

スーパーSEセミナーのインストラクショナルデザイン： IT技術者のための系統的教育システム

Instructional Design of the Super SE Seminar : A Systematic Education System for Professional IT Engineers

掛下 哲郎
Tetsuro Kakeshita

大月 美佳
Mika Ohtsuki

佐賀大学 理工学部 知能情報システム学科
Department of Information Science, Faculty of Science and Engineering, Saga University

要旨

日本のソフトウェア産業は国際競争力があるとは言えない。その大きな原因としては、系統的なソフトウェア教育を受けたIT技術者が大幅に不足していることが挙げられる。我々は2004年4月から現役IT技術者を対象としてスーパーSEセミナーを開講している。スーパーSEセミナーは、ACM等が提案しているCCSE2004カリキュラムに準拠しており、教育プログラムの構築には、佐賀大学におけるJABEE認定プログラムの構築・運用経験が活かされている。本稿では、スーパーSEセミナーにおける教育システムの設計について報告する。

1. はじめに

日本のソフトウェア産業は国際的競争力があるとは言えない[1]。ソフトウェアの輸出は、ソフトウェア輸入の1%程度に留まっており、日本国内のソフトウェア開発案件が中国等にアウトソーシングされる事例や、国内における各種ITプロジェクトの開発を外国企業が受注する事例も増えている。(社)情報サービス産業協会(JISA)が会員企業に対して行なったアンケート調査[2]では、受託ソフトウェア開発において、納期は一応守られているものの、プロジェクトの見積り、計画、管理には重大な課題が潜んでいる可能性が指摘されている。

上述の問題が発生する根本的な原因としては、ソフトウェア技術者に対する専門教育の不足が挙げられる。我々は、ソフトウェア技術者の技術力不足問題に対応するために、2004年度からスーパーSEセミナーを開講している[3]。スーパーSEセミナーを運営するに当たって、我々は、系統的な教育システムを構築することが重要だと考えている。これにより、共通化された教育コンテンツやカリキュラムを活用して各種のセミナーを実施することや、セミナーの実施を外部に委託することも可能になる。本稿ではインストラクショナルデザイン[4]の手法を活用して教育システムを設計する。

以下、2~3節ではセミナー受講者のプロフィール分析および企業のニーズ分析を行い、教育目標および達成度レベルを設定する。4~5節では、設定した教育目標を達成するための教育コンテンツおよび演習課題を設計する。6節では達成度評価および教育システムへのフィードバックについて検討する。

2. 受講者プロフィールとニーズの分析

スーパーSEセミナーを受講した現役のソフトウェア技術者の特徴を以下にまとめる。

- (1) 一定の実務経験があり、それぞれの業務の中で異なる経験を積んでいる。
- (2) コンピュータサイエンスやソフトウェア開発に関する系統的な教育を受けていない受講者が多い。
しかし、自らの知識を確認するためにセミナーを受講するレベルの高い受講者も一部存在する。
- (3) 学んだことを実務に活用することに対する意欲が一般的な大学生と比較して強い。
- (4) セミナー受講に割ける時間が限られている。

上記を考慮すると、セミナーの長さは比較的短期(例:5日間×7~8時間=35~40時間)のものとし、座学のみでなく実技を含めることで、実務への活用を容易にできる。また、実技の中にグループ討論やロールプレイングなどを含めると、各受講者の経験から他の受講者が学習できるため、相乗効果が得られる。当然のことながら、短期セミナーの中で系統的な教育を行なうためには、座学と実技の連携が重

要になる。また、短期セミナーで教育効果を上げるためには目的を限定する必要がある。

JISA の調査によると、ソフトウェア企業で不足している人材としては、プロジェクトマネジメント (91%)、セールス・マーケティング (60%)、コンサルタント (40%) などが挙げられる[5]。これらの業種で不足しているスキルに関する調査結果は以下のとおりである。なお、プロジェクトマネージャが品質管理の中心人物を兼ねているケースも多い。

プロジェクトマネジメント	問題分析能力・問題解決能力 (40%)、リスク管理能力 (19%)
セールス・マーケティング	提案・企画力 (72%)、顧客とのコミュニケーション能力 (20%)
コンサルタント	問題分析能力・問題解決能力 (30%)、提案・企画力 (25%)

この調査結果から、問題分析能力、問題解決能力、提案・企画力などの重要性が高いことが分かる。

また、大学の情報工学系教育への期待を見ると、システム設計・ソフトウェア設計 (78.9%)、通信・ネットワーク (59.8%)、プロジェクトマネジメント (59.8%)、プログラミング技術 (55.0%)、ソフトウェア工学 (39.2%)、コンピュータサイエンス (35.9%) の順になっている[6]。

3. 教育目標と達成度レベルの設定

2 節で行った受講者の分析およびソフトウェア企業のニーズ分析に基づいて、教育分野を以下のように設定する。このうち、1 と 2 は 3~4 の前提である。また、1~4 は 5 の前提となっている。1 回のセミナーでは、5 つの教育分野の中から 1~2 分野程度を教育する。

1. ソフトウェア技術者としての基本的な習慣
2. コンピュータサイエンスと系統的プログラミング
3. ソフトウェア設計
4. ソフトウェア品質保証
5. 問題分析と問題解決、自ら学ぶための技術

Bloom は認知領域に基づいて教育目標を知識 (Knowledge)、理解 (Comprehension)、応用 (Application)、分析 (Analysis)、統合 (Synthesis)、評価 (Evaluation) の 6 つに分類した[7]。情報処理学会では、これを参考にして、大学学部における達成度レベルとして以下のレベル 0~5 を設定している。しかし、学習前の受講者のレベルが 0~4 の間で広く分布していること、比較的短期間のセミナーであり、本格的な実務においても、作業の方向性はプロジェクト内で与えられることを考慮して、セミナー終了時の達成度レベルとしては、レベル 3~4 を設定する。

レベル	表記例	意味
0	聞いたことがない	その項目について受講生は何も説明できない。
1	..を知っている ..の知識を持つ	該当項目について講義で聞いていて、その概要を述べることができる。
2	..を説明できる ..を理解している	該当項目を知識として理解しており、その知識を説明できる。または、具体例を挙げるができる。
3	..を使用できる ..を実行できる	概念と使い方が分かるレベル。限られた条件下 (演習課題等) で使っている。または、具体的な指示を与えられれば、該当項目を実行できる。
4	..を活用できる ..を..に応用できる	詳しく理解し応用できるレベル。現実に近い状況で応用できる。大まかな方向性を与えられれば、該当項目を実行できる。
5	..に熟達している	実務 (現実社会の問題解決) に応用している。

4. 授業コンテンツの設計と開発

3 節で設定した教育目標に対応した教育コンテンツを整備する際には、ACM と IEEE Computer Society が共同で開発した CCSE2004[8]、およびそれを日本向けに改良した Jpn1[9]が参考になる。これらのカリキュラムは大学学部レベルを想定しているが、ソフトウェア工学に関連する知識を体系的にまとめているため、スーパーSE セミナーでは、これらのカリキュラムに実務的な内容を追加して授業コンテンツを

開発している。以下の表では、CCSE2004の知識区分に従って授業コンテンツの概略を示す。

コンピュータサイエンス	アルゴリズムの設計と解析, 系統的なアルゴリズム作成, 高水準データ構造, コーディング標準, Java/C++プログラミング, ソフトウェア開発環境 (Eclipse, Ant, CVS, JUnit, PostgreSQL, Visual Studio.Net 等), DB, SQL, OS, コンピュータネットワーク
技術者としての能力	ロジカルシンキング, 文書作成技術, ブレーンストーミング, 目標設定, 自己啓発, コミュニケーション
ソフトウェアのモデル化と分析	コストモデル, 要求分析, オブジェクト指向分析, UML
ソフトウェア設計	構造化設計, データ設計, オブジェクト指向設計, デザインパターン, アーキテクチャパターン, EA, DB 設計, UI 設計
ソフトウェアの検証とテスト	ユニットテスト, 受入テスト, TDD, レビュー, ウォークスルー, インспекション
ソフトウェア発展	保守プロセス, リファクタリング
ソフトウェアプロセス	プロセスモデル, PSP, TSP
ソフトウェア品質	メトリクス, ソフトウェア品質保証, IEEE 標準規格
ソフトウェア管理	ソフトウェア構成管理, CMM ファミリ, プロジェクト管理
アプリケーション分野の知識	XML, CGI, J2EE, Struts, EJB

5. 演習課題の設計と開発

達成度レベル3~4を実現するためには、適切な演習課題を用意する必要がある。受講生のレベルには大きなばらつきがあるため、それに合わせて以下のような様々なレベルの演習課題を開発している。

- 初歩的な課題 (レベル3) としては、手順を与えた上でのソフトウェア操作実習, アルゴリズムを与えた上でのプログラミング (評価はコーディング標準に基づいて行なう), チェック項目を指示した上でのプログラムの相互レビュー, コードリーディングなどがある。
- 中程度の課題 (レベル3) としては、アルゴリズム設計を含むプログラム開発, テストケース設計, 構造化設計に従ったモジュール設計, オブジェクト指向分析/設計, アルゴリズム設計やモジュール設計を対象とした相互レビューやグループ討論などがある。
- 発展的な課題 (レベル4) としては、グループ討論やロールプレイングを通じた仕様分析やモジュール設計, TDD (Test Driven Development) に従ったコーディング, PSP (Personal Software Process) やTSP (Team Software Process) に従ったソフトウェア開発および定量的な計測などがある¹。

我々は、プログラミングのガイドライン (<http://www.cs.is.saga-u.ac.jp/syllabus/GuideLine/Cguide.html>) やアルゴリズム作成のガイドライン (<http://www.cs.is.saga-u.ac.jp/syllabus/GuideLine/algorithm.html>) を作成して系統的なプログラミングの指針とすると同時に、レビューを行なう際の基準にもしている。また、モジュール設計やアルゴリズム設計を支援するための教育用ツールとして、我々は Perseus を開発している[10]。Perseus を用いると、設計したモジュール、ルーチン、アルゴリズム、データ構造などを標準的なコメントとして埋め込んだ骨格プログラムを生成できるので、それに基づいてコーディングを系統的に進めることができる。

6. 達成度の評価とフィードバック

セミナー受講者の達成度評価は、評価を行なう目的や時期に応じて様々な手法が必要になる。我々は以下に列挙する各種の方法を用意している (開発中のものも含む)。

- セミナー実施前に行なう達成度評価は、セミナーの教育レベル設定が主要な目的である。そのため簡易な方法として、受講者に対するアンケート調査を行い、セミナーでの教育項目毎に達成度

¹ PSPおよびTSPはCarnegie Mellon大学の登録サービスマークである。

レベルの自己申告を求める方法を採用している。

- セミナー実施中に行なう達成度評価は、受講者へのフィードバックが主要な目的である。そのために、5節で挙げた各種ガイドラインに基づいた演習課題の評価用チェックシートを用意している。
- セミナー前後に達成度評価を実施することで、セミナーによる教育効果を正確に測定できる。そのために、我々は Web 上でのオンラインテストを用いた技術力評価システムを共同開発中である。

また、受講者や受講者派遣企業からのフィードバックを得ることは、個別の受講者に対して適切な指導を行なうと同時に、セミナーを継続的に改善する上で不可欠なものである。これを実施するための方策としては、セミナー実施後に派遣企業の意見を聴取する方法の他に、大福帳を用いた方法²がある。大福帳は、個別の受講者に対して配布される教師との間の連絡用紙である。以下に1日分の大福帳の様式（スペースの関係で記入欄は縮小している）を示す。

	言いたいこと・聞きたいこと・なんでもあり、の 伝言板	先生からの返事
月 日		

受講者には伝言板の欄に意見や質問などを自由に記述して毎回のセミナー後に提出することを義務付ける。セミナー講師は返事を記入して次回のセミナー時に返却することにより、受講者の反応をほぼリアルタイムで知ることができ、素早い個別対応が可能になる。また、受講者は大福帳に書くべきことを考えながらセミナーを受講するため、能動的な学習を促進できる。我々の経験によれば、セミナー受講者数が30名以下ならば、返事を書くための時間も30分～1時間程度（1～2分/受講者）で済む。

7. まとめと今後の課題

これまでに実施したスーパーSEセミナーを通じて、約120時間分の授業コンテンツと約50時間分の演習課題（正解例を含む）の開発が完了している。これらの教材（授業コンテンツおよび演習課題）は、教育目標に従ってモジュール化されているため、個別企業からの要望に対応したコースとして再構成することもできる。今後は、教材や評価システムの充実を図ると共に、指導者用教材の整備を進めたいと考えている。

ソフトウェア技術者の技術力不足問題には、教育以外にも様々な要因が複合的に関連している。それらの要因も考慮した総合的な対策の重要性は、今後ますます高まることが予想される。

参考文献

- [1] 松原友夫, ソフトウェア産業にもデフレがやってくる, 情報処理, Vol. 44, No. 4, 2003, pp. 375-383.
- [2] 情報サービス産業協会, 情報サービス産業における受注ソフトウェア開発の実態アンケート調査結果報告, 2005.
- [3] 掛下哲郎, 佐賀のIT戦略は教育から, 情報処理, Vol. 46, No. 7, 2005, pp. 844-846.
- [4] ウォルター・ディック他著, 角行之監訳, はじめてのインストラクショナルデザイン, ピアソン・エデュケーション, 2004.
- [5] 情報サービス産業協会, 平成16年度 情報サービス産業における国際化に対応した高度人材育成に関する調査研究報告書, 2005.
- [6] 情報サービス産業協会, IT サービス産業における新卒人材の採用等に関する実態調査, JISA 会報, No. 78, pp.107-116, 2005.
- [7] B. S. Bloom, D. R. Krathwohi, Taxonomy of Educational Objectives: The Classification of Educational Goals, Handbook I: Cognitive Domain, Longmans, Green, 1956.
- [8] Software Engineering 2004, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, <http://sites.computer.org/ccse/>
- [9] 情報処理学会ソフトウェアエンジニアリング分科会, 大学学部等におけるソフトウェアエンジニアリング教育・訓練カリキュラムモデル例「Jpn1」, <http://blues.se.uec.ac.jp/acc-se/IPSJ-SE-Curriculum.html>
- [10] 藤崎友美, 掛下哲郎, 構造化されたアルゴリズムを対象とした教育支援ツール Perseus, 電子情報通信学会研究報告 SS2004-55, 2005.

² 大福帳方式は織田 揮準 教授（元 三重大学 教育学部）が考案された方式である。