

プログラム開発体験に基づく上流工程SE育成 カリキュラムの開発と実施

橋本千恵子

日立製作所 ビジネスシステム開発センタ

上流工程SEに要求される能力は(1)問題発見・解決能力(2)システム構築能力(3)コミュニケーション能力(4)プロジェクト遂行能力である。これらの能力開発及び向上を目的とし、企業内新人教育で実施するプログラミング教育を開発した。本教育では概要設計、ユーザ・インタフェース、プログラム書法に重点を置いてプログラムを組ませる。またシステム分析からテストまでのシステム開発演習をグループで実施し、開発手順と作業内容、チーム内コミュニケーションの重要性、ドキュメンテーションの必要性を理解させる。さらに作成システムについての成果発表の場を設け、プレゼンテーション能力の向上も図る。本教育を平成6年度新人教育に適用した。本稿ではその適用状況及び成果を述べる。

DEVELOPMENT AND IMPLEMENTATION OF SYSTEMS ENGINEER CURRICULUM FOR UPSTREAM ACTIVITIES THROUGH PROGRAM DEVELOPMENT EXPERIENCE

Chieko Hashimoto

Hitachi, Ltd.

Institute of Advanced Business Systems

2-2 Kagahara, Tuzuki, Yokohama, Kanagawa, 224 Japan

The following four skills are required for upstream systems engineers:(1)problem finding and solving skills, (2)system constructing skill, (3)communication skill, (4)project collaborating skill. A programming curriculum was made for developing and improving these skills. In this curriculum, the conceptual design, the user interface and the programming notation are emphasized. By going through in groups the systems development exercises from analysis to tests, the students are expected to understand the systems development procedures, the importance of group communication, and the necessity of documentation. Moreover, an occasion is provided to explain the systems which the students have constructed in order to improve their presentation skills. The curriculum was utilized in the education for the first year employees in 1994. This paper describes how the students reacted to the curriculum and what the outcome was.

1. はじめに

当事業所は顧客に対して業務のあるべき姿のデザインや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コンサルティングなどを行っている。このような業務に携わる者には情報技術はもちろんのこと、加えて以下の能力が要求される。

- (1) 問題発見・解決能力
- (2) システム構築能力
- (3) コミュニケーション能力
- (4) プロジェクト遂行能力

ここに挙げた能力は従来OJTにより身につけるものとされ、集合教育では教育不可能とされてきた。特に入社後、実施する新入社員導入技術教育（以下新人教育と略す）では、情報基礎技術の習得に主眼をおき、プログラミング文法の習得のみを目的としたプログラミング教育を実施してきた。

しかし竹田¹⁾は先に挙げた能力はプログラミングの初歩から教育していくことが可能であると発表している。そこで当事業所では竹田の提案した「プログラム開発体験に基づく情報技術者育成カリキュラム」を参考に、「上流工程SEの育成を目的としたプログラミング教育」を開発し、平成6年度新人教育で実施した。本教育では先に述べた能力の向上を目的とする。

本論文では

- (1) カリキュラム作成方針
- (2) カリキュラム内容
- (3) 実施結果
- (4) 今後の課題

について述べる。

2. カリキュラム作成方針

先に挙げた能力をプログラミング教育の中でどのように向上させていくかを検討し、教育方針をたてた。要求能力と教育方針の関連を表1に示す。

表1 要求能力と教育方針

要求能力	教育方針
問題発見・解決能力	・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
システム構築能力	・C言語基礎文法を習得させる ・プログラム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能力	・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
プロジェクト遂行能力	・共同作業を進める上での留意点を認識させる

3. カリキュラム内容

3.1 カリキュラム概要と時間数

本教育では教育を効果的にかつ効率的に行うために、教育過程を初級編、中級編、総合演習の3フェーズに分けた。先に挙げた教育方針と各フェーズとの関連を表2に示す。また各フェーズのカリキュラム概要と予定時間数を表3に示す。

表2 教育方針と各フェーズとの関連

教育方針	初級編	中級編	総合演習
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	×	△	◎
C言語基礎文法を習得させる	◎	△	△
プログラム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	×	◎	◎
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	○	×	◎
共同作業を進める上での留意点を認識させる	×	×	◎

◎：最重要視 ○：重視 △：一部重視 ×：無視

表3 カリキュラム概要と予定時間数（単位：時間）

フェーズ	単元	予定時間
初級編	C言語の基本文法	6 6
	・入力-計算-出力	6
	・実数データの扱い	5
	・制御構造	1 0
	・配列	1 0
	・関数	2 0
	・ファイル処理	1 5
中級編	個人演習による金銭出納帳システム作成演習	4 9
	・住所管理プログラムの解説	5
	・金銭出納帳システム作成	2 5
	・構造体	1 0
	・金銭出納帳システム作成（構造体を用いて）	9
総合演習	グループ演習によるダイレクトメールシステム構築演習	6 8
	・システム分析	8
	・システム計画	6
	・システム設計	1 0
	・プログラム設計	6
	・プログラム作成	1 8
	・テスト	8
	・報告会準備	8
	・報告会	4

3.2 初級編カリキュラム

初級編では一通りの基礎文法を習得させる。講義方法は以下の通りである。

(1) 例題を用いてのアルゴリズム、文法事項の習得

(2) 例題の確認と概要設計

解説した例題の一部を改造，変更する程度の演習問題を与え，概要設計をさせてからプログラムを組ませ文法内容を確認させる。ここではプログラムを打ち込む前に何をどう実現させるのかを考えさせる。この概要設計にはHCPチャート²⁾を用い記述させる。これは物事を系統立てて考える訓練であり，それを文章化させることにより文書作成能力も養う。

(3) 演習

演習問題を複数与え，その中で必須問題をこちらが指定し提出させる。問題は難易度に幅をもたせる。これは受講生が各自のプログラミング経験にあわせて演習が進められるようにするためである。講師はここで提出された結果から受講生の理解度を見ることができる。

(4) 作成プログラムについての公開レビュー

これはすべての演習問題に対して行うわけではないが，プレゼンテーションの練習も兼ね時間の許す限り行う。

これら一連の講義において以下の2点を強調する。

・ユーザインタフェースの設計

入力プロンプト，出力形式，エラー時のメッセージ表示などユーザにプログラムの動きが分かるような設計をさせる。どんなやさしいプログラムでもユーザを意識してプログラムを作成するよう指導する。

・プログラム書法の徹底

インデントの使い方，コメント行の挿入，プログラムヘッダの挿入など他人が見ても分かりやすいプログラムを作成させる。実業務では作成者以外がプログラムを読むことも生じるため，プログラムは作成者以外の人も読みやすいものでなければならないことを理解させる。

このように初級編ではプログラム作成の演習を中心に進めていくためタイプ速度が遅いと講義に支障を来す。そこでキー入力速度を高めるよう初級編では毎日午後の始めの20分間にタッチタイプ練習³⁾を実施する。

3.3 中級編カリキュラム

中級編は先に述べた竹田の提案したカリキュラムを流用する。ここでは800ステップ程の「金銭出納帳システム」を開発させる演習を中心に進める。演習の実施を通じて部品の再利用，プログラムの拡張，仕様の変更といった開発現場で実際に起きる作業を体験させる。また文法事項では構造体を新たに講義する。以下に講義内容をしめす。

(1) 例題プログラムの提示

演習を実施するにあたり，問題となるのは受講生のシステム構築における経験の乏しさである。そこで課題によく似た機能を持つシステムを例題として提示し，画面の動き，

コマンド体系，データ構造，モジュール構造などを理解させる。

(2) 「金銭出納帳システム」の仕様決定

例題プログラムを参考にして，データ構造，画面遷移，モジュール構造を設計する。

(3) 「金銭出納帳システム」の作成(第1版の作成)

ここでは一からシステムを作成するのではなく，例題プログラムを用いて部品の再利用を行う。例題プログラムで流用可能な部分は流用し，変更が必要な箇所は変更し，新たに必要なものは作成させる。

(4) 完成システムの機能拡張(第2，第3版の作成)

例題システムの機能拡張を参考に，(3)で作成したシステムを第2，3版へと段階を踏んで機能拡張させる。

(5) 完成システムのデータ構造の変更

構造体の講義を実施し，その後各自の完成システムに対してデータ構造を配列から構造体に変更させる。変更点はプログラム全域にわたり時間を要する。このような体験を通して初期設計の重要性を学ばせる。

本演習では(3)までを必修課題とし，ある程度完成した時点で構造体の講義に入り(5)へと進む。(3)が早く完成した受講生は(4)へと進ませ，ソフトウェアの進化を体験させる。受講生によって進捗状況が異なってくるが，各自の実力に見合ったものを決められた時間内で作成するよう指導する。また演習の進捗状況は口誌を通じて報告させる。

3.4 総合演習カリキュラム

本演習では自社のシステム開発標準手順を学んだ後，その手順に沿った設計書を作成しながらシステム分析からテストまでのシステム開発をグループで実施する。自社のシステム開発標準手順を学ばせること，ドキュメンテーションの重要性を理解させること，共同作業におけるコミュニケーションの重要性を理解させることが狙いである。内容は以下の通りである。

(1) システム分析

現状システムを説明しどこが問題か，どこを改善すべきか，どこをシステム化すべきかなどのテーマを与え，各グループで話し合わせる。最終的に新システム案をまとめさせ発表させる。

(2) システム計画

こちらから大まかな要求仕様を出す。これらを満たし，加えて(1)の結果を踏まえたシステムの仕様を決定させる。また各工程にかかる人数，日数などの大口程計画を立てさせる。

(3) システム設計

画面入出力設計，帳票設計，ファイル設計，機械処理設計などを行わせる。

(4) プログラム設計，作成，テスト

HCPチャートを用いてモジュール設計を行わせる。プログラムが完成すれば単体テスト，組合わせテストに入らせる。単体テストについてはテストデータも受講生に作成させる。

(5) 報告会

報告会では開発システムの特徴，工夫点，また本演習を通じて学んだことなどを中心に発表させる。

本演習はグループ単位で進めるため，システム仕様，互いの進捗状況などを把握させる必要があり，設計書，進捗管理表を共有させ，それを基にレビューを行わせる。このように基本的には受講生の自主性に任せるが，講師も状況に応じてフォローを入れる。そのためにもグループの進捗状況を講師へ報告させる。これは日誌と口頭の両方で行わせ，コミュニケーション能力を養う。

4. 実施結果

本教育を平成6年度新入社員に対して実施した。本教育実施前に受講生のコンピュータ基礎知識を知るために，事前テストとアンケートを行った。アンケートで，学生時代にC言語を勉強したことがある人は半数であった。テストには50ステップほどのプログラムを読み，4つの穴埋めをさせるC言語プログラミングの問題を出題した。その結果を表4に示す。

表4 事前テスト結果

正解数	4問	3問	2問	1問	0問
正解者率	7%	7%	14%	7%	65%

以下本教育の結果と成果について述べる。始めに本教育の時間数と内容レベルについて述べる。

4.1 実施時間

各フェーズの時間数についての受講生の意見を図1に示す。

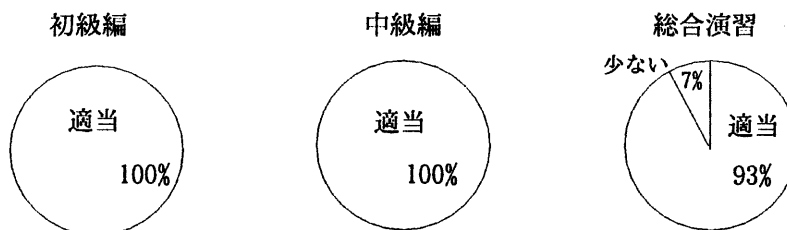


図1 時間数に関する受講生アンケート結果

受講生が今回の時間数をほぼ适当と見なしているとおりに，どのフェーズも全体的に大きな遅れは生じなかった。初級編の演習ではエディタ操作やキータッチに慣れないこともあり予定時間を越えることもあったが，演習を重ねるに連れこれらの問題は解決された。キータッチについては毎日行ったキータッチ練習の効果があつたと思われる。

4.2 実施内容

各フェーズの内容レベルについての受講生の意見を図2に示す。

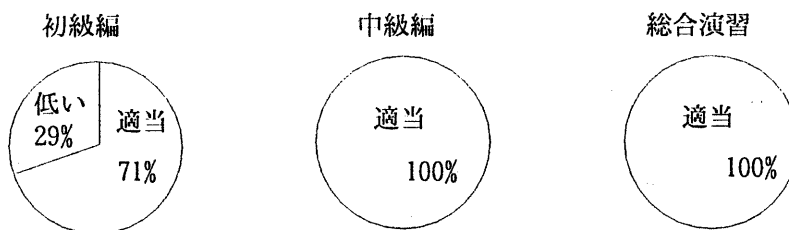


図2 内容レベルに関する受講生アンケート結果

初級編を除いては内容レベルは適当であるという結果がでている。これは3章で述べたように受講生が各自の実力に応じて取り組める演習であったことも大きい。個人により到達目標が違うことは、レベルの違う集団に対する教育に有効であった。一般にプログラミング未経験者は経験者と一緒に教育を受講することに不安を抱いている場合が多い。到達目標が異なることで精神的にゆとりを持って教育に取り組めた、とは受講生の感想である。課題の提出状況や試験結果などからも今回の内容レベルはほぼ適当であったと言える。

4.3 教育方針についての考察

次に2章であげた教育方針に対する達成度を検証する。

(1) プログラム作成における概要設計を重視する

初級編ではプログラム経験者に概要設計の重要性を理解させることができなかった。原因としては初級編の演習レベルが低かったこと、個人でプログラムを作成するため特にドキュメントの必要性が感じられなかったことが挙げられる。しかし中級編以降はレベルも上がり、さらに総合演習は共同作業である。これらの演習では、モジュール分割を必要とするプログラムでは最初の概要設計が重要であること、また概要設計書を読めば他人が作ったプログラムもある程度理解できることなどを実感させることができた。提出された概要設計書も初級編に比べて詳細なものであった。また教育終了時のアンケートでは設計の重要性について、よく理解できたが36%、理解できたが64%という結果であった。

以上のことから本教育全体を通じて、受講生はプログラムを作成する上での設計の重要性を認識できた。また対象がプログラミング経験者であれば、ある程度難しい課題を与える、共同作業を経験させるなどの配慮が必要なことも分かった。

(2) C言語基礎文法を習得させる

教育終了時に教育成果を見るために事後テストを実施した。その中で事前テストと同レベルのC言語問題を5題出題した。その結果を表5に示す。

表5 事後テスト結果

正解数	5問	4問	3問	2問	1問	0問
正解者率	43%	43%	14%	0%	0%	0%

表4の事前テスト結果と比較してその違いは明らかである。提出課題についても、未提出者は

なく全員が合格であった。以上のことより基礎文法を習得させるは達成できた。

(3) プログラムを系統的にとらえる

初級編を通じて

(a) ユーザが使いやすいプログラムを作る

(b) 他人が読みやすいプログラムを書く

の2点を強調した。

(a) については徹底が図れた。これは入力プロンプトの表示、見やすい結果表示、入力に対するエラーチェックなどが演習結果に見られたことからわかる。また他人や他グループとの差別化を画面の見やすさ、システムの使いやすさなどで図ろうとする気風もあり、本教育中「人に優しいプログラム」というフレーズが流行語にもなったほどである。他人や他グループとの競争意識が(a)についての意識を高めたように思う。

(b) については他人のプログラムを読む機会が多々あったことから、その重要性を認識させることはできた。しかし中級編以降の演習ではプログラムの機能向上に時間を取られ、コメント行の挿入を怠る人が出てくるなど徹底は図れなかった。これはアンケートや成果発表会でも反省点として挙げられていた。

(4) システム開発一連の作業を理解させる

初めに中級編の演習結果について述べる。表6は完成プログラム状況である。必須課題は第1版までとしたため全員が合格である。

表6 中級編演習の完成状況

完成プログラム	第3版	第2版	第1版
完成者率	72%	21%	7%

最初のデータ構造設計、画面設計については設計書を作成させ、それを基に講師とのレビューを行った。持ってきた設計書を見ると、早く端末に向かいたい一心からディスプレイに入りきらない画面設計、画面遷移があいまいなものなどいい加減に書いてあるものもあり、これらは何度も修正を入れさせた。

既存のプログラムを部品として使用する作業は思ったより難しかったようだ。よく似た構造を持つプログラムを改造するだけなので、簡単にできると思ったのだろう。しかしデータ構造、画面表示が変われば変更点多岐に及び関数の引数、戻値も同じものになるとは限らない。第1版、2版までしか終わらなかった受講生は、ひとつにこの関数間のデータ引渡ししがうまくできなかったことが挙げられる。もうひとつとして、モジュール設計ができなかったことが挙げられる。長いプログラムでは適切なモジュール分割が必要だが、モジュール間のつながりを把握できない設計力の弱さが原因である。

次に総合演習の実施状況について述べる。

総合演習ではシステム開発における各工程を体験させることができた。またどのグループも提出日までに要求仕様以上の機能を持つシステムを作成できた。ここで下流工程だけでなく上流工

程を体験させたことにより、実務との繋がりを持たせることができ、受講生に実務を意識させる効果もあった。

各工程作業は初めての体験のため失敗も多々あったがその都度グループで話しあい、時には講師から助言を得て問題を解決していった。

システム設計での画面設計、ファイル設計などは中級編で一度経験済みのこともあり、講師からの細かい指示は必要なく計画通りに進んだ。しかしプログラム作成は仕様に凝りすぎたこともあり定時間後残って作業をするグループも出てきた。仕様の決定段階で定時間内で作成可能なシステムに変更させるべきかは講師の判断によるが、今回は本演習に対する受講生の熱意を尊重し、定時後の作業も認めた。事実、プログラム作成に夢中になって取り組んでいる印象が強く、その結果どのグループもこちらが予想していたものよりレベルの高いシステムに仕上がった。

(5) システム分析能力を養う

中級編で自分で仕様を決めシステムを設計するといった作業を行ったが、本格的なシステム分析を実施したのは総合演習のみである。前述の通り分析までの手順を提示したこともあり、どのグループも長時間の話し合いの末、新システムをまとめることができ、これを発表させた。提案内容の大枠は後で講師が提示する要求仕様に近いものであり、ほぼ的を得た分析であった。しかし細かい所になると分析不足があり、発表後の質疑応答で講師や他グループからあいまいな点、実現が難しいと思われる点、改善点などの指摘が見られた。

今回の教育ではシステム分析能力に対する評価基準を設けていない。そのためこの演習のシステム分析能力を養うことに対する効果は図れなかった。しかし受講生にとって貴重な体験であったことは日誌やアンケートの感想欄から読み取れた。

(6) 共同作業を進める上での留意点を理解させる

総合演習の成果発表では、大きく分けて2つの反省が挙げられた。ひとつめは、グループ内コミュニケーションの不足である。そのために受講生が経験した失敗例を以下に挙げる。

- (a) 単体テストで動くことを確認した各自作成のプログラムを組み合わせ一つのシステムにすると動かない。
- (b) システムとして動いてからも画面に統一性がない、入力方法が統一されていない、出力メッセージが統一されていない。
- (c) プログラムを作成していく過程で仕様を変更したが、それがグループ内に伝わっていない。

(a)、(b)はファイル設計、画面設計といったシステム設計の段階で、(c)はプログラム作成の段階でグループ内の統一が図れていないためである。これらは全てグループ内レビュー不足が原因である。

またグループ進捗を講師に報告する際、報告者が状況を把握していないことがあった。これはプログラム作成に入ってから多くみられた。これも各自の作業のみに集中し、互いの進捗を報告しあうといったコミュニケーションを怠ったためである。

ふたつめはグループ作業の意識が欠けていたことである。一人の作業効率の低下がグループ全体の作業効率を低下させる。このように個人がグループ全体に及ぼす影響を実感したようだ。し

かし各自の作業速度は個人の能力によるところも大きく、最初の作業分担が適切であったかも問われるところである。

初めての共同作業ゆえこのような反省が挙がることはあらかじめ予想できた。こちらの狙い通りであり意味のある失敗といえる。コミュニケーションの重要性、共同作業における個人の責任の重さなどは説明して分かってもらえるものではない。これらを理解させるのに実体験に優るものはないであろう。試行錯誤しながらの体験が最も説得力を持つことが分かった。

(7) ドキュメンテーション能力を養う

設計書、日誌、プレゼンテーション資料と本教育を通じてドキュメントを作成する機会を与え、毎日何かを書かせる訓練をした。

そのなかでも日誌をつける作業は文章を書くことに慣れていない受講生にとって苦痛であったようだ。書いてきた文章を見ても主語、述語がはっきりせず意味不明のもの、単なる感想のみで作業報告になっていないもの、1行ほどの極端に短いものなど日報としての役割を果たしていないものが多かった。講師は日誌を毎日添削し受講生に返却した。これを2ヶ月続けた結果、教育終了時には多少の改善はあったものの、全員が分かりやすい文章を書けるようになったとは言いがたい。

また総合演習で用いた設計書はグループで共有するものであり、読みやすい文書を書くことが要求される。仕様をレビューするにも正確に書いた設計書がなければ難しい。しかし総合演習の発表で設計書の軽視を反省に挙げたグループがあったように、作成された設計書は読みやすいものとは言えなかった。

以上のことから本教育によりドキュメンテーション能力に大きな向上があったとは評価できない。しかし本教育はドキュメンテーションの重要性を認識させることに効果があったと言える。

(8) プレゼンテーション能力を養う

先に述べた通り、講義の中でプレゼンテーションの場を何度か設けた。講師からは発表内容とは別に、プレゼンテーションにおける注意点で気付いたところはその都度指摘した。受講生各自もコンサルテーションといった実務を意識してか、問題意識と向上意識を持ってプレゼンテーション能力の向上に努めていることが、日誌などから読み取れた。

最後の総合演習の成果発表は発表準備の時間も設けた本格的なものであり、これは本教育を通してのプレゼンテーション能力に対する成果発表の場でもあった。これを聞いた講評者の意見をまとめる次の2つになる。

- ・発表内容にまとまりがあり、言いたいことが明確であった。
- ・発表資料が見やすくわかりやすいものであった。

講師から見ても教育の始めの頃と比較して発表内容、発表資料、加えて発表態度に進歩が見られた。これは教育中にプレゼンテーションを多く実施した効果と言える。

5. 今後の課題

本教育の実施結果から課題に挙げられるものを以下に述べる。

(1) デバッガに関する講座の導入

アンケートによると、本教育で一番苦勞した点として多かったのがデバッグ作業である。プログラム作成に時間がかかるという回答もあったが、これもデバッグ作業に費やす時間が多かったからであろう。前述の通り中級編、総合演習に入ってからほどの箇所エラーが発生しているのかがわからないため、作業が停滞する受講生も出てきた。その解決策として s d b * の簡単な操作を教えたところ、これがその後のプログラム作成に大きな効果を発揮した。

本教育はプログラム作成における設計、プログラム書法、インタフェースなどを重視している。しかしデバッグ作業に大部分の時間をとられ、これらの重要事項がおろそかになることは望ましくない。

以上のことから、来年度はデバッガに関する講座を実施したい。実施時期は初級編終了時が適当と思われる。また講義時間は8時間あれば、概要と簡単な操作方法は習得できる。今回のキータッチ訓練と同様、プログラムの作成効率向上に効果があると思われる。

(2) コミュニケーション能力における講師の指導力

既に述べてきたように、本教育ではコミュニケーション能力の向上も目的の一つである。その手段として受講生に書かせる、発表させるなどの機会を多く設けた。しかし場数を踏ませるだけでなくその時々で講師による適切な添削、助言が必要である。またこれらの能力に対する評価は講師の主観によるところが大きく、明確な評価基準や指導要綱が必要である。

以上のことから、講師はプログラミング技術だけでなくコミュニケーション技術にも長けていることが望ましい。特に業務経験が浅い講師はコミュニケーション能力の指導は難しく、何らかの対策が必要である。対策としては以下のものが挙げられる。

- ・講師の能力向上を目的とした研修の実施
- ・コミュニケーション能力指導マニュアルの作成
- ・コミュニケーション技術を専門に指導するサブ講師をつける講師編成

6. おわりに

今回上流工程 S E の育成を目的としたプログラミング教育を開発し実施した。以下はじめに挙げた能力向上に対する成果を述べる。

(1) 問題発見・解決能力

本教育が本能力の向上に与えた効果は評価できなかった。しかし概要設計を通じて問題が提起されたとき目的を明確にし、解決までの手段を考える訓練は実施できた。これより考える過程の重要性を認識させることはできたと思う。

(2) システム構築能力

初級からの一連の演習作成過程とその結果から、本教育を通じて本能力を高めることができたと判断する。

* s d b symbolic debugger

(3) コミュニケーション能力

プレゼンテーション能力は向上が見られたが、ドキュメンテーション能力の向上は受講生全員には見られず、課題が残った。しかし、総合的に見て本教育は本能力向上に効果があったと判断する。

(4) プロジェクトを遂行する能力

共同作業の体験によりプロジェクト遂行のための留意点を認識させることができた。これは本能力の開発につながったと判断する。

以上のように一部目的は達成できなかったが、本教育を通じて受講生には大きな技術的成長が見られた。加えて共同作業を通じての協調性の向上やリーダーシップの発生、プログラム達成までの根気強さなど、精神的成長も見られた。

またコンピュータに対する苦手意識を与えることなく、積極的に取り組める演習課題であったことは本教育で特筆すべき評価である。

最後に本カリキュラムの作成にあたりご指導いただいた慶応大学環境情報学部の大岩元教授に感謝の意を表する。

参考文献

- 1) 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 pp. 944-954 (1992-7)
- 2) 長野宏宣監修, 浅見秀雄, 忠海均著: HCPチャート-階層化プログラム設計図法-: 第2章, オーム社 (1993)
- 3) 大岩元著: 5時間10分キー入力習得法: マグロウヒル出版 (1990)