

若年技術者向けソフトウェア 開発研修プログラムの開発と評価



大森 久美子[†]

若年技術者の設計力及び開発力の底上げを目的としたソフトウェア開発のPBL研修を、5年間にわたって実施しながら改善し、ソフトウェア開発研修プログラムとして集大成した。

本研修プログラムは、要求分析から受入れテストまでの全工程を6日間で経験できる基礎研修と、さらに技術スキルの深掘りを行う2日間のフォロー研修で構成されている。

研修プログラムには、研修を効果的にかつ効率的に進めるための、受講者の気づきに対するフィードバック方法や要求分析の新しい手法など、様々な工夫を考案して取り込んでいる。

評価の結果、ソフトウェア開発プロセスの網羅的な内容を含み、かつ、受講者自らが問題の原因と対策を考えることができる研修にできたことを確認した。

本報告では、研修プログラムの開発経緯とその内容及び評価結果について報告する。

Software engineering training program for entry level engineers and its evaluation

Kumiko Ohmori[†]

A software engineering training program has been developed for 5 years.

This program consists of 6 days primary training course and 2 days follow up training course.

The primary training course includes whole software development process.

The follow up training course includes further higher level contents to improve engineers' technical skill.

This program includes various new methodologies.

For example, awareness feedback methodology and requirement analysis methodology are developed.

From the evaluation, this program showed to accelerate engineers' ability to think.

This paper describes the details of this program and its evaluation.

1. はじめに

システムに占めるソフトウェアの比率が増大しており、すべての若年技術者に対してソフトウェアに対する素養が必要とされている。これは、単にプログラムが書けるということではなく、ソフトウェア開発の設計、製

造、テストの全工程をこなせることを意味する。

【脚注】

† NTT サービスイノベーション総合研究所 ソフトウェアイノベーションセンター
NTT Service Innovation Laboratory Group, Software Innovation Center

近年、産業界からの要望などにより、大学において実践的なソフトウェア開発のPBL (Project based Learning) が普及してきた。しかし筆者の組織においては、様々な大学の異なる分野の卒業者を採用しているためPBLの経験者が5%程度と少なく、上記の取り組みの成果を享受するには至っていない。

そこで若年技術者を含む新入社員研修の一環として、PBL形式でソフトウェア開発の研修を実施することにした。まず、世の中の研修が利用できないか調査したが、次のような問題があった。

- ・課題を与えられて設計からテストまでを行うのが一般的であり、筆者の組織で重要視する要求分析工程が含まれていない
- ・定められた期間内にゴールまで到達できるように、穴埋め式の研修が多いため、知識は身につくがスキルが身につくとは言い難い

このような状況から、筆者は新たに研修の設計を行って、実際に研修を実施した。初年度の成果はSECジャーナルで報告したが [1]、その後4年間にわたって研修を実施し、その都度研修内容について評価と改良を行ってきた。その結果、8日間で企業が必要とするソフトウェア開発の全工程を経験し、基本的なスキルを身に付けることができる研修プログラムを開発することができた。

本研修プログラムは、次の特徴を有している。

- ・問題発見、要求分析から受入れテストまで、ソフトウェア開発の全工程を含んでいる
- ・顧客（発注者）と開発者（受注者）の両方の立場を経験できる構成になっている
- ・フォロー研修を追加して、知識の定着とスキルの高度化を図っている
- ・要件定義工程は、後述する7要素法を考案して一般的に新入社員には困難といわれる要求分析作業も可能にしたり、その他の工程にも様々な工夫を凝らしている
- ・研修全体にわたって気づきのフィードバックを行うようになっており、受講者自らが自分の成長を実感できるようになっている

本論文では、研修の改良の経過と最終的な研修プログラムを紹介する。

2. 研修プログラムの設計

研修プログラムは、ID (Instructional Design) に基づいて設計した [2]。以下に研修プログラムの概要を示す。

2.1. 対象とする受講者

本研修プログラムでは、新入社員を含む入社数年程度の若年技術者を対象としている。

実際に受講したのは、年度により異なるが、平均約60名で、これを2クラスに分け、さらにクラスは5名程度のチームに分けた。

前提条件として、Java、PHPなどを用いてプログラミングができることとしているが、実際にはプログラミング経験のない社員もかなりいたので、最近では技術講座として事前にJavaなどのプログラミングを教える研修も導入している。

2.2. 研修目標

企業においては、単なる知識を身に付けるだけでは業務に役立たないので、知識を使えるスキルを身につけさせることを目標とした。具体的には、以下の目標を設定した。

- ・ソフトウェアの開発工程の全体像、及び各工程の目的、成果物を理解し、実践できる
- ・各開発工程で問題が生じた際、その原因と対策を考えることができる。特に上流工程の曖昧さが下流工程、及びソフトウェアの品質に与える影響を経験から学び、上流工程の重要性を知る。

2.3. 研修内容と研修期間

研修は、6日間の基礎研修と、その数カ月後、知識やスキルの深堀りを行う2日間のフォロー研修で構成している。

表1に基礎研修の日程を、表2にフォロー研修の日程を示す。

表1 基礎研修の日程

	講義	実習
1日目	開発プロセス	要求分析
2日目	要件定義	要件定義
3日目	外部設計	外部設計
4日目	内部設計、製造	内部設計、製造
5日目	テスト、品質管理	製造、テスト
6日目	プロジェクト管理	受入れテスト、成果発表

表2 フォロー研修の日程

	講義	実習
1日目	要求分析	要件定義書の見直し
2日目	機能設計 品質設計	方式設計の見直し 非機能要求の見直し 事例分析

2.4. 教材

研修は、筆者らが所属する組織の課題を踏まえ、伝え

たいことを決定後、IPAの書籍などを参考にしながら、独自に開発した。詳細は、4章で述べる。

2.5. 研修環境

研修で開発するソフトウェアは受講者がチームで相談して決めるものとするが、普通は、Webサーバとデータベースサーバを用いたものとなることが多い。

研修環境としては、これらの環境を構築するために、以下の要件を満たす必要がある。

- ・ ネットワーク環境が構築できること
- ・ 必要なソフトウェアがダウンロードできること
- ・ 必要なサーバ環境を構築できること

2.6. 講師

講師は、以下の要件を満たす必要がある。

- ・ PBLや技術の指導ができること
- ・ 研修中に生じる様々な課題に対して、その課題と関連する経験内容を話して受講者に考えさせることができること
- ・ 研修を行う企業の文化を理解し、メンターとしても対応できること

本研修では、筆者が全体をとりまとめ、PBLの指導は開発経験が豊富なOB社員2名に支援を依頼した。

2.7. 評価方法

評価は、以下に示す3つの側面から実施した。

- ・ 受講者の評価
- ・ 研修の構成の評価
- ・ 指導方法の評価

これらの評価を行うために、以下の情報収集を行って分析した。

- ・ 研修の最初と最後に用語レベルの知識確認テストを実施し、受講者の知識レベルの変化を確認する
- ・ 成果物の内容を見て、受講者のスキルの到達状況を確認する
- ・ 研修中に受講者が書いた気づきシートから、受講者

ID:	所属:	氏名:	月日:
気づき			
内部設計が固まらないまま、スケジュールの都合で製造を行っている。			
内部設計があいまいなため、大まかな正常系しかテストシナリオを作成できない。			
異常系について何も取り決めをしていないことに気づいた。			
正常系でも未定義なフローがあり、追加の打合せを行う必要があった。			

図1 気づきシートの例

のスキル、マインドの変化を把握する[3]。図1に気づきシートの例を示す。この例では、製造工程に入ったところで、それまでの工程の進め方の不備に気づいている。

気づきシートには、研修内容や講師に対する要望も記述されるので、研修の構成や指導法の評価にも活用できる。

3. 研修プログラムの改良

2章で示した研修プログラムは、5年間にわたる研修の改良の結果である。詳細は4章に示すが、ここでは、他の組織においても本研修プログラムを参考にさせていただけるよう、改良の過程を紹介する。

【1年目】不足部分をフォロー研修で補足

最初は、ソフトウェアの要求分析から納品までの全工程を経験できるPBLを開発した。

時間不足で受入れテストができないチームがあったので、フォロー研修を設計し、受入れテストの実施、及び設計書の記述不足を補う設計書の見直しをテーマとした。

フォロー研修で受入れテストを実施したところ、受講者自身が特に上流工程の品質保証の重要性に気づいたので、品質管理の研修を追加した。したがって、この年は、基礎研修後にフォロー研修、品質管理研修と2つの追加研修を実施した。

【2年目】要件定義と受発注関係を追加

2年目は、チーム作業において時間(納期)を意識させた結果、受講者にシステム開発の責任感が芽生え、基礎研修の中で受入れテストまで実施できた。

2年目の大きな特徴は、チーム作業に受発注の関係を導入したことにある。すなわち偶数チームを編成し、Aチームが行った要件定義をBチームが開発しAチームが受入れる、Bチームが行った要件定義はAチームが開発しBチームが受入れることとした。こうすることにより、要求を相手に伝えることの難しさ、曖昧な要求を受け取ると後の工程が進まないこと、要求が曖昧であると受入れの際の確認項目も曖昧になり、受入れ判断ができないことを伝えることができた。

さらに、ソフトウェア開発を一通り経験して品質に対する認識が高まった結果、受講者は要件定義の重要性に気づいたので、基礎研修後のフォロー研修として要件定義の研修を行った。

【3年目】方式設計を追加

3年目は、要件定義に続く外部設計の部分に改良を加え、方式設計のプロセスを研修に追加した。

表 3 フォロー研修の設計

基礎研修の結果 明らかになった課題		フォロー研修で採用した研修項目					記事
		要求分析		機能設計		品質設計	
		システム 全体像の 図表現	商用 システム の観点	方式設計	インタ フェース 設計	非機能 要求の 抽出	
受講者の 課題認識	考えていることを正確に伝えるのが難しい	○					
	非機能要求の抽出が難しい					○	
	作業の見積が難しい						OJT で実施
	実務ではどうなっているか知りたい						○
講師の 課題認識	全体像の把握が曖昧	○					
	商用化システムに対する観点が不足している		○				
	方式設計まで検討が進んでいない			○			
	インタフェース設計が甘い				○		
	非機能要求項目の抽出が甘い					○	○

【4年目】上流工程の改良と最新技術の導入

4年目は、ここまでの改良の過程を反映し、性能などの基本的な非機能要求を考慮した要件定義や方式設計のプロセスを基礎研修の中で伝えることにした。

さらに、4年目は、Androidの開発も組み込み、本格的なクライアントサーバ型システムの開発を実施した。

【5年目】研修プログラムを集大成

ここまで、基礎研修の後に、フォロー研修と品質管理研修と2つの研修を実施してきたが、基礎研修後の2つの研修を1つにまとめた。

フォロー研修は、基礎研修での課題を認識し、解決のための知識とスキルの深掘りが目的であり、主に要求分析、方式設計、品質管理をテーマとした(表3)。

なお、見積もりについては、経験がないと実感がわからないので、フォロー研修後にOJT(On the Job Training)で実施することにした。

このようにして、若年技術者のソフトウェア開発スキルを段階的に高める研修プログラムが完成した。基礎研修、フォロー研修を通して、開発の一連の流れを深く経験できるプログラムとすることができた。

4. 研修プログラムの工夫点

4.1. 研修プログラムの構成

(1) 短期間で効果的な研修

研修プログラムは、当初は最初の基礎研修とフォロー研修2回で構成していたが、フォロー研修の内容を基礎研修に組み込むことにより、現在は、フォロー研修1回で構成している。

(2) 自らが気づく研修

毎回の研修は、基礎技術の講義の後に、実習を行う。時間配分は、講義12時間に対して実習44時間の割合

となっている。なお、日程は1日7時間の基礎研修を6日間実施し、1か月程度の間隔をおいて2日間のフォロー研修を実施している。

毎回の研修終了時には、気づきシートを回収し、次の研修の最初に、気づきのフィードバックを行う。フィードバックでは、受講者の疑問に答えたり、注目すべき気づきを紹介したりする。こうすることで、情報の共有とともに、気づきを深掘りするためのアドバイスも行っている。気づきのフィードバックの例を図2示す。

研修の最後には、成果発表を行ってチーム活動を振り

- **さまざまな気づき** 共有したい受講者の気づき
 - 品質管理には定量的な基準が必要
 - プロジェクトの火の見櫓で監視するだけでなく、火消しも必要
 - 課題管理表をはずして仕事をしていた
- **議論の発展に向けて** 講師からのアドバイス
 - 問題指摘だけでなく、「何故」を考え、どうすればよいかを考えてみよう
 - ソフトハウスに対する不満が書かれていたが、ソフトハウスの立場でも考えられないか
 - 顧客の文化を捉えきれないのは何故か

図 2 気づきのフィードバック例

- **何ができるようになったか**
 - 何のための機能なのか考えられるようになった。
 - これまで「どのように実現するか」ばかり考えていたことに気づき、「何を実現したいか」を考えられるようになった。
- **自分に不足しているものは何か**
 - ソフトウェアの作成を文書で管理するという感覚に慣れていないため、設計書作成のスキルが必要である。
- **これから何をすべきか**
 - 要件の詰めが甘かったのもっと細かく検討しなければならない。
 - 協力会社とのパートナーシップのあり方について考えていきたい。

図 3 気づきの振り返りの例

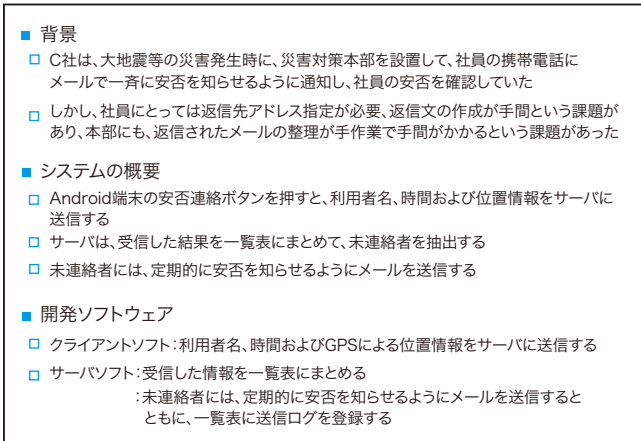


図 4 テーマの概要



図 5 教材の書籍化

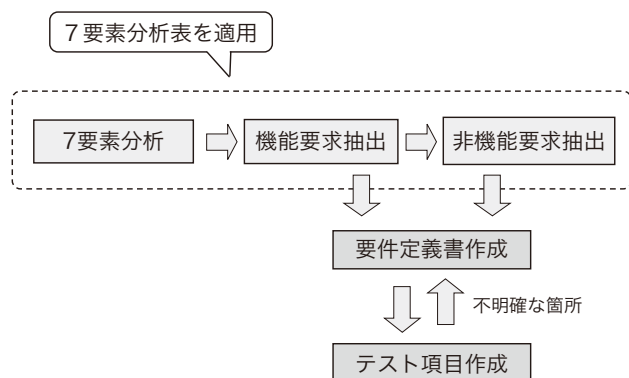


図 6 7要素分析法

返り、さらに、気づきシートを返却して受講者個人の気づきを振り返る機会を与えている。気づきの振り返りの例を図3に示す。

(3) 教えずに気づかせる指導

気づきのフィードバックは、指導の重要な位置を占める。受講者が気づいたことを紹介することも多く、教えずに気づかせるための有効な手法となっている。

以下に、気づきの例と対応例を示す。

- ・ 気づきシートに「時間管理が重要」と書かれていた場合は、よい気づきとして紹介し、進捗管理の例を紹介する
- ・ 気づきシートに「相談して意識合わせを行った」と書かれていた場合は、報告・連絡・相談を表す「ほうれんそう」を説明するとともに、PMBOKの考え方も説明する

・ 気づきシートに「全体像は図で表すと分かりやすかった」と書かれていた場合は、リッチピクチャやUMLのダイアグラムを説明する
基礎技術は講義で説明するが、気づきの状況から次回の講義内容やレベル調整も行っている。

(4) 大枠のみ指定する自由度の高いテーマ設定

当初、研修で開発するソフトウェアの選定は受講者に任せていたが、時々アルゴリズム開発など、ソフトウェアの開発工程を理解するという研修目的と異なるテーマ設定も見受けられたので、以下のような対処を行うことにした。

- ・ 事前に導入研修などで組織のソフトウェアの特徴や組織が期待するスキルを示し、研修に取り組む際の意識付けを行う
- ・ 研修のテーマ検討時、例えば「安否確認のシステムをクライアントサーバ型で開発すること」というように大まかな枠組みを示す(図4)

(5) 教材の書籍化

教材は新規に開発したが、その結果は、指導方法も含めて以下の内容で書籍化している(図5)。

- ・ 基礎研修 [4]
- ・ 開発管理(品質, 見積) [5]
- ・ 要求分析 [6]
- ・ 開発時の様々な問題対応 [7]
- ・ Android 業務アプリ開発 [8]

4.2. 要求分析で用いる7要素分析法

要求分析では、要求獲得の一環としてステークホルダの明確化が重要である。

本研修では、ソフトシステム方法論(SSM) [9]のCATWOEに「目的, 手段, 活動内容など」を追加した7要素分析法を考案した。具体的な7つの要素を以下に示す。

- ・ 利用者(Customer)
- ・ 活動に関わる人(Actor)
- ・ 入力・変換過程・出力(Transformation process)
- ・ 世界観(Weltanschauung)
- ・ 活動を止められる人(Owner)
- ・ 環境上の制約(Environmental constrains)
- ・ 目的, 手段, 活動内容など(今回追加)

これは、まずシステムの機能要素を7要素に着目して抽出し、それを機能に展開する方法である。要求の抽出や文書化に慣れていない若年技術者にとって、7つの要

表 4 7 要素分析表の例

Soft Systems Methodology (SSM) による要素							
システム名	利用者 (Customer) (受益者)	Actors (活動に関わる人)	目的、手段、活動内容など	Transformation process (入力、変換過程、出力)	Weltanschauung (活動を意味づける世界観)	Owner (活動を止められる人)	Environmental constraints (環境上の制約)
最適温度時間力 カウントダウンシス テム	武蔵野研究開発セ ンタ勤務者		温度時間を知りたい 温度時間までをカウントダウンしたい バス発車時刻に合うように温度したい	【入力】 自席位置、バス停名 【保持しておくパラメータ】 各バス停の時刻表情報 自席から(エレベーター階段) までの所要時間 各階から1階までの所要時間 各階から1階までの所要時間	(「ギリギリまで仕事をする」) 最速時間の計算 移動の効率化 ストレスの軽減	システム会社 (サービス事業者) (バス運用会社) (運営業者)	動作環境(ブラウザ条件)

機能要求・要件			非機能要求(非機能要求の4大要素)				
方式モデル (活動モデル)	機能要件	可用性	性能・拡張性	運用・保守性	移行性	セキュリティ	システム環境・エコロジー
方式モデル (活動モデル)	- 温度時間計算 - 温度時間表示 - 温度時間計算 - カウントダウン(現時刻と温度時間を参考)	平日6:00~24:00で 運用	入力できる自席箇所は 本館の階数のみ 時刻の粒度は30分ご と			Webでの実装	
バス運送に対応	- 温度時間計算(自席からバス停までの最短時 間) - バス運送情報の取得		バスの時刻表の改訂 に対応 バスの運送に対応			該当する施設の利用 者のみが利用可能	

要素を示すことにより、システムの基本的な機能を一つずつ抽出することが可能となる。

しかし、異常系の抽出は困難であるので、抽出した機能に基づき、IPA の非機能要求グレード [10] を用いて異常系などの機能を抽出することにした (図 6)。

7 要素を用いた分析は、7 要素分析表を用いて行う。表 4 は 7 要素分析表の記述例である。

なお、SSM では、CATWOE を用いた分析に先立ち、リッチピクチャ (ポンチ絵) を書くのが一般的である。本研修では、7 要素分析表を書いた受講者が、システムの全体像を表すポンチ絵が必要だと気づきシートに書いた後、リッチピクチャを説明した。次年度以降は、ポンチ絵やアクティビティ図やシーケンス図による図表現の説明を事前に行うことにした。

SSM では、CATWOE の要素を対等に扱うが、若年技術者の研修では、世界観がチームのメンバーの意識合わせに効果があったとの報告が多かった。これは、コンセプトの共有が有効だったことを意味していると思われる。

7 要素分析を行った後、要件定義書を作成した。要件定義書のあいまいさを減らすために、要件定義書からテスト項目を作成する作業を追加した。テスト項目を作ることができない要件については、要件定義書の見直しをさせた。要件定義書へ反映できない項目については、次工程へ引き継ぐための検討項目の一覧 (管理表) を作成させ、プロジェクトマネジメントに関する指導も併せて行った。

4.3. 設計工程に方式評価を導入

若年技術者は、課題に対して自分の持っている知識のみを頼りに、決めうちで方式を設計する傾向にある。企業では、システムのアーキテクチャを定め、さらに個々の機能の実現方式について複数の方式 (代替案) をあげ

表 5 方式評価の例

案	概要	性能	拡張性	小規模適用時の作りやすさ	総合評価
1	動的にルートと所要時間を計算する	×	○	×	×
2	あらかじめ全てのルートと所要時間を計算しておく	○	×	○	○
3	動的にルートと所要時間を計算した結果を保存しておく 同じパターンの場合は再利用する	△	○	×	△

○：優れている △：普通 ×：劣っている

て比較評価し、最も適切なものを選択するのが一般的である。

本研修では、IPA が提供している OSS 教材の「コンピュータシステムアーキテクチャ」[11] を参考にしながら、方式の設計方法を教えた。例えば、Web 三層モデルや MVC モデルでアーキテクチャを説明した。その後、機能の実現方式について、例えばデータの渡し方を、メモリ利用型にする場合、ファイル経由型にする場合、データベース利用型にする場合などについて、利害得失を比較し、最終判断する方法を示した。

教材は、受講者が基礎研修で開発したソフトウェアを対象とした。表 5 に複数の方式評価例を示す。

この工程では、教えるたびに指導する項目が増えてきている。例えば、Android 端末を利用したシステムでは、無料ソフトを利用したり、トラフィックの増加を無視した方式を採用したりと、商用化の観点を考慮せずで作る傾向が見られた。こういう面の指導は、フォロー研修で対応することになっている。

4.4. 製造、テスト工程で活用する OSS

製造工程では、ネットワーク環境と開発ツールを整備する必要がある。

表 6 気づきの変化の評価

No.	気づきのレベル			気づきの特記事項（課題認識注目すべき気づき）		
	基礎研修	フォロー研修 1 回目	フォロー研修 2 回目	基礎研修	フォロー研修 1 回目	フォロー研修 2 回目
1	3	5	5	知識以外の気づきが多かった	自分の改善マインドを自覚した	研修で学んだことを自グループにフィードバックする
2	3	4	4			作業標準が必要学んだことを実務にフィードバックする
3	2	3	3		方針や観点の共有が重要	ノウハウの蓄積が重要

表 7 気づきのレベル変化（受講者全員の平均）

	基礎研修	フォロー研修 1	フォロー研修 2
基礎研修から参加	2.26	3.13	3.17
フォロー研修 1 から参加	—	2.00	2.93
フォロー研修 2 から参加	—	—	2.11

開発環境は、インターネットに接続できる環境が必須であり、企業においては、セキュリティポリシー上、業務環境とは別のネットワークとしなければならないことが多い。開発に使用する PC やスマートフォンなどのクライアント端末も別途調達する必要がある。本研修プログラムでは、来訪者用のインターネット接続環境を利用するとともに、PC 及びスマートフォンはレンタル品を調達した。

開発ツールは、OSS ツールを活用して進めた。ツールの選定は受講者に任せしたが、LAMP といわれる Linux, Apache, MySQL, PHP, Perl に加えて、JavaScript が多く利用された。開発環境としては、Eclipse に加えて、構成管理が重要と考えたチームは Subversion などを利用した。

4.5. 品質管理で活用するゾーン分析

品質管理は、品質保証を行うために必要であることを理解させた後、品質管理の詳細について説明した。定量的品質管理と定性的品質管理を説明し、実際には両者が補い合って品質を高めることを事例紹介と演習を交えて教えた。

品質管理の教材には、IPA の書籍「定量的品質予測のススメ」[12] を活用し、特にゾーン分析については、社内の実際のデータを用いて演習を行った。

品質管理に加えて、見積手法についても教えた。教材には、IPA の「ソフトウェア開発見積りガイドブック」[12] を活用し、FP 法を紹介した。

ただし、FP 法を数日の開発研修で身に付けるのは困難であり、研修では紹介するにとどめ、適用技術は、OJT で身に付けることとした。

5. 評価

5.1. 受講者の評価

(1) 知識レベルの評価

受講者の知識レベルの向上度は、知識レベルの確認テストを研修の最初と最後に実施して確認した。その結果、開始時には平均 65 点程度で分散も大きいですが、研修終了時には 85 点程度に向上し分散も小さくなったので、知識付与の目的は達成できたと判断している。しかし、フォロー研修の最初と同じテストを行うと点数が 70 点程度に下がっていた。2 回目のフォロー研修でも同様の傾向を示したので、研修は継続して実施する必要があることが確認できた。

知識レベルの変化は毎年同じ傾向になっている。本研修の知識レベルの目標は達成できていると判断し、後述するスキルと思考力の向上に向けた改善に注力することとした。

(2) スキルの向上

基礎研修受講時には、要件定義書や外部設計書の目次構成にしたがって記述することで精一杯だったが、フォロー研修では内容にまで踏み込んだ記述ができるようになった。特に、非機能要求についての配慮ができるようになっており、研修の目標は達成できたと判断している。

(3) 思考力の向上

気づきシートを何百枚も読んでフィードバックしていくうちに、次のような基準で気づきのレベルを振り分けることを考案した [13]。

- レベル 1 気づきなし（指示され受け身の姿勢で研修を受講しただけ）
- レベル 2 知識は習得した
- レベル 3 問題を発見した
- レベル 4 原因を考察した
- レベル 5 問題の対策を考えた

初年度の研修について、このような基準で評価した個人ごとの評価結果を表 6 に、研修参加回数による受講者全員の気づきのレベルの変化を表 7 に示す。

これから、フォロー研修により気づきのレベルが上がっていることが分かる。

チームのリーダーは、だいたい研修目標に掲げた問題の原因と対策を考えるレベル 5 に達している。

気づきのレベルが低くなる原因は、チーム活動による他人への依頼のマインドが影響していると考えられる。そこで、最近では、チーム活動で他人に頼る傾向が見られる場合は、チームで議論する前に一定時間個人で考える時間を与えるようにしている。

なお、考案した方法は、個人ごとに課題認識レベルの違いが分かるので、今後の技術者育成の基礎情報としても活用が期待されている。

5.2. 研修の構成の評価

本研修プログラムは、2 章で述べたように新しい取り組みを適用するごとに評価して改良を重ねてきた。

その結果、研修は、基礎研修とフォロー研修で構成するのが効果的であるという結論にたどりついた。

研修内容については、要求分析は初心者には難しいが、基礎技術としては基礎研修で教えておき、フォロー研修で深掘りするのが効果的であると判断した。

この結果を受けて、筆者の所属する組織全体の研修体系も、基礎研修と応用研修に整理して実施することになっている。

5.3. 指導方法の評価

講師がめざしたことは、教えずに気づかせることである。本研修では、開発するシステムは受講者が決め、その開発過程の中でシステム開発のポイントを受講者が自覚できるようにした。

そのために次の行動をとった。

- ・ 受講者がとった判断や行動について、なぜそうしたのか質問をする。
- ・ 受講者の質問に対してどうするべきかを示さず、講師が経験した類似の経験を紹介する。
- ・ 技術を説明する場合は、複数の方法を示し、それぞれに長所と短所があることを示す。

こうすることにより、プロジェクト管理の重要性などについて、受講者自らが気づいてくれることが多くなってきた。

6. おわりに

若年技術者の設計力及び開発力の底上げを目的としたソフトウェアエンジニアリングの PBL 研修について、現在に至るまでの改善経緯と現段階での研修プログラムについて紹介した。評価結果で示したように、単に知識を身につけるだけではなく、受講者自らが考える研修とすることができた。

この研修プログラムの内容は書籍化して公開しており、最初に出版した書籍 [4] は 6 刷を重ね出版数は約 16000 部となった。その他の書籍も 3500 から 5000 部が出版されており、複数の大学や研究機関でも活用されている。電子情報通信学会の先端オープン講座でもテキストとして採用されている。

開発した研修プログラムは、研修実施のたびにその内容について評価し、改良を加えていった。見方を変えれば、講師陣が研修をとおして Faculty Development (FD) を行い、その結果、研修プログラムも改良されていったといえる。

例えば、受講者への質問方法については、最初の研修ではあまり気づかなかつたが、研修の指導を重ねるうちにだんだん 5.3 に示したような対応方法に気づいてきた。

すなわち、研修の改良と FD は車の両輪であり、今後も継続して、両者の向上に取り組んでいきたい。

【参考文献】

- [1] 大森, 神沼: 問題形成から受入れ検査までを含んだ PBL 型ソフトウェア開発研修とその評価, SEC journal, Vol.5, No.3, Jun2009
- [2] 実践的ソフトウェア教育コンソーシアム: 教育デザイン入門, オーム社, 2007
- [3] 神沼, 黒田: 気づきシートの活用と分析, 研究報告 2009-IS-107, 情報処理学会研究報告 Vol.2009, No.32, 2009
- [4] 大森他: ずっと受けたかったソフトウェアエンジニアリングの新人研修, 翔泳社, 2009
- [5] 大森他: ずっと受けたかったソフトウェア開発管理の集中研修, 翔泳社, 2010
- [6] 大森他: ずっと受けたかった要求分析の基礎研修, 翔泳社, 2011
- [7] 大森他: ずっと受けたかったソフトウェアエンジニアリングの新人研修 開発現場編, 翔泳社, 2012
- [8] 大森他: ずっと受けたかった Android 業務アプリ開発の新人研修, 翔泳社, 2013
- [9] ピーター チェックランド他: ソフト・システムズ方法論, 有斐閣, 1994
- [10] IPA: 非機能要求グレード,
<http://www.ipa.go.jp/sec/softwareengineering/reports/20100416.html>
- [11] IPA: OSS 教材,
<http://www.cybersoken.com/oss10/index.html>
- [12] IPA: SEC BOOKS,
<http://www.ipa.go.jp/sec/publish/index.html#ent>
- [13] 黒田, 大森: 気づきシートを用いた教育研修の分析と評価, 研究報告 2010-IS-113, 情報処理学会研究報告, Vol.2010, No.113, 2010