

## 付録

### ●第1回（例会）：レビュー演習：猪塚 修（横河ソリューションサービス株式会社）

#### ■概要：

本演習は、ソフトウェア成果物の改善を目的としたレビューについて学ぶ。

演習内容は以下の通りである。

1. レビューの勘所を養うため、簡単な仕様書に対するレビューを個人演習として行う。
2. 他の人の視点を知るためレビューの内容を共有する。

#### ■有効性：

レビューはソフトウェアテストと同様に欠陥を見つける手段であり、ソフトウェアテストより前の工程で行うので低コストで欠陥を発見することができる。また、ソフトウェアテストにて実施が難しい確認がレビューでは簡単にできる。

ソフトウェアの設計は唯一絶対ではないので、レビューを行うことにより専門家など多くの人の意見をもらうことができ、より良い設計にできる。

業務を遂行しつつ、参加者の教育ができる手段である。

#### ■留意点：

レビューアのス킬によって視点が変わるため指摘内容の有効性が変わる。また、レビューで指摘される基本的なことをレビューイが事前に理解し作りこみができていれば、レビューによる指摘の質がより向上する。

#### レビューアの成果物の査読方法の例

1. 理解する
2. もれなくダブリなく記載されているかを確認する
3. レビューチェックシートの項目を思い浮かべながら作業する
4. プロジェクトとしての重要事項を常に気にする
5. 視点を変える

#### レビューイのドキュメントの書き方

1. できるだけ同じ並びで書く
2. 関連を明確にする
3. 同じ言葉を同じに使う
4. インターフェース・ファイル設計を明確にする
5. 上位ドキュメントとの対応を確認する
6. 異常処理を考える
7. テスト仕様書は条件・結果を明確にする

## ●第2回（例会）：オブジェクト指向分析設計：井上 樹氏（豆蔵）

### ■概要：

オブジェクト指向分析設計の「モデリング」手法を学び、開発でどのように活かすことができるかを、演習を通して学習する。演習内容は以下の通りである。

- ・要求のモデリング

システム要求を把握することを目的とする。

ストップウォッチを題材とし、ユースケース図・ステートマシン図を作成した。

- ・設計のモデリング

ソフトウェアの構造を見える化することを目的とする。

ストップウォッチを題材とし、シーケンス図を作成した。また、クラス図からプログラム構造の問題を指摘する演習も行った。

### ■有効性：

各演習を通し、以下の有効性を体感した。

- ・要求のモデリング

複数の図を使用することにより、システム要求の不備・不足を多角的に発見できる。

特にユースケース図はユーザー視点を起点として描くため、使い勝手といった非機能要件の把握にも期待できる。

- ・設計のモデリング

ソースコードに比べソフトウェア構造を広い視点で俯瞰することができるため、ソフトウェア構造の問題点（モジュール間の依存関係など）に気が付きやすい。そのため設計段階での問題点摘出が期待できる。また、視覚効果が高いため、レビュー効率向上も期待できる。

### ■留意点

上記の有効性を活かすためには、モデリング手法を正確に記述する訓練が必要となる。

実際の開発に応用するためには長い期間と経験が必要となると思われる。具体的な期間は以下ようになる。

- ・スケッチレベル → 数ヶ月
- ・ソースコード生成レベル → 数年

日々の業務改善など、小さい範囲からモデリング手法を作成する訓練を積んでいくと良いと考える。

### ●第3回（合宿）：アーキテクチャ設計・評価：長谷川 裕一氏（合同会社 Starlight & Storm）

#### ■概要：

本演習は、アーキテクチャの品質特性とそれを表すシナリオ、一般的な実現手法と、アーキテクチャの分析評価の概要について知ることを目的とし、「ドーナツ屋の起業」をテーマに演習を行った。

アーキテクチャとは形のある決まった概念ではなく、ビジネス要件、システム要件などに基づいて導かれるシステムの骨格であり、目標とする品質特性を明らかにして、それを満たすアーキテクチャを設計・評価する必要がある。

アーキテクチャの設計手法としては幾つかあるが、今回は ADD (Attribute Driven Design) を使用した。ADD では、目標とするシステム要件の品質特性シナリオを作成することで非機能要件を明確にし、作成した品質特性シナリオから実現手法の候補を挙げてアーキテクチャを設計する。品質特性シナリオの作成では「どこに」「何をしたら」「どうなってほしい」を具体的にする。

分析・評価の手法としては ATAM (Architecture Tradeoff Analysis Method) を学んだ。ATAM は品質特性に関する要求を満たしているアーキテクチャが設計されているかを評価することで、実現手法のリスクや、品質特性に肯定的に作用する重要ポイント、他の品質特性に影響を与えるトレードオフポイントが明確になる。

#### ■有効性：

ソフトウェア開発においてアーキテクチャを適切に設計することで、目標の明確化と品質の実現が可能となる。また、アーキテクチャの決定過程などシステムの理解を助け、実装やメンテナンスでの問題発生時のリスクを緩和することができる。

#### ■留意点：

品質特性シナリオの作成では、最も脅威となる破滅のシナリオを考え、それを防ぐためのシナリオを中心に考えると作り易い。また、具体的な実現手段を考えてしまうと非機能要件を十分に抽出できない恐れがあるため、実現手段は考えない。同様に、様々なステークホルダの観点から検討を行わないと、複数の観点からの非機能要件は抽出できない。

ATAM を利用した分析/評価では、必ずしもマニュアル通りに実行するのではなく、各開発部門の環境に合わせてカスタマイズし、軽量化してから使用するのが望ましい。

**●第4回（臨時会）：利用者視点でCS向上を図るUXデザイン手法の体験学習：金山 豊  
浩（株式会社ミツエーリンクス）村上 和治（東京海上日動システムズ株式会社）**

**■概要：**

UXデザイン（User Experience）のプロセスと設計手法の概要を学び、ユーザーの本質的な要求を具体化するための手法として「ストーリーボード」「スケッチ&プロトタイプ」「ナビゲーションマップ」について講義と演習を通して学習した。内容は下記の通りとなる。

**1. 講義**

UXデザインの設計プロセスである「ストーリーボード」「スケッチ&プロトタイプ」「ナビゲーションマップ」の有効性を体系的に学ぶ。

**2. ストーリーボード作成**

ペルソナ行動シナリオをベースに、ペルソナが目的を達成するための行動/振る舞いをコマ毎にストーリー化した「ストーリーボード」を作成する。

**3. スケッチ&プロトタイプ作成**

コマ毎に作成したストーリーボードをベースに、必要な「情報」と「機能」を抽出し、ユーザーの振る舞いをイラストで表現（スケッチ）したプロトタイプを作成する。

**4. ナビゲーションマップ作成**

コンテキストの維持に注意しながら複数のスケッチを統合した「ナビゲーションマップ」を作成する。作成したスケッチの操作感などを評価し、スケッチ・ナビゲーションマップの作成・修正を繰り返し行うことで、ユーザーにとって価値あるものを作り上げていく。

**■有効性：**

サービスやプロダクトの開発において、ユーザーにとって価値を探しながら作るべきものを明確にする仮説検証の手法として非常に活用できる。「機能」を中心に考えるのではなく、「ユーザーへの価値」を中心に考え、ユーザーが目的を達成するための行動/振る舞いを「ストーリー」として視覚化することで、ユーザーが評価しやすく、本質的な要求を早期に発見できる。また、「ストーリー」をスケッチすることは手間やコストがかからず、容易に実証・実験が可能である点でも非常に有効な手法である。

**■留意点：**

デザインを評価するフェーズでは、「クライアント」や「エンドユーザー」の協力が必要不可欠である。デザインフェーズのコンセプト・検証はあくまで仮説に過ぎない。そのため、企画段階から「ユーザーへの価値」を一緒に考えて、「プロトタイプ」を作り上げることで有効性が発揮される。また、成果物は評価者によって変更する必要がある。

評価者がエンドユーザーであれば、ビジュアルデザインまでされたモックアップを操作して評価してもらう。プロトタイプのような見た目の色や形のビジュアルが未完成で操作感が想定できないものでは、エンドユーザーにとっては実際に使用するイメージが湧かなく、良いフィードバックが得られないと考える。

## ●第5回（例会）：要求工学（要求分析）：中谷 多哉子氏（放送大学）

### ■概要：

システム開発において重要な技術である『要求工学（特に要求抽出）』を実践した。近年では要求に対する解は複雑化し、更には状況の変化によって要求そのものが変化する問題が増加している。（Cynefin フレームワークにおける Chaotic[混沌とした]な問題）

問題解決に向けて、「要求は不安定である」を理解し、「要求の根拠を明らかにし、要求の安定性を予測する」手法を学ぶために、以下の要求分析の演習に取り組んだ。

#### ① ペルソナ法

システムを利用する架空の人物を設定し、その人物がシステムを使用する状況を想像することで要求分析を行う手法。人物を特定するために必要な情報を整理することでシステムの対象を明確にできる。人物の行動まで想像することで具体的な要求抽出が期待できる。

#### ② リッチピクチャを用いた現状分析

問題に関連する人たちを、表情を持った人物の絵と発言の吹き出しで描き、大きな1枚の絵として表現する手法。様々な立場の意見や、問題を取り巻く環境の全体を俯瞰できる。直感的に現状把握ができるため、関係者と円滑な情報共有や、新たな問題に気付くことが期待できる。

#### ③ CATWOE 分析

問題の状況を CATWOE の6つの頭文字の要素で表現し、要求分析を行う手法。

- ・C: Customer（顧客）
- ・A: Actors（Tを行う人）
- ・T: Transformation Process（ある状態からWに合致する状態へと変換するプロセス）
- ・W: World View（Oの世界観）
- ・O: Owners（Tを実現することで、Wを達成したいと考えている人）
- ・E: Environment（T達成に必要な環境と資源、守るべき制約とルール）

6つの要素を定義することで誰（Owners）が何（World）を求めているかを明確にできる。

また、同じ問題に対して Owners の定義を変更して分析を繰り返すことで、立場の違いによる要求のズレや、要求の根拠を把握することができる。

#### ④ ゴール指向分析

ゴール（＝システムが要求を満足することによって達成できる目的）を、ゴールを達成するために必要なサブゴールに分解し、階層的に記述する手法。最終ゴールを達成するために必要な作業のモレに気付いたり、サブゴールを誰が実施するのかを整理したりすることができる。

### ■有効性：

要求の根拠まで分析し要求抽出を行うことで、状況の変化に対して安定した要求を獲得することが期待できる。また、分析時の状況（条件）が残っているため、抽出した要求が新しい状況下でも利用可能かどうかを予測することが可能となる。

### ■留意点：

分析に時間をかけすぎると、安定していた要求も再び不安定になったり、タイムリーなシステムの提供ができなくなったりする恐れがある。

解決したい問題や、分析にかけるべき時間などを考慮して、適切な分析手法を選択することが重要だと考えられる。

## ●第6回（例会）：アジャイル開発：天野 勝氏（株式会社永和システムマネジメント）

### ■概要：

多くのアジャイル開発手法で採用されているイテレーション開発を、折り紙で多面体を作成するプロジェクトを通して擬似的に体験した。演習の具体的な体制や内容については、下記のような形式で実施した。

#### {工程}

2リリース（2イテレーション）を実施。1イテレーションは4日間。

初日にはイテレーション計画会を実施。2～3日目は朝会を実施後、開発を実施。

4日目は朝会・開発の後にレビュー会を実施し、リリースを行う。

2イテレーション目は、上記作業に加え、初日に振り返り会を実施する。

#### {体制}

プロダクトオーナーが1名。他のメンバーは開発者として実施。

#### {要件管理・作業見積もり}

プロダクトオーナーがリリースで実現する要求を決定し、プロダクトバックログを作成し、優先順位を決定する。その後、開発者が優先度の高い要求からタスクに分けて見積もりを実施。

#### {進捗管理}

各開発者の作業の可視化にはタスクボードを使用。プロジェクト全体の進捗状況については、バーンダウンチャートを使用。

#### {振り返り}

メンバー全員で KPT（Keep/Problem/Try）を実施。

### ■有効性：

アジャイル開発を実際に体験し、バーンダウンチャートやタスクボードによる進捗状況の透明化、朝会・振り返り会の実施による現状の可視化の容易性を肌で感じる事が出来た。正しく軌道に乗せられれば短納期での開発手法として有効であると感じた。

### ■留意点：

演習にて最も進捗速度が上がったタイミングは、皆が同じゴールに向けて進んでいると感じた瞬間であった。また、予想外の出来事が発生した際、皆の動きがバラバラになり、効率が一気に落ちたと感じた。

アジャイル開発では担当者が主として進めていく為、他の開発手法以上にチームとして纏まっていないと効率が上がらず、この点についても考慮が必要であるだろうと感じた。

## ●第7回（例会）：工数見積りモデルの構築手法：石谷 靖氏（株式会社三菱総合研究所）

### ■概要：

工数見積りの根拠が見えないと、発注者側は妥当性が分からない、受注者側は妥当性を説明できない、という問題に繋がる。このような問題の解決策として見積り根拠の「見える化」が重要であり、その方法として「CoBRA（Cost estimation, Benchmarking and Risk Assessment）法」を学んだ。

CoBRA 法は、見積りモデルを作成する方法である。この方法は、「同じ規模（例：プログラム行数）のプロジェクトでも、かかった工数（例：人月）に違いがある」という状況をベースの生産性  $\alpha$ （規模に比例する部分）と、そこからのオーバーヘッド（CO）（プロジェクト特有のコスト変動要因（以下、変動要因とする））の二つに分けて考える。

$\alpha$  は、過去のプロジェクトのデータを利用し算出する。CO は、熟練者の経験を利用して、モデルを作成する。

演習内容は、以下の通りである。

- ・グループで各自の経験に基づいて変動要因の洗い出しを行い、変動要因を定義した（影響度レベル（工数影響の大きさ）の定義を含む）。

- ・IPA で公開されているツールとモデル例を使用して、下記の手順でモデル構築とモデルの改善を行った。

1. ツールの起動
2. 変動要因の選択
3. プロジェクト実績データの入力
4. 見積りモデルの構築および改善

- ・グループで定義した変動要因に対する改善策をグループで議論した。

### ■有効性：

過去のプロジェクト実績に加えて、熟練者の経験を変動要因（要件の不安定さ、性能要求のレベル、開発期間の制約等）としてモデル化しているため、従来の「勘」ではなく、説明力のある見積りが実現可能である。

熟練者2-3名の協力と10件程度の過去プロジェクト実績データによりモデル構築ができるため、容易に構築できる。

ツールを利用することで、変動要因レベルを変動させれば工数超過リスクがどの程度になるかをシミュレーションすることができる。また、その結果をもとにリスク・コストのコントロールや改善を行うことが可能となる。

### ■留意点：

規模の見積りは、CoBRA 法の範囲外のため、別途、規模を見積もる必要がある。

変動要因は数多くなりがちであるが、10 個程度に抑える方がよい。数が多くなった場合、一旦、7 個程度にして、妥当なモデルになるまで数を増やしていくとよい。

見積り誤差が大きい場合は、以下の点を確認する。

- ・規模、工数の実績データに誤りは無いか。
- ・当該プロジェクトに特有な要因の抜けが無いか。
- ・特殊性が無いか（規模が巨大または極小か、開発が新規または改修か等）。

特殊なプロジェクトである場合は、モデルを分けるかモデルからの除外を検討する。

## ●第8回（例会）：テスト演習：鈴木 三紀夫氏（ASTER）

### ■概要：

テスト技法の基礎を学ぶことを目的とし、演習を通して各技法の特徴や使用方法を学んだ。

演習は、まず各受講者が演習問題を解き、その解答を基にグループ討論を行った。その後、演習問題に適用できるテスト技法の特徴や使用方法の解説を受けた。テスト技法としては、以下について学んだ。

- ・制御フローパステスト
- ・同値分割
- ・境界値分析
- ・ドメインテスト
- ・デシジョンテーブルテスト

### ■有効性：

テスト技法を適用する事で、下記の様な有効性がある事を学んだ。

1. 条件分岐が複雑な場合、制御フローを書く事で、条件の整理を行う事ができ効率良くテストケースの作成が行える。
2. デシジョンテーブルを適用する事で、ドメインから仕様に落とし込む事も可能であり、テストケースの抜けや漏れを防ぐことができる。
3. 同値分割では、要素が増えた場合、ドメインという考え方を取り入れる事で、ケースの洗い出しを積み上げて検討する事ができ、したがって効率よくテストケースを作成する事ができる。
4. テスト技法を適用する事で、テストケース作成者の違いによるバラツキを抑える事ができ、テストの抜け・漏れを防ぐことができる。

### ■留意点：

テスト目的やどの様なバグを検出するかを考慮せずに、単にテスト技法を適用しただけでは、意味のないテストケースになってしまうため、テスト目的やどの様なバグを検出するかも考えて、テスト技法を適用する必要がある。

また、テスト技法も万能ではなく、テストレベルや目的に応じた技法を適用する必要があるため、テスト技法の特徴や種類、得手不得手も事前に知っておく必要がある。

テスト技法を知らない場合、テストケース作成者の違いによってバラツキが発生するため、テスト基準を明確にしたガイドラインを作成する事で、関係者間で認識を合わせる事ができる。

●第9回(臨時会):メトリクスの基礎と GQM 法によるゴール指向の測定:鷺崎 弘宜(早稲田大学/国立情報学研究所)

■概要:

ソフトウェアの品質を把握し改善するために、品質測定評価方法と落とし穴、コツの説明と演習を実施した。

ソフトウェア開発は厳しいものであるが、品質を把握し改善するためのメトリクスの重要性和限界を知り、意思決定に役立てる必要がある。

① 代表的なメトリクス

主なメトリクスとして「規模:コード行数 (LOC)」「規模:ファンクションポイント法」「複雑さ:サイクロマティック複雑度」「欠陥:欠陥数, 欠陥密度, 欠陥除去率」などがある。

② メトリクスの落とし穴「コツ」

メトリクスの落とし穴として、「ホーソン効果 (注目された結果, 取得したメトリクスのみ見かけ上のみ上がる)」「オレオレ品質 (井の中の蛙状態で品質がいいと判断してしまう)」などがあり, コツとして横に広げて多面的にみることがあげられる。

③ ゴール指向の測定 GQM

GQM (Goal-Question-Metric) は明確に目標を据えて、目標に対して必要なメトリクスを対応付けるゴール指向な枠組みである。

測定上の目標 (Goal), 目標の達成を評価するための質問 (Question), 質問に回答するために必要な定量的データを得るための主観的・客観的メトリクス (Metric) で構成される。

④ 組織目標, 戦略測定の整合化 GQM+Strategies

GQM の枠組みをつかって、目標に対してメトリクスを取得し評価することができたとしても、有効な組織アクションに繋がっていないという落とし穴に陥ることがある。

そのための手法として、「GQM+Strategies」というモデルがある。

上位組織目標と戦略に対してメトリクスを設定し、その上位戦略を下位の組織目標と戦略に分解しメトリクスを設定するという、縦にアクションを繋げる方法である。

■有効性:

GQM+Strategies により、組織目標にひもづいた測定と改善が実施可能となる。

会社の上位組織目標や戦略からのメトリクス収集および改善を実行することも可能であるが、自部門の目標や戦略からのメトリクス収集および改善を実行するなど、実施規模を調節することが可能である。

■留意点:

GQM+Strategies における測定プログラムを運用するにあたり、まずは、人材と責任を含めた組織とのコミットメントが必要である。その上で、適用範囲や目標を決めて測定、分析、改善のプロセスを実施し、必要であれば目標や戦略を見直ししながら繰り返すことが必要である。

以上