

演習コース「ソフトウェア工学の基礎」 2015 年度 活動報告

Report on Practice Course of Software Engineering Foundations in 2015

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 : 中田 絢子 (ソニー株式会社)
齋藤 綾乃 (株式会社インテック)
村瀬 邦彦 (テックスエンジニアソリューションズ株式会社)
吉井 誠 (株式会社リンクレア)
江口 大祐 (株式会社インテック)
高木 聡 (アズビル株式会社)
植村 光吉 (ソニーイーエムシーエンス株式会社)
飯田 勲 (東京海上日動システムズ株式会社)
油谷 啓之 (日本トラスティ・サービス信託銀行株式会社)
根岸 翔 (テックスエンジニアソリューションズ株式会社)

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 11 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 2005-2015, 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]を参照されたい）。そこで 2015 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、2015 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 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 メトリクスと GQM
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
	プロセスアサセ メントのマネジメン ト	構成管理		要求分析 要求工学・オブジェクト指向・ ペーパープロトタイピング
	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、アー キテクチャ	レビュー アーキテクチャ
	監査のマネジメン ト	プロジェクトマネ ジメント全般 見積り	テストのマネジメ ント	テスト テスト
	教育のマネジメン ト		品質評価のマネ ジメント メトリクス	品質分析・ 評価 GQM
	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 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。2015 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 2015 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） 2015/5/15：

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） 2015/6/12：

オブジェクト指向分析設計：

井上 樹氏（豆蔵）

●第 3 回（合宿） 2015/7/9～10：

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） 2015/8/5：

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） 2015/10/9：

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 2015/11/27：

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 7 回（例会） 2015/12/18：

テスト演習：

猪塚 修（横河ソリューションサービス株式会社）

●第 8 回（例会） 2016/1/15：

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）

●第 9 回（臨時会） 2016/2/1：

要求工学（要求分析）：

中谷 多哉子氏（放送大学）

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.

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：

序盤にレビューの目的や手法などについて座学形式で学び、実際に品質特性やシナリオベースのレビューの演習を行うことでよりレビュー方法の違いによる有効性を学ぶことができた。

また、ポストイットに各自のレビュー結果を記載しグループ内討議をすることで他者のレビュー観点など新たな気づきを得ることができた。

■有効性：

ソフトウェア開発では、後工程になってから発覚する欠陥ほど、修正に掛るコストが増加する。レビューは欠陥をより上流工程で発見するための有効な手段のうちの 1 つである。そのため、各工程（要件定義工程、設計工程、開発工程など）でレビューを実施することは、品質の観点からとても有効である。

また、シナリオベースのレビューを実施することでユーザーの視点をレビューに取り入れたり、機能性、信頼性、使用性、効率性、保守性、移植性といった品質特性の観点でのレビューを取り入れたりするなど、様々な視点でレビューを行うことでレビューの効果をより高めることができる。

■留意点：

レビューは、限られた時間の中で実施し効果をださなければならないため、レビューでは品質特性のどの特性に対してレビューを実施するのかなどレビュー実施時の目的が重要である。また、複数人でレビューの観点を変えて実施するなどの効率的に欠陥を指摘するための検討も必要である。

レビューを受けるレビューイは、レビューアがどのような観点でレビューを実施しているかを知ることや、レビューの重要性を念頭におき、円滑なレビューを実施できるよう心掛けることが重要である。

●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：

「モデリング」はオブジェクト指向分析設計の基盤である。本演習では与えられた課題に対して実際に UML による「モデリング」を行ない、考慮すべき点や注意点、モデリングが開発でどのように役立つのかといったことを学ぶ。演習は大きく「要求のモデリング」、「設計のモデリング」、「モデルとコードの関係」の 3 つで構成される。

「要求のモデリング」は何を要求されているかを分かりやすい形で表現することが目的である。演習ではストップウォッチを題材に「ユースケース図」と「ステートマシン図」を作成した。

「設計のモデリング」はソフトウェアの構造を見える化することが目的である。演習では自動ドアを題材に「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」を作成した。

「モデルとコードの関係」では、UML から C++ へどのようにマッピングされるかを学んだ。これらを実際に行ってみることで「ユースケース図とステートマシン図を使った要求分析の方法を人に説明できる」、「クラス図を使った設計の改善方法を人に説明できる」、「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人に説明できる」ようになることが本演習の目標である。

■有効性：

「要求のモデリング」を行うことにより、開発に携わる関係者間で完成イメージの共有が促進され、要求の不備不足が発見しやすくなる。「設計のモデリング」を行うことにより、ソースコードで読み取れる範囲よりも広い範囲を俯瞰でき、より効率的なレビューを行うことができる。これらのメリットによって、開発プロセス全体の手戻りが抑制されることが期待できる。

UML を使ったモデリングでは、クラス図でシステムの静的な側面を、ステートマシン図で動的な側面を、アクティビティ図で機能的な側面を扱うことができる。これら 3 つのモデルで対象を捉えることによって、モデリングツールからソースコードの自動生成が可能になり、コーディングの工数削減やコードの品質向上に繋がる。

■留意点：

モデルベース開発で重要な要素として 7 つの鉄則と呼ばれるものがある。

1. 高いモデリングスキル, 2. ツールの理解, 3. モデルベース開発に対応したマネジメント, 4. モデルベース開発に対応したプロセス, 5. モデルの構成管理, 6. スモールイン, 7. 長い目

一定水準のモデリングスキルを身に付けるための学習コストは C 言語等のプログラミング言語の学習コストに比べて非常に高く、習熟に必要な期間の個人差も大きい。モデルベースでうまく開発ができるようになるまでには、年単位の期間と高い学習コスト、それを許容できる組織体制が必要であるが、メリットを享受する前に挫折する事例が多いため、地道な啓蒙活動と小さな案件から実績を積み重ねていくことが重要である。

●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：

本講座では「ドーナツ店内のレイアウト」をテーマにアーキテクチャ設計の分析/評価について演習を行った。アーキテクチャはビジネス要件、システム要件、前提条件、制約条件などに基づいて導かれるシステムの骨格である。

まず、アーキテクチャ導出の手法のひとつとして品質特性シナリオを学んだ。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特性からシナリオを作成し、品質の目標を明確にする手法である。

次に ADD (Attribute Driven Design) によりアーキテクチャ設計を行った。ADD では品質特性シナリオまたはユースケース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。そして、それぞれに実現手法を割り当て、アーキテクチャを設計する。

最後に、アーキテクチャ設計の分析/評価の手法として ATAM (Architecture Trade-off Analysis Method) を学んだ。ATAM によって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どうかを評価できる。また、重要ポイントとリスクとトレードオフポイントを抽出できる。

■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、目標を明確にし、品質の実現が可能となる。また、システムの高品質化や開発後のメンテナンス性を向上できる。

■留意点：

品質特性シナリオの作成では、非機能要件を十分に抽出するため、具体的な実現手段を考えない。加えて、各利害関係者を含むメンバーで検討を行うことで、非機能要件をさらに獲得できる。ATAM は開発部門の環境によってマニュアル通りの実施が難しいことが多い。このため ATAM はプロジェクトに適用できるようにカスタマイズすることが望ましい。

●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：

ペーパープロトタイピングは、ユーザーの要求を満たす解決策を作成する手法の一つであり、使いやすさやデザイン上の問題点を早期に発見・修正するために有効である。

本講義ではUX・人間中心設計を講義形式で、ペーパープロトタイピングを演習形式で学び理解を深めた。

<実施内容>

1. シナリオ作成

ペルソナを元にアクティビティシナリオを作成し、ユーザーの”活動（コト）”を物語形式で記載する。アクティビティシナリオを元にインタラクションシナリオを作成し、ユーザーが目標に向かうための具体的な”操作（モノ）”を物語形式で記載する。

2. ストーリーボード作成（UXフロー）

2つのシナリオを元にストーリーボードを作成する。シナリオをステップに分け、ステップ毎に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する。

3. ワイヤフレーム作成

「2. ストーリーボード作成（UXフロー）」と並行して、ワイヤフレームを作成する。ストーリーボードとワイヤフレームが1対1となるように過不足確認する。

4. ウォークスルー（思考発話法）

チーム外の第三者に被験者として参加してもらい、ウォークスルーを実施する。被験者には、ストーリーボードを読み、ワイヤフレームのインターフェースを操作して思ったことを声に出してしてもらい、ウォークスルーで見つかった問題点を改善し、再度別の被験者にてウォークスルーを実施する。

■有効性：

ターゲットユーザーのペルソナに則したシナリオを作成し、ユーザーを中心に考えるため、ターゲットユーザーに特化したサービス、デバイスに依存しないサービスの検討が可能となる。

シナリオ・ストーリーボード・ワイヤフレーム作成にて情報を外化（見える化）することで、チームメンバー間の共通認識ができるため、スムーズに議論を進めることができる。

ウォークスルーにて、前段階で気付かなかった事について新たな気づきを得ることができる。今回はシナリオ作成から開始したが、ワイヤフレーム作成とウォークスルーのみでも充分新たな気づきを得ることが可能だと感じた。

■留意点：

ウォークスルーで実施した思考発話は、被験者の慣れが必要である。声を出しながら操作するという事は通常利用時は実施しないため、最初は意識して発話しないと沈黙してしまう。また、簡易的に本来のユーザーではない人に被験者となってもらい場合も注意が必要であり、開発担当者が実施するとインターフェースの評価をしてしまう傾向がある。

ワイヤフレームの作成時、シナリオからアプリケーションのユーザーインターフェースを考えるにはUI設計の知識が必要である。知識がなければ、どのように表現することが正しいのかが判断できない。

有効なペルソナやシナリオ作成には、前段階のユーザー調査含めて相応の知識と経験が必要となる。よって、実プロジェクトで人間中心設計プロセスを全て実施する場合はかなりの労力と時間を要すると考えられる。

演習では1つのシナリオしか作成しなかったが、半日程度を要した。実プロジェクトでは複数のシナリオがあると想定されるため、作業の効率を高める方法を検討する必要がある。

●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：

多くのアジャイル開発手法で採用されているイテレーション開発について、折り紙の多面体を作成するプロジェクトを題材に疑似体験した。

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した（1 日は 11 分）。

イテレーションの初日はイテレーション計画のみ行い、2 日目からは朝会と作業を繰り返し実施した。作業の進捗管理はタスクボードとバーンダウンチャートを利用した。イテレーションの終了時は KPT（Keep/Problem/Try）による振り返りを行った。

■有効性：

作業を細分化し、短いサイクルで区切りながら開発を進めるため、要求の変更に対しても柔軟に対応できる。

タスクボードを使用することで作業が可視化でき、バーンダウンチャートでプロジェクトの進捗状況を容易に把握することができる。これと併せ、毎日の朝会にてチーム状況をメンバーと共有することで問題点に早く気づき、迅速な対策や改善を行うことができた。開発状況に応じて、タスクの見直しや分担の最適化を行いチームとしてのパフォーマンスを高めていくことができる。

KPT による振り返りも短いサイクルで行うことができ、作業やチームの質が向上していく。

■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が大きな課題となる。

●第 6 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構築手法の一つである CoBRA (Cost estimation, Benchmarking and Risk Assessment) 法をテーマとして、工数見積りの考え方、見積りモデル構築手順や活用時の留意点などを学んだ。

プロジェクト開発の中で工数と開発規模の 2 つの因子間だけで説明しきれない要因を熟練者の知識や経験をもとに”コスト変動要因”として定量化し見積りモデルの中に組み込むことで、説明性の高い見積りモデルの構築が可能となる。

講義で CoBRA 法の概要を理解した後、演習にてコスト変動要因を内容や影響度合いを自分達で定義するまでのプロセスを体験した。IPA で公開されているツールを利用して、見積りモデルを構築するプロセスを体験した。

■有効性：

数十件の過去 PJ データでモデル作成が可能になるため、導入にかかる初期コストが少ない。

また、熟練者の経験や勘を定量化しモデルに組み込むことで、見積り根拠の見える化が可能となる。見える化により、工数に影響度の高い要因が把握できるようになり、効果的な対応策の導入が可能になる。

■留意点：

モデル作成にあたり、以下の点が留意点となる。

規模の測定方法・単位が組織内で統一されていることが前提となる。モデルに複数の測定単位が混ざると、作成したモデルの信頼性が欠如する

変動要因を初期の段階で抜け漏れなく挙げる

影響度レベル設定は、定性的な表現だと個人の感覚に依存してしまうので、定量的かつ、測定可能な表現にすることが望ましい

モデル運用の際には、計画見直し時や要件確定時などにメンテナンスを行い、見積りの精度を上げていく。

●第7回（例会）：テスト演習：猪塚 修（横河ソリューションサービス株式会社）

■概要：

ソフトウェアテストを実際に行う事によって、緻密さが必要なことを理解したうえで、単体テストの基本的なやり方や、テスト技法についての知識を習得し、テストケース設計の重要性を学んだ。演習では Myer' s の三角形という問題を元に、テストケースを実際に考える演習問題に取り組んだ。各々の回答を元に、グループ内でテストケース抽出の経緯などを議論した。その上で、単体テストの設計、レビュー、準備、実施、完了までの流れを詳細に学んだ。最後にテスト技法として、ホワイトボックステスト、ブラックボックステスト、デシジョンテーブル、同値と境界値、パステストについて、演習問題に取り組んだ。

■有効性：

ソフトウェアの品質を考える上で、テストを行うことは必要不可欠だと考える。今回の演習問題のように、実際にテストケースを考え、グループ内で議論を重ねることにより、新たな観点や、考え方に触れることができた。その上でテスト技法を学ぶことによって、テストケース設計を行う際に、より少ないケース数で、よりバグが見つかり、漏れや重複の無い、良いテストケースを考えることができる。

■留意点：

より多くのテストを行う事で、より高い品質のソフトウェアを作ることができた。しかし、テストを行うコストも相対的に高くなるので、求める品質を実現するために必要なテストケースとは何かを考える事が重要であると考えられる。

●第 8 回 (例会): 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)

■概要:

メトリクスの基礎と Goal-Question-Metric (GQM) 法について講義で学んだ後, GQM 法を用いた演習を実施した。

メトリクスは測定の方法と尺度を表したものであり, プロダクトあるいはプロセスの品質を定量的に評価するために有効なものである。ただし, メトリクスはあくまで手段であるため, 測定の目的を明確にしたうえで, その目的に沿って使用することが重要である。

GQM 法はまず「目標(Goal)」を設定し, その達成を評価するための「質問(Question)」, 質問に回答するために必要なデータを得るための「メトリクス(Metrics)」をトップダウンに設定する手法である。

<実施内容>

1. コード行数の測定 (個人演習)

メトリクスの演習として, スライド 1 枚に収まる程度の少量のコードについて, 各個人で行数を測定し, それぞれ結果を発表した。これにより, 対象が少量のコードであっても測定の目的によって測定方法が変わること, それにともなって測定結果も変わることよく理解できた。

2. GQM グラフ作成 (グループ演習)

2 チームに分かれてそれぞれのチームでテーマを設定し, GQM グラフを作成した。演習では「質問」導出にあたって仮定した事柄を「仮定」として設定する点も含め実施した。「目標」を起点とすることで「目標」に対して必要なメトリクスを設定することができることを理解した。

■有効性:

抽象的かつ複雑なプロダクトであるソフトウェアの開発において, その品質を評価するためには, メトリクスによる定量的な分析が欠かせないものとなっている。

GQM 法においては, 目標を明確に据えたうえで, それに対応したメトリクスを設定する枠組みが提供されているため, 目標と整合したメトリクスの設定・計測を実施することができる。

■留意点:

メトリクスによる定量的な分析は不可欠ではあるが, あくまで一つの側面でしかないことに注意する必要がある。一つのメトリクスに頼らず多面的に測定することと, 最後は現物の確認が重要であること忘れてはいけない。

メトリクスはあくまで目標達成の手段であるが, メトリクス達成自体が目的化してしまい, 本来の目標を見失う「負のホーン効果」が発生することがある。これを防ぐためには GQM 法等により目標とメトリクスの関係を明確にすることが有効である。

●第9回（臨時会）：要求工学（要求分析）：中谷 多哉子氏（放送大学）

■概要：

システム開発では、ソフトウェアやシステムに求められる発注者（オーナー）の要求を正しくとらえ抽出し、機能要件や非機能要件として要求仕様書に定義していく。要求はオーナーの価値観や置かれている背景に基づくものであるため、開発者側では把握しにくい。また、要求は状況に応じて変化するため不安定でもある。このため、要求定義にはコミュニケーションを伴い、要求分析が必要となってくる。

本コースでは、要求分析の手法として、リッチピクチャを用いた分析、CATWOE 分析、ゴールモデル分析について演習を通じて学んだ。

1. リッチピクチャを用いた分析

リッチピクチャとは、システム化に伴う関係者（利用者、オーナー、関連機関等）が持つ意見を分析するために、絵（漫画）と吹き出しの文字で描くものである。リッチピクチャを作成し俯瞰することで、関係者が持つ意見や問題意識等の現状把握を直感的に行うことができる。

2. CATWOE 分析

CATWOE 分析とは、システム化に伴う関係者が持つ「真の要求」を明らかにするために、要求がどのような根拠（価値観、世界観）に基づいているのかを次の6つの要素（※）で分析する手法である。

※Customer, Actor, Transformation-process, World-view, Owners, Environment

例えば、観光スポットから最寄り商店街へ観光客の流れを取り込む仕組み作り（要求）を検討する事例において、「観光客が観光スポットから途中迷わずに最寄りの有名店で食事をしたいと望んでおり、シャトルバスで移動する仕組みを提供する」場合のCATWOEは、「C（観光客）、A（シャトルバス）、T（有名店までのルートが不明という状況が、シャトルバスで迷わずスムーズに移動できるという状況になる）、W（速く楽に目的地に移動したい）、O（観光客）、E（シャトルバスが通るルートやバス停の整備ができてい）」となる。

3. ゴールモデル分析

ゴールモデル分析とは、目的（ゴール）の達成のために必要となるタスク（要求に当たる）や、そのタスクを実施するための前提条件となるサブゴールを、木（分岐図）を用いてトップダウンで分解して洗い出す手法である。

■有効性：

真の要求はオーナーの世界観に基づくものであり、開発者側で知ることは容易ではないが、要求分析手法を使うことで明らかにすることができる。また、要求分析手法で可視化したアウトプット（リッチピクチャ等）を用いて、オーナーや関係者と共有し、確認できる利点もある。

■留意点：

要求分析手法は有効性も多いが、作業に時間がかかり、多面的に分析するには複数人必要となる。要求分析手法を全ての開発にそのまま適用するのはロードがかかるため、必要に応じて適用したり、要求分析手法の考え方を頭に置いて簡易的な方法で代替したりするなどの柔軟性も必要だと考える。

以上