研究論文

鳴門教育大学情報教育ジャーナル No.6 pp.37-43 2009

失敗ソースコードを用いたプログラミング授業環境に関する研究 -情報系を専攻としない学生による参画型授業モデルの考察-

皆月昭則* 林 秀彦**

プログラミングの授業は，大学で情報を専攻としない非情報系の学生を対象とした場合，数少ない実習的な科目の１つであり，いわばモノづくりを行う貴重な機会を提供している。しかし，単にサンプルソースコードの入力に終始する授業構成では，創造的なモノづくりを希求する学生の期待を喪失させることになる。このような希求に応じるため，本稿では，非情報系特有の分野に対応した実践的テーマを創出させてシステムを作り込んでいく授業構成を提案し，実践成果を関連研究と比較考察して述べる。システムの作り込み支援では，過年度の学生の情報や知識リソースが活用可能なしくみを構築し，作り込みを促進させる組織形成手法で作りこみグループを組織化して学生間相互評価を実施した。これらの提案方法を実現した３年間の授業実践では，学生が主体的に授業に取組み，授業から派生した新たなエクステンションが創出されており，プログラミング授業の新たな可能性を見出したことを関連研究との比較からも述べる。

［キーワード： JAVA，組織論，技能伝承，SECI，ナレッジマネジメント］

1. はじめに

プログラミングの授業は，大学で情報を専攻としていない非情報系の学生を対象にした場合，数少ない実習的な科目の１つであり，いわばモノづくり実体験の場である。しかし，単にサンプルソースコードの入力に終始する授業構成では，実用的な技能習得を希求する学生の期待を喪失させることになる。このような状況を改善するため，本稿では，非情報系学生専攻分野[1]を含めた実践的課題を併用した環境でシステムを作り込むための支援システムを構築提案し，授業実践による成果を述べる。システムの作り込み支援では，過年度の学生のミニ講義や失敗事例の活用が可能になるしくみを構築し，システム開発を促進させる組織を形成して実践した。これらの提案方法を実現した３年間の授業実践では，学生が主体的に授業に取組み，授業から派生した課外活動の創設及び地域貢献などもみられ，プログラミング授業の新たな可能性を導出したと考えられる。

2. 作り込み支援による授業実践の概要

本研究では，総じて最終課題のプログラミングによるシステムを作り込む際の学生による各組織が，授業の座席配置や課題に関連した学外における調査等の諸活動など取組みの全面に活かされている。初回の授業では学生の合意形成によってグループを作り組織化する。グループは，単なる授業課題を解決するための学内勉強会のための参集意識だけでなく，システムニーズ調査などの学外での諸活動を実施しながら製作構想を推進する。このように推進した結果から製作された成果物と組織化の関係については[2]に既に述べている。以下では，グループ形成後の課題の推進や，学生相互による評価及び考察に至るまでの一連の授業実践について，そのプロセスの概要を述べる。

システム構築を進める過程では，図１に示すようにグループというよりも高い参画意識と目標を持つ組織としての発展をねらいにしている。よって，授業環境として，さまざまな仕組みを準備する必要がある。例えば，教室の座席配置においても，初回を除き，活発なコミュニケーションと作り込みを実施するための組織のメンバーどうしが近くに意識できるようにした。また，本授業において作り込みを実施した組織自体への評価としては，教師からの評価に加えて最終課題の発表会の評価を追加する。これによって学生の意識は，作り込みプロセスの諸活動の一部始終からシステムの完成にいたる評価点を単位取得の前提に据えるため，学生は所属した組織内で関係を綿密にする必要が生じてくる。

* 釧路公立大学 経済学部
** 鳴門教育大学 大学院 生活・健康系教育部

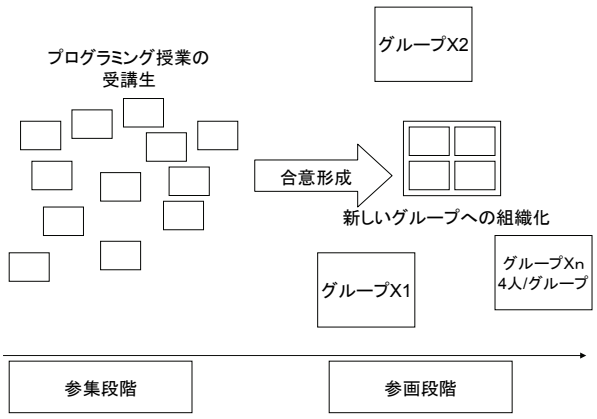

図１　高い参画意識を生み出すための組織化

組織化の中で，学生どうしの相互評価では，①完成発表したシステムが他の組織のメンバーによって他のシステムと比較評価される(評価①とする。)。評価①では，個人だけでなく組織としての作り込みパフォーマンスをどのように向上させるのかを意識させている。よって，外部の評価の意識に加えて，②作り込み組織内の個人について同僚メンバーからの評価も受けることを意識した評価（評価②とする。）の仕組みを用意した。すなわち，評価①と評価②を合わせて，最終成果として外部（他者）に提示しなければならないシステム完成タスクと各自が組織内で参与や参画している度合いを高めて内部メンバーに示さなければならないような仕組みを構築し，内外の相互評価を意識させながら協働的なパフォーマンスを高く維持するような仕組みに沿って授業を実践した。

作り込み組織では，社会調査で得られた情報を議論して知識を抽出したりするための本稿が提案するSECI改モデル[2][3]の概念や作り込みにおける構想を支援するためのKJ法などの創造的技法[4]を積極的に使用しており，図２が示すようにシステムによって何を実現するのかという概念を定めてからシステム実装させる一連のモノづくりを実体験させた。

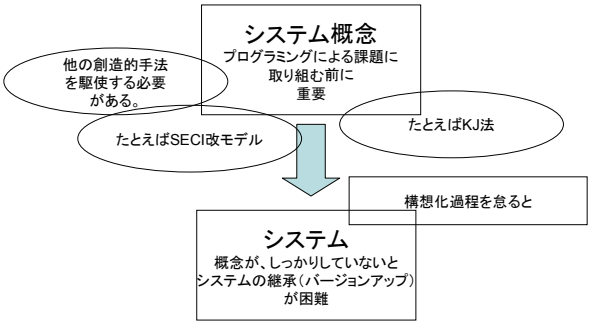

図２　システム概念の重要性
（プログラミングする前に）

システムの概念を明確に据える課題は，理工系のモノづくりの分野に共通することであり，概念をあいまいにしておいた場合のシステム実装の失敗事例を示すことはプログラミング教育でも必要であり，特に失敗ソースコードの閲覧が構想化で有用であった。作り込むシステムの概念を組織的に構想する際の失敗ソースコードの閲覧効果では，システムの発展に寄与しなかったソースコードのコーディングの問題点などを考察しており，少し手を加えることで成功ソースコードになった成果などがシステムの作りこみで用いられていた。また，システムの概念導出の重要性では，失敗ソースコードなどと比較することで，拡張（バージョンアップ）できない，あるいは重大な欠陥を継承する可能性があるなど，永遠に更新プログラム対策に依存しなければならないなどと学生達は，身近なところで自分たちが使用している実在のシステム事例ともに議論し考察していた。

3.　関連研究

3.1 非情報系学生向けのプログラミング能力格差の補間的な授業環境づくり

関連研究[5]では，プログラミング能力の向上を優先して課題を出題し，学生の能力差に着目している。これは，能力差を小さくしようとする試みでもあるが，前提になるのは課題提出を継続させる試みでもあると考えられる。本授業では，学生の継続性の牽引要素として，作り込み組織の学内外の諸活動を通じた互いのエンカレッジ環境に委ねており，プログラミング能力の格差が大きく生じて（生じている前提）いても，授業におけるサンプル・プログラムなどの個別課題の達成に偏重することなく，最終成果としての完成システム発表時に実施する学生どうしの相互評価までの活動まで継続目標をおいている。すなわち，作り込みの諸活動によって組織内の参与・参画意識を高めさせて，授業から学内外の諸活動まで取り組む継続性を担保している。したがって，学生は，作り込み組織の協働的活動に継続的に貢献していれば，高い評価が得られると確信してくるようになると考えられる。本稿の授業実践において学生の個々の能力に及ぶ正確な検証はしていないが，このような授業環境に学生達のさまざまな能力差の補間と発展および取り組む楽しさを期待感が生じてくると考えられる。また，関連研究[5]ではティーチングアシスタントと教師による課題演習の指導補助を実施しているが，本授業では，関連研究[5]より，さらに学生どうしの距離を近づけ密にする作りこみ組織を作らせており，参集意識の向上で学内外のシステムニーズ調査などの課題にも直接的に向き合う参与意識をもたせるための作り込み組織に，さまざまな課題が遂行されることを委ねている。関連研究[5]でも，プログラミングの能力差が，どの程度まで向上したのかを明らかになっていないが，継続性が向上（11.3ポイント）していると

報告されている。また，本授業では，作り込み組織内での得意分野（例えば，街のことに詳しく外では活動的な人）における参与・参画する意識を重要視しており，与えた授業環境においては，受講生の 95%が継続して出席しており単位を取得した。関連研究[5]のような方法では，プログラミングの能力の向上をねらいにする場合，かなりの長い時間が必要であると考えられる。そして，関連研究[5]のように推測した学生の能力に応じた個々の課題を与えることはプログラミング能力の向上にも直結するが，低い課題に取り組む学生のやる気に対するモチベーションの低下が懸念される。特に懸念する例としては，毎回，他の学生よりも簡単な課題に取り組む際の劣等感が現実化し授業に参加する継続性が失われてしまう可能性が考えられる。また，関連研究[5]による例では，「解けたときの大きな達成感が味わえるように，推測された得点が最も低い演習課題を出題する」とあるが，低い課題が継続する場合，課題に取り組む際の高揚感が次第に失われてしまう可能性があると考えられるため，本研究では，授業課題に取り組む参集環境以外にも作り込み組織内の勉強会や議論で解決することに期待し，プログラミング能力が低い学生への参与・参画の機会をできるだけ多くアドホックに作れるようなグループ決定手法を実践した。また，作りこみを強力に推進させる例としては，作り込み組織に先輩を呼んでヒントを聞いて解決するような方法や，組織的な諸活動を通じて同時同所的な環境において学習と問題解決ができることを会得させた。近年，初等中等教育でも，算数などの授業が能力別のクラスで少人数指導するメリットやデメリットが議論されているが，図３が示すように高等教育の情報系の授業において，特にプログラミング授業においては，能力差に分けた指導に対して同様の懸念や問題点もあるため，本研究では能力差を補間するためのグループを作る参集手法から工夫を施し，所属決定グループへの参与・参画を強く意識させた作り込み組織を重視している。

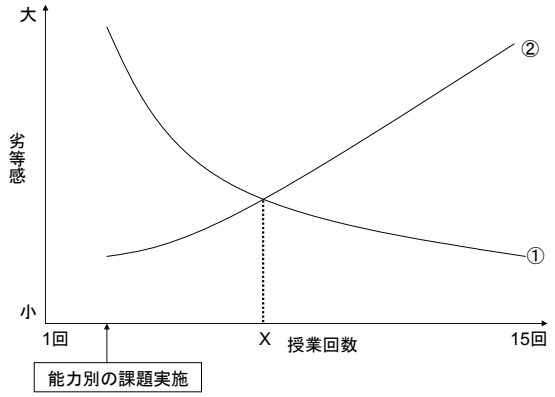

図３　能力別にした課題提出の懸念

図３では，能力別に課題を実施した場合の懸念として，①と②の曲線種別による劣等感差が生じると考えられる。すなわち，能力別に分けた課題を継続していくと，取り組み意欲にも影響すると考えられるため，授業回数の中盤(X)以降は，個別課題ではなく組織化によるメンバーの創意を結集するような授業環境が必要であると考えられる。

関連研究[5]における大きな指摘としては，授業の継続性が低下するというよりも，プログラミングやモノづくりに関する概念や構想化などの諸活動にも将来的に関心意欲が失われてしまう懸念があると考えられる。すなわち，算数における授業の児童のように，過去の演習課題履歴で点数が低い場合には，その理解状況が次の課題選出に反映してしまう例もあり，場合によっては，授業の最終週まで簡単な問題に取り組まざるをえない阻害された気持ちになるような学生もいると考えられる。このような懸念における仮説としては，プログラミングを通じた作り込み組織による製作や構想化の楽しさなどの参与・参画意識が芽生える可能性は低く，劣等感は将来に長きにわたり継続する可能性が考えられる。関連研究[5]の著者らが学習意欲の継続を今後の発展としているところは，前週の演習課題の提出が滞るなどにより，理解できなかった場合の挫折した学生を救済する方法などの根本的な現実の問題として指摘している。本研究の授業環境の救済例では，前週に何らかの要件で授業に欠席して理解できなかった場合でも，作り込み組織によるフォローアップが可能になっている。作り込み組織の学生にフォローアップ内容をヒアリングすると，次のように聞かれる。「A 君は，プログラミングの苦手意識もあり勉強会には継続的に出席しているが，社会調査などのフィールドワークの方が向いていると考えられるため，大いにチャンスをあげたいと思う」などと，得意な分野で参与・参画するためのタスクバランスとして作り込み組織内で考慮調整した親睦的進捗が見られる。また，本研究の授業環境を用いた場合，教師は，個々の学生の能力差をあまり意識することなく，作り込み組織のフォローアップを期待した高度な内容の授業も可能であると考えられる。

3.2 学生参画による授業実践

関連研究[6]では，少人数グループによる学習という点では，取組みの類似性があるが，ピア・レビューの単位が２名１組であるため，本授業のグループの規模と異なる。すなわち，関連研究[6]は，２名それぞれを教師が決定していく方法である。本授業では，グループの構成メンバーの決定過程における教師の役割は，グループのメンバーを決めることができないようになっている。よって，教師はグループの編成数を決めることは可能であるが，メンバーはグループ所属予定の１名単位の選出からでも，学生互いの合意形成で決めていく必要性が生じる。本授業では，このようなグループの決定手法において関

連研究[7][8]として提案しており，この提案したグループ決定法は，単なるグループへの参集意識ではなく，参与から参画[9]に向かうための諸活動を通じて強い意識をもたせる重要なプロセスの開始点である。関連研究[6]では，「お互いに教えあい不足した知識を補強し，獲得する」と述べられているが，教師が決定したグループの単なる参集意識だけでは学生の能動的な意識への移行が円滑でないため，ピア・レビューにおける参与・参画の意識は必ずしも高くないと考えられる。一方，本手法では，グループを決定するプロセスで簡単な動機付けを与えているため，当初のグループの参集メンバーに参与・参画意識を与えることが授業実践によって検証できた。また，関連研究[6]では，数回の学習単元の知識を活用したプログラムを制作する演習課題を提示しており，これらはサンプル・プログラムの勉強会を中心とした学生どうしの相互効果に期待していると考えられる。本研究では，関連研究[6]の演習課題の取組みにおける学生どうしの相互効果に加えて図４が示すような参与・参画意識に期待している。

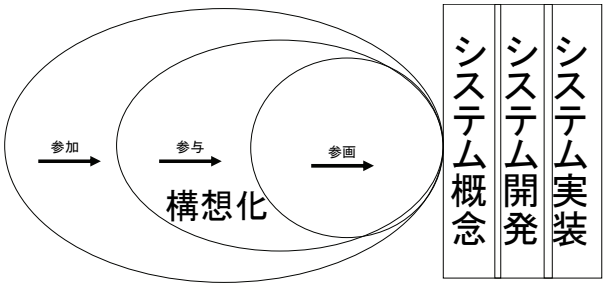

図４　参加・参与・参画

すなわち，システム開発のための学内外の諸活動から得られた発想や創造的手法を駆使することによって，システムの概念化や構想化過程を重んじた最終課題のシステム構築の評価までを重視している。学生どうしの相互効果の例としては，仮に演習課題によるプログラミングが苦手であっても，組織が実施する学内外の諸活動に十分な貢献が認められることによって，メンバー個人の評価が得られることが可能であり，さらに，そこで得られた情報をプログラミングに活用できる知識とみなしてシステムに盛り込むことを意識させる指導を実施している。関連研究[6]のグループ学習手順では，レビューを中心にしているが，学習素材においてサンプル・プログラムおよび演習課題に終始しており，本格的なシステムの作り込みの実践までには到達していない。本授業では，学習領域を拡大しており，その支援のためにSECI改モデルを意識させるなど，毎授業終了時に学内外の諸活動報告やシステム構築の進捗報告会をグループ毎に実施している。関連研究[6]では，設計論理をレビューで解説しているが，本授業では，設計論理を作りこみ過程のReadmeファイルの作成時点で組織内の相互レビューを実施し，その後に実装テストしてソースコードを類別してサーバで公開し，さらには，本格的な作り込み前に設計概念として中間発表会で評価している。関連研究[6]の評価は，ピア・レビューで得られたフローチャートの評価を毎回実施する定点評価であるが，肝心のグループ学習の評価としては，本授業が毎週複数回実施していることに対して関連研究[6]では5.74週に１回と少なく，本手法の評価回数との相違もみられる。

4.　考察

本稿で述べた授業実践における展開方法は，作りこみ過程において前工程と後工程という分類をして実施しているが，過去の一貫したプログラミング教育の方法論を軽視するのではない。プログラミング言語によるモノづくりに限らず，どのように作り込むのかという本質的な部分は，学生の自己学習力で構築されていくものと考える。過去の授業実践では，冗長的に基本に着目させるようなサンプルソースコードの入力で始まりそして終わる授業展開がなされていたが，昨今の学生の期待を保持することは難しいため，本稿の提案方法では，そうした部分を補っている。すなわち，システム開発の必要性を大局的に意識させる切り口によって，例えば，小規模で短期的テーマを否定するわけではないが，社会科学系学生の興味が伴うような学外の調査などを含めた大規模テーマを設定した授業環境を構築して実践することで，学生の関心・意欲を持続させながらシステムの作り込みが実現できた。よって，本授業実践では，サンプル・プログラムや演習課題による従来のプログラミング学習を延伸した環境によって，学内外の諸活動を通じて幅広いフィールドから課題を自らのグループで発見抽出し，システムを開発する。課題を整理してシステム概念を作らせるために，サンプルコードそのものではなく，システム化が必要不可欠である理由やシステム実装に向き合うという動的な意識付けが重要であると考えられる。すなわち，使用した言語はJavaであるが，プログラミング言語のコード入力という小規模・短期的な実装だけでなく，学内外の諸活動で課題を発見してシステム化の必要性の根拠を明確にさせて，その後はシステム概念を正確に構想化して社会の諸課題を解決する姿勢に期待している[10][11][12]。その期待を組織的に，かつ，システマテックに実現するため，過去の成功・失敗プログラム例や学内外の諸活動に基づいてKJ法やSECI改モデルを使用した方法論を駆使させている。よって，授業では，新たなシステムの構想力を会得させるプログラミング授業提案の型として従来の授業型と異なることを強調している。

本稿の授業テーマのように学内外の諸活動を含める場

合，幅広い範囲の情報を調査してくることから，システム概念を作るための知識変換など合意形成による時間も要するが，学生は積極的に時間を確保し，主体的に行動することが観察できた。これは，授業における作り込み組織の形成が有効に機能したことを示唆している。また，複数年度学生にわたるReadmeファイルやミニ講義は，技能伝承であり，単年度だけでは実現できないような作り込み成果を実現できた。これらの成果を関連研究と比較すると，プログラミングに関する授業において，過去の学習者の履歴を活用することで学習意欲の低下を防ぐ研究[6]や，学生によるピア・レビューによって学習効果を高める研究[5]が報告されており，本提案においても，これらの報告を支持する結果が得られたと考察できる。

本稿では，使用している授業での作り込み組織形成の方法を詳述しないが，ソフトウェアを作り込む企業では，プログラミング能力だけでなく，作り込み過程を発展的に推し進めるような組織が必要であり，そのような組織化について企業の開発マネージャは模索するところであろう。企業の開発現場では，各ソフトウェアの作り込み工程に適切なメンバーが組織化されており，各作業はシステム概念のもと適切にシェアされている。近年では，エキストリーム・プログラミングなど，開発を短期で終わらせるような組織構成もIT企業で導入されている。大学のプログラミング教育では，そのような企業事例に等しくあわせる必要性はないが，失敗してもいいからという意識と，構想的な作り込み方法と組織で，システム概念の創出やモノづくりを経験させることは重要である。

また，本稿で述べた授業実践は，これからの大学教育の在り方の1つとして示唆するものである。本稿の実践的取り組みでは，授業終了後あるいは単位取得後も継続的に授業での成果を社会あるいは地域のために活用するという学生の思いが創出される。このような創出は異例と考えられるかも知れないが，今後，プログラミング教育あるいは情報教育のめざす一目標として据えることも可能である。つまり，「社会や地域で貢献するきっかけづくり」としての授業の在り方に1つの指針を与えることができた。

プログラミングの授業では，基本文法となる個人の知はサンプル・プログラムのコード学習で高めることができるが，システムの概念化やプログラミングの複雑な構造化には実用的・実践的なプログラミングのコツあるいはグッドプラクティスを学ぶことが必要であり，集団的なグループの知を結集する学習環境において，問題解決が可能になる。本実践における授業観察によると，学生どうしの議論の長さが，学生個人のプログラミングスキルの向上に寄与しており，例えば初回の授業では小規模のサンプルコードを真似る程度にとどまっていた学生であっても，授業の後半では学生どうしの議論を通して，より大規模なシステムを設計している。これは，単にサンプルコードを真似るだけでは達成できないスキルであり，サンプルコードを理解して新しいオリジナリティのあるシステム設計に応用できるスキルをも学生は身につけた結果であると考える。つまり，既存のサンプルコードを理解する基礎スキルに加えて，学生の掲げる新たなテーマを設定してシステムを構築するに至る基礎スキルを身につける授業実践の結果であることが考察できる。なお，このような観点は，授業観察からのみではなく，学生の視点（学生が回答する授業アンケート）からも支持される結果であった。

例えば，FDの一環の授業アンケート感想によって，学生の習得スキルを抽出しており，次のような感想があった。

・サンプル・プログラムは，変数の宣言から制御文・配列など理解できましたが，自分たちで設定したテーマのシステム化が，うまく，できませんでした。こうしたいと思うのですが，エラーが発生してうまくできませんでした。また，私たちが作ろうとしているシステムが，自分たちのスキルをこえていると思いました。(Aさん，３年，2003年受講生)

・サンプルコードは，すべて理解した。グループで決定して取り組むテーマは壮大であるが，次年度の学生への継続性が期待できるため，今年度までの目標を立て，できることはすべてしておくことにした。特に，社会調査は予備的調査に実施して問題点や必要な情報発信のシステムの要望を抽出して，Webサイト内にJavaアプレットを組み込む概念設計して，ASPで基本テストをした。アプレットの動的な工夫が，ユーザインターフェースとのユーザビリティに障害的になっていることが明らかになったが次年度学生に任せる。(Bさん，３年，2004年受講生)

・前年度の受講生のシステムを一覧し，グループで検討した結果，使えるモノと使えないモノがあることがわかった。特に，エラーによる失敗プログラムを閲覧していくと，笑える失敗箇所があり，僕らも，このように次年度学生に笑われないような作りこみをしていきたいと考えた。失敗プログラムが，誰によるものかわからないが，さらし者にならないようにするために，プログラミングに親切にコメントを付加して頑張った証をシステムにして残す。(Cさん，３年，2005年受講生)

これらの結果は，いわゆる「実践から学び創出する」ことによってスキルを習得していく過程の断片を学生の

視点で表している。本実践では，一斉授業に対する評価の場合とは異なり，学生が一律に習得していく学習の過程を示すことは容易ではないため,学生がどのような過程を経てスキルを習得していったのかの詳細を分析することは今後の課題であるが，これらのアンケート事例を代替として考察すると，学生の視点からも先の授業観察の結果を支持する結果が得られていることが示唆される。これは，モノ作りの楽しさに学生が気づいて自発的に学習を進めた結果であると考察している。

また，表1は，授業専用サーバ内のソースファイル数を示している。「①失敗ソース」は，受講年度学生の生成したコードのうち学生によるReadmeファイルとともにある失敗ソースコードのファイル数である。「②閲覧ソース」は，サーバ内に蓄積されている閲覧が可能なソースコード全てのファイル数である。「③失敗/閲覧」は，年度ごとに②を①で割った値であり，閲覧ソースに対する失敗ソースの割合であるため,いわば失敗率を意味している。③の結果から年度毎に失敗率が減少する傾向が示された。これは本提案手法が有用であることを示唆する一つの結果である。ただし，本指標は一つの目安であるが，必ずしも失敗ソースの数が年度毎に減った結果ではない。例えば，2005年度は2003年度と2004年度に比べて失敗ソースの数は減少しているが，2004年度は2003年度に比べて失敗ソースの数は増えている。これは，テーマ設定の違いによるアルゴリズムの難易度の違いによる影響やグループの諸活動の質的な相違による閲覧ソースファイルの活用度合いの違い等さまざまな要因が推察される。

表１　サーバ内のソースファイル数の推移

年度	２００３	２００４	２００５
①失敗ソース	３０	５７	１９
②閲覧ソース	５５	１２８	２２３
③失敗/閲覧	０．５５	０．４５	０.０９

5. まとめ

大学で情報を専攻していない非情報系の学生を対象としたプログラミング授業について，社会調査などの大規模テーマを導入した実践的課題の中でシステムを作り込む新たな視点を導入した。そして，複数年度の学生によるプロジェクト継承およびシステム・プログラミング技能伝承と学習組織形成による授業環境の構築を提案し,3年間の授業実践により得られた成果に基づき考察した。直接的な成果では，プログラミング専用サーバの活用と過年度の学生のミニ講義による複数年度の学生の技能伝承などを通して，非情報系学生に対してプログラミングに興味関心が向けられて「できる」から「(実際に) つくる」ための構想力の育成を実現した。3 年間にわたる授業実践では，学生が主体的に授業や課題に取組む姿勢がみられ，授業から派生した新たな課外活動につながる成果を示した。

近年，日本のソフトウェア開発は欧米だけでなくアジアにアウトソーシングする割合が高くなっていることは，モノづくりを推進する人材が日本で育っていないことを意味する。しかし実際にJavaやC#言語の仕様を駆使して提案手法を実践すれば，非情報系の学生でも構想力を養い高度なモノづくりができるようになることが期待できる。今後は，本稿で提案した方法を高等学校で適用できるよう検討を進める予定である。

参 考 文 献

[1] Forte, A. and Guzdial, M.: Motivation and non-majors in computer science: identifying discrete audiences for introductory courses. IEEE Transactions on Education, Vol.48, No.2, pp.248-253 (2005).

[2] 皆月昭則：失敗ソースコード学習と学生主体組織によるプログラミング授業に関する研究,鳴門教育大学情報教育ジャーナル, Vol,6, pp.47-55 (2008).

[3] 野中郁次郎,竹内弘高,梅本勝博:知識創造企業, 東洋経済新聞社, (1996).

[4] 高橋誠：新編創造力事典, 日科技連出版社(2002).

[5] 田口浩,糸賀裕弥, 毛利公一, 山本哲男, 島川博光：個々の学習者の理解状況と学習意欲に合わせたプログラミング教育支援,情報処理学会論文誌,Vol.48, No.2, pp.958-968 (2007).

[6] 生田目康子: ピア・レビューを伴うグループ学習の評価 - 一斉型プログラミング授業への適用 -, 情報処理学会論文誌, Vol.45, No.9, pp.2226-2235 (2004).

[7] 村上雅俊, 柿崎達哉, 松元初美, 皆月昭則：合意形成型グループ決定法を利用した最適協調性の定量評価, 情報処理学会第 70 回全国大会, 第 4 分冊, pp.921-922 (2008).

[8] 松元初美, 千代谷典広, 佐藤大希, 森谷智史, 皆月

昭則：抽選型と合意形成型グループ決定法の考察，情報処理学会研究報告，2008-CE-095(5)，Vol.2008，No.64，pp.29-34（2008）．

[9] 林義樹：参画教育と参画理論　ー人間らしい「学び」と「くらし」の探究，学文社（2002）．

[10] 結城浩：Java 言語プログラミングレッスン上・下，ソフトバンクパブリッシング(2000)．

[11] 鈴木正人：ソフトウェア工学，サイエンス社(2003)．

[12] 山田健志：オブジェクト指向ワークブック，翔泳社(2003)．