

ソフトウェア工学演習コース 2007 年度 活動報告

Report on Software Engineering Practice Course in 2007

鷺崎弘宜(国立情報学研究所) 田村一賢(東芝ソリューション株式会社)
阿部修久(キヤノンシステムソリューションズ株式会社) 安藤 元伸(ブラザー工業株式会社)
古村仁志(日本電気株式会社) 保栖真輝(日本電子株式会社)
溝口文康(株式会社アルゴ 21) 山本文彦(TIS 株式会社)
猪塚修(横河制御エンジニアリング株式会社)
および
ソフトウェア工学演習コース 2007 年度 研究員一同

2005 年度より継続して「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・先進的な種々の代表的ソフトウェア工学技術を学習した。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、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。

Following the success of previous courses in 2005-2006, the software engineering practice course has been 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. 設置背景と狙い

社会で求められるソフトウェアは、量・質・種類の全てについて増大の一途にある[1]。一方で、品質と納期に対する要求は厳しさを増している。例えば携帯電話の場合、搭載プログラムのコードサイズが 10 数年で 10 倍以上増大したのに対して、開発期間は半分程度にまで短縮されたという報告がある[2]。このような厳しい状況において、大規模・高品質・多品種・高生産を確実に追求するためには、理論や経験に裏づけされた技術の体系を適切に用いることが不可欠である。

ソフトウェア工学(Software Engineering)とは、そのような要請にこたえるための技術体系であり、「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]と位置づけられる。ソフトウェア工学を適切に用いることで、属人性を排した一定以上の品質保証(高品質)と高生産(大規模・多品種)の達成が期待できる。しかし現実には、ソフトウェア工学の習得と適切な活用は容易ではなく、その潜在的な効果を十分に産業において享受できていないと考えられる。例えば、ソフトウェア開発プロジェクトの品質やコストおよび納期に関する発注者からみた成功率が 30% を切るという報告[4]があるなど、今日のソフトウェア開発には工学の適用による改善の余地が十分にあると想定される。

このような問題意識から、本コースは 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た([5][6]を参照されたい)。そこで 2007 年度も引き続いて、産学両面に通じた講師をお招きし、計 17 名の研究員が参加して、全 10 回にわたり実践的・先進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、本コースの構成と目標の達成度合い、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、研究員の分担執筆による貢献が大きい。

2. コースの設計と自己評価

本コースの設置にあたり、ソフトウェア工学を習得し実践する上での問題を考察し、その解決に向けて内容および構成を設計した。考察した問題 P1~P6、および、本コースの設計および実施における取り組

み S1～S6 をそれぞれ対応付ける形で表 1 にまとめた。

表 1: ソフトウェア工学の習得と実践に向けた問題と本コースにおける解決

習得と実践に向けた問題	コース設計上の解決
<p>P1: ソフトウェア開発はソフトウェア、ハードウェア、ネットワーク、外界・環境、組織・プロジェクト・人間などが関与する複雑な知的活動である。従ってソフトウェア工学の扱う領域および関連領域は広大であり、必要な種々の科学技術・工学技術・知識を体系だって把握することを難しくしている。</p>	<p>S1: SWEBOK (Software Engineering Body of Knowledge) [1]を用いてソフトウェア工学領域全体および関連領域を俯瞰し、知識体系に対応付ける形で主要なソフトウェア工学技術を整理し、演習で取り上げるように設計した。対応付けた結果を図 1 に示す。SWEBOK は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位置づけの確認に適している。図 1 より本コースは SWEBOK の領域を、知識領域のレベルで広く扱っている。副知識領域のレベルでは、個別の演習内容が代表的な技術/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。</p>
<p>P2: 工業化を推し進める中で、主として大規模化に起因して工程に対応した専門化が進められつつあり、例えばテスト技術者はテスト工程とその周辺のみ、プログラマは実装工程とその周辺のみという具合に、ライフサイクルプロセスの全体を通じた技術・知識の見通しを悪くしている。</p>	<p>S2: 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技術を学習できるように演習を配置した。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。</p>
<p>P3: 技術革新の早さがある。本質的で不変な技術・知識が多数ある一方で、特に実装や運用について基礎技術の蓄積の上で日々革新があり、最新の技術動向の把握は容易ではない。</p>	<p>S3: 要求工学やテストといった普遍的で実装によらない実践性の高い技術に加えて、モデル駆動開発や形式検証といった最新の先端的技術もあわせて取り上げ、最新動向も把握できるように構成した。前者に該当する 7 種(要求工学、設計レビュー、品質要求定義、オブジェクト指向分析設計、テスト、アジャイル開発、組込みソフトウェア開発プロセス構築)について、各研究員は実適用に向けた技術習得に成功したと考えられる。後者の 3 種(モデル駆動開発、形式検証・モデル検査、アスペクト指向プログラミング)について、研究員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基礎形成に成功したと考えられる。</p>
<p>P4: 「産業—アカデミア・ギャップ」[7]も一因である。あらゆる学問と同様にソフトウェア工学の進展にはアカデミアと産業界のシナジーが不可欠であるが、アカデミアが実践性を追及できず、一方で、産業界ではソフトウェア工学の習得と導入が欠けて</p>	<p>S4: 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的技術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。</p>

いると指摘されている[7][8]。	
P5: P1 に関連して、ソフトウェア開発の難しさには人的要因や要求・問題の多様さ(ビジネスの多様さ、システム境界の曖昧さ)などが関係しており、プロジェクトや組織を超えて難しさやノウハウを共有することがときに困難である。	S5: 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(研究員を変更しての)グループ演習を通じて、同じような問題意識や目的を持った仲間作りに成功した。また、参加研究員の立場は様々であり、異なる立場、異なる視点の意見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。
P6: ソフトウェアは本質的に質量を有さず不可視[9]であるため、他の工学分野(例えば建築)とは異なり、その原理や働き・ダイナミクスの把握がときに難しい。	S6: 取り上げる全ての技術について手を動かす実習を、実用的なツールも交えて徹底的に実施し、その原理や働きを体感すると同時に実践に向けた適用力を養った。また、複数の回(要求工学、オブジェクト指向分析設計、モデル駆動開発など)において不可視性の解決に有効な UML[10]やモデリング技術を重点的に活用した。

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 構造とアーキテクチャ 設計品質の分析評価 設計のための表記 OO分析設計, モデル駆動開発 設計戦略および手法	構築の基礎的概念 形式検証 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 アジャイル開発 計量	プロセス実現および変更 プロセス定義 プロセス構築 プロセスアセスメント プロセス計量	ツール 開発手法 (各回での扱い)	品質の基礎的概念 品質要求定義 マネジメントプロセス 実践上の考慮事項

図 1: 演習内容と SWEBOK における知識領域の対応

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と自己評価している。

3. 演習の概要と気づき

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技術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。

- 第1回: 要求工学: 中谷 多哉子(筑波大学)
 - 概要: 真の要求が何なのかを分析する技法に触れるのが目的であった。演習では「会議を円滑に運営するシステムを構築する。」といったテーマが与えられ、どんなシステムを構築すべきかを分析していった。手法として、リッチピクチャとCATWOE分析といったものを活用し、「誰のどのような問題が存在し、それをどのようにすると会議が円滑になるといえるのか。」を分析していった。
 - 有効性: この演習により、ひとつの状況の中にも立場の違いにより、さまざまな世界観、価値観が存在することが認識できた。その中のどの要求をどのように満たすことが、本来おこなうべき事なのかを考えるべきであるということが体験できた。これは、研修の翌日から活用できる内容である。技法の習得よりも、自分自身の中で意識をすることが非常に重要で、それに気づくことができたことが一番の有効性があるポイントであるといえる。

- 臨時会: ソフトウェア品質と設計レビュー : 猪塚 修(横河制御エンジニアリング株式会社)
 - 概要: どのような視点でのレビューをおこなうべきかを学ぶことが目的であった。演習では「簡単なWEBメールシステムを新人がシステム設計をおこなった際に、レビューするポイントをあげる。」といった内容で実施した。まずはレビューポイントをポストイットに次々に記入していく作業を実施した。次にISO9126[11]の品質特性を意識してレビューポイントを増やしていく作業を実施した。
 - 有効性: この演習によりISO9126といったレビューの視点が存在することの有効性を体験することができた。さらにはレビューチェックリストを活用することで、ある一定の品質を確保することができることを学ぶことができた。先人達が作り上げた知識の結晶であるレビューチェックリストを、うまく活用していくことがいかに無駄のないことなのかを実感できたのである。演習後、研究員自身が実務において演習内容をレビューポイントの明確な定義に活用し、レビューアが複数人に分散しても一定レベルの品質を確保することに成功した。

- 第2回: 品質要求定義: 東 基衛(早稲田大学)
 - 概要: 不明確になりがちな品質要求を、マトリクスを用いて定義するのが目的であった。演習では「シナリオから機能要求と品質要求を定義し、対応する品質特性と優先度を定義していく。最終的には品質評価のための評価基準の定義と測定法を定義する。」といった内容であった。
 - 留意点: 実現可能性の検討なく顧客の要望を全面的に受け入れて品質要求を明確にしすぎると、開発者自身の首を絞めるケースがないとも限らないため、品質要求定義の技術を適材適所に活用することが重要と考えられる。

- 第3回: オブジェクト指向分析/設計: 井上 樹(株式会社豆蔵)
 - 概要: オブジェクト指向分析と設計を体験してみるというのが目的であった。時間も多く必要とする内容であるため、2日間にわたり合宿で演習がおこなわれた。演習では「自動ドアの制御システムをオブジェクト指向で分析し設計していく」といった内容であった。成果物として、UMLのユースケース図、ユースケース記述、分析モデルのクラス図・シーケンス図、設計モデルのクラス

図・シーケンス図を記述していった。

- 有効性: この演習により、オブジェクト指向とはどんなものなのか、どんな点が有効なのか、どんな点に注意すべきなのかを体感することができた。講師がオブジェクト指向の世界でも有名なコンサルタントであったことが、演習を一番有意義なものにしたと感じた。ただのUMLの研修ではなく、「オブジェクト指向とはこうすべきです」と芯のある内容で、研究員達の質問にも一貫して明確に答えていたのが、演習全体を通して説得力を感じることができた重要なポイントである。
 - 留意点: 今回の演習で、講師の言葉だけではなく、オブジェクト指向への移行が簡単なものではないことも体験することができた。組織として取り組むことの重要性が強く感じられ、その体制が整わないかぎり失敗することが見えているので、それまでは実用しないという導入方法が考えられる。
- 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)
 - 概要: モデル駆動開発のメリット及びモデル変換の基礎知識、技術用語に関する講義を受けた後、演習では自律走行ロボット制御ソフトウェアのプラットフォーム独立モデルから依存モデルへの変換及びコード変換を実施した。
 - 有効性: 演習を通じてモデル駆動開発の基本的な概念を理解した。現在実務で問題となっているプログラミング工程の非効率及びコード品質の属人性をモデル駆動開発により解消または低減可能であることを習得し、モデル駆動開発の有用性と将来性を感じた。
 - 留意点: 実務での実用化はまずモデリング技術の習得が前提となり、その先のモデル駆動開発は敷居が高いと感じられる。しかし演習を通じてモデル駆動開発はツールを利用せずともExcel や PHP を利用してモデルをソースに変換可能であると理解できた。モデル駆動開発の利点を理解した上で面倒くさい仕事を自動化できるよう努めていきたい。
 - 臨時会:形式検証・モデル検査:吉岡 信和(国立情報学研究所)
 - 概要: ソフトウェアの仕様に対して、実行可能なパターンを網羅的かつ自動的に検証する手法であるモデル検査を、複数人によるスキャナとプリンタの取り合い、HDD レコーダから DVD レコーダへのコンテンツのダビングの2種類の比較的身近な題材を用いた演習を通じて習得した。
 - 有効性: ネットワーク家電を代表とする複雑化、大規模化、分散化してきているソフトウェアでは、多種多様な外部環境、様々な動作パターンを考慮する必要があり、人海戦術によるテストだけで品質を確保するには限界があり、モデル検査を利用することで解決できる可能性があることを確認できた。
 - 留意点: モデル検査の導入には高いスキルが必要なため、導入の敷居が高いが、将来においては導入したい技術であり、本コースの内容に部分的に協力している国立情報学研究所トップエスイー(R)プロジェクト[8][12]においてもモデル検査講座が開講されており、今後の展開に期待したい。
 - 第5回:ソフトウェアテスト:鈴木 三紀夫(TIS 株式会社)
 - 概要: 演習問題を通してソフトウェアテスト技法の基礎を学ぶことが目的であった。具体的には、演習を通して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、デシジョンカバレッジ)、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、デシジョンテーブルテスト)を学んだ。
 - 有効性: テスト項目の洗い出し作業は我々が想像するよりも個人間のバラツキがある。そのバラツキを軽減させながら効率良いテスト設計をするためには、細かなルールの設定が必要になってくる。これらのルールを文章化し言葉でプロジェクトメンバへ伝えるのは困難で、その解決策として今回の課題をメンバと共に取り組むことが有効だと感じた。

- 留意点: 実運用にあたって、スキルや技術的な問題は特に無い。問題があるとするならば、テスト項目のバラツキに関する話題をどこで切り出すかということである。プロジェクトの発足時、チーム育成の一環として取り組むことが理想と思われるが、実データに基づく分析結果があると意図が伝わり運用し易くなると感じた。
- 第6回: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: 日々刻々と変化するソフトウェアに対する要求に対して、柔軟に対応できる手法として用いられつつあるアジャイル開発に関して、基本的な概念、手法やマインドを工作で庭を作成するといった形で学んだ。
 - 有効性: 通常イテレーションを繰り返すのは時間的に無理があるが、工作という一見ソフトと関係の無いものを利用して行ったことによって、機能の分割や要求に対する柔軟な対応の仕方などが学べた。また、実際のソフトウェア開発でもペアプログラミングや XP (eXtreme Programming) などのプログラミング方法などがあるが、共同で作業することによる品質の偏りが低減されることを認識できた。また、決まった手法ではなく如何に柔軟に顧客の要求を満足させられるかという観点で作業を進めるといった視点が現代の製品に対するニーズにマッチしていると感じられた。
 - 留意点: 顧客要求に柔軟に対応するという事はいいことである反面、顧客要求を際限なく受け入れるということにもなりかねない危うさがあるように感じられた。この辺りの駆け引きが難しい可能性があると思われる。また、顧客要求が変化しないようなプロダクトについては従来のウォーターフォールに基づく開発の方が、全体像が見えるのでより顧客に安心感を与える場合も存在しうると考えられる。
- 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、横河デジタルコンピュータ株式会社)
 - 概要: 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 [13])をベースおよび参考として、組込みソフトウェア開発における現状や開発プロセスの構築における留意点について紹介を受けた。また、開発に考えられるリスクの分析や最適な開発プロセス構築に向けたアクティビティの抽出などを演習を通して学習した。
 - 有効性: 演習は組み込みソフトウェアについての内容であったが、特に組込みソフトウェアに限らず適用や実践が可能と考えられる。どのような要因が品質に影響するかについての分析方法や、開発プロセスをどのように構築・運用すればよいかについてのアイデアは様々な局面で活用できると考える。
 - 留意点: ESPR は組込み領域に限らず幅広く適用可能な場合があると考えられるが(例えばエンタープライズシステム開発)、その適用にあたっては組込み開発との共通性や相違に留意する必要がある。また、ESPR ではアクティビティやプロセスの種類について国際規格である ISO/IEC12207[14], 15288[15]との関係を整理しているが、収録している成果物テンプレート等についても IEEE 等の標準への適合性を整理できると、より即効性が高まる可能性がある。
- 臨時会:アスペクト指向プログラミング:鷺崎 弘宜(国立情報学研究所)
 - 概要: アスペクト指向の概念、並びに、それがオブジェクト指向のどのような問題点を解決するかについて講義を受けた後、Java のアスペクト指向環境である AspectJ を用いて、演習をおこなった。演習では、ソースのいたるところに埋め込まれているロギング処理を、AspectJ によって一元化するという作業を通じて、その効果を確認するというものであった。
 - 有効性: アスペクト指向は、プログラムに割り込みをかけ、共通処理(アスペクト)を挿入するという仕組みが必須となる為、この機能を実現するための優れた環境が必要である。これについて、

AspectJ は、既存のプログラムを一切変更することなく、さまざまな条件設定に基づく割り込み位置指定が行える為、十分な実用性が確保できていると感じた。

- 留意点: 強力な割り込みが行える半面、チームとして活用する場合は、何らかの統制を行わなければ、逆に混乱を引き起こす可能性がある。これについては、利用方法に関するベストプラクティス、アンチプラクティスの蓄積・共有化や、プロジェクト内で利用規約を制定するなどの活動が必要と感じた。

4. おわりに

本コースでは狙い(1章)にあるように、指導講師による講義と演習を通じて、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と発表を行うことで、より深い理解を促した。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。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] 鷺崎弘宜, 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] 平山雅之, 組込みシステム開発における品質向上の施策, ET2002 TB-6, 2002.
- [3] 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. (最新版は <http://www.swebok.org/> より取得可能)
- [4] 中村建助 他, プロジェクト成功率は26.7%, 日経コンピュータ2003年11月17日号, 日経BP, 2005.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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 (ICSE), ACM Press, 2007.
- [9] F.P. Brooks, The Mythical man-Month : Essays on Software Engineering, Addison-Wesley, 1975 (邦訳『人月の神話 - オオカミ人間を撃つ銀の弾丸はない』)
- [10] OMG, "UML 2.0 Specification," 2004, <http://www.uml.org/>
- [11] ISO/IEC9126:2001, "Software engineering - Product quality - Part1: Quality model", 2001.
- [12] 国立情報学研究所トップエスイープロジェクト, <http://topse.jp>
- [13] 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.
- [14] ISO/IEC 12207: Information technology -- Software life cycle processes, 1995.
- [15] ISO/IEC 15288: Systems Engineering - System Life Cycle Processes, 2002