演習コース「ソフトウェア工学の基礎」 2009 年度 活動報告

Report on Practice Course of Software Engineering Foundations in 2009

鷲崎弘宜（早稲田大学） 田村一賢（東芝ソリューション株式会社）
野中誠（東洋大学） 加藤岡弘一（伊藤忠テクノソリューションズ株式会社）
上村秀一（株式会社東芝） 高田祐布子（オムロンアミューズメント株式会社）
中島碧莉（IT ホールディングス株式会社） 古木健（東京海上日動システムズ株式会社）
森崎一邦（東芝電波システムエンジニアリング株式会社） 横内和城（日本電子株式会社）
吉川真吾（伊藤忠テクノソリューションズ株式会社） 村上真一（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 5 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 2005-2008, 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

エンタープライズシステムから組込みシステムに至る隅々にまでソフトウェアが搭載され、その役割の重大さは日々増大している。このような状況で、携帯電話のリコールや、銀行システムに代表されるインフラの停止など、ソフトウェアの品質・不具合に起因するトラブルは後を絶たない。その要因には、ソフトウェアに対する要求によるものと、ソフトウェアを生み出す人材によるもののいずれもが含まれると考えられる。具体的には、ソフトウェアの規模・種類が増大する一方で、相反するかのように、コストや納期に対する要求が厳しさを増している状況[1]は、ソフトウェアの品質に影響を及ぼしている。さらには、質の高い人材を生み出すシステムが機能していないこと[2]も大きな一因である。つまり、（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[2]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[3]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[4]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)[4]や SE2004（旧 CCSE）[5]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（[6][7][8][9]を参照されたい）。そこで 2009 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。

- G1: 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会得
- G2: 個人・組織の開発力強化のための基盤形成
- G3: 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

コースの設計にあたり、ソフトウェア工学における知識体系の広がりとしての SWEBOK における各回の位置づけを提示することで、実践に近い体験を通じた技術習得と同時に、演習結果を体系的な知識整理・学習につなげられるように配慮した。また、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コース全体を設計し、同じく体系的な知識性・学習を促すように配慮した。得られた設計結果として、SWEBOK との対応付けを図 1 に示す。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（G1）に有効であったと自己評価している。

(2) 各回の振り返りと1年間を通じた実践の促し

さらに各回における工夫として、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が感じられた。

さらに、本コースでは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（G2）について一定の達成をみたと評価している。

(3) 交流の促進

例会終了後の懇親会、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（G3）に成功したと自己評価している。なお、その交流にあたり、研究員の立場は様々であり、異なる立場や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。ただし、懇親会の開催については各自の多忙さ等の理由により数回の実施にとどまったため、より多くの開催による深い親睦と交流を次年度以降の課題としたい。

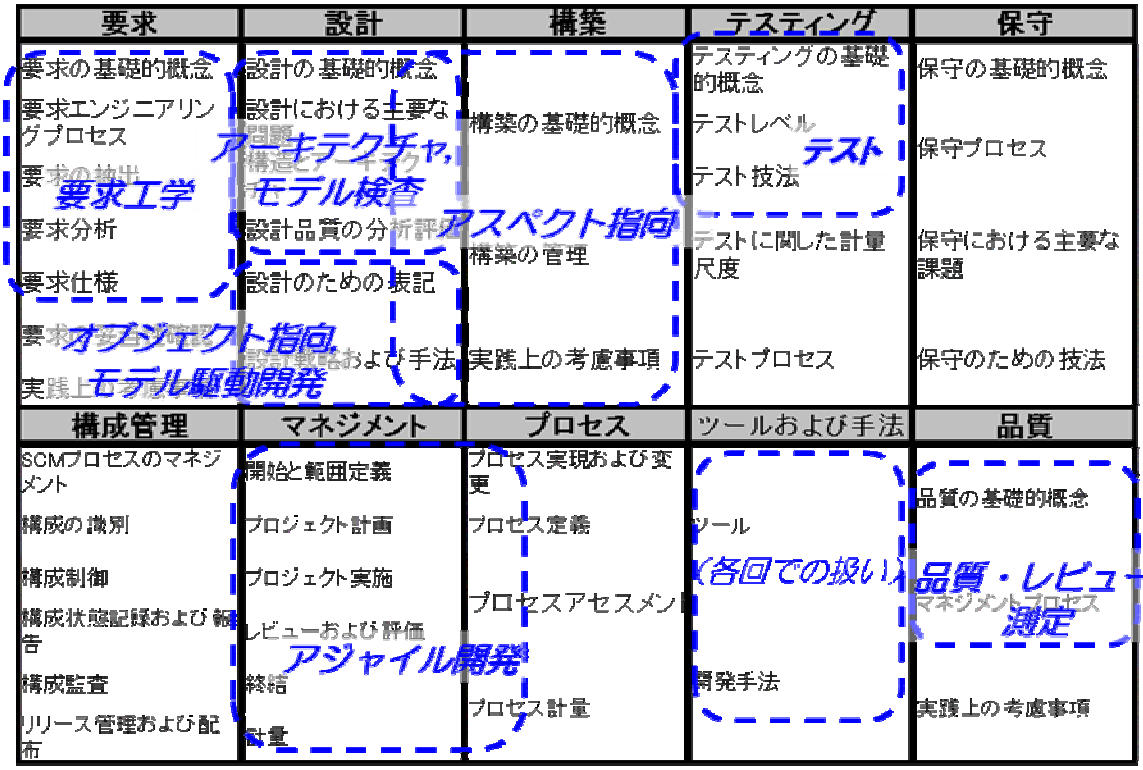

図 1: 演習内容と SWEBOK における知識領域の対応

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。2009 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 2009 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:ソフトウェア品質・レビュー、講師 (敬称略) : 猪塚 修 (横河ソリューションズ株式会社)
  - 概要: ソフトウェアの品質を確保する為行なう各種レビューについての演習を行った。品質特性を使ったレビュー、シナリオベースのレビュー、重点を絞ってレビュー、視点を変える、チェックリストの考え方等の演習を行った。
  - 有効性: ISO9216 に沿って品質レビューチェックリストを作成するのは効果的だと理解した。レビュー時にチェックシート以外で出た項目については、チェックシートに追加し、チェックリストを常に改善していく事も有用だと理解した。
  - 留意点: レビューは必要なものだと認識をしているが、実際の開発現場では忙しい等の理由か

ら正しく行なわれない事が見受けられる。また、バグをつぶすという意識でレビューを行なう事が重要。レビューの定義、レビューの観点、チェックシートの運用、レビューを行なうタイミング等をプロジェクトメンバー全員で最初に正しく認識あわせをすることが重要。

- 活用事例: プロジェクトで必要なレビューの一覧の洗い出しを始めた。また、過去のレビュー結果を元にチェックリストの作成を開始した。

- 臨時会: アジャイル開発、天野 勝(株式会社永和システムマネジメント)
  - 概要: ソフトウェア開発工程にて発生する種々の変更に柔軟に対応する手段としてアジャイル開発がある。アジャイル開発をチームで進める上で基本となるふりかえりやタスクボードを利用した仕事の進め方について、協調ゲームの高得点を目指すカイゼン、家の模様替えを家族で期間内に実施するというシミュレーションをおこないながら学んだ。
  - 有効性: プロジェクトにおける現状の Problem からその改善策を探し Try する。ふりかえりにより、うまくいったものは Keep してうまくいかなかったものは更なるカイゼンをおこなう。このような KPT によるカイゼンサイクルを繰り返すことでプロジェクトを成功へと導きメンバーの意識の共有にもつながることを実感できた。またタスクボード等を有効に使うことで現在のプロジェクトが抱える作業やメンバーの負担状況を見える化し、スケジュールを管理することに役立つことが実感できた。
  - 留意点: アジャイル開発は臨機応変に対応できることが魅力であるが、開発日程におさめるための作業の優先度付け、要求者との交渉は必ず必要である。また割り当てられたタスクをすぐにこなしてしまうような能力のある人間に仕事の負荷が集中することが考えられる。これらのことからもある程度作業の方向性が一致している小規模プロジェクトでより効果を発揮する開発スタイルであると感じた。
  - 活用事例: 現在携わっているプロジェクトのタスクボードを作成し、メンバーの負荷状況や積み残しの作業を「見える化」した。また TODO から DONE へとタスクのラベルが移動していくのが一目瞭然であり各自の達成感を実感できるようになりモチベーションの向上につながっている。KPT をベースとしたふりかえりミーティングを実施しカイゼンを継続実施している。
- 第 2 回: ソフトウェアテスト、鈴木三木夫(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に、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 5 種類の学習を演習を通して行った。演習は、まず個人で問題に取り組み、その後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まとめ、グループ間で発表し合うという形式がとられた。
  - 有効性: テストの行い方や、制御フローパステストの事前条件・事後条件のようなテスト要素の規定の仕方は、企業によってまちまちであり、どうあるべきかという議論は多くの場合うやむやにされている。本演習では、主要なテストの考え方や、それぞれのテストの勘所などが丁寧に説明され、ソフトウェアテストの基礎を理解することができた。
  - 留意点: テストの種類はたくさんあり、テストする対象によって決められる。テストしたい対象や、テストの意図が複数あるような場合は、無理にテストをまとめようとせず、分けてテストを行うようにすべきである。
- 第 3 回: アーキテクチャ設計・評価: 長谷川 裕一(合同会社 Starlight&Storm)
  - 概要: アーキテクチャとは、何か？から始まり、アーキテクチャの評価をどのようにするかを演習を通じて体験する。演習では、アーキテクチャ設計の座学を実施し品質特性に着目したアーキテクチャ評価の ATAM (The Archietecture Tradeoff Analisis Method) を活用し、アーキテクチャ評価の演習を実施した。ATAM は、品質特性シナリオを作成するので、非機能要件の理解にも繋がる演習である。
  - 有効性: 現状、どの組織でもレビューでアーキテクチャの評価を実施している。現状ではそこまで評価するのか不透明感が残るケースがある。本演習では、品質特性シナリオを活用するので、

アーキテクチャで重要な非機能要件抜け漏れが防げる点では非常に有効と認識しアーキテクチャの評価する一助となる演習である。

- 留意点: ATAM は、開発現場で適用できるようにカスタマイズをする必要がある。現状、組織やプロジェクトで、ATAM と同様な評価している場合は、ATAM の良いところだけをを参考にすることが重要。もし、初めて ATAM を適用する場合は、ATAM の考え方を参考に ATAM の良いところだけを取り入れることから始めることを認識することである。愚直に ATAM をそのまま活用するのは、かなり危険である。理由は、そのまま ATAM を活用した場合、作業量ばかり増えるだけで開発現場で受入れらない可能性があるため。
- 活用予定: 現在のプロジェクトにおいて、ATAM をもとにアーキテクチャ評価のチェックリストをカスタマイズしていく予定。

- 臨時会: 要求工学、中谷 多哉子 (筑波大学)
  - 概要: システム開発における要求工学の重要性を理解する目的で、演習「要求抽出の3つのプロセス」をテーマに要求工学を実践した。要求獲得の手法 (リッチピクチャ、CATWOE分析、ゴール指向分析) を適用し、ステークホルダの意見抽出、課題の背景の理解、ステークホルダの世界観を理解し、ゴールである要求獲得の演習を行った。
  - 有効性: 3つのプロセス (Ⅰ.システムが導入される対象の情報 (背景) を得る段階、Ⅱ.システムに求められる要求 (目的) を獲得する段階、Ⅲ.獲得した要求を取捨選択し新たな要求に整理する段階) を経て仕様 (要件) を固めることが遠回りのように思えたが、無駄のない標準化されたプロセスであることを演習を通し体験できた。特にリッチピクチャは、初期のプロセスでシステムの置かれる背景 (課題や要素) をステークホルダから抽出するのに有効である。
  - 留意点: 納期、コスト面から獲得した要求の取捨選択を行う場合、捨てられた要求事項やあいまいな要求事項についても記録 (課題) として残すことが重要で、問題の先送りや要件の抜け漏れ防止に繋がることがわかった。
  - 活用事例: リッチピクチャを使い、職場での課題抽出とゴール指向分析を参考に行った。ゴールまでの全体像が具体化されることで、取組みの弱い部分が早期に発見でき情報共有の素材になっている。
- 臨時会: オブジェクト指向分析設計、井上樹 (株式会社豆蔵)
  - 概要: オブジェクト指向分析設計の効果を知ることを目的とし、要求定義、要求分析、設計をオブジェクト指向で行った。要求定義では、「ユースケース図」、「ユースケース記述」の演習を、要求分析では、「シナリオ分析」、「オブジェクト指向分析」の演習を、設計では、「オブジェクト指向分析」の演習を行った。
  - 有効性: 口頭や文章よりも、多くの情報を簡潔かつ正確に伝えることが出来るため、伝達コストを軽減できる。オブジェクト指向は、問題領域の概念構造＝ソフトウェアの構造を目指しているので、解決方法が変わっても、ソフトウェアの構造に影響しない。
  - 留意点: きちんと作らないと、オブジェクト指向開発に変更しても、開発コストは下がらない。新規に作り直す時に、オブジェクト指向に変更するのが好ましい。オブジェクト指向開発に変更後、3プロジェクトは我慢しないと、開発コストは下がらない。
  - 活用事例: シーケンス図を使い、複数会社合同開発プロジェクトにて、仕様整合を行った。今までの仕様を詳しく知らない、新規で加わった人も、今までの仕様と、変更内容を理解しやすく、仕様整合あいまいによる不一致を軽減出来ている。第 5 回: モデル駆動開発、久保秋真 (株式会社アフレル)
- 第 5 回 モデル駆動開発 久保秋 真 (株式会社アフレル)
  - 概要: モデル駆動開発 (MDD) の基礎として、組込み開発を背景として、モデル駆動開発の意義、モデル駆動アーキテクチャ、MDD のない開発とある開発の比較、MOF, QVT, XMI といった要素技術、実際の MDD の流れなどを学んだ。またそれらの学習の後、演習として、ロボットを

使って実際に MDD による組込みプログラムの生成を行った。

- 有効性：MDD を利用することによって、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、コード生成のための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位置に甘んじているモデルが非常に多い。MDD を利用することで、設計レベルでソフトウェアの品質を作りこめるという、モデル本来の意義を最大限に生かすことができる。
- 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、ソフトウェアを自動で生成してくれるツールとしてではなく、コードの生成から属人性を排除し、品質を一定に保ったコードを得るツールとして認識すべきである。

- 第 6 回: モデル検査・形式検証、吉岡信和（国立情報学研究所）
  - 概要: 複雑化するシステムに対し、その仕様を網羅的かつ自動的に検証するモデル検査について、実際にツールを動かしながら、その有効性や実施方法などを学習した。
  - 有効性: 本ツールを使えば検証を自動化でき、より短時間で品質があがる可能性がある。技術やシステムの複雑・分散化がさらに進む今では、検証するためにいずれ採用される手法とも感じた。また、LTSA の場合、結果をオートマトンで表現し連携や動作が「見える化」されるため理解しやすいため、若手教育にも使えそうだった。
  - 留意点: 現在ツールを使うためには、設計仕様をツールに合わせてモデル化しなければならない。その点が大きな負担になる。モデル化する方法をより簡略化できれば（例えば、モデルへの変換ツールができる、モデル化せずにコード本体を検証できる）より利用可能性が高まるだろう。
- 第 7 回 ： ソフトウェア測定・メトリクス、野中 誠（東洋大学）
  - 概要: ソフトウェア開発において定量的な管理を行う重要さと、ソフトウェア開発での測定が品質管理での意思決定に欠かせないことを学んだ。演習では、メトリクスについて Excel シートを利用して実際的に経験した。特に、欠陥に関して、テスト管理図、欠陥発見計画、欠陥除去率、レイリーモデルによる予測について演習を行った。
  - 有効性: 過去のソフトウェア開発プロジェクトの実績を利用して、現開発での欠陥の発生状況を予測し、開発規模や実測値と照らし合わせて、欠陥が残っていないか、試験やレビューが十分か確認できる。テスト管理図を使いながら、現在進行中のプロジェクトの欠陥発生状況を把握し、それを次のプロジェクトへのインプットにして欠陥発生、欠陥除去の計画を立てることも可能である。
  - 留意点: 欠陥に関する本手法を利用するには、過去の類似の開発プロジェクトの欠陥の状況をある程度蓄積していなければいけない。各フェーズでの欠陥摘出状況、顧客への出荷後の欠陥発生状況、開発チームの欠陥是正効率を把握しておく必要がある。過去に類似のプロジェクトがないならば、注意深く予測値を設定して利用しなければいけない。
- 第 8 回: アスペクト指向開発、鷲崎 弘宜（早稲田大学）
  - 概要: ソフトウエア開発で注目されているアスペクト指向を演習中心で判りやすく教えていただき、アスペクト指向の本質が理解できる演習である。従来、ソフトウエア開発時（方式設計時）に要求に関係なく、ロギングやキャッシュの仕組み（＊1）を検討する必要があり、各クラスにハードコーディングすることになる。アスペクト指向は、極論すればビルド（コンパイル）までロギングやキャッシュに関することを検討する必要がないため、要求の本質を見定める設計と実装に必要になる情報を分けて考えられる。本演習では、AspectJ を用いてアスペクトプログラミング（ロギング実装）を演習する。（＊1）：関心事と横断的関心事がある。関心事：実装に登場する様々な観

点に基づく知識・技術横断的関心事：ロジック・ロギング・セキュリティなどのソフトウェア構造の各所に横断的に散在して実現される関心事を指す。

- 有効性: アスペクト指向は、本来の要求に対して設計に集中できる点で、非常に有効な考え方だと認識している。Java では、AspectJ を活用すればビルドに至るまで、関心事に関することは意識しないため実装中にロギングの仕組みなどを検討外に出来る点ではかなり有効だと実感した。今後は、アスペクト指向が上流工程の設計アクティビティまで発展するれば、かなり強力な指向であり非常に有効な技術となると感じた。
- 留意点: アスペクト指向の適用する場合は、IDE が重要である。IDE の強力なサポートがなければ、アスペクトは実現できない。Java の場合は、Eclipse＋AspectJ でなければデバックが難しい。また、初めてアスペクト指向を取り入れるときは、適用する範囲を狭めたほうが賢明である。理由は、機能に着目した場合、アスペクトはモジュールより粒度が小さいため情報が発散する危険がある。まずは、アプリケーションログなどターゲットを狭めて開発することを推奨する。

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。

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] 鷲崎弘宜, 情報学探求 – 大規模ソフトウェアの効率的開発技術の追求, 情報通信ジャーナル 5 月号, 2007.
[2] 阿草清滋, 西康晴, 沢田篤史, 鷲崎弘宜, 〈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.
[3] 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 .
[4] 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.
[5] IEEE CS, ACM, Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, 2004. 〔最新版は http://sites.computer.org/ccse/ より取得可能〕
[6] 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.
[7] 鷲崎弘宜, 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.
[8] 鷲崎弘宜, 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.
[9] 鷲崎弘宜, 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.