

# USDM 小冊子

## 基礎編



ver 1.3

2016年5月10日

# 小冊子の構成

USDМ 入門(T2)が作成している小冊子の構成は、次のとおりです。



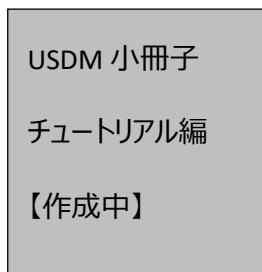
- … 本書です。  
USDМ を作成する上で、基礎的なことが記載しています。  
最初に読んで頂きたい小冊子です。



- … USDМ を作成する手順を、簡単にまとめた小冊子です。  
必要最低限の情報のみ記載されています。  
リファレンスとして参照してください。



- … USDМ の作成についての応用や解説をまとめた小冊子です。  
「要求の分割」、「仕様から要求の立て方」、「理由の聞き出し方」などが記載する予定です。現在作成中です。



- … USDМ の作成について、チュートリアルの小冊子です。  
「要求のまとめ方」から「仕様の書き出し」までの手順をまとめたものです。現在作成中です。

本書は、(株)システムクリエイツの清水吉男氏が著した『要求を仕様化する技術 表現する技術』の改定第 2 版をもとにしています。本格的に展開する場合は、この本を使って読書会をすることを推奨します。

# 目次

1	前提条件.....	3
1.1	前提知識について.....	3
1.2	ソフトウェア開発について.....	3
1.3	小冊子の対象者および登場人物について.....	3
1.4	USDM を作成する人について.....	3
1.5	USDM を取り入れていないソフトウェア開発について.....	3
2	USDM とは何か.....	4
2.1	USDM とは.....	4
2.2	USDM の特徴.....	6
3	USDM のメリット.....	14
3.1	USDM のメリットについて.....	14
4	USDM の書き方.....	18
4.1	基本フォーマット.....	18
4.2	要求の書き方.....	20
4.3	理由の書き方.....	23
4.4	仕様の書き方.....	24
4.5	USDM 作成時の注意点.....	26
5	USDM の取り組みの糸口.....	28
5.1	取り組みのポイント.....	28
5.2	どこから手をつければ良いか.....	28

# 1 前提条件

---

この小冊子を読む前に、次のような前提条件があります。

## 1.1 前提知識について

・ソフトウェア開発における「要求」、「仕様」、「設計」の関連した知識が必要です。

## 1.2 ソフトウェア開発について

コンピュータでのソフトウェアの開発のことで、ソフトウェアを作成することを指します。

「組み込み系」や「エンタプライズ系」などの「ソフトウェア開発」の分け方がありますが、ここでは特に意識していません。

## 1.3 小冊子の対象者および登場人物について

・「依頼者」・・・ソフトウェア開発を発注する人。企業や組織を代表している人。業務に精通している人。1人または複数人。

・「ユーザー」・・・ソフトウェアを使用する人。1人または複数人。

・「作業員」・・・ソフトウェアの開発を行う人。

## 1.4 USDM を作成する人について

・「作業員」を想定しています。場合によっては「依頼者」が作成することもあります。少数です。

## 1.5 USDM を取り入れていないソフトウェア開発について

・「現状のソフトウェア開発」と表現しています。「現状」および「現行」は「現状」と表現しています。

# 2 USDMとは何か

ここでは USDM(Universal Specification Describing Manner の略)の概要や特徴について説明します。

また、USDM ができた背景やメリットについても説明します。

## 2.1 USDMとは

～USDMとは・・・

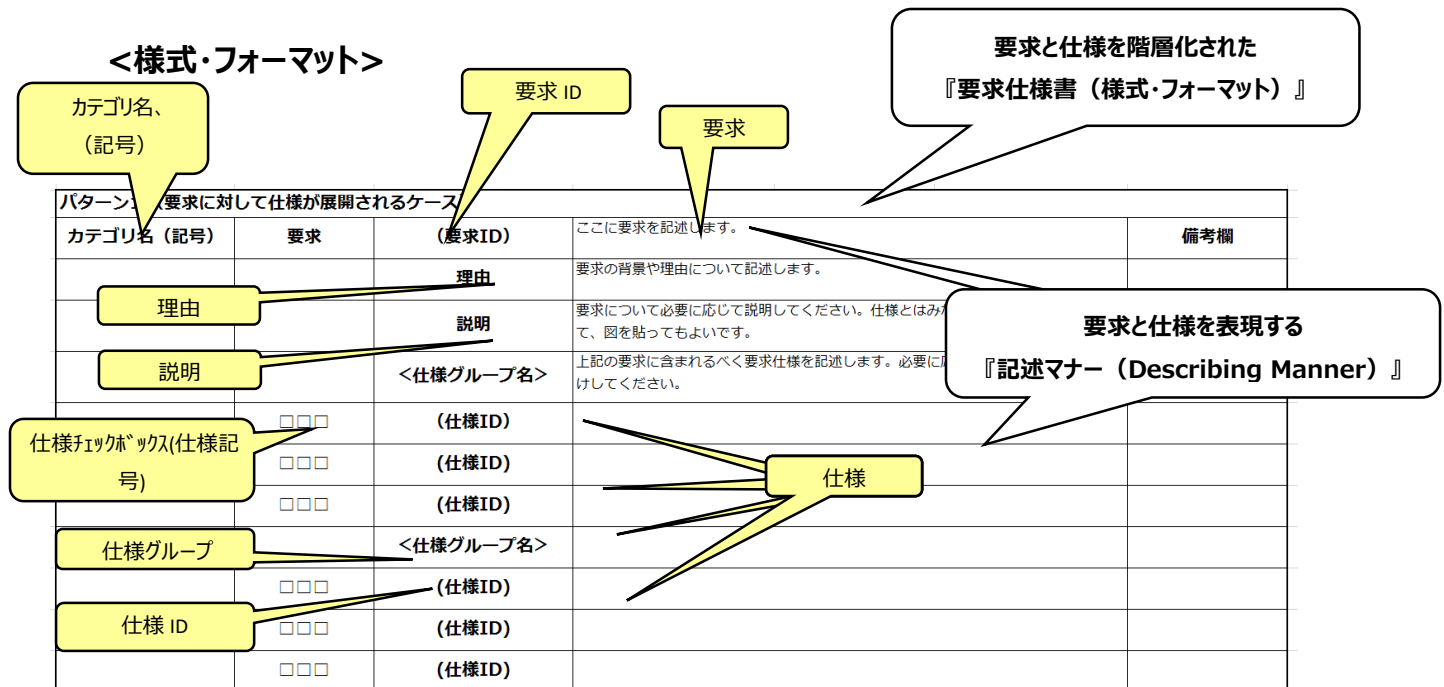
### 2.1.1 USDMの概要

USDMは、(株)システムクリエイツの清水 吉男 氏が提唱する表記法です。

USDMは、

- 要求と仕様を階層化された構成に記述するための『要求仕様書（様式・フォーマット）』
- 要求と仕様を表現する『記述マナー（Describing Manner）』

から成ります。



パターン2（要求がもう一段階階層化されるケース。これ以上の階層化は避けたほうがよい）				
カテゴリ名（記号）	要求	（要求ID）	ここに要求を記述します。	
		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。仕様とはみなされません。セルを広げて、図を貼ってもよいです。	
		要求	（要求ID）	ここに階層化された要求を記述します。要求IDは一段下げられます。
			理由	範囲を狭めた要求について背景や理由を記述します。
			説明	要求について必要に応じて説明してください。仕様と見なされません。
		<<仕様グループ名>>		主分割記号・・・全体を通して共通の分割基準として決めます（オプション）。<<前提条件>><<内部処理>>
			<仕様グループ名>	補助分割名・・・主分割のなかに異なるテーマの仕様が存在する場合に補助分割記号を使って、純度の高い集合を作ります。
		□□□	（仕様ID）	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。
		□□□	（仕様ID）	
		□□□	（仕様ID）	
			<仕様グループ名>	
		□□□	（仕様ID）	
		□□□	（仕様ID）	
		□□□	（仕様ID）	

### <記述例>

売上分析 (URIAGE)	要求	URIAGE-01	画面のドロップダウンリストで店舗を指定するところから、店舗コードを取得し、取得した店舗コードをもとに売上テーブルを検索し、検索した結果を編集し、編集したデータをもとに売上一覧表を表示するところまで行う。
		理由	店舗ごとの売上の傾向を知りたい。それで売上が少ない店舗を対策を早めに立てたい。売れ筋の商品を知りたい。
		説明	実装時期は、10月ごろ。
		<ドロップダウンリストの表示>	
	□□□	URIAGE-01.001	店舗テーブルを次の条件で検索し、店舗コードと店舗名を取得する。 ・廃止した店舗（廃止フラグが1のもの）は除く ・有効期限内（本日が、開店日と終了日の間にある） ・店舗コードの昇順
	□□□	URIAGE-01.002	取得した店舗コードは、ドロップダウンリストの値として設定する。
	□□□	URIAGE-01.003	取得した店舗名は、ドロップダウンリストの表示名として設定する。
		<ドロップダウンリストの指定後>	
	□□□	URIAGE-01.101	ドロップダウンリストの指定が発生した場合（changeイベントが発生した場合）に、売上一覧表処理を実行するようにする。

### 2.1.2 USDM が出来た背景

ソフトウェア開発での不具合の大半は上流工程で発生しますが、不具合を上流工程で見つけることは、難しいです。下流工程（結合テスト以降）で仕様漏れを発見した場合、対応コストは上流と比較して 10 倍から 30 倍といわれています(参考: <http://www.ipa.go.jp/files/000005272.pdf>)。

清水氏は、上流工程で不具合が混入しやすく、発見し辛い理由は、要求仕様書のフォーマットに明確な基準がなく曖昧な表記を許しすぎている事にあると考えました。

厳密な仕様記述の記法としては形式手法がありますが、USDM はより一般的なツール（Excel）で気軽に導入でき、仕様漏れを防ぐ効果も高い技法です。

また、ソフトウェア開発の多くは新規開発ではなく既存システムの一部改修、すなわち「派生開発」です。

派生開発における品質向上のための技法「XDDP」における中心成果物には、USDM の記法が利用されています。「XDDP」を用いることで、担当者がそのシステムを完全に把握しない状況でも、品質を守りながら最小の時間で開発を完了させることができるようになります。

## 2.2 USDM の特徴

### 2.2.1 「要求」を表現します。

USDM の特徴は、要求仕様書に「要求」を記述して表現することです。（補足：囲み参照）

USDM の要求仕様書の 様式・フォーマット は、「要求」と「仕様」を階層化の構造の中で 1 つのドキュメントに記述していきます。（2.1.1 <様式・フォーマット> 参照）

このため、仕様化の作業の中で、すぐに要求を確認することができ、作業を進められます。

また、仕様がその要求の範囲を満たしているか確認することができ、漏れがあることに気づきやすくなります。

#### [補足]

要求を要求仕様書にまとめて、要件（仕様）を要件定義書に別々にまとめるドキュメント体系を採用している組織の方は、USDM の要求仕様書は要件定義書にあたるものと思ってください。

あるいは、「“要求仕様書”に要件（仕様）を書き加えることで、要件定義書の役割も持たせる」というイメージの方がつかみやすいかもしれません。

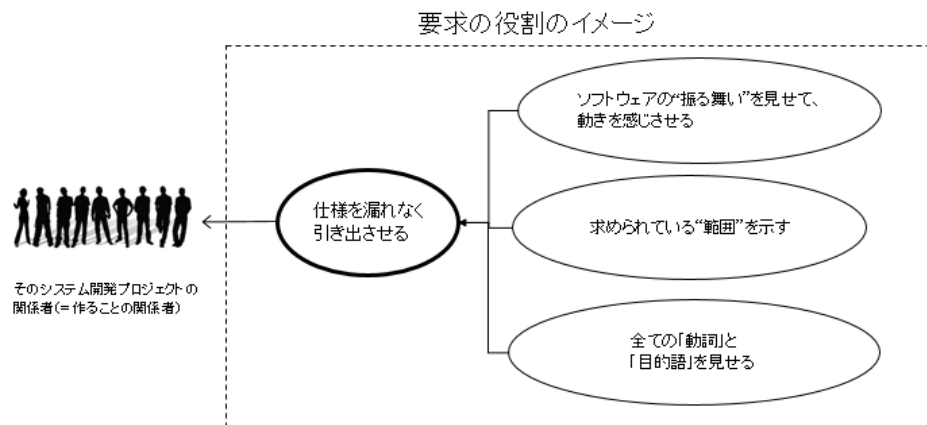
## 2.2.2 「要求」を表現する理由 = 「要求」の役割

USDM で、「要求」を要求仕様書に表現する理由は、“仕様を漏れなく引き出させる”という重要な役割を「要求」に持たせているためです。次の 3 つの役割を「要求」に担わせることで、その役割を果たさせています。

- 要求されている“ソフトウェアの振る舞い”を見せて、動きを感じさせる役割
- 要求されている“範囲”を示す役割
- 要求に含まれる全ての“動詞”と“目的語”を見せる役割

「要求」を表現するときは、この 3 つの役割を果たせるようにしてください。

<図 2>



## 2.2.3 「要求」されているソフトウェアの“振る舞い”を見せて、動きを感じさせるように表現します。

- 要求されているソフトウェアの“振る舞い”を表現します

要求とは、依頼者が求めている“ソフトウェアで実現して欲しい動作（処理）”のことです。

USDM では、依頼者が求めているソフトウェアの動作の一連の流れ、すなわちソフトウェアの“振る舞い”を文章にして表現します。

この“振る舞い”で表現する文章には、

“イベントに始まり、「入力処理」→「変換処理」→「出力処理」までの一連の処理）を行って止まるまで”

の範囲でソフトウェアの動作の一連の流れを盛り込みます。5 ページの URIAGE-01 の例のように、要求中に含まれる動詞をすべて書ききるようにします。

“振る舞い”を表現するときは、1 つ 1 つの動作を、具体的な動詞で表現します。



また、“振る舞い”で表現された一連の動作には、入力処理から出力処理までの動作が含まれますが、それぞれの動作に着目すると、入力処理に係る動作には入力データが、出力処理に係る動作には出力データがあります。

例えば

計測データを受信し、10件ごとに平均値を計算して、画面に表示する。

という“振る舞い”の場合には、

- ・ “計測データ”が、入力データ
- ・ “平均値”が、出力データ

となります。こうした入力データや出力データは、文章では目的語”で表現されます。

このように、“振る舞い”を文章で表現するときは、動作を目的語と動詞をセットにします。

具体的には、

目的語 動詞  
「～～を～～する。」

となります。

“振る舞い”は、ソフトウェアの動作の一連の流れですので、それを文章にすると目的語と動詞が交互に連鎖した形で表現されます。具体的には、

「～～を～～して、～～を～～して、… ～～を～～して、～～を～～する。」

となります。

➤ 動きを感じさせるように表現します

動きを感じさせるためには、“動詞形”で表現することが重要です。

次の2つの例を見比べて、感じ方の違いを実感してみてください。

例1 平均値を表示する

例2 平均値の表示

例1では、動詞が“表示する”と動詞形で表現されています。例2では、動詞が“表示”と名詞形で表現されています。

どちらも意味は同じですが、例1「表示する」のように動詞形にすることで、動きを感じさせることができます。

例 2「表示」のように名詞形で書いてしまっは、動きを感じさせることはできません。  
 要求を表現するときは、必ず“動詞形”で表現します。

➤ 要求を“振る舞い”で表現し、“動詞形”で表現する理由は、動きを感じさせることが狙いです。

要求を“振る舞い”で表現し、“動詞形”で表現して動きを感じさせるようにすることで、ユースケース図や P F D (Process Flow Diagram の略) のようなことを、要求仕様書に持たせることができます。

〔 通常の要求仕様書で見られる要求の表現 〕

2.5 画像拡大機能

- ・ 画像表示域の下に「スライダー」がある。
- ・ スライダーの「ボタン」を右に移動すると、画像を移動した分だけ拡大する。
- ・ 逆に「ボタン」を左に移動すると、画像を移動した分だけ縮小する。

「画像拡大機能」という機能カテゴリのタイトル (小見出し) があって、その下にいきなり仕様が“散文”の中で表現されている。  
 要求が書かれていないため、動きの流れを感じさせることはできない。  
 タイトル (小見出し) は、その機能の呼び名でしかなく、要求の 3 つの役割を果たせない。

〔 USDМ での要求の表現 〕

画像表示 (Dpic)	要求	Dpic30	画面に表示された画像を、スライダーの操作によって、任意の大きさに拡大・縮小する。
		理由	画像の一部分を拡大して細部を見たり、逆に画像を縮小して全体像を見たい。
		説明	
		<スライダーの表示>	
	□□□	Dpic30.10	画像表示域の下に「スライダー」がある。
		<画像の拡大>	
	□□□	Dpic30.20	スライダーの「ボタン」を右に移動すると、その移動分だけ画像を拡大する。
		<画像の縮小>	
	□□□	Dpic30.30	スライダーの「ボタン」を左に移動すると、その移動分だけ画像を縮小する。

要求を“振る舞い”で表現することで、そして、動作を“拡大する”、“縮小する”といった“動詞形”で表現することで動きが感じられるようになり、仕様が引き出しやすくなる。  
 また、常に要求を見ながら、仕様を引き出す作業ができるため、要求にそぐわない仕様や、仕様のモレがあった場合にも気付きやすい。

#### 2.2.4 求められている“範囲”を示す。

要求にまつわる範囲には、

- 動作（処理）の範囲
- データの範囲

などがあります。

- 動作の範囲を示す

動作（処理）の範囲とは、“振る舞い”で表現された一連の動作の流れについて、“何々の動作”をするところから始まって、“何々の動作”をして終わるまでをいいます。

例えば

計測データを受信し、10件ごとに平均値を計算して、画面に表示する。

という“振る舞い”（要求）の場合には、動作の範囲は、

- ・ “計測データを受信する”ところから、動作が始まって、
- ・ “（平均値を）画面に表示する”ところまで、動作が終わり

となります。

- データの範囲を示す

データの範囲とは、入力処理や出力処理で扱われる入力データや出力データについて、その上限値や下限値、あるいは正常値と異常値の基準などのことをいいます。

例えば

天気予報をネットから入手して、最初の2日分は時間単位で表示し、その後の5日分はそれぞれ1日単位で予報を表示する。

という“振る舞い”（要求）の場合には、「天気予報」のデータについて、

- ・ 「1週間分の予報データ」が求められていて、
- ・ 「最初の2日分は、時間単位の予報データ」が求められ、
- ・ 「その後の5日分は、1日単位の予報データ」が求められていること

が範囲として示されています。

### 2.2.5 「要求」と「理由」をセットにして表現する。

依頼者が発する「要求」には、それが必要な「理由」（背景）があります。

また「理由」は「要求」に比べ変化することが少ないです。

要求と合わせて、その理由をつかんでおくことで、

- 「要求」の意味が理解しやすくなる。
- 作ることの依頼者や作業者の「要求」に対する認識のズレを調整できる。
- 「要求」に根拠がないことに気付くこともできる。
- 「仕様」の引き出しや設計方法に配慮できることがある。
- 依頼者も気が付いていない要求を引き出せることもできる。

などのメリットが得られます。

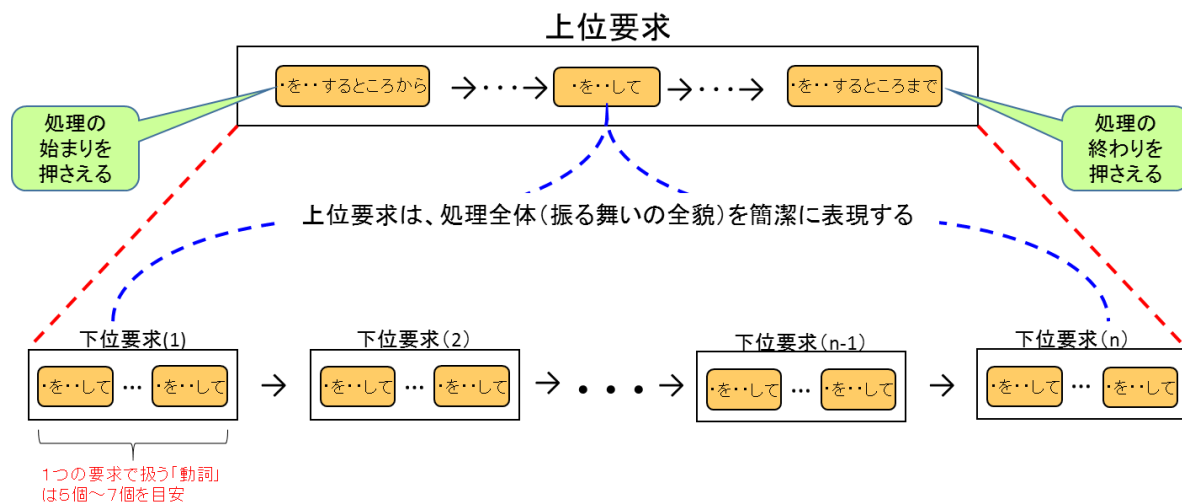
### 2.2.6 「要求」を分割して範囲を区切り、階層化する…全ての“動詞”と“目的語”を引き出す。

要求の範囲が広いと感じたときは、要求されている“振る舞い”を分割して範囲を区切り、階層化します。

要求を分割して階層化することで、範囲が狭められて限定された要求の集まりに変換します。

その範囲の中に含まれている動きを全て引き出して動詞と目的語で表現します。

範囲が限定された中で作業することで、動きが見えるようになり、モレに気付きやすくなります。



〔 要求を区切って分割し、細分化してグループ分けするイメージ 〕

依頼者からの要求書又はヒアリング結果資料の例

**1.1 書籍情報管理**

- ・ 所有している書籍の情報を、データベースに登録して、検索・閲覧できるようにしたい。
- ・ 書籍情報の登録は、スマホのカメラで ISBN 情報をスキャンして、ネットで検索し、その結果を登録できるようにしたい。

通常の要求仕様書で見られる要求の表現例

区切って分割

スマホのカメラで ISBN 情報をスキャンする。

ISBN 情報でネットを検索する。

検索結果をデータベースに登録する

データベースを検索し、書籍情報を表示する

USDMの要求欄に配置

<b>1.1 書籍情報管理(1)</b> スマホのカメラでISBN情報をスキャンする。			
ISBN取込	要求	RdIs10	書籍のISBN情報をカメラで画像取り込みして、OCR機能でテキストデータ化する。
(RdIs)		理由	
		説明	
		<仕様分類名>	
		□□□ (仕様番号)	
<b>1.1 書籍情報管理(2)</b> ISBN情報でネットを検索する。			
書籍情報取得	要求	GtBI10	ISBN情報で、インターネットを検索して、書籍情報を取得する。
(GtBI)		理由	
		説明	
		<仕様分類名>	
		□□□ (仕様番号)	
<b>1.1 書籍情報管理(3)</b> 検索結果をデータベースに登録する			
書籍情報登録	要求	IsBI10	取得された書籍情報から、ISBN番号や書籍名などの項目を抽出して、データベースに登録する。
(IsBI)		理由	
		説明	以下の項目を表示する予定。 ・ ISBN情報 ・ 書籍名 ・ 著者名 ・ 出版社 ・ 出版日
		<仕様分類名>	
		□□□ (仕様番号)	
<b>1.1 書籍情報管理(4)</b> データベースを検索し、書籍情報を表示する			
書籍情報表示	要求	DsBI10	キーワードでデータベースを検索し、検索結果から書籍名を抽出し、一覧で表示する。
(DsBI)		理由	
		説明	
		<仕様分類名>	
		□□□ (仕様番号)	
	要求	DsBI20	一覧表示の中から選択された書籍の書籍情報を画面に表示する。
		理由	
		説明	
		<仕様分類名>	
		□□□ (仕様番号)	

必要ならば、分割後にカテゴリでグループ化する。

依頼者からの要求書やヒアリング結果には、“ユーザーの振る舞い”のみが記述され、“ソフトウェアの振る舞い”が隠れていることがあるので、この時点で“ソフトウェアの振る舞い”を引き出して表現を調整する。

〔 USDMフォーマットの中でさらに細分化し、仕様を引き出す。 〕

### 2.2.7 限定された範囲の中で「仕様」を引き出します。

「要求」を“振る舞い”で表現し、「要求」を階層化することにより、1つ1つの動作（動詞と目的語）が明確になります。そして、明確になった動詞や目的語それぞれに着目して仕様を引き出すことで、必要な仕様をモレなく記述することができます。

**仕様とは、「明確にされた動作（動詞と目的語）の1つ1つに対して、具体的な“制約”や“処理内容”や“選択肢”などを記述したもの」です。**

仕様化するときは、それぞれの動作（動詞と目的語）について、

- 何ができれば良いのか？
- どのような状況で動けば良いのか？
- 結果はどのような形で返せば良いのか？

といった観点で仕様を引き出します。

### 2.2.8 設計工程に入る前に、「仕様」を出し切る必要があります。

設計作業の手戻りを解消するには、設計作業を始める前に要求をもとに考えられる仕様を、それを適切な（もちろん！USDMの）フォーマットでドキュメントにして、設計作業に引き継ぐことです。

### 2.2.9 要求仕様書は、作るためのドキュメントです。

USDMでは、要求仕様書とは、「要求」すなわち実現して欲しいこと（Requirement）について、“作ることの関係者”が認識を特定（Specify）できている文書と位置づけています。

具体的には、要求について、機能要求に加えて、非機能要求（品質要求）も記述して、作るために必要となる情報を要求仕様書に盛り込んでいきます

また、作ることの関係者はプロジェクト計画書などであらかじめ分かっているので、要求仕様書の記述内容の詳細度を関係者それぞれに合わせて変えることで、関係者全員が認識を特定できる状態に持っていくことが可能です。（経験の浅い者が担当する部分は、詳しく記述し、プロジェクトのドメインの経験が豊富な者が担当する部分は、ポイントだけ押さえた必要最小限の記述で済ませることも可能です。）

# 3 USDMのメリット

～USDM を使うことによって受けるメリットは・・・

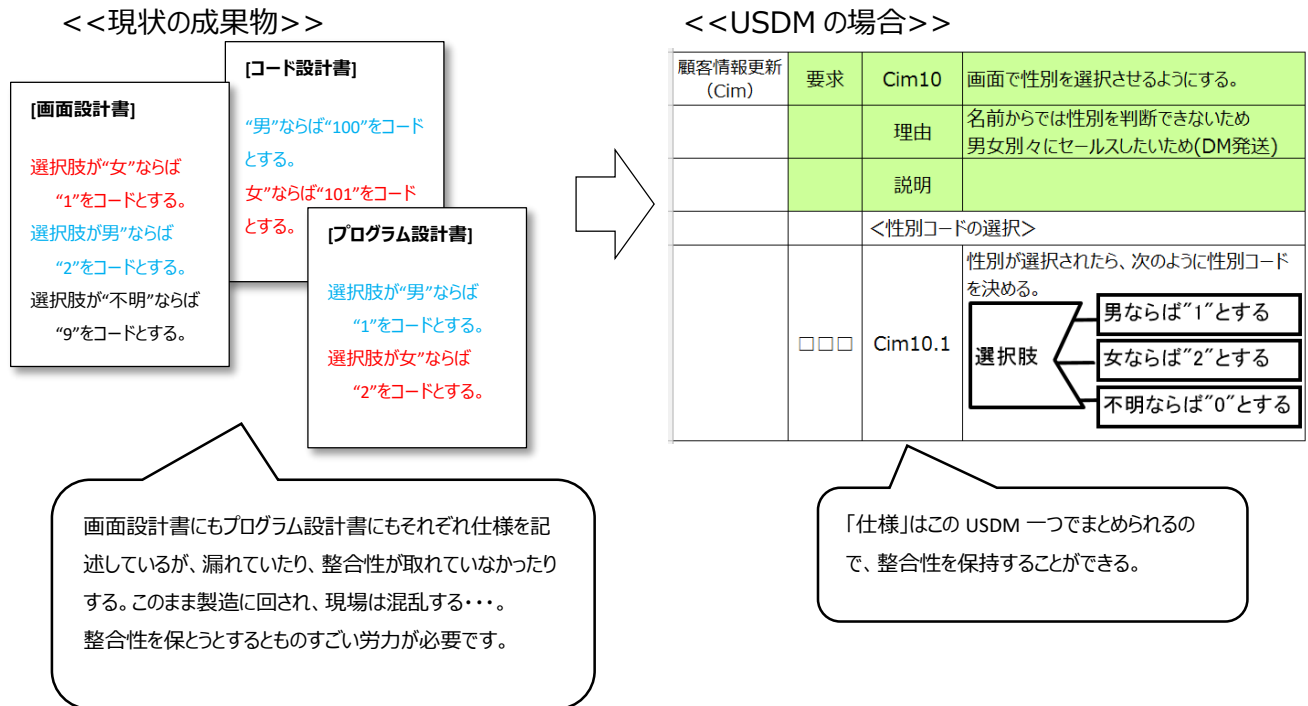
## 3.1 USDM のメリットについて

「ソフトウェア開発」で USDM を取り入れると、次のようなメリットがあります。

- ① USDM では、「要求」および「仕様」が一覧形式として記載されるため、「漏れ」や「重複」などの矛盾を“気づき”やすくなります。

現状のソフトウェア開発で作成している要求定義、仕様書、設計書などの成果物では、散文になりやすく、「漏れ」や「重複」が発生しやすくなります。

(ロジカルシンキングの“MECE(漏れなく重複なく)”が取り入れられています)



また、USDM では、「要求」、「仕様」を一覧表として扱います。USDM 自体がチェックリストとして使用することができるため、仕様どおりにプログラムが実装されているか、確認することができます。

② USDM では、「要求」、「仕様」、「理由」、「説明」と分けて記載します。

その書き方によって記述すると、それぞれの内容が明確になります。

またトレーサビリティがとれるようになります。

現状の仕様書では、「仕様」のところに説明が記載されていたり、「備考」や「説明」の欄に仕様が記載され、非常にわかりにくいものになっています。実装時、仕様が漏れてしまいます。

<<現状の成果物>>

**[プログラム設計書]**  
画面で性別を選択させるようにする。選択肢は、“男”、“女”  
“不明”とする。  
(営業部の〇〇部長からの要望です)

上記の書き方は、何となくわかるような気がする。  
ところで「営業部の〇〇部長からの要望」って、なに？  
何か特別な実装をしなければならぬのか？

「要求」、「理由」、「説明」、「仕様」が明確に分かれているので、わかりやすい。

<<USDM の場合>>

顧客情報更新 (Cim)	要求	Cim10	
			画面で性別を選択させ、選択時に性別コードを決定し、DB格納にする。
		<b>理由</b>	名前からは性別を判断できないため男女別々にセールスしたいため(DM発送)
		<b>説明</b>	営業部の〇〇部長からの要望です。
			<顧客画面で性別を選択項目>
		□□□ Cim10.1	顧客画面の氏名の横に、性別のラジオボタンを設ける。
			<性別コードの選択>
		□□□ Cim10.5	性別が選択されたら、次のように性別コードを決める。 <b>選択肢</b> → 男ならば“1”とする 女ならば“2”とする 不明ならば“0”とする
			<格納DB>
		□□□ Cim10.9	性別コードは、DBでは顧客テーブルに設ける。

③ USDM では「理由」を記載するため、「要求」および「仕様」の根拠がわかりやすくなります。

「なぜ、そうなったのか？ 理由はこうだから」とか「そういう理由だから、こうなった」と説明しやすく、内容が論理的になります。現状の成果物では、ほとんど「理由」が記載されることがなく、なぜこのような「要求」や「仕様」になったのか経緯がわからなくなっています。「要求」が妥当なものか判断しにくくなっています。

(ロジカルシンキングの“So What/Why So”が取り入れられています。)

<<現状の成果物>>

**[プログラム設計書]**  
顧客画面で性別を選択させるようにする。選択肢は、“男”、“女” “不明”とする。

なぜ、性別の選択肢を追加したのだろう？理由が書いていない。項目は顧客画面の一番下側でいいのだろうか？

「理由」が記載されているので、「仕様」が確定しやすい。  
また「理由」に合わない「要求」や「仕様」があれば、判別しやすい。

<<USDM の場合>>

顧客情報更新 (Cim)	要求	Cim10	
			画面で性別を選択させ、選択時に性別コードを決定し、DB格納にする。
		<b>理由</b>	名前からは性別を判断できないため男女別々にセールスしたいため(DM発送)
		<b>説明</b>	営業部の〇〇部長からの要望です。
			<顧客画面で性別を選択項目>
		□□□ Cim10.1	顧客画面の氏名の横に、性別のラジオボタンを設ける。
			<性別コードの選択>
		□□□ Cim10.5	性別が選択されたら、次のように性別コードを決める。 <b>選択肢</b> → 男ならば“1”とする 女ならば“2”とする 不明ならば“0”とする
			<格納DB>
		□□□ Cim10.9	性別コードは、DBでは顧客テーブルに設ける。



- ④ USDM では「要求」の配下に「仕様」を記述するため、「要求」の範囲に収まるような「仕様」を作成することができます。「要求」の範囲からはずれた「仕様」を記述しても、レビュー時、違和感があるため簡単に見つけられます。

「要求」の配下で「仕様」を記載することによって、「仕様」が「要求」どおりのものか判断することができます。極端な例ですが「音楽を流す」という「仕様」は、ここに書かれている「要求」とは全く異なります。この場合は削除するか、別途「仕様」を満たす「要求」を立てる必要があります。

顧客情報更新 (Cim)	要求	Cim10	画面で性別を選択させ、選択時に性別コードを決定し、DB格納にする。
		理由	名前からでは性別を判断できないため男女別々にセールスしたいため(DM発送)
		説明	営業部の〇〇部長からの要望です。
			<顧客画面で性別を選択項目>
	□□□	Cim10.1	顧客画面の氏名の横に、性別のラジオボタンを設ける。
			<性別コードの選択>
	□□□	Cim10.5	性別が選択されたら、次のように性別コードを決める。 <div style="border: 1px solid black; padding: 2px; display: inline-block;">           選択肢           <ul style="list-style-type: none"> <li>男ならば"1"とする</li> <li>女ならば"2"とする</li> <li>不明ならば"0"とする</li> </ul> </div>
			<格納DB>
	□□□	Cim10.9	性別コードは、DBでは顧客テーブルに設ける。
	□□□	Cim10.10	ついでに音楽を流す。

- ⑤ USDM では「要求」の1つ1つの動詞を「仕様グループ」と記載します。その「仕様グループ」に対して、「仕様」を導くことができます。

<<現状の成果物>>

<<USDMの場合>>

**【プログラム設計書】**  
顧客画面で性別を選択させるようにする。選択肢は、「男」、「女」「不明」とする。

これで「わかるような気がした」と思っても、どうコーディングして良いのかわからない……。

「仕様グループ」というスコープまたはカテゴリの配下に「仕様」を記載することによって思考のフォーカスを絞り込むことができ、容易に「仕様」が書きやすくなります。また、重複や漏れも見つけやすくなります。

顧客情報更新 (Cim)	要求	Cim10	画面で性別を選択させ、選択時に性別コードを決定し、DB格納にする。
		理由	名前からでは性別を判断できないため男女別々にセールスしたいため(DM発送)
		説明	営業部の〇〇部長からの要望です。
			<顧客画面で性別を選択項目>
	□□□	Cim10.1	顧客画面の氏名の横に、性別のラジオボタンを設ける。
			<性別コードの選択>
	□□□	Cim10.5	性別が選択されたら、次のように性別コードを決める。 <div style="border: 1px solid black; padding: 2px; display: inline-block;">           選択肢           <ul style="list-style-type: none"> <li>男ならば"1"とする</li> <li>女ならば"2"とする</li> <li>不明ならば"0"とする</li> </ul> </div>
			<格納DB>
	□□□	Cim10.9	性別コードは、DBでは顧客テーブルに設ける。

⑥ USDM では「仕様」を、曖昧性を排除し具体的に記述するため、迷うことなく実装することができます。

<<現状の成果物>>

**[プログラム設計書]**  
 顧客画面で性別を選択させるようにする。選択肢は、“男”、“女” “不明”とする。

「仕様」が具体化されているので、設計およびコーディングしやすくなります。

<<USDM の場合>>

顧客情報更新 (Cim)	要求	Cim10	画面で性別を選択させ、選択時に性別コードを決定し、DB格納にする。
		理由	名前からでは性別を判断できないため男女別々にセールスしたいため(DM発送)
		説明	営業部の〇〇部長からの要望です。
<顧客画面で性別を選択項目>			
	□□□	Cim10.1	顧客画面の氏名の横に、性別のラジオボタンを設ける。
<性別コードの選択>			
	□□□	Cim10.5	性別が選択されたら、次のように性別コードを決める。 <div style="border: 1px solid black; padding: 5px; display: inline-block;">           選択肢           <ul style="list-style-type: none"> <li>男ならば“1”とする</li> <li>女ならば“2”とする</li> <li>不明ならば“0”とする</li> </ul> </div>
<格納DB>			
	□□□	Cim10.9	性別コードは、DBでは顧客テーブルに設ける。

⑦ 「要求」を「振る舞い」と記述するため、「要求の動き(順序性)」と「範囲」を持たせることができます。

(「2.2.3「要求」されているソフトウェアの“振る舞い”を見せて、動きを感じさせるように表現します。」参照)

# 4 USDM の書き方

ここでは、USDM を書き方について説明します。ツールはスプレッドシートのあるものを使います。推奨は Microsoft EXCEL です。

## 4.1 基本フォーマット

USDM の基本フォーマットは次の 2 つがあります。

<要求が 1 階層の場合>

①カテゴリ名	パターン1 (要求に対して仕様が展開されるケース)			
カテゴリ名 (記号)	要求	(要求ID)	ここに要求を記述します。	備考欄
		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。仕様とはみなされません。セルを広げて、図を貼ってもよいです。	④説明
		<仕様グループ名>	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。	
	□□□	(仕様ID)		⑤仕様
	□□□	(仕様ID)		
	□□□	(仕様ID)		
		<仕様グループ名>		
	□□□	(仕様ID)		
	□□□	(仕様ID)		
	□□□	(仕様ID)		

<要求が 2 階層の場合>... 1 つの要求の範囲が大きいと判断した場合

①カテゴリ名	パターン2 (要求がもう一段階階層化されるケース。これ以上の階層化は避けたほうがよい)			
カテゴリ名 (記号)	要求	(要求ID)	ここに要求を記述します。	
		理由	要求の背景や理由について記述します。	②要求
		説明	要求について必要に応じて説明してください。仕様とはみなされません。セルを広げて、図を貼ってもよいです。	③理由
		要求	ここに階層化された要求を記述します。要求IDは一段下げられます。	③理由
		理由	範囲を狭めた要求について背景や理由を記述します。	
		説明	要求について必要に応じて説明してください。仕様とはみなされません。	④説明
		<仕様グループ名>	主分割記号...全体を通して共通の分割基準として決めます (オプション)。 <<前提条件>><<内部処理>>	
		<仕様グループ名>	補助分割記号...主分割のなか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、純度の高い集合を伴います。	
	□□□	(仕様ID)	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。	⑤仕様
	□□□	(仕様ID)		
	□□□	(仕様ID)		
		<仕様グループ名>		
	□□□	(仕様ID)		
	□□□	(仕様ID)		
	□□□	(仕様ID)		

<各項目の説明>

項目名	説明	備考
① カテゴリ名	カテゴリ名を記述します。	
② 要求	要求を記述します。 要求 ID も記述します。 「3.2.要求の書き方」を参照のこと。	
③ 理由	理由を記述するところです。 「3.3.理由の書き方」を参照のこと	
④ 説明	要求に対しての補足などを記述するところです。	
⑤ 仕様	仕様を記述するところです。 仕様グループ、仕様 ID も記述します。 「3.4.仕様の書き方」を参照のこと	

<記述例>

検索するWebページの要求仕様書（例）

WEBページ検索 (SER)	要求	SER10	
			利用者が入力欄に入れたキーワードを用いて、検索を開始させ、検索結果を一覧で表示させ、該当Webページにジャンプできるようにしたい。
		理由	ユーザーがすばやく目的のWebページに移動できるようにするため。
		説明	
		<検索結果の表示>	
	<input type="checkbox"/>	SER10.10	検索ボタンを押下後、検索結果をリスト表示する。
	<input type="checkbox"/>	SER10.12	検索リストの要素は、ページタイトル、URL（ハイパーリンク）、ヒットしたキーワードの前後100文字とする。
	<input type="checkbox"/>	SER10.14	ヒットした件数を表示する。件数が10件を超える場合、10件単位でページングする。
	<input type="checkbox"/>	SER10.16	ページング数は、10個まで表示し、以降は・・・の記号で省略する。

(以下略)

## 4.2 要求の書き方

USDMにおける要求とは、作業者に作ってほしいという依頼のことです。

依頼者は、こういう機能を、こういう性能で、こういう作り方で、作ってほしいと依頼します。

要求の書き方として、依頼者がソフトウェアに求めている機能（振る舞い）を記述します。

機能をユーザーがどう使うかをイメージし、文章にします。

ユーザーの操作（動詞）を省略せずに記述してください。

### 4.2.1 要求をうまく引き出す

要求を持つのは依頼者であって、作業者は欲しいものはわかりません。

しかし、専門知識のない依頼者も、本当に欲しいものを先んじて考えることは難しいです。

過去からの類推が効く分、作業の方が欲しいものを見つける能力は高いです。

USDMは、作業者が要求を理解し、表現することを助けます。

USDMは、要求を文章化することによって、その文章を「てこ」にして本当に欲しいものが導出できます。

USDMを書くにあたって、まず「何を入力として作るのか？」ということになります。

要求仕様を作成する中心となるのは依頼者へのヒアリング作業です。

また、現在ある成果物から作ることもできます。

### 4.2.2 要求文章の記述ルール…動詞を引き出す

要求を文章化して表現します。

その際、動詞をできる限り細かく出すことをします。

その上で、動詞を時間軸にまとめ、5～7個程度にまとめて下位要求とします。

たとえば、依頼者から「ユーザー検索画面を作って」と言われているものとします。

- XX画面から**遷移**し、
- 組織データとユーザーデータをデータベースから**取得**し、
- 組織データを選択条件に**プリセット**し、段階的に**絞込ができる**ようにし、
- 絞込を行うたびにヒット件数を**表示**し、
- ヒット件数が閾値を下回った場合、ユーザー情報を**一覧で表示**する

といった内容があります。

これらを要求の文章として表現します。**太字**が動詞です。

このように、具体的な機能の動きをイメージして、**動詞**を引き出します。

動詞が 8 個以上になると仕様が多すぎて見づらくなります。その場合、要求を分けます。

分け方として、「要求を階層化する(垂直方向)」または「別の要求とする(水平方向)」があります。

上の例でいうと、絞込の部分は「別の要求とする(水平方向)」として分けてもよいです。

また、異常系も明確にします。

例えば、

- 絞込の結果、ヒット件数が 0 件になった場合、条件変更を促すメッセージを表示する

なども記載します。

#### 4.2.3 要求を階層化する

依頼者から提示される要求の範囲が広い場合(仕様に含まれる動詞の数が 8 以上)、そのままでは仕様を出すことは難しいです。

そのため、要求を階層化し、上位要求および下位要求に分割します。

要求をまとめるための要求(上位要求)を作り、その配下に分割対象の要求を含めるようにします。

上の例だと、

- 組織階層をもとにユーザーを絞込し、一覧表示したい(上位要求)
  - ◇ 遷移条件による振る舞いの違い～(下位要求)
    - XX 画面から遷移し……
  - ◇ 初期表示時点での振る舞い～(下位要求)
    - 組織データとユーザーデータをデータベースから取得し……
    - 組織データを選択条件にプリセットし、段階的に絞込ができるようにし……
  - ◇ 絞込時点での振る舞い～(下位要求)
    - 絞込を行うたびにヒット件数を表示し……
    - ヒット件数が閾値を下回った場合、ユーザー情報を一覧で表示する
  - ◇ ユーザー情報を一覧表示する際の振る舞い～(下位要求)
    - ユーザー情報を一覧で表示する……

と階層化します。

下位要求では動詞の数は減りますので、仕様が書きやすくなります。

また、下位要求にも理由をつけます。

#### 4.2.4 要求の分割方法

要求を分割する際は、基本的には時系列で書いていきます。

USDMとしては、4 パターンの分割基準を設けています。

- 時系列分割・・・時間軸に要求を分割する方法
- 構成分割・・・「機能別」や「構成別」で要求を分割する方法
- 状態分割・・・「状態」という概念で要求を分割する方法
- 共通分割・・・共通する機能を切り出して独立する方法

#### 4.2.5 ID の付与

要求には ID を振ります。

依頼者とやりとりする際、ID をもとにすると、「話の対象が明確になる」、「類似の項目との誤解もなくなる」など、話がしやすくなります。また、トレーサビリティを確保しやすくなります。

#### 4.2.6 仕様を導出しやすいよう要求の粒度を小さくする

要求は、仕様を束ねる**範囲**の事です。

適切な範囲を設定することで、仕様を出しやすくなります。

仕様に関する気づきを得るためには、要求を十分小さい粒度にします。

そのときの粒度は、「関係者が認識を特定（Specify）できる」程度にして下さい。

#### 4.2.7 仕様が要求に正しく含まれているか検証する

逆に、仕様から要求を導出することもあります。

仕様を記述後、その仕様が正しく要求の配下にいるか？という観点で、検証を行います。

その結果、よりふさわしい要求の配下に仕様を移動します。

ある仕様が、現時点で出された要求のすべてとうまく対応しない場合、**よりふさわしい要求を、新たに作る**こと

が必要になります。

仕様から要求を導出することは、依頼者の要求がこうである、と新たな仮説を立てる事と言えます。

#### 4.2.8 要求の共通化

例えば、要求として

- 絞込の結果、ヒット件数が 0 件になった場合、条件変更を促すメッセージを表示する

なども記載するとします。

そうすると、細かいユーザーの誤入力までいちいち書くのか？という話が出ます。

このような要求は、共通化できますので、「共通として別の要求とする(水平方向)」とします。

### 4.3 理由の書き方

依頼者が求める要求が、どういう背景のもとで生じたかを記述します。

突き詰めると、要求の目的は「儲かりたいから」となりますが、とても理由になりません。

できるかぎり、「要求に特有の理由」、「依頼者が重きをおいているポイント」を押さえます。

例えば、19 ページの記述例では、「速さ」と「正確性」に重きをおくことが理由になります。

#### 4.3.1 理由を書く

機能を求めるに至った背景、すなわち、現在までどうやったのか？これからどうしたいのか？という情報が出てきます。実際に使うユーザーが、どういう状況のもとその機能を使うのか…という事を把握します。

理由は要求の一部といえます。USDM では、書く欄を分けます。

…なお、「理由を聞く」というのは難しく、単純に「どうしてこの機能が必要ですか？理由を教えてください」と訪ねても、「業務を効率化するためでしょ」とか、「俺が考えた案が気に入らないの？」など、明確でない理由が返ってくることもあります。聞き方は工夫しましょう。



## 4.4 仕様の書き方

要求を達成するために、具体的な仕様を記述します。

単純なイメージとしては、「設計の概要」と捉えてください。「ソースを書く際の論理的なまとめ」とも言えます。

### 4.4.1 仕様とは何か

仕様とは、要求に含まれる動詞を具体的に表現したものです。つまり、

- 要求から導出される（すべての仕様は、要求に属する）
- 設計をイメージできる（仕様毎に作業タスクと作業時間が導出できる）
- 検証をイメージできる（仕様をテストする手段が導出できる）

条件を満たす粒度で要求を具体化します。

### 4.4.2 仕様グループを設定する

まず、要求のうちひとつの「動詞」と、付随する「目的語」に着目し、仕様グループを定義します。

仕様グループのもとに、仕様を記述します。

要求（分割されている場合は、下位要求）に含まれる動詞から、仕様のグループを設定します。

「検索条件をプリセットする」という動詞に対応して、<検索条件のプリセット>というグループを明記します。

### 4.4.3 仕様の粒度をどう設定するか

仕様は、何を作るか（What）を定義するものです。よって、最終的にはソースコードに対応させることができます。また、仕様は依頼者と合意するため、依頼者が理解できる粒度にします。仕様の記述粒度をどこまでにするかは、

- 依頼者が理解できること。
- 作業者が大まかな作業の流れがイメージできること。
- 要求から仕様に落とす際のモレが生じないこと。

です。

1項目は1つの内容（細目）で記述してください。細目ごとに曖昧さがないことをチェックするため、細目にある「動詞」と「目的語」に着目し、関係者の間で同じイメージを持てるかを確認してください。

依頼者と作業者との認識が合っている場合は、仕様の記述粒度を粗く書いてもよいです。  
逆に作業者が新入りベンダーで認識があまり合っていない場合は、仕様の記述する粒度を細かく書かいたほうがよいです。

#### **4.4.4 ID の付与**

仕様にはそれぞれ ID を振ります。

依頼者とやりとりする際、ID をもとにすると、「話の対象が明確になる」、「類似の項目との誤解もなくなる」など、話がしやすくなります。

## 4.5 USDM 作成時の注意点

### 4.5.1 なぜ要求の引き出しと仕様化を分けるか

仕様グループの作成までを、「要求の引き出し」作業とし、仕様化と分けることを推奨します。

要求の引き出しの「ついでに」仕様化もやれば良いじゃないか、という意見があります。

しかし、USDM の作業プロセスとしては、要求の引き出しと仕様化は明確に分けるべきです。

なぜ分けるかというと、分業のためです。

要求の引き出しは、本来的には依頼者の作業です。

依頼者やその背景に眠っている要求を探る作業は、作業員だけで完結できない作業です。

本質的に属人的で、ゆえに人を増やしても効果が上がらない。言い換えれば、生産性が読めません。

一方、仕様化の作業は、要求が適切に表現されてさえいれば、これまでの組織のやり方にあわせた解決策を過去の経験ないし類推から導出できます。USDM での要求仕様書作成時、仕様数と作業時間を記録することで、生産性を把握でき、作業工数を見積できるようになります。

### 4.5.2 仕様化の作業員について

特に強調したい点は、**仕様化の作業員は、次工程の設計担当とすること**です。

異なる担当をつける場合、要求仕様書を確かに理解できた、という引継を保証するための仕掛けが必要になるためです。

要求からうまく設計に繋げるために、同じ作業員を割り当てましょう。

### 4.5.3 コピペ(コピー&ペースト)は最小にする

一部だけ仕様が違うという場合、文章をコピペで作り一部修正という形で書くと、差異がわかりにくくなります。

読み手に伝わりづらい注意すべき表現なので、USDM では「ペースト作文」と呼び、避けるようにしています。

仕様欄に表や図を追加して、見やすい形にしてください。

フローチャートやディビジョンテーブルなど、テスト工程で条件の分類を行う時の書き方を流用するとよいでしょう。

### 4.5.4 条件分岐は網羅するようにする

「受信環境が Wifi (無線 LAN) の場合は、バックアップを有効にしない。」という仕様があったとします。これを実装する際は条件文で IF 文になります。

IF 文という事は、すべての場合分けを考える必要があります。

有線 LAN の場合は？パケット通信の場合は？通信帯域が遅い場合は？…といった、隠れた条件をすべて記述する必要があります。

文章に「もし…の場合は」という形の表現があれば、「else は何？」と考える癖をつけましょう。

このような「隠れた IF 文」が生じやすいのは、文章の末尾が「～しない」という場合が多いです。そのため、USDM では「否定表現」とよび、注意すべき表現になります。

#### 4.5.5 「T. B.D.」および「等」は具体的に書く

表のセルが縦に長くなるのが嫌だし、見栄えが良くないから…という理由で、「等」や「etc」で仕様のパターンを止めてしまう場合があります。**現タイミングで確定できる事であれば、すべて書きましょう。**

表示幅が狭いのであれば、複数セル使うなり、別資料にまとめて参照を記載するなど、工夫して書き切ります。

現タイミングで未確定であれば、仕様の合意時期を決め、それを記載します。

仕様漏れを防ぐことができます。

# 5 USDMの取り組みの糸口

---

～USDMを始めるには・・・

## 5.1 取り組みのポイント

設計作業と仕様化作業を明確に区別しましょう。

設計工程で仕様化するよりも、要求仕様書を作成する中で仕様化したほうが、作業の手戻りを防ぐことができます。トータル工数が短くなります。

## 5.2 どこから手をつければ良いか

いきなり現状のソフトウェア開発で採用している要求仕様書をUSDMのものに入れ替えることはハードルが高いものがあります。まずは、自分だけでUSDMで要求仕様書を書いてみます。最初のステップとして、補足編を自分で転記してみて全体像を掴むのが良いでしょう。

例えば、完了したプロジェクトの設計資料のうち、自分の担当部分の要求や仕様を一部抜粋して、USDMの要求仕様書を用いて清書してみるイメージで、

- 要求を振る舞いで表現すること、
- その要求の中の動詞をすべて表現してそれを分割して表現すること、
- 分割された要求の下に仕様を配置していくこと

に取り組んでみます。

要求と仕様を見比べて、仕様にモレがないか、曖昧な表現はないかを確認します。

曖昧な表現のうち、見つけやすいものに以下の3つがありますが、これらの表現を修正して無くすことから始めると良いです。

- 「等」、「e t c」
- 否定表現
- ペースト作文

「USDMを開発でどう使うか」も参照してみてください。

もし、完了したプロジェクトといえども実際の設計資料を練習に使うのは難しい（使えない）場合は、この小冊子の USDM の例示を、自分で USDM のフォーマットに入力して見ることも、オススメです。

© 派生開発推進協議会 2016  
作成者: AFFORDD T2 研究会 (USDМ 入門)  
作成日: 2016 年 5 月 10 日 (平成 28 年)