

演習コース「ソフトウェア工学の基礎」 2016 年度 活動報告

Report on Practice Course of Software Engineering Foundations in 2016

主査 : 鷺崎 弘宜 (早稲田大学/国立情報学研究所)
副主査 : 猪塚 修 (横河ソリューションサービス株式会社)
研究員 : 内藤 拓 (富士通クオリティ・ラボ株式会社)
田口 浩嗣 (日本電子株式会社), 加藤 蔵次 (株式会社デンソー)
加地 孝敏 (アズビル株式会社), 太田 延樹 (アズビル株式会社)
大森 淳夫 (パイオニア株式会社), 中村 和哉 (株式会社インテック)
辻本 亜樹 (ヤンマーエネルギーシステム株式会社)
飯塚 宏明 (テックスエンジソリューションズ株式会社)
原 好宏 (テックスエンジソリューションズ株式会社)
市川 大輔 (株式会社リンクレア)
金成 祐介 (伊藤忠テクノソリューションズ株式会社)
大澤 裕太 (東京海上日動システムズ株式会社)

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 12 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 2005-2016, 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

演習コース I 「ソフトウェア工学の基礎」

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]を参照されたい）。そこで 2016 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、2016 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）。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。

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
	プロセスアセスマ 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

演習コース I 「ソフトウェア工学の基礎」

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。2016 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 2016 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） 2016/5/20：

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） 2016/6/10：

オブジェクト指向分析設計：

井上 樹氏（豆蔵）

●第 3 回（合宿） 2016/7/7～8：

アーキテクチャ設計・評価：

長谷川 裕一氏（合同会社 Starlight & Storm）

●第 4 回（臨時会） 2016/8/9：

ペーパープロトタイピング：

浅野 智氏（UX/HCD コンサルタント）

●第 5 回（例会） 2016/10/14：

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 6 回 2016/11/25：

テスト演習：

鈴木 三紀夫氏

●第 7 回（例会） 2016/12/19：

要求工学（要求分析）：

中谷 多哉子氏（放送大学）

●第 8 回（例会） 2017/1/13：

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（臨時会） 2017/2/7：

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（早稲田大学/国立情報学研究所）

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

演習コース I 「ソフトウェア工学の基礎」

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,

演習コース I 「ソフトウェア工学の基礎」

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, 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] 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.

以上

演習コース I 「ソフトウェア工学の基礎」

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：

本演習は、ソフトウェア成果物の改善を目的としたレビューについて学ぶ。
演習内容は以下の通りである。

1. レビューの勘所を養うため、簡単な仕様書に対するレビューを個人演習として行う。
2. 他の人の視点を知るためレビューの内容を共有する。

■有効性：

レビューはソフトウェアテストと同様に欠陥を見つける手段であり、ソフトウェアテストより前の工程で行うので低コストで欠陥を発見することが出来る。また、ソフトウェアテストにて実施が難しい確認がレビューでは簡単にできる。

ソフトウェアの設計は唯一絶対ではないので、レビューを行うことにより専門家など多くの人の意見をもらうことができ、より良い設計にできる。

業務を遂行しつつ、参加者の教育が出来る手段である。

■留意点：

レビューアスキルによって指摘内容の有用性が変わる。また、レビューで指摘される基本的なことをレビューイが事前に理解し作りこみができていれば、レビューによる指摘の質がより向上する。

レビューは成果物に対して行うものであり、設計の途中レビューを行っても効果は期待できない。

また、チェックリストを埋めるなどのレビュー自体が目的となっている形式的な確認に関しても、効果は期待できない。

●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：

本演習は、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かを知り、モデリングを始めるきっかけにすることを目的として、以下の達成を目標とする。

- ・ユースケースを使った要求分析の方法を学ぶ
- ・ステートマシン図を使った要求分析の方法を学ぶ
- ・クラス図を使った設計の改善方法を学ぶ
- ・クラス図とステートマシン図から C++ のプログラムが作れることを学ぶ

演習は以下の構成で実施。

- (1) 要求のモデリング
ストップウォッチや自動ドアを題材として、ユースケース図やステートマシン図を使うことにより、要求事項が分かりやすい形で表現できることを体感する。
- (2) 設計のモデリング
設計モデルのレビューを題材として、クラス図を使うことにより、ソフトウェアの構造が見える化できることを体感する。
- (3) モデルとコードの関係
設計モデルから C++ のコードが作成できることを体感する。

■有効性：

- (1) 要求のモデリング
 - ・完成した製品やシステムのイメージを共有できる。
 - ・要求の不備不足が発見できる。
- (2) 設計のモデリング
 - ・ソースコードで読み取れる範囲より広い範囲で俯瞰できる。
 - ・ソースコードを書く前に問題を発見できる。
- (3) モデルとコードの関係
 - ・オブジェクト指向に対応したプログラム言語であれば、詳細に書かれた UML モデルからコードに変換できる。

■留意点：

一定のモデリングスキルを身に付け、モデリング手法を導入してから効果が出るまでには相当の期間とコストがかかることが想定されるため、組織の理解と協力を得た上で、長期的な計画のもとに小規模開発から試行導入することが望ましい。

●第 3 回（合宿）：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：

本演習は、アーキテクチャの品質特性とそれを表すシナリオ、一般的な実現手法とアーキテクチャの分析評価の概要について知ることを目的とする。

アーキテクチャはビジネス要件やシステム要件などから構築されるシステムの土台である。演習では「ドーナツ店内のレイアウト」を題材にグループ演習を行い、理解を深めた。

アーキテクチャの設計手法として、ADD (Attribute Driven Design) を学んだ。ADD ではシステムの品質の目標となる品質特性をシナリオ化して品質特性シナリオの作成を行う。これは「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」といった品質項目からシナリオを作成し、非機能要件を抽出する手法である。作成した品質特性シナリオから実現手法を選択し、アーキテクチャを設計する。

分析・評価の手法としては ATAM (Architecture Tradeoff Analysis Method) を学んだ。ATAM は品質特性から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。また、リスクや重要ポイントと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：

ソフトウェア開発においてアーキテクチャを適切に設計することで、目標の明確化と品質の実現が可能となる。また、システムの理解も容易になるため、開発完了後のメンテナンス性の向上も期待できる。

■留意点：

品質特性シナリオの作成では、具体的な実現手段を考えず、非機能要件を抽出するようにする。また、様々なステークホルダの観点から検討を行うことで、複数の観点から非機能要件を抽出できる。

ATAM を利用した分析/評価では、必ずしもマニュアルに沿って実行するのではなく、工数人員などの環境に合わせ、開発部門ごとにカスタマイズすべきである。

●第4回（臨時会）：ペーパープロトタイピング：浅野 智氏（UX/HCD コンサルタント）

■概要：

サービスデザインやUX（User Experience）、HCD（人間中心設計）などの最近の設計手法の概要を学び、ユーザーの本質的な要求を具体化するための手法としてのペーパープロトタイピングについて講義と演習を通して学習した。内容は下記の通りとなる。

1. 講義

サービスデザインやUX、HCDなどの最近のデザイン、設計手法を学びながら、ペーパープロトタイピングを含むプロトタイピングの有効性を学ぶ。

2. アクティビティシナリオ作成

ビジョン提案型デザイン手法を学び、「モバイルによるクッキングレシピサービス」を題材にユーザーの活動を具体的に「アクティビティシナリオ」を作成する。

3. ストーリーボード作成

アクティビティシナリオをベースに、ユーザーの目標に向かうための操作を具体的に「インタラクションシナリオ」を作成する。作成した「アクティビティシナリオ」と「インタラクションシナリオ」からユーザーの振る舞いをイラストで表現し、対応するタスクを記載した「ストーリーボード」を作成する。

4. ワイヤフレーム作成

ストーリーボードに対応したインタラクションを「ワイヤフレーム」として画面イメージ作成する。今回はスマホでのアプリという想定で作成した。

5. ストーリーボーディング（思考発話法）

作成したストーリーボードとワイヤフレームを第三者に評価してもらう。評価者は利用状況や操作感などを声に出しながら行う。実際に出た意見を参考にし、ストーリーボードやワイヤフレームを修正・評価を繰り返しながら、ユーザーにとって価値あるものを作り上げていく。

■有効性：

サービスやプロダクトの開発において、ユーザーにとって価値を探しながら作るべきものを明確にする仮説検証の手法として非常に活用できる。「機能」を中心に考えるのではなく、「ユーザーへの価値」を中心に考え、ペーパープロトタイピングとして目に見える形にして、評価することで、ユーザーの本質的な要求を満たすコト、モノを早期に発見できる。

■留意点：

ベンダーの立場であれば、クライアントの協力が必要不可欠である。企画段階から「ユーザーへの価値」を一緒に考えて、「ペーパープロトタイピング」として作り上げることで有効性が発揮される。さらに、エンドユーザーの評価も必要不可欠である。社内やクライアントだけで評価しても仮説の域をでない。

評価してもらう成果物は、評価者によって変える必要がある。エンドユーザーであれば、ビジュアルデザインまでされたモックアップを操作して評価してもらう。ペーパープロトタイピングのような見た目の色や形のビジュアルが未完成で操作感も想定できないものでは、エンドユーザーにとっては実際に使用するイメージが湧かなく、良いフィードバックが得られないと考える。

●第 5 回（例会）：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）

■概要：

アジャイル開発手法として最も採用されているイテレーション開発を、折り紙の多面体を作成するプロジェクトを題材に演習した。開発プロセスはスクラム開発をベースにリーン開発や XP(extreme programming)開発のプラクティスを取り入れ、下記のような形式で演習を実施。

{体制} PO(プロダクトオーナー)が 1 名：要件管理、製品リリースの責任者。

SM(スクラムマスター)は設定せず。

開発者が 6 名：イテレーション計画会、朝会、開発作業、振り返り会等を実施。

{工程} 1 イテレーションを 4 日間の設定とし、2 イテレーションを実施（1 日は 11 分）。

イテレーションの初日は計画会を実施。

2 日目～3 日目は最大 4 分間の朝会と開発作業を実施。

4 日目は朝会と開発作業に加え、リリースレビューを実施。

第 2 イテレーションは上記に加え初日に振り返り会を 4 分間実施。

{要件管理} 顧客要求を PBL(プロダクトバックログ)として管理。

イテレーションの始めに PO が PBL の順位付けを行い、その後 PO と開発メンバーが各イテレーションで開発する内容をイテレーションバックログとして合意を取る。

{見積もり} 1 タスクを 2 分でできる作業として規模・工数を見積もり。

第 1 イテレーションの振り返りを受けて作業規模の見積りを精査。

{進捗管理} 各開発者の作業可視化にはタスクボードを使用。

プロジェクト全体の進捗状況の可視化にはバーンダウンチャートを使用。

{振り返り} PO と開発者が参加し KPT (Keep/Problem/Try) を実施。

■有効性：

アジャイル開発では短いサイクルに分けて開発を進めるため、開発中の顧客からの要件変更に対してより柔軟に対応できる。またバーンダウンチャートおよびタスクボードによる進捗管理や、朝会での情報共有、振り返り時の KPT など常に開発を可視化することで、イテレーションごとに開発プロセスを改善させることもできる。

演習では上記の管理手法を紙ベースで実施したが、現在はアジャイル開発向けの管理ツールが豊富に揃い、アジャイル開発を行いやすい環境が整ってきていると考えられる。

■留意点：

アジャイル開発では開発規模をタスクの相対的重み付けから見積もる。そのために正確な見積もりを行うためには作業への習熟が必要になる。実際に本演習では規模見積もりの精度が低く進捗の予実と大きな乖離を発生させてしまった。

またアジャイル開発において PO の役割を明確化する必要がある。PO の役割が不明確な場合、要件の順位付けやイテレーションごとのレビューが不十分となる。アジャイル開発は計画変化に強いプロセスではあるが、上記のように変化させてはいけないプロセスもあることを考慮しないとイケない。

最後にソフトウェア品質に関する所感を記載する。アジャイル開発は比較的新しい開発手法であり、また短いイテレーションごとの開発である。そのために品質保証の方法が確立しているとは言い難い。アジャイル開発における品質保証の方法は今後も大きな課題となる。

演習コース I 「ソフトウェア工学の基礎」

●第 6 回（例会）： テスト演習：鈴木 三紀夫氏

■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、制御パステスト、同値分割、限界値分析、デシジョンテーブルテストについて学んだ。

■有効性：

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。テスト技法を適用することで、テストケース作成者の経験値によらず、一定の水準以上のテストを実施することができる。また、グラフや表を作成することで、テストケースが明確になり、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、テスト技法の種類および特徴を知っていることが前提となる。

同値分割のように、テスト技法を適用しただけでは、テストケース作成者によりテストケースが変わるものがあるため、テスト技法によっては経験を積む必要がある。また、制御パステストのように、目標とする品質によってテストの基準が変わるものもあるため、関係者間で認識を合わせることが重要となる。

演習コース I 「ソフトウェア工学の基礎」

●第7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重要な技術である『要求工学（特に要求抽出）』を実践した。「要求は不安定である」「要求の根拠を明らかにすれば、要求の安定性を予測できるはずだ」といったことを念頭に置いて、以下の3つの要求分析の演習に取り組んだ。

① リッチピクチャを用いた現状分析

1つの意見や問題に対して、1コマ漫画（表情を持った人物+その人物の発言の吹き出し）を1枚の付箋に描いていく。描き上がった複数枚の付箋を大きな紙の上や机上で並び替えて、全体を俯瞰することによって、直観的な現状把握を行うことができる。

② CATWOE 分析

要求の信念、根拠、世界観といったことを明らかにするために、CATWOEの6つの頭文字の要素に分けて分析する手法である。

- ・C: Customer（顧客）
- ・A: Actors（Tを行う人）
- ・T: Transformation Process（ある状態からWに合致する状態へと変換するプロセス）
- ・W: World View（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）

「短時間の会議を開催したい」という要求のCATWOE分析の例は、以下の通りである。

- ・C: 短時間の会議を開催したいと望む人
- ・A: 会議室の手配者
- ・T: 会議を開催する場所がない状況が、場所が提供される状況になる。
- ・W: いつでも誰でも自由使用できる共有会議室があれば、気軽に使用できて便利だ。
- ・O: 会議室の手配者
- ・E: 短時間の会議専用の会議室が提供されている。

上記の「O: Owners」を変更して分析してみると、「T: Transformation Process」と「W: World View」が変わる。CATWOE分析によって、人の立場による世界観の違いを理解できる。

- ・T: 短時間の会議を開催できない状況が、短時間の会議を開催できる状況になる。
- ・W: 短時間の会議を開催して、意思疎通の円滑化による、業務の生産性向上は重要だ。
- ・O: 短時間の会議を開催したいと望む人

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）を階層的に少しずつ詳細にしていくことによって、最終的なゴールを導き出すことができる。

■有効性：

複数の要求分析手法を取り入れることで、多角的に要求を抽出し分析できる。同時に、要求漏れを少なくすることが期待できる。

■留意点：

複数の要求分析手法を行うと、当然、時間がかかるし、重複した要求が挙がるだろう。顧客に合った要求分析手法を選択したり、自身が所属する組織が得意とする要求分析手法を採用したりするのが良いと思われる。

●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：

開発プロジェクトを発注/依頼する場合、工数見積りの根拠が見えないと、発注者側は妥当性が分からない、受注者側は妥当性を説明できない、という問題がある。このような問題に対応すべく、本演習では、見積りの根拠を「見える化」する工数見積りモデルの構築手法「CoBRA (Cost estimation, Benchmarking and Risk Assessment) 法」を学んだ。

CoBRA 法は、見積りモデルを作成する方法である。この方法は、「同じ規模（例：プログラム行数）のプロジェクトでも、かかった工数（例：人月）に違いがある」という状況をベースの生産性 α （規模に比例する部分）と、そこからのオーバーヘッド (CO)（プロジェクト特有のコスト変動要因（以下、変動要因とする））の二つに分けて考える。 α は、過去のプロジェクトのデータを利用し算出する。CO は、熟練者の経験を利用して、モデルを作成する。

演習内容は、以下の通りである。

- ・グループで変動要因の洗い出しを行い、変動要因を定義した（影響度レベル（工数への影響の大きさ）の定義を含む）。
- ・IPA で公開されているツールとモデル例を使用して、ハンズオン形式でモデル構築とモデルの改善を行った。
- ・グループで定義した変動要因に対する改善策を、グループで議論した。

■有効性：

熟練者の経験を利用しているが、熟練者の経験を変動要因としてモデル化しているため、従来の「勘」ではなく、説明力のある見積りを実現している。

熟練者 2-3 名の協力と 10 件程度の過去プロジェクト実績データによりモデル構築ができるため、比較的容易に構築できる。

見積り時に、変動要因のレベルを変動させることにより、どのような場合に工数超過のリスクがどのくらいになるかを分析できる。また、その結果を元にリスク・コストのコントロールや改善を行うことが可能となる。

■留意点：

規模の見積りは、CoBRA 法の範囲外のため、別途、規模を見積もる必要がある。

変動要因は、数多くなりがちであるが、10 個程度に抑える方がよい。数が多くなった場合、一旦、7 個程度にして、妥当なモデルになるまで数を増やしていくとよい。

●第9回(臨時会):メトリクスの基礎と GQM 法によるゴール指向の測定:鷺崎 弘宜(早稲田大学/国立情報学研究所)

■概要:

メトリクスは、測定の方法と尺度であり、プロセスやプロダクトの品質の評価や改善に有用である。ただし、評価や改善という目的に合致した適切なメトリクスを測定する必要がある。GQM (Goal-Question-Metric) 法は、目標 (Goal) を定め、質問 (Question) を積み重ねて、測定するメトリクス (Metric) を定める手法であり、適切なメトリクスの選定に有効である。

メトリクスの基礎知識、代表的なメトリクス、使いどころと落とし穴について学んだ。GQM 法についてと詳細な手順、コツとして、以下のことを学び、それにしたがって、グループに分かれて GQM モデル (グラフ) を作成した。

- ・目標テンプレートを使用して、目標の明確化と共有。
- ・Question 導出の観点、上位ゴール、仮定、解釈を考えて、質問やメトリクスの検討。
- ・メトリクスを取る際のメカニズムの検討。

そして、作成した GQM モデルを見直し、発表を行い、今後の取り組みを考えた。

その後 GQM 法の応用である GQM+Strategies 法について学習した。

■有効性:

- ・GQM 法はソフトウェアの改善だけでなく、一般的な業務でも使える。
- ・仮説を明確にすることで他者に説明しやすく、メンテナンスしやすくなる。
- ・測定すべきメトリクスが明確になると、個人レベルで測定できるか、組織で対応しないと測定できないかが明確になる。
- ・メトリクスの相関を見つけられれば、予測が可能になる。

■留意点:

- ・ゴールが抽象的だと難しいのでサブゴールへの分解などが必要。
- ・同じメトリクスでも人により定義や測定方法が異なり、測定結果が異なることがあるので、定義や測定方法の共有が必要。
- ・GQM を一回やっただけでは不十分で、PDCA を回してブラッシュアップが必要。
- ・メトリクスを決めると、そのメトリクスのみ達成しようとして、本質を見失うことがある、負のホーソン効果が起きないように、多面的にみるなど気をつけないといけない。(負のホーソン効果とは、注目して測定したメトリクスは見かけ上改善されるが、意図してない他の重要な側面に悪影響を及ぼすこと。)

以上