

演習コース「ソフトウェア工学の基礎」 2014 年度 活動報告

Report on Practice Course of Software Engineering Foundations in 2014

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 : 藤原 聡子 (株式会社東芝)
村上 淳 (NECソリューションイノベータ株式会社)
中村 壮志 (株式会社デンソー)
市川 勝規 (日本電子株式会社)
染谷 知宏 (テックスエンジソリューションズ株式会社)
岩村 義明 (アズビル株式会社)
山本 真成 (東京海上日動システムズ株式会社)
仲野 恭平 (伊藤忠テクノ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 10 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 2005-2014, 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]を参照されたい)。そこで

演習コース I 「ソフトウェア工学の基礎」

2014 年度も引き続き、産学両面に通じた講師をお招きし、計 8 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 SWEBOOK およびソフトウェア品質知識体系 SQuBOK 上で、2014 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）。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
	プロセスアサセスマ 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>	
	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 <i>品質・レビュー、ア ジャイル</i>	レビュー <i>アーキテクチャ</i>
	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</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		運用・保守のマ ネジメント	運用・保守

3. 各演習における気づきと活用状況

演習コース I 「ソフトウェア工学の基礎」

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 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。2014 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 2014 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) 2014/5/9 :

レビュー演習 :

猪塚 修氏 (横河ソリューションサービス株式会社)

●第 2 回(例会) 2014/6/13 :

オブジェクト指向モデリング :

井上 樹氏(豆蔵)

●第 3 回(合宿) 2014/7/17 :

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) 2014/8/19 :

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) 2014/10/10 :

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 6 回 2014/11/7 :

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 7 回(例会) 2014/12/20 :

メトリクスと GQM :

鷺崎 弘宜 (早稲田大学/国立情報学研究所)

●第 8 回(例会) 2015/1/16 :

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 9 回(臨時会) 2015/2/13 :

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)

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

演習コース I 「ソフトウェア工学の基礎」

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.

演習コース I 「ソフトウェア工学の基礎」

[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。

演習コース I 「ソフトウェア工学の基礎」

● 第 1 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

レビューにおける目的や定義について学び、品質特性を使ったレビューやシナリオベースのレビューについて演習を行った。

演習では、各自ポストイットに記載した内容を模造紙に説明しながら貼り付け、他の人の視点や考え方について学ぶことを大切に、レビューの視点(勘所)を増やすことを目的とした。また、レビューを受ける側(レビューイ)についても焦点を当て、効率的にレビューを受けるためのノウハウについて学んだ。

■ 有効性

レビューは、単体/結合/総合テストより早い段階でソフトウェアの欠陥を除去できる最も効率的な方法である。レビューの効果を高めるには、レビューの目的の明確にすることをはじめ、レビュー実施者(レビューア)やレビュー内容について重点を絞ることが重要である。

レビューイについては、チェックシートを有効活用して成果物について品質向上を高めレビューを受けるための準備(心構え)が重要である。

■ 留意点

- ・レビューに使える時間は限られているため、レビューア・レビューイともに効果的なレビューを実施しようとするための準備や姿勢が重要となる。

- ・レビュー内容について重点を絞らない場合、レビューア(開発担当者、運用担当者、お客様など)により、それぞれ着目する視点が異なるため、レビュー内容について重点を絞った場合と比べると指摘件数は多く出されるが、致命的(クリティカル)な指摘事項が出ない可能性がある。

- ・レビューイは、チェックシートのチェック項目に対して形式的な確認とならないよう、なぜそのチェック項目があるのか考えることが大切である。

演習コース I 「ソフトウェア工学の基礎」

●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：

本演習の目的は、オブジェクト指向分析設計の基盤である「モデリング」手法について、ソフトウェア開発現場での活用方法を、講義と演習を通じて習得する事である。

本演習で使用した「モデリング」手法には、「要求のモデリング」と「設計のモデリング」があり、それぞれの概要は以下の通りである。

(1) 要求のモデリング

要件定義工程(要望→要求→要件)のうち、(要求→要件)で利用するモデリング手法である。要求から要件を「抜け漏れを防ぎ」「わかり易い形で」記述する事ができる。

要求のモデリングのうち、「ユースケース図」「ステートマシン図」について、ストップウォッチを題材に、演習課題の形式で学んだ。

(2) 設計のモデリング

コーディングの前の、詳細設計工程で使用するモデリング手法である。ソースコードに比べ、「レビューがし易い」「コーディング前の問題発見がし易い」という特徴がある。

設計のモデリングのうち、「クラス図」「シーケンス図」について、自動ドアを題材に、演習課題の形式で学んだ。

■有効性：

要求、設計のモデリング手法は、「大規模開発」「立場の異なるステークホルダとの合意形成」「レビュー」など、関係者が多いソフトウェア開発の現場において、その「視覚効果の高さ」が共通認識形成の面で有効にはたらくと期待できる。

また、複数のモデリング手法を組み合わせることで、多角的な問題発見に繋がり、手戻りによるコスト増大の抑制が可能である。

設計のモデリングについては、モデリングに加えて自動生成手法(MDA)を導入すると、設計時のコストは増大するが、「設計の見える化」と「製造工程の簡略化」の効果が期待できる。

■留意点：

(従来実施しなかった)モデリング手法の導入の為にコストが必要となる事、モデリング手法導入による効果が見えるには年単位の期間が必要な事から、大規模開発に即導入するのではなく、小規模開発から導入を行い、効果を確認した後に徐々に広める事が望ましい。

演習コース I 「ソフトウェア工学の基礎」

第 3 回 (合宿) : アーキテクチャ設計・評価 : 長谷川 裕一郎 (合同会社 Starlight & Storm)

■概要 : 本講座では、「ドーナツ店の起業」をテーマにアーキテクチャの設計と分析/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構築されるシステムの土台である。目標とする品質特性を明らかにして、目標を満たすアーキテクチャを設計する必要がある。アーキテクチャの設計手法の一つとして ADD (Attribute Driven Design) を学習した。ADD では非機能要件と品質特性に注目して、まず品質特性シナリオの作成を行い、作成した品質特性シナリオからもっと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、選択した品質特性シナリオに対し実現方法を割り当て、アーキテクチャを設計する。

アーキテクチャの分析・評価の手法として、ATAM (Architecture Tradeoff Analysis Method) を学んだ。ATM は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否かを評価する。ATM によって、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なる。

■有効性 :

ADD を用いてアーキテクチャを適切に設計することで品質の実現が可能となる。また、ATM を用いてアーキテクチャを分析/評価することによって実装やメンテナンスで問題が発生するリスクを緩和することができる。

■留意点 :

品質特性シナリオはもっとも脅威となる破滅のシナリオを考え、それを防ぐためのシナリオを中心に考える。また、品質特性シナリオは様々なステークホルダの観点から見た場合、抜けているシナリオがある場合がある。そのため何度かシナリオの見直しを行ったほうがよい。

トレードオフポイントでは、どの品質特性に重きを置くかによって判断する。0 対 100 で判断するのではなくバランスを見て決めるのが現実的である。

演習コース I 「ソフトウェア工学の基礎」

●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、実際の開発を行う前にできるだけ実物に近い形でコンセプトをシミュレーションして、問題点や使いにくさを早期に発見・修正できるものである。本講義では、既存の様々な手法の概要を学びながら、主には演習を通して手法のイメージを掴んだ。演習では「モバイルによるクッキングレシピサービスの提案」をテーマに、チームに分かれて下記手順で作業・議論をした。

<演習手順>

1. ペルソナとアクティビティシナリオの作成 (1時間)

1.1 仮想のペルソナの作成

名前や家族構成、性格、料理のきっかけ、調理の頻度や料理の目的、料理をする際の傾向、情報への接し方といった多数の観点で仮想的に具体的にペルソナを設定する。

1.2 バリュースナリオの作成

ペルソナが望んでいる典型的なシナリオを想定し、具体的に様々なシーンを挙げる。

1.3 アクティビティシナリオの作成

1.2のうち1つのシーンを選択して、物語風に具体的にシナリオを書き出し(アクティビティシナリオ)、そのシナリオ中にすべきユーザの行動(タスク)を洗い出す。

1.4 インタラクションシナリオの作成

1.3のタスクに対し、実装に依存したユーザの作業を書き出す。

2. ストーリーボードとワイヤーフレーム作成 (1時間)

シナリオをステップに区切って、そのステップごとに、アクティビティとワイヤーフレームを用紙に書き上げる。アクティビティには、シナリオとユーザのタスクを併記し、そのステップの状況に応じたワイヤーフレームになっているかを3.で検証する。

3. ストーリーボーディング(思考発話法)での評価 (1時間)

演習では、被験者・司会・記録係の3人の役割を設けた。被験者は、ストーリーボードに沿ってワイヤーフレームのインタフェースを操作する。被験者は思ったことを口に出しながら試行評価し、記録係が記録する。記録係は、被験者の手が止まった部分・迷った点を付箋にメモし、該当箇所付近に貼り付ける。司会は、被験者が考え込んでいる場合や声を出していない場合に「何に迷っていますか」「声に出してください」と注意を促す。最終的に、挙げられた付箋を見て、問題点を共有・発見し、改善ポイントを議論する。

■有効性：

- ・ペルソナやシナリオを具体的に設定しコンセプトを明確にすることでイメージ共有できて新たな発見や問題点が見えやすくなる
- ・「検索する」「調査する」といった抽象的な用語を用いて誤魔化していた箇所や、出来たと思っていた箇所に対して、第三者が評価することで早期に問題発見できる
- ・被験者主体で作業してもらうこと、その場に同席することで、問題点に気づきやすい

■留意点：

- ・ペルソナをとっても具体的に設定することが必要である。演習においても具体的に設定したつもりが、まだ足りなかった。無難に抽象的な設定にすると、後の作業で新たな気づきを得にくい。また、中途半端なペルソナでは、方針がぶれる・都合のいい解釈をしてしまうことがあった。ペルソナの設定次第でその後の方針に大きな影響を与えるため、ペルソナの設定自体がととても重要である。
- ・アクティビティシナリオ作成の際も、開発予定のシステムを使う前後だけでなく、その状況を想像しやすいようにととても具体的にシナリオを作っておくとイメージ共有しやすい。
- ・演習にも関わらず時間がととても掛かった印象であり、実際の現場ではさらに負荷が掛かると思われる。慎重に適切に、かつ、効率的に作業を進める手法も合わせて習得したい。

演習コース I 「ソフトウェア工学の基礎」

● 第 5 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

折り紙の多面体を作成する模擬プロジェクトを通して、アジャイル開発の進め方を体験した。

2つのチームに分かれ、各チーム1名はプロダクトオーナー、残りの3名は開発者役とした。

プロダクトオーナーはどの形状の多面体をいくつ作るかを決定し、開発者は2回のイテレーションで多面体を作成した。

1日を11分として、1イテレーションを4日間とした。各イテレーションの初日にはイテレーション計画の作成、イテレーション終了ごとにKPTによる振り返りを実施した。各日の始めには朝会を実施した。

■ 有効性

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チームメンバーの作業負荷を可視化できた。これと併せ、毎日の朝会にてチームの状況をメンバーと共有することで問題点に早く気づき、迅速に対策を打つこと、改善をかけることができた。

タスクの粒度を揃えること、開発状況に応じてタスクをこなすことでチームメンバーの負荷が平準化され、プロジェクトの生産性を高めることができた。

■ 留意点

アジャイル開発は小規模プロジェクトでは効果を出しやすい。大規模プロジェクトにアジャイル開発を適用する場合、チーム編成、チーム間および顧客とのコミュニケーションのとり方が大きな課題となる。

演習コース I 「ソフトウェア工学の基礎」

●第 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:

システム開発において重要とされる要求工学について、ステークホルダ分析・要求抽出を中心に、主に演習を通して手法の理解を深めた。特に、価値観や関心の持ち方によって見える世界が異なることに留意し、ステークホルダとその立場での価値観を広く洗い出したうえで慎重に実現手段を考えていくべきと学んだ。演習では「押上地区観光客誘導システム」を課題に要求抽出を体験した。演習の流れの概要は以下のとおり。

- (1) リッチピクチャを用いた現状分析により、関係者や意見を洗い出す
- (2) 役割依存モデルを作成する

今回の演習では簡略化し、リッチピクチャ上に挙げられた人のうち、Actor・Customer・Owner(後述)の候補、重要な課題を検討した。

- (3) CATWOE分析を行う

Ownerを設定しCATWOE分析を行う。

CATWOEは、C(Customer: サービスを受ける人, 受益者, 犠牲者), A(Actor: 遂行者), T(Transformation process: 望ましい状況から望ましい状況へ変換するプロセス), W(World view: Ownerの世界観, 信念, 根拠), O(Owner: Tを推進する人, 決定者), E(Environment: 望ましい状況を達成するために必要な環境と資源, 守るべき制約とルール)から成る。

- (4) 所有者別ゴールモデルの作成, KAOS(Knowledge Acquisition in autOated Specification)の適用

Ownerの望みをトップゴールとしてsubゴールへ分解し、対象システムに求められる機能やゴールを達成するための手段を見つけていく。その際、ゴールを満たすAnd条件・Or条件を考えながら責任の割り当てを検討し、システムの満たすべき責務の明確化をする

■有効性:

- (1) リッチピクチャについて

絵を描いてみることで視点が広がり、多くの価値観・要求を挙げる事ができた。例えば、ステークホルダは導入先の会社/部門という単純な想定だけではなく、押上地区の地元老舗の店主や道路を行き交う人々等を見過ごさずに広く挙げる事ができた。

- (3) CATWOE分析について

A, C, Oを切り替えながら考えることで、より要望・要求を明確化しやすく、抜け漏れに気づきやすい。ただし初心者は、A, C, Oをうまく切り替えられず訓練が必要である。

- (4) KAOSについて

- ・ステークホルダを広く分析したうえでKAOSの分析をすると、実現手段の妥当性の確認をしやすい。例えば、1つのシナリオを選択して分析した結果が妥当だと思えても、別のCustomerにとっては不満に感じる場合がある場合がある

- ・実現すべき役割の一部を**部門/**システムの責任に任せる、といった役割分担を設定することができるため、システムの境界を明確にしやすい

- ・AND条件, OR条件を考えながら分解するため、代替手段を検討する等を通して最適な手段へ落としこみやすい

■留意点:

- ・どんな手法でも言えることだが、顧客の状況をどれだけ知っているか・細かく想定できるかが、顧客の価値観に気付けるか・要求を漏れなく抽出できるかに大きく影響する。

- ・Ownerを具体的に設定することでアイデアが生まれやすい。

- ・価値観や要求を発散することはできたが、それらを集約する一連の手法やノウハウは別途、習得する必要がある

- ・実践するには、重要なものを見定めるために十分な事前の情報収集や十分な分析をするための時間を十分に確保することが重要である

演習コース I 「ソフトウェア工学の基礎」

●第7回：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：

メトリクスの基礎と Goal-Question-Metric (GQM) 法を学んだ後、GQM 法を用いた演習に取り組んだ。

メトリクスとは品質特性などを測定する際の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。第7回例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、欠陥密度(*1)などであった。一方、プロセスメトリクスは、欠陥密度(*1)、欠陥除去率などを学んだ。

GQM 法とは、設定した目標に対して必要なメトリクスを「目標(Goal)」、「質問(Question)」、「メトリクス(Metrics)」の順にトップダウンで決定する手法である。GQM 法を拡張し、ゴールの更に上位に「ビジネスゴール」、ゴール、質問、メトリクスの間に「仮定」、「解釈」等の項目を置くことでゴールと測定するメトリクスの対応付けを補完できる。

(*1) 欠陥密度は、プロダクトとプロセス両方の観点から活用できるメトリクスである。

■有効性：

メトリクスは、ソフトウェア開発の管理や、品質の把握に欠かせない存在である。測定したメトリクスをヒストグラムや散布図などで可視化することで、作ったソフトウェアの異常値の発見、事象の関係性の把握が出来る。

GQM 法は、目標とメトリクスの整合性や妥当性を明確化することで、測定自体が目的となることを防ぐ。目標に対する質問の導出が難しい場合、目標が成り立つ、或いは成り立たない仮定を設定すると、質問の導出がやり易くなる。GQM の中で解釈(条件)を設定した場合、目標に対するメトリクスの測定基準は明確になるが、解釈に対しての閾値の正確な設定が難しい。閾値に関しては、最初は大きい、小さいなどの大まかな設定にし、データの蓄積や精査を行った後に、精密な設定値を設ければ良い。

■留意点：

メトリクスを用いる際は、まず、組織内における定義を統一しておくことが大事である。例えば、「1件」、「1行」などの定義は人によって様々である。そのため、定義を統一しない状態でメトリクスを用いると、測定結果が不正確になり、誤った基準値を設定してしまう等のリスクが高くなる。

次に、測定の目標を明確にしないと測定自体が目的となり、測定結果が活用されず、現場に「やらされ感」しか与えない可能性がある。また、測定メトリクスを達成させることが目的となり、本来の測定目標を見失う「負のホーソン効果」が起こる可能性もある。GQM 法等を活用することで目標とメトリクスの関係を明確化し、このような事象を防ぐことができる。

演習コース I 「ソフトウェア工学の基礎」

第8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：

ソフトウェアテスト技法の基礎を学ぶと共にテスト技法の使い方のコツや指導方法を身につける事を目的とし、5問の演習問題に取り組んだ。演習では個人演習で導いたそれぞれの回答を基にグループで検討を行い、テストデータ抽出の経緯を議論した。また、制御パステスト、同値分割、境界値分析、デシジョンテーブルなどのテスト技法の使い方について学んだ。

■有効性：

制御フローパステスト、同値分割、境界値（限界値）テスト、デシジョンテーブルなどのテスト技法を使用する事により、テストの内容が明確になり説明が容易になる。また、テストデータを分別することで必要最低限なテストに絞る事が出来る為、コストの削減につながる。

■留意点：

テスト技法は、事前に演習等を実施して使用方法や特性を把握しておく必要がある。さまざまなテスト技法が存在しているがそれぞれ検出できる不具合が異なる為、使用する場合にはどのような不具合を見つきたいのかを明確にして、それに合うテスト技法を選択する必要がある。また、テスト技法によりテストケース数を削減することにより全網羅で無くなる事による漏れのリスクもあるが、コスト削減のメリットもある為、リスクとコストを考慮してあらかじめ網羅率を決めておく。

演習コース I 「ソフトウェア工学の基礎」

- 第 9 回(臨時会)：工数見積りモデルの構築手法 (C o B R A 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である「C o B R A 法」を学んだ。本手法で見積りモデルを構築し、活用する事でプロジェクト工数を求めることができる。経験ベースの見積もりと、データから導き出す見積りみの混合型である「C o B R A 法」は、見積りみに精通したメンバ 2～3 名の協力と、過去の見積り事例 10 件程度で、見積りモデルを構築することができる。見積り精通メンバの「勘 (経験則)」と見積り事例のデータから構築された見積りモデルをツール化することで属人性が排除され、誰でもプロジェクトの工数を求めることができる。

■有効性：

「C o B R A 法」のツールが、I P A より提供されているため、見積りモデルをツール化し誰でも見積りを行うことができる。作業積み上げやWBSなどが存在しない時点の見積り (概算見積りや予備見積り等) 工程で活用するのが望ましく、モデルの見直し (改善) を行うことで、「勘 (経験則)」による見積り精度の誤差を小さくすることができる (±20%程度)。モデルにより数値化された個人の「勘」や「経験則」は、見積りみの妥当性の説明する際に有効である。また、見積りモデルを構築する過程で検討される「変動要因」と「影響度レベル」は、見積りみが増える要因をリスクや課題と捉えることができ、リスク管理やプロセス管理にも有効活用できる。

■留意点：

モデルを活用するにあたり、プロジェクトの規模は「KLOCやSTEP数などで、定量的に目処が付いている」ことが前提となる(プロジェクト規模を算出するモデルではない)。モデルを構築する上で必要となる、工数の変動要因と影響度レベルは、

- ・変動要因は所属組織に合わせて検討する。
- ・変動要因は見積り時点で使用可能な要因を検討する。
- ・影響度レベルは、定量的に測ることができるように設定する。
- ・変動数値が均一になるように影響度レベルを設定する。

等、検討や設定が難しい面がある。また、様々なプロジェクト特性を取り込んでしまうとモデルの精度が上がらないため、全社共通モデル等と呼ばれるような大きなモデルは推奨できない。プロジェクト特性が同じ組織内でモデルを構築することが望ましい。