

演習コース「ソフトウェア工学の基礎」2010年度 活動報告

Report on Practice Course of Software Engineering Foundations in 2010

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
野中 誠 東洋大学 小倉 徹 株式会社山武 ビルシステムカンパニー
鈴木 尚 株式会社山武 ビルシステムカンパニー 片山 拡充 キヤノンソフトウェア株式会社
古谷 伸一 イーソル株式会社 中田 陽大 テクマトリックス株式会社
升谷 雄二 テクマトリックス株式会社 吉田 麻紀 株式会社インテック 本田 繁 日本電子株式会社
長嶋 聖 株式会社山武 塩浜 龍志 早稲田大学 下條 清史 早稲田大学

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の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6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。

Following the success of previous courses in 2005-2009, 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]を参照されたい)。そこで2010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、2010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, 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。

要求	設計	構築	テスト	保守
要求の基礎的概念 要求エンジニアリングプロセス 要求の抽出 要求分析 要求仕様 要求の妥当性確認 実践上での考慮事項	設計の基礎的概念 設計における主要な問題 構造ヒエラルキヤ モデル検査 設計品質の分析評価 設計のための表記 設計戦略および手法	構築の基礎的概念 構築の管理 実践上の考慮事項	テストの基礎的概念 テストレベル テスト技法 テストに関する計量尺度 テストプロセス	保守の基礎的概念 保守プロセス 保守における主要な課題 保守のための技法
構成管理	マネジメント	プロセス	ツールおよび手法	品質
SCMプロセスのマネジメント 構成の識別 構成制御 構成状態記録および報告 構成監査 リリース管理および配布	開始と範囲定義 プロジェクト計画 プロジェクト実施 レビューおよび評価 最終 計量	プロセス実現および変更 プロセス定義 プロセスアセスメント プロセス計量	ツール (各回での扱い) 開発手法	品質の基礎的概念 品質・レビュー マネジメントプロセス 測定 実践上の考慮事項

図 1: SWEBOK における各技術の位置づけ

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 要求分析 要求工学, オブジェクト指向
	プロセスアセスメン トのマネジメント	構成管理		レビュー 品質・レビュー, アーキテクチャ, 形式手法
	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	テスト テスト
	監査のマネジメント	プロジェクトマネ ジメント全般	テストのマネジメ ント	品質分析・ 評価 品質データ分析(測定)
	教育のマネジメント		品質評価のマネ ジメント	運用・保守 運用・保守
	法的権利・責任の マネジメント		運用・保守のマ ネジメント	

図 2: SQuBOK における各技術の位置づけ

(2) 各回の振り返りと1年間を通じた実践の促し

各回における工夫として、2010 年度に引き続き演習終了後に必ず振り返りの時間を設定し、各自の気づきを箋紙に一件一葉に記述してボード等に張り出して説明し、皆で共有することとした。得られた気づきは、取り上げた技法の実務における有効性や、適用における留意点など多岐にわたり、集団で同じ技法に取り組み多様な観点や立場で考察することの効果を感じられた。

さらに、1年間の冒頭で各自が、習得技法の活用・実践について目標を設定し、その達成について中間状況と最終状況を互いに報告する機会を設けた。これにより、ただ漫然と各回の技法を習得するのではなく、強い目的意識をもって技法を習得しそれを各自の業務や個人へと積極的に役立たせることを幾らか奨励できた。結果として、次節で述べるように実際に業務にて活用できた技法が多数出現し、その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。

(3) 交流の促進

例会終了後のほぼ毎回の懇親会開催、研究員の年間を通しての固定、および、毎回のグループ演習を通じて、同じような問題意識や目的を持った仲間作りに成功した。交流にあたり、研究員の所属や立場は様々であり、同業他社における事例や、異なる立場・視点に接する機会としても本コースは機能したと考えられる。

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以下に述べる。

下記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。2010 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 2010 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:レビュー:猪塚 修(横河ソリューションズ株式会社)

- 概要:レビューとは、対象物がある目的(欠陥を発見する、より良い設計にするなど)のために査読する技術である。重要なことはレビューの目的を明確に定義し関係者間で共有すること、そして色々な視点でレビュー対象を確認することである。
講義ではレビューの定義や、レビュー対象の読み方、手法としてチェックリストを用いたレビュー(CBR)、シナリオベースのレビュー(SBR)について学んだ。演習では同じレビュー対象に対して、①何も準備されていない状況->②CBR->③SBR(グループ演習)と、段階的に手法を導入して指摘内容の変化を体験した。
- 有効性: 1 回目の演習では何を指摘してよいか分からなかったが、2 回目はチェックリストを使うことによって視点が定まり、指摘が出てくるようになった。3 回目はシナリオによる視点の変化に加えて、グループ演習によって他のメンバーの視点が自分と異なっていることを確認でき、また異なる指摘項目を発見できた。演習の度に視点が変わり、指摘項目が増えていくことを実感できた。
また、2 回目の演習ではチェックリストとして ISO9126 の品質特性を使用した。要求分析や基本設計段階の機能抽出の網羅性を確認するチェック項目として有効と感じた。
- 留意点: チェックリストはレビュー実施時だけではなく、開発担当者が開発作業を行う前にも確認するべきである。
また、ISO9126 品質特性にはトレードオフの関係にあるものがあり、定義されている全ての品質特性を同時に満たすことはできない。レビューの視点として使用する場合はレビュー対象に必要な品質を選定する、または優先度を設定する必要がある。
- 活用事例:基本設計工程におけるレビューチェック項目として、ISO9126 の品質特性を導入した。レビュー対象に必要な品質特性を定義し、その特性を基に具体的なチェック項目を検討してチェックシートに追加した。チェックシートの内容を関係者に事前展開し、レビュー実施前にチェックシートの確認を実施するようにした。

●臨時会:要求工学:中谷多哉子氏(筑波大学大学院ビジネス科学研究科)

- 概要: システム開発を失敗させないためには、要求の曖昧さを無くし、できるだけ完全な真の要求を抽出する必要がある。要求抽出を実践するために様々な手法があることを理解し、各手法の特徴を、グループワークを通じて理解した。
- 有効性: 要求抽出手法のリッチピクチャは、直感的な現状把握する点において効果的だと理解した。また、RODAN の役割依存モデルでは、要求者の立場・役割を分析し、依存関係を明らかにすることで、課題をより深く抽出できる点で効果的であると理解した。
- 留意点:開発現場では、要求の変更・修正・追加・抜け漏れ等、修正を求められる場面が多々ある。これらの修正を減らすためには、要求を「獲得する」ではなく「抽出する」という意識が重要である。また、要求者の根拠・背景、そして解決すべき本質的な課題(ゴール)をはっきりさせることが重要である。

●第 2 回例会:オブジェクト指向モデリング:井上樹氏(株式会社 豆蔵)

- 概要:モデリングが開発においてどのように役立つのかを知ることを目的とし、UML を用いて要求、設計のモデリングを行った。要求のモデリングでは、「ユースケース」、「ステートマシン図」を作成し、シナリオ分析、モデルベース分析を行った。設計のモデリングでは、オブジェクト指向設計の演習を

行った。

- 有効性:モデリングを行うことによって、開発者間のシステムに対する理解が深まり、イメージを共有できる。要求分析では、UML を活用することで動的、静的及び機能的な側面から対象を捉えることができ、よりの確に要求のモデル化が行える。オブジェクト指向は、システム化の対象世界をオブジェクト同士の振る舞いとして捉え、モデル化することで修正しやすく、シンプルなプログラム作成につながる。
- 留意点:システム化対象の構造をしっかりと理解してモデリングを行わないと、開発者間におけるイメージ共有の妨げとなってしまう。また、UML モデルのコード変換をする場合、十分な情報量を持つモデルを正確に書かなければいけない。

●第3回例会(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:目標のないアーキテクチャ = どうでもいいシステム を作り出さないためにも、目標とする品質特性を考慮してアーキテクチャ設計を実施することが重要となる。設計したアーキテクチャは、ATAM (The Architecture Tradeoff Analysis Method)という手法(ドキュメントなど)を用いたり、マトリクス測定(ソースコード)を実施したりして評価をする。演習では品質特性シナリオを作成、それに肉付けをして最後に評価を行うという一連の流れを実施した。
- 有効性:アーキテクチャが適切に設計されていればメンテナンスコストを削減することができる。そのためアーキテクチャの分析、評価を実施することは非常に重要なことであることを理解できた。またそれに関連して、アーキテクチャの評価の際にツールを使用してチェックを実施することも有効であることを理解した。
- 留意点: 品質特性シナリオは実際の運用環境で確認する必要があるが、完成するまでのタイムラグ、実施コストの問題がある。完成後かつ問題発生前に確認するためには PDCA のサイクルでうまく管理する方法を考えることがポイントになる。

●臨時会:ソフトウェアテスト:鈴木三紀夫氏(TIS 株式会社)

- 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、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を行った。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、最後に解答を模造紙に書いてグループ毎に発表する形式で行った。
- 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グラフを用いる方法や、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。演習後には各技法の目的、考え方、要点などを解説していただき、ソフトウェアテスト技法の基礎を理解することができた。
- 留意点:演習ではグループ全員の解答が一致することはほとんどなく、人によって視点や考え方が違ってさまざまな解答が出てきた。このようにテスト技法は人によって解釈が変わるのでいかにしてばらつきを抑えるかが重要である。
- 活用事例:テスト漏れをなくすこと、テストケースをわかりやすくすることを目的として、デシジョンテーブルを使ってテストケースを作成した。複雑で組み合わせの多いシステムのテストケースの整理には有効なものだと実感できた。今後はいかに上手くテーブルを圧縮して効率良くテストを行うかが課題。

●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: モデル駆動開発(MDD)として、時代背景、関連する用語(MDD、MDA、CIM、PIM、PSM)、メリット、開発事例およびモデル変換の基礎を学んだ。また演習として、バグの仕込まれたロボットのプログラムを、モデルからデバッグ・修正し、プログラム自動生成を行った。

- 有効性: MDD の有効性として、プログラム自動生成によるメリット以外に、アプリケーションのモデルはアプリケーション分野の専門家が開発し、ターゲット用にコード生成するモデルは生成するコードの専門家が担当するという実装技術と機能仕様の分離を行うことができる。また、現状ではコードと正確な対応が取れておらず、スケッチや参考資料に過ぎない設計(モデル)が多い中、MDD によりモデルレベルからプログラムの品質を意識した作り込みが可能となる。
- 留意点: MDD はコードの自動生成によって工数を短縮する技術だと理解されがちであるが、実際の工程のうち、自動生成によって短縮が期待される部分は全体から見て大きくはなく、過度な期待は禁物である。MDD は、成果物を自動で生成してくれるツールとしてではなく、成果物の生成から属人性を排除し、品質を一定に保った成果物を得るツールとして認識すべきである。また、ツールなどの開発環境が整わなければ MDD を導入できないと考えられがちであるが、モデルから成果物への「変換規則」を整備し、誰でも同じ成果物を生成できる様にするのも MDD として重要な点である。

●第 5 回例会:モデル検査・形式検証:吉岡信和氏(国立情報学研究所)

- 概要: 複雑かつ高度分散化するソフトウェアに対して、その使用における可能な実行パターンを網羅的かつ自動的に検査することで検証を行うモデル検査について、LTSA という実際のツールを用いた演習によりその実施方法および有効性・有用性を学習した。
- 有効性: ソフトウェアテストでは振る舞いの網羅性や検証の厳密性において部分的または不確実であり、これらの短所が致命的になりつつある。しかし、モデル検査では網羅的かつ厳密に検証できるため、複雑・分散化が進む現在のシステムに対してかなり有効に感じた。また、モデル検査ツールでは検証の自動化を実現しているため、コストを抑えつつ品質を上げることができる可能性がある。
- 留意点: LTSA でモデル検査を実施するにあたり、状態遷移図の生成には FSP による記述が必要である。そのため、難解な FPS の記述方法を習得する負担が LTSA を導入する際の障害となる。状態遷移図生成の容易化をはじめとする利用容易性の向上を実現することができれば、さらに利用可能性が高まるだろう。

●第 6 回: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: アジャイル開発とは、価値を生み出すソフトウェアを、手早く無駄なく作る手法の総称である。ソフトウェアの変更コストカーブが平坦であることが大きな特徴である。アジャイルによる開発として、KPT(Keep: 維持したいこと、Problem: 問題点、Try: 次のイテレーションで実施したいこと)による振り返りのやり方、タスクボードとバーンダウンチャートを使った時間管理について学んだ。演習では、折り紙を使った多面体作成をチームで実施し、アジャイル開発を実際に体験した。演習を通し、KPT による振り返りで作業効率が向上すること、見積の精度が向上することを体験することができた。
- 有効性: アジャイル開発では、短い期間で区切りながら開発を進め、その区切毎の KPT による振り返りで作業の質が向上していく。タスクボード、バーンダウンチャートを有効に使うことで、プロジェクトメンバーの作業負荷、プロジェクトの進捗状況、開発スコープの妥当性が見える化できる。その結果として、スケジュールの大きな誤り、プロジェクトの大きな失敗がなくなる。時間を細かく区切ることにより、その時点での作業者のパフォーマンスを最大限に引き出せることができる。
- 留意点: アジャイル開発では、QCD を変えずに開発のスコープを変更するため、アジャイル開発の趣旨を顧客としっかり共有することが大切である。開発のスコープが変わるため、契約が難しいところも注意点である。作業者のモチベーションが高いときには、KPT による振り返りで作業効率が向上していくと考えられるが、作業者のモチベーションが下がったときに、際限なく作業効率が低下していくのではないかと感じた。負のループには注意を払っていくこと、細かくスケジュールを管理しつつも、全体としての最適化も考える必要がある。振り返りによるフットワークの軽さは、開発だけでなく、新しい技術を追求していくような研究の分野にも適用できる手法ではないかと感じた。

●第7回例会:ソフトウェア品質データ分析(測定)、野中 誠氏(東洋大学)

- 概要: 上流工程で、作り込み欠陥数が不明のまま欠陥除去率の目標値を設定するだけでは、目標値の有用性はほとんどない。欠陥数を予測できれば、上流工程の欠陥除去率の予測値が得られ、目標値に有用性をもたせることができる。適当な欠陥数の予測値を得るために、適当な欠陥数の数え方や、レイリーモデルなどの統計モデルを用いた欠陥数の予測方法について学んだ。演習として、統計解析ソフト「R」を使用し、様々なデータと欠陥数の関係を分析した。
- 有効性: 高い信頼性のある欠陥数の予測値を得ることで、欠陥除去の適当な目標値を設定できる。目標値に向かって技術や手法を改善することで、高品質なソフトウェアを開発できる。
- 留意点: 最も留意すべき点は、「欠陥数の数え方・欠陥 1 件の粒度」をプロジェクトメンバーで統一することである。個々で数え方が違うと、適当な予測値が得られなくなってしまう。また、説明変数としてリッカート尺度¹を使用した場合は、データ処理の方法に注意する必要がある。リッカート尺度は質的変数であるため、単純に加減算できないためである。レイリーモデルがよく適合し、早期に欠陥が除去されているプロジェクトが必ずしも成功プロジェクトではないということも、念頭に置いておく必要がある。

●第8回:アスペクト指向:鷺崎 弘宜(早稲田大学/国立情報学研究所)

- 概要:アスペクト指向とは、本質と横断的関心事を分離して捉えることで複雑な結びつきを解消し、織り込みルールにより1つに纏め上げる構造化手法である。利点として既存のオブジェクト指向以上にモジュール間の結びつきを弱くできるため、部品の再利用化が容易に行える。開発環境へのプラグインであるAspectJを用いてアスペクト指向プログラミング(AOP)を体験することができた。
- 有効性:構造を簡略にできるため高い可読性や保守性を見込める。保守性結合ルールさえあれば本質のプログラムに対してロギングなどの処理を容易に後付け可能であるため、既存のコードを修正する必要がなくデバッグ作業などにも有効である。また、別の横断的関心事と織り込みルールで織り込むことで別のプロダクトラインを展開することも可能であり、高い再利用性があると言える。AOPはJavaに限らずC/C++言語などでもAspectCやAspectC++がオープンソースで開発されているため容易に体験可能である。
- 留意点:従来とは異なる手法のため工数などの初期コストが高かかってしまう。本質を分離しすぎると構造が分かりづらく全体の見通しを悪くしてしまい、保守性の低下や信頼性の低下にも繋がる。横断的な処理は織り込みルールで定めた処理に対して実行するため、予想外の動作を起こさぬよう事前に関数名などの命名ルールを定めておく必要がある。構造や結びつきが分かりづらくなる点については開発環境などツールの手助けが必須であろうと感じた。織り込みルールや構造を把握した者でなければテストを行いづらく、テスト漏れを起こしやすい印象を受けた。

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。

¹ 質問項目に対する回答者の態度を測定する簡便な方法。5段階や7段階の尺度がよく用いられる。

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。

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 2004一, 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.