

# 演習コース「ソフトウェア工学の基礎」 2012年度 活動報告

## Report on Practice Course of Software Engineering Foundations in 2012

主査 : 浜田 浩史 (伊藤忠テクノソリューションズ株式会社)  
副主査 : 鷺崎 弘宜 (早稲田大学/国立情報学研究所)  
猪塚 修 (横河ソリューションズ株式会社)  
研究員 : 朝井 与志哉 (株式会社ニコンシステム)  
加藤 尚樹 (日本電子株式会社)  
楠森 賢佑 (株式会社 NTT データ)  
久原 健一 (日本電子株式会社)  
駒井 利之 (株式会社アドバンテスト)  
鈴木 勝統 (矢崎総業株式会社)  
鈴木 達郎 (富士ゼロックスアドバンストテクノロジー株式会社)  
田中 孝一 (日本電気通信システム株式会社)  
東久保 理江子 (アンリツ株式会社)  
永瀬 孝紀 (アズビル株式会社)  
森 俊樹 (株式会社東芝)

### 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8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 2005-2012, 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<sup>[1]</sup>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<sup>[2]</sup>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<sup>[3]</sup>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)<sup>[3]</sup>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]を参照されたい）。そこで 2012 年度も引き続いて、産学両面に通じた講師をお招きし、計 11 名の研究員が参加して、全 10 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、2012 年度に取り上げた各技術の位置付けを識別し、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 2）。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 品質・レビュー	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 品質データ分析(測定)
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 アジャイル開発	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		要求工学・要求分析 プロジェクト指向
	検査のマネジメント 監査のマネジメント	リスクマネジメン ト 品質・レビュー プロジェクトマ ネジメント全般	レビューのマネ ジメント アーキテクチャ	レビュー 形式手法
	教育のマネジメント		品質評価のマネ ジメント 品質データ分析(測定)	品質分析・ 評価 テスト テスト
	法的権利・責任の マネジメント		運用・保守のマ ネジメント	運用・保守

図 2 SQuBOK における各技術の位置づけ

### 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)を招いて実施した。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 10 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。2012 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 2012 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) 2012/5/11 :  
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)
- 第 2 回(例会) 2012/6/15 :  
レビュー演習 :  
猪塚 修氏(横河ソリューションズ株式会社)
- 第 3 回(合宿) 2012/7/19 :  
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)2012/9/4 :  
オブジェクト指向モデリング :  
井上 樹氏(豆蔵)
- 第 5 回(例会) 2012/10/12 :  
要求工学(要求分析) :  
中谷 多哉子氏(筑波大学大学院ビジネス科学研究科)
- 第 6 回(臨時会)2012/10/25 :  
モデル駆動開発 :  
久保秋 真氏(株式会社アフレル)
- 第 7 回(例会) 2012/11/16 :  
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 8 回(例会) 2012/12/21 :  
ペーパープロトタイピング :  
浅野 智氏(横浜デジタルアーツ専門学校)
- 第 9 回(例会) 2012/1/18 :  
ソフトウェアテスト :  
鈴木 三紀夫氏(MRT コンサルティング)
- 第 10 回(臨時会)2013/2/1 :  
メトリクスによるプログラムソースコードの品質把握と改善  
- GQM 法とメトリクス動向 - :

鷺崎 弘宜氏（早稲田大学）

#### 4. おわりに

本コースでは、指導講師による 10 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.

## 付録1：各回のまとめ

●第1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：

折り紙による多面体作成のグループ演習を通して、アジャイル開発の基礎を理解することを目的とする。演習では10分を1日として扱い、仮想的にアジャイル開発のプロセスを体験した。

1日目：作成する多面体の種類と数を決定し、細かいタスクに分割して、タスクボードに貼り出す

2～4日目：朝会(4分)、作業(6分)を繰り返す(タスクボード、バーンダウンチャートを使用)

5日目：結果の報告及びふりかえり(KPT)を行う

### ■有効性：

朝会や、タスクボード・バーンダウンチャートを使用することで、プロジェクトの進捗状況やプロジェクトメンバーの作業負荷の「見える化」ができる。これにより、プロジェクトの問題や失敗に早く気付くことができるため、大きな失敗を抑え、「早く小さく失敗できる」ことがメリットである。また、短期間で小さな開発を繰り返すため、要求の変更に對して柔軟に對応できる。

### ■留意点：

あらかじめ定めた作業期間を守ることが、アジャイル開発の特徴であるため、事前の見積の精度が低いと、タスクの積み残しが増えてしまう。実際のプロジェクトでアジャイル開発を行うためには、要求側とコミュニケーションを密に取り、開発の状況をしっかりと理解してもらうことが重要である。また、大規模プロジェクトでアジャイル開発を適用する場合には、プロジェクトメンバー間の情報共有や、要求側との連携をどのように上手く取っていくのかが大きな課題となる。

●第2回(例会)：レビュー演習：猪塚 修氏（横河ソリューションズ株式会社）

■概要：

演習を通じて、ソフトウェアレビュー技法の基礎や、レビューの視点（観点）を学ぶ事を目的とする。序盤、レビューの目的、定義、チェックシート、品質特性などをテーマに解説をし、個人演習を行った。個人演習の結果は、一件一葉でポストイットに記入し、模造紙に説明をしながら貼り付ける形式で発表した。その後、グループ演習やシナリオベースのレビューに関する解説を通じて、レビュー技法の基礎を学んだ。

■有効性：

レビューはトラブルを上流工程で未然防止するための有効な手段の一つである。本演習で学んだ技法を通じて改善・効率化することで、より多くのトラブル削減（コスト削減）を計ることができる。具体的には、チェックシートの利用によりレビュー対象が要求を満たしていることを、より確実に確認することができる。また、シナリオベースのレビューを行うことで、ユーザー視点から見た要件に関して、客観的に検証をすることができる。

■留意点：

しばしば、レビューをすること自体が目的になることがある。メリハリのないレビューや、形式的な確認しか行わないレビューは無駄にすぎず、手段と目的を取り違えないよう、注意が必要である。また、チェックシートを作成・運用するにあたり、リストが肥大化しないようにすること。汎用的に作りすぎてチェックの観点が不明確になったり、リストが肥大化しすぎて運用されなくなったりするのは良く見られるケースである。レビューを受ける側は、目的、ポイントを明確にし、円滑なレビューが実施できるよう務めることが大切である。

● 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：

アーキテクチャの設計と分析/評価について学び、「ドーナツ店の起業」をテーマにグループ演習を行った。アーキテクチャの設計手法として ADD(Attribute Driven Design)を学習した。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を作成する。システムの品質特性としては、“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”が挙げられる。作成した品質特性シナリオから実現手法の候補を挙げてアーキテクチャを設計する。次に、アーキテクチャの分析/評価を行う。分析/評価手法として ATAM(Architecture Trade-off Analysis Method)が存在する。ATAM は、アーキテクチャからリスク、トレードオフポイントを抽出して評価を行う手法である。開発後のアーキテクチャ管理では、VG(McCabe Cyclomatic Complexity)等のメトリクス、DSM(Dependency Structure Matrix)といった分析手法を用いることができる。

■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、目標の明確化及び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り、開発完了後のメンテナンスも実施しやすくなる。

■留意点：

品質特性シナリオの作成では、具体的な実現する手段まで考えてはいけない。また、品質特性シナリオの構成要素の応答測定は、測ることができるものを設定する。アーキテクチャの分析/評価を開発部門で初めて実施する場合、マニュアルに沿って全て実施しようとするのではなく、現在の開発プロセスに合わせてカスタマイズして活用すべきである。

●第4回(臨時会)：体験 オブジェクト指向モデリング：井上 樹氏 (豆蔵)

■概要：

UMLを使って要求、設計をモデル化することを学び、演習問題にて内容を確認し、演習問題にて理解を深めた。要求のモデル化ではユースケース図、ユースケースシナリオ、ステートマシン図、シーケンス図を用いて何を要求されているかをわかりやすい形で表現をする方法を学んだ。演習ではストップウォッチ、自動ドアの例で実際にUMLを書いて理解を深めた。設計のモデル化においては、与えられたクラス図から何が悪いのか？と共に設計の善し悪しの判断について学んだ。またモデルとコードの関係から詳細に書かれたUMLモデルからはコードを出力するなど可能な事を学んだ。

■有効性：

モデル化により、要求、設計、それら共通として下記のメリットがある。

- ・要求のモデル化によるメリット⇒漏れの発見
- ・設計のモデル化によるメリット⇒ソフトウェア構造が見える化でき、広い範囲の俯瞰、ソースコードを書く前に問題を発見できる。
- ・要求・設計共通のモデル化によるメリット⇒非常に見やすい形になる

モデルとコードの関係においては詳細なUMLからのコード生成によって生産性の向上(モデル上で設計が可能による手戻りの削減など)が期待できる。

■留意点：

一概にモデル化といっても、上記のように要求についてのモデル化と設計についてのモデル化は別であり、目的も若干違うことに注意する。またUMLモデルからのコード出力においては、詳細なUMLの記述、高いモデリングスキル、ツールの理解などが必要で、容易かつ速やかに実践できるものではない。



●第5回(例会)：要求工学(要求分析)：中谷 多哉子氏(筑波大学大学院ビジネス科学研究科)

■概要：

要求とは不安定なものである。その不安定な要求に対し要求抽出の目的は、「要求変更に対するリスクの低減」、「本質的に取り組む課題の明確化」である。

本演習では、RODANの役割依存モデルに基づき、リッチピクチャー→役割依存モデルの作成→CATWOE分析→ゴールモデルの作成から要求抽出について考えた。

以下に、本演習で取り組んだ各手法の概要を示す。

- (1) リッチピクチャー：漫画や吹き出しを用いてステークホルダ(利害関係者)の意見を俯瞰し、直感的な現状把握を行う手法。
- (2) 役割依存モデルの作成：RODANの役割依存モデルを考え、リッチピクチャーで抽出した意見に対して役割(Owner、Actor、Customer)を定義しその関係を把握する。
- (3) CATWOE分析：上記で把握した関係を図示してまとめ、Customerの状態の変化がOwnerの意見・問題意識(世界観)から逸脱していないことを明確にする。このとき、Ownerが考える「Customerの状態を変換するプロセスが合理的であると説明する理由」がOwnerの世界観に相当し、特に重要である。
- (4) ゴールモデルの作成：上記で導き出したOwnerが考える「Customerの状態を変換するプロセスが合理的であると説明する理由」をTopGoalに据えて木構造のモデルを作成する。このとき、TopGoal以下のSubGoalにはCATWOE分析で明確にした変換プロセス実行後の状態やそのプロセスを実現するために必要なリソースなどを記入する。また、and、or 接続端子や貢献度(++、+、-、--)等の情報を付加することもできる。

■有効性：

Ownerの世界観(要求の本質的な根拠)を考えることで、不安定な要求に対して変更可能性を把握し、複数の要求に対してもその依存関係(競合関係や相乗関係)を把握することができる。

また、この関係を資料(見えるもの)として残すことができるので、要求の変更時や複数要求で競合関係が生まれた際の顧客を納得させる資料にもなり得る。

さらに、複数プロジェクトを通して考えた場合、要求の可変部分と共通部分を把握することもできる。

■留意点：今回演習で取り組んだ内容は要求工学全体から見ても一部でしかない。要求工学の手法は数多くあり、それぞれその特色(使用すべきところ、強い点など)は異なる。そのため、マニュアル化した手法をこなすのではなく、それらの特徴を十分に考慮してその時々により最適な手法を選択していくことがプロジェクトの成功にとって大切なこととなる。そのため、要求工学の広い世界を知ることが重要となる。

●第6回(臨時会)：モデル駆動開発：久保秋 真氏 (株式会社アフレル)

■概要：

モデル変換の演習を通じて、モデル駆動開発 (MDD) に関する用語とモデル変換方法を理解することを目的とする。モデル駆動開発 (MDD) の基礎知識についての座学と演習を段階的に行い、無理なく理解を深めることができた。座学ではモデル駆動開発の意義、技術用語、モデル変換に使う仕様や技術、事例などを学んだ。演習ではクラス図やメタモデルを各グループで作成し解説した。また、自律走行ロボットの組込みソフトウェア開発として、MDDによるモデルからコードを自動生成するプロセスを実体験した。

■有効性：

ソフトウェアが大規模化・複雑化する中で、一般的なモデルはスケッチとして各工程の担当者が作業を理解する単なる助けであるのに対し、MDD は正確なモデルを使って自動化を可能にする。モデル変換を行うことによって、誰でも同じことができるようになり、開発を進めることができる。自動生成によりプログラミングの効率化、品質の均一化、安定化につながる。また、コードだけでなく、テスト用モデルや分析モデル、テストケースやテストデータ、ビルドや配布用スクリプトなどのモデルや関連文書を生成できる可能性がある。

特に、ロボットを用いた演習は、得た知識をその場で触りながら体験できた。大変興味深く、印象に残った。

■留意点：

モデル変換技術は、モデリングとプログラミングの紐付けであり、両者を結びつける技術である。いかに人の知識経験を処理可能なデータにするか、つまり人が紐付け・付加する情報のデータ化、手順の自動化などの変換規則が鍵となる。この結びつける方法や知識が見える化し、コード作成の技術を属人化しないことで、再利用を可能にする。

●第7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：

工数見積りモデル構築手法である CoBRA 法の説明を中心に、工数見積りの考え方、モデル構築上の注意点等を学んだ。CoBRA 法は、1997 年に独フ라운ホーフ財団実験的ソフトウェア工学研究所（IESE）で開発された、比較的新しい工数見積り手法。規模と工数の比例関係を仮定した上で、現実のプロジェクトにおいて生じる様々なコスト変動要因を考慮することにより、工数見積りに適切な補正をかけることができる。CoBRA 法の講義の後、演習では、IPA SEC から公開されている CoBRA 法支援ツールを使って実際に CoBRA 法を体験し、工数見積りモデル構築の一連の流れを理解した。

■有効性：

工数見積りには様々な手法があり、大きくは、デルファイル法などの熟練者の経験に依存した手法と、COCOMO などの過去のプロジェクトデータに基づいたモデルベースの手法に大別される。CoBRA 法は、その両者の特徴を併せ持ち、良いところ取りをしたような手法であり、比較的少数の過去データ（約 10 件）でモデルの作成を始めることができ、熟練者の経験・ノウハウ等の定性的な情報をコスト変動要因として取り込むことで、定量的な工数見積りモデルに反映することができる。また、モデル構築・見直しの一連のプロセスを通じて、熟練者の経験・ノウハウを形式知化し、工数の変動要因についてより深く理解することが可能となる。

■留意点：

コスト変動要因の洗い出しとその影響度レベルの設定が、CoBRA 法適用の成否を分ける一番のポイントであり、3 名程度の熟練者によるブレインストーミングを行うなど、幅広い視点で、様々な可能性を考慮して実施する必要がある。最初から最適な見積りモデルが構築できるとは限らない。予測と実績の乖離をよく分析して、コスト変動要因の見落としがないか、影響度レベルの設定は適切か、よく検討すべき。また、過去データにおいて精度の良い見積りモデルが構築できたとしても、外部環境等の変化により、予測がずれてくる可能性もある。継続的に見積りモデルの見直しを行うことが重要。

●第8回(例会)：ペーパープロトタイピング：浅野 智氏（横浜デジタルアーツ専門学校）

■概要：

ペーパープロトタイピングとはデザイナーやエンジニアが開発実施前にラフな（≠雑な）紙で作ったプロトタイプを作成しシミュレーションを行うことで、そのデザインの持つ問題点や使いにくさを早期発見する手法である。ペーパープロトタイピングには設計視点の手法（紙で作成したワイヤーフレーム、モックアップなど）や、評価視点の手法（発話思考法、オズの魔法使いなど）といった種類がある。本演習ではストーリーボーディングによる演習を行った。

演習手順

1. ペルソナ、およびペルソナのアクティビティシナリオ（※1）を作成
2. シナリオをタスクに分解し、タスクシートとワイヤーフレームを作成
3. 被験者が発話思考法（※2）を行いながら、操作のシミュレーションを行う
4. 被験者が発言した内容で気付いた問題点や気になる発言をポストイットに記入し、タスクに貼り付ける

※1 ユーザーが行う作業をシステムの用語を使用せずに日常的で抽象的な用語で記述するシナリオ

※2 被験者が頭に浮かんだことをアナウンスしながらテストする手法

■有効性：

操作のシミュレーションにより、容易に不足点や改善点を把握することができる。例えばシミュレーションで被験者が操作を見ることで UI 部品の不足や画面遷移の不足が見つかり、発話思考法で「〇〇があればいいのに」といった発言により改善点を把握できる。

■留意点：

複数のペルソナが存在する場合や多くの機能を持つアプリケーションの場合には全てのシナリオでペーパープロトタイピングを行うには時間がかかるため、全てのシナリオ・機能に実施するのは現実的ではなく、シナリオ・機能の絞り込みが重要である。被験者としてシナリオ作成や UI 作成に参加していない関係者を必要とするため、プロジェクトによっては被験者の確保が難しいケースも考えられる。

●第9回(例会)：ソフトウェアテスト：鈴木三紀夫氏（MRT コンサルティング）

■概要：

演習問題を通して、ソフトウェアテスト技法の基礎を学ぶことを目的とする。受講者は事前にホワイトボックス、ブラックボックス、デシジョンテーブル、境界値、同値、パステストといった用語を予習した上で演習に臨んだ。

演習問題は、制御フローパステスト、同値分割テスト、境界値テスト、ドメインテスト、デシジョンテーブルテストのテストケースを作成する。演習はまず個人で問題を解き、その解答を持ち寄ってグループで検討し、最後にグループ毎に解答を模造紙に書き、テスト設計の意図を説明する形式で行った。

■有効性：

演習を通して各技法を用いたテスト設計の方法を身につけることができた。ドメインテストの設計において、マトリックスの代わりに数直線を用いる方法や、デシジョンテーブルの書き方のコツについての説明は特に役に立ちそうだと感じた。解説は演習後とすることで頭でっかちにならず、かつ自らの実力を測ることもつながり有用な教育手法にも出会えた。

■留意点：

やらないテストをどのように他人に納得してもらうか、その為には事前にテスト意図を明らかにし説明責任を果たせるようにする。制御フローパステストではフローグラフを書きカバレッジ基準を決める、同値分割テスト項目レビューでは想定した考え方が妥当か確認する、境界値テストではバグを見つけるための規則性へ着目をする。

●第10回(臨時会)：メトリクスによるプログラムソースコードの品質把握と改善 - GQM法とメトリクス動向 - : 鷺崎 弘宜氏 (早稲田大学)

■概要：

メトリクスと Goal-Question-Metric (GQM) 法の基礎を学んだ後、GQM法を用いたワークショップ演習に取り組んだ。メトリクスとは測定の方法と尺度であり、プロダクトメトリクスとプロセスメトリクスに分類できる。本臨時会で学んだプロダクトメトリクスは、コード行数、ファンクションポイント、サイクロマティック複雑度などであった。

一方、プロセスメトリクスは、欠陥除去率などを学んだ。

GQM法とは、メトリクスを「目標」、「質問」、「メトリクス」の順にトップダウンで決定する手法である。

GQMを拡張した考え方では、これに「ビジネスゴール」、「仮説」、「解釈」、「メカニズム」が加わる。

ワークショップ演習では、チーム毎に目標（開発プロセスの遵守、出荷後品質の改善、自動化テストの容易化）を設定し、拡張した考え方のGQM適用に取り組んだ。

■有効性：

メトリクスは、ソフトウェアの定量的な品質管理に必要不可欠である。定量的な品質管理によって、ヒストグラムや散布図などによる可視化、欠陥数予測モデルの構築などが可能になる。GQM法は、測定目的とメトリクスとの対応関係が明確にできる。対応関係の明確化によって、測定自体が目的になってしまうのを防ぎ、目標の達成度合いも把握しやすくなる。

■留意点：

メトリクスを用いる際は、その目的が重要である。なぜなら、目的が曖昧だと、測定自体が目的になってしまうからである。また、測定値を意思決定に用いる場合、複数のメトリクスで判断するのが重要である。なぜなら、メトリクスには「実世界で測定できるのは、測定したい概念の一部に過ぎない」という限界があるからである。