

## <セッション>

### シフトレフトテストにおけるシステム要求の獲得 ～テキストマイニングを用いたアプローチ～

#### <質問1>

使用されているツールは、KH Coder のみでしょうか？他にテキストマイニングで使われているツールがあればご教示ください。

#### <回答>

KH Coder の他では、有償ツールでは NTT データ数理システムの "Text Mining Studio" を使用したことがあります。

#### <質問2>

システムの振る舞い、という言葉が理解しにくかったのですが、少しずつわかってきました。これは USD M 上での共通概念ワードなのですか？

#### <回答>

USD M における振る舞いは『イベントに始まり、入力～変換～出力処理を経て止まるまでの動作』とあり(書籍「要求を仕様化する技術・表現する技術」)、これを共通概念と捉えてよいと思います。ただ、実際にはシステムの大きさ・複雑さによって上記の定義では書きにくいケースがあり、今回の事例では工程別に動作を切り出して要求による範囲を小さくしています。

#### <質問3>

QA との連携として具体的に考えていることはありますか？

#### <回答>

USD M による要求仕様書を QA との共通ドキュメントとして、出荷判定や受け入れテストのベースとしての活用を考えております。QA 活動において、USD M の上位要求や仕様との関連性の判りやすさ(透明性)が重要と考えております。

#### <質問4>

単体テストや結合テストは開発者がそのまま担当されることがあると思いますが、そういった場合でもシフトレフトという考え方は適用できるのでしょうか？

#### <回答>

テストを行うのが誰であれシフトレフトは適用できると考えております。アジャイル開発では開発とテストは一体での活動となりますが、ウォーターフォールにおいてもアジャイルと同じような活動を目指したものがシフトレフトの考え方になります。テスト実行だけでなく、設計の段階でテストビリティ(テスト容易性)を考慮し、テストツールを並行して作る、といった活動もシフトレフトと言えます。

### <質問5>

シフトレフトでテスターの方が要求段階から参画するためには、ハード・ソフトの製品知識を習得する必要があるかと思います。どのように習得されていますか？

### <回答>

ハードウェアの知識の習得は実際には難しく、レビューにハードウェア開発のチームメンバーに入ってもらうこともありました。ポイントとしては、ハードウェア自体の理解を深めようとしがちになるのを制止して、『結局それは何をするためのものであるか』『それがソフトウェアとどう連携しているか』をまず掴む、という方針で進めました。

### <質問6>

[テキストマイニングによるシステム要求抽出]と[テストシフトレフト]の関係性を教えてください。  
要求抽出をテストチームが実施したほうが、要求抽出の品質があがるということでしょうか？

### <回答>

本発表ではテストチームが主導でしたが、理想は要求抽出にテストチーム『も』参加すること、だと考えております。

### <質問7>

派生開発では変更要求仕様書の USDM は「作業視点」になると思いますが、差分要求のようなシステム目線でない部分や要求が不完全な部分に対してのアプローチをお聞かせ下さい

### <回答>

変更要求仕様においても、システムの振る舞い、それを実現するソフトウェアの仕様との構造においては何も変わりはないと思います。ソフトウェアの変更のみであっても、テストとしては関連するシステムの振る舞いを確認する必要があるためです。要求不完全のアプローチですが、こう動くだろうと仮定の要求として書き、レビューで認識を合わせるということはしておりました。

### <質問8>

テキストマイニングツールのアウトプットと、それを元にした人手の作業の境目はどの部分でしょうか（ツールのアウトプットはどのようなものでしょうか）？

### <回答>

品詞別の語の一覧、共起ネットワーク、階層化クラスタ分析、これらはテキストマイニングツール(またはPythonのライブラリ)によって出すことができます。

### <質問9>

開発ドキュメント問う言うのが、いろいろな人が書かれていて、言葉や粒度がばらばらで、目的もずれているばあい、形態素分析するとバラバラになって、システム要求にまとめるのが大変になるように思うのですが、そのようなばらつきについて、どうしたらよいか、ご意見をいただけるとありがたいです。

### <回答>

表記ゆれ(同じ意味で異なる表現)は悩ましい問題ですが、ある程度はツール(KH Coderであればプラグインや設定ファイルの書き換え)や、前処理を行うモジュールによってまとめることが可能です。自然言語処理を行う中で、開発ドキュメントの書き方の統一性のなさや、誤字の多さといった問題点が浮き彫りになりますが、そこを起点にドキュメントの書き方・標準化への提案に繋がると考えております。

**<質問10>**

USDM を作成物として作成することになると認識しましたが、  
間違っていないということをどうやって保証するのでしょうか？

**<回答>**

USDM はテストチームで完結するドキュメントではなく、開発チーム、QA に共通するドキュメントとして、レビューは必須にしています。