

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

レビューについて学び、演習を実施した。

演習はレビューの視点を養うことと他の視点を知ることを目的とし、「インターネットの広告付き無料WEBメールシステム」の簡単な設計図を題材として実施した。

演習内容は以下の通りである。

- ①レビューを実施する。
- ②ソフトウェアの品質特性の視点でレビューを実施する。
- ③他の視点を知るため、レビューの指摘内容を共有する。

■有効性

- ・レビューはソフトウェアテストより早い段階で実施され、ソフトウェアテストと同様に欠陥を発見できるため、低コストで欠陥を発見できる。また、ソフトウェアテストでは発見することが難しい欠陥を見つけることができる。
- ・専門家などの意見を聞くことにより、より良い設計にすることができる（設計は唯一絶対ではないため、専門家などの意見は貴重である）。
- ・レビューを実施する時間は限られているため、合意の上でポイントを絞りレビューを実施することで、効率的に改善点を発見できる。

■留意点

- ・レビューは公式レビューと非公式レビューがあるが、定義は組織により異なるため、注意する必要がある。
- ・レビューに使用するチェックシートは、レビュー時に確認するのではなく、設計前に学習すべき内容である。
- ・レビューの実施が目的となる形式的な確認では、効果のないレビューとなる。
- ・メリハリのないレビューは、レビュー効率が落ちるため、目的を明確にしてレビューを実施する。

●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基盤である「モデリング」が開発でどのように役立つのかを知るために、要求のモデリング、設計のモデリングに関する演習を行った。

要求のモデリングに関しては、まずストップウォッチを題材にしてシナリオ分析のひとつであるユースケースを使ったモデリングを演習で体験した。ユースケースによるモデリングではシステムを利用するユーザであるアクタとシステムとの相互作用をユースケース図で示した後、ユースケース記述により事前条件、事後条件、イベントフローを詳細に定義する。次に同じストップウォッチを題材にしてUMLを使ったモデリングをステートマシン図とシーケンス図の作成を通して体験した。ステートマシン図によるモデリングではシステムの状態とイベント、状態間の関係、状態遷移に伴うアクティビティを定義し、シーケンス図では各クラスの動作シーケンスをトレースする。

設計のモデリングに関しては、クラス図から設計の問題点を指摘する演習を行い、オブジェクト指向設計は分析モデルを基盤として設計モデルを構築することにより、要求と設計を繋ぐ手法であることを学んだ。

■有効性:

①要求のモデリング

要求をモデリングすることにより、要求の漏れ、課題、未決定事項、矛盾等が見えてくる。ユースケースのようなシナリオ分析は習得に時間がかからないわりに効果の大きい方法であり、UMLを使ったモデリングは、正しくルールに従って記述する必要があるため、学習コストが高いが、静的構造、動的構造、機能的構造等を正確かつ詳細にモデリングすることができる。

②設計のモデリング

設計のモデリングとはソフトウェアの構造が見える化することであり、ソースコードで読み取れる範囲よりも広い範囲を俯瞰することができる。これによりレビューがしやすくなるだけでなく、ソースコードを書く前に問題を発見することもできる。

■留意点:

要求として出てくるのは要求元が興味、関心のある部分のみでありヌケモレが多いため、モデリングにより要求を分析し、抜けている部分の要求も抽出することが大切である。また、対象システム、分析スキル等に適したモデリング方法を選択することで効率の良い分析を行うことができる。

さらには静的構造、動的構造、機能的構造をUMLでモデリングしてモデルベース分析を行い、要求分析モデルをベースに設計モデルを構築することでオブジェクト指向設計が可能になり、モデル上での設計・シミュレーション、ソースコードの自動生成を活用したモデル駆動開発で生産性の向上に貢献することができる。

●第3回(合宿):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)の概要を学び,用意された題材に対して「ストーリーボード」「スケッチ」「ペーパープロトタイピング」を実践した.内容は以下の通り.

1. オリエンテーション
演習の流れおよびUXデザインのプロセス概要説明
2. 題材説明
テーマとペルソナの説明を受け,解決案のフリーディスカッションを行う
3. ストーリーボード
ペルソナ視点で体験を記述する
4. スケッチ&プロトタイプ
機能,データを洗い出してグループ化・階層構造化し,スケッチによるプロトタイプを作成する
5. テスト準備
プロトタイプによるテストを準備する
6. テスト
ポスターセッションを利用してプロトタイプのテストを行う.
7. 振り返り
テストの結果を共有する.

■有効性:

ペルソナの性格や趣味趣向を詳細に決め,ペルソナの体験をストーリーとして記述することで,ユーザの視点や感情を反映した脳内モデルを構築できる.

スケッチは安価であるため,何度も失敗できる.そのため,多くの案を出して比較・評価することで,良いアイデアを模索することができる.

■留意点

ストーリーボードを作成する際には,ペルソナを主語として行動・振る舞いを書く.また,ペルソナの体験・期待・希望をペルソナの言葉で記述し,ペルソナの感情を表す絵を入れる.ストーリーボードは,システムの要件や画面遷移ではなく,ペルソナにとっての区切りごとにコマ割りをを行う.常に主語がペルソナであることを意識しないと,つい開発者の視点でモデルを捉えがちなので,注意が必要である.

スケッチの前に,機能・データの関連性を見つけて整理する.それをもとに,グループ・階層・掲載順序について検討する.その後,実際に複数のレイアウトをスケッチし,ベストなものを選び,プロトタイプとする.テスト時にターゲットユーザーが戸惑うことなく操作できるよう,プロトタイプ作成時に,視線の誘導を意識したり,操作の結果を実感しやすいギミックを用意したりすることで,よりスムーズにテストが行える.

●第4回(臨時会):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:

アーキテクチャの品質特性とそれを表すシナリオ, アーキテクチャの一般的な実現手法と分析評価の概要について学び, 演習ではその手法を用いて, 「ドーナツ店の店内レイアウト」をテーマにしたアーキテクチャの分析/評価を実施した。

アーキテクチャは, ビジネス要件, システム要件, 前提条件, 制約条件などに基づいて導かれるシステムの骨格である。

本演習では, アーキテクチャの分析/評価の一連の流れとして, ①品質特性シナリオ ②ADD ③ATAM について学んだ。

① 品質特性シナリオ

アーキテクチャの設計にあたり, まず目標とする品質特性を明らかにするため, 「可用性」, 「変更容易性」, 「性能」, 「セキュリティ性」, 「テスト容易性」, 「使いやすさ」といったシステムが備えるべき品質特性のシナリオを記述する。

品質特性シナリオを作成することによって, 具体的にない非機能要件の分析を可能にし, 目標とする品質を明らかにすることができる。

②ADD(Attribute Driven Design)

ADDは, 品質特性と機能要件を両立させるアーキテクチャを目標とする設計手法である。

まず, 作成した品質特性シナリオの中から, 最もアーキテクチャに影響を与えるものを選択する。そして, 選択した品質特性シナリオに対し実現方法を割り当てることで, アーキテクチャを設計していく。

③ATAM (Architecture Tradeoff Analysis Method)

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手法である。アーキテクチャの評価・分析手法としてATAMを適用することで, 実現手法のリスクや品質特性を満たすための重要ポイント, およびトレードオフポイントが明確にすることができる。

■有効性:

アーキテクチャによってシステムの品質が実現されるため, ソフトウェアシステムの開発において, アーキテクチャを適切に分析/評価することで, 目標品質の実現に貢献することができる。また, アーキテクチャの決定過程においてシステムの理解を助け, 問題発生時のリスクの緩和にも期待できる。

■留意点:

アーキテクチャ設計における品質特性シナリオの記述では, 非機能要件の洗い出しを行う際に, 具体的な実現手段を考慮しないよう留意する。これは, 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である。また, 複数の観点から非機能要件を抽出するためには, 様々なステークホルダの観点から検討を行う必要がある。

ATAMによる評価においては, 実施すべき手順が多く, 工数や人員等の問題でマニュアル通りの実施が困難となる場合が多いため, 適用するシステムの環境に合わせ, 都度カスタマイズして実施することが望ましい。

●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、利用者や顧客、また状況に応じ変化する要求を抽出する技術（要求工学）には様々な手法がある。それらの手法の適用には、明らかにする対象を定め、最適な手法を選択・適用する必要がある。

本演習では、ステークホルダ分析・現状分析から要求の抽出までの一連の流れについて、「大型ショッピングモールでの買い物シーン」をテーマに各種手法の演習を行った。実践した手法の概要を以下に示す。

1. ペルソナ法

システムを使用する架空の人物像（ペルソナ）を設定し、その人物の使用状況を想像することで要求分析を行う。人物像を事細かに設定することでシステム対象を明確にでき、行動を想像することで具体的な要求を抽出できる。

2. リッチピクチャによる現状分析

漫画・吹き出し・その位置関係を用い、各関係者（ステークホルダ）の利害関係・意見・ギャップを表現する。問題の状況を可視化、直感的な現状（世界観）の把握に役立てる。

3. CATWOE 分析

以下の6視点で、それぞれの立場から、問題をどのように考えているのかを把握し、立場による世界観の相違を理解する。

C : Customer（顧客、Tの受益者・犠牲者） **A** : Actor（Tを行う人）

T : Transformation Process（ある状態からWに合致する状態へ変換するプロセス）

W : World View（Oの世界観）

O : Owner（Tを推進/中止すべきと考える人）

E : Environment（Tの達成に必要な環境と資源、守るべき制約とルール）

4. ゴール指向分析（KAOS法）

システムが要求を満足すると達成できる目的（＝ゴール）を詳細化、いくつかのサブゴールに分解を繰り返す。分解により、目的達成のための手段（＝要求）を明確にする。

■有効性：

モデル化手法を使い分け、関係者から要求の根拠・背景を幅広く引き出すこと、完成したモデルを他関係者で共有し、要求抽出における抜け漏れを防止できることを体験した。

例えば、ペルソナ法やリッチピクチャは、同手法の認識合わせにより参加者の議論が促進され、個人ワークに加えて、多数の視点がグループワークで挙げられた。

ゴール指向分析は、サブゴールで上位のゴールの依存関係を抜け漏れなく検討する効果を実感した。

各手法は、その視覚的な情報整理の性質から、関係者の手法理解度に差があっても、最低限の説明を行い活用すれば、一定の効果があると学んだ。顧客やプロジェクトメンバの事前知識を揃えられない実案件適用にあたり、今回の演習で得られた重要な気付きである。

■留意点：

各手法に留意点がある。例えばCATWOE分析は、各ステークホルダの立場による世界観の違いを捉えることが重要である。ゴール指向分析は、ゴールの『状態』を明確にし、曖昧な表現を避ける必要がある。

●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:

昨今は、ビジネススピードが速くなりそれに伴いユーザの要求が日々変化していくような状況であり、その中で今までのソフトウェア開発の代表的なスタイルであったウォーターフォールを実施していると、完成した時点ですでに要求が古くなっている可能性がある。そこでビジネススピードに合わせてソフトウェアを提供していこうと考えられたのが「アジャイル」という開発手法である。

アジャイル開発が登場した背景やその標準的な開発の進め方を学んだうえで、多くのアジャイル開発手法で採用されているスプリント開発を、折り紙で多面体を作成するという疑似プロジェクトを利用しながら体験することで理解する。

疑似プロジェクトの進め方は以下のような進め方を行った。

<体制と役割を決定する>

PO(プロダクトオーナー) 1名: ビジネスの責任者として、要件管理/製品リリースの責任を持ち、成果物の価値を最大化する。

SM(スクラムマスター) 1名: プロジェクトを円滑に進め、チームのパフォーマンスを最大化する。

開発者 4~5名: 計画会、朝会、開発作業、振り返りを行いながら要求を具現化する。

<スケジュールについて>

1スプリントを4日間(1日=11分とする)とし、2スプリントを実施。

1日目はスプリント計画会を実施。

2~3日目は朝会(4分)と開発(7分)を実施。

4日目は朝会、開発をおこなったうえで、レビューを実施しリリースする。

2スプリント目は上記の前に1スプリント目の振り返り会を実施する。

<要件管理の方法>

POが要求を決定しPBL(プロダクトバックログ)として管理する。PBLの優先順位についても最終決定はPOが行う。

<見積>

優先度の高い要求から、実現に必要なタスクを開発者が洗い出し、1タスク2分程度でできる作業(タスク)に分けて見積を実施。

<進捗管理の方法>

タスクボードを用いて各開発者の作業の可視化をおこなう。プロジェクト全体の進捗管理にはバーンダウンチャートを用いる。

<ふりかえり会の実施>

メンバー全員でKPTA(Keep/Problem/Try/Action)を行う。

■有効性:

アジャイル開発では、メンバー全員による朝会により開発状況を把握することができる。それにより、問題点を早く認識でき速やかな対策を打つことが可能になる。また、スプリント毎に実施する振り返りにより、プロジェクトを進めていくうえでの良かった点/悪かった点が共有でき、次のスプリントに向けての改善ができるため、チーム全体の生産性や質をあげることができる。

■留意点：

今回の「折り紙多面体作成プロジェクト」の品質要求では「作成した多面体が自立していれば良い」程度であったために、単純に見た目で判断できるが、実際のプロジェクトにおいては、品質保証に関する活動計画について初期の時点で検討し実施しなければ、後になって膨大な品質保証活動及び開発のやり直しが発生する可能性がある。また、ウォーターフォール開発のように開発前に要求やそれに対する成果物が明確に決められており、それに向かって適切な期間をかけて開発をする場合は、見積→受注に至るまでスムーズに進められるが、アジャイル開発においては、ニーズ（要求）が変化することも十分にあるために、発注側と受注側の契約内容や条件等に十分な考慮が必要になってくると考えられる。

●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：

工数見積りモデルの一つである「CoBRA（Cost estimation, Benchmarking, and Risk Assessment）法」について理解する。

CoBRA 法の特徴は見積りモデルを生産性と開発規模（例：SLOC）だけで構成するのではなく、工数に影響する様々な変動要因（例：顧客の性能要求レベル、チームの知識・経験等）も考慮してモデルを構成する点である。

CoBRA 法の見積りモデルは計算式で表現すると以下となる。

$$\text{工数} = \frac{\alpha \times \text{開発規模}}{\text{生産性}} \times (1 + \sum \text{CO}_i) \quad \text{変動要因 (CO: コストオーバーヘッド)}$$

生産性は過去プロジェクトのデータを利用して算出し、変動要因は見積り熟練者の知見を基に洗い出し、定量化する。

演習では以下をグループワークで実施した。

- ・演習参加者が見積り熟練者となって、変動要因の洗い出しと定量化を実施
- ・IPA 公開ツールを使用し、CoBRA 法の見積りモデルを構築
- ・構築したモデルの改善

■有効性：

- ・変動要因を考慮しているため、実態に合った見積りが可能で、見積り誤差を低く抑えられる。
- ・見積り根拠が「見える化」され、発注者と受注者間で根拠に基づいた価格交渉が可能。
- ・IPA 提供のツールがあり、過去プロジェクト（10 件程度）の開発規模および実績工数のデータと、見積り熟練者 2, 3 名の協力があれば、容易に CoBRA 法の見積りモデルを構築できる。
- ・生産性および変動要因は CoBRA 法における算出プロセスが定義されている。

■留意点：

- ・モデルを最初から適切に構築することは非常に難しいため、構築後のモデルに対して改善を繰り返し、モデルの見積精度を向上させる必要がある。
- ・構築後のモデルの改善時には変動要因も見直し対象になるが、見直しの都度見積り熟練者を集めることが困難な場合が想定される。見積り熟練者が集められた際には予め多めに変動要因を洗い出し、定量化しておくことが有効。
- ・開発規模は CoBRA 法における算出プロセスはないため、独自で算出する必要がある。

●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。

◆ホワイトボックステスト技法

- ・制御フローパステスト

◆ブラックボックステスト技法

- ・同値パーティション
- ・境界値テスト
- ・決定表

■有効性：

目的に応じたテスト技法を選択することで、以下の効果が得られる。

1. 一定水準の品質を保ったままテストケースを絞り、テスト実施にかかる工数を削減することができる。
2. グラフや表を作成し、テストケースを可視化することで、テスト内容の誤解やテスト漏れを防ぐことができる。

■留意点：

1. 単にテスト技法を適用しただけでは、意味のないテストになってしまうことがある。そのため各テストケースが、どのような不具合を検出できるか、どのような不具合が潜んでいるか、想像を巡らせてテストケースを作成していくことが望ましい。
2. テスト技法は万能ではない。各テスト技法の得手不得手を事前に把握し、目的に応じたテストケースを適用する必要がある。
3. 組織内、関係者内でテストケース作成方法や、テストケースの表現方法が異なると効率的なテストが実施できないため、共通の方法でテストを実施する必要がある。

●第9回（臨時会）：メトリクスによるソフトウェア品質把握と改善：鷲崎 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、落とし穴や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にメトリクスの重要性と限界を知り、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）や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や結合度」「欠陥：欠陥密度や欠陥除去率」などがある。それぞれ使いどころがあり，ソフトウェアの特定の側面を捉えるために使用する。

② メトリクスの落とし穴と「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，そのメトリクスの見かけ上の結果が上がる）」「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，多面的や客観的な視点で，メトリクスを見ていくことが重要。

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。GQMの拡張では，過程（Hypothesis, Assumption）と解釈（Interpretation）が追加される。過程は，質問導出にあたり仮定している事柄であり，解釈は目標の達成・未達成を判断する基準である。

④ 組織目標，戦略測定の整合化 GQM+Strategies

GQMの枠組みを使い，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」という概念モデルがある。

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，組織目標にひもつ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も可能である。

■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ながら繰り返す必要がある。