

<セッション>

ライブ USDM はこうやって書く！～ライブで解説しながら実際に USDM を作成してみます～

<質問1>

システム要求仕様を USDM で書く場合に、「書きにくい」アプリケーション事例はあるのでしょうか？あるいは、どんな事例でも工夫次第で書けるのでしょうか？ご教示いただければ幸いです。

<回答>

合意形成のためのドキュメントなので、要求が曖昧な場合、仕様が実装に近い表現しかできない場合（ソースからリバースで USDM を作る場合）、仕様の不確実性が利用可能なデータに依存する場合（機械学習プロジェクトの場合）、発注側が仕様の妥当性を評価できない場合（非機能要件）、API 仕様など協議により仕様のあり方が変わりうるものなどが難しいです。が、USDM でなくとも難しいです。

<質問2>

要求の時点で矛盾が生じている場合の解消方法はどのようなものがあるのでしょうか。今回のケースだと、「起きるね」と話している段階で既に起きていると思います。

<回答>

依頼者は二度寝がしたいので、彼の中では矛盾していません。生じた疑問はヒアリングして確認すればよいでしょう。ただし、聞き方として「矛盾してませんか？」って言うと怒る人もいるので、実際の運用イメージを具体例で確認したり、別の解決手段を提案すると本音が聞き出せます。

<質問3>

要求が条件分岐する場合、要求を2つに分けて書くべきでしょうか？

スマートスピーカーに「起きるね」と話しかけると、リビングのライトをつけて、TVの電源をONにして、晴れの日にはカーテンと窓を開け、コーヒーマーカーの電源をONにしてコーヒーを淹れる。

スマートスピーカーに「起きるね」と話しかけると、リビングのライトをつけて、TVの電源をONにして、雨の日にはカーテン開け、お湯を沸かす。

のような場合です

<回答>

動詞が異なってくる場合は分けたほうが良いでしょう。上位要求を抽象的に書いて共通部分と差分を分けて書くか、メインの処理と特殊な場合の処理で要求を分けるかします。少しの差であれば仕様グループで場合分けすることも可能です。

<質問4>

そもそも、要求に手段が入るとなぜいけないのでしょうか？下段の記述と重複するだけだと思いますが・・・。

<回答>

実現手段が限定されると、より合理的な別の実現手段（オプション）が検討されにくくなるためです。

<質問5>

ソフトウェアの要求仕様書として USDM を使用する場合、要求はだれの目線で書くべきでしょうか？（ソフトウェア？ユーザー？システム？）

<回答>

画面であれば画面の利用者ですが、バッチなど内部コンポーネントの場合はシステムの効率化の観点で作成されるため直接的な利用者がおらず書きづらいことがあります。コンポーネントの利害関係者は存在するので、その人達が同じイメージを持てるように要求と理由を記述します。

<質問6>

要求の階層の理想があれば教えて下さい。

また、階層の深い、浅いに対するメリット・デメリットがあれば教えて下さい。

<回答>

作業対象の案件やプロジェクト、組織で活用する範囲で統一したほうが便利かと思います。上位/下位に分割する前提の組織テンプレートを見かけることが多いです。

要求そのものをどのように分割すべきかは以下資料が詳しいです。

https://affordd.jp/conference2015/affordd_conference2015_tutorial.pdf