演習コースⅠ

演習コース「ソフトウェア工学の基礎」 2020 年度 活動報告

Report on Practice Course of Software Engineering Foundations in 2020

研究員 ：二瓶 秀信（キヤノン株式会社），齋藤 美穂（株式会社ベリサーブ）
和田 久（エヌ・ティ・ティ・コミュニケーションズ株式会社）
辻 鉄也（エヌ・ティ・ティ・コミュニケーションズ株式会社）
谷口 文隆（エヌ・ティ・ティ・コミュニケーションズ株式会社）
高松 毅（旭化成エレクトロニクス株式会社），芳沢 圭一（株式会社オージス総研）
金丸 周平（株式会社東京精密），西畑 翔（アズビル株式会社）
村上 拓也（テックスエンジソリューションズ株式会社）
甲斐 流（テックスエンジソリューションズ株式会社）

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 ：鷲崎 弘宜（早稲田大学/国立情報学研究所）
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 16 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 2005-2020, 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的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]を参照されたい）．そこで 2020 年度も引き続いて，産学両面に通じた講師をお招きし，計 13 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，2020 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）．

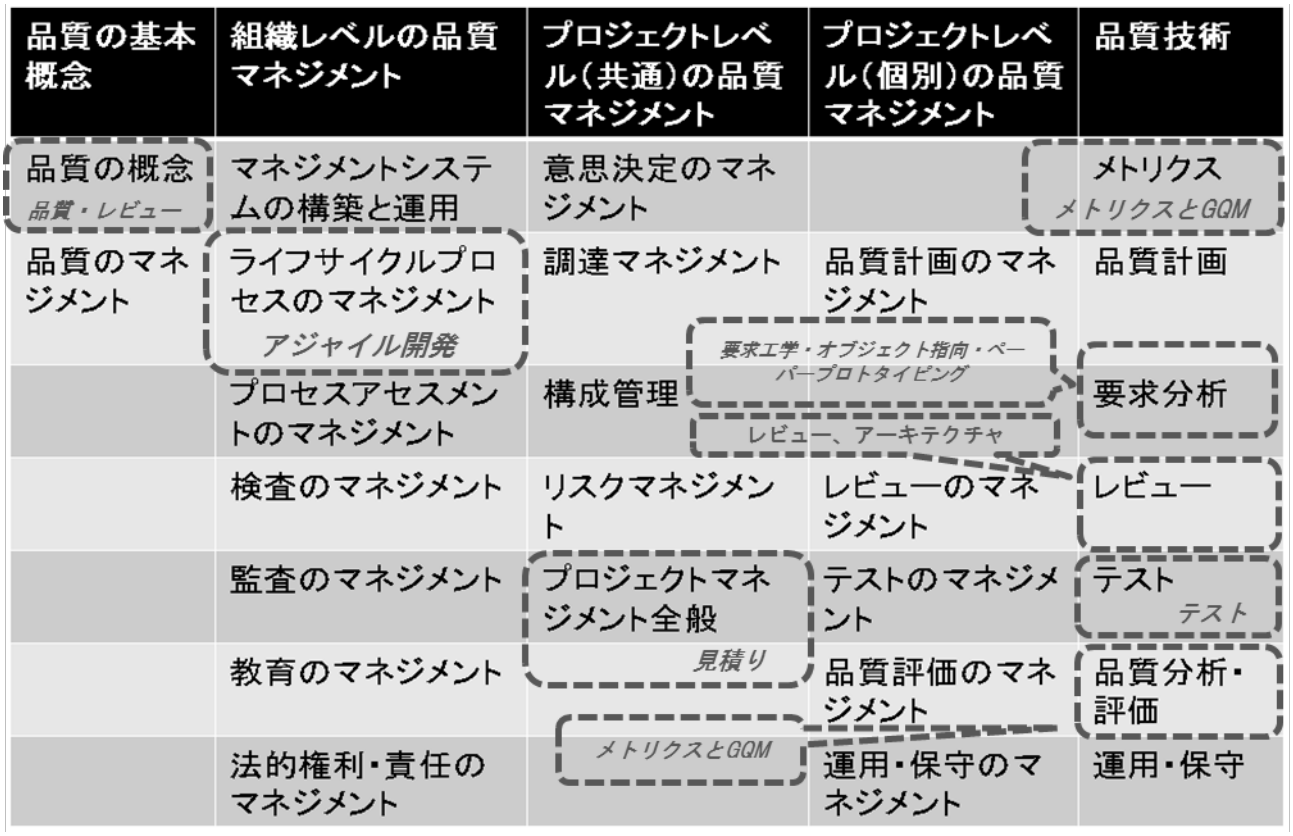

図 1：SQuBOK エリアと演習のマッピング

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．2020 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 2020 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） 2020/4/24
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）
●第 2 回（例会） 2020/5/22
オブジェクト指向分析設計：
井上　樹氏（豆蔵）
●第 3 回（合宿 1 日目） 2020/6/25
人間中心設計（UX デザイン手法）:
金山　豊浩（株式会社メンバーズ)、村上　和治（東京海上日動システムズ株式会社）
●第 4 回（合宿 2 日目） 2020/6/26
メトリクスの基礎と GQM 法によるゴール指向の測定：
鷲崎　弘宜（早稲田/国立情報学研究所）
●第 5 回（臨時会） 2020/8/20
アーキテクチャ設計・評価：
長谷川　裕一氏（合同会社 Starlight & Storm）
●第 6 回 2020/10/16
要求工学（要求分析）:
中谷　多哉子氏（放送大学）
●第 7 回（例会） 2020/11/15
アジャイル開発演習：
天野　勝氏 （株式会社永和システムマネジメント）
●第 8 回（例会） 2020/12/11
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）
●第 9 回（例会） 2021/1/8
テスト演習：
鈴木実紀夫（ASTER）

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， 鷲崎弘宜， 〈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] 鷲崎弘宜， 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] 鷲崎弘宜， 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] 鷲崎弘宜， 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] 鷲崎弘宜，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] 鷲崎弘宜，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] 鷲崎弘宜，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，鷲崎弘宜，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，鷲崎弘宜，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，斉藤慶太郎，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] 鷲崎弘宜，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] 鷲崎弘宜，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] 鷲崎弘宜，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] 鷲崎弘宜，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，河嶌 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 鉱也，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，髙田 紀行，内藤 次郎，藤井 俊久，鷲崎弘宜，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，草薙 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 槙 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

以上

●第 1 回（例会）：レビュー：猪塚 修（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．
演習はレビューのポイント，品質特性の考え方を取得することを目的に，「SNS システム」の簡単な設計図をテーマに実施した．演習内容は以下の通りである．
（準備）フォーマットの重要性
① レビューの実施（個人）
② 重視する品質特性を決めた再レビュー（個人）
③ 他人のレビュー結果の共有と集約（グループ）

■有効性：
・レビューはソフトウェアテストよりも早い段階で実施されるため，安く欠陥を見つけることができる．
・必要ならば専門家の意見を聞くことで，より良い設計にすることができる．
・テストで見つけにくい，または見つけられない欠陥（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，メンバー間の視点の共有やメンバーの教育ができる．
・レビューにおいて「てにをは」は重要ではない．性能が重要なシステムであれば性能が出ること，セキュリティが最重要であればセキュリティが担保されていることを重視してレビューを行う．

■留意点：
・レビュー対象のドキュメント/ソースコードのフォーマット（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，または欠陥を検出できなくなる．
・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．
・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重大な問題を早く見つけることが重要．
・品質特性の項目はトレードオフのものが多々あるため，システムに対してどの特性を重視するのか決めておくことが必要である．例えば性能重視のシステムであればソースコードが読みにくくなること（保守性悪化）はありえる．

●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」について理解することを目的に以下のテーマについて演習を実施した.
① 要求モデリング
② 設計モデリング

各テーマの概要
①要求モデリング
要求を正しく捉え，整理する方法を学ぶために「ストップウォッチ」を題材として，
以下の内容を実施した.
・ユースケース図，ユースケース記述，ステートマシン図の作成
　ユースケース図：システムを利用するユーザとシステムとの相互作用を表した図
　ユースケース記述：ユースケース図で記載した各ユースケースの詳細を定義したもの
　ステートマシン図：システムの動的な全体像を表した図
　　（全てのユースケースを網羅するように作成する）

②設計モデリング
UML を利用することで、ソフトウェア構造の問題点を発見しやくなることを実感するために，以下の内容を実施した.
・クラス図を用いたソフトウェア構造の欠陥の探索
　クラス図：システムの静的な全体像を表した図
　　（全てのユースケースを網羅するように作成する）

■有効性：
　要求のモデリングの際に，システムのユースケースを考慮することで要求の不備・不足を減らすことが可能である．特に，各ユースケースに対してユースケース記述を作成することで，例外フローのような異常系の動作を設計時に考慮しやすくなる効果がある.

　ステートマシン図等の UML 設計を行う際に，要求モデリングで作成したユースケースを引き続き利用し、ユースケースを網羅できているかの確認を行うことができる．また，UML を用いた設計は，コードを直接調べるよりもシステムの構造を俯瞰的に把握しやすく，システムの欠陥に気づきやすい点で優秀である.

　オブジェクト指向に対応したプログラミング言語を利用することで，詳細に記載された UML はコードに直接変換することも可能なため，モデルからコードへの変換コストを削減することも可能である.

■留意点：
　ユースケースの作成目的は，設計者とユーザとの共通認識を作るために使用するものであるため，正確に作ることにこだわりすぎないことに注意する.

　ユースケース記述では，要求の抜け・漏れを発見することが目的であるため，システムの課題や T.B.D.（To Be Determined：現在未決定だが，将来決定する内容）を積極的に記載することに留意する.

●第 3 回（例会）：人間中心設計（UX デザイン手法）：金山豊浩氏（株式会社メンバーズ）

■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」「機能・情報洗い出し」「スケッチ」「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る
3. コンセプト化
   アイデア出しのブレスト，解決案のフリーディスカッションを行う
4. ストーリーボード
   ストーリーボードの描き方の説明を受け，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：

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，ユーザのニーズを明らかにすることができるため，企画バグの混入防止に繋がる.

また，スケッチは安価であり何度も失敗できるため，たくさんのスケッチを作成しその中からベストなものを選択することが出来る.

■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，行動・振る舞いを書く際は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起承転結の 4 コマに収める形で表現し，コマ毎に必要な機能・必要な情報を洗い出す. この際も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，重要度・優先度について検討する. その後，機能と情報について複数のアプローチでスケッチし，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（例会）：①メトリクスによるソフトウェア品質把握と改善,②GQM+Strategies による組織目標と戦略の整合化：鷲崎　弘宜（早稲田大学/国立情報学研究所）

■概要：
・講義の前半ではメトリクスの基本的な事項とそのメトリクス作成に有効なGQMを,後半ではそのGQMを組織目標策定時に活かすGQM+Strategiesを学んだ.
・メトリクスにはプロダクトに関するもの（コード行数やサイクロマティック複雑度など）とプロセスに関するもの（欠陥除去率や消化済みテストケース数など）がある.
・GQMはGoal-Question-Metricの略で,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（目標指向）な枠組みである.
・GQM+Strategiesは,各社の各部署がそれぞれ目標を立てているが,その目標が繋がっておらず,有効な組織アクションにつながっていないことを背景にドイツのフラウンホーファー研究機構のIESEにて開発された手法である.
・通常,組織構造に沿って,各部署の目標や戦略を連鎖させ,各目標の達成可否判断に必要なメトリクス（測定の方法と尺度）を表現する.ここでいうメトリクスは所謂KPIのことで,目標や戦略の導出の根拠として,事実（Context）や仮定（Assumption）も明示する.
・演習では実際にGQM+Strategiesを作成し,GQMでどのように組織目標や戦略を連鎖させるかを学んだ.

■有効性：
・メトリクスを有効活用できれば,発生事象の関係を捉え,未来を予測して変えることができる.
・メトリクスを定める際に GQM を用いれば,単なる評価指標のメトリクスだけでなく,そのメトリクスを測定する目標（Goal）を定めることが可能となる.
・GQM+Strategies での仕組みを適用することで,組織構造上の上位において決定した戦略を下位部門が実現していないことや,逆に下位の戦略が組織に貢献していないこと,あるいは,測定データが組織目標と結び付いていないといった不整合を可視化でき,各部署における責任の明確化を行うことができる.

■留意点：
・メトリクスは測定することが目的にならないようにすることが大原則.
・メトリクスを定める場合は,どういう目的・目標があって,だからこれを測定すれば良いという論理が成立していることが重要.
・メトリクスを定める対象が測定対象になる場合とそうでない場合があり,そうでない場合は,代替の指標が必要となる（ソフトウェア自体に測定可能な物理量が無いため）.
・開発者や組織が,自分よがりの品質を押し付けてしまうオレオレ品質（本講義の造語）には注意が必要.それを防ぐため内部品質⇒外部品質⇒利用時の品質といった具合に段階的にどのような影響があるかを特定し,だからこのメトリクスを測定するといった具合に考えなければならない
・GQMで定める目標は狭すぎたり,広すぎたりしないように,目標を構造化しどこにフォーカスするかを定めた方が良い.
・GQM+Strategies では目標・戦略・メトリクス（目標達成可否判断に必要な指標）を立てた後は,「実行計画の策定」や「計画の実行」を行い,更に「結果の分析」を進めることで,継続的なプロセス改善を実現することが重要.

●第 5 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：

システムに求める品質を設定する「品質特性シナリオ」・アーキテクチャを設計する「ADD(Attribute Driven Design)」・アーキテクチャを分析・評価する「ATAM(Architecture Trade-off Analysis Method)」，それぞれの手法について概要を学んだ．そして，「ドーナツ店の起業・店内レイアウトの決定」を題材にした演習を通じてそれぞれの手法を用いたアーキテクチャの設計及び分析評価を実施した．

「アーキテクチャ」に決まった定義はないが，本演習ではビジネス要件・システム要件・前提条件・制約条件などに基づき導かれるシステムの土台と定義した．

各手法の概要

① 品質特性シナリオとは「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」といったシステムが備えるべき観点から品質目標をシナリオとして記述したものである．
品質特性シナリオを作成することで具体的になっていない非機能要件を分析することが可能となり，目標とする品質を設定することができる．

② 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設計する手法である．
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．次に，選択した品質特性シナリオに対して実現方法を割り当てることでアーキテクチャを設計する．
ADD によってアーキテクチャに重要な要件や設計の過程が明確となり，アーキテクチャの説明が容易になる．

③ ATAM は設計したアーキテクチャに対して品質特性に関する要求が満たされているかを評価する手法である．
まず，アーキテクチャに関する紹介を行い，アーキテクチャの調査・分析を行う．次に，利害関係者それぞれの視点から課題と解決先を議論する．
ATAM によって品質特性の重要ポイントや実現手法のリスクやトレードオフポイントを明確にすることができる．

■有効性：

アーキテクチャはシステムの土台であり，システムの品質に影響を与えるため，適切に設計や分析・評価を行うことが求められる．適切にアーキテクチャを設計することでシステムへの理解を深めることが可能になる．その結果，システム開発における「Quality」,「Cost」,「Delivery」の改善やメンテナンス性の向上，問題発生時のリスク緩和などが期待できる．

■留意点：

具体的な実現手法を考慮することで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記述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．また，単一観点からの非機能要件の抽出による要件漏れを防ぐために，異なる知見を持った様々なステークホルダと品質特性シナリオを作成することが望ましい．

ATAM による分析・評価は実施すべき手順が多く，工数や人員等の問題でアニュアル通りの実施が困難となることが多い．そのため，適用するシステムの環境に合わせて調整して実施することが望ましい．

●第６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ける要求抽出プロセスを工学的に定式化する技術である．要求抽出を実践するためのさまざまな手法があるが，それらの適用には，明らかにしたい対象を定め，最適な手法を選択し適用することが大切である．

本演習では,「郊外の大型ショッピングモールでの買い物シーン」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適用を体験した．

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．人物像をよりリアルに定義することにより，ユーザ目線の要求抽出が可能になる．

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 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）

③ 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．最終ゴールの達成に必要な作業の漏れを防止することができる．

■有効性：

要求工学のさまざまなモデル化手法を適用することによって，要求の根拠，背景を知ることができる．また，完成したモデルを関係者で共有することで，意思疎通やネゴシエーション等に利用することも可能である．

ペルソナ法やリッチピクチャのグループ演習においても､メンバー間の議論が活発化され，多くの視点が挙げられることを実感できた．ゴール指向分析では，設定するゴール如何により抽出される要求が大きく変わってくることを確認できた．

■留意点：

要求工学の各手法で分析を行う際には，必要以上に時間をかけず，タイムリーなシステム構築を目指すことも大事である．また、適切な分析手法を選択することも重要である．

●第 7 回（例会）：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）

■概要

昨今，ビジネスの進展スピードは高速化し，要求の変化が激しくなっている．そのため，ソフトウェアの開発においても，要求の変化に迅速に対応することが一層求められるようになっている．このような状況下では，従来の開発の初期段階で全ての要求内容を確定することを前提としたウォーターフォール型の開発では対応できず，小さく作って大きく育てるアジャイル開発のプロセスが採用されている．

代表的なアジャイル開発の手法としては，リーン（Lean Software Development），スクラム（Scrum），XP（Extreme Programming）がある．

スクラムでは，1 週間～4 週間のタイムボックスをスプリントと言い，スプリントは，スプリントプランニング，デイリースクラム（朝会），開発作業，スプリントレビュー，スプリントレトロスペクティブ（振り返り）で構成される．

講義ではアジャイル開発の進め方を学び，スクラムの手法に則り演習を行った．

アジャイル開発の要素である「チームで協力しながらゴールに近づいていく」をスクラムの体制（ロール），イベントを模して体験した．

【演習内容】

・丸，三角，四角のパーツが書かれている手本をもとに，同じ図形を描く．
・プロダクトオーナー，スクラムマスター，開発チームの役割になり，決められた時間内に上記の作業を行う．
・スプリントは 2 回．
　スプリント 1 の制約は，個人ごとにできる作業を制限する．
　スプリント 2 の制約はなし，個人ごとに作業の制限はない．
・スプリントでは，プランニング，朝会（3 回），レビュー／リリース，振り返りを行う．

・プロダクトオーナーと開発チームでゴールを決める．
　目標として何個の図形を完成させるかを決める（見積る）．
・開発チームは進め方を決める．
　誰が丸，三角，四角を担当するか，チェッカーは誰かを決める．（スプリント 2 では制約なし）
・スクラムマスターは進行を行う．
・バーンダウンチャートに基準線を引いて用意をする．
・開発チームは朝会で進捗状況などを確認し（3 回），レビュー／リリースを行う．
・リリース後，KPTA(Keep，Problem，Try，Action）振り返会を行う．

・スプリント 2 でも，スプリント 1 と同様に作業を進めていく．

【演習での気づき】

・一致団結して解決していけるような雰囲気になれた．
・バーンダウンチャートは目標，達成度が可視化されてよい．
・KPTA は問題やアクションが明確になり，話し合いを進めやすくなる．
　ノウハウ化できるものよい．
・みんなが同じ方向を向けるわかりやすいゴール設定が必要だと思った．
・作業の見積りは難しい．

■有効性

アジャイル開発は以下の領域が適している.

・要求の変化が激しく，あらかじめ要求が固定できない領域
・リスクの高い領域
・他社に先駆けて製品・サービスの市場投入が必要な領域

アジャイル開発の出現当初は開発者 10 名程度の規模が適していると言われていたが,大規模開発においても，複数チームによるアジャイル開発が行われている.

■留意点

ウォーターフォールに比べて，失敗が少ないと言われているアジャイル開発であるが，アジャイル開発においても失敗は発生する．その根本的な原因はアジャイル開発に関する知識不足である.アジャイル開発を行う上では,アジャイル開発に適した領域のものであるのか,アジャイル開発を行う準備が整っているかを確認した上で実施する必要がある.

以下にアジャイル開発の典型的な失敗例を示す.

・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.
　　経営層が勘違いしたアジャイルを，現場に押し付ける.
・開発チームが，プロダクトオーナーが要求を整理するのを待っている
　例：プロダクトオーナーに業務知識が不足していて，要求をまとめられない.
・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.
・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.

●第 8 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.

また得られる効果として,工数の説明力向上,コストマネジメント力向上,プロセス改善,見積リスク把握,アセット化と属人生排除などが挙げられる.

CoBRA 法の見積りモデルは計算式で表現すると以下となる.

工数(コスト) ＝ α × 規模 × （1 ＋ Σ COi）

生産性　　変動要因（CO：コストオーバーヘッド）

生産性は実績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
・変動要因に対する改善策の提案

■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 法においての算出プロセスが定義されている.
・変動要因を洗い出す際に,影響度合いの大きさや出現頻度の多さなどから,
工数コントロールのポイント(影響度の高い要因)を把握できる.
・CoBRA モデルを組織で活用し,各プロジェクトの変動要因データを蓄積・分析することで,
共通する要因の軽減・解消に向けた対策を検討できる.

■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である.
・開発規模は CoBRA 法においての算出プロセスはないため,独自で算出する必要がある.

●第 9 回(例会):テスト演習 : 鈴木 三紀夫氏 (ASTER)

■概要:
ソフトウェア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,受講者が演習問題を解き,その解答をグループ内で共有し,各グループで発表を行った.その後,解説を受け,演習問題に適用できるテスト技法の特徴や使用方法を学んだ.テスト技法としては,以下について学んだ.

・ブラックボックステスト技法
-同値分割法
入出力の関係性に基づいて,入力の条件をグループにまとめる
-境界値分析
数値の境界を識別し,境界に対するテストケースを作成する
-デシジョンテーブルテスト
条件の組み合わせの相互作用をデシジョンテーブルで整理し,それを網羅する
テストケースを作る
・ホワイトボックステスト技法
-制御フローテスト
処理の流れを制御フローで整理し,それを網羅するテストケースを作成する

■有効性:
目的に応じたテスト技法を選択することで,以下の効果を得ることができる.
・テストの目的に応じた技法を適用することで,品質を落とすことなく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.
・グラフや表を作成することで,テストケースが可視化され,テストの抜け・漏れを防ぐことができる.

■留意点:
・テストの目的に応じたテスト技法を適用する必要があるため,テスト技法の種類や特徴を把握しておく必要がある.
・様々なテスト技法が存在しており,各技法は検出できる不具合が異なる.そのため,テスト技法を使用する場合には,どのような不具合を見つけたいのかを明確にし,それに合うテスト技法を選択する必要がある.

以上