

# 学習者同士の相互レビューを通じたプログラム設計教育

松澤 芳昭<sup>†</sup> 杉浦 学<sup>†</sup> 大岩 元<sup>‡</sup>

<sup>†</sup>慶應義塾大学 政策メディア研究科 <sup>‡</sup>慶應義塾大学 環境情報学部

macchan@crew.sfc.keio.ac.jp

## 概要

プログラム設計の基礎として、目的と手段の構造が整理された、第三者にも分かりやすい HCP チャートを作成させる教育を行っている。本稿では、この教育をなるべく指導者に頼らずに、学習者同士の相互レビューによって発見的におこなう教育方法の提案と考察をする。提案する方法は、効果的なレビューができるように、1) 第三者が作成した設計を、実装することを前提としてレビューさせる 2) レビューの視点を指導する、などの工夫をしている。教育実践した結果、a) 設計の重要性や、設計文書によるコミュニケーションの難しさを効果的に理解できる b) レビューの視点を教育することにより、学習者同士のレビューのみでもかなりの学習効果が得られることが確認できた。

## 1. はじめに

我々は、竹田[1]の考えに基づき、プログラマから出発して、システム設計ができるようになることを目的としたプログラミング教育を行っている。この教育では、プログラミング言語の知識を獲得するだけでなく、他人（第三者）に分かりやすく、保守の容易なプログラムの設計を実践できる能力を育成することを教育目標としている。

また、我々は、竹田カリキュラムにオブジェクト指向の概念を取り入れたカリキュラムを開発、実践している[2]。この経験により、竹田カリキュラムの段階で、コメントや HCP チャートを用いた「第三者に意図が伝わりやすいように、目的と手段の構造を整理して表現する能力」を獲得していることが、オブジェクト指向設計の学習の前提能力として必須となることが分かってきた。

この教育を効果的に行う方法の一つとして、竹田は学習者の成果物に対してレビューをする方法を挙げている。駒谷[3]は、学習者同士のレビューを取り入れたカリキュラムが学習効果を向上させることを示した。

しかし、駒谷の成果はコーディング規約や文法誤りなど下流工程での学習効果にとどまっている。筆者の教育経験から上流工程の

「構造が整理されているか」という視点でのレビューに関しては、指導者が担う役割が大きいと考えられる。竹田も、このタイプの教育において、指導者の開発経験や教授力が教育効果に大きな影響を及ぼすと述べている。

本稿では、この教育をなるべく指導者に頼らずに、学習者同士の相互レビューによって発見的におこなう教育方法の提案と考察を行う。

提案する方法の概要は、学習者に 2 人ずつのペアを組ませてそれぞれに設計文書を作成させる。そしてそれを交換して相手の作成した設計に従って実装を行わせるというものである。

我々はこの方法を、IT 実務者、新入社員、大学生と 3 種類の異なる教育対象に対して実践した。本稿では、これらの実践経験から得られた知見を報告する。

## 2. 教育の目的

本教育の目的は、「第三者に意図が伝わりやすいように、目的と手段の構造を整理して表現する能力」を育成することである。

この目的を達成することは、設計の入門教育として非常に重要である。システム設計では上位の目的が下位の手段選択の理由になる。上位の目的を明確にし、共有することができなければ、豊富な設計の知識（手段）を持っていてもそれを適切に選択できない。

目的と手段の概念自体は誰もが知っている知識である。目的と手段を整理して人にももの

を伝えることは、実はそう簡単ではない。

これを実践できる能力をもった人材は、コミュニケーションを円滑に進め、設計だけでなく、学習や仕事をスムーズに遂行できると我々は考えている。

### 3. 教育の方法

#### 3.1. HCP チャートを用いる

本教育で利用する設計言語として、HCP チャート[4][5]を用いた。HCP チャートは目的-手段の構造を表記できる。フローチャートのように、実装の手段のみを記述する表記法は本教育の目的にそぐわない。

なお、本教育は、オブジェクト指向設計の教育ではなく、構造化設計の教育である。オブジェクト指向設計と比較して構造化設計のほうが易しいので、設計教育入門として適していると考えためである。

#### 3.2. 実装することを前提に第三者の設計レビューをする

本教育法の特徴は、学習者同士のレビューを通して発見的な教育を促進することである。しかし、レビュアーである学習者のモチベーションがあがらない場合は、消極的なレビューになり、教育効果が得られないと思われる。

そこで我々は、「第三者に実装されることを前提に設計文書を書き、第三者の書いた設計文書を、実装することを前提にレビューする」という演習を行った。その演習の手順は以下の通りである。

1. 学習者同士で2人ずつのペアを作る
2. その2人に指導者が作成した仕様書が与えられ、互いに設計を行う
3. 互いに作った設計を交換して、実装することを前提にレビューをしあう
4. レビューを受けて自分が作成した設計を修正する
5. 修正された（相手の）設計に従って、実装を行う

この演習では、設計の構造を整理して伝えることができないと、相手に理解されず、実装してもらえない。それゆえ、通常のプログラミング初歩教育では難しい、設計の目的を学習者に明確に伝えることができる。また第三者に見せることが必要となるので、質の向上も期待できる。

この演習では、相手が行った設計に従って

実装を行わなければならない。それゆえ、相手の設計を完全に理解するようにレビューしないと、自分が実装するときに困ることになる。学習者同士の積極的なレビューが期待できる。

教育目的は設計教育であるため、仕様書は指導者が作成して与えている。仕様書の内容は、お互いに第三者の視点でレビューできるように、ペアで異なるものになるようにしている。

#### 3.3. レビューの視点を統一する

学習者同士のレビューをうまく機能させるためには、レビューの視点を統一することが重要である。レビュアーは指導者ではないため、我々の意図とは異なる視点でレビューが行われる可能性がある。

そこで我々は、本教育の学習目標である「目的と手段の構造が明確か」という点に焦点を絞ったレビューの視点を考案した[6]。このレビューの視点は、これまでの教育経験に基づいて、レビュー項目のパターンを分類したもので、レビューの視点となる分類は以下の4つである。

- 日本語不良：  
日本語の不備により記述の意味が不明なもの
- 目的不明確：  
その記述がどのような目的で必要なかが明確でないもの
- 構造不良：  
上位レベルとその下位レベルの記述が「目的-手段」の関係になっていないもの
- 粒度不良：  
あるレベルにおける、記述の大きさが不揃いであるもの

演習を始める前に、このレビューの視点を学習者に指導した。さらに、実装者が実装時に勝手に設計を変更してはならず、必要なら設計者に変更してもらうように指導した。レビュアーが、不明確なチャートを勝手に解釈して実装してしまうことを避けるためである。

## 4. 実践結果

### 4.1. 実践環境

我々が提案する教育方法を、IT実務者、新

入社員、大学生と 3 種類の異なる教育対象に対して実践した。対象者と環境をまとめて表 1 に示す。

	プログラム経験	受講者数 (評価対象)	講義時間 (hour)
IT 実務者	1~15 年	32(18)	15
新入社員	0 年~	16(12)	40
大学生	0.5 年~	66(37)	15

表 1 実践対象

プログラム経験は、学習経験も含めており、IT 実務者以外は実務の経験がなかった。

IT 実務者と大学生は、プログラミングの文法にはほとんど時間をかけず、講義時間の 2/3 ほどを設計の教育に使った。新入社員に対しては、全くの経験がない者が多く、文法の教育も同時に行ったので、設計の議論を行ったのは、講義時間の 1/5 ほどである。

なお、受講前に HCP チャートの知識がある者はいなかった。

#### 4.2. 受講者の感想

受講者の感想は総じて、「レビューを受けて間違いを指摘してもらうことで、理解が深まった」「レビューを受けると、気づかなかったことがたくさん発見される」など、本教育方法に対して肯定的だった。

ただし、演習の初期には「重箱の隅をつつくようなレビューを強要された」「レビューはどこまでが指摘すべき範囲なのかの線引きが分からない」などの、実習の効果に否定的な受講者がいた。

ところがその否定的な受講者も、実習を 1 度か 2 度ほど経験すると、「他人に理解してもらえるようにチャートを作成する作業はとても難しい」「頭ではわかっていることを人に伝えるのは難しい」「どこまで詳細に書けばよいか難しい」「言葉の意味が重複してしまったり、あいまいになってしまうことが多くて大変だ」など、文書によるコミュニケーションの難しさと、設計という作業の難しさに対する意見に変化していき、最終的には先述のような肯定的な意見になる傾向があった。

#### 4.3. 教育後の成果物の評価

教育効果を測定するために、受講後の受講者が作成した HCP チャートの評価を行った。3

段階の基準を用い、チャート全体の質を評価した。

基準を表 2 に示す。

A	構造がよく整理されている
B	構造は比較的整理されている 不明確な記述が散見する
C	構造が整理されていない 日本語が不明確である

表 2 評価の基準

表 2 で示した基準を用いて評価を行った結果を表 3 に示す。

	総数	A	B	C
IT 実務者	18	7(39%)	4(22%)	7(39%)
新入社員	12	3(25%)	4(33%)	5(42%)
大学生	37	8(22%)	6(16%)	23(62%)

表 3 評価結果

#### 4.4. レビュー内容の傾向

受講者がどのようなレビューをしているかを調べるために、受講者の記述したレビューログを分析した(新入社員のログは入手不可能であった)。結果、指摘の多くが 3 種類に分類できることが分かった。以下にその分類を示す。

- 目的不明確指摘：  
目的やその構造が整理されていないことを指摘したもの
- 手段不明確指摘：  
手段が不明確、またはアルゴリズム不良など、手段の不良を指摘したもの
- その他・不明：  
その他  
レビューの記述自体が不明確なもの

目的不明確指摘は、3.3 節で述べたレビューの視点に基づいて指摘している項目である。この分類の指摘ができる受講者は、演習前に指導したレビューの視点を理解し、それを実践に生かしてレビューを行っていると考えられる。

手段不明確指摘については、演習前に指導は行っていない。ただし、受講者にレビューをさせるときに、指導したレビューの視点以外の項目も見つけたら指摘してよい、としたため、学習者が独自に指摘したものと考えられる。

この2つの分類を用いて、IT実務者と大学生の指摘傾向をまとめたものが、表4である。

	IT実務者	大学生
目的不明確指摘率	37.5%	17.4%
手段不明確指摘率	43.8%	71.4%

表4 環境による指摘傾向

## 5. 考察

### 5.1. 設計に対する意識・モチベーションの向上

本教育の演習では、指導者がレビューする場合と異なり、受講者は「教えられている」のではなく、実際問題として自分の意思が隣の受講者に伝わらないという体験をする。その体験は相当印象強いことが受講者の感想とその変化の様子から考察できる。この結果から「伝わる設計を作成すること」の重要性を理解させるために、本教育法は有効であると考察できる。

また現場では、受講者が楽しそうに相互レビューを行っている姿が観察された。特に、今回受講したIT実務者にとって、このような経験はあまりないらしく、活発な議論が行われていた。このことから、レビューに対するモチベーション向上に関しても本教育法は概ね有効であることが考察できる。

### 5.2. 設計文書の質の向上

受講前にHCPチャートの知識を持つものがいなかったことと、HCPチャートの知識のみを教えた段階でのHCPチャートの質が全体的にかなり悪かったことを考慮すると、ある程度の教育効果を確認できたと考えられる。

しかし、実践環境によって教育効果に差がある。特に大学生に対する実践では、意外に効果が上がらなかった。

### 5.3. 実施環境による影響の考察

本節では、実施環境による影響の考察として、何故大学生への実践であまり学習効果がみられなかったのかを考察する。

#### 1) 相互レビューの質

「表5 環境による指摘傾向」より、大学生の指摘は、手段不明確に大きく偏っていることが読み取れる。レビューログを観察しても、レビューの質の差は歴然としている。

これは、実装することを前提とするレビュー環境を設定したことによる負の影響だと考えられる。学習者が「実装しなければならない」というプレッシャーを感じてしまうため、レビューの視点がプログラム実現の手段に偏りがちになってしまうのである。

HCPチャートは、使い方によってはフローチャートのように実装の手段を羅列することもできる。そして、小さなプログラムであれば、目的を共有しなくても、詳細な手段を共有することにより、相手は実装できてしまう。しかし、それでは実装プログラムを交換して、写しているだけに過ぎなくなる。

このことから、レビュー演習を行う前にレビューの視点をしっかり指導しておくことが重要だったといえる。さらには、レビュー演習開始後も、しばらくの間は指導者がレビューの状況、特に手段不明確指摘率を観察しておき、指導を行う必要がある。

#### 2) マッチング

大学生への実践では、「レビューしても相手が設計を修正してくれない」と指導者に泣きつく学生が散見された。

この問題の原因の一つに、大学生は社会人に比べて個々人のモチベーションの差が著しいという原因が考えられる。

レビュー演習において、設計を相手が修正してくれないことは、自分の実装ができず、最悪の場合課題が提出できない可能性もあることを意味する。優秀な学生が不真面目な学生とペアを組んでしまった場合、努力が報われず、優秀な学生のモチベーションまで下げてしまう可能性があるため注意が必要である。

#### 3) コミュニケーション環境

「レビューしても相手が設計を修正してくれない」という問題に対して考えられるもう一つの原因は、コミュニケーション環境である。実装演習では、実装が始まってから設計の不具合が見つかり、設計者に問い合わせることもある。社会人を対象とした教育では、何か不具合があればすぐに聞ける環境での実践だった。しかし、大学での実施では授業時間が限られているため、実装を宿題とした。そのため、設計に不具合が見つかった場合にすぐに設計者に問い合わせることができる環境ではなかった。このことが原因となり、上記のような不満を持つ学生が少なからずいた。

### 5.4. 実装能力の影響

HCPチャートを用いた指導していると、気

になることがある。実装が得意な学習者がなかなか目的の構造をまとめられず、手段を詳細に示した HCP チャートを書く傾向があることである。

レビューの指摘傾向にも、同様の現象が起こることが、表 5 で示した指摘傾向の比較により考察された。

	実装得意	実装不安
目的不明確指摘率	25.7%	69.5%
手段不明確指摘率	59.8%	5.5%

表 6 実装能力による指摘傾向(IT 実務者)

個々のデータを観察によると、必ずしも実装の得意な人が設計を苦手とするわけではなく、実装能力と設計能力自体の相関はあまりないようである。

あまり実装が得意でないグループでは、目的不明確指摘率が高いというよりも、手段不明確を指摘できないといった傾向が強いことが考察された。

### 5.5. マッチングの影響

設計能力、実装能力や指摘率よりも、コミュニケーション能力や相性が良く活発な議論をし、レビューの指摘件数が多いペアが若干学習効果を高めている傾向が観察された。しかし、実装が得意な人同士が手段不明確の指摘を活発に行っている場合もあるので注意が必要である。

## 6. まとめ

本稿では、プログラミング設計教育を受講者同士のレビューを通して発見的に行うための方法として、第三者の書いた設計に対して、実装することを前提として学習者にレビューを行わせる方法を提案した。

この教育を企業の新入社員、IT 実務者、大学生に対して行った結果、以下の 4 点が考察された。

1) どのペアでも、ほぼ間違いなく学習者同士で意図が伝わらないという状況が起こる。これは学習者にとって貴重な体験である。この体験を通して、設計の重要性や、設計文書によるコミュニケーションの難しさといったことを学習者が理解できる。

2) 学習者が「実装しなければならない」というプレッシャーを感じるため、レビューの視点がプログラム実現の手段に偏りがちに

なる。この傾向は、実装が得意な学習者ほど強くなる。指導者は、学習者のレビューの視点が論理構造に向くように、事前に十分な教育をおこない、始めのうちは学習者のレビューの質を観察する必要がある。

3) 演習を行う環境は、学習者同士がいつでも会話できる環境であることが望ましい。演習を宿題にする場合などでは、演習実施時にペアの学習者同士の時間と場所が共有されないため、教育効果が上がらない。

4) ペアリングを決める際には、実装能力や設計能力よりも、モチベーションやコミュニケーションの相性を考慮した方が良いと思われる。学習者同士でより多くの議論ができる環境をつくるように配慮すると良い。

## 7. 今後の課題

レビューの視点をうまく指導すればもう少し効果的な相互レビューを行うことができるという感触がある。レビューの視点自体がまだ未熟なので、学習者により分かりやすくなるよう改良していく必要がある。

レビューの手段は文書と会話の基本であるが、インターネット上で行うツールがあると学習者にとってコミュニケーションの手段が増えるので便利である。そのようなツールにより、指導者も学習者同士のレビューを追跡しやすくなるはずである。レビューをサポートするシステムの開発も、今後の課題である。

## 参考文献

- [1] 竹田尚彦, 大岩元. プログラム開発経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, Vol33, No. 7, pp. 944-954, 1992.
- [2] 松澤芳昭, 青山希, 杉浦学, 川村昌弘, 大岩元. 「目的の表現」に注目したオブジェクト指向プログラミング教育とその評価. 情報処理学会研究会報告(CE-72-11), pp. 77-84, 2003.
- [3] 駒谷昇一. 新入社員に対するレビューを中心としたプログラミングの教育方法. 情報処理学会研究会報告(CE-38-12), pp. 67-72, 1995.
- [4] 長野宏宣, 浅見秀雄, 忠海均. 階層化プログラム設計図法-HCP チャート. 企画センター, 1992.
- [5] 花田收悦. プログラム設計図法. 企画センター, 1983.
- [6] 松澤芳昭, 杉浦学, 大岩元. プログラム設計教育における HCP チャートのレビュー手法.

情報処理学会第 66 回全国大会 (1C-2), pp. (4-343)-(4-344), 2004.

[7] 橋本千恵子. プログラム開発経験に基づく上流工程 SE 育成カリキュラムの開発と実施. 情報処理学会研究会報告 (CE-35-1), pp. 1-12, 1995.