

## ソフトウェア工学演習コース 2008 年度 活動報告

### Report on Software Engineering Practice Course in 2008

鷲崎弘宜、城間祐輝(早稲田大学) 田村一賢(東芝ソリューション株式会社)  
溝口文康(株式会社アルゴ 21) 大橋剛和(富士通株式会社) 覚井真吾(株式会社東芝)  
白井孝明(ブラザー工業株式会社) 草場康男(株式会社 CIJ) 松宮宏明(株式会社インテック)  
安藤良治、佐藤和人(アンリツエンジニアリング株式会社) 柴田和也(ソニー株式会社)  
實藤博(セイコーエプソン株式会社) および ソフトウェア工学演習コース 2008 年度研究員一同

「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。コースとしては 2005 年度から継続的に設置して 4 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について報告する。

Following the success of previous courses in 2005-2007, 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. コースの狙い

今日の社会においてエンタープライズシステムから組み込みシステムに至る隔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(高品質)と高生産(大規模・多品種)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]を参照されたい)。そこで 2008 年度も引き続いて、産学両面に通じた講師をお招きし、計 15 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

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会得
- 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に有効であったと自己評価している。また、それを通じた開発強化のための基盤形成についても、次章で述べるように有効と判断された技術が多数であり、一定の達成をみたと評価している。さらに、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。

| 要求                                                                                                  | 設計                                                                                                                           | 構築                                                             | テスト                                                                     | 保守                                                               |
|-----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|-------------------------------------------------------------------------|------------------------------------------------------------------|
| 要求の基礎的概念<br>要求エンジニアリングプロセス<br>要求の抽出<br><b>品質要求定義</b><br>要求仕様<br><b>要求工学</b><br>要求の妥当性確認<br>実践上の考慮事項 | 設計の基礎的概念<br>設計における主要な問題<br><b>アーキテクチャ</b><br>構造とアーキテクチャ<br><b>モデル検査</b><br>設計品質の分析評価<br>設計のための表記<br><b>分析設計</b><br>設計戦略および手法 | 構築の基礎的概念<br><b>評価</b><br>構築の管理<br><b>モデル駆動開発</b><br>実践上の考慮事項   | テスティングの基礎的概念<br>テストレベル<br><b>テスト</b><br>テスト技法<br>テストに関する計量尺度<br>テストプロセス | 保守の基礎的概念<br>保守プロセス<br>保守における主要な課題<br>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<br>構成の識別<br>構成制御<br>構成状態記録および報告<br>構成監査<br>リリース管理および配布                               | 開始と範囲定義<br>プロジェクト計画<br>プロジェクト実施<br>レビューおよび評価<br><b>アジャイル開発</b><br>終了<br>計量                                                   | プロセス実現および変更<br>プロセス定義<br><b>プロセス構築</b><br>プロセスアセスメント<br>プロセス計量 | ツール<br><b>(各回での扱い)</b><br>開発手法                                          | 品質の基礎的概念<br><b>品質要求定義</b><br>マネジメントプロセス<br><b>測定</b><br>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

## 3. 各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、さらには自身における姿勢などの変化や必要なアクションなど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。7回の定例会(第4回はソフトウェア品質シンポジウムのため演習なし)、および、3回の臨時会を加えた計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。

- 第1回:ソフトウェア品質要求定義:東 基衛(早稲田大学)
  - 概要: 情報システムを提案する、という演習を通して、機能要求・品質要求を整理していく技法を学んだ。提案する情報システムの概要は、事前に宿題としてまとめておき、当日の演習ではシナリオ化、機能要求定義、品質要求明確化、品質を確認するメトリクスと測定法の定義、という手順で要求仕様を明確化していった。
  - 有効性: 品質要求を定義するにあたって、ISO/IEC 9126-1 で定義されている品質特性を参照した。うっかりすると、機能面以外の品質要求は見落としがちだが、体系付けられている品質モデルを参照することで、見落としを防ぐことができる。また、演習の進め方として、班分けによって受講生同士の議論ができたことで、普段とは違った視点で品質を捉える機会ができた。
  - 留意点: 最初のシナリオ抽出の段階で、システムに対するステークホルダをうまく洗い出すことがポイントである。これについては体系付けられた分類があるわけではないので、ここで見落とすと最後まで気が付かない。この段階では、とりわけ複数の人の目によるレビューが重要だと思われる。また、機能面以外の品質のメトリクスと測定法を見つけるには多少のアイデアが必要である。そのため、多くの人と議論をしたり、経験を積んだりすることも重要であると感じた。
- 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「会議を円滑に運営するシステムを構築する。」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を行った。
  - 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、様々な世界観があり、その世界観が異なるため、ステークホルダのシステムへの要求が異なり、仕様(基本的)を固めることを困難にしていることが理解できた。言いかえれば、世界観(W)の違いに基づくものと分かれば、ユーザ側で調整可能であり、その世界観抽出にCATWOE分析は有効であると判断する。
  - 留意点: ゴール指向分析において、パーティションで区切られたOwner間のゴールの貢献度を、ゴールモデルだけで判断する事は困難であり、当事者(Owner)を集め、ゴールモデル図の説明の中で、Owner自身が気づくか、あるいはシステム屋が気づくことが重要であると分かった。
- 第2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。演習では「自動ドア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」といった内容であり、成果物として、UMLのユースケース図、ユースケース記述、設計モデルのクラス図を記述した。
  - 有効性: 演習により、目的に応じて絵や図を使うことで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できることを実感でき、モデリング/オブジェクト指向とはどのようなものであり、導入することで何がうれしいのか、どのような点に注意すべきなのかを体感し、理解することができた。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り、ただのUMLの研修ではなく、「オブジェクト指向とはどういうものか」を芯のある内容で、説明して頂き、説得力を感じることができた。
  - 留意点: オブジェクト指向を使う場合、組織としての体制や、推進する体制が必要であり、その体制が整わないかぎり失敗する確率が高い。また体制の中には、オブジェクト指向の経験者の参加が必要であり、リーダーシップが求められる。オブジェクト指向を導入することで、大きな効果を得られるには、さまざまなノウハウを蓄積・共有・再利用を行えるまで、多少の時間が必要で

ある。

- 第3回:アーキテクチャ設計評価:鷺崎 弘宜(早稲田大学)
  - 概要: 品質要求に着目したアーキテクチャ設計と評価方法を体験することが目的であった。演習では、「遠隔監視システム」の要求から品質シナリオを記述することによって、ADD (Attribute Driven Design)を用いてアーキテクチャ設計をし、ATAM(Architecture Trade-off Analysis Method)を用いて品質特性を中心としたアーキテクチャの評価を行った。
  - 有効性: この演習によって、品質要求(例えば実装技術上の制約など)を満たす堅実な設計と早期リスクの特定、アーキテクチャの本質的な機能と特性をステークホルダと共有できることに有効性を感じることができた。また、通常、機能要求を満たすため、品質に関する要求はアーキテクチャ決定時のトレードオフによって暗黙知となりがちだが、ここで学んだことを実践することによって、暗黙知を撤廃し、前工程での品質確保や、後戻り工程の削減に貢献できるのではないかと考えている。なお、これは、期間の厳しい開発であればあるほど有効に機能するのではないかと考える。
  - 留意点: ADD のアプローチは、各品質特性と紐づけられたアーキテクチャパターンに精通していると、より効果が発揮されるため、経験の少ない組織やチームでは、学習が必要だと考えられる。また、ATAM においては、経験に依存した主観的な評価手法であり、ステークホルダともリスクを共有することが重要であるため、組織ぐるみでの体制構築などが必要であると考えられる。
- 臨時会: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: モデル駆動開発の基礎知識として、抽象化したモデルからモデル変換を繰り返しコードや成果物を自動的に生成する仕組みに対する講義と自律走行ロボットのクラス図を作成し、モデル変換を繰り返しソースコードに自動変換されるまでの過程を、演習を通して体感した。
  - 有効性: モデル変換はプラットフォーム非依存モデル→プラットフォーム依存モデル→ソース・コードへとそれぞれの変換規則により自動変換するため、設計からコードまでの一貫性が確保でき生産効率が高いことが理論上理解できた。さらにコード変換規則にコーディング作法を組み入れればコード品質を均一に保つことが期待できる。コード変換規則を確立すれば、従来開発に比べ短納期での開発が可能であると思われる。
  - 留意点: プロジェクト適用時には、教育の実施や習熟期間、コード変換規則の確立に多大な時間を要するため生産性が低下しプロジェクトへの負担が大きくなると考えられる。プロジェクト規模を考慮しプロジェクト全体への適用とするか特定の機能やプログラムに限定して適用するかを考慮すべきである。演習を通じて、今後の開発における設計、製造、テスト工程において「自動化は可能か?」という思考を持つことが大切であると感じた。
- 第5回:ソフトウェアテスト:鈴木 三紀夫(TIS 株式会社)
  - 概要: ソフトウェアテスト技法の基礎を、制御パステスト、同値クラステスト、ドメインテスト、デシジョンテーブルテストに関する演習を通して習得した。
  - 有効性: それぞれの演習におけるキーワードは以下の通り。制御パステスト(C0、C1 カバレッジ、目標カバレッジ率)、同値クラステスト(クラスの分け方、代表値の選択方法)、ドメインテスト(Weak Point、On point、Off point、Strong Point)。当演習における問題は、基礎的な問題であったが、解となるテストケースは、1 つではなく、テスト設計(戦略)に依存するものであった。それぞれのテスト技法について理解することも重要ではあるが、当該製品の品質目標に対して、どのようなテストでどうテストを行い、品質を確保していくのがポイント(有効)であると、演習を通して学んだ。
  - 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、テスト設計(戦略)を詳細に行い、いかにテストを行うかである。この度学んだソフトウェアテスト技法を駆使し、テスト戦略、テストケース等を明確にする必要があると考える。
- 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: 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正多面体を作成するといった形で学んだ。
- 有効性: 開発における問題点(Problem)を見つけ、その改善案を試行(Try)、よいものは継続(Keep)の PTK のサイクルを繰り返すことで注意点や失敗、メンバーの意識などの共有にもつながることを実感できた。ウォーターフォール型の開発とは異なり、レビューを早い段階で行うことにより、顧客とのギャップをうめやすく仕様変更の要望や、仕様の欠陥にも対応しやすいと感じた。また、新人など不慣れなメンバーの教育を初期の段階で行うことにより、後半にその教育の成果が生かせると感じた。
- 留意点: 顧客の要求をどこまで受け入れるかという問題がある。どの時点でどの程度の仕様変更を受け入れるかといったことをあらかじめ決めておくなどの対応が必要だと感じた。また、ウォーターフォール型の開発よりも個人の負担が増すと考えられるため、一人ひとりの能力が重要になってくると考えられる。この点から、ウォーターフォール型の開発に比べ、アジャイル開発は個人の能力を生かす開発手法で、小規模の開発に向いていると考えられる。PTK のサイクルでは、問題点を挙げるのが重要になるため、積極的に問題点を挙げるのできる雰囲気作りが必要であると感じた。
- 第 6 回: モデル検査による形式検証: 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、仕様上可能な振舞いを網羅的かつ自動的に検証するのに有効な手法であるモデル検査について学んだ。
  - 有効性: 演習では実際にモデル検査ツールを用いて分散システムを例としたソフトウェアの様々な振舞いのパターンを自動的に検証した。その結果、デッドロック等の不整合を効果的に検出することができ、その有効性を確認できた。これは近年、大規模化・複雑化・分散化しているソフトウェアに対して、非常に有効な検証方法であると感じた。
  - 留意点: モデル検査ツールを使う際は、ソフトウェアの仕様をモデル記述言語に変換する必要がある。いかにモデル検査ツールが仕様の検証に効果的であっても、この変換作業でミスや漏れがあっては効果は期待できない。今後は変換作業においてミスや漏れをどのように減らしていくかが課題であると考ええる。
- 第 7 回: ソフトウェア測定: 野中 誠(東洋大学)
  - 概要: 普段何気なく実施している「測定する」ことの重要性、また、プロダクトとプロセスの代表的なメトリクスとその測定方法、管理方法、測定時の留意事項などを習得した。演習ではインスペクションの有効性・費用対効果の評価、実際の開発プロジェクトで測定した"共有された価値観"の結果などの定量的管理に関して意見交換を行った。
  - 有効性: 本演習コースで繰り返し学んできた品質を定量的に管理・評価することの大切さを改めて学んだ。品質(顧客満足)と開発期間の短縮をコントロールするための定量的管理方法、プロジェクト管理の状況判断に有効な各種手法を得ることができたことは有効だった。
  - 留意点: ソフトウェア品質の定量的な測定は重要である一方、測定対象の曖昧さ、データ入力の不安定さ、開発現場へのフィードバックの難しさがある。定量的管理に対する組織・メンバーのコミットをいかに取り付けるか、データや結果・評価をいかに共有するかが測定効果を高めるポイントであり、プロジェクト成功率の向上への鍵と考える。
- 第 8 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: 講義資料に加え書籍[9][10][11]を用いて講義・演習が展開された。その講義内容は、「組込みシステム」における「ソフトウェア開発」の現状が近年の事例や講師の経験を交えて解説された。その中で「組込みソフトウェア開発」における課題やリスクの傾向が示され、それを解決する為の"気づき"が演習体験や講義資料、更に講師の生の声を通して与えられた。そして、

その具体的な解決策として ESPR[9]が紹介される。また、初版発行間もない ESQR[10]に関する講義も展開され該分野での先進的な取組みに触れる機会も得られた。

- 有効性: 「ソフトウェア開発」の最前線にいる技術者にとって「プロセス」=「面倒臭いもの」と映る場合が多い。もし、現場の技術者が本講義を受講したならこの誤った認識を解き、「開発プロセス」の意味と必要性を理解できる。逆に、「プロセス」を管理・運営する立場の人であれば、現場を疲弊させる「開発プロセス」の構築をしては意味が無いこと。無理のない最適な「開発プロセス」を構築する事の大切さが理解できる。また、ガイドラインとして ESPR を利用する事で「開発プロセス」の構築が遅れている現場でも比較的容易に体系的な「開発プロセス」が得られると考えられる。ESPRは「組込みソフトウェア向け」であるが、関連する規格(ISO/IEC12207,15288等)は「組込みシステム」だけを対象にしているものではない。また、「組込みシステム」と言っても近年は多様化、大規模化が進み様々な性質のものが存在するが、これらに対応できるように ESPR は考慮されているようである。この事からも他ドメインでの開発において ESPR をガイドラインとして活用する事は有効と考えられる。
- 留意点: 「開発プロセス」を構築する事が目的ではない。それを定義する事で開発者にとって安心・安全な開発工程を手に入れる。その結果としてプロダクト品質の向上につなげる事が真の目的である。開発現場に合致した適応範囲を見極め、いたずらに利用者の拒否反応を引起す事のない様に「開発プロセス」を創り出す必要がある。

#### 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も、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] 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.
- [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 一, 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] 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.
- [8] 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.

- [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 改訂版 (Embedded System development Process Reference: ESPR ver2.0), 翔泳社, 2007.
- [10] 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開発向け品質作り込みガイド (Embedded System development Quality Reference: ESQR), 翔泳社, 2008.
- [11] 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, 組込みシステムの安全性向上の勧め, オーム社, 2006.