

システム開発マネジメントを意識した プログラミング教育の試み

A Trial of the Programming Education for System Development Management Education

木 下 和 也

Kazuya KINOSHITA

和文要旨

筆者が担当している講義科目であるプログラミングおよびシステム設計を中心に、経営学部学生にシステム開発を意識し職業意識を持たせる授業設計を試みた。これらは、授業だけではなく課外活動との連携で実現されるものである。教育効果としてプログラミング・スキルと同時にエンジニアとしての知識を身につけさせることが期待されるが、結果として、この教育プログラムと課外活動は、参加した経営学部の学生に一種の職業人としてのエンジニアの意識をもたらすことができた。

英文要旨

This paper shows a technical education program and extracurricular activities for students in faculty of management, which consists of computer programming and system development. Education results show that this technical education program including extracurricular activities can give excellent effect to obtain the programming skill and engineer's knowledge. In addition, this education program and activity lets students experience a kind of simulation-like occupation. These education results indicate that this education program is an effective education system for faculty of management students to obtain the sense as the careerist.

和文キーワード：プログラミング, 適性, 学習, 課外活動

英文キーワード：programming, fitness, learning, extracurricular activities

目 次

はじめに

1. 学生の情報関連知識とプログラミングの理解
2. システム開発を目的とした授業における学生の理解度
3. 課外活動の活用

まとめ

はじめに

システム開発に必要な知識やスキルを経営学部の専門科目として他の科目とより馴染んだものとするため、筆者が担当しているプログラミングの授業を主として活用できないか検討して

きた。プログラミングという科目は、経営学部においては、その他の専門科目とどのように関連があるのかが学生にとっては曖昧でわかりにくいようである。むしろ、パソコンを学ぶといったリテラシーに近い印象を持っている学生が多い。積極的にマネジメントの分野であることを

アピールするような授業や、それを取り巻く環境を構築すべきであろう。筆者がこれまで行ってきた授業とそれを取り巻く課外活動の目的はまさにそれであり、以下では、この活動の経緯や、結果について述べる。

1 学生の情報関連知識とプログラミングの理解

1.1 事前知識との関係

現在の学生は、中学から高校にかけて体系的な情報に関する知識とスキルを身につけてから入学してきている。これは、高等学校における普通教科「情報」の新設によるところが大きい。以下は高等学校学習指導要領より抜粋したものであるが、情報という科目が高等学校で必要となった経緯を述べることで、大学で学ぶ学生の事前知識について述べたい。1996年7月に、中央教育審議会は「21世紀を展望した我が国の教育の在り方について」と題する答申を行った。その中で、これからの情報社会に生きていく子供たちにどのような教育が必要か、また、教育の改善・充実のためにコンピュータや情報通信ネットワークをどのように生かしていくのかの観点から、情報化と教育について推進すべきこととして、次の4点を示した。

- ①情報教育の体系的な実施
- ②情報機器、情報通信ネットワークの活用による学校教育の質的改善
- ③高度情報通信社会に対応する「新しい学校」の構築
- ④情報社会の「影」の部分への対応

この答申から始まった改革の目玉の一つが教科「情報」の創設である。情報教育調査研究協力者会議の第1次報告は、情報教育の目標を「情報活用の実践力」、「情報の科学的な理解」、情報社会に参画する態度」の3つの観点にまとめている。情報活用の実践力は、課題や目的に応じて情報手段を適切に活用することを含めて、必要な情報を主体的に収集・判断・表現・処理・創造し、受け手の状況などを踏まえて発信・伝達できる能力のことである。また、情報の科学的な理解は、情報活用の基礎となる情報手段の特性の理解と、情報を適切に扱ったり自らの情報活用を評価・改善するための基礎的な理論や

方法の理解のことである。さらに、情報社会に参画する態度は、社会生活の中で情報や情報技術が果たしている役割や及ぼしている影響を理解し、情報モラルの必要性や情報に対する責任について考え、望ましい情報化社会の創造に参画しようとする態度のことである。

これらの観点から新設された普通教科「情報」は必修科目であり、「情報A」、「情報B」、「情報C」の3科目で組織されている。したがって、現在入学してくる学生は、少なくとも3科目のうちどれか一つを選択し学んできていることになる。同時に日常的にパソコンや携帯電話などに接しているため、情報機器への苦手意識などは感じられなくなってきている。

1.2 学部専門科目との関連性

以前であれば、情報系の知識やスキルを身につけていたのは、商業高校などの専門科目を学んできた学生のみであり、とくに情報処理科出身の学生は、プログラミングの知識もある程度身につけていた。普通科高校出身の学生は、趣味で使える程度の学生はいても、体系的な知識に乏しく、苦手意識を持つ学生も多数存在していた。

このような段階での情報教育では、ユーザとしてのスキルを中心に教えるリテラシー教育が中心に行われたが、2006年度以降に入学した学生については、すでに実習を含めて高校までに体系的な知識とスキルを学んでいる。そのため、商業高校出身の学生以外であっても、より専門的な情報教育、とりわけシステムやソフトウェア開発に関する教育が行いやすくなっているといえる。プロジェクトマネジメントを扱うようなマネジメント教育は経営学部との親和性は高く、多くの専門科目と関連がある。したがって、プログラミングなどの専門科目では、スキルアップ中心の授業内容に偏らず、マネジメントの知識を織り交ぜて行うことが、科目の関連を学生に実感させやすいはずである。

木下[3]および[4]で述べているように、経営学部の科目の中で、そのほかの科目とは関連のない孤立した知識を学ぶ科目であると学生が認識してしまえば、学生が何のために学ぶのか目標を見失ってしまう。情報リテラシー教育ではなく、あくまでも、開発に関連したマネジメ

ントの分野に直結した科目であるという印象を与えることが、プログラミングを含む情報関連科目の専門性が経営学部専門科目としての関連性を学生に理解させやすいものと考えている。

2 システム開発を目的とした授業における学生の理解度

2.1 授業における適性と理解に関する問題

学習理解度に関してシステム開発に不可欠な知識であるプログラミングの大きな特徴は、適性に左右されるということである。最終的に理解できたとしても長い時間をかけなければならない学生と、そうではない学生が存在する。どんなにわかりやすい授業を考えても、適性によっては、なかなか理解してもらえないことが多い。

いかなる理由であれ、現状として適性がなくてもプログラミングやシステム開発に興味を持つ学生は増える可能性はあるし、今日のような情報化社会においては、利用者としての知識だけでなく、開発者としての知識の基本を持つべきであろうから、できるだけ幅広い適性の範囲で多くの学生に、開発にかかわる基本的な知識を教えることに努力すべきであろう。もちろん、その一方で、適性と関心の高い学生には、より専門的で高度な内容を教えることで、情報化社会の担い手となって欲しい。

そもそも適性の有無は0か1というようなものではなく、段階的に捉えられてよいものである。適性が高いか低いか、その中間なのか、適性にはグレードがあり、授業についていけるかどうかでそれを判断するのは難しい。プログラミングの授業スタイルは毎回新しい知識の積み重ねによって進むため、どこかで躓いてしまうとその先に進むことができなくなってしまう。躓いても深く追求し理解して次の授業に備えれば、まったく問題は生じないが、そこまで懸命に取り組む学生が多いとはいえない。

IT開発マネジメントを理解し、体験するまでに学生を引き上げるためには、基本的なプログラミングスキルを持つ学生の母数をできるだけ多く確保しなければならない。スキルを持ちそれを活かせることが、開発マネジメントの理解を促すからである。

2.2 適性の影響を排除するためのプログラミングの学習

ここで、プログラミング言語の特徴を述べるために、その比較対象としてHTMLについて簡単に述べたい。HTMLの特徴は文字や画像の配置及びリンクを基本として、タグの使い方を理解するだけで多様なWebページ作成はできる。しかしながら、プログラミング言語を理解するためには、言語特有の書き方だけではなく、さまざまな処理のアルゴリズムを理解しなければならない。多くの場合、プログラミングを不得意とする学生はこのアルゴリズムを理解できていない。

HTMLとプログラミング言語を比較すればわかるように、1行ごとに画面表示として結果を返すHTMLに対して、プログラミング言語は複数行に渡って構成されるループ処理や、サブルーチン、関数ごとに意味のある処理がなされるため、処理の流れを理解する必要がある。

このようなアルゴリズムの理解には、前述のように適性の有無が大きく関わってくるが、初級プログラミングにおいては、練習によって理解度を上げることができるといふ仮説を立て、反復練習を取り込んだ多言語によるプログラミング学習を展開した。多言語によるプログラミング学習は、同じアルゴリズムを複数の言語で繰り返し練習することである。

適性が比較的低い学生にも興味やモチベーションを持たせたまま、適性のある学生に退屈させず同様のモチベーションを維持させるプログラミング教授方法を求めて試行錯誤した結果、この2年間である程度の成果を得た。それが多言語学習である。プログラミング言語が異なっても、アルゴリズムは同じであるため、言語を変えてアルゴリズムを復習することが目的となる。複数回繰り返すことで退屈する学生が現れないように、言語を変えることによって対応する。言語については、それぞれ特徴的な記述方法や互いに利用目的の異なるものを用意することが学習意欲を高めることにつながるようである。

筆者の授業ではBASIC、C、Perlの3言語を順に利用する。これらの言語を選択した理由は、記述方式に明らかな違いがあり、言語の特徴が分かりやすいこと、それぞれの言語の誕生時期の違いにより使用環境や利用目的の違いが大きい点からも言語の違いを理解しやすいからである。

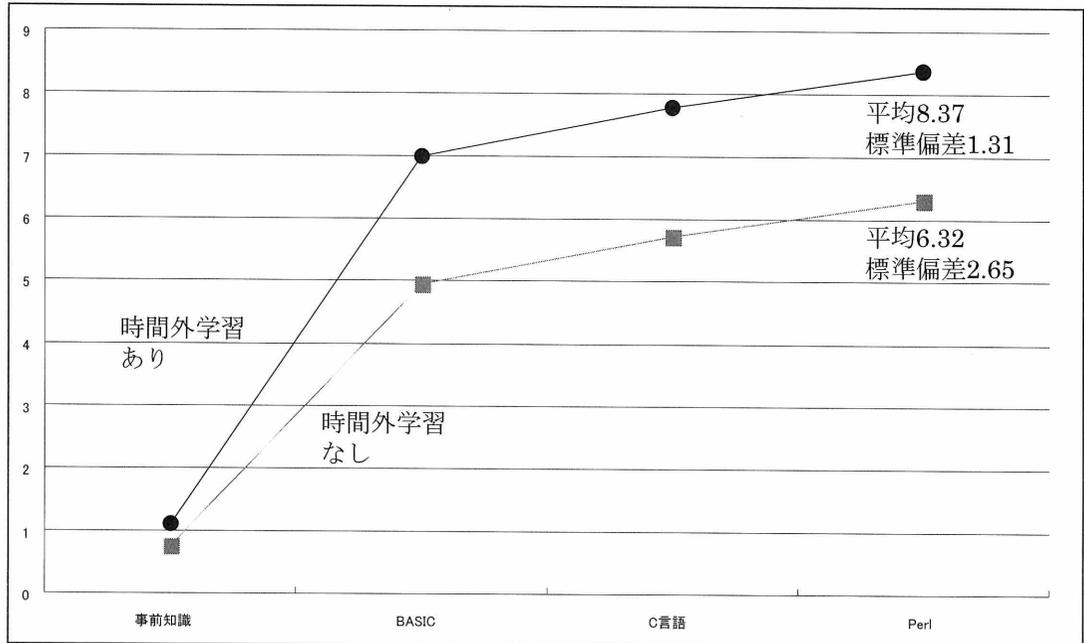


図1 アルゴリズムの理解度の推移

多言語プログラミング学習を導入して3年目になる2009年7月の授業において効果を測定するためのアンケートを実施したが、そのアンケートを分析した結果、図1のような結果を得ることができた。

全体的な傾向として繰り返し学習するたびにアルゴリズムの理解度が向上していることが分かる。さらに詳細にみると、時間外学習している学生としていない学生との比較においては、どちらも事前知識はほとんどない状態からスタートし、最終的にはある程度の理解に達しているといえるが、時間外学習を行っている学生の方が明らかに理解度が高い。さらに、最終的な理解においては、標準偏差が小さく、ほぼ同じような理解度に達していることがわかる。時間外学習をしていない学生の標準偏差は比較的大きく、授業だけで理解できる学生とそうではない学生の差が見て取れる。したがって、時間外学習の有無に関わらず、反復練習による理解度の向上には効果があるが、時間外学習を行わなければその効果が薄れることがわかる。この時間外学習の効果については、2.3節で述べる。

この方法を導入して3年目となるが、それ以前と比較すると学期半ばで授業に来なくなった学生が明らかに減少しており、最後まで興味を

持たせ継続させるという点においても効果があったといえよう。その意味で、プログラミング・スキルを活かした職業への興味とIT開発マネジメントへの関心が深まる素地を作る授業になりうるものと確信している。

問題点としては、授業内容がこの学習法を導入する前と比較して明らかにレベルダウンしていることである。これを補うための手段として課外活動を通じて、より興味のある学生への対応を行っている。第3章ではこの課外活動に参加する学生の活動が、さらに授業へフィードバックされるという、授業を取り巻く環境とネットワークの活用について述べてい。

2.3 システム開発関連授業における学習効果の測定方法

上述の時間外学習の効果を測定するための手法として今後取り入れたいと考えている事例について述べてい。先行研究として兼宗・長[6]においては、eコマースプログラミング論という授業において、eコマースサイトを構築するプログラミング体験を通して、実際のeコマースシステムの仕組みを学ぶことを目的とした授業を展開し、そこで得た学生の特性を分析している。この分析にはすべてをWeb画面上で構築

でき、学生の行動を分析するためのアクセスログを用いて、学生の行動と学習効果の相関が測定できる点で画期的だといえる。

本研究のエッセンスについては筆者が担当しているシステム設計などの授業で応用できると考えている。この測定方法は、自習率とクリエイティブ指数を使用した相関と、失敗率とクリエイティブ指数を使用した相関によって、グラフ上の学生の位置を把握することで、理解度を決定することができる(図2及び図3参照)。

この2つのグラフから、クリエイティブ指数が低い学生の中には失敗率の低い学生と高い学生が混在しているが、クリエイティブ指数が高い学生は失敗率が低く安定していることがわかる。さらに、自習率が高い学生ほどクリエイティブ指数が高いわけであるから、興味関心を持って真面目に取り組めば失敗率が下がる。しかし、能力や適性などによっては失敗率は下がってもクリエイティ

ブとはいえない作品しか作れない学生もいるということが明確になったわけである(注1)。

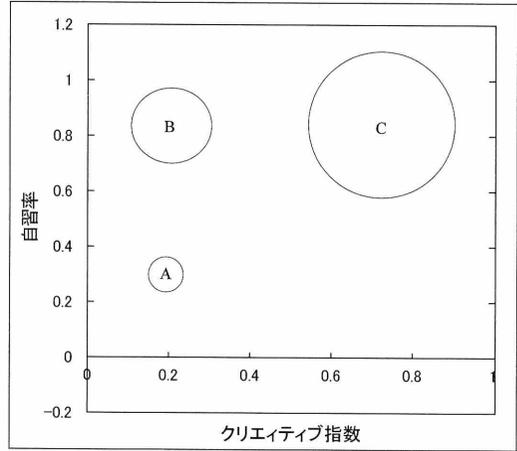


図4 理想的な被験者の分布

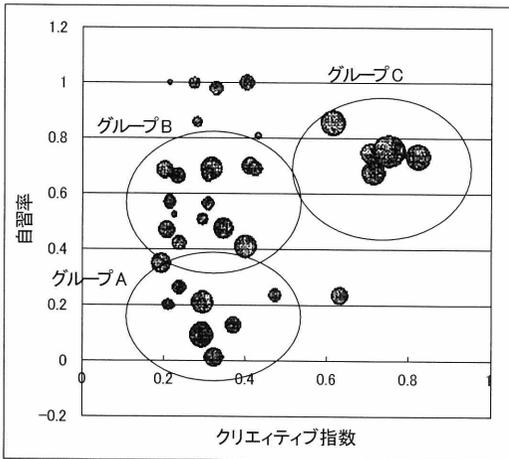


図2 自習率とクリエイティブ指数
(出所 兼宗・長[6] P.146)

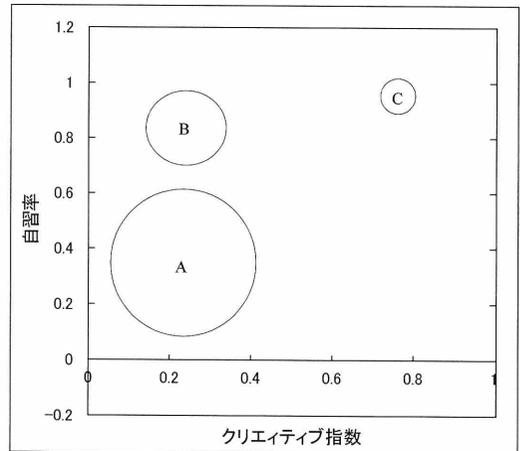


図5 授業困難な状況を表す分布

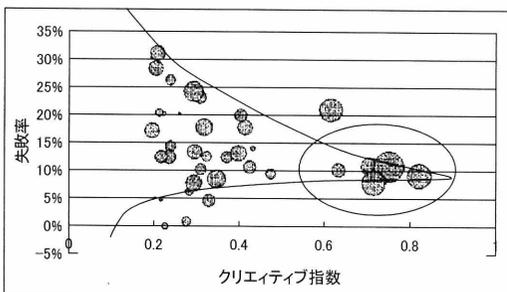


図3 クリエイティブ指数と失敗率
(出所 兼宗・長[6] P.147)

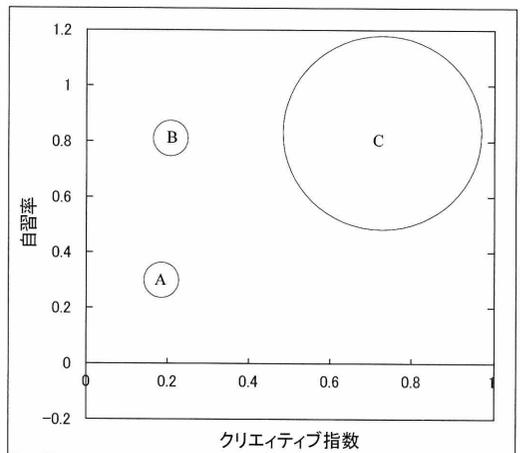


図6 より理想的な被験者の分布

このグラフを筆者の授業に人数配分を考慮して適用すると、図4、図5、図6の様に描くことができる。

失敗率とクリエイティブ指数の関係から、Aグループの中には失敗率の低い学生も含まれるわけであるから、Aグループの中から脱落者を出さない方法を考えることがCグループの増加に繋がると考えられる。また、Bグループの学生はいわゆる真面目なタイプであるといえるが、お手本に忠実に、真面目に作成するタイプといえる。このような学生は、アイデアの源泉をどこかに探してやる必要があるであろう。

ここでは、外的な原因による学生の低い意欲について、これを向上させることは述べない。しかし、たとえ外的な要因で意欲が低い学生が多くなったとしても、そうではない学生も含まれているわけであるから、少しでも理想となる分布を目標に教員が努力しなければならないことは言うまでもない。必要なことは、常にこのような分布がどのように変化しているかを教員が注意深く見ていくことであり、その行為が理想の分布へと近づけることになろう。その意味で、この概念を授業評価に取り入れる試みは意味がある。

3 課外活動の活用

3.1 プログラミングの魅力を伝える機会の創出

設備等により人数制限などの制約があるため、プログラミングを扱う授業だけで、興味のある学生を募り教育することは不可能である。

そこで、課外活動とゼミおよびプログラミングをはじめとする関連科目を連携させ、授業で学ぶ学生へ、さまざまな情報をフィードバックすることを試みた(図7を参照)。

本来経営学を学ぶつもりで入学してきた学生に、一見して経営学とは異なる印象の科目名であるプログラミングを、経営学と同様に魅力ある科目であることを伝え、将来の職業の選択肢を広げることから、この実験的活動は始まった。

3.2 課外活動と授業の連動

課外活動は2005年から本格的に開始したが、このときの学生の共通点はプログラミングの授業を履修していないことと、筆者のゼミ所属で

はないということであった。このことが、ゼミや授業と関係なく興味があれば誰でも休み時間や放課後に学べるという印象を作ったと考えられ、その後の学生募集のヒントとなった。つまり、課外活動が完全に授業と分離している状態でスタートしたのである。

活動内容は、それぞれが、ノートパソコンを持ち寄り、比較的小規模の教室で自主的に学習することであった。筆者がある程度の基本的な解説と、何らかの課題を与え、それを自主的に開発するためのプログラムに作り上げることが基本的な学習スタイルであった。

このようなスタイルで学習しながら、成績評価とは異なる方法で自らの実力を試し、また達成感を得る機会を設けることが次のステップであった。それが学部内のシステム開発に取り組むことであり、この成果を、プログラミングの授業で報告したことが、新たな参加者が集う契機となった。また、課外活動で開発されたシステムを実際に提示して、プログラミングの授業を実施するスタイルも生まれた。

3.3 課外活動内容と学生の行動

課外活動として活動を始めたのが2005年である。創設時のメンバーはプログラミングの経験もなく筆者のゼミの所属でもないため、まったくの未経験者であった。しかし、卒業までの限られた時間の中で何らかの成果を出さなければならず、開始後半年で開発プロジェクトに挑戦することとなった。その際に請け負ったのが事前登録システムの開発である^(注2)。構造は比較的単純であるが、登録者の認証や二重登録の

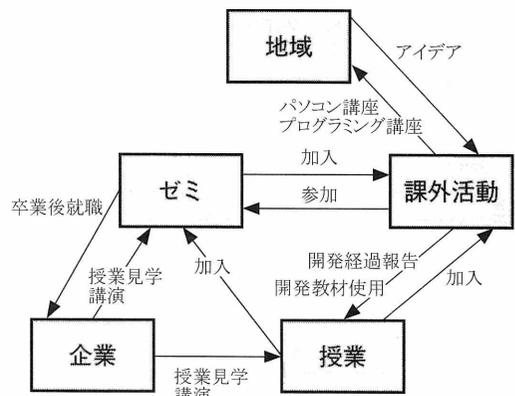


図7 課外活動とその周辺の関係

禁止などのプログラムの工夫、わかりやすいインターフェース、信頼性と安定性など、多くの部分で不安を抱えたまま、学部の正式なシステムとしての開発が始まった。参加する学生の実力や適性がまちまちである上にチームワークの問題など多くの問題を抱えた開発であったが、2006年秋学期に正常に稼働し、このことがメンバー全員の自信となった。また、学生が直接利用するシステムであるため、学生による開発であることが口コミとなったこともあり、興味を持つ学生が増える契機となった。しかしながら彼らは、この開発を最後に、後輩を育てる間もなく卒業することとなった。この段階では活動すべてに教員の存在が不可欠で、知識が蓄積したり、継承したりすることができない状態であったため、年度を越える段階で最初からやり直しという状態であった。

また、学生であるがゆえに避けられないことは卒業という年限である。そのため、たとえ個人のスキルが向上したとしても卒業してしまえば、それ以上の成果は望めないと思われていた。しかしながら、組織への所属感が、上級生からの知識習得の意欲を高め、さらに上級生にとっても下級生への知識継承の意欲が高められているように思われる場面を多く見てきた(図8参照)。

2009年現在の課外活動における学生のスキルはこれまでの最高レベルに育っている。個人的にスキルアップしたわけではなく、確実に上級生の影響を受けて現在のスキルが身につき、さらに組織としての総合的な実力も向上している。この状況は組織としてのスキルが学生そのものを媒体として蓄積している状態と捉えられるであろう。

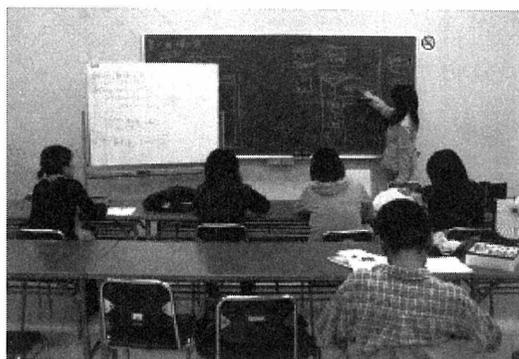


図8 学生が自主的に後輩指導に当たる課外活動の様子

この要因のひとつは第2期及び第3期メンバーに教職課程の学生がそれぞれ一人ずつ所属していたことが大きいように思われる。その意味で教職課程との連携も今後の選択肢の一つといえよう。

3.4 課外活動による学生の開発事例

プログラミングに興味を持ち課外活動に参加している学生は学部内のシステムを中心に要件定義から開発・運用にかかわっている。彼らの活動は成績として評価されるわけではなく、自発的に活動していることが特徴である。その活動のうち代表的なものを以下に挙げる。

- ①履修登録の一部である人数制限科目の Web 登録システム (事前登録データ収集システム)
- ②携帯電話による出席管理システム
- ③歯学部附属病院の消耗品・歯科材料の発注納品在庫管理システム
- ④携帯電話によるアンケートシステム
- ⑤時間制作成システム
- ⑥学部コース登録システム
- ⑦文学部の事前登録データ収集システム

特徴的なことは、学部内のシステム開発だけではなく、学内の他学部においてもその活動を広げていることにある。学生はこれらの活動を仮想的な会社として行っており、その意味で経営学部という環境でプログラミングを学ぶことに大きな意味があると思える。

3.5 学生によるパソコン講座およびプログラミング講座

図7に示されている地域貢献として、パソコン講座やプログラミング講座を実施している。講座内容は筆者が協力したものの、大半は学生の発想によって構成されている。さらに、授業で用いる教科書や教材も学生が協力して作成し、授業そのものも学生が行う形式であった。

2008年度及び2009年度と複数年度に渡って実施された講座であるため、卒業により学生が入れ替わっているが、第1回目の講座では教職課程に所属している学生が授業を担当した。この授業にはケーブルテレビの取材も入るなど、地域への貢献が全面的に押し出されたものとなった。教職課程の学生が参加したのはこの第1回目の講座だけで、翌年度は教職とはまった

く関連のない学生だけで実施されたが、第1回目の講座においてアシスタントとして参加していた学生がこの役を引き継いでいたため、特に問題は生じなかった。

第3回目の講座はこれまでのパソコン講座ではなく、プログラミング講座として企画された。この講座では、これまで以上の準備が必要となり、学生による事前準備はさらに努力を要するものとなった。その結果、大きなトラブルもなく、多くの参加者に受け入れられるものとなった。また、プログラミング講座に関しては、それ以前のパソコン講座と参加者の年齢層や参加目的がまったく異なるものであり、さらに多くの知見を得ることができ、さまざまな内容での地域貢献の可能性が広がった。

特にプログラミング講座の場合、学生によって作成された教科書が、著者の授業でも使われることにもなり、学生の課外活動を介して正規の授業に多くの経験と知識がもたらされることとなった。

3.6 卒業生及び現役エンジニアによる講演

課外活動およびゼミの学生が卒業後に積極的に、学部教育にかかわってもらうことも、キャリア教育の観点から非常に重要なことである。システム開発関連企業の社長講義などは、卒業生を通して招聘することが多い。また、比較的規模の大きな会社の場合、このようなことが困難であれば、卒業生本人が先輩として報告会を開くこともある。この場合、就職活動の経験を含み業界の説明などを行うため、より身近なキャ

リア教育となって、プログラミングを学ぶ意義が伝えられている。

プログラミングの授業では主に名古屋市内にあるシステム開発関連企業の社長または現役のエンジニアに、1年に1度か2度、この業種についての業務とSEの職務についての講義を依頼している。企業は毎回変わるもので、前年に履修した学生も出席し、複数の企業の講義を聴くことで就職活動に役立っている。協力いただいている企業は、筆者のゼミまたは課外活動に参加していた卒業生が勤務している。そのため、筆者の活動や授業の内容について理解していただいている。プログラミングの授業が社会に対して、また将来の職業に関して、どのような関係があるのかを理解できるよう配慮された内容となっている。今ではこの講義がシステム開発マネジメントとプログラミングの授業をつなぐ大きな役割となっている。

3.7 課外活動における学生の活用の教育効果

学生の理解度と同様に、スキルと難度との関係から、学生のモチベーションについての関係もわかってきた。とくに課外活動では、学生が参加するシステム開発において、彼らの達成感と負担には図10のような関係があるものと推測される。この関係をうまく利用することで学生のモチベーションを上げ、さらにスキルレベルを効率的に上げる手立てになるであろう。

図10は附属病院のシステムを開発したときの学生の行動から推測したものである。このシス



図9 学生主導で企画運営されるプログラミング教室の様子

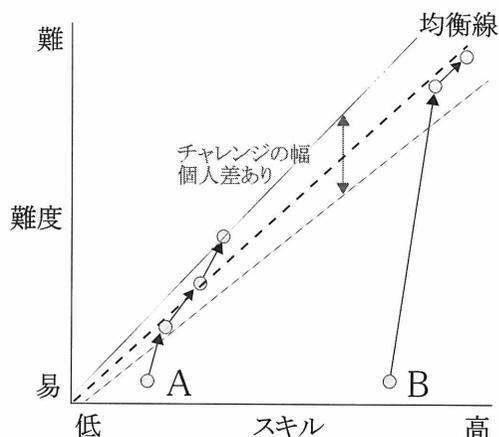


図10 スキルとプログラム難度の関係

テムの開発においては、当初の仕様が変更され、追加機能が急増したこと、その追加に伴う基本設計の変更など、短期間に負荷が増加した。このため、学生にとっては当初想定した能力以上の負荷が短期間にかかったため、開発に対するモチベーションが急激に低下するという事態に陥った。これまでの学生の反応から、スキルとプログラムの難度には図 10 のグラフのような関係があり、スキルにちょうど均衡するような難度のプログラムを担当するときに達成感やモチベーションが上がるように見受けられた。これは、Csikszentmihalyi [10] が提唱するフロー状態に相当するものと考えられ、スキルレベルを大きく超えた難度の高いプログラムに挑戦することは、モチベーションの低下につながると思われる。

この開発事例では、グラフ上に示した学生 A が上述のようにこの状態を経験し、開発から離脱することとなった。そのまま学生が卒業したため、開発は筆者によって継続されたが、2008 年末にスキルを高めていた学生 B を稼働中の当該システムの修正業務に投入したところ、問題なく業務をこなすことができた。この業務にかかわるようになり、学生 B のモチベーションはさらに上がり、多くの業務や行事に積極的に参加している。この事例から、学生 A および B のスキルと難度の関係はグラフに描かれたような軌跡をたどったと考えられる。

これらのことから、学生による開発においては、教員が常に学生のスキルを見ながら、適切なレベルの開発に参加させモチベーションを高め、さらにスキルを高めるという連鎖の中で指導することに留意しなければならないことがわかった。

まとめ

経営学部でのプログラミングの教育にはいくつかのハンディキャップがあり、それを克服しなければ、経営学という社会科学の学問領域で、効果的な教育を行うことは難しい。ハンディキャップとしては、

- ・システム開発が専門の学部ではない
- ・したがって隣接分野の科目が少なく、経営学とは無関係な科目と判断されてしまいがちで

ある

- ・設備等によりすべての学生が受講できるわけではない
- ・適性に左右される分野であるため、過度に授業内容のレベルを上げることができない

ということを挙げることができよう。これらを乗り越えなければ、プログラミングの授業で経営学部としての教育効果を出すことは難しい。また学生参加のシステム開発も不可能である。

結局、この 4 年間で得た成果として、経営学部の学生に対して実践的なプログラミングを教育するためには次のようなサイクルが機能していなければならなかった。

- ・経営学部専門科目との関連をより明確にする
- ・授業そのものの工夫として、受講者のモチベーションをどのように高めるのか、いかに不適格者を出さないかを考える
- ・課外活動と連携することで、より専門性を深め、その成果を授業に還元する
- ・学習内容と現実の開発の関係を理解させ、モチベーションを高めるとともに、キャリア教育の一環とする

これらを一つのサイクルの中に入れて運営していくことが、経営学部の中でプログラミングの教育とその実践である学生によるシステム開発の両立を実現させるものと考えている。

システム開発マネジメントを意識したプログラミングの授業は、経営学部のその他の専門科目との繋がりや、現実のシステム開発を通じて得られる社会や職業との繋がりを意識したものでなければならない。その実践には、いわゆる落ちこぼれを作らないように受講生に興味を持たせながらも適性のある学生には課外活動を通じた現実の開発や社会活動への機会を与え、さらに、それら課外活動の成果を授業に還元し、授業の質を向上させることに繋げることが重要である。

脚注

(注 1) クリエイティブ指数の計算について

クリエイティブ指数は、独創的なプログラムには、他のユーザが使わないような単語がたくさん含まれているという仮説に基づいて、各学生について次のような方法で算出される値である。

全ユーザのプログラムファイルから単語（日本語、

英語、プログラム中の要素)を取り出し取り出された各単語 w について、その単語を使用したユーザの相異なる数 $df(w)$ を求める。

さらに受講者、教員、TAの人数を組み入れ $idf(w)$ を計算する

あるファイルに出現した相異なる単語について、 idf の平均を出す。これをそのファイルのオリジナリティとする。ユーザの作った各ファイルのオリジナリティの「平均を学生のクリエイティブ指数とする。また自習率はログに記録された数値を元に計算している。

(注2) 人数制限のある科目の履修登録については、事前に抽選を行っており、その希望をwebページから学生に入力させ、データを記録するシステム。

参考文献

- [1] 浅井淳 岩田晃 「プログラミング学習入門期に必要な力について」日本教育心理学会総会発表論集第49巻 2007年 pp.69
- [2] 神村栄一他 「大学生のコンピュータ操作およびBASICプログラミング能力獲得に及ぼす要因の検討Ⅱ:性格特性, コンピュータ不安とパフォーマンスの関連」日本教育心理学会総会発表論文集 1993年 pp.531
- [3] 下和也 「学生による経営学部内Webシステム構築について」愛知学院大学経営管理研究所紀要 第15号 2008年 pp.9~18
- [4] 木下和也 「プログラミング・スキル向上を目的とした学生参加によるWEBシステム開発について」愛知学院大学経営学研究第17巻2号 2008年 pp.35~44
- [5] 木下和也 「大学附属病院の消耗品および歯科材料発注・納品・在庫管理システムの構築に関して~各セクションでの発注依頼及び納品処理の構造~」愛知学院大学経営学研究第17巻3号 2008年 pp.65~78
- [6] 兼宗進 長慎也 文科系大学におけるサーバサイドプログラミング授業の試み 情報処理学科研究報告 141~147 2006
- [7] 志村信幸 他 「知的生産における理解度分析手法論 - 情報システム・エンジニアの概念形成教育における受講者評価に関する研究 -」情報処理学科研究報告 61~68 2006
- [8] 鈴木直義他 「ソフトウェア開発教育と地域情報化」情報処理学科研究報告 9~24 2006
- [9] 八木達矢 「変化対応を考慮したシステム構築の考察」UNISYS TECHNOLOGY 89 56~68 2006
- [10] Mihaly Csikszentmihalyi (大森弘訳) 「フロー体験とグッドビジネス—仕事と生きがい」世界思想社 2008年
- [11] 文部科学省 「高等学校学習指導要領解説 情報編」開隆堂出版 2008年