

# 演習コース「ソフトウェア工学の基礎」 2013 年度 活動報告

## Report on Practice Course of Software Engineering Foundations in 2013

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サービス株式会社)  
研究員 : 小間 香保里 (アズビル株式会社)  
杉山 浩一 (株式会社リンクレア)  
染原 一仁 (日本電子株式会社)  
佐々木 愛美 (株式会社東芝)  
中村 考宏 (株式会社東京ビジネスソリューション)  
森 哲史 (三菱プレジジョン株式会社)  
斉藤 慶太郎 (東京海上日動システム株式会社)  
新田 佳祐 (株式会社アドバンテスト)  
安部 晃嘉 (株式会社メタテクノ)

### 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 9 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 2005-2013, 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][10][11]を参照されたい）。そこで 2013 年度も引き続いて、産学両面に通じた講師をお招きし、計 9 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、2013 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
	プロセスアセスマ ントのマネジメント	構成管理		要求分析 <small>要求工学・オブジェクト指向・ペーパープロトタイプ</small>
	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 品質・レビュー、アーキテクチャ	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 見積	テストのマネジメ ント テスト・モデル駆動開発	テスト
	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

図 1 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。2013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2013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) 2013/5/10 :  
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) 2013/6/14 :  
アジャイル開発 :  
西村 直人氏 (株式会社永和システムマネジメント)
- 第3回(合宿) 2013/7/11 :  
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(例会) 2013/10/11 :  
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5回(例会) 2013/11/8 :  
オブジェクト指向モデリング :  
井上 樹氏 (豆蔵)
- 第6回(臨時会) 2013/11/29 :  
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) 2013/12/20 :  
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8回(例会) 2014/1/17 :  
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(臨時会) 2014/2/14 :  
モデル駆動開発 :  
久保秋 真氏 (株式会社アフレル)

### 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は、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.

## 付録1：各回のまとめ

### ●第1回(例会)：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）

#### ■概要：

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る。

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ため、ソフトウェア開発プロセスにおいて重要視されている。

本演習では、品質特性（パフォーマンス、セキュリティ、信頼性など）を使ったレビューやシナリオベースのレビューなどからレビューの勘所を養い、グループ演習を通じて他の人の視点を学んだ。

#### ■有効性：

レビューで大事なことは、視点を換えることである。

どの部分（前提の資料との整合性、あいまいな表現、品質特性など）に重点を置いてみるかで指摘が変わる。

また、開発者、発注者、ユーザ、保守、運用担当者、それぞれの視点で見ると要求が変わることもある。

これらポイントを意識してレビューすることで、レビューの効果を高めることができる。

#### ■留意点：

レビューに使える時間は限られている。

レビューを受ける側は、事前に目的やポイントを明確にし、複数のレビューアでポイントを分担してもらうなど、円滑なレビューが実施できるよう努めることが大切である。

●第2回(例会)：アジャイル開発：西村 直人氏（株式会社永和システムマネジメント）

■概要：

アジャイルとはソフトウェア開発手法群の総称である。本演習では、絵画の展示会を運営するプロジェクトを想定したグループ演習を実施した。

演習を通し、アジャイルでよく活用されるスクラムのチームの動き方や、必要となる用語を理解し、アジャイル開発プロセスを体験した。

演習時は、以下をルールとした。

1. 4人で1チームとする
2. 1スプリント（反復型開発プロセスにおける1回の開発期間）を10分とする
3. スプリント終了ごとに振り返りを実施し、スプリントバックログ（次のスプリント内で達成すべきチームのタスクリストのこと）を作成する
4. 上記を4スプリント繰り返し、展示会を実現する

■有効性：

スプリントバックログを作成し、作業を書き出すことにより、完成のイメージと具体的に何をすべきかをメンバ全員で共有することができる。さらにそれらをタスクボードに貼り出すことで、プロジェクトの進捗状況やプロジェクトメンバの作業負荷の「見える化」ができる。

あわせて朝会を実施し、全員で進捗を確認することで、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」ことがメリットとなる。

また、スプリント毎の振り返りではプロダクトオーナー（顧客側の意思決定が出来る人物）にも参加してもらうことにより、要求とのギャップを埋めることや、要求の変更に対しても柔軟に対応でき、プロジェクトの起動修正が早くできる点もメリットといえる。

■留意点：

スクラムチームでは、全員で取り組む姿勢が必要であり、メンバ同士が共通の意識を持たないとうまく機能をしない。共通意識を持つにはメンバ個人が持つ不安や疑問に合わせてアプローチを続け、一体感を作ることが必要となる。

バックログ等の作業管理は作業者自身が行う必要がある。小さく失敗するためには、朝会で自分の管理している作業が予定通りに進んでいないことを伝えることが重要となる。予定通り進んでいないことは、プロジェクトにとってのリスクである。それを1日ごとに確認できる朝会という場合は、アジャイル開発において必要である。

スプリント毎の振り返りでは、要求側の意向を汲み取り反映させた上で、次スプリントで必須の作業を洗い出し、それに対して、より良いやり方を考えることが課題となる。

● 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、「ドーナツ店の企業」というテーマの演習を通じてアーキテクチャの設計・評価を学んだ。アーキテクチャとはビジネス要件やシステム要件から構築されるシステムの土台である。まず、そのベースとなる要件の抽出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。要件が十分に抽出されたのち、ADD (Attribute Driven Design) によりアーキテクチャの設計が行われる。ADD はアーキテクチャドライバの分割・詳細化を行う手法であり、品質特性シナリオと照らし合わせ実現方法を割り当てる。分析・評価の手法としては ATAM (Architecture Tradeoff Analysis Method) を学んだ。ATAM は品質特性から矛盾なくアーキテクチャが設計されているかを評価する。準備、分析、検証、フォローアップの4つのフェーズによって実行され、フェーズはさらにステップに分けて実行される。

■有効性：

アーキテクチャは、システムの土台であり、適切な設計・評価を行うことが求められる。ADDによる設計やATAMによる分析・評価を行うことでシステム全体の高品質化や開発後のメンテナンス性の向上が可能となる。

■留意点：

品質特性シナリオの作成では、「どのような刺激が発生するか」という点からシナリオを作成し、実現方法に関しては触れないようにする。実現方法まで考えてしまうと、非機能要件を十分に抽出できない恐れがある。また、異なる知見を持ったメンバーで検討を行うことで複数の観点からの非機能要件が抽出できる。ATAMを利用した分析・評価では、マニュアルに従ってすべて実行するのではなく、各開発部門の環境に合わせてカスタマイズし実行することが望ましい。これは工数や人員などの問題でマニュアル通りの実行が難しいためである。

●第4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ことを目的とし、いくつかの演習問題に取り組んだ。演習では個人演習・グループでの検討を行い、テストデータ抽出の経緯を発表した。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、技法の使い方についても学んだ。

■有効性：

境界値テストにおけるマトリックスの代わりに2次元グラフを用いる方法やデシジョンテーブルの書き方、ユーザの使用頻度に重きを置く方法など、テストデータ作成のコツを学ぶとともに柔軟な発想が必要であることを肌で感じた。

■留意点：

テストは限られたコストの範囲内で製品の品質を評価する必要がある。そのためより少ないテストケースで多くのバグを発見し、漏れがないようにテスト対象を網羅することが求められる。テスト設計では適切なテスト技法やテストデータの選択を行うとともに、その妥当性については説明責任を果たさなければならない。

●第5回（例会）：オブジェクト指向モデリング：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活かされているのかを講義と演習問題を通して理解することを目的とした。

演習では、ストップウォッチを具定例として、「要求」と「設計」の2つの観点から代表的なUML表記法を用いて理解していき、モデルとコードの関係からモデルベース開発のポイントを押さえていった。以下に本演習で取り組んだUML表記法の概要を記載する。

（1）要求のモデリング

- ・ユースケース図：ある場面でのシステムとアクタ（利用者・関係者・外部システム）との相互作用を表したもので、ユーザの視点からシステムの利用の流れを検証する図。
- ・ステートマシン図：ある対象が生成されてから消滅するまでの間の状態の遷移図を示した図。

（2）設計のモデリング

- ・シーケンス図：オブジェクトがメッセージのやり取りをして処理を進めていく様子を時間の流れに添って表現した図。

■有効性：

以下に要求・設計の各々のモデリングのメリットを記載する。

（1）要求のモデリングのメリット

何を要求されているのかをわかりやすい形で表現するため、完成したシステムのイメージを共有し、要求の不備発見を促進することが期待できる。

（2）設計のモデリングのメリット

ソフトウェアの構造が見える化するため、ソースコードを書く前に問題の発見が可能。また、ソースコードで読み取れる範囲よりも広い範囲で設計を俯瞰できる。

モデルを中心とした開発を実施することで、手戻りを削減し、自動生成ツール（MDA）を利用することで、コーディング工数も削減が可能となる。

■留意点：

システム要件定義や設計の工程では、モデルベース開発の導入によるメリットが受けられることが期待される。

ソフトウェア機能設計や詳細設計、実装などの工程では、モデルベース開発を導入したことによる（従来プロセスでは実施しなかった）工数がかかるため、設計品質が向上し、結果として手戻りが減少する工数と、どちらの比重が大きいかはケースバイケースである。

●第6回(臨時会):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の構築手法 CoBRA 法 (Cost estimation Benchmarking and Risk Assessment) を用いて、工数見積りモデルの構築手法を学んだ。ドイツのフ라운ホーファー協会 IESE (実験ソフトウェア工学研究所) で開発された CoBRA 法は、少数の実績データと、見積り熟練者の知識、経験、勘を「コスト変動要因」として組み合わせて、可視性の高い見積りモデルを構築する手法である。

演習においては、IPA (独立行政法人 情報処理推進機構) にて公開されている「CoBRA 法に基づく見積り支援ツール」を利用して CoBRA 法を体験し、工数見積りモデルを構築する流れを理解した。

■有効性:

CoBRA 法は、10 件程度のプロジェクト実績と、3 名程度の見積り熟練者により初期モデルが作成可能であるため、容易に初期モデルを構築できる特徴を持つ。

新たに発生した変動要因を継続的にモデルに取り込み、モデル再構築・見直しを行うことにより、類似プロジェクトの見積りは、変動要因と影響度レベルを設定するだけで、工数・コストオーバーヘッド (変動要因により発生するコスト) の予測を容易に行うことができる。

■留意点:

コスト変動要因と、各変動要因がコストへ与える影響度レベルの設定が、コストオーバーヘッドを算出する要因となるため、様々な視点から変動要因を洗い出す必要がある。

生産性とコストオーバーヘッドの順位を分析し、過去プロジェクトと相対誤差が 20% 以内に収めることにより、コスト変動要因と影響度レベルの正当性を担保され、継続的に精度の高い見積りモデルとなる。

規模により変動要因が大きく影響するため、ある程度の規模のプロジェクトを対象として、過去プロジェクト事例は規模の粒度を合わせることを推奨される。

新しい変動要因が無いかを見積り時に検討し、継続的に分析を行うことにより、高精度かつ漏れの無いモデルを構築し続けることが重要となる。

●第7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：

本講座では、要求工学の中でも特に要求抽出に焦点をあて、要求工学がシステム開発において重要な技術であることを学んだ。演習では、「押上地区観光客誘導システムの提案」を題材として、要求抽出を実践するための手法であるリッチピクチャ、役割依存モデル、CATWOE分析、ゴールモデルを用いて、要求抽出の一連の流れを体験した。

■有効性：

「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」という考えから、要求者の世界観を把握し、不安定な要求の共通性と可変性を見極めて設計することで、要求変更に対するリスクを低減することができる。要求の共通性と可変性を見極めるには、各要求者の信念(こうあるべき)の背景を明示し、関係者間の理解を得ることがポイントである。

また、要求抽出の過程で作成したドキュメント(要求の依存関係や正負の貢献関係)は、要求の良し悪しを判断した根拠になるとともに、後に要求変更が発生した時の判断材料としても有用である。

■留意点：

要求工学を実践するための手法は、本講義で取り上げた手法以外にも様々な手法が存在する。手法を適用する際は、まず明らかにしたい対象を定め、それを知るためには何がわかればよいのかを検討する。その上で、目的に応じた手法を適切に選択する必要がある。各種手法の特徴をよく把握した上で選択することが重要である。

●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、開発実施前の仕様書段階でラフなプロトタイプを作り、ユーザビリティ評価を行う手法である。ペーパープロトタイピングを行うことにより、問題点の早期発見につながる。

本演習では、モバイルによるクッキングレシピサービスの提案を課題として、ストーリーボーディングおよび思考発話法を用いたユーザビリティ評価を行った。

手順は以下の通りである。

1. ペルソナとアクティビティシナリオの作成

サービスのユーザを特定するため、ペルソナを定義し、利用状況を分析する。

2. ストーリーボードとワイヤーフレームの作成

アクティビティシナリオをタスクごとに分割して、ストーリーボードとユーザが操作するインターフェース（ワイヤーフレーム）を作成し、並べて掲示する。

3. ストーリーボーディング

被験者がストーリーボードを読み、ワイヤーフレームのインターフェースを操作する。被験者には頭に浮かんだことも発話してもらうように促す。被験者の行動と発言のそれぞれについて気づいたことを、ポストイットに記載しワイヤーフレームに貼付けて可視化する。

■有効性：

製品のプロトタイプだけでなく、作業中の気づきの可視化が、問題点の発見とともに、関係者間での認識の共有にもつながる。また、被験者の行動を注意深く観察し、「なぜ今そのように行動（操作）したのか」を知ることにより、改善案のヒントを得ることができる。

■留意点：

アクティビティシナリオを作成する際は、実装に依存せず抽象的な表現にする。シナリオには、ユーザビリティの定義（①ある特定のユーザが ②特定の利用状況下で ③指定された目標を達成するために用いられる際の ④有効さ、効率およびユーザの満足の度合い）が表現されていることが望ましい。特に開発者は思考が実装に捕われがちであるので、ユーザが本当に求めるアクティビティを見失わないことが大切である。

この手法はユーザのニーズを知るため、ユーザ調査が鍵となる。ある程度の時間が経過してからアンケートやインタビューでは、その人の真意を知ることが難しいため、質問のタイミングや言葉を選ぶなど、工夫が必要である。

●第9回(臨時会)：モデル駆動開発：久保秋 真氏 (アフレル)

■概要：

本講座では、モデル駆動開発について学んだ。

モデル駆動開発(MDD: Model-Driven Development)とは、モデル駆動アーキテクチャ(MDA: Model-Driven Architecture)に基づいた開発手法である。モデル駆動アーキテクチャとは、ソフトウェア開発にモデルを活用する枠組みのことで、ここで言うモデルとは問題解決を図る際に問題解決に必要な関心事だけを抜き出したもののことである。

本講座では、まずモデル駆動開発に関する基本事項について座学にて学んだ。

次に UML で書かれたクラス図を変換規則に基づいて別の書式(簡易メタモデルのインスタンス図、ソースコード)に手作業で変更する演習を行った。この演習を通じてモデル化におけるモデル変換規則の重要性について理解を深めた。

最後にモデル駆動開発ツール"BridgePoint"を使って自動搬送ロボットの設計と実装を行った。この演習で"BridgePoint"の概要とモデルを中心とした設計の進め方について理解を深めた。

■有効性：

MDD の有効性として、実装技術と機能仕様を分離して管理できることがあげられる。この際、実装技術は企業のコア資産として個々のモデルとして管理し、機能仕様についてはモデルの組み合わせによって管理する。このような管理を行うことで、コア資産に大幅な変更を加えることなく多彩な顧客要求に対応できるようになる。

また MDD はモデルを中心とした開発手法なので、モデルに対する変更規則を変えるだけで、ソースコード、設計文書、テストデータ等、様々な成果物を生み出すことが可能になる。

その結果、上記成果物が常に同期された状態になり、成果物間の乖離を防止できる。

さらに変換規則による変換を自動化することにより、上記成果物に関する品質、生産性の向上が実現できる。

■留意点：

MDD で重要なことは変換規則である。あるモデルを別のモデルに変換する際、変換規則が曖昧な状態ならば、属人性を排した安定した品質のモデルを生み出すことができない。またモデル変換の自動化も変更規則が確立されることによって初めて可能になる。

「まずは自動化ありき」が MDD ではなく、「まずは変更規則ありき」が MDD の肝である。

また MDD を実務に取り入れる際には、全ての開発に MDD を適用するのではなく、従来の開発手法を適用する部分と MDD を適用する部分に整理して適用することが重要である。

以上