

ソリューション・ビジネス指向技術者育成のための導入教育

角 行之[†] 橋本 千恵子[†]

企業におけるシステムエンジニアの新入社員教育に、ソリューション・ビジネス指向技術者の育成を目的とした「SOL 型人材の育成」コースを導入した。このコースの教育時間を従来コースの半分に縮め、グループ単位で要求仕様を開き出し、システム仕様を決めさせ、システム開発をさせた。このコースとは別に、従来どおり「ソフトウェア・エンジニアリングの基礎概念の修得」を主目的とするコースを並行して実施した。両コースで教えたソフトウェア・エンジニアリングのキーコンセプトの理解度を測定した結果、新コースは従来コースの1.3-1.4 倍の有意差が認められた。この教育方法は実務を想定した「擬似OJT 型教育の有効性」を示唆している。

Vocational Education and Training for Novice Solution Business Oriented Engineers

TSURAYUKI KADO[†] and CHIEKO HASHIMOTO[†]

Authors introduced the new course for novice solution business oriented engineers named SOL cohort into the vocational education and training. The trainees divided into several groups grasped the requirement specifications, decided the specifications of the system, developed the software systems completely. On the other hand, the conventional courses named OLD cohort to aim mastering of key concepts of software engineering were executed. A regression analysis on the pre-post training aptitude mark explains that the time required to understand the key concepts on software engineering in SOL is only half that in OLD, and the average mark after training of the SOL is 1.3 to 1.4 times larger than the OLD. This may suggest an effectiveness of OJT based training.

1. はじめに

経済産業省が推進しているITSS (IT スキル標準)¹⁾ は、IT サービス産業における職種、専門分野、レベル、必要なスキルや経験、知識を体系化し、内容を詳細に定義した標準として注目されている。ITSS ではSE やプログラマーという職種はなく、必要なスキルをもった技術者の総称である。ITSS が登場した背景には、情報産業界の技術革新の迅速性と多様性がある。このため情報システムを提案する情報技術者には高度な専門性が要求されるようになった。また、複雑な情報システムをすばやく構築できる技術者の必要性も高まっている。このように社会の変化に対応して、情報技術者の育成方法も再考せざるをえなくなってきた。

著者の事業所でもこの認識が高まり、ビジネスモデルの形態を「ソリューション型」と「サービス型」に区分し、それぞれのビジネスモデルに特化した専門家を育成する決断をした。第一段階として、ソリューション・ビジネス指向の技術者を新入社員から選抜し、従来型のコースとは切り離して独自の教育を試み、成果をあげた。本論文は、この導入教育の成果と効果測定についての考察である。

2. 求められる情報技術者像と後継者育成

情報システムの構築は情報システム全体の仕様を決める時点が分岐点になる²⁾。「システム仕様を決めること」とは、「必要な情報技術を的確に選択し、それらを体系立てた企業固有の情報システム構想様式を決めること」すなわち「情報システム・アーキテクチャを決めること」³⁾であり、「概念形成と問題解決」でもあるから⁴⁾、この職種を「ソリューション・ビジネス」と定義する。

[†] 日立電子サービス株式会社
Hitachi Electronics Services Co., Ltd.

仕様を確定したあと、その仕様に従ってシステムを構築するプロセスは、顧客にサービスを提供することであるから「サービス・ビジネス」と定義する。すなわち情報システムの構築プロセスは、システム仕様（アーキテクチャ）の確定を分岐点にソリューション・ビジネスとサービス・ビジネスの二つのプロセスに区分できる。ITSSの11職種との対応では、前者が、マーケティング、セールス、コンサルタント、ITアーキテクトの4種類に、後者が、プロジェクトマネジメント、ITスペシャリスト、アプリケーションスペシャリスト、ソフトウェア開発、カスタマーサービス、オペレーションおよびエデュケーションの7種類に細分化されていると考えてよい。

情報システムの運用で生じるトラブルの多くは、ソリューション・ビジネスでの未熟さ、不具合に原因しており、「23%はシステム仕様不全」と報告されている²⁾。このプロセスで造りこまれるシステム不良は、顕在化と修復に時間がかかるため、社会的影響を及ぼすこともよくない。このため、各企業でソリューション・ビジネスを担当する技術者は経験を積んだベテランSEであり、OJTによって後継者や若手を育成せざるをえないのが現状である。しかし、OJTでは育成に時間がかりすぎることで、育成のためにベテランSEの貴重な時間を犠牲にするなどの弊害も生じることになる。そこで、若手後継者候補の層を厚くするとともに、現場に配属される前に、すこしでも必要なスキルを高める工夫をし、現場での育成時間を短縮できるよう教育部門で予備教育を強化する必要性が生じてきた。

3. 従来の研修方法

情報システム構築を主業務とするSEを育成する企業内教育の特徴は、プロフェッショナルとしてのシステム設計能力の基礎技術を身につけさせることにある。決められたシステム仕様を忠実に守り、いかに品質の高いシステムを期限内で構築できるかの力量が問われる。主としてこの業務に従事する技術者を、上述の定義に従って「サービス・ビジネス指向技術者（SER型人材）」と呼ぶことにする。多くの企業では、SER型人材にはソフトウェア・エンジニアリングの手法と作法を修得させることを狙いとする教育が主流になっている。現場に配属されてからは、応用動作は身につくものの、基礎知識の修得に時間を割くことが難しくなる。とくにコストパフォーマンスを重視する企業内の導入教育では、座学が主、演習を従とする教育が多くなる。本論文で報告する事業所の例でも、伝統的に座学

主体の教育方法を採用していた。ただ、この事業所の座学教育で留意したことは、受講生の精神衛生を考慮して、担当インストラクタが一方向的に講義をし、受講生が拝聴するという方法を避け、随所に質疑応答の時間を設けた。この受講生の反応で、インストラクタはクラス全体の平均的な理解度を類推し、進度を調整した。

この事業所の新入社員83名のうち69名を情報システム構築を主業務とする従来型の教育コースに、残りの14名は後述の新コースに投入した。従来型コースの69名は、修得すべき言語でクラス分けし、24名のCOBOL組（OLD1と呼ぶ）と45名のC言語組に分けた。さらにC言語組も、学生時代に情報処理教育を受けなかった25名の未経験組（OLD2と呼ぶ）と、20名の経験組（OLD3と呼ぶ）とに分けた。これらのOLDコースの教育内容と配分時間を表1に示す。OLDコースの特徴はグループ学習ではなく個人学習である。表1の中でOLD1/2組の「システム開発」とOLD3組の「CS開発実践」とは、この事業所が定める「標準システム開発技法」に則った開発手法を身につける作業をいう。受講生は、二ヶ月半の教育期間中に、この手法の修得を通して、与えられた仕様のソフトウェアを開発し、「ソフトウェア・エンジニアリング」の基礎を学ぶ。

4. ソリューション・ビジネス指向教育の試み

情報システムの全体の仕様を決める業務に従事する技術者を上述の定義に従い「ソリューション・ビジネス指向技術者（SOL型人材）」と呼ぶことにする。SOL型人材に期待される能力は、顧客業務のあるべき姿のデザインや、顧客の情報システム化計画立案の支援、さまざまな問題を解決するための提案とコンサルティングなどである。このためには、情報技術はもとより、問題発見・解決能力、コミュニケーション能力だけでなく、システム構築能力とプロジェクト遂行能力などが要求される。このシステム構築能力とプロジェクト遂行能力は、SER型人材に要求されるほどの専門性は要求されないが、システム開発の実現性を担保して、顧客に安心感をもたらすために必要なのである。これらの能力は属人性が強いので、OJTによって身につくものとされており、集合教育の対象からは外していた。そこで、集合教育において「擬似OJT方式」を導入し、現場でのベテランSEによる育成の負担を少しでも減らすことを教育目的にした。具体的には、ベテランSEの役割をインストラクタが演じる方法を採用した。

表1 OLD型教育の内容

Table 1 Training contents for OLD cohort

OLD 1 (24名) :	374H
・ コンピュータ入門	4H
・ プログラミング入門 (COBOL、C言語)	130H
・ コンピュータの基礎知識	124H
・ システム開発 (総合演習)	88H
・ SE業務の関連知識	28H
OLD 2 (25名) :	364H
・ システム技術入門	8H
・ プログラミング入門 (C言語)	112H
・ コンピュータの基礎知識	104H
・ システム開発 (総合演習)	88H
・ SE業務の関連知識	52H
OLD 3 (20名) :	374H
・ プログラミング入門 (C言語)	78H
・ コンピュータの基礎知識	120H
・ CS (Client & Server) 開発実践	148H
・ SE業務の関連知識	28H

問題発見・解決能力、コミュニケーション能力、システム構築能力、プロジェクト遂行能力は、程度の差こそあれ、すべての情報技術者に求められる能力ではある。しかし、「情報システム全体の仕様を決めること」を使命とする SOL 型人材には必須の要件になる。SOL 型人材にとって、これらの能力のひとつひとつが高度であるばかりでなく、これらの能力に対する顧客（発注者）からの信頼と安心感を得ることが存在意義そのものといえる。

新入社員の導入教育であるから、前提知識や実施時間に制限がある。そこで、要求する能力を「問題発見・問題解決の初歩手順」（「問題発見・解決」と略す）、「システム構築に必要な基礎概念の修得」（「システム構築」と略す）、「グループ学習に最小限必要なコミュニケーション能力の修得」（「コミュニケーション能力」と略す）、「プロジェクト遂行に最小限必要な基礎概念と初歩的な管理技法」（「プロジェクト遂行」と略す）に絞った。これらを一体化したカリキュラムを組み、擬似 OJT 方式で体験させるために、教育コースにミニプロジェクトを設けた。擬似 OJT 方式とは、ミニプロジェクト内部でリーダーを決めさせ、そのリーダーを中心にグループ学習を自発的に進める方法をいう。この過程で、担当インストラクタが、疑問を投げかけ、質問を促し、受講生が納得する明快な回答をした。

SE 希望の技術者がどのような特性（得手・不得手）をもっているかは入社時にはわからない。そこで、ソリューション・ビジネスに興味と期待をもって入社した 14 名を SOL 型人材候補者として認定した。候補者には教育範囲を絞りこみ、小規模でも自分でシステム仕様を決めて開発できる技量を短期間で身につけさせる新たな教育方法を採用し、効果があるかどうかを調べることにした。

SOL 型人材の教育の特徴は、インストラクタが発注者になり、開発システムの要求仕様を聞き出させ、システム仕様を決めさせる作業から始めたことである。この過程を通して、たとえば「問題発見・解決とは具体的にどのようなことを意味するのか」を理解させることにした。

教育時間は、従来の教育時間約 370 時間（二ヶ月半すなわち正味 54 日）の半分の時間 183 時間で試みた。この半減の根拠は三つある。第一が、現場から「教育時間を短縮し、早く配属してほしい」との要請がでていること。第二は、教育部門からも「入社直後は緊張して聴講するが、一ヶ月を過ぎると居眠りや私語が散見され、能率が急激に低下する」との報告がでていたためである。第三は、[竹田・大岩論文]⁵⁾ から、同種の教育には 150 時間は必要との示唆があったためである。教育時間を半減しても従来コースでの理解度と有意差がなく、「必要最小限の技術理解度」が保証されるなら、教育方法と期間を抜本的に見直す必要が生じる。そこで、「要求仕様を決めるというシステム開発の上流工程から、稼働確認をする下流工程までを一貫して経験させ、SOL 型人材候補としての作法を修得させる」ことを狙った。要求する能力と修得させるための教育方針とを表 2 のように定めた。

教育を効果的・効率的に行うため、教育過程を初級編（66H）、中級編（49H）、総合演習（68H）の三フェーズに分け、合計 183 時間を予定した。（実績は 167 時間）。

初級編では、プログラム文法を中心に講義し、受講生がひととおりプログラムを組めるように指導した。中級編では、システム開発演習として、自分たちで仕様を決めて、システムを設計させた。総合演習では、事業所独自のシステム開発技法に従い、システム分析から稼働確認までをグループ学習させた。表 2 の教育方針と各フェーズとの関連を表 3 に示す。受講者 14 名のうち、学生時代に情報処理教育を受けなかった受講者は 2 名（学部卒）で、12 名は情報処理教育を受けている。この 12 名の内訳は、学部卒 4 名、修士卒 7 名、博士卒 1 名になっている。したがって、入社前に情報処理教育を経験したという点で SOL コースは OLD 3 コースと同程度の予備知識を有するとみてよい。

表2 SOL 型人材への要求能力と教育方針

Table 2 Requests for ability and policy in SOL cohort

要求能力	教育方針
問題発見・解決	・プログラム作成における概要設計を重視する ・システム分析能力を養う
システム構築	・C言語基礎文法を修得させる ・プログラミングを系統的にとらえさせる ・システム開発の一連の作業を理解させる
コミュニケーション	・ドキュメンテーション能力を養う ・プレゼンテーション能力を養う
プロジェクト遂行	・共同作業をすすめる上での留意点を認識させる

表3 SOL型人材への教育方針と各フェーズとの関連

Table 3 Priority of policy in each phase for SOL cohort

教育方針	初級	中級	総合
概要設計の重視	◎	○	○
システム分析能力	×	△	◎
C言語基礎文法の修得	◎	○	△
系統的なプログラミング	◎	○	○
一連のシステム開発作業の理解	×	◎	◎
ドキュメンテーション能力	○	○	◎
プレゼンテーション能力	○	×	◎
共同作業での留意点の認識	×	×	◎

◎：最重要視 ○：重視 △：一部重視 ×：無視

5. 有効性の評価

5.1 評価の方法

新入社員の導入教育というレベルで考えた場合、ソリューション・ビジネスに特化させる SOL 型人材の教育と、従来型の教育とで共通する領域の一つに「ソフトウェア・エンジニアリング」がある。従来型の教育では、「ソフトウェア・エンジニアリングの基礎概念（キーコンセプト）の修得」を主目的としているが、SOL 型人材の教育では副目的にした。したがって、先述の「必要最小限の技術理解度」として「ソフトウェア・エンジニアリングのキーコンセプトの修得」を選ぶなら、教育効果の比較がより鮮明にできると考えた。このため、従来型の教育とは教育内容と方法を変え、教育時間も約一ヵ月半に半減させた SOL コース（183 時間）と、従来どおり二ヵ月半で実施した

OLD コース（約 370 時間）で、ソフトウェア・エンジニアリングのキーコンセプトの理解度に差が生じるかどうかを比較した。

実験で使用した共通の理解度テスト問題を表4に示す。全25問で、各問0～4点の配点をした。各項目は教育中のキーコンセプトになっており、「訓練された SE なら説明できなければならない」項目である。事前テストは導入教育の開始初日に実施し、事後テストは各コースの終了日に実施した。出題の特徴は、事前テストと事後テストが同一問題であることである。同一でも意味・意義がある問題として25個のキーコンセプトを選び、この理解度が事前と事後でどう変化するかを調べた。各項目の理解度を0～4点の5段階に分けて判定した。5段階評価にした理由は、国際的な基準と考えられる ACM (Association for Computing Machinery) の判定区分に合わせたためである。ACM の表現は、

no knowledge, recognition, literacy, usage, application となっている⁶⁾。

0点：理解していない (0)

1点：すこし理解している (1)

2点：だいたい理解している (2)

3点：かなり理解している (3)

4点：完璧に理解している (4)

システム思考的理解を要求するコンセプトでは、目的・全体と部分・応用適応性を問診で確認する。たとえば「オブジェクト指向」では、キーコンセプトを構成するキーワードとして「データ、手続き、オブジェクト、再利用、隠蔽、継承」をあらかじめ設定し、意味・目的・メリットとディメリット・使用言語との関係・適用の制約を問診する。「ウォーターフォール・モデル」では、「工程分割、ドキュメント成果物、レビュー、上流工程、下流工程、逆流」をキーワードにした。判定は、各キーコンセプトを構成する複数のキーワードがいくつ含まれているか、論旨に曖昧さがないかどうかでまず判断し、ドキュメンテーション能力とプレゼンテーション能力の有無も加味して総合判定した。判定はコーディネータと正副クラス担任があたった。この方法は判定作業に時間のかかる短所はあるが、かなり客観的に理解度が判定できる長所がある。

評価分析において「キーコンセプトの理解度」を導入した理由は二つある。第一は、評価する側の主観は入るものの、数量的に表現できるメリットがあるため、比較しやすい点にある。各キーコンセプトには、あらかじめそれを構成するキーワードを決めておき、受講

生が解答するときに、それらをいくつ使用したかにより判定した。この方法は、採点者の個人差をなくすために考案した。上記判定基準の () 内の数字は、解答で使用したキーワードの個数を表している。第二は、ソフトウェアは思想・概念の集合体であり、キーコンセプトの表現内容により理解度が推定できるのではないかと考えたからである。

表4 共通問題 (25個のキーコンセプト)

Table 4 25 key concepts

- ・仮想記憶方式 ・スーパーバイザ ・ジョブ管理
- ・アルゴリズム ・TCP/IP ・Vi エディタ
- ・サーバ ・クライアント ・COBOL ・C/C++
- ・オブジェクト指向 ・プロトタイプ ・UNIX
- ・モジュールテスト ・システムDA ・ソフトCIM
- ・プロジェクト管理 ・CASEツール ・RDB
- ・コンパイラ ・DB/DC ・ISO-9000 ・LAN
- ・ウォーターフォールモデル ・プログラム設計

5.2 定量分析

受講者全員に対して、共通の事前テストと事後テストを実施し、回帰成就値⁷⁾を求めた。回帰成就値は、事前テストの影響を考慮して異なる二つの学習形態の成績比較を行う手法である⁸⁾。

比較する三コース (SOL、OLD 2、OLD 3) の事前テストと事後テストの得点を表5に示す。教育中の学習言語が共通 (C言語) の3コースを評価の対象として、教育方式の違いによる「キーコンセプト理解力」を測るために回帰分析をする。事前テストと事後テストの結果の分布を、図1にSOL対OLD 2、図2にSOL対OLD 3を示す。使用言語の異なるOLD 1との比較は省略する。図中の直線は、SOLコースとOLDコースの、それぞれの回帰直線である。楕円はそれぞれの分布の傾向をわかりやすくするため事前テストと事後テストの平均値を中心にして1σの楕円を描いた。図1、図2からSOLコースの事後テストのパラツキがOLDコースより小さいことがわかる。また回帰直線の傾きが小さくなっている。これはSOLコースのほうが成績上位群より下位群の成績向上度合いの大きいことを示す。

定量的比較のためのデータ諸元を表6、表7に示す。

SOLコースはOLDコースにくらべて、事前テストの平均値はほぼ同じだが、事後テストの平均値は1.3~1.4倍に向上し、標準偏差も小さくなっている。

SOLコースとOLD 2/3コースの回帰成就値の差を偏差得点とすると、表6、表7から、

$$14.56 - (-8.15) = 22.71, 11.19 - (-7.83) = 19.02$$

となり、ともに検定の結果、SOLコースのほうがOLD 2/3コースよりも、1%水準で有意差があった。

5.3 定性分析

事前テストでの得点が平均点未満の者を成績下位群、平均点以上の者を成績上位群と定義する。これは表5に破線で区切っており、SOLとOLD 2、OLD 3の事前テストの平均点は、それぞれ32.4、31.7、34.0である。

成績下位群と成績上位群で事後テストがどう変化したか、成績向上の度合いがどうであったかを調べる。表8に各コースの事前テストの下位群と上位群の比較を示す。

各コースの事前テストでの下位群と上位群が事後テストでどう変化したかを、図3、4、5に示す。

表8と図3からわかるように、SOLの下位群では事前テストが20.0点に対して、事後テストが79.3点で、差は59.3点、伸び率は4.0倍である。上位群では、事前テストが48.8点に対し、事後テストが86.8点、差は38.0点、伸び率は1.8倍である。同様に、OLD 2の下位群の伸び率は2.4、上位群の伸び率は1.6である。OLD 3では下位群の伸び率は2.4、上位群の伸び率は1.7である。したがって、SOLコースとOLDコースの上位群の伸び率はほぼ同じであるが、下位群の伸びに差が生じている。

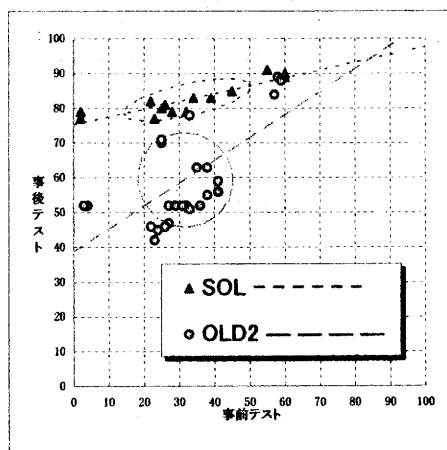


図1 SOL 対 OLD 2 の得点分布

Fig. 1 Distribution of scores for SOL/OLD 2

表5 三コースでの得点
Table 5 Scores in 3 courses

SOL		OLD2		OLD3	
事前	事後	事前	事後	事前	事後
2	77	25	70	26	56
2	79	25	70	33	62
22	82	25	71	10	65
26	81	27	52	10	65
25	80	29	52	31	60
28	79	23	42	28	55
23	77	31	52	26	57
32	79	4	52	32	58
60	89	24	45	41	69
39	83	27	47	38	65
45	85	26	46	40	63
55	91	22	46	38	67
60	90	3	52	35	59
34	83	33	78	39	67
		35	63	39	66
		38	62	62	95
		58	89	34	61
		32	52	35	63
		59	88	39	60
		41	59	43	65
		38	55		
		41	56		
		57	84		
		36	52		
		33	51		
平均	平均	平均	平均	平均	平均
32.4	82.5	31.7	59.4	34.0	63.9

表6 SOLとOLD2のデータ諸元
Table 6 Data for SOL/OLD2

項 目		SOL	OLD2
受講者数		14	25
平均点	事前テスト	32.36	31.68
	事後テスト	82.50	59.48
標準偏差	事前テスト	18.40	13.54
	事後テスト	4.67	13.75
楕円	傾き (度)	12.7	33.5
回帰成就値	平均	14.56	-8.15
	標準偏差	4.83	10.78

回帰成就値 t 検定 0.00 %

表7 SOLとOLD3のデータ諸元
Table 7 Data for SOL/OLD3

項 目		SOL	OLD3
受講者数		14	20
平均点	事前テスト	32.36	33.95
	事後テスト	82.50	63.90
標準偏差	事前テスト	18.40	11.21
	事後テスト	4.67	8.32
楕円	傾き (度)	12.7	24.2
回帰成就値	平均	11.19	-7.83
	標準偏差	2.30	6.92

回帰成就値 t 検定 0.00 %

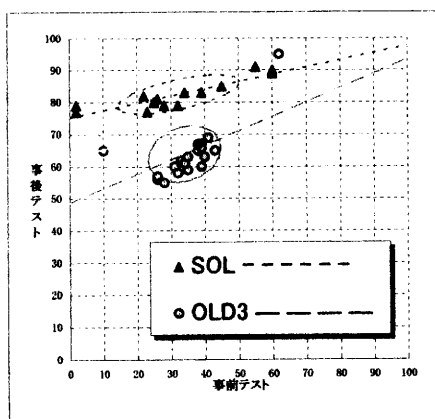


図2 SOL 対 OLD3 の得点分布
Fig.2 Distribution of scores for SOL/OLD3

表8 上位群と下位群の比較
Table 8 Comparison Higher and Lower

SOL	事前	事後	伸び率
下位群	20.0	79.3	4.0
上位群	48.8	86.8	1.8

OLD2	事前	事後	伸び率
下位群	22.4	53.6	2.4
上位群	41.8	65.8	1.6

OLD3	事前	事後	伸び率
下位群	24.5	59.8	2.4
上位群	40.3	66.7	1.7

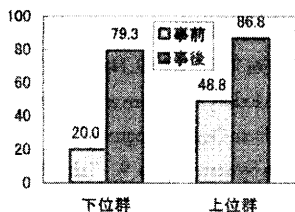


図3 SOLの下位群と上位群の比較

Fig. 3 Comparison Lower and Higher in SOL

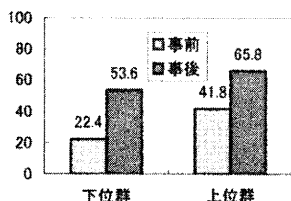


図4 OLD 2の下位群と上位群の比較

Fig. 4 Comparison Lower and Higher in OLD 2

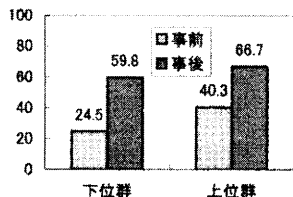


図5 OLD 3の下位群と上位群の比較

Fig. 5 Comparison Lower and Higher in OLD 3

6. 考察

SOL 型人材の育成を目的とした新コースと従来コースとの有意差の原因を考察する。両コースとも標準システム開発技法をベースにシステム開発を経験させた点は共通である。しかし SOL コースでは、オブジェクト指向やウォーターフォール・モデルなどのキーコンセプトについて、グループ単位で実例を示しながら、OJT 方式でインストラクタが教えたため、ほぼ全員がその意味を理解したものと推測している。具体的には、「システム構築」能力を養う「システムのプログラミング」と「一連のシステム開発作業」を指導するときに工夫を凝らした。すなわち、関係する「キーコンセプト」を、初級・中級・総合演習の各フェーズで、繰り返し表現を変えて説明し、感覚情報や短期

記憶ではなく、長期記憶に定着させる努力を続けた。この努力が、事後テストのバラツキをすくなくすることに寄与したものと解釈している。

一方、OLD コースでは、講義と演習とを分離して教えたため、コンセプトの意味を理解する受講生にバラツキが生じたものと思われる。たとえば「ウォーターフォール・モデル」で、知的生産の方法としての意味は理解できるが、品質を射影するドキュメントの重要性までは考えが及ばない。また、頭の中だけの理解のため、各工程をどこで打ち切るかの判断が工程全体の納期に影響することまでは思いが至らない。これらは、個人の資質の問題ではなく、教育方法の問題に属すると考えるべきだろう。教育中は、講義でも演習でも、わからないことは積極的に質問するように仕向けてはいるが、人前では質問をせず、講義終了後に個別に聞きにくる傾向がある。個人学習ともあいまって、これがバラツキを生じさせた原因と思われる。

この結果、経験も知識も乏しい新入社員に概念を刷り込む場合、作業に即して教えることの重要性を痛感した。

SOL 型人材の早期育成の第一段階、すなわち、発注者から要求仕様を聞き出し、自分たちで仕様を決めてソフトウェアを開発する訓練は初期の目的を達した。従来の半分の時間での「指定したソフトウェア・エンジニアリングのキーコンセプトの理解」も達成できた。受講生も、自ら SOL 型人材を希望しただけに、「必要な能力がなにかを再認識した」と認めている。予定時間 (183 時間) をみずからの努力で 167 時間に短縮できたことによって、「やればできる」という自信を植えたことは、受講後のアンケートからも窺える。

しかしながら、これだけでは問題発見・解決能力などの SOL 型人材に必要と定めた能力が身についたかどうかの客観的判断はできない。このため、今回の被験者の 3 年後、5 年後の追跡調査を継続し、確認する必要がある。

また、今回対象にした被験者数 (母集団) は、標本の大きさが (14, 25, 20) であり、標本の大きさとしては少量である。統計データの標本の大きさに最低基準はないが、この種の調査では数 100 以上でないと充分とはいえない。しかしながら、新入社員が数 100 人を超える事業所などは情報産業界にはなく、高々数 10 人を数える程度である。しかも今回採りあげた領域の問題意識をもって新入社員の導入教育を実施している事業所からの報告はまだない。SOL 型人材の早期育成は急を告げており、百年河清を俟つわけにはいかない。よって、本データでの結論を公開し、陳勝吳広とした。

7. まとめ

新入社員に必要な不可欠なソフトウェア・エンジニアリングのキーコンセプトを理解させるという観点から、新設のSOLコースと従来のOLDコースの教育方法の違いによる定量的効果をまとめる。

- (1) SOL コースは、OLD コースの約半分の教育時間(167 対 364、374) で目標が達成できたうえに、事後テストの平均値も1.3-1.4 倍を記録した。
- (2) 成績上位群の伸びに差はないが(1.8 対 1.6、1.7)、下位群の伸びに有意差があった(4.0 対 2.4、2.4)。

本研究は、新入社員の導入教育において、仕様を自分で決めるシステム開発作業を体験させ、システム提案に必要な技量と見識をOJT 式に教える方法が効果的であることを示しているので、追試をして検証してほしい。

なお、「ソリューション・ビジネス」という新職種を担当するSOL型人材の育成には、今後とも斬新な専門教育コースを用意しているが、中核となる科目をつぎのように設定している：

- ・ 概念形成と問題解決 (3 日)
- ・ 問題解決からビジネス・モデリングへ (3 日)
- ・ ビジネスにおける論理思考 (2 日)
- ・ ビジネスにおけるシステム思考 (2 日)
- ・ 見積りの本質と評価 (2 日)
- ・ 情報技術の変遷と経営へのインパクト (2 日)
- ・ リーダーシップと取りまとめ手腕 (2 日)
- ・ 表現術の勘どころ (2 日)
- ・ プロジェクト/プログラム・マネジメント (3 日)
- ・ 倫理と専門性および法律 (1 日)

これらを学ぶことにより、経営者(発注者)が期待するシステム提案と構築に力量を発揮する情報技術者がひとりでも多く誕生することを期待している。

参考文献

- 1) 経済産業省「ITスキル・スタンダードβ版」(2002)
- 2) 角 行之：情報システムの運営と管理、情報処理学会第63 回全国大会チュートリアル(1) pp.39-46 (2001)
- 3) 岩田裕道：情報システム・アーキテクチャ概説、情報処理学会第64 回全国大会チュートリアル pp.1-10 (2002)
- 4) 手島歩三：概念形成と問題解決、情報処理学会第63 回全国大会チュートリアル(1) pp.15-20 (2001)

- 5) 竹田尚彦、大岩 元：プログラム開発体験に基づくソフトウェア技術者育成カリキュラム、情報処理学会論文誌 Vol. 33, No. 7 pp.944-954 (1992)
- 6) ACM: IS'97 Model Curriculum and Guide Lines for Undergraduate Degree Programs in Information Systems、pp. 9 Table 1 (1997)
- 7) 池田 央：テストと測定(教育学大全集25)、第一法規出版 pp. 70-71 (1982)
- 8) 角 行之、保田勝通、山本洋雄、大槻 繁：企業における上級ソフトウェア技術者養成のための実践訓練コースSEPの開発と評価、教育システム情報学会誌 Vol. 18, No. 1 pp.111-120 (2001)