

## 演習コース「ソフトウェア工学の基礎」2024年度活動報告

### Report on Practice Course of Software Engineering Foundations in 2024

研究員：村田 優 (株式会社デンソー), 松野 航大 (株式会社アドバンテスト),  
國分 晴利 (エヌ・ティ・ティ・コミュニケーションズ株式会社),  
鈴木 優輝 (東芝システムテクノロジー株式会社), 堀江 眞太 (日本電気株式会社),  
前田 良浩 (株式会社日立ソリューションズ・クリエイト),  
中田 賢治 (プライシス株式会社), 櫻井 貴寛 (三菱電機ソフトウェア株式会社),  
石田 貴世子 (三菱電機ソフトウェア株式会社), 藤原 みさき (アズビル株式会社)  
主査：猪塚 修 (横河ソリューションサービス株式会社)  
副主査：鷺崎 弘宜 (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)エクスマーション)  
長谷川 裕一 (合同会社 Starlight&Storm)

#### 研究概要

演習コース「ソフトウェア工学の基礎」を設置し、演習と議論を通じて実践的および先進的な種々の代表的ソフトウェア工学の考え方や技術を学習した。コースとしては 2005 年度から継続的に設置して 20 年目となる。本稿では、コースの設置背景と狙い、各回における演習の概要、議論や振り返りを通じた実務におけるソフトウェア工学技術適用に関する問題認識、参加した各研究員における活用実践状況について報告する。

**Abstract** Following the success of previous courses in 2005-2024, the practice course of software engineering foundations has been opened in this year. This article reports on the primary aims of this course, summaries of each practice in regular meetings, problem recognition and preliminary application experiments on software engineering techniques learned in the course.

#### 1. コースの狙い

扱う対象がしばしば抽象的で、自由度が高く極めて難しいソフトウェア開発という行為の成功には、理論や経験に裏打ちされたソフトウェア工学技術が欠かせない。しかし、その適用が場当たりのではかえって複雑さを増すばかりである。そこで、体験や実践を通じて使いどころや留意点を含めて「深く」習得した技術群を体系的に使いこなすことが重要であるが、(特に我が国の) ソフトウェアの多くは、きちんとソフトウェアエンジニアリング (ソフトウェア工学) を学んでおらず、また企業でも十分な体系的教育を受けていない技術者によって作り続けられている [1] と指摘されている。

ソフトウェア工学 (Software Engineering) とは、ソフトウェアを開発する際に駆使すべき技術 [2] であり、ソフトウェアの開発、運用、および保守に対する系統的で規律に基づいた定量的アプローチ [3] と捉えることができる。ソフトウェア工学の習得と適切な利用により、属人性を排した一定以上の品質保証と高生産の達成が期待でき、上述の品質問題の解決を期待できる。具体的には、Software Engineering Body of Knowledge (SWEBOK, ソフトウェアエンジニアリング基礎知識体系) [3] などの参照による体系的なソフトウェア工学知識の整理と学習に加えて、実践あるいは実践に近い体験を通じたソフトウェア工学技術の習得が必要である。

このような問題意識から本コースは、主に演習と議論を通じてソフトウェア工学技術群を

習得する場として 2005 年度より継続して設置され、ソフトウェア工学技術の会得に有効であったとの評価を得ている（[4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22]を参照されたい）。そこで 2024 年度も引き続いて、産学両面に通じた講師をお招きし、計 10 名の研究員が参加して、全 9 回にわたり代表的なソフトウェア工学技術に関する講義と演習を実施した。

本稿では以降において、本コースの構成、および、各回における講義・演習の概要、および、議論や振り返りを通じた実務におけるソフトウェア工学技術適用に関する問題認識について報告する。なお、以下の報告は、主に各研究員の分担執筆による。

## 2. コースの設計と自己評価および工夫

本コースは、設置にあたり以下の3点を目的とした。

- ・演習を通じた主要なソフトウェア工学技法の体系的かつ深い習得
- ・個人・組織の開発力強化のための基盤形成
- ・仲間作り（データ収集、技法発展）

その着実な達成のため、本コースでは以下の取り組みを実施した。

### (1) 知識体系における位置づけの提示と徹底的な演習

コースの全体構成の設計にあたり、ソフトウェア工学知識体系 SWEBOK およびソフトウェア品質知識体系 SQuBOK 上で、2024 年度に取り上げた各技術の位置づけを識別し、マネジメントを除くエンジニアリング系として主要な知識領域を概ね網羅できていることを確認した（図 1）。

品質の基本 概念	組織レベルの品質 マネジメント	プロジェクトレ ベル(共通)の品質 マネジメント	プロジェクトレ ベル(個別)の品質 マネジメント	品質技術
品質の概念 <i>品質・レビュー</i>	マネジメントシ ステムの構築と運用	意思決定のマ ネジメント		メトリクス <i>メトリクスとGQM</i>
品質のマ ネジメント	ライフサイクル プロセスのマ ネジメント <i>アジャイル開発</i>	調達マ ネジメント	品質計画のマ ネジメント	品質計画
	プロセスア セスメント のマ ネジメント	構成管理	<i>要求工学・オブジェクト指向・ペ ーパープロトタイピング</i> レビュー、アーキテクチャ	要求分析
	検査のマ ネジメント	リスクマ ネジメン ト	レビューのマ ネ ジメン ト	レビュー
	監査のマ ネジメント	プロ ジェクトマ ネジメン ト全般 <i>見積り</i>	テストのマ ネジ メン ト	テスト <i>テスト</i>
	教育のマ ネジメント		品質評価のマ ネ ジメン ト	品質分析・ 評価
	法的権利・責任 のマ ネジメント	<i>メトリクスとGQM</i>	運用・保守のマ ネ ジメン ト	運用・保守

図 1：SQuBOK エリアと演習のマッピング

そのうえで、演習の各回ができるだけ開発プロセスの流れにそって上流系技術から下流系技術と順に並ぶように全体を設計し、各回の「点」と「点」を結び付けて「線」を成し、体系的な学習を促すように配慮した。以上のコースの設計および徹底的に手を動かす演習ベースの講義構成により、本コースはソフトウェア工学技術の体系だった深い習得に有効であった。

本年度の特筆事項として、すべての演習をオンラインで実施した。初対面の時から一度も実際に顔を合わせないで行ったが、TOOLの活用で例年同様の学びを得ることができた。

### 3.各演習における気づきと活用状況

本コースでは、ソフトウェア工学技術の特にソフトウェア開発技術およびマネジメント・プロセス・品質技術に関する以下の演習について、それぞれ個別に講師（敬称略）をお招きして実施した。さらに全演習の終了後、各受講者が本コースを通じて得られた「気づき」をそれぞれに報告し、整理してまとめた。具体的には、実務におけるソフトウェア工学技術の活用という観点から気がついた有効性や留意点、さらには各自の所属先や個人における実践・活用状況を各研究員がそれぞれに考察した。本コースに限らず学習行為一般について、その最終目的は学習した事柄によって自身およびその周囲について何らかの変化をもたらすことにあり、「気づき」を整理検討することは重要である。計9回の演習について、それぞれ整理した結果を付録に記載する。

付録における活用事例とは、本コースのある参加者が実際に、習得した各技術を自身や所属組織等において活用した結果を報告している。2024年度において既に多数の技術について活用が始められており、前述のように実践を通じて開発強化のための基盤形成について一定の達成をみた。また特にコースの後半にて取り上げた技法については、主に時間的な問題から2024年度中の活用には至らなかったため今後の活用が期待される。

#### ●第1回（例会）2024/5/10：

デザイン思考&ペーパープロトタイピングと周辺：

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ョン）

#### ●第2回（例会）2024/6/21：

オブジェクト指向分析設計：

井上 樹氏（株式会社豆蔵）

#### ●第3回（例会）2024/7/25～26：

アーキテクチャ設計・評価：

長谷川 裕一氏（合同会社Starlight & Storm）

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）

#### ●第4回（例会）2024/10/18：

アジャイル開発の基礎知識：

天野 勝（株式会社永和システムマネジメント）

#### ●第5回（例会）2024/11/8：

工数見積モデルの構築手法（CoBRA法）：

石谷 靖氏（株式会社三菱総合研究所）

#### ●第6回（例会）2024/12/13：

メトリクス：

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ョン）

#### ●第7回（例会）2025/1/17：

ソフトウェアテスト：

湯本 剛氏（株式会社 ytte Lab/ フリー株式会社）

#### ●第8回（臨時会）2025/2/7：

要求工学（要求分析）：

斎藤 忍氏（日本電信電話株式会社）

#### 4.おわりに

本コースでは、指導講師による9回の講義・演習を通じて、ソフトウェア開発プロセスの上流から下流までの主要な工学的技術を深く会得した。研究員各位には、本コースを通じて習得した技術や「気づき」を活用し、自身や組織への適用を通じたソフトウェア工学の実践に積極的に取り組まれることを願う。

次年度も、演習内容を改善した上で本コースを実施する。研究員各位には、次年度も本コースに参加して議論を深める、あるいは、他の分科会にて習得技術を適用・発展させるなど、自身や周囲、社会、さらには日科技連へのフィードバックにご貢献いただければ幸いである。また本稿が、この演習コースに対する興味に結びつき、次年度以降の演習コースへの新たな参加につながれば幸いである。その延長線上として、日本のソフトウェア産業の発展に少しでも貢献できれば、著者として望外の喜びである。

謝辞 本稿の執筆にあたって、研究員の方々に草案を分担執筆いただきました。ここに厚く御礼申し上げます。また、毎回の演習をご指導いただいた講師の皆様に、この場を借りて厚く御礼申し上げます。

#### 5.参考文献

- [1]阿草清滋，西康晴，沢田篤史，鷺崎弘宜，〈特集〉情報専門学科カリキュラム標準J07：ソフトウェアエンジニアリング領域（J07-SE），Vol. 49，No. 7，pp. 25-31，2008.
- [2]Pressman, R. S. : Software Engineering -A Practitioner's Approach, McGraw-Hill, 2005. (邦訳) 西康晴，榊原彰，内藤裕史訳，実践ソフトウェアエンジニアリング，日科技連出版社，2005.
- [3]ISO/IEC/JTC1/SC7: ISO/IEC TR 19759:2005, Software Engineering -Guide to the Software Engineering Body of Knowledge (SWEBOK), ANSI, 2007. (最新版は<http://www.swebok.org/>より取得可能) (邦訳) 松本吉弘監訳，ソフトウェアエンジニアリング基礎知識体系—SWEBOK 2004—，オーム社，2005.
- [4]野中誠，ソフトウェア工学演習コース活動報告，日本科学技術連盟第21年度ソフトウェア品質管理研究会成果報告集，2006.
- [5]鷺崎弘宜，猪塚修，田村一賢，濱正知美，麓博之，ソフトウェア工学演習コース2006年度活動報告，日本科学技術連盟第22年度ソフトウェア品質管理研究会成果報告集，2007.
- [6]鷺崎弘宜，田村一賢，阿部修久，安藤元伸，古村仁志，保栖真輝，溝口文康，山本文彦，猪塚修，ソフトウェア工学演習コース2007年度活動報告，日本科学技術連盟第23年度ソフトウェア品質管理研究会成果報告集，2008.
- [7]鷺崎弘宜，城間祐輝，田村一賢，溝口文康，大橋剛和，覚井真吾，白井孝明，草場康男，松宮宏明，安藤良治，佐藤和人，柴田和也，實藤博，ソフトウェア工学演習コース2008年度活動報告，日本科学技術連盟第24年度ソフトウェア品質管理研究会成果報告集，2009.
- [8]鷺崎弘宜，田村一賢，野中誠，加藤岡弘一，上村秀一，高田祐布子，中島碧莉，古木健，森崎一邦，横内和城，吉川真吾，村上真一，演習コース「ソフトウェア工学の基礎」2009年度活動報告，日本科学技術連盟第25年度ソフトウェア品質管理研究会成果報告集，2010.
- [9]鷺崎弘宜，猪塚修，野中誠，小倉徹，鈴木尚，片山拡充，古谷伸一，中田陽大，升谷雄二，吉田麻紀，本田繁，長嶋聖，塩浜龍志，下條清史，演習コース「ソフトウェア工学の基礎」2010年度活動報告，日本科学技術連盟第26年度ソフトウェア品質管理研究会成果報告集，2011.
- [10]鷺崎弘宜，猪塚修，浜田浩史，奥井健，千代出，阿部悦子，清水里美，南齋雄一，高橋大輔，坂静香，道脇直紀，山崎春奈，大橋昭，演習コース「ソフトウェア工学の基礎」2011年度活動報告，日本科学技術連盟第27年度ソフトウェア品質管理研究会成果報告集，2012.
- [11]浜田浩史，鷺崎弘宜，猪塚修，朝井与志哉，加藤尚樹，楠森賢佑，久原健一，駒井利之，鈴木勝統，鈴木達郎，田中孝一，東久保理江子，永瀬孝紀，森俊樹，演習コース「ソフトウェア工学の基礎」

- 2012年度活動報告，日本科学技術連盟第28年度ソフトウェア品質管理研究会成果報告集，2013.
- [12]浜田浩史，鷺崎弘宜，猪塚修，小間香保里，杉山浩一，染原一仁，佐々木愛美，中村考宏，森哲史，齊藤慶太郎，新田佳祐，安部晃嘉，演習コース「ソフトウェア工学の基礎」2013年度活動報告，日本科学技術連盟第29年度ソフトウェア品質管理研究会成果報告集，2014.
- [13]鷺崎弘宜，猪塚修，藤原聡子，村上淳，中村壮志，市川勝規，染谷知宏，岩村義明，山本真成，仲野恭平，演習コース「ソフトウェア工学の基礎」2014年度活動報告，日本科学技術連盟第30年度ソフトウェア品質管理研究会成果報告集，2015.
- [14]鷺崎弘宜，猪塚修，中田絢子，齋藤綾乃，村瀬邦彦，吉井誠，高木聡，飯田勲，油谷啓之，根岸翔，演習コース「ソフトウェア工学の基礎」2015年度活動報告，日本科学技術連盟第31年度ソフトウェア品質管理研究会成果報告集，2016.
- [15]鷺崎弘宜，猪塚修，内藤拓，田口浩嗣，加地孝敏，大森淳夫，辻本亜樹，飯塚宏明，原好宏，市川大輔，金成祐介，大澤裕太，加藤蔵次，太田延樹，中村和哉，演習コース「ソフトウェア工学の基礎」2016年度活動報告，日本科学技術連盟第32年度ソフトウェア品質管理研究会成果報告集，2017.
- [16]鷺崎弘宜，猪塚修，衛門一樹，黒田昌憲，原田英之，岩本真司，向井祐人，西村智樹，奥山亜耶子，佐藤弘幸，岩崎洋，河嶋浩子，西田尚弘，村田龍一，片山伸輔，羽多野顕，演習コース「ソフトウェア工学の基礎」2017年度活動報告，日本科学技術連盟第33年度ソフトウェア品質管理研究会成果報告集，2018.
- [17]日比秀二，荒川健太郎，加藤鋳也，岡本沙紀，吉田拓矢，樋口雄基，西村高志，関谷絵里，岩崎真士，川俣陽輝，高田紀行，内藤次郎，藤井俊久，鷺崎弘宜，猪塚修，演習コース「ソフトウェア工学の基礎」2018年度活動報告，日本科学技術連盟第34年度ソフトウェア品質管理研究会成果報告集，2019
- [18]壁谷考洋，増田知彰，藤崎智宏，前川大智，岡野真也，井置一哉，滝川翔，草薨明彦，片桐汐駿，渡辺貴之，岡内裕希，米陀政人，岩根正典，嶋幸生，田原正崇，島岡良規，吉野正崇，小田健人，邱章傑，増田充，房野亮一，宮村充弘，庄野慎2019年度活動報告，日本科学技術連盟第35年度ソフトウェア品質管理研究会成果報告集，2020
- [19]二瓶秀信，和田久，辻鉄也，谷口文隆，高松毅，芳沢圭一，金丸周平，西畑翔，村上拓也，甲斐流，2020年度活動報告，日本科学技術連盟第36年度ソフトウェア品質管理研究会成果報告集，2021
- [20]杜馨瓏，諏訪航司，牛久裕輔，村瀬明日香，浜田泰之，清水梨帆，重村恵美，会沢優花，漆原康行，庭野雄平，藤井広宣，白浜陽一，村田秀和，重本章吾，後藤香織，2021年度活動報告，日本科学技術連盟第37年度ソフトウェア品質管理研究会成果報告集，2022
- 相原啓人（日本電子株式会社）
- [21]森有紀乃，萩原美樹，三原一真，稲垣英幸，谷川壮洋，堤弘法，梶原裕，鈴木理昂，松本英樹，川勝光季，笹本実穂 2022年度活動報告，日本科学技術連盟第38年度ソフトウェア品質管理研究会成果報告集，2023
- [22]佐々木瑛太，戸田優作，鬼武祐介，大塚駿，芦田啓慈，炭矢瑠奈，關樹，川名大地，松田考将，村山光一郎，後藤昌巳 2023年度活動報告，日本科学技術連盟第39年度ソフトウェア品質管理研究会成果報告集，2024

以上

●第1回（例会）：デザイン思考&ペーパープロトタイピングと周辺：  
鷺崎 弘宜氏（早稲田大学グローバルソフト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 / 株式会社エスクモーション）

## ■概要

デザイン思考とは、問題の本質を見極め、多くのアイデアを出して解決に導くプロセス&早くたくさん失敗して経験から学ぶ考え方である。原理としては、デザイナーの思考や感性、方法を用いて技術的に利用可能なもの、および実行可能なビジネス戦略により顧客価値や市場機会へ変換され得るものと人々の要求を一致させることを指す。

本演習では、デザイン思考のプロセスを理解し、代表的な技法を使って、ユーザビリティテストを体験した。内容は以下の通り。

## ●デザイン思考のプロセス

### 1. 発想

課題の設計を工夫し、ものごとの新しい見方を探す。

- 取り組み：よく聞く,よく見る,予想外を受け入れる
- 代表的な技法・・・ペルソナ法,構造化シナリオ法

### 2. 創造

アイデアを生み出し、実行可能なものにする。

- 取り組み：改善策を考えるため多くアイデアを出す
- 代表的な技法・・・プロトタイプ

### 3. 改良

被験者からのフィードバックをもとに実験を繰り返し継続する。

- 取り組み：利用者の意見を聞き,気づきをリスト化して改善。

## ●技法概要

- ペルソナ法：実在するユーザの代わりに使われる厳密に定義されたユーザ像。
- 構造化シナリオ法：  
以下3つのステップで、ユーザの目標や活動を検討する。
  - サービスシナリオ：  
ユーザー一般にとっての価値とビジネス的価値を記述する。
  - アクティビティシナリオ：  
ペルソナにおける活動全体のフローや情動を記述する。
  - インタラクションシナリオ：  
ユーザが目標に向かうシステムや実装に依存した具体的活動を記述する。
- プロトタイプ（プロトタイピング）：  
実物に近いモデル・模型を作成して仕様を検討する。
- ペーパープロトタイピング：  
ユーザが行いたいことを,ストーリーボードとしてワイヤーフレームをシナリオに沿って並べ,ユーザビリティテストを行う。テストの被験者は,ペルソナになりきって,思いついたことを発話する。作成者は結果を記録して,改善のヒントとする。
- システム/デバイスによるプロトタイピング：  
簡易なモック,Raspberry Pi の利用など

## ■有効性

- ペルソナ：

プロジェクトの関係者全員がユーザ像を共通認識でき、目的が明示的になるため、画面設計などの開発を遂行しやすくなる。

- 構造化シナリオ：  
ユーザ視点のイメージが深まり、問題の本質を分析できる。
- プロトタイピング：  
コーディング不要であり、容易にテストを実現できる。更に繰り返し実施することで、UIの完成度を高められる。また、ペーパープロトタイピングではツールや環境に依存せず、修正も容易であるため、試行回数を増やすことができる。

#### ■留意点

- ペルソナ：  
ユーザ像は具体性が重要であり、特にバックボーンや現実性に留意すること。想像が困難な場合は実製品の利用者ログなどを活用すること。
- 構造化シナリオ：  
ペルソナ体験に沿ったユーザ視点で作成すること。開発者視点で作成すると、実装方法によらないユーザの本質・欲求を識別できない。ユーザ視点でのシナリオ作成が困難な場合はいくつかのシナリオを通して、段階的にユーザの要求に近づけていくアプローチを検討すること。
- プロトタイピング：  
開発早期に問題発見することが目的のため、プロトタイプの内容に拘るあまり、工数をかけ過ぎることは避けること。また、被験者にかかる工数も考慮して、再テストの頻度を決定すること。

●第2回：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について、演習を通して学んだ。

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である。これにより、システム完成イメージの早期共有、要求の正しい理解、不備不足の発見といったメリットが得られる。演習では以下の手法により要求のモデリングを行った。

- ・シナリオ分析：利用者の視点からシステムの利用の流れ、相互作用を想定して要求分析をする手法である。ユースケース図、ユースケース記述、シナリオリスト、シナリオ詳細記述と具体的にユースケースを書き進めることによって問題点などを洗い出すことができる。
- ・モデルベース分析：UML（統一モデリング言語）を使い、要求をモデル化する手法である。静的構造、動的構造、機能的構造の3つの側面から対象を捉えることが基本的な考え方であり、定義されたモデルを用いて自然言語を書き直すことで抜け、もれに気付くことができる。

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を見える化することである。これにより、ソースコードから読み取れる範囲よりも広く俯瞰でき、ソースコードを書く前に問題を発見できるといったメリットが得られる。演習では、設計モデリングツールとして、クラス図／ステートマシン図／アクティビティ図／シーケンス図の設計モデルのレビューによりソフトウェア構造の問題点の抽出を行った。

■ 有効性：

- ・ソフトウェアエラーは、後工程になるほど修正コストが増大する。要求分析段階で要求をモデリングし、早期に要求の不備不足、認識違いを解消することは、修正コストの削減に効果的である。
- ・異常系や暗黙の要求は、要求として明示されないことが多い。シナリオ分析により、ユースケースを詳細化していく過程で、代替フロー、例外フロー、制約等を明確にすることで、要求の網羅性を高めることができる。
- ・プログラマの扱うコード量は10年前と比較し、100倍以上に増えているとの調査結果が、第一回特別講義で示された。ソフトウェアの構造をコードから把握するのは限界がある。クラス図、ステートマシン図等で構造を見える化することは、ソフトウェアの早期理解につながるほか、改修・修正時の影響把握や、内部一貫性、全体最適に寄与する。

■ 留意点：

- ・要求は4つのステージからなる。利害関係者が実現してほしいと思っていることを示す「要望」、要望のうち、システムで実現することを定めた「要求」、要求を満たすためのシステムのふるまい・特性を定めた「要件」、要件が網羅された要件の集合である「仕様」である。それぞれを区別することが重要である。
- ・設計のモデリングにおいて、詳細に書かれたUMLモデルはコードに変換可能であるが、相応の情報量を持つモデルを正確に書く必要がある。



● 第3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（Soft/Hard）間の関連、思想であり、骨格、基盤、土台、権力などと喩えられる。システム全体の構造や振る舞いを定義するものであり、その決定がシステムの品質・開発効率、保守性などに大きな影響を与えるため、適切な設計を行うことが重要である。

（1）品質特性シナリオ

品質特性がどのようにシステムに影響を与えるかを具体的に示すシナリオのこと。品質特性には「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」があり、品質特性シナリオの構成要素として「刺激の発生源」「刺激」「環境」「成果物」「応答」「応答測定」からなる。

（2）ADD（Attribute Driven Design）

品質特性を考慮しアーキテクチャを決定する手法。品質特性シナリオの応答測定を満たすように実現手法やアーキテクチャパターンを選択し、それを実現するために必要があればモジュール分割を行い詳細化する。

（3）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。システムのアーキテクチャに関するトレードオフ（妥協点）を分析し、最適な品質特性を満たすアーキテクチャの決定する手法。

有効性

- ・品質特性シナリオは、品質特性を具体的なシナリオとして定義することでどの品質特性を重視すべきかを明確にできる。
- ・ADDは設計の初期段階で品質特性を定義するため、非機能要件の達成が確実に becoming いく。
- ・ATAMは品質特性のトレードオフを評価し、最適なアーキテクチャを決定できる。

留意点

- ・これらの手法は、設計の初期段階で品質特性を明確化し、アーキテクチャ選択を評価するために有効であるが、適用するタイミングが遅くなりすぎないようにする必要がある。
- ・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。

## ●第3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

- 「レビュー」の定義は組織によって違うが、本コースでは「ソフトウェアテストと同様に欠陥を見つける手段」という解釈に基づき講義や演習が行われた。
- 演習では、メールシステムを題材にレビューを体験することにより、レビュー時の視点の重要性を理解することが出来た。また、グループ討議により複数人によるレビューの有効性（多様な視点の確保）を理解すると共に、他の人の考えや意見を聴くことにより、自身とは異なる視点やアプローチに気付くことができた。
- また、演習での経験や考えをベースにその後の講義を受けることで、行間を補うことができ理解を深めることができた。

### ■有効性

- レビューは静的テストに分類される活動であり、効果的に行うことで設計段階からシステムの品質を向上させることが可能である。
- 設計は唯一絶対ではないため、レビューにより専門家などの意見を取り入れることでより良い設計にすることが可能である。
- テストでは発見が不可能（or 困難）な問題を見つけることが可能である。
- 経験の浅いメンバーをレビューに同席させることにより、設計のポイントを学ぶことが可能である。

### ■留意点

- チェックシートや品質特性などの観点を予め準備しておくことで、より効果的なレビューを実施できる。
- 「レビュー」には様々な手法があるため、各手法の特徴を理解し、適宜使い分けることが重要である。以下に代表的な手法を記載する。
  - ウォークスルー
  - インスペクション
  - 回覧（パスアラウンド）
- レビューを効果的に実施するためのポイントを以下に記載する。
  - レビューの目的を明確にする。
  - システム要求を明確にする。
  - 重点を絞る。
  - 視点を変える。（例えば、開発者／発注者／ユーザ／保守／運営担当者のそれぞれの視点で見ると要求が変わってくる。）

## ●第4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

アジャイル開発とは、機能ごとに優先順位を定めて開発の単位を小さくし、高頻度でリリースする開発手法である。主なメリットとしては、早い段階から価値を提供できることや、要件の変更にも柔軟に対応できることが挙げられる。今回は標準的なアジャイル開発の進め方についてスクラムのフレームワークを用いて演習形式にて理解を深めた。

演習では以下のルールに従って、各チームにてメンバーを3つの役割に分けて開発作業を実施した。

#### 【ルール】

チームにて折り紙を用いて多面体を作成することにより、完成した多面体やその数に応じた得点が得られる。他チームよりも多くの得点の獲得を目指す。

#### 【役割】

- ・ プロダクトオーナー（1名）：サービスの責任者。プロダクトバックログを管理して方針を決定する。
- ・ スクラムマスター（1名）：スクラム管理の責任者。プロダクトオーナーの方針を開発に適用する。
- ・ 開発者（3名）：成果物を作成する。

#### 【開発作業】

以下の流れを1スプリントとし、これを2回実施した。スプリントではスプリントプランニングを1回、朝会および開発を3回、最後に朝会、スプリントレビュー/リリースおよび振り返り会を1回行った。

- ① スプリントプランニング：プロダクトオーナーがスプリントの目標や戦略を決定する。
- ② デイリースクラム（朝会）：スクラムマスターを中心に作業の役割分担、開発の進め方を議論し、決定する。
- ③ 開発：開発者およびスクラムマスターが成果物を作成し、スプリントバックログに反映させる。
- ④ スプリントレビュー/リリース：成果物をレビューし、リリースする。
- ⑤ レトロスペクティブ（振り返り会）：KPTAを用いてスプリントの振り返りを行い、次のスプリントに向けた改善点を議論する。

### ■ 有用性

- ・ アジャイル開発ではスプリントごとに目標を定めて開発および改善を繰り返すため、ウォーターホール開発と比較して短い時間で価値を最大化できる。
- ・ 朝会や振り返り会など、チームメンバー同士のコミュニケーションが行える場が多く設定されているため、開発目標に関する認識を適宜合わせられる。
- ・ 朝会の際にバーンダウンチャート、スプリントバックログなどでチームメンバーの進捗状況の詳細を把握できるため、そこで理解した得手不得手を開発の作業に反映できる。

### ■ 留意点

- ・ アジャイル開発においてリーダーは、メンバーを支援するサーバント型とメンバーに細かく指示を出す指示命令型のリーダーシップを状況に応じて使い分ける必要がある。
- ・ スクラムマスターはプロダクトオーナーから伝えられた方針を元に開発者に指示を出すため、すれ違いが起こらないようにそれぞれの立場のメンバーとしっかりと認識を合わせる必要がある。

●第5回（例会）：工数見積りモデルの構築手法：石谷 靖（株式会社三菱総合研究所）

■概要

ソフトウェア開発における見積りでは金額(工数)の妥当性が分からない、見積りの妥当性を説明できないなどの見積り根拠が見えないことによる問題が発生しがちであり、見積り根拠の「見える化」= コスト構造の「見える化」が重要であると言える。

その見える化のために工数見積りモデルの構築手法(CoBRA 法)を学び、演習(グループワーク形式)で「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」を実施した。

ここで、CoBRA 法の見積式を下記に示す。

$$\text{工数（コスト）} = \alpha \times \text{規模} \times (1 + \sum C0i)$$

$\alpha$ は「生産性」を表しており、実績データに照らして、変動要因とその定量化を検証し、算出するものである。C0は「コスト変動要因のオーバーヘッド」を表しており、経験豊富なPL等の熟練者の知見を基に変動要因とその影響を定量化したものである。

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、見積りモデルが完成する。

■有効性：

- 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することで工数の説明力が向上する。
- 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。
- 影響度の高い要因の把握による、コストマネジメント力の向上。
- 工数変動量から予算超過確率を把握することができる。
- 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。
- 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：

- 変動要因については、最大でも10個程度が望ましい。  
初期段階で影響の大きい7個程度に変動要因の数を絞込み、初期モデルを構築し、精度をチェックし必要であれば変動要因を追加していくのが理想的である。
- 最初から適切なモデルを構築することが難しいため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必要がある。
- モデルの構築に3名程度の熟練者が必要となり、かつ10件程度のプロジェクト実績データ(規模、工数)が必要となる。
- 見積り誤差が大きいプロジェクトに特殊性(規模や開発種別の違い等)があった場合、モデルを分けることやモデルから除外を検討する必要がある。

●第6回(例会)：メトリクスによるソフトウェア品質把握：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)SI&C / (株)エクスマーシオン)

■概要：

メトリクスはソフトウェアの品質把握，及び改善において重要である。  
メトリクスの利用には目的が必要であり，目的設定にGQMの考え方が有効であり，また，GQMはGQM+Strategiesとして拡張することにより，企業活動にも利用可能である。  
講義内容は以下となり，グループ演習を交えながら理解を深めた。

(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を表わす。メトリクスは測定の方法と尺度を意味し，定量的であることが重要である。

主なメトリクスは以下となる。

- ・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)ゴール施行の測定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，以下で構成される。

- ・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

目標を具体化するには，横に広げて多面的に見ることが必要であり，仮定と解釈によってより精密な目標を設定する。

(3)組織目標，戦略，測定の整合化GQM+Strategies

GQMを用いて，目標に対するメトリクスを取得，評価しても組織目標につながっていないことがある。

そのような場合の手法として，GQM+Strategiesがある。

GQM+Strategiesは上位の組織目標と戦略に対してメトリクスを設定し，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，開発の管理，品質把握と改善に用いることができる。また，現状の把握だけでなく，未来の予測や計画に用いることができる
- ・GQM+Strategiesグリッドを用いて「目標連鎖」の見える化をすることで，部門を超えた企業/事業の方向性を共有することができる

■留意点：

- ・メトリクスは闇雲に収集すれば良いわけではなく，目的を明確にすることが重要である
- ・メトリクスには限界がある。メトリクスの限界を知り意思決定に役立てる

●第7回(例会)：テストを作る技術(テスト技法演習)：湯本 剛氏(株式会社 ytte Lab)

■概要

本演習の目的は、演習を通してソフトウェアテスト技法の基礎を学ぶことである。テスト技法を知らない人の場合は使い方のコツ、知っている人の場合は他者への指導方法の習得することが出来た。

● テストを作る技術

テスト作成のためには、テストスキル以外にソフト/IT に関するスキルやドメインの知識についても必要となる。

テスト分析の内容からテスト設計を実施する際にテスト技法を活用する。

● テスト技法

➤ 同値分割法

入力値の全体集合をある条件に沿って区切る技法。

➤ 境界値分析

同値クラスの代表値として境界値を使う技法。

➤ デシジョンテーブルテスト

入力条件の組み合わせによって出力が変化することを確認するための技法。

➤ 状態遷移テスト

処理順序の組み合わせを確認するための技法。

● テスト分析と設計で行うモデル化

テスト技法を適用するためには適用可能なサイズまでシステムを分解する必要がある、そのためのモデル化方法を学習した。

■有効性

● 同値分割法/境界値分析：

同値クラスの代表値をテストすることで、実施するテスト数を削減することが出来る。また、代表値として境界値を設定することで、欠陥が含まれやすい箇所を重点的に確認出来る。

● デシジョンテーブルテスト：

条件を全て書き出すことで考慮漏れを防止し、試験の網羅性を担保することが出来る。

● 状態遷移テスト：

処理順序による欠陥を防止する。また、一連の流れとしてテストを実施することが出来るため、テスト内容をイメージしやすい。

■留意点

● どの技法を適用する場合においても、仕様が明確であることが重要となる。動作が不明瞭な状態だと動作確認などの手戻りが発生する。

● テストの入力条件のパラメータの組み合わせの数は簡単に増えてしまう。同値クラスを大きくする、任意の2つの組み合わせを必ずテストするように条件を組むなど、組み合わせ数を絞ることが重要となる。

● テスト実施者との認識のずれを防止するため、入力内容と期待結果が分かる状態まで詳細に試験を記載する必要がある。

●8回（臨時）：要求工学：斎藤 忍 氏

■概要

要求定義とは、顧客のニーズに即したシステムを定義することである。顧客の要望を満たすシステムを提供するためには、要求からサービスの全体像をデザインすることが重要である。

本演習では要求工学の技法として以下について学習し、それらを作成する演習を行った。

【技法】

● ペルソナの生成

ユーザーの情報（名前、性別、年齢、スキル等）や好み、生活スタイルなどを記述し、詳細な人物像を作り上げる。

● サービスシナリオの作成

開発対象のサービスをペルソナが日常生活においてどのように利用するかというシナリオを記述する。

● カスタマージャーニーマップの作成

ユーザーの時系列の行動、他者やモノとの関わり方、サービスを利用する際の思考や洞察を洗い出す。

● ユーザーストーリーマップの作成

ユーザーストーリーと呼ばれるユーザー視点で書かれた要件を、ユーザーの行動ごとに洗い出し、優先順位を付ける。

■有効性

● ペルソナ

ユーザーのイメージや認識が一致するため、ユーザーの問題点や課題を議論するための時間やコストを削減できる。また、ユーザーのニーズに沿ったプロダクトやサービスを開発するためのアイデアが生まれやすくなる。

● カスタマージャーニーマップ

ユーザーの体験を視覚的に表現することにより、問題点や改善点を発見することができる。

● ユーザーストーリーマップ

優先順位を付けてユーザーストーリーを整理することにより、MVP(Minimum Viable Product)を切り出すことができる。

■留意点

● ペルソナ

先入観や思い込みを排除して作成する。現実的なペルソナを生成するためには、周囲の人を参考にするとよい。幅広いユーザー層を満足させるサービスを開発するのではなく、ただ一人のペルソナ(=メインペルソナ)を満足させるようにサービスを開発する。

● カスタマージャーニーマップ

ユーザーの思考や洞察を記述する際は、ポジティブな感情もネガティブな感情も書き出すことが大切である。

● ユーザーストーリーマップ

MVP(Minimum Viable Product)を絞り込む前に、ユーザーストーリーを発散させることが重要である。それにより、ユーザーやサービスへの理解を深めることができる。