると煩雑になる
 - ❖ 階層的に理解していくときの理解性が悪くなる
 - ❖ 階層は2階層以内になるように工夫する
 - ❖ 上位の要求は必要か？
 - ❖ 要求の粒度が合うようにグループを利用して、シンプルな“動き”の単位にする

要求の
<グループ>
だけを見れば、
「時系列」
になる



<グループ> を使って
要求をまとめる



2. 1 要求を上手く表現するには

2. 2 要求を上手く仕様化するには

2. 3 画面仕様を上手く表現するには

2. 4 非機能要求を上手く表現するには

- ▶ 仕様化が曖昧なことが、多くの問題を引き起こしている

➤ 要求の仕様化

- ❖ 要求が適切に表現されなければ仕様は正しく引き出せない
- ❖ ソフトウェアとしてソースコードに変換されるのは仕様である
- ❖ 仕様の条件
 - ❖ 要求に含まれる“**具体的**”な「**処理**」や「**動き**」を表現したもの

- 仕様は「**要求**」から**導出**される
 - 全ての仕様は、いずれかの**要求**に属する

- **実現可能性**が見える(設計者の立場から)
 - 前提条件や処理や振る舞いのための「**制約**」なども含むこと
 - 具体的であるために“**仕様間の矛盾**”が見える
 - 設計の様子がイメージできることがある

- **検証可能性**が見える(評価担当者の立場から)
 - その仕様が実現していることを検証する方法が見える
 - 具体的であることで、関係者の間で違った意味に解釈されない

- **品質要求**も例外ではない
 - 品質要求にも、実現可能性と検証可能性は求められる

2.2 要求を上手く仕様化するには

➤ 仕様も「グループ化」する

❖ 事前に設定されたグループの仕様を埋める

- ❖ 動詞に着目し動詞の単位に仕様のグループを設定する
- ❖ 仕様のグループ化も要求の「分割」と同じ
- ❖ グループによる分割に「すき間」がないこと
- ❖ 設定された<グループ>の中に仕様を埋める

要求で見た「動詞」の単位でグループにする

要求	MAL01.03	検索結果を 表示 し、見つかったときは「subject」などの情報を 一覧で表示 する
	理由	目的のメールが一つとは限らないので、絞り込めるような操作がしたい
	説明	
		<検索結果の表示>
<input type="checkbox"/>	MAL01.03.01	検索されたメールの件数を一覧の上に表示する
<input type="checkbox"/>	MAL01.03.02	該当するメールが存在しないときは「該当無し」を表示する
		<検索されたメールの情報の表示>
<input type="checkbox"/>	MAL01.03.05	検索されたメールの「Subject」を一覧で見せる
<input type="checkbox"/>	MAL01.03.06	検索されたメールに連続番号をつけて表示する
<input type="checkbox"/>	MAL01.03.07	検索されたメールの件数が10件を超えるときはスクロールバーを見せる

※動詞でグループ化すれば、ほぼ「時系列分割」になる

2.2 要求を上手く仕様化するには

➤ 仕様はどこまで詳細化するか

❖ 開発者および関係者が「Specify (特定)」できる状態まで詳細化する

❖ (要求)仕様書 (Specification)は、「Specify」という考え方に基づく

❖ 開発者がプログラミング時に

・何を実現するのか

・どのように実現するのか

と“悩む”ならば

❖ 上記の「何を実現するのか」は仕様である

何を実現するのか」で悩むということは、

仕様が「Specify」されていないことになる

関係者の能力の違い によって書き方を変えるべきという認識が必要

要求仕様書は今回の関係者が「Specify」できたことが問われる文書

❖ 「詳細な仕様は設計に入ってから抽出するもの」は**固定概念**

❖ 「Specify」されていないまま進むと、**手戻りの原因**になる

2.2 要求を上手く仕様化するには

- ▶ 仕様にも、必要なら理由や説明を付ける
 - ❖ 要求から明らかに想像できる仕様であれば理由や説明はいらない
 - ❖ 仕様の背景にある「理由」や「説明」が必要な場合がある
 - ❖ なぜ、この仕様が必要なのか
 - ❖ 理由は、設計上で配慮されることがある
 - ❖ 説明は、仕様の動きなどを補足するものであって、ソースコードにはならない

要求	CH-02	印刷開始時にインクの残量を調べて、最後まで印刷できない可能性があれば印刷依頼者のPC上に知らせて欲しい
	理由	印刷内容を知っている人に、最初に危険性を知らせたい
	<チェックのタイミング>	
□ □ □	CH-02-1	印刷開始の操作を完了した時点でプリンターと交信してインクの残量(全カートリッジ分)を知る 【理由】印刷ボリュームと比べて判断する必要があるから
	<確認操作>	
□ □ □	CH-02-2	「注意」または「警告」のアナウンスを出したときは、「確認」の操作によって印刷処理を続ける 【説明】インクの交換はこの間に行い、交換後に「確認」操作によって印刷を続ける

2.2 要求を上手く仕様化するには

➤ 仕様から要求を立てる -1-

❖ 要求仕様を記述していると、仕様が「要求」と「理由」を共有していないことが起こることがある

❖ “みにくいアヒルの子”現象

❖ 関連する仕様を書いている時に、“ふっと”必要に気が付いて追加した

❖ 読み返している時に処理の“流れ”の具体的なイメージがわき、その先の仕様が浮かんで追加した

この仕様は、
要求と理由を
共有しない

要求	MAL01-02	いくつかのキーワードを組み合わせて検索できる
	理由	可能性のあるキーワードを簡単に探したい
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	MAL01-02-1	検索したいキーワードを入力できる
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	MAL01-02-2	複数のキーワードを「AND」と「OR」で繋ぐことができる
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	MAL01-02-3	キーワードは最大8個まで指定できる
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	MAL01-02-4	検索されたメールの「Subject」を一覧で見せる

❖ このような場合、以下のように対応する(3種類ある)

- ① この仕様が収まる要求を探して移動する
- ② 「要求」と「理由」の表現を変えて、この仕様を受け入れる
- ③ 新たに適切な「要求(と理由)」を設定し、この仕様をそこに移動する

2.2 要求を上手く仕様化するには

▶ 仕様から要求を立てる -2-

❖ 新たに要求と理由を設定したことで、**仕様の欠如に気付くことがある**

要求	MAL01-03	検索結果を扱いやすく表示し、そこから選択したい
	理由	目的のメールが一つとは限らないので、絞り込めるような操作がしたい
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	<検索結果の表示>	
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	MAL01-03-1	検索されたメールの件数を一覧の上に表示する
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	MAL01-03-2	該当するメールが存在しないときは「該当無し」を表示する
	<検索メールの表示>	
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	MAL01-03-5	検索されたメールの「Subject」を一覧で見せる
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	MAL01-03-6	検索されたメールに連続番号をつけて表示する
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	MAL01-03-7	検索されたメールの件数が10件を超えるときはスクロールバーを見せる
	<メールの中身の表示>	
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	MAL01-03-10	一覧から1つのメールを選んで、その内容を見ることができる
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	MAL01-03-11	開示されたメールの中で検索キーワードと一致している文字列を赤色で表示する
	<メールの選別>	
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	MAL01-03-15	不要なメールを選んで一覧から消すことができる
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	MAL01-03-16	一度不要として消されたメールを復活することができる

一般の要求仕様書の構成では、このテクニックは使えない！

2.2 要求を上手く仕様化するには

▶ 仕様から要求を立てる -3-

❖ 要求と要求仕様の混在に対応する

❖ 依頼者の“要求”には「要求」と「要求仕様」が混じっている

❖ 聞き取りの中で、依頼者が“具体的な例(=仕様)”を示すことがある

❖ この時の「仕様」は、依頼者が表現している要求とは別の要求に含まれるべく仕様の一部である場合がある

このまま設計しようとするから“仕様の漏れ”が発生する

❖ 要求が不明なときは、いったん「仕様」の欄に記述し、インタビューの中で「要求」と「理由」を探す

〇〇機能	要求		
		理由	
	□ □ □	SHT-	仮予約状態の座席を解除して「空」状態に戻す

日常の中で仕様から要求を立てる練習をすること

2.2 要求を上手く仕様化するには

▶ ペースト作文による仕様記述は避ける -1-

❖ 「ペースト作文」は仕様モレを招く

- ❖ “コピー&ペースト”によって、一部を変更して文章を完成させる方法
- ❖ 「ある条件のときに、ある値になる」というような仕様を1行ずつ書く(またはペースト作文によって作成する)と、条件が漏れることがある
- ❖ このような記述は、読み手にとっても苦痛極まりないものである
- ❖ 条件の組合せや必要な変数を見落としてしまう

□□□	THM01-1	前回温度が基準値1以下で、かつ、今回温度が基準値1以下の場合は、「A」
□□□	THM01-2	前回温度が基準値1以下で、かつ、今回温度が基準値1を超える場合は、「B」
□□□	THM01-3	前回温度が基準値2以下で、かつ、今回温度が基準値1以下の場合は、「B」
□□□	THM01-4	前回温度が基準値2以下で、かつ、今回温度が基準値1を超える場合は、「C」
□□□	THM01-5	前回温度が基準値2を越え、かつ、今回温度が基準値1以下で、自動設定ONの場合は、「D」
□□□	THM01-6	前回温度が基準値2を越え、かつ、今回温度が基準値2以下で、自動設定OFFの場合は、「E」
□□□	THM01-7	前回温度が基準値2を越え、かつ、今回温度が基準値2を超える場合は、「F」

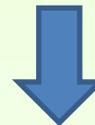
このような場合、表やディシジョンチャート、図などを活用する

2.2 要求を上手く仕様化するには

▶ ペースト作文による仕様記述は避ける -2-

❖ 違いが分かるような表現に変える

□□□	THM01-1	前回温度が基準値1以下で、かつ、今回温度が基準値1以下の場合は、「A」
□□□	THM01-2	前回温度が基準値1以下で、かつ、今回温度が基準値1を超える場合は、「B」
□□□	THM01-3	前回温度が基準値2以下で、かつ、今回温度が基準値1以下の場合は、「B」
□□□	THM01-4	前回温度が基準値2以下で、かつ、今回温度が基準値1を超える場合は、「C」
□□□	THM01-5	前回温度が基準値2を越え、かつ、今回温度が基準値1以下で、自動設定ONの場合は、「D」
□□□	THM01-6	前回温度が基準値2を越え、かつ、今回温度が基準値2以下で、自動設定OFFの場合は、「E」
□□□	THM01-7	前回温度が基準値2を越え、かつ、今回温度が基準値2を超える場合は、「F」



□□□	THM01-1	液体温度の基準値による処理設定を下表に示す			
		前回温度	今回温度	自動設定	処理
		≦ 基準値1	≦ 基準値1	—	A
			> 基準値1	—	B
		≦ 基準値2	≦ 基準値1	—	B
			> 基準値1	—	C
		> 基準値2	≦ 基準値1	ON	D
			≦ 基準値2	OFF	E
			> 基準値2	—	F

デシジョンチャート

2.2 要求を上手く仕様化するには

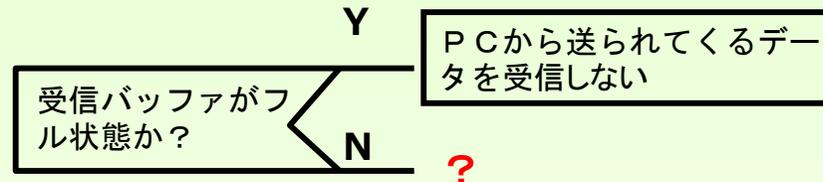
➤ 否定表現の問題

❖ 否定表現の仕様記述は不十分

❖ 否定表現とは、

❖ …が…の時は、…しない

ex. 受信バッファがフルの状態の時は、データを受信しない



❖ 聞き取りや文章の表現の中では自然に使われることがありその場では、それで「分かった」と認識されてしまう

❖ コーディング時に「IF」の片側が定義されていないことに気付くが、それでは遅い！

❖ 「否定表現」が早い段階で認識されることで適切な対応方法が可能になる

❖ “肯定表現”でも片側の定義が欠けるケースがある

❖ …が…の時は、…する

ex. 「在庫があることを確認できたときは、速やかに出庫の手配をする」

もう一方の仕様も忘れずに記述する！

2.2 要求を上手く仕様化するには

▶ 曖昧な表現を避ける

❖ 文章の中に“等”の記述も避ける

LOG01.02	コマンドを受信したときは、その時点でのコマンド名等のログデータを、規定のフォーマットでログファイルに出力する。
----------	---

関係者が
「Specify」できる記
述ではない

❖ これでは、仕様として役に立たない

❖ 仕様の文章にうまく記述できないときは、その下に具体的に列記する

LOG01.02	<p>以下のコマンドを受信したときは、その時点での出力するデータを、規定のフォーマットでログファイルに出力する。</p> <p><対象となるコマンドの種類></p> <ul style="list-style-type: none"> ・Start command ・Stop command ・Set command ・Write command ・Read command <p><ログファイルに出力するデータ></p> <ul style="list-style-type: none"> ・受信時間 ・コマンド名 ・受信データ
----------	---

- ▶ 確定しない仕様は「暫定仕様」として要件管理で対応
 - ❖ 要求定義時に仕様がなかなか決まらない場合がある。
 - ❖ 残りの数%の仕様を確定させるのに、必要以上の時間を待つことはない
 - ❖ 要求仕様書のベースライン設定時に、まだ合意できていない仕様が有れば、それを明示し「暫定仕様」で進める方法がある
 - ❖ 値や方式が決まっていなくても、「選択肢」を示すことはできる
 - ❖ 分かっていることをすべて記述し、合意できていない箇所を明示していれば、一般に設計に取り掛かることは可能
 - ❖ 仮設の仕様として通常の仕様と同様に記述し、「暫定仕様」である事を明示する
 - ❖ 仕様を確定させるまでの“期限”を限る方法も有効
 - ❖ その後に仕様が確定した時は「要件管理(変更管理)」で対応する
 - ❖ 「暫定仕様」の部分は後で変更される可能性が高いので、「要件管理」の「追跡機能」でカバーする

2.2 要求を上手く仕様化するには

➤ 「認定仕様」という考え方

- ❖ USDMでは、**特殊なケース**として「要求」の状態であっても、その状態で実装可能とするケースを「**認定仕様**」として容認する
- ❖ 「要求」や<仕様グループ>単位で対応する
- ❖ 「Specify」の考えを活用したもので、「**要求仕様書**」だけに適用できる

要求仕様			
要求	MAL01	事前に指定された受信および送信した電子メールをキーワードで検索してメーラー上で再利用したい	
	理由	メールが多くて、関連するメールを探せない	
	説明		
要求	MAL01-01	検索グループを指定できる	
□□□	理由	検索グループが複数あるから	
	説明		
要求	MAL01-02	いくつかのキーワードを組み合わせて検索できる	
	理由	探したいメールにを見つけやすくするため	
	説明	...	
		<キーワード入力>	
□□□	MAL01-02-1	検索したいキーワードを入力できる	
□□□	MAL01-02-2	複数のキーワードを「AND」と「OR」で繋ぐことができる	
□□□	MAL01-02-3	キーワードは最大8個まで指定できる	

「仕様」として
めたことを示す

認

「要求」“MAL01-01”は、
この下に仕様を 展開
しない

書けるけど書かないのと、書けなくて書かないのとでは、
まったく違うことに注意 !

2. 1 要求を上手く表現するには

2. 2 要求を上手く仕様化するには

2. 3 画面仕様を上手く表現するには

2. 4 非機能要求を上手く表現するには

- ▶ 最近では設定や動作などの多くの操作が「画面」を通じて実現できる

2.3 画面仕様を上手く表現するには

➤ 画面仕様にもUSDMを適用する

❖ 画面仕様も、要求と仕様の構成を取り入れることで、精度の高い画面仕様書となる

I. 操作系全体の「画面要求」をまとめることで、システムとして操作系に何が求められているかが明確になる



要求	#1	ファイルと接続に関する設定項目は、全て画面上で行えること
要求	#2	セキュリティの設定ができること
要求	#3	セキュリティのレベルに応じて、見える画面、操作できる画面やボタンを設定できる
要求	#4	どの操作中でも、ベースの画面に直接戻って、そこから新しい画面操作ができること
要求	#5	接続されている機器の計測データは全てリアルタイムで表示し、異常が判断されるときはオペレータに適切な注意を喚起すること
要求	#6	操作エラーを含めて、エラー表示は、エラーのグループ(種類)が分かるようにすること
要求	#7	エラー表示に対する操作は、簡単にヘルプで確認できること
要求	#8	見えにくい文字のサイズや色を使わないこと(品質要求)
要求	#9	マウスによる選択では、操作ミスを起こしにくくすること(品質要求)

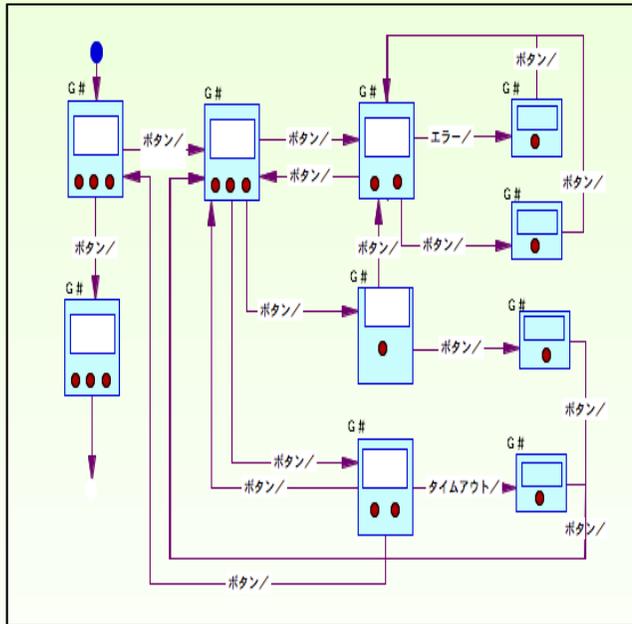
2.3 画面仕様を上手く表現するには

画面仕様書の構成

画面に関する仕様は、次のものがある

- 画面遷移図: システム全体の画面構成を示す(画面全体を見渡せること)
- 画面共通要求仕様: 各画面に共通な個別画面の要求の方向性(思想)を示す
- 個別画面要求仕様: 個々の画面の構成要素(画面要素)ごとの要求と仕様を表す

画面遷移図



画面共通要求仕様

画面共通要求仕様	
＜画面共通要求＞	
要求	R0-01
理由	
説明	
＜……＞	
要求	R0-01-01
理由	
説明	
＜……＞	
要求	R0-01-01-01
理由	
説明	
要求	R0-01-01-02
理由	
説明	
要求	R0-01-01-03
理由	
説明	
要求	R0-01
理由	
説明	
＜……＞	
要求	R0-01-01
理由	
説明	
＜……＞	
要求	R0-01-01-01
理由	
説明	
要求	R0-01-01-02
理由	
説明	
要求	R0-01-01-03
理由	
説明	
要求	R0-01-01-04
理由	
説明	
＜……＞	
要求	R0-01-01
理由	
説明	
＜……＞	
要求	R0-01-01-01
理由	
説明	

個別画面要求仕様

番号	要求	説明
REQ001	画面全体のレイアウト	画面全体のレイアウトを決定する。
REQ002	タイトルバーの表示	画面のタイトルバーを表示する。
REQ003	メインコンテンツ領域の表示	画面のメインコンテンツ領域を表示する。
REQ004	フッター領域の表示	画面のフッター領域を表示する。
REQ005	ボタン配置	画面下部にボタンを配置する。
REQ006	エラーメッセージ	エラー発生時にメッセージを表示する。
REQ007	タイムアウト	操作が一定時間続かない場合、タイムアウトメッセージを表示する。
REQ008	検索機能	検索ボックスと検索ボタンを配置する。
REQ009	リスト表示	データリストを画面中央に表示する。
REQ010	ページネーション	リストのページネーション機能を実装する。
REQ011	ヘルプ機能	ヘルプボタンを配置し、ヘルプ画面を開く。
REQ012	戻るボタン	戻るボタンを配置する。
REQ013	戻るボタン	戻るボタンを配置する。
REQ014	戻るボタン	戻るボタンを配置する。
REQ015	戻るボタン	戻るボタンを配置する。
REQ016	戻るボタン	戻るボタンを配置する。
REQ017	戻るボタン	戻るボタンを配置する。
REQ018	戻るボタン	戻るボタンを配置する。
REQ019	戻るボタン	戻るボタンを配置する。
REQ020	戻るボタン	戻るボタンを配置する。
REQ021	戻るボタン	戻るボタンを配置する。
REQ022	戻るボタン	戻るボタンを配置する。
REQ023	戻るボタン	戻るボタンを配置する。
REQ024	戻るボタン	戻るボタンを配置する。
REQ025	戻るボタン	戻るボタンを配置する。
REQ026	戻るボタン	戻るボタンを配置する。
REQ027	戻るボタン	戻るボタンを配置する。
REQ028	戻るボタン	戻るボタンを配置する。
REQ029	戻るボタン	戻るボタンを配置する。
REQ030	戻るボタン	戻るボタンを配置する。

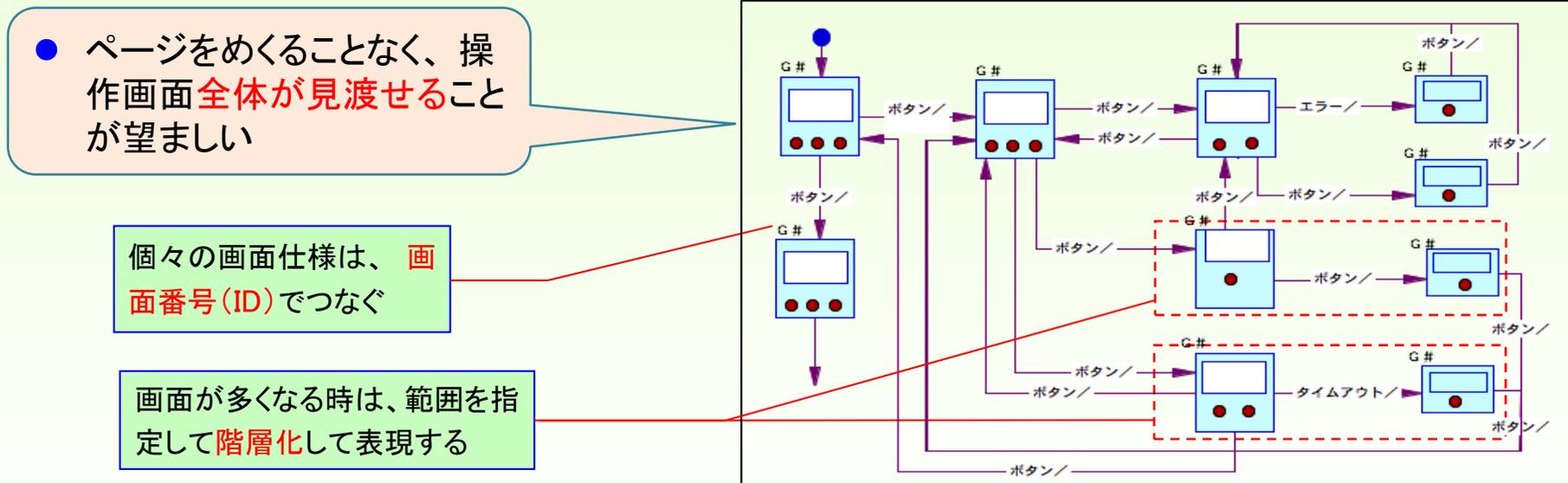
2.3 画面仕様を上手く表現するには

➤ 画面遷移図

❖ 画面遷移図はきちんと表現する

- ❖ 画面は「状態」の1種であり、画面遷移は**状態遷移の4つの要素**（画面状態、遷移、イベント、アクション）で正確に表現する
 - ❖ 画面遷移図は「操作システム設計書」の構成要素
 - ❖ “イメージ”だけの遷移図は設計書としての効力を発揮しない
- ❖ 全体を俯瞰できるようなレイアウトにすることで、モレや勘違いを発見しやすくなる

画面遷移図



2.3 画面仕様を上手く表現するには

➤ 画面仕様にもUSDMを適用する

- ❖ 画面全体に関わる「要求」の仕様化は、個々の画面要求の方向性を示す
- ❖ 操作画面全体に対する要求は、システム全体に対する「思想」の意味を持つ
 - ❖ デザイン的な思想や制約的を示す
 - ❖ 画面の設計や、その先の仕様化に対する「枠」となる

画面別対応

				#1	#2	#x
要求	OP1	どの操作中でも、「ベースの画面」に直接戻って、そこから新しい画面操作ができること				
	理由					
	□□□	OP1.1	操作画面が重なった時でも、「ベース画面」は隠れないようにする			
	□□□	OP1.2	ベース画面に直接戻った時はそれまでの操作は無効になるので、入力データの設定の変更などが行なわれた場合は、その場で確認を求めること			
要求	OP2	接続されている機器の計測データは全てリアルタイムで表示し、異常が判断されるときはオペレータに適切な注意を喚起すること				
	理由					
	□□□	OP2.1	異常の表示には最大3段階(緊急、警戒、注意)で表示する			
	□□□	OP2.2	原則として異常を表示している時も計測を継続する			
	□□□	OP2.3	異常の種類によっては、異常表示中でもデータはリアルタイムで捉え続けていて、異常が解除されたときは、即座に「今」の状態を表示できること			

2.3 画面仕様を上手く表現するには

➤ 個別画面仕様 -1-

❖ 個別画面の配置図のページに、画面の構成要素ごとに要求と仕様を表現する

- 通常の**要求仕様の表記法をそのまま適用**できる

画面が大きいときは、「要素」にして小さくするか、別の場所で表現する

この画面が表示されるときの仕様を記述する
・領域を指定する仕様
・ボタンや種類の仕様
・色や背景書の仕様など

それぞれのカテゴリ毎に「要求」を設定し、その下に「仕様」を展開する

画面遷移図と対応する

UI:01 モニタ画面仕様 画面番号:画面名

初期画面表示	要求	UI.01.01	この画面の初期状態を表示		
		理由			
		説明			
		<画面構成>			
		□□□	UI-01.01.01	タテ○○ヨコ○○の表示領域をXXXXで表示する	
		□□□	UI-01.01.02		
表示領域					
表示域	要求	UI.01.02	モニタ情報、異常が発生したときの情報を表示してほしい		
		理由			
		説明			
		<初期画面>			
		□□□	UI-01.02.01	○○データと○○データを表示する	
		□□□	UI-01.02.02		
売れ筋ボタン					
売れ筋ボタン	要求	UI.01.03	売れ筋ボタンを押したら、ここ1週間の売れ筋商品を一覧で表示してほしい		
		理由			
		説明			
		<ボタン押下の条件>			
		□□□	UI-01.03.01	常に押下できること	
		□□□	UI-01.03.02		
		□□□	UI-01.03.03		
		<押したときの動作>			
□□□	UI-01.03.04	ボタンの色を「赤」に変化させる			

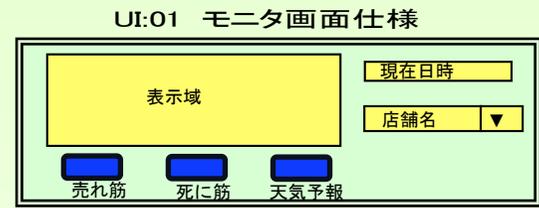
「ボタン」や「表示領域」などの画面を構成する要素で区切る

2.3 画面仕様を上手く表現するには

➤ 個別画面仕様 -2-

❖ 画面要素ごとに要求と仕様を表現する

❖ 個々の画面構成のすべての「要素」に対して「要求」を表現し、「仕様」を展開する



*****	要求	WES001	表示領域の初期化を要する
		理由	初期状態を確保する
		説明	表示領域を初期状態に戻す

表示領域

***	要求	WES002	天気予報ボタンが押されたときに天気予報を表示する
		理由	天気予報の表示を促す
		説明	天気予報ボタンを押したときに天気予報を表示する

天気予報ボタン

天気予報ボタン	要求	WES05	本日と明日の時間単位の天気予報と、その後の6日間の1日単位の天気予報をネットから入手して表示したい	
		理由	商品の仕入れや陳列方法の判断に使うため	
		説明		
			<表示フォーマット>	
		<input type="checkbox"/>	WES05.1	本日と明日の分は24時間分を4時間毎に区切って6マスで表示する
		<input type="checkbox"/>	WES05.2	その後の6日分は1日単位で6マスで表示する
			<各マスの表示データ>	
		<input type="checkbox"/>	WES05.5	本日と明日の一つのマスには以下のデータを表示する ・時間(4時間単位の最初の時間) ・天気記号 ・予想降雨量、ただし本日で過ぎた時間のマスには降雨実績を表示する ・予想気温
		<input type="checkbox"/>	WES05.6	その後の6日分の一つのマスには以下のデータを表示する
			<ネット経由でのデータの獲得>	
		<input type="checkbox"/>	WES05.10	天気ボタンの押下によって、ネットに接続し〇〇から天気データを手に入る
	<input type="checkbox"/>	WES05.11	本日の24時間分のデータから表示対象のデータを表示する	

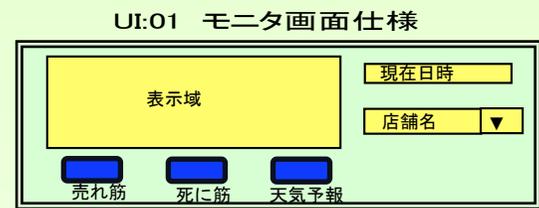
一般には、「動詞」が少なく、ほとんど「1階層」で対応可能

2.3 画面仕様を上手く表現するには

▶ 個別画面仕様 -3-

❖ 画面要素ごとに要求と仕様を表現する

❖ 個々の画面構成のすべての「要素」に対して「要求」を表現し、「仕様」を展開する



GUIツールで確認する項目などが有る時に付加

UI要素	要求	仕様	GUI
表示領域	UI:01	この領域の初期状態を定義	
	UI:02	表示領域の初期状態を定義	
天気予報ボタン	要求	WES05 本日と明日の時間単位の天気予報と、その後の6日間の1日単位の天気予報をネットから入手して表示したい	
	理由	商品の仕入れや陳列方法の判断に使うため	
	説明		
		<表示フォーマット>	
	□□□	WES05.1 本日と明日の分は24時間分を4時間毎に区切って6マスで表示する	●
	□□□	WES05.2 その後の6日分は1日単位で6マスで表示する	●
		<各マスの表示データ>	
	□□□	WES05.5 本日と明日の一つのマスには以下のデータを表示する ・時間(4時間単位の最初の時間) ・天気記号 ・予想降雨量、ただし本日で過ぎた時間のマスには降雨実績を表示する ・予想気温	●
	□□□	WES05.6 その後の6日分の一つのマスには以下のデータを表示する	●
		<ネット経由でのデータの獲得>	
	□□□	WES05.10 天気ボタンの押下によって、ネットに接続し〇〇から天気データを手に入る	
□□□	WES05.11 本日の24時間分のデータから表示対象のデータを表示する		

2.3 画面仕様を上手く表現するには

機能仕様との接続

- ❖ システムの仕様のほとんどが**画面仕様書**だけで表現できる場合、画面仕様を主体として画面仕様から機能要求仕様を参照するようにする
 - ❖ 画面仕様だけで対応出来ない場合、操作ボタンや画面表示の処理の中で、別に記述されている機能要求を参照するようにする
 - ❖ 別に記述されている機能の実行は、仕様や下位要求のレベルで接続する
 - ❖ 「画面仕様書」は、画面を起点にして**画面に直接関わることに専念**する

要求仕様			
	要求	DSA.01	計測したデータをリアルタイムで表示し、異常がある時はオペレータに知らせる
		理由	接続先の状況の変化に素早く対応する必要があるため
		説明	
	<通常表示>		
	□□□	DSA.01.1	計測したデータは、1レコードを1行に表示する
	□□□	<有効値計算>	
	□□□	DSA.01.10	有効値計算は、「機能仕様書 #VAL-03」を参照

仕様の中に直接記述

関連仕様書として記述

または

要求仕様			関連仕様書
	要求	UI.01.02	計測したデータをリアルタイムで表示し、異常がある時はオペレータに知らせる
		理由	接続先の状況の変化に素早く対応する必要があるため
		説明	
	<通常表示>		
	□□□	DSA.01.1	計測したデータは、1レコードを1行に表示する
	□□□	<有効値計算>	
	□□□	DSA.01.10	有効値計算を実施する
			#VAL-03

2.3 画面仕様を上手く表現するには

機能要求との連携

❖ システムの仕様が機能要求で構成されているなかで画面との連携が必要な場合、USDMMのフォームに「関連画面仕様」の列を追加して、画面要求仕様を参照するようにする

要求仕様			関連画面仕様
要求	ALM	設定した時刻になったら、好みのアラーム音を鳴らしてほしい	
	理由	...	
	説明	...	
要求	ALM01	アラームを鳴動させる時間を設定する	
	理由	...	
	説明	...	
<アラーム時刻の設定>			
<input type="checkbox"/>	ALM01.01	H ボタンが 1 回押される毎に、アラーム時刻に...	UI:001
<input type="checkbox"/>	ALM01.02	M ボタンが 1 回押される毎に、アラーム時刻に...	UI:001
<アラーム時刻の表示>			
<input type="checkbox"/>	ALM01.11	アラーム時刻設定中は、アラーム時刻を点滅させる	UI:001
要求	ALM02	好みのアラームを設定する	
	理由	...	
	説明	...	
<アラーム音の表示>			
<input type="checkbox"/>	ALM02.01	鳴らすことが出来るアラーム音のタイトルを...	UI:002
<input type="checkbox"/>	ALM02.02	選択できるアラーム音が10より多い場合は...	UI:002
<アラーム音の選択>			
<input type="checkbox"/>	ALM02.05	表示されたアラーム音のリストから...	UI:002

この列を追加する

連携する画面仕様書のIDを記述する

- 2. 1 要求を上手く表現するには
- 2. 2 要求を上手く仕様化するには
- 2. 3 画面仕様を上手く表現するには
- 2. 4 **非機能要求を上手く表現するには**

- ▶ 機能要求を実現することしか求めてこなかったために、「作り方の品質」を実現するスキルが身についていない

▶ 作り方の品質要求(非機能要求)の扱い

❖ 品質も「要求」と「仕様」を表現する

❖ 個々の品質特性に対して実現したいレベルや範囲を要求として表現する

❖ 品質要求に対して、それを実現するための仕様を展開する

❖ 「仕様」であることの条件は、通常の仕様と同じ

❖ しかし、保守性などの品質特性については、プログラムを実行して確認することはできない



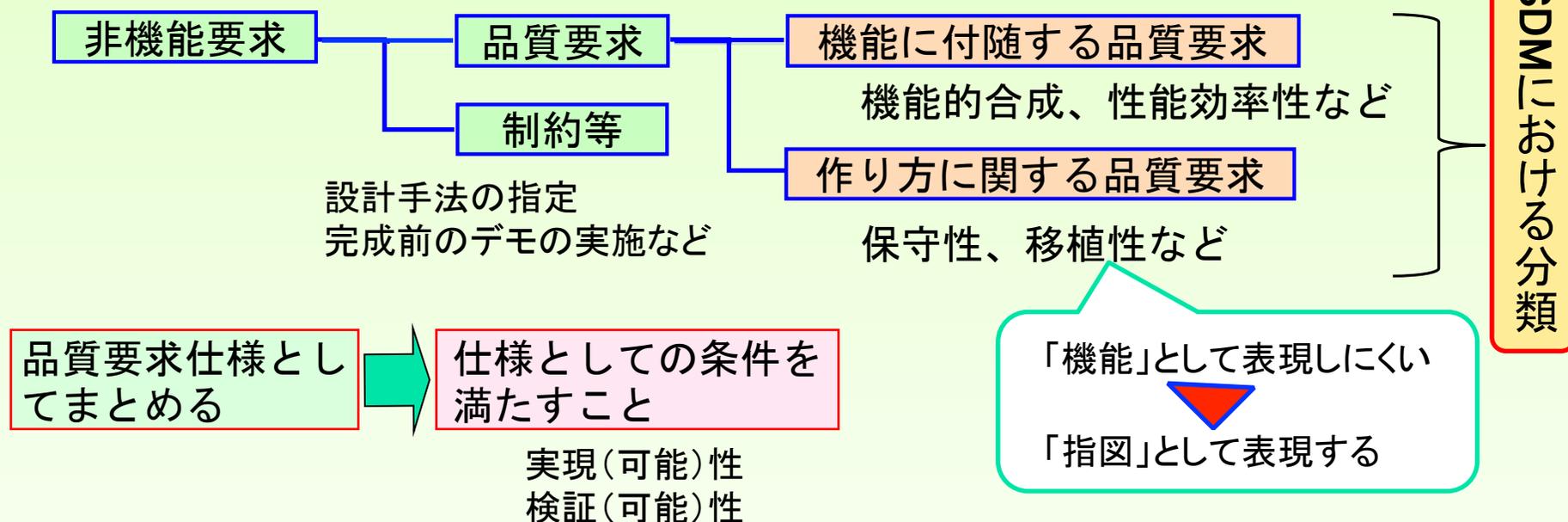
静的テストで
確認

2.4 非機能要求を上手く表現するには

➤ 非機能要求にもUSDMを適用する

❖ 非機能要求とは

- 非機能要求に含まれるもの



❖ 品質要求の内容によっては設計に大きな影響を与える

- ❖ 複数のモジュールが連携して実現することが多い
- ❖ モジュール構造やデータ構造の設計を間違えると要求は実現できない

➤ 品質要求の表現は難しい？

❖ 現実には、品質要求は明文化されていないことが多い

- パフォーマンスやセキュリティのような品質は表現しやすい
- 操作性や回復性も表現しやすい
- しかし解析性や変更容易性の表現は難しい

これって「機能仕様」の
イメージになっていませ
んか！

品質要求

機能に付随する品質要求

機能要求と同様に書ける

作り方に関する品質要求

作業員への「指示」として書く

USDMから
の提案

もともと「要求仕様書」は、作業員(今回の関係者)に
対して**作って欲しいことが書かれた文書**

2.4 非機能要求を上手く表現するには

➤ 品質要求の表現

❖ 品質要求も「要求」として表現する

- ❖ 要求として表現することで、品質要求を仕様化ができる
- ❖ 対象の品質特性を明記することで、関係者が同じイメージでレビューできる
- ❖ 「パフォーマンス」などでは、「目標」が数値で表現されることがある

「実現可能性」のレベルで表現する

品質特性

要求	DSP01-02	【時間効率性】「表示」ボタンを押してから2秒以内に画面の表示が完了すること
	理由	ここで表示するデータの処理量が多く、表示が遅いとハングアップしたと勘違いする可能性があるため
要求	QUA.02	【変更性】ソフトウェアの保守性を確保して変更簡単に崩れないシステムを作って欲しい
	理由	この後5年以上(30回以上)、ここから製品展開したいからデリバリを的確に実現することでシェアをひっくり返したい
要求	QUA.03	【障害許容性】接続する機器が故障しても、全体の動きを止めないで欲しい
	理由	工場のラインを制御している装置だから

2.4 非機能要求を上手く表現するには

▶ 非機能要求も適用範囲を考慮する

- ❖ ある「機能要求」のみに関係する「非機能要求」は、その機能要求（上位要求）の下位要求として捉える

要求	RQ1
	理由
	説明
<非機能要求: 時間効率性>		
要求	RQ1.1	検索結果が表示されるまでの時間を2秒以内にしてほしい
	理由	画面に表示されるまでの時間が長いとフリーズしたと思われるため
	説明	
□□□	RQ1.1.01	検索対象のデータ量がXXXXByteより大きい場合は、検索進展に対するプログレスメータを表示する
□□□	RQ1.1.02	検索対象のデータ量がXXXXByte未満の場合は、検索進展を示すプログレスメータは表示しない

- ❖ システムの「機能要求」に共通する「非機能要求」は、上位要求として独立させる

<非機能要求: 変更性>		
要求	RQ1	ソフトウェアの変更に対する耐久性(崩れにくさ)を向上させてほしい
	理由	この後7年間で50回以上製品のバージョンアップを行い商品展開をしたいから
	説明	製品の競争力維持が必要
<モジュールの設計>		
□□□	RQ1.01	モジュールの複雑度は17以下とし、超える場合は事前に了解をとること
□□□	RQ1.02	モジュールの凝集度が手順的凝集度以下になる場合は、事前に検討すること
□□□	QR1.03	処理と管理は明確に分離し、関数の呼び出し深さが6を超えるときは事前に了解をとること

2.4 非機能要求を上手く表現するには

- ▶ 非機能要求を機能で実現することもある
 - ❖ 非機能要求によっては、機能で要求を満たす場合がある
 - ❖ 例 交差点の安全性を高めたい
 - ⇒ 信号機, 立体交差
 - ❖ 時計の場合

		非機能要求仕様	
機能性	要求	ALM01	時刻の誤差はできるだけ小さくしてほしい
		理由	時刻のずれが大きくなると、時刻合わせを行わなければならないため
		説明	
<正確な時刻の維持>			
正確性	要求	ALM01.01	時刻の誤差は平均月差を10秒以内としてほしい
		理由	時刻のずれが大きくなると、時刻合わせを行わなければならないから
		説明	
<運針の正確性維持>			
	<input type="checkbox"/>	ALM01.01.1	水晶振動子を含む基準信号発生器から基準信号を受信する
	<input type="checkbox"/>	ALM01.01.2	基準信号をもとに運針信号を作成する
	<input type="checkbox"/>	ALM01.01.2	作成した運針信号を使用して秒針の運針を制御する
			:
時刻の校正	要求	ALM01.02	決められた時刻に、自動的に時刻を合わせてほしい
		理由	時刻を常に正確な状態に維持したいから
		説明	
<時刻の正確維持>			
	<input type="checkbox"/>	ALM01.02.1	標準電波の送信局から信号を受信する
	<input type="checkbox"/>	ALM01.02.2	複数の電波を受信している場合最も強度の強いものを選択する
	<input type="checkbox"/>	ALM01.02.3	受信した信号から現在時刻を抽出する
			:

2.4 非機能要求を上手く表現するには

品質要求の備え

❖ それぞれのシステムに応じて想定される「品質要求」を(JISの)品質特性を基に把握して自分たちの組織で使用する言葉で表現しておく

(副)品質特性	認識	要求表現事例	仕様表現事例
モジュール性	変更が他のコンポーネントへの最小の影響ですむように構成されている度合い	処理やデータ構造を変更した時に、影響が拡散しないように作って欲しい	
再利用性			

品質特性	機能適合性 (Functional Suitability)	性能効率性 (Performance efficiency)	互換性 (Compatibility)	使用性 (Usability)	信頼性 (Reliability)	セキュリティ (Security)	保守性 (Maintenability)	移植性 (portability)
副品質特性	機能完全性 (Functional completeness)	時間効率性 (Time behaviour)	共存性 (Co-existence)	適切度認識性 (Appropriateness recognizability)	成熟性 (Maturity)	機密性 (Confidentiality)	モジュール性 (Modularity)	適応性 (Adaptability)
	機能正確性 (Functional correctness)	資源利用性 (Resource utilization)	相互運用性 (Interoperability)	習得性 (Learnability)	可用性 (Availability)	インテグリティ (Integrity)	再利用性 (Reusability)	設置性 (Installability)
	機能適切性 (Functional appropriateness)	容量満足性 (Capacity)		運用操作性 (Operability)	障害許容性 (Fault Tolerance)	否認防止性 (Non-repudiation)	解析性 (Analysability)	置換性 (Replaceability)
				ユーザーエラー防止性 (User error Protection)	回復性 (Recoverability)	責任追跡性 (Accountability)	修正性 (Modifiability)	
				ユーザーインターフェース快美性 (User interface aesthetics)		真正性 (Authenticity)	試験性 (Testability)	
				アクセシビリティ (Accessibility)		(JIS X 25010、品質特性・副品質特性より引用)		

2.4 非機能要求を上手く表現するには

➤ 品質要求を仕様化する

- ❖ 品質要求も具体的な仕様として定義されないものは実現しない
- ❖ 必要な品質特性に応じて達成すべき要求と
- ❖ それが達成したといえるための仕様を記述する

- ここでいう「2秒」は目標(=要求)であって仕様ではない
- プログラムコード上に現れることはない

要求	DSP01-02	【時間効率性】「表示」ボタンを押してから2秒以内に画面の表示が完了すること
	理由	ここで表示するデータの処理量が多く、表示が遅いとハングアップしたと勘違いする可能性があるため
□□□	DSP01-02-1	画像処理するデータ量が「2MB」以内の時は、通常通りに処理する
□□□	DSP01-02-2	画像処理するデータ量が「2MB」を越えるときは、処理中は「プログレスバー」を表示する

- ❖ この仕様を満たせば、「2秒以内に表示が完了した」と“みなす”

「要求」と「仕様」の階層構造が「みなす」という概念を成立させる

2.4 非機能要求を上手く表現するには

➤ 保守性も仕様化しないと実現しない

- ❖ **保守性**や**移植性**の品質要求も、仕様化しないと実現しない
 - ❖ ただし、“**品質を織り込む**”**設計技術**が必要
 - ❖ 通常のテストで確認できないが“**仕様としての条件**”は同じで、実現していることの**検証は「静的テスト」**で行う

要求	QUA.02	【変更性】ソフトウェアの変更に対する耐久性(崩れにくさ)を向上させて欲しい
	理由	この後5年以上最低30回のバージョンアップをここから展開したいから
□□□	QUA.02-1	一つのクラスに含まれるメソッドは 10個以下とし、越える場合は事前に了解をとる (*2)
□□□	QUA.02-2	モジュールの複雑度(*1)は17以下を原則とし、越える場合は事前に了解をとる (*2)
□□□	QUA.02-3	モジュールの凝集度も“手順的凝集度”以下になる場合は事前に検討する (*2)
□□□	QUA.02-4	処理と管理は明確に分離し、関数の呼び出しの深さが5を越えるときは事前に了解をとる (*2)
□□□	QUA.02-5	タスク間でアクセスし合うグローバル・データを作らない。アクセス時間等による制限から、グローバル・データとなることが避けられない時は、品質検討会議で承認されること (*2) 【説明】割り込み処理との間でのみ共有するケースはこの制限を受けない

- ❖ この仕様を満たせば、「保守性」を満たしていると“**みなす**”

(*1)=マッケイブ (McCabe) の循環的複雑度 (Cycromatic Complexity)
(*2)=PLが対応したり、品質検討会議のような組織で対応する

▶ 画面操作の品質要求も忘れずに

❖ 画面操作によっては、**その画面特有の品質要求**が存在することがある

❖ 応答性

❖ 操作に対する**スムーズな反応**

❖ 入力データの表示の速やかさ

❖ 配色や操作性

❖ 操作画面として目が疲れないこと

❖ 画面内の項目の見易さや選択しやすさ

❖ 操作する人の年齢を考えた配色や操作性であること

❖ 適切に状況を伝えることで誤操作を防ぐ

1. USDMのおさらい
2. 上手く表現するために
3. **こんな場合には**
4. この表現は如何ですか

▶ 要求がうまく出てこない

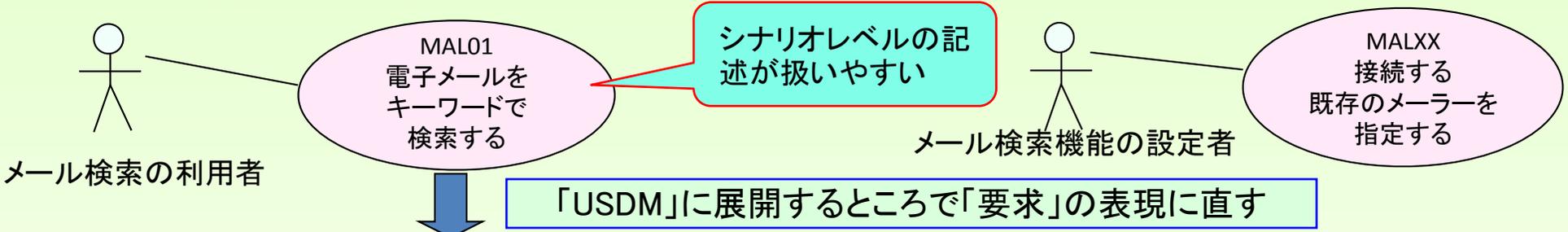
- ❖ 要求を引き出すために他の技術と連携する
 - ❖ ユースケースと連携
 - ❖ “状態遷移” から導出
 - ❖ 構造化分析と連携
 - ❖ マインドマップと連携
 - ❖ 「操作画面の要素」から導出

3. こんな場合には(要求がうまく出てこない)

ユースケースと連携

ユースケースと対応させる

- ある程度のところまでユースケース(=要求)で抽出を進めていき、途中でUSDMの表記に切り替える



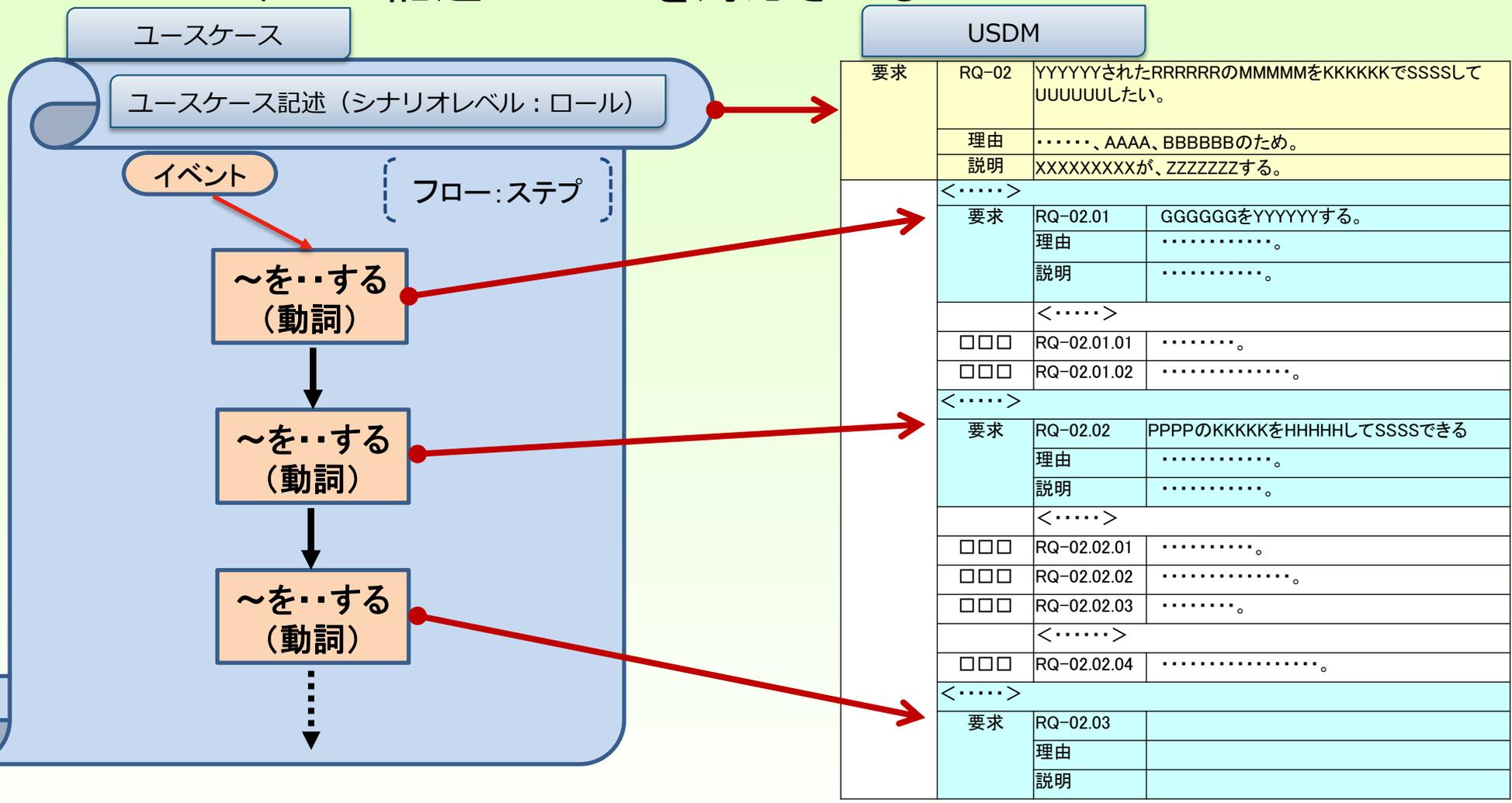
メール検索機能	要求	MAL01	受信および送信した電子メールをキーワードで検索して再利用したい
		理由	メールが多くて関連するメールを探せない
		説明	
			要求の下に仕様を引き出す

- ユースケース全体を上位要求に展開
- 各ステップは下位要求あるいはグループに展開
- 各ステップの実現方法を仕様に展開

3. こんな場合には(要求がうまく出てこない)

➤ ユースケースからUSDMへ

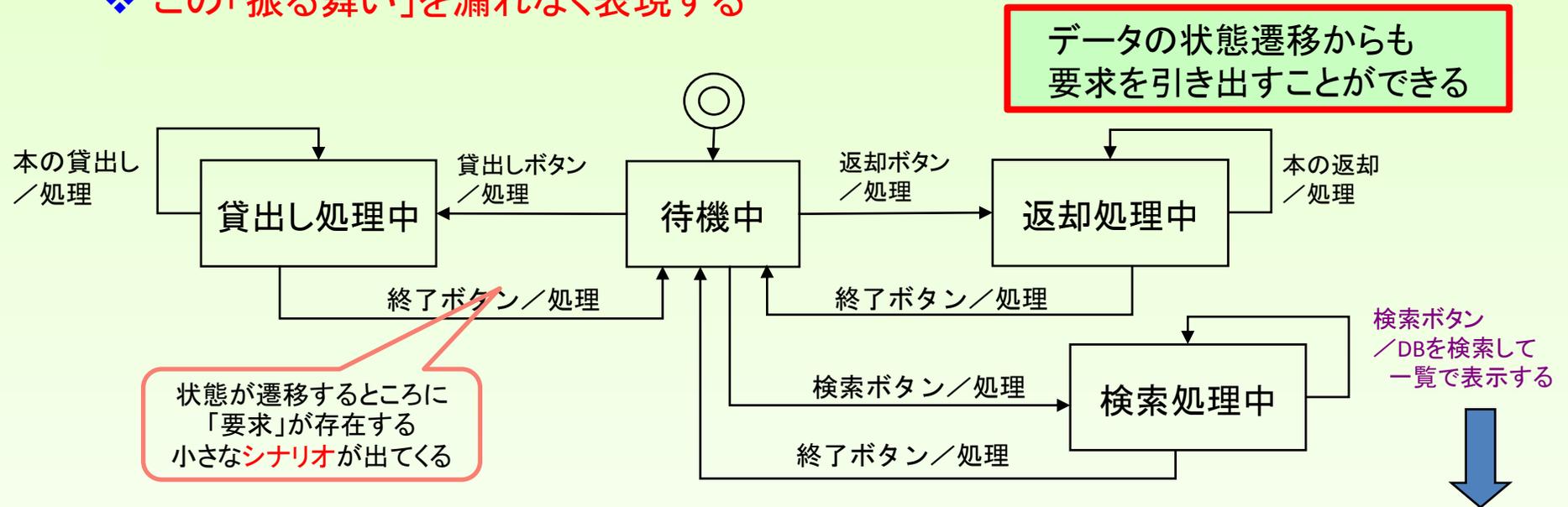
❖ ユースケース記述とUSDMを対応させる



3. こんな場合には(要求がうまく出てこない)

➤ 状態遷移”から導出

- ❖ 遷移する際に「振る舞い」を持つので、「要求」を引き出しやすい
 - ❖ 状態遷移のきっかけとなる“イベント”とそのイベントに付随する”振る舞い“がある
 - ❖ この「振る舞い」を漏れなく表現する



検索ボタン	要求	BOK01	入力された書籍情報を元に書籍DBを検索し、発見した書籍を一覧で表示する
	要求	BOK02	新たな書籍情報を使って、一覧で表示された中から表示を絞り込む
貸出しボタン	要求	BOK03	入力した書籍情報と個人情報から当該書籍を当該個人に対する「貸出し中」の状態にする
返却ボタン	要求	BOK04	入力した書籍情報から書籍DB内の当該書籍を返却済みとし、個人データから抹消する

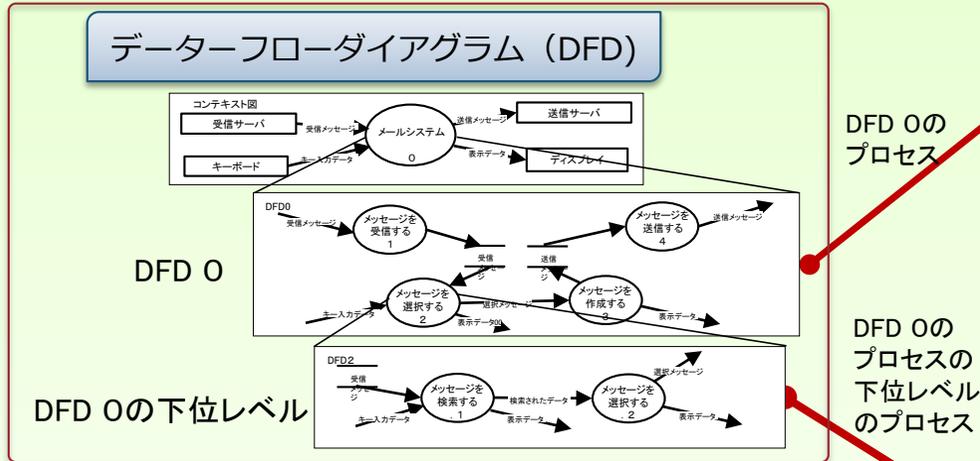
3. こんな場合には(要求がうまく出てこない)

➤ 構造化分析と連携

- ❖ データーフローダイアグラムのプロセスとUSDMを対応させる
- ❖ プロセスの名称では「振る舞い」の表現としては不適切なので編集が必要

構造化分析：

USDM



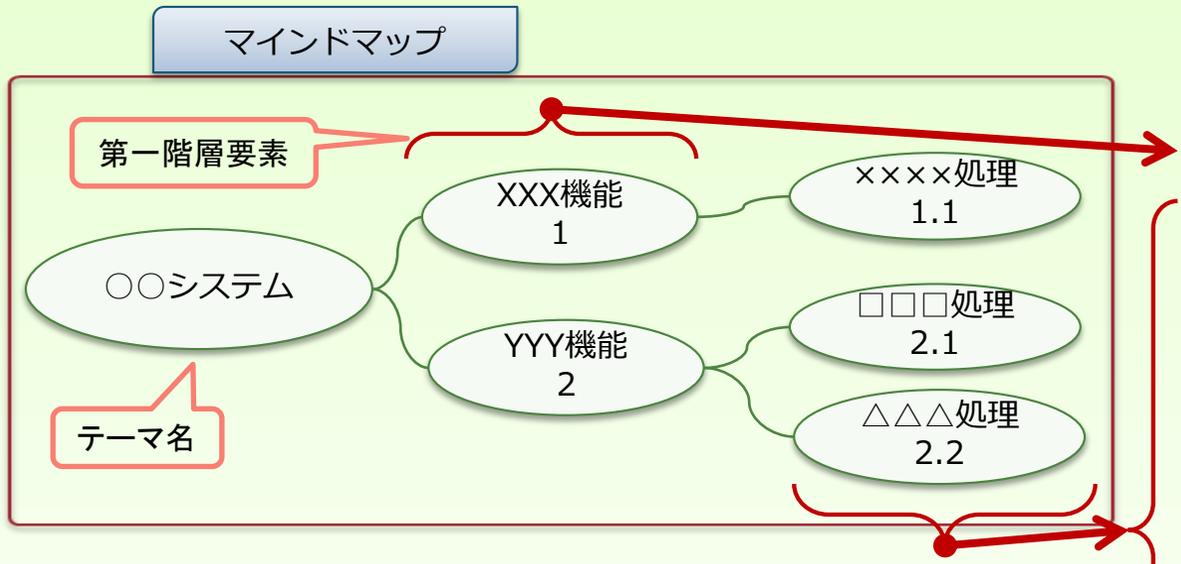
要求	RQ-02	YYYYYYされたRRRRRRのMMMMをKKKKKKでSSSSしてUUUUUUしたい。
理由	、AAAA、BBBBBBのため。
説明		XXXXXXXXXXが、ZZZZZZする。
<.....>		
要求	RQ-02.01	GGGGGGをYYYYYYする。
理由	。
説明	。
<.....>		
□□□	RQ-02.01.01。
□□□	RQ-02.01.02。
<.....>		
要求	RQ-02.02	PPPPのKKKKKをHHHHHLしてSSSSできる
理由	。
説明	。
<.....>		
□□□	RQ-02.02.01。
□□□	RQ-02.02.02。
□□□	RQ-02.02.03。
<.....>		
□□□	RQ-02.02.04。

- 一般的には
 - 第一階層の要求:DFD0に記載されているプロセス
 - 第二階層の要求:DFD0に記載されているプロセスの下位レベルとして記載されているプロセス
- 開発規模が小さくDFDに階層が無い場合
 - 第一階層の要求(DFD0に記載されているプロセス)だけの対応:第二階層が無いケース

3. こんな場合には(要求がうまく出てこない)

➤ マインドマップと連携

- ❖ 第一階層の要求: テーマ名から派生した第一階層の要素(上位要求)
- ❖ 第二階層の要求: 第一階層の要素から派生した次の階層の要素
- ❖ 要素の表現では「振る舞い」の表現としては不適切なので編集が必要



USDM

要求 (YYY機能)	RQ-02	YYYYYYされたRRRRRRのMMMMMをKKKKKKでSSSSしてUUUUUUしたい。
理由	、AAAA、BBBBBBのため。
説明		XXXXXXXXXが、ZZZZZZする。
<.....>		
要求 (□□□処理)	RQ-02.01	GGGGGGをYYYYYYする。
理由	。
説明	。
<.....>		
□□□	RQ-02.01.01。
□□□	RQ-02.01.02。
<.....>		
要求 (△△△処理)	RQ-02.02	PPPPのKKKKKをHHHHHしてSSSSできる
理由	。
説明	。
<.....>		
□□□	RQ-02.02.01。
□□□	RQ-02.02.02。
□□□	RQ-02.02.03。
<.....>		
□□□	RQ-02.02.04。

- ❖ 同一面に全ての要素が書かれているので
 - ① 階層の検討がしやすい
 - ② グループの検討がしやすい

3. こんな場合には(要求がうまく出てこない)

➤ 「操作画面の要素」から導出

❖ 画面に配置された表示領域やボタンなどの要素も「振る舞い」を持つ



一般の要求仕様書と同じパターンで表現できる

- ❖ 画面も「状態」の一種
- ❖ 状態遷移の考え方で「画面遷移」を作り、そこから要求を引き出すことができる
- ❖ 画面レイアウトの実現も「振る舞い」として表現する
- ❖ 画面に配置されたボタンの押下(選択)をトリガーとして遷移する画面に着目して「画面遷移」を作成する
- ❖ 画面に配置されたボタンに付随する処理を「振る舞い」として表現する

3. こんな場合には

- ▶ 要求がうまく分割階層化できない
 - ❖ 上位要求の表現が「動詞」「目的語」を使った「振る舞い」になっているか
- ▶ 振る舞いにはなっていると思うが下位要求が……
 - ❖ 表現されている“振る舞い”の記述が適切でない
 - | | | | |
|--|----|------|-------------|
| | 要求 | DA07 | 計測データを表示できる |
|--|----|------|-------------|

(「名詞＋する・できる」の組合せで表現しただけ)

 - ❖ 振る舞いのように表現されてはいますが……
 - ❖ 理由が要求とマッチしているか
- ▶ このような場合次の階層を展開するのは容易ではない
 - ❖ 表現されている“振る舞い”を洗練させる

▶ 下位の要求多くなりすぎる

❖ 上位要求の要求範囲が曖昧

- ❖ どの範囲まで記述してよいかリミットが分からない。

- ❖ 適切な表現で要求の範囲を明確にする

- ❖ この状態だと振舞の境界が見えにくくなり多くの要求が発生する

❖ 上位要求の範囲が広すぎる

- ❖ 上位要求の範囲を狭くする

- ❖ 上位要求にもグループに分ける（範囲が狭くなる）

3. こんな場合には

▶ 階層が深くなってしまふ

- ❖ 要求と要求仕様は階層を成して連鎖している
- ❖ レベルの異なる要求が混ざっているかも
 - ❖ レベルをそろえる
- ❖ 最上位の要求の抽象度が高すぎる場合
 - ❖ 振る舞いとしての表現が抽象すぎる
 - ❖ 詳細化するために階層が深くなる
 - ❖ 上位要求を分割して範囲を狭くする
 - ❖ そのうえで「振る舞いの抽象度」を下げた表現にする
 - ❖ 場合によっては上位要求にもグループの考え方を導入する
- ❖ 下位要求に分割基準をそろえるためだめの要求がある
 - ❖ <グループ>を適用して要求の粒度をそろえる

➤ USDMの展開

- ❖ 作成したUSDMによる要求仕様を仕様の明確化だけに終らせずこれと連携したソフトウェアエンジニアリング成果物に構成する
 - ❖ 要求仕様書＋スケジュール
 - ❖ 要求仕様書＋TM
 - ❖ 要求仕様書＋進捗
 - ❖ 要求仕様書＋追跡
 - ❖ 変更要求仕様書＋TM

3. こんな場合には

➤ USDMの展開 ❖ 要求仕様書＋進捗

要求仕様 進捗確認 (成果物一覧表 業 見取り表 業 進捗確認表)

プロジェクト名:

日付:

パターン1(要求に対して仕様が展開されるケース)

カテゴリー名 (記号)	要求 (要求番号)	理由	説明	備考欄	レビュー結果	収録モジュール名	見直しボリューム (コード行数)	実用ボリューム (コード行数)	消去行数	新規行数	変更・修正行数	見直し不具合件数	実不具合件数	設計 担当	工数	作業開始期日 (予定)	作業終了期日 (予定)	作業開始期日 (実績)	作業終了期日 (実績)	確認	
カテゴリー名 (記号)	要求 (要求番号)		ここに要求を記述します。		<input type="checkbox"/> OK <input type="checkbox"/> NG																
	理由		要求の背景や理由について記述します。																		
	説明		要求について必要に応じて説明してください。仕様とは見做されません。セリを広げて、図を貼っても良いです。																		
	<<仕様分類名>>		主分割記号・全体を通して共通の分割単として決める。(A77777)																		
	<<前提条件>><<内部処理>>																				
	<仕様分類名>		補助分割名・主分割の中に異なるテーマの仕様が混在する場合に補助分割記号を使って、純度の高い単を作る																		
	<input type="checkbox"/> <input type="checkbox"/> (仕様番号)	要求 (要求番号)		上記の要求に含まれるべく要求仕様を記述します。必要に応じて<>>でグループ分けしてください!		<input type="checkbox"/> OK <input type="checkbox"/> NG															
	<input type="checkbox"/> <input type="checkbox"/> (仕様番号)	理由				<input type="checkbox"/> OK <input type="checkbox"/> NG															
	<input type="checkbox"/> <input type="checkbox"/> (仕様番号)	説明				<input type="checkbox"/> OK <input type="checkbox"/> NG															
	<input type="checkbox"/> <input type="checkbox"/> (仕様番号)	<仕様分類名>				<input type="checkbox"/> OK <input type="checkbox"/> NG															

パターン2(要求がもう一段階層化されるケース、これ以上の階層化は避けたほうが良いです)

カテゴリー名 (記号)	要求 (要求番号)	理由	説明	備考欄	レビュー結果	収録モジュール名	見直しボリューム (コード行数)	実用ボリューム (コード行数)	消去行数	新規行数	変更・修正行数	見直し不具合件数	実不具合件数	設計 担当	工数	作業開始期日 (予定)	作業終了期日 (予定)	作業開始期日 (実績)	作業終了期日 (実績)	確認	
カテゴリー名 (記号)	要求 (要求番号)		ここに要求を記述します。		<input type="checkbox"/> OK <input type="checkbox"/> NG																
	理由		要求の背景や理由について記述します。																		
	説明		要求について必要に応じて説明してください。仕様とは見做されません。セリを広げて、図を貼っても良いです。																		
	要求 (要求番号)	理由	ここに階層化された要求を記述します。要求番号は1段下げられます 範囲を狭めた要求について背景や理由を記述します。		<input type="checkbox"/> OK <input type="checkbox"/> NG																
	説明		要求について必要に応じて説明してください。仕様とは見做されません。																		
	<<仕様分類名>>		主分割記号・全体を通して共通の分割単として決める。(A77777)																		
	<<前提条件>><<内部処理>>																				
	<仕様分類名>		補助分割名・主分割の中に異なるテーマの仕様が混在する場合に補助分割記号を使って、純度の高い単を作る																		
	<input type="checkbox"/> <input type="checkbox"/> (仕様番号)					<input type="checkbox"/> OK <input type="checkbox"/> NG															
	<input type="checkbox"/> <input type="checkbox"/> (仕様番号)					<input type="checkbox"/> OK <input type="checkbox"/> NG															

1. USDMのおさらい
2. 上手く表現するために
3. こんな場合には
4. この表現は如何ですか

4. この表現は如何ですか

➤ さあ、皆さんで考えてみましょう
これはいかがでしょうか？

誤差補正処理要求仕様				優先度
ユースケース		ER.001	計測データの補正を行うための誤差補正データを算出する。	
□□□	要求	CAR.01	RL情報/OD情報/SP情報/TW情報を入力測情報として誤差補正データを算出し、誤差補正処理に提供すること	
		理由	経時変化や温度変化に起因する情報の誤差をソフトウェアで補正することにより、精度の高い計測データを維持するため。	
		説明	振動などによる計測器のズレや温度によるセンサーのゆがみを原因とした計測データの誤差をソフトで吸収し、常に精度の高い計測性を維持するための補正データを算出する。	
□□□	要求	CAR.01.01	起動時にAFF110から補正データ類を読み込むこと	
		理由	補正処理の動作に必要な補正值(補正データ)がAFF110のメモリに保存されているため。	
		説明	計測領域の情報や、最新(前回終了時)の誤差補正データなど誤差補正処理に必要な補正データ類を取得する。	
□□□	仕様		仕様の件名	
		□□□ CAR.01.01.01	AFF110から自動誤差補正制御データを取得する。 (補正データ一覧) WW値 HH値 VRS値 VRE値 HHS値 HHE値 RAN値 LDV値	
		□□□ CAR.01.01.02	AFF110から誤差補正データを取得する。	
		□□□		
□□□	要求	CAR.01.02	X11から取得したRL情報からVer誤差補正計算を行い、算出された誤差補正データをX11とAFF110に保存する。	
		理由	Hor誤差補正はVer情報が補正されている必要が有るため、Ver誤差補正を先に行う。	
		説明	HoR情報、VrP情報、VrR誤差、LiR誤差の補正を行い算出結果をX11とAFF110に渡す。	
□□□	仕様		仕様の件名	
		□□□ CAR.01.02.01	RL情報をX11から取得する。	
		□□□ CAR.01.02.02	Ver誤差補正計算を行う。	
		□□□ CAR.01.02.03	計算結果の誤差補正データをX11とAFF110に保存する。	
		□□□ CAR.01.02.04	X11とAFF110に保存されている誤差補正データと計算結果に差がない場合はX11とAFF110への保存は行わない。	
		□□□		



➤ 「仕様の問題」は要求モレであって仕様モレではない

ゴールの範囲がイメージできる状態

適切な範囲に限定された状態

分割基準が共有できる状態

並びが整理された状態

動詞がすべて表現された状態

で「要求」が表現されれば
仕様モレはほとんど起きない

- ❖ 仕様モレの多くは「要求モレ」や「要求の動詞抽出モレ」である
- ❖ 「要求」を表現しない状況では、すべて“仕様モレ”と認識されている
- ◆ 仕様モレはUSDMで解決できるが、曖昧表現は完全には解決できない
 - ◆ 名文ではなく、明文を！

- 品質を織り込むには要求に合った技術が必要
 - ❖ 品質要求は、機能を実現する設計技術とは別の設計技術が必要
 - ❖ 品質要求が課されなかったことで、これらの知識や技術の習得を怠った

知識や技術	有効な品質要求
<ul style="list-style-type: none"> ・データ構造 ・アルゴリズム 	<ul style="list-style-type: none"> ・性能効率性などの性能に関する品質要求
<ul style="list-style-type: none"> ・モジュールの分割基準 ・モジュールの尺度(凝集度、結合度、複雑度など) ・クラスの分割基準(デザインパターンなど) ・クラスの尺度 	<ul style="list-style-type: none"> ・保守性や移植性などの品質要求
<ul style="list-style-type: none"> ・マルチタスクシステムに於いて、タスクの外にグローバルデータを出さない設計技術 ・「Switch文」を使わない状態遷移の設計技術 	<ul style="list-style-type: none"> ・保守性、移植性、性能効率性に関する品質要求

- 清水吉男：要求を仕様化する技術・表現する技術
（技術評論社）【改訂第2版】
- 清水吉男：AFFORDD勉強会資料「USDMによる要求仕様の書き方」【text_usdmV13.pdf】



- (株)eXmotion：USDMトレーニング教材「USDM実践トレーニング」

御清聴ありがとうございました

『そうだったのか！USDm』

梶本 和博

派生開発推進協議会
株式会社エクスマーシオン