

●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の欠陥抽出に重要な「レビュー」について学んだ。

演習は Google Jam board 等のツールを利用しながらグループ討議を中心としたオンライン形式で実施された。

本演習コースのテーマである「レビュー」に関しては演習コース参加者の「レビュー視点」を増やす事を演習全体の目的として以下内容をポイントとして学習しながら

「個人演習」⇒「他の参加者との共有」⇒「解説」⇒「グループ演習」⇒「グループ討議」
⇒「発表」⇒「解説」⇒「まとめ解説」

と演習コースならではの「手を動かす事」を中心とした流れで演習は進められた。

【演習のポイント】

- ・レビューの目的
- ・レビューの定義(種類・手法・アンチパターン等)
- ・レビューにおけるチェックシートの活用と留意点
- ・レビューの視点(ステークホルダによる違い、「品質特性」利用の有効性等)
- ・具体的なレビューの方法(手順/査読例、レビューイ視点でのノウハウ等)

■有効性：

- テストと比較して低い実施/対応コスト, 早期欠陥抽出に伴う対応コスト含めて品質向上に対して費用対効果の高い検証手段である。
- 成果物を同じ分野の専門家が精査することにより, 対象となる成果物を常識的に要求されている水準を超えたものにする事が出来る。
- テストで見つけにくい, または見つけれない欠陥 (例「ソースコードにマジックナンバーが埋め込まれていること」) を検出できる。
- レビューに同席することにより, 他のレビューアとの視点の共有やポイントを学び, メンバーの教育ができる。また, レビューイにとってもレビューアが何を考えて指摘をしているかを知ること, 指摘された原因を考えることでレビューイ自身のスキルアップにつながる。

■留意点：

- レビューではランダムな指摘ではなく, システムに求める品質特性, レビューの目的を明確にして実施することが重要である。そのためには品質特性, 異なるステークホルダの視点等を勘案したレビューの視点の決定 (視点の割り振り) が重要である。
- 品質特性をレビュー観点とする際には, 各品質特性の関係性においてトレードオフのものが多々あるため, プロジェクトの特性を鑑みて, どの品質特性を重視するのか優先順位を決めておくことが必要である。ただし, 劣後された事により使い物にならない成果物になっては意味がない。劣後された特性をどれだけ救えているかを見るの事も大事。
- レビュー対象の成果物の成果物の標準化. 特にフォーマット (書式, 用語, 記述ルール等) が揃っていることが重要である。フォーマットが揃っていないと時間がかかるためレビューの効率が落ちる, または欠陥を見逃してしまう。
- レビューアだけでは良いレビューは出来ない。組織/プロジェクトの準備やレビューイによる読みやすい成果物の作成, セルフチェック (チェックシート利用) による一定レベルでの品質確保が重要になる。

※組織/プロジェクト準備の例：

レビュープロセス整備, 成果物の標準化, 設計指針整備, チェックシート, プロジェクト計画, レビューの視点の決定, 等

●第2回（例会）：オブジェクト指向分析設計：井上 樹 氏（株式会社豆蔵）

■概要：

オブジェクト指向分析設計とはシステムの実現手段を分析し、オブジェクトという単位で設計することである。その基盤となるのが「モデリング」である。本演習では要求のモデリングと設計のモデリングを行った。詳細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。要求のモデリングに用いられる手法にはユースケース・クラス図・ステートマシン図・アクティビティ図・シーケンス図が挙げられる。

(2) 設計のモデリング

設計のモデリングとは、作成するソフトウェア・システムの構造を見える化することの意味する。設計のモデリングに用いられる手法にはクラス図・ステートマシン図・アクティビティ図・シーケンス図が挙げられる。

■有効性：

- 要求をモデリングすることで、製品の完成イメージの早期共有ができ、関係者間での認識祖語を早い段階でなくすことにつながる。その結果、手戻りコストの削減および要求通りの製品作成が可能になる。
- 設計をモデリングすることで、ソースコードよりも高い視点から設計内容を確認することが可能になり、質の高いレビューにつながる。

■留意点：

- 要求のモデリング
 - 要望，要求，要件，仕様はそれぞれ意味が明確に異なることに留意する。
 - 各モデリング手法は利用する目的が異なる。各モデリング手法の目的を理解し、作成する製品にあったモデリング手法を選択する必要がある。
- 設計のモデリング
 - オブジェクト指向に対応したプログラミング言語ならば UML とプログラムが 1 対 1 に対応するが、非オブジェクト指向言語（C 言語）の場合は変換方法が複数通り存在する。
 - 設計の主となる入力は要件（要求）である。
- 全体
 - 大半のモデリング手法においてモデル作成用のツールが存在するため活用する
 - モデルからコードを自動生成するツールを利用することでコーディング工数を削減できるが、そのためには正確にモデリングを行う必要がある。

●第3回(例会)：プロトタイピング (UX デザイン手法)：

金山 豊浩 氏 (株式会社メンバーズ) / 三井 英樹 氏 (Weblysts.com) / 村上 和治 氏 (株式会社 SHIFT)

■概要：

UX (User Experience) デザインプロセスの概要を学び、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，題材の説明を受ける
2. 提供価値明確化
ペルソナに提供する価値 (Be ニーズ /Do ニーズ /Have ニーズ)の説明を受ける
3. コンセプト化
アイデア出しのプレストを行う
4. ストーリーボード
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
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)・テストを行い，結果をまとめる
8. 振り返り
テストの結果を共有する

■有効性：

- ストーリーボードでペルソナの行動や振る舞いを記述し，ユーザの体験を可視化することで，チームでユーザーニーズを共通理解することができ，問題発見や意思決定の助けになる。
- スケッチは低コストかつ短時間で作成できるため，素早く改善を繰り返すことでより良いUIを作成することが出来る。
- プロトタイプはコーディング不要でスケッチで簡易的に作成可能であるため，ストーリーボードで作成したユーザ体験の実現性などの問題点や不足点など修正しながら何度もテストを行うことで，完成度を高めること出来る。

■留意点：

- ストーリーボードにはシステム要件やアプリの動作など開発者視点の内容は記述せず，ペルソナの体験にフォーカスして記述すること。
- ストーリーボード作成時にストーリーが長すぎると解釈が分かれてしまう可能性があるため4コマ程度で完結するのが望ましい。
- スケッチはアイデアを生み出す作業であるため，細部にこだわらず多くのアイデアを書き出すよう意識する。
- ウォークスルーを実施する時はユーザになりきる精度を高めるために，ターゲットとなるペルソナを確認し，ユーザのニーズや目的を考慮すること。

- 第 4 回(例会)：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 / 株式会社システム情報 / 株式会社エクスマーシオン）

■概要：

ソフトウェア開発の管理、品質把握や改善において、メトリクスは非常に重要な指標である。しかし、メトリクス取得するだけでは、あまり役に立たない。そこで、GMQ や GMQ+Strategies というモデル化手法と組み合わせることで、測定から改善へのプロセスを明確にすることができる。講義の内容は以下の通りであり、随時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。メトリクスとは、ある測定方法に基づいて測定された尺度を表し、ソフトウェア開発においては、以下のような指標がある。

- 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 GMQ

GMQ とは、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)な枠組みである。

- 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

(3) 組織目標、戦略、測定の整合化 GMQ+Strategies

GMQ+Strategies とは、上位組織目標と戦略に対して測定指標を設定し、さらにその戦略の根拠となる下位目標と戦略に対して測定指標を設定するという方法である。GMQ+Strategies を用いて、各組織をつなぐ組織目標(ゴール)と戦略を木構造で表現したものを GMQ+Strategies グリッドという。

■有効性：

- メトリクスを有効活用することで、ソフトウェア開発の現状を把握し、評価することができる。
- GQM の考え方とメトリクスの測定方法を知ること、未来の予測や計画に用いることができる。
- GMQ+Strategies グリッドを用いて分析することで、組織の上位と下位間の整合性を確保することができる。また、各組織の目標と戦略を洗い出すことで、各組織における責任を明確化できる。

■留意点：

- メトリクス自体を測定することを目標にしない。
- 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必要がある。また、横に広げて多面的に見る必要がある。

●第 5 回(例会)：アジャイル開発の基礎知識：
天野 勝 氏(株式会社永和システムマネジメント)

◎概要

アジャイル開発とは、ソフトウェア開発プロセスの一種であり、ソフトウェアの要求の変化や目標変化に応じて、短期間で「計画、実行、評価」を繰り返す反復（イテレーション）型の開発手法である。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。

スプリントは以下のイベントと開発作業から構成される。

- ・プロダクトオーナー（1名）：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。
- ・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（2～4名）：スプリントのプロセスを通じ、価値を作る。

①スプリントプランニング

プロダクトバックログの中から、今回のスプリントに取り入れる項目を選定する。

②デイリースクラム

朝会と呼ばれる毎日決まった時間にミーティングを行う。

③スプリントレビュー

スプリントの最後に成果物をレビューし、フィードバックをもらう。

④スプリントレトロスペクティブ

スプリントレビュー後に、スプリントの振り返りを実施し、次のスプリントに向けた改善を図る。振り返りのツールとして、KPTA、YWT、バーンダウンチャートを用いる。

◎有効性

アジャイル開発では、最小限の機能から作っていき、改善をし続けることで、提供価値を増加させ、価値提供までの時間を短縮できるメリットがある。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、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。ほかにも、アジャイルがうまくいく条件として、チームメンバーが自律的に動ける環境は雰囲気的重要であり、スクラムマスターやプロダクトオーナーとの密に連携することでプロジェクトがうまくいく。アジャイル開発が他の開発手法に比べて必ずしも優れているわけではなく、プロジェクトの性質に合わせて、最適な開発手法を選ぶ必要がある。

●第6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:

アーキテクチャとはシステムの基盤でありシステムの品質を決定する重要なものである。そのため、アーキテクチャが頑強でないとシステムテストや運用後に大問題が発生する可能性がある。そこで、本演習ではアーキテクチャを分析・評価する方法として「品質特性シナリオ」、「ADD」、「ATAM」について学んだ。

(1)品質特性シナリオ

「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」などといった品質特性をシナリオ化することで品質の要件を明確にする方法。品質特性シナリオを作成することでアーキテクチャの設計や評価が行える。なお品質特性シナリオの構成要素は「刺激の発生源」、「環境」「成果物、アーキテクチャ要素」「応答」「応答測定」である。

(2)ADD(Attribute Driven Design)

アーキテクチャが目標とする機能要件と品質要件を明確化できる設計手法のこと。品質特性シナリオの成果物が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、モジュール分割をおこなうことで実現、詳細化する。ADDのメリットとしてアーキテクチャがなぜこのようなアーキテクチャになったのかの説明が容易にできることがあげられる。

(3)ATAM(Architecture Trade-off Analysis Method)

ATAMは品質特性に関する要求をアーキテクチャが満たしているかを評価する方法である。ATAMを適用することで「アーキテクチャの簡潔なプレゼンテーション」、「シナリオの集まりで表現された品質要求」、「アーキテクチャ上の決定と品質要求の対応関係」「重要ポイントとトレードオフポイント」「リスクと非リスク」「主題となるリスク」を得ることができる。

■有効性:

- アーキテクチャはシステムの品質を決定、実現する土台であるため、品質特性シナリオや早い段階での分析・評価を行うことが重要である。
- アーキテクチャを適切に設計することでシステムの理解やシステムの品質を実現することが可能となる

■留意点:

- 非機能要件を十分に抽出できない可能性があるため、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方がよい。
- 複数の観点から非機能要件を抽出するために、異なるステークホルダと品質特性シナリオを作成するとよい。
- ATAMによる分析評価は実施項目が多く実施困難な場合が多い。
- プロセス遵守に重点をおくため、成果物(ドキュメントやソースコードなど)が品質を満たしているかの踏み込みが甘くならないように注意。

●第7回（臨時）：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ユーザ要求を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るが、最適な手法を選択して適用することが大切である。

本演習では、「リカレント（リスクリング）教育」をテーマに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各種手法の演習を行った。

① ペルソナ法

システムを利用する架空の人物像（ペルソナ）を設定し、その人物の利用シーンを想像することで要求抽出を行う。人物像の設定をより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 (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)

④ 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）を詳細化、いくつかのサブゴールに分解し、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）の関係を可視化する。

■有効性：

要求工学の様々なモデル化手法を適用することで、要求の根拠や背景を知ることができ、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、実際に手法を用い参加者で議論し多くの視点が挙げられることが実感できた。

■留意点：

システム開発で解決しようとしている問題を理解し、手法を適用するためには、明らかにしたい対象を定め、それを知るための手法を選択し、適用することが重要である。

●第8回(例会)：工数見積モデルの構築手法～勘を見える化する手法～：
石谷靖氏（株式会社三菱総合研究所）

■概要：

ソフトウェア開発の工数見積もりの際に問題となりるのが、開発中にどれだけ工数が変わる可能性があるかということである。この問題を解決するのが工数見積りモデルの一つである CoBRA (Cost estimation, Benchmarking, and Risk Assessment) 法である。この講義では「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：

- 揺らぎがあるものに対して、変動要因を考えて精度を向上できる。
- 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。
- 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。
- 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。
- 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。
- 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。

■留意点：

- 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。
- 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。

●第9回（例会）：ソフトウェアテスト：鈴木 三紀夫 氏（ASTER）

■概要：

本演習は演習問題を通じて、ソフトウェアテスト技法の基礎を学ぶことを目的としている。各テスト設計の演習課題に対し「個人演習」>「グループ内議論」>「講師による回答例の解説」という流れで進めた。

個人演習およびグループ内議論でテスト設計を検討した後でテスト設計技法を用いた解法に触れることで、より深くテスト設計技法を理解することが出来た。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おけるテスト技法である。

プログラム制御を制御フローグラフで表現し、グラフを網羅する基準(カバレッジ)に従ってテストケースを設計する方法である。

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） … 命令文を少なくとも1回実行する

●C1（分岐網羅） … 分岐条件を少なくとも1回実行する

(2)同値分割法

ブラックボックステストにおけるテスト技法である。

仕様からデータのある観点でグループ（同値パーティション）に分類し、そのグループの代表的な値を用いてテストを行う技法である。

(3)境界値分析

ブラックボックステストにおけるテスト技法である。

同値分割法の拡張で、仕様条件の境界となる値とその隣接値に対してテストを行う技法であり、同値パーティションの最小値および最大値を選択してテストを行う。

(4)デシジョンテーブルテスト

ブラックボックステストにおけるテスト技法である。

複雑な判定ロジックの整理に有用な決定表（デシジョンテーブル）の考え方「規則／ルール」に基づき、入力条件の組み合わせと対応する出力結果を整理することでテストケースを設計する技法である。

■有効性

- ・実施するテストの目的に応じたテスト技法を用いることでコストパフォーマンスが高いテストを実施することができる。
- ・テスト設計技法の活用で合理的、抜け漏れなくテストを設計することが可能である。
- ・「制御フローテスト」におけるカバレッジの網羅率や「境界値分析」で扱う値を定義することで、品質のバラつきを排除する効果も期待できる。

■留意点

- ・テスト設計には正解がなく、テストの目的に応じた設計を行うことで最適なテスト技法を選択する必要がある。
- ・様々なテスト技法の特徴、短所、長所を理解し、テスト技法を都度柔軟に最善の技法を使い分ける必要がある。

以上