

## ソフトウェア工学演習コース 2006 年度 活動報告

### Report on Software Engineering Practice Course in 2006

ソフトウェア工学演習コース 研究員・主査・副査一同

鷲崎弘宜(主査, 国立情報学研究所) 猪塚修(副査, 横河制御エンジニアリング株式会社)  
田村一賢(東芝ソリューション株式会社) 濱正知美(東京海上日動システムズ株式会社)  
麓博之(ジャパンシステム株式会社) 東出智成(横河制御エンジニアリング株式会社)

我々は昨年度に引き続いて「ソフトウェア工学演習コース」を設置し、演習と議論を通じて実践的かつ先進的な種々の代表的ソフトウェア工学手法を学習した。本稿では、その主な狙い、例会での演習概要、臨時会の議論に端を発し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。

The software engineering practice course was opened in this year. This article reports on the primary aims of this course, summaries of each practice in regular meeting, and problem recognition about software engineering technique application in practical contexts.

#### 1. 演習コースの狙いと設置背景

ソフトウェア工学 (Software Engineering) とは、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 [1] であり、ソフトウェア製品を作り運用/保守するための理論、原則、技術のまとまりと認識することができる。現在、ソフトウェアが大規模・複雑化する一方で、品質と納期に対する要求は厳しさを増している。この困難な状況に対処するために、ソフトウェア工学を基盤とした「ものづくり」技術を適切に適用することが求められている。

しかし、ソフトウェア工学領域では古くより指摘される「産業 - アカデミア・ギャップ」[2] と呼ばれる問題があり、特に日本において深刻である [3]。産業とアカデミアそれぞれの立場を以下にまとめる。

- 産業: 多くのソフトウェア産業技術者は、実務におけるソフトウェア開発の豊富な経験を持つ。しかし、ソフトウェア工学を体系的に学んだ経験が十分でなく、最新の技術動向を把握している人も多くはない。組織のソフトウェア開発力を強化するためには、技術者がソフトウェア工学に関する確固たる基盤技術を持ち、最新技術の適用を試みる取り組みが求められる。その基盤の上に、問題領域ごとの専門性や付加価値を高めていくことが重要である。
- アカデミア: 多数のソフトウェア工学研究者が存在し、国際的な連携の中で種々の基盤的なソフトウェア工学技術の研究と発展を促している。しかし、研究者は実務(あるいは実務に近い文脈)における産業ソフトウェア開発の経験が十分ではなく、「大学でのソフトウェア工学研究・教育が実践的でなく、役立たない」という批判がある [4]。従って、研究成果を産業技術者に伝達し、その結果を共に基盤技術の発展に反映させる取り組みが不可欠である。

このような問題意識から、本コースは 2005 年度に初めて設置され、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た。そこで 2006 年度も引き続いて、産学両面に通じた講師を招き、実践的かつ先進的なソフトウェア工学技術に関する講義と演習を企画した。本コースは、産業界に対して主要かつ最新のソフトウェア工学技術/手法の体系的学習機会を提供し、同時に、アカデミア出身の講師に産業技術者による研究成果を含む工学手法を伝達する機会を提供することにより、上述のギャップの克服に寄与したと考えられる。

本稿では以降において、本コースの構成と目標の達成度合い、例会における演習概要、および、演習

を通じて得られた「気づき」を報告する。なお、以下の報告は、受講者の分担執筆による貢献が大きい。

## 2. コースの全体構成と評価

### 2.1 コース設計

主査・副主査は関係者の助力を得て、以下を基本方針としてコース全体を設計した。

- ソフトウェア開発の職業人として必要な種々の知識領域について、偏りなく基本的な理論・原則および応用知識を学習できること。
- ソフトウェアライフサイクルプロセスのうちの開発プロセスを中心として、要求からテストに至る流れに沿って主要かつ最新の工学的手法を学習できること。

設計の結果、次の 9 つの技術を取り上げることが決定した：要求分析、品質要求定義、オブジェクト指向分析設計、(アーキテクチャ)レビュー、ソフトウェアパターン、モデル駆動開発、テスト、形式手法、組込みソフトウェア開発プロセス構築。

各技術を、SWEBOK (Software Engineering Body of Knowledge) [1]における副知識領域に対応付けた結果を図 1,2 に示す(ただし両図における SWEBOK のバージョンは 0.95 である)。SWEBOK は、職業人としてのソフトウェア工学技術者に必要な知識体系をまとめたものであり、本コースによって得られる知識の整理および知識体系全体における位置づけの確認に適している。図 1,2 より本コースは SWEBOK の領域を、知識領域のレベルで包括的に取り上げることが成功したと考えられる。さらに副知識領域のレベルでは、個別の演習内容が各知識領域における代表的な技術/手法に焦点をあてたものであるため、特定の領域を重点的に取り上げたと考えられる。演習で扱わなかった副知識領域については、受講者各自が他の学習機会を通じて扱うことが期待される。

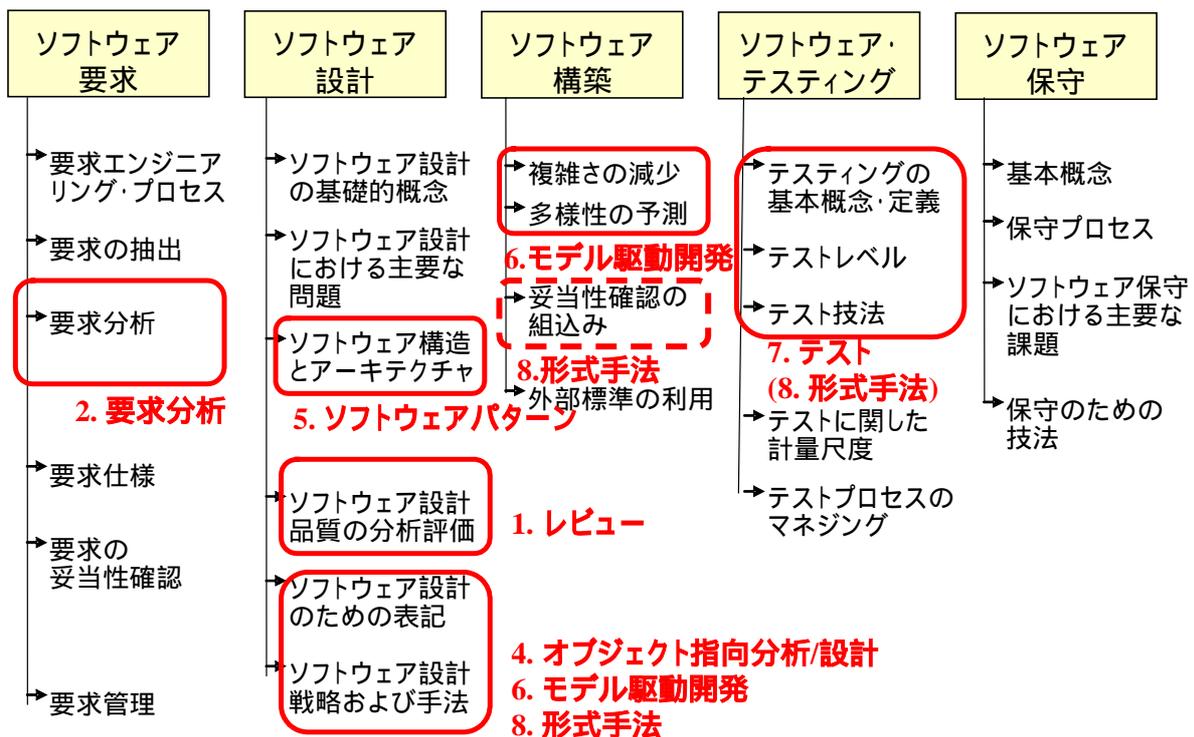


図 1 演習内容と SWEBOK ver0.95 における開発系知識領域の対応

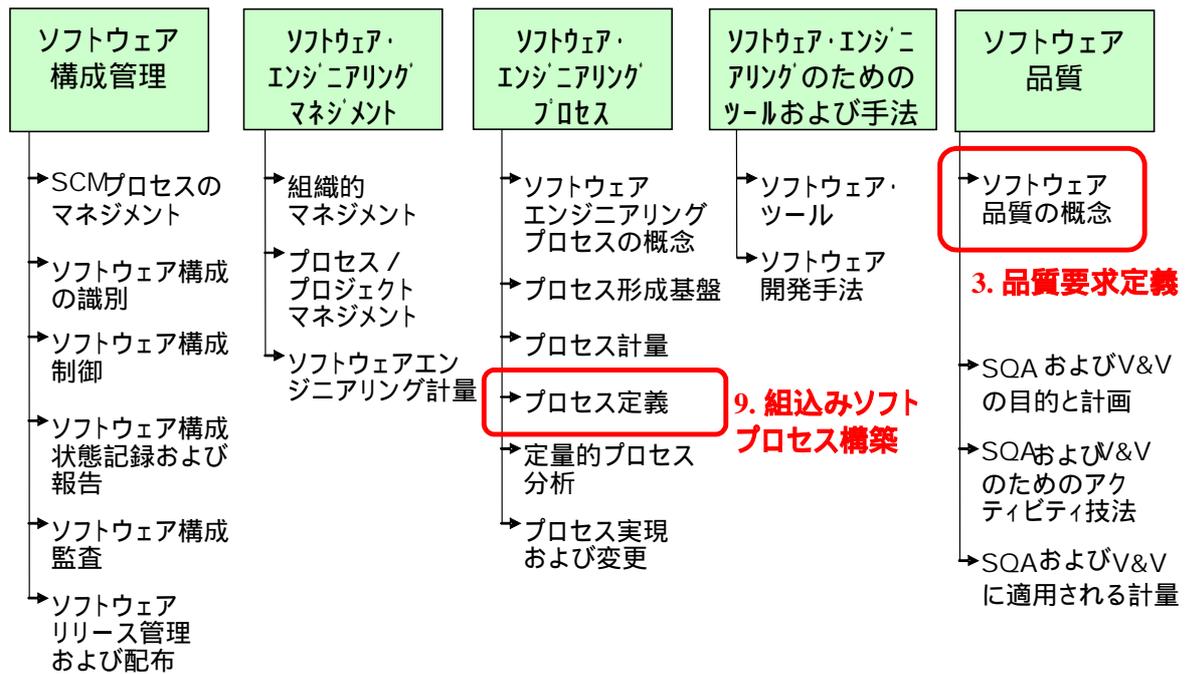


図2 演習内容とSWEBOK ver0.95における支援系知識領域の対応

## 2.2 全体評価

本コースの参加者(研究員(受講者)および主査・副主査)は20余名であり、臨時会を含む全9回の集中的な演習および活発な議論を通じて、種々のソフトウェア工学手法の学習に有効であったと考えられる。年度当初に掲げた本コースの目標、および、全ての例会を終了した後の達成度評価を以下に述べる。

- 主要な工学的技術の演習を通じた深い会得: 取り上げた9つの技術は、実開発への適用可能性の観点から、極めて実践的なものと、先端的かつ将来の適用が期待されるものの2種に大別できる。前者に該当する5種(レビュー、品質要求定義、オブジェクト指向分析設計、テスト、組み込みソフトウェア開発プロセス構築)について、各受講者は実開発への適用に向けた技術習得に成功したと考えられる。後者の4種(要求分析、ソフトウェアパターン、モデル駆動開発、形式手法)について、受講者は基礎概念を理解し、将来において適用を検討するための知識および適用時の基礎の形成に成功したと考えられる。総合すると、開発プロセスの流れに沿って各技法を配置したこともあり、受講者それぞれが開発プロセス全体を俯瞰する中で主要な技術を習得することに成功したと考えられる。
- 個人・組織の開発力強化のための基盤形成: 上述の前者5種については、各受講者が知識/技術を持ち帰って所属組織へと展開することが可能なレベルの深い会得が概ね達成されており、取り上げた技法はいずれもプロダクトやプロセスの品質の観点からソフトウェア開発における開発力向上に繋がるものであるため、その強化のための基盤形成に幾らか成功していると考えられる。後者4種については、各受講者が実際に役立つ技術として用いることができるようになるために、復習や一步先の自主的学習による深い会得、および、開発現場における適用/導入方法の検討が必要である。
- 仲間作り: 例会終了後の懇親会、全受講者の年間を通しての固定、および、毎回の(メンバを変更しての)グループ演習を通じて、同じような問題意識を持った仲間作りに成功した。

## 3. 例会での演習

本コースでは、ソフトウェア開発技術およびマネジメント技術に関する以下の演習について、それぞれ

個別に以下の講師(敬称略)を招いて実施した。各演習内容について、その概要を紹介する。

- 第1回:レビュー :猪塚 修(横河制御エンジニアリング株式会社)

ソフトウェア工学の分野でよく取り上げられているレビュー手法を用いて、受講者に実際にレビューを実施してもらい実施方法、有効性について議論した。演習では、WEBメールの簡単な設計に対して、各自でレビューをして、その後、グループで議論をし、レビューのやり方、特に ISO/IEC 9126-1 [5]の品質特性の観点からのレビュー、検証の視点からのレビューなどの解説と、グループでの議論を交互に繰り返し理解を深めると共に、最後に各グループの成果を発表した。

- 臨時会:要求分析:中谷 多哉子(筑波大学)

顧客ニーズをゴールという観点で捉えて要求定義を行う技法を扱った。開発のやり方と、要求の獲得から入り、ステークホルダ分析、リッチピクチャ、CATWOE、問題フレーム、ゴール指向分析、IEEE 830 [6]に示された要求仕様書(SRS: Software Requirements Specification)、要求仕様の品質特性について、解説がなされた。この解説に基づいて、会議スケジューリングシステムの要求仕様を定義するグループ演習を行った。

- 第2回:品質要求定義:東 基衛(早稲田大学)、中島 毅(三菱電機株式会社)

品質要求、品質測定技術、品質モデル、および品質評価が体系化されることが重要という解説がなされた。これを受けて、演習では、インターネットによる通信販売を課題として、グループに分かれて実際に品質要求分析と定義を行った。その方法は、システム機能要求の定義、外部品質要求の選別と優先順位付け、品質要求基準の設定という流れを、10段階からなるステップに分けて行うというものであった。今回は予習を含めた5つのグループ演習により、要求品質の仕様化と品質要求基準の設定方法を、より具体的に理解することができた。品質特性は、その概念は広く理解されているものの、実際に利用した経験を持つ人は少なく、第1回、臨時会、第2回、第3回の各演習で、品質特性を利用したことによって、理解する良い機会になったといえる。

- 第3回:オブジェクト指向分析/設計:井上 樹(株式会社豆蔵)

オブジェクト指向(OO)分析/設計、オブジェクト指向モデリングの効果を UML(Unified Modeling Language [7])で表現する事によって、絵(図)による理解の容易性、共通言語で作成するメリット、要求定義、要求分析、設計の仕方を講義、演習を繰り返すことによって、今後試してみようと思えるまで体験した。OOの経験者にとっても、各自の理解の整理、初めての受講者にとっても、概要を理解する上で、非常に良い機会になった。(UMLなどは以降の演習でも出てきたので、ここで基本を押さえてあったのは役立った)

- 臨時会:ソフトウェアパターン:鷺崎 弘宜(国立情報学研究所)

ソフトウェアパターン全般についての講義の後、各自の成功体験をノウハウとして一般化し、パターンとして抽出する手法の演習、および、オブジェクト指向設計におけるデザインパターン[8]の適用に関する演習を実施した。演習を通じて、パターンの概念、および、各パターンにはメリット・デメリットがあり適切な取捨選択と適用が重要であることを学習した。

- 第4回:モデル駆動開発:久保秋 真(サイバービーンズ株式会社)

自律走行ロボット制御ソフトウェアのオブジェクト指向およびモデル駆動開発を題材として、プラットフォーム独立モデルから依存モデルへの変換の演習を実施した。演習を通じて、モデル駆動開発の基本的な概念を理解し、同開発に関わる技術群の整理と将来の実践に向けた基礎の形成に成功した。特に、普段の開発において意識することが時として少ない「メタ」の概念を扱う重要な機会となった。

- 第5回:ソフトウェアテスト:鈴木 三紀夫(TIS 株式会社)

グループ単位により徹底的に、テスト技法として基本的な構造ベース・ホワイトボックステスト手法(特にステートメントカバレッジ、デジジョンカバレッジ)、および、仕様ベース・ブラックボックステスト手法(特に同値分割法、境界値分析、ドメインテスト、デジジョンテーブルテスト)の演習を実施した。演習を通じて、各テスト手法が導出された背景や概念を深く会得し、実践のために必要な手順や考え方を整理し、さらなる手法学習への十分な道筋を得ることができた。

- 第6回:形式手法:中島 震(国立情報学研究所)

信頼性を高めるための種々の形式手法(VDM,Zに代表されるモデル規範型や、SPINに代表される並行システム向け手法など)の発展経緯と概念について学習した後に、形式手法を実務に役立てる上での障壁や効果的な活用方法等について議論した。議論を通じて、数学的な背景理論への習熟の必要性や、テスト技法による信頼性向上が困難な並行処理等への部分的な集中的適用の効果の可能性などを整理し、さらなる手法学習への十分な道筋を得ることができた。

- 第7回:組込みソフトウェア開発プロセス構築:室 修治(独立行政法人情報処理推進機構・ソフトウェアエンジニアリングセンター)

IPA/SECにおいて策定された組込みソフトウェア向け開発プロセスガイド(Embedded System development Process Reference: ESPR [9])の目的および特徴を学習し、続いて、ESPRにおいてリファレンスガイドとしてまとめられている種々のタスク・サブタスクを実開発におけるプロセス定義の観点から集団レビューした。レビューを通じて、プロセス定義におけるESPRに代表されるリファレンスの重要性や、組込み開発とエンタープライズ開発における必要タスクの違い、上流フェーズからの品質作りこみのためのタスク実施の重要性を整理した。

#### 4. 工学技術の実践に向けて

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」を持ち寄り、議論を通じて整理してまとめた。具体的には、実務におけるソフトウェア工学手法の活用という観点から気がついた事柄、自身における姿勢などの変化、次に必要なアクションなどを考察し議論した。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、姿勢の変化や次の必要アクションを含めた「気づき」を整理検討することは重要である。整理した結果を以下に述べる。

##### 4.1 学習内容を実務に役立てやすいテーマ

受講者それぞれが本コースの振り返りを行ったところ、演習を通して様々な気づきがあったという意見が多数出た。その中でも実務への即座の役立てやすさという観点から意見の多かった4つの演習について、受講者の気づきを紹介する。

- 要求分析

- ソフトウェア開発において、表面的に要求されている仕様だけでは満足度や品質は上げられない。非機能要求の指針が品質向上のポイントとなる。
- 演習によって要求を分析する視点が再発見できたので、要求をより明確にするための工夫点が補強できた。
- ゴール指向分析手法である*i\**(アイスター)を適用して期待するサービスを発見するというアプローチ方法は、今まで属人的に行っていた要求分析作業において、自分が行った要求分析の背景・根拠を得るうえでも有用である。
- 一般に不具合を修正するコストは、その不具合を検出した工程が下流であればあるほど増加す

るといわれており、要求分析手法を用いて上流で品質を上げることで、コスト面でも効果がありそう。

- 品質要求定義
  - 今までは「機能性」や「信頼性」などの品質特性についてどのように目標を立てて結果を評価すればよいのか分からなかったが、今回の講義で具体的な品質目標の立て方やその評価方法についてなんとなく理解できた。
  - 定量的な品質測定方法として紹介された、ソフトウェアマトリクス の例が非常に有用だった。
  - 利用者は必ずしも自分のニーズを明確に把握しているとは限らない。その潜在的なニーズを引き出すことがシステムエンジニアとして必要である。
  - 演習で学んだ内容を踏まえて社内のレビュー教育の演習内容を見直した。その演習結果から教育内容が有効であるとの回答を得ることができた。
- ソフトウェアテスト
  - 実務で行ってきたテストの視点が演習で確認できた。
  - テストを実施するうえで完了基準を設けることの重要性を学んだ。自社でのプロジェクトについても今後、網羅率を完了基準として利用したい。
  - 多くの時間をテストに費やしているが、それを改善するためのポイントが得られた。
  - 演習に対してグループディスカッションをしてみると、人によってテストケースの粒度にばらつきがあり、各会社によってソフトウェアテストに関する「文化」に違いがあると感じた。
  - 「どこまでテストをすれば十分なテストを行ったといえるか」ということについていつも疑問を持っていたが、テストケースの十分性に関する根拠の示し方がわかった。
  - 開発手法の改善というと設計から製造までを主に意識していたが、テスト手法およびテスト基準について考えることの重要性がわかった。
- 組み込みソフトウェア開発プロセス構築
  - 演習で使用した ESPR は、ソフトウェア開発の各工程の実施項目や、それに伴うインプットとアウトプットが明確になっている為、新たに開発管理の標準化を図りたい、または現在行われている開発管理を改善したい際に参考にできると感じた。また、演習のグループディスカッションにて「組み込みソフトウェア向け開発プロセスガイド」と、各自が実際に現場で使用している開発管理とを比較したが、組み込み系の受講者と非組み込み系の受講者との間で多少の意見の違いがあったが、非組み込み形の開発管理にも十分に参考にできると感じた。
  - 組み込み系とエンタープライズ系のプロセス(文化)に違いがあり、同じソフトウェアでも製品特性によりプロセスを柔軟に最適化する必要があると考えさせられた。(エンタープライズ系のほうが、下工程で決定する事項が多いと感じた)
  - 適切な開発ステージにおける明確な目標や仕事の流れを持つことにより、安定した品質を保つことができると考えられ、その点で ESPR を有効活用できると考えられる。

#### 4.2 受講者の実務内容とのギャップがあるテーマ

本年度の受講者の多くは、C 言語などの非オブジェクト指向言語を実務で用いているため、オブジェクト指向分析設計およびソフトウェアパターンについて、学習内容を実務に直接に結びつけることは幾らか困難と考えられる。しかしながら、問題領域の抽象化の考え方や追跡性を持った分析設計の流れ、ノウハウの文書化アプローチがソフトウェア開発に役立つことを十分に実感することができた。

以降において特にオブジェクト指向分析設計について得られた気づきをまとめる。

- オブジェクト指向分析 / 設計

今回の演習では最もポピュラーと思われるテーマであるが、実際に現場で活用している受講者は少数であった。演習ではオブジェクト指向による要求分析～設計を UML を用いて自ら考えて実践することで、それぞれ自分なりの興味や価値を持つことができた。オブジェクト指向の長所や短所を考察し、現場への今後の導入の可能性を考えるきっかけとなった。

#### 4.3 将来において実務に役立てたいテーマ

工学演習コースの内容の中には、現状は困難であるが将来的には適用したい技術として受講者が捉えたテーマもあり、その捉え方もさまざまであった。ここではいくつかのテーマに絞り、受講者がどのように感じたかを簡単に紹介する。

- 形式手法

ソフトウェアの品質として信頼性は意識しているが、それを向上させるための手法がわからない。その指針としてのテーマである形式手法では、その可能性についてディスカッションが行われた。そこでの意見としては、導入の価値は高いがスキルや環境による障害が大きいなど、現状の問題点が多く挙げられた。しかし、モデル検査の面を活かして、例えば状態遷移が複雑なシステムの検証や、実機での確認が物理的に不可能なパターンの検証といった、テスト工程での導入の可能性も論じられた。

- モデル駆動開発

モデルを解析してコードを自動生成する手法は、現状での設計からコーディングにおける不整合に悩まされる場面では大きな助けになると思われる。特に、「面倒な仕事は自動化する」という考え方を体現しているモデル駆動開発に興味をもった受講者は多い。しかし、現場での実用化までは敷居が高く即効性を求めるのは難しいが、実際にモデル駆動開発を扱ったイベントも数多く開催され、今後の導入のしやすさに期待が持てる場所である。

#### 5. ソフトウェア工学演習コースの学習の仕組み

本分科会では狙い(1章)にもあるように、指導講師による講義、演習を通じて、ソフトウェア開発工程の上流から下流までの主要な工学的技法を深く会得することを目的とした。特に演習を重視しており、各回のテーマに基づいて4～5名の小グループでの議論、まとめ、発表を行うことで、より深い理解を促した。

##### 5.1 演習形式の有効性

演習の際には、模造紙とペン、付箋紙、PCなどの道具を用いて手を動かして作業した。この方法は単なる座学と比べて各種工学的技法を身につけるのに大変有効であった。受講者からは特に「議論の有効性」「まとめ、発表の有効性」に関する好意的な意見が多くあげられた。

##### 5.2 議論の有効性

演習ではグループ毎にテーマに沿った演習課題の検討、議論を行い、その過程で以下に示す効果が得られた。

- 異業種エンジニアとの交流: 業種は異なっても同じような問題意識を持っており、それを共有することができた。
- 専門性を持った指導講師との質疑応答: 指導講師の専門性が高く、参加者が既に習得済みの技術であっても、より深い質疑応答ができた。
- 理解の深まり: 他の受講者の意見を聞き、議論することで、異なる視点で物事を考えることができた。その中で自業務への技法の適用のヒントを得ることができた。

##### 5.3 まとめ、発表の有効性

グループで議論した結果をPowerPoint<sup>(R)</sup>等にまとめ、発表することで、以下に示す効果が得られた。

- 時間内で成果をまとめる能力の習得: 新QC七つ道具にも含まれている新和図法(KJ法:川喜田二郎氏考案)などを用いることで、制限時間内にグループで議論した結果をまとめることができた。
- 理解の深まり: 他グループの発表を聞き、意見交換することでテーマにあげられた工学手法をより深く理解することができた。

## 6. おわりに

受講者各位には、本コースを通じて今までの自分たちのやり方だけが手法でないという、自分自身に対する気づきをもって、各組織の中で、熱意を持って「ソフトウェア開発をエンジニアリングと呼べる状態」にするべく努力されることを願う。また、次年度も本コースに参加して議論の深耕をしたり、他の分科会にて新たに取り組むなど、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。

次年度も、演習メニューを改善した上で本コースを実施する。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、受講者の方々に草案を分担執筆いただいた。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も、この場を借りて厚く御礼申し上げます。

## 参考文献

- [1] 松本吉弘, 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. (最新版および ver0.95 は <http://www.swebok.org/> より取得可能)
- [2] K. Beckman, N. Coulter, S. Khajenoori, and N. R. Mead, “Collaborations: Closing the industry-academia gap,” IEEE Software, Vol.14, No.6, pp.49–57, 1997.
- [3] S. Honiden, Y. Tahara, N. Yoshioka, K. Taguchi, and H. Washizaki, “Top SE: Educating Superarchitects Who Can Apply Software Engineering Tools to Practical Development in Japan,” Proc. 29th International Conference on Software Engineering, ACM Press, 2007 (to appear).
- [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] ISO/IEC9126:2001, “Software engineering Product quality Part1: Quality model”, 2001.
- [6] IEEE Std 830-1998, “IEEE Recommended Practice for Software Requirements Specifications”, IEEE Computer Society, 1998.
- [7] OMG, “UML 2.0 Specification,” 2004, <http://www.uml.org/>
- [8] E. Gamma, R. Johnson, R. Helm, and J. Vlissides., “Design Patterns: Elements of Reusable Object-Oriented Software,” Addison-Wesley, 1995.(邦訳)本位田真一, 吉田和樹 監訳, オブジェクト指向における再利用のためのデザインパターン, ソフトバンククリエイティブ, 1999.
- [9] 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター, “組込みソフトウェア向け開発プロセスガイド”, 翔泳社, 2006.