

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

■概要 :

ソフトウェア開発工程における欠陥を見つける手段としてレビューについて学んだ。
演習は「SNS システム開発」を例として個人でレビューを行い、その後グループにわかれ結果を共有し、集約した結果を研究員全員に対して発表した。

当日の流れは以下となる。

- ① 個人でのレビュー実施
- ② 個人で重視する品質特性を決めレビュー
- ③ グループでの結果の共有と集約, 発表

■有効性 :

- ① レビューを実施することでテストよりも早い段階で欠陥を検出することができ、手戻りが少なくなり、費用の削減や工数の短縮につながる。
- ② 専門家からの意見を得られる機会となるため、よりよい設計を行うことができる。
- ③ テストでは見つけにくい、見つけられない欠陥を検出することができる。
- ④ レビューに同席することで設計のポイントを学ぶなど教育の場としても活用することができる。
- ⑤ 担当者ごとに視点が違うため、観点に縛られず確認を行える。

■留意点 :

- ① 書式や用語の認識などのフォーマットをそろえない場合は効率が落ちてしまい、欠陥を見つけることが困難になる。
- ② フォーマットがそろっていない場合はレビューアの負担が大きくなるため、セルフチェックの実施やチェックシートを用いた認識の共有などが必要となる。
- ③ レビューにかけられる時間は限られているため、重点を決めて問題を早く見つけ出すことが必要となる。
- ④ 品質特性の項目はトレードオフのものが多く存在するため、どれを優先し、そのために犠牲になった項目がどれだけ救えているのかを見ることが大事となる。
そのため、ある品質特性を優先するあまりほかの品質特性が使い物にならない場合は問題となる。
- ⑤ 運用後に問題が発生した場合、どうすれば気づけたのかをグループで共有する。
- ⑥ チェックシートは設計前に作成しておき、設計に取り入れる(作りこみの品質を向上させる目的)

●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とモデリングについて学んだ。

目的は、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ことと、モデリングをはじめめるきっかけ作りである。

演習内容は以下の通りである。

(1) 要求のモデリング

要求のモデリングとは何を要求されているのかを分かりやすい形で表現することを意味する。

- ① ユースケース図を用いて要求の全体像を把握する
- ② ユースケース記述を用いてシステムの完成イメージを明らかにしつつ、要求の不足を洗い出す
- ③ ステートマシン図を用いて要求の不備を洗い出す

(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を見える化することを意味する。

- ① クラス図を用いてソフトウェア構造の問題点を洗い出す

■有効性：

- 要求のモデリングを行うことで、製品/システムの完成イメージの早期の共有、要求の正しい理解、不備不足の発見が可能となり、手戻りコストの削減につながる。
- 設計のモデリングを行うことで、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のでレビューがしやすくなり、さらに、ソースコードを書く前に構造の問題を発見できるので品質向上、手戻りコストの削減につながる。
- モデルを正確に記載すれば、オブジェクト指向言語のコードへの変換が可能となり、モデルからソースコードへの変換コストの削減につながる。

■留意点：

- 要望、要求、要件、仕様をきちんと区別したうえで、正しく要求のモデリングを行わなければならない。
 - 要望：利害関係者が実現してほしいと思っていること
 - 要求：要望のうち、システムで実現することを決めたもの
 - 要件：要求を満たすための「システムの振る舞い・特性」を定めたもの
 - 仕様：要件が網羅された状態
- 分析モデルを設計モデル検討・構築のスタートポイントの一つとして利用する。
- モデルをコードに変換するためには、それだけの情報を持つモデルを正確に(※)記載しなければならない。

(※)正確に：UML 文法として誤りがない，かつ内容に誤りがない

演習コース I 「ソフトウェア工学の基礎」付録

●第 3 回 (例会) : 人間中心設計 (UX デザイン手法) : 金山 豊浩氏 (株式会社ミツエーリンクス) / 村上 和治氏 (東京海上日動システムズ株式会社)

■概要 :

UX (User Experience) デザインの概要を学び、用意された題材に対して「ストーリーボードの作成」「機能・情報洗い出し」「スケッチの作成」「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, UX デザインのプロセス概要の説明を受ける
2. 題材説明
テーマとペルソナの説明を受ける
3. コンセプト化
アイデア出しのブレストを行う
4. ストーリーボードの作成
ストーリーボードの描き方の解説を受け, ストーリーボードを作成する
5. 機能・情報洗い出し
機能・情報洗い出しの説明を受け, 機能・情報洗い出しを行う
6. スケッチの作成
UI デザインパターンの説明を受け, スケッチを作成する
7. プロトタイピング
プロトタイプを使ったテストのやり方の説明を受け, プロトタイプの準備(プレテスト)・テストを行う
8. 振り返り
テストの結果を共有する

■有効性 :

- ・ストーリーボードによって, ユーザのニーズを明確にすることができる。
ストーリーボードには, シナリオ(モデル化したユーザの行動)を記述する。ストーリーボードの作成によって, ユーザの行動の前提や制約を整理することができる。作成したストーリーボードと, そこから導き出したニーズを共通認識として, 機能・情報洗い出し以降のプロセスを有利に進めることができる。
- ・スケッチによって, より良い UI デザインを早期に作成することが出来る。
スケッチは低コストで作成できる。そのため, スケッチをたくさん作成して, ユーザに評価してもらうことで, ユーザのニーズに合った UI デザインを早期に作成することが出来る。
- ・プロトタイピングによって, 完成品の質を高めることができる。
プロトタイプを作成し, それをユーザが実際に使用することで, ユーザの視点で機能・情報を検証することができる。検証結果をフィードバックすることで, より高品質な完成品を作成できる。

■留意点

- ・ペルソナの設定が曖昧だと, その設定から考えたシナリオも曖昧になる。そのため, ストーリーボードを作成する際は, ペルソナのプロフィールを具体的に設定する。ペルソナの年齢, 性別, 家族構成など, 各要素について詳細に設定するのが望ましい。
- ・シナリオの作成にあたっては, 網羅性が重要となる。そのため, シナリオとプロトタイプの作成にあたっては, 質より量を意識する。

●第 4 回（例会）：要求工学（要求分析）：中谷 多哉子氏（放送大学）

■概要：

要求工学とは、ソフトウェア開発において要求抽出を実践して仕様化するプロセスの技術である。要求抽出には様々な手法があり、最適な手法を選択して適用することが重要である。

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、「新型コロナワクチン接種のシーン」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、その人物の利用状況を想像することで要求分析を行う。人物像の設定をより深く詳細に設定することでユーザのニーズに合ったシステムの要求を抽出することができる。

② リッチピクチャを用いた現状分析

現在の状況を把握するために、表情を持った人物・吹き出し・位置関係を漫画で書き出し、状況を取り巻く世界観を可視化する手法である。

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, T の犠牲者もしくは受益者)
- ・ A : Actors (T を行う人々)
- ・ T : Transformation Process
(ある状態から世界観 W に合致する別な状態へと変換するプロセス)
- ・ W : World View (O の信念, 根拠, 世界観)
- ・ O : Owner (T を実現すれば, 世界観 W を達成できると考えている人)
- ・ E : Environment (T 達成に必要な環境と資源, 守るべき制約とルール)

③ ゴール指向分析

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（＝ゴール）をいくつかのサブゴールに分解して、システムが仕様を満たすことによって、実現されること（＝要求）および要求を満たす手段（＝仕様）の関係を明確化する。

■有効性：

様々なモデル化手法を選択・適用することで、ソフトウェア開発における要求抽出の根拠や背景を共有できるとともに、抜けや漏れを抑止することが可能である。

ペルソナ法やリッチピクチャのグループ演習においては、手法を用いることで要求抽出の方向性を参加者で共有ができ、様々な議論が促進されたことを実感できた。

■留意点：

手法の適用については、ソフトウェア開発の内容に合った要求工学の手法を選択することが重要である。

リッチピクチャを使用する際は、ただ単に人物や吹き出しを感想のように描くのではなく、世界観を表現した人物・吹き出しとして描かないと正しい分析結果が得られないため注意が必要である。

●第 5 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) 法を学び、「CoBRA 構築モデルプロセスの実施」「CoBRA モデルの改善」「CoBRA モデルを使ったリスク分析・コストコントロール」「変動要因と改善策」を実施した。

ここで、CoBRA 法の見積式を下記に示す。

$$\text{工数(コスト)} = \alpha \times \text{規模} \times \left(1 + \sum CO_i\right)$$

α は「生産性」を表しており、実績データを用いて算出するものである。

これに対し CO は「コストオーバーヘッド」を表しており、経験豊富な PL 等の熟練者の知見を基に定量化したものである。

つまり CoBRA 法とは、従来の「経験ベース型」と「データ駆動型」の「混合（ハイブリッド）型」といえる。

構築したモデルに対し、要因関係図の見直しや、規模・工数・変動要因のレベルの見直しによる改善を繰り返し行うことにより、CoBRA モデルが完成する。

■有効性：

- 組織固有のコスト変動要因をモデル化することが出来る。
- コスト変動要因に、熟練者の優れた「勘」「経験」を反映させられる。
- コスト変動要因と影響度が見える化によって工数の説明力(価格交渉にも活きる)を向上させられる。
- 影響度の高い要因の把握により、コストマネジメント力を向上させられる。
- 工数変動量から予算超過確率を把握することで見積りリスクの把握ができる。
- 熟練者の優れた知見をモデル化し、共有・活用することでアセット化と属人性排除が可能である。
- 組織に共通する要因を把握し、軽減・解消することによるプロセス改善が可能である。

■留意点：

- モデル構築には、熟練者 2, 3 名の協力と、過去プロジェクト 10 件程度が必要になる。
- 概要に記載した通り、いきなりモデルを完成させることは難しいため、一旦構築したモデルに対し改善を繰り返し行う必要がある。
- 見積り誤差が大きいプロジェクトに特殊性(規模が巨大もしくは極小、新規・改修による開発種別の違い等)があった場合、モデルを分けることや、モデルからの除外を検討する必要がある。

●第 6 回(例会)：アジャイル開発の基礎知識：天野 勝 氏(永和システムマネジメント)

■概要：

ビジネス展開が急速に進む昨今，システム開発もそのスピード感に追従することが求められている．その中でシステム開発を行うには，変化が早く大きい市場のニーズに対してタイムリーに価値を提供して行くこと，作ったシステムがその後のニーズの変化に柔軟に対応することで価値の維持/向上をしていくことが非常に重要である．ウォーターフォール，イテレーション，フロー…様々な開発手法があるが，より機敏に動けるよう考えられたのが「アジャイル」という開発手法である．アジャイルにはリーン，スクラム，XP 等の代表的な開発手法があるがスクラム開発の中の要素の一部である「スプリント」を演習にて体験した．

演習の進め方は下記の通り．

・役割の決定

プロダクトオーナー：製品リリースに責任を持ち，プロダクトの価値を最大化する． /1 名

スクラムマスター：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． /1 名

開発者：スプリントのプロセスを通じ，価値を作る． /2~4 名

・スプリントを回す

下記の①～⑥を 1 回のスプリントとし，2 回スプリントを回す

① スプリントプランニング：プロダクトオーナーと開発者で目標や戦略を決める．

② 朝会/開発：朝会では現状とその日の行動の確認を行う．開発時，スクラムマスターはチームの問題点を発見し取り除き，プロダクトオーナーは次のスプリントの戦略を考える．

③ 朝会/開発

④ 朝会/開発

⑤ レビュー/リリース：スプリントの成果物を確認し，フィードバックを得る．

⑥ 振り返り：チームが現状より良くなるよう，KTPA(Keep/Problem/Try/Action)の振り返りを行う．

上記を回す中で，作業の進捗管理にはバーダウンチャートを用いる．

■有効性

アジャイル開発では 1 回のスプリントの中で目標や戦略を検討することができるため，ウォーターフォールに比べ要求の変化に対応しやすい．また，チームでのコミュニケーションの機会が多く，日々の朝会やスプリント毎に実施する振り返りを通じて良かった点/悪かった点を共有でき，即座に改善を実施することができる．

■留意点

演習においては高すぎる目標設定をした際に成果物の品質低下が見られた．プランニング時はチームの成果を最大限にでき，かつ品質が低下しない程度の目標を設定することが重要である．また，事前の品質目標の設定，品質保証のプロセスを定義しておくことも必要である．アジャイル開発を進める上で十分な知識がないと失敗に陥りやすい．アジャイル開発の典型的な失敗例は下記の通り．

・そもそもアジャイル開発ではない．

例：アジャイル開発についての理解が乏しいため，ルール違反が多発する．

・開発側が，プロダクトオーナーが要求を整理するのを待っている．

・フレームワーク(基盤)と，アプリケーションを別チームで並行開発する．

・スプリント期間に無理に完了させる．

演習コース I 「ソフトウェア工学の基礎」付録

- スプリント期間内に完了していないのに，計画を見直さない.
- 初期の要求が変更されないまま開発を終える.
- 開発者の成長を見込んで，スキル不足の開発者だけで開発を始める.
- スキルが不足しているのに長いスプリントを採用する.

演習コース I 「ソフトウェア工学の基礎」付録

●第 7 回 (例会): 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:

アーキテクチャの一般的な定義は、コンポーネント(ハード/ソフト)間の関連、思想などで、骨格、基板、土台、権力などで喩えられる。骨格や土台のため、失敗するとシステムテストや運用後に大問題が発生する。アーキテクチャは品質に与える影響が大きく、システムのライフサイクルを通して、維持管理する必要がある。

(1) 品質特性シナリオ:

非機能要件を基に、品質の目標を具体的に記述したシナリオのこと。

アーキテクチャの設計や評価、テストなどに利用する。「刺激の発生源」「刺激」「環境」「成果物」「応答」「応答測定」の構成要素からなる。

(2) アーキテクチャの設計 ADD(Attribute Driven Design):

品質特性シナリオの成果物を、応答測定を満たすような実現手法や、アーキテクチャパターンを割り当て、それを実現するためにモジュール分割を行い、詳細化する。

(3) アーキテクチャの分析・評価 ATAM(Architecture Trade-off Analysis Method):

アーキテクチャが品質特性に関する要求を満たしているかを評価する手法である。

評価者、プロジェクトの意思決定者、利害関係者のメンバにて、準備、発表/調査/分析、検証/報告、フォローアップのステップで活動を推進する。品質特性の重要ポイントとトレードオフポイント、実現手法のリスクを明確にすることができる。

上記の(1)~(3)の手法をグループ演習で学んだ。演習では成果物を“ドーナツ屋”として「品質特性シナリオの作成」、「実現手法の選択(店のレイアウト図作成)」、「アーキテクチャの分析・評価(演習結果発表)」の手順で実施した..

■有効性:

- ・アーキテクチャには、品質特性シナリオ、早い段階の分析/評価が必要。
- ・品質特性シナリオの成果物に対し、実現手法を組み合わせることをアーキテクチャ手法と言うが、1つの成果物に対して、複数のアーキテクチャ手法が出てくる。どのアーキテクチャ手法がベストか、論理的に選択することが重要な仕事。
- ・アジャイル開発では、文書が残りにくいいため、ソフトウェアアーキテクチャを容易に記述できる図として、C4Modelの作成が提言されている。C4とは、以下の頭文字を集めたもの。コンテキスト(Context)、コンテナ(Containers)、コンポーネント(Components)、コード(Code)。
- ・近年のシステムは、複雑で多様化しており、アーキテクチャの重要性がますます高まる。

■留意点:

- ・品質特性シナリオを作成する段階では、具体的な実現手法を考慮しないことが望ましい。
- ・プロセス遵守に重点をおくため、成果物(ドキュメント、ソースコード等)が品質を満たしているかの踏み込みが甘くなる。品質の適切な作り込みと評価が必要。

●第 8 回：ソフトウェアテスト：鈴木三紀夫氏（ASTER）

■概要：

本演習は各テスト設計課題に対し「各自でテスト設計」>「グループ内議論」>「講師による回答例の提示」という流れで進めた。

個人およびグループで考え抜いた後にテスト設計技法を用いた解法に触れることで、Before/After として技法活用による効果を明確に認識することができた。

演習テーマとして扱われたテスト技法は以下の通り。

(1) 制御フローテスト

ホワイトボックステストに適用する。

プログラムで命令が実行される流れを定めた「制御フロー」からテストケースを設計する。通常以下のようなカバレッジ基準を定めて実施する。

- C0（命令網羅，ステートメントテスト）… 命令文（ノード）を少なくとも 1 回実行する
- C1（分岐網羅，デシジョンテスト）… 分岐（エッジ）を少なくとも 1 回実行する

(2) 同値分割

同等に処理されると想定したデータ全てを同じパーテーションに振り分け，各パーテーションから少なくとも 1 個の値を選んでテストする。

(3) 境界値分析

同値分割法の拡張で，パーテーションが数値または順序付け可能な値で構成される場合に，パーテーションの最小値と最大値(または，最初と最後)を選んでテストする。

(4) デシジョンテーブル

条件と結果的に起きる動作を表の行，条件の一意的な組み合わせとその結果として実行するアクションにより表される判定の規則を表の列として，規則ごとにテストする。

■有効性：

- テスト設計技法の活用は設計根拠や十分性を説明するのに有用である。
- また、「制御フローテスト」におけるカバレッジの度合いや「境界値分析」の際の値の取り方などをチーム内で定めておくことで，品質のバラつきを排除する効果も期待できる。

■留意点：

- テストの 7 原則でも “テストは状況次第” と表現されるとおり，テスト設計には正解がないことを留意する必要がある。
- 使い手は方法論の丸暗記だけではなく，本質を理解し都度柔軟に最善の手法を選択できる状態を目指す必要がある。

●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 – 演習を交えた品質測定評価の落とし穴とコツの習得 –：鷺崎弘宜氏（早稲田大学グローバルソフトウェアエンジニアリング研究所 / 国立情報学研究所 / (株)システム情報 / (株)エクスマーシオン)

■概要：

メトリクスは、ソフトウェア開発の品質把握と改善において重要である。メトリクスを有効活用するには GQM の考え方が役立つ。さらに GQM は、GQM+Strategies として企業活動にも応用できる。講義内容は以下の通りであり、(2)、(3)についてはグループ演習も実施した。

(1) ソフトウェアの品質とメトリクス

ソフトウェア品質とは、ソフトウェアの使用時に必要性を満たす能力を決定する属性全体のことである。メトリクスは測定の方法と尺度を意味し、主に以下のようなものがある。

- プロダクト：コード行数 (LOC)、ファンクションポイント (FP)、サイクロマティック複雑度 (CC)、凝集度、結合度、欠陥密度 (DD)
- プロセス：欠陥密度 (DD)、欠陥除去率 (DRE)、消化済みテストケース数

(2) ゴール指向の測定 GQM

GQM は、明確に目標を定めて、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（目標指向）な枠組みであり、以下 3 つから成り立つ。

- 測定目標 (Goal)：測定上の目標
- 質問 (Question)：目標の達成を評価するための質問
- メトリクス (Metric)：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

また、GQM の拡張では仮定 (Hypothesis, Assumption) と解釈 (Interpretation) が追加される。

(3) 組織目標、戦略、測定の整合化 GQM+Strategies

GQM+Strategies は、測定を基にして目標 (Goal, 状態) と戦略 (Strategies, 状態を実現するための施策) を整合するための手法であり、各組織の目標と戦略をつないで木構造で表現したものが GQM+Strategies グリッドである。

■有効性：

- メトリクスは、現在の把握・評価および未来の予測・計画に用いることができる。
- GQM+Strategies グリッドによって見える化することで、組織目標と紐づいていない測定指標や下位組織目標と繋がっていない戦略を見つけることができる。

■留意点：

- メトリクスはソフトウェアの抽象的な特定側面を捉えるものであり、測定方法が異なれば測定結果も変わること留意すべきである。
- GQM において、質問導出の際は、測定対象そのものを目的から見て明確にする、対象の属性を評価者の視点から見て明確にする、対象の特徴を評価者の視点から見て評価する、などの観点が重要である。また G-Q および Q-M はそれぞれ 1:n の関係性であり、横に広げて多面的に見る必要がある。

以上