

プログラミング授業の導入としての「お絵かきプログラム開発演習」

A Proposal Of “Describing Illustration with Language” For Introduction Education Of Programming

荒木 恵* 松澤 芳昭** 杉浦 学* 大岩 元***
Megumi Araki* Yoshiaki Matsuzawa** Manabu Sugiura* Hajime Ohiwa***

慶應義塾大学大学院政策・メディア研究科* 静岡大学** 慶應義塾大学***
Graduate School of Media and Governance, Keio University*
Faculty of Informatics, Shizuoka University**
Keio University***

〈あらまし〉 本発表では、「お絵かきプログラム開発演習」というプログラミング教育の導入のワークショップを紹介する。この演習では、「魅力的な絵を、期限内に、正確に、だれにでも描けるような日本語の"プログラム"」を発注し、設計し、実装し、それを第三者が評価して、その結果を討論する。我々はこの授業を大学生を中心に、高校生や社会人にも実践しているが、間違いなくクラスが盛り上がり、言語コミュニケーションの難しさを中心とするソフトウェア開発の本質的な討論がなされる。この授業は60分で実践でき、文系・理系問わず学習者が楽しめてプログラミングに対する姿勢をポジティブにする効果もある。

〈キーワード〉 情報教育 情報システム アンブラグド コミュニケーション 教材開発

1. はじめに

プログラミング教育はコンピュータを使いこなすための情報教育として行われてきたが、近年ではアプリケーションソフトウェアの普及に伴い、プログラミングそのものはコンピュータを使用するためには必須のものではなくなりつつある。しかしながら、プログラミング教育は、プログラミング能力の獲得だけでなく、そのプロセスを通して論理的な問題解決能力を育成することを目的とする教育であり、この目的を主とし、情報教育の柱として発展していくべきと、我々は考えている。

こうしたプログラミング教育は、複雑な状況から問題を同定し、多様に視点からの評価を経て問題解決をはかる、情報技術分野の一つである「情報システム (Information Systems)」のアプローチを取り入れていく必要がある (大岩 2007)。このとき、プログラミング教育は、言語を習得するだけではなく、コンピュータの本質を知るだけでなく、「自分ができる何らかのタスクを他人 (または装置) がやれるように、十分に詳細に、完全にその手続きを記述する」(UNESCO

2002) ことが目指されなければならない。

このような教育は、実際にプログラミングを利用する職業を目指していない文系の学生にも重要である。なぜなら、文系の学生といえども、社会に出て、ソフトウェアの発注や評価などの開発に関わる機会は少なくないからである。

30年前に書かれた、ソフトウェア工学で有名な「ブランコの漫画」(London University Press 1973, 有澤 1988)がある。これは、発注者の要求が、設計、実装に工程に誤って伝えられて、ユーザの要求に合わないものを作ってしまうソフトウェア開発のコミュニケーションを皮肉った漫画である。この問題はソフトウェア工学が発展した今なお健在である。仕事で他人 (または装置) に正確に物事を伝える方法論は、どのような職業でも必要である。

コンピュータを専門としないプログラミング初学者にとっても、ブランコの漫画に示されているようなコミュニケーションの問題が起こりうることを知ることは意義がある。起こりうる問題をあらかじめ知っておくことで、何を学ばばよいか分かり、プログラミ

ングの学習の動機付けになるからである。

しかしながら、この漫画をソフトウェア開発体験のない初学者にただ見せても、その意図するところが分からない。この漫画で提示されている問題を理解するには、ソフトウェア開発プロジェクトの全体像を知り、コミュニケーションの問題を体験する必要がある。

プログラミングを通じてソフトウェア開発やコミュニケーションの難しさとは何かを知るには、技術力と、それを習得するための長い時間が必要である。それまでに退屈な言語の使い方だけを覚えさせられては、プログラミングはおろか、コンピュータも嫌になってしまう。

そこで我々は、プログラミング教育の導入授業で行うことのできる、コンピュータを使わずに (Unplugged)、ソフトウェア開発の視点を身につけることができるワークショップ「お絵かきプログラム開発演習」を開発し、実践している。このワークショップには、われわれの"プログラミング"に関する考え方が抽象化されて詰め込まれており、初学者は楽しみながら、これから学ぶことの全体像を知ることができる。

このワークショップでは、先にあげたブランコの漫画で示されているような、様々なコミュニケーションの問題が再現されるので、社会人でも新たな発見があり、議論を楽しめる。それだけでなく、プログラミング初心者や文系学生にも好評で、プログラミングに対する不安感を取り除く効果や、その後のクラスの雰囲気が変わり、受講者同士のコミュニケーションが円滑に進むようになるアイスブレイキングの効果もある。

本稿では、このワークショップ「お絵かきプログラム開発演習」を紹介し、このワークショップの実践方法を示す。

2. ワークショップ

プログラミングの初学者がソフトウェア開発におけるプログラミングの位置づけを理解し、ソフトウェア開発の過程でおきるコミュニケーションの問題を体験できるワークショップ「お絵かきプログラム開発演習」の実施方法を紹介する。

2.1. ワークショップの方法

このワークショップの中心となる活動は、「魅力的な絵を、期限内に、正確に、誰にでも描けるような日本語プログラム」を開発することである。開発された日本語プログラムに対して実行テストを行い、プログラムの正確性を検証する。

絵を、特定の手順に従って描くというワークショップには、他に「Kid Fax」(Tim Bell ほか 2007)や「図形の作文ワークショップ」(佐藤ほか 2004)が挙げられる。提案するワークショップは、上記のワークショップと、「コンピュータそのものの仕組みに興味を持たせ」「コンピュータサイエンスの面白さに直感的に触れる」という点で共通している。

このワークショップではコンピュータサイエンスだけでなく、情報システムの考え方も取り入れている。ソフトウェア開発で起こるコミュニケーションの問題を体験してもらうために、要件定義、設計、実装、テスト、そして評価を含む「開発プロジェクト」を実施する。このワークショップで行うプロジェクトは「お絵かきプログラム開発プロジェクト」である。

2.2. プロジェクトの概要

お絵かきプログラム開発プロジェクトは、5名のメンバーで構成される。お絵かきプログラム開発プロジェクトメンバーに与えられる「問題」は「魅力的な絵を、誰でも、正確に、期限内に描けるような日本語プログラムを開発する」ことである。

プロジェクトは要求分析、設計、実装、テストの4つのフェーズで構成される。これらのフェーズは、発注者、設計者、プログラマー、テスターが担当するが、すべて異なる人が作業を行う。各フェーズ間のコミュニケーションは書面でのみ行われる。それぞれのフェーズにおいては、時間期限も定められている。

プロジェクト終了後には、メンバー全員がそれぞれの立場からプロジェクトの評価を行い、プロジェクトの成功や失敗について議論する。

2.3. プロジェクトのプロセス

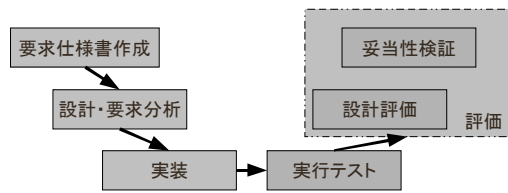


図2 プロジェクトのプロセス

お絵かきプログラム開発プロジェクトは、**図2**のプロセスで進行する。

ここでは「三日月」を描く日本語プログラムを開発するプロジェクトを例にとって、実際の成果物例(**図3**)を挙げながら説明する。

要求仕様書の作成

施主が、日本語でどのような絵を描くプログラムが欲しいかを要求仕様書に書く。

要求分析・設計

設計者が施主から要求を受け取り、要求を満たす絵を設計し、設計書に設計を記す。

ここでは、日本語以外に、図面を用いて設計を行ってよい。

実装

プログラマーが設計者から設計を受け取り、設計された絵を誰でも、正確に、期限内に描けるようなプログラムを記述する。ここでは、日本語のみしか用いてはならない。

テスト

テスターが、プログラマーが記述した日本語プログラムにしたがって絵を描く。

プログラマーによるプログラムの記述のよしあしと、テスターによるプログラムの解釈のよしあしを区別するために、2名のテスターによって一つのプログラムが検証される。人間がテストを行うため、同じプログラムでも解釈や読解力の差によって違う実行結果が得られることがある。

評価

プロジェクトメンバー全員がそれぞれの立場からプロジェクト評価を行う。施主は要求を満たすものができたかを中心に評価し、それ以外のメンバーは成功や失敗の原因、改善点を考察する。評価の際にはプロジェクトの成果物をすべてまとめ、プロジェクトの全体像を見られるようにしてから評価を行う。

3. 授業

このワークショップをプログラミングの導入授業で活用するための方法を紹介する。

3.1. 導入

授業のはじめに、コンピュータを使わないプログラミングワークショップを行うことを伝える。その後、ソフトウェア開発のプロセス(要求・設計・実装・テスト・評価)について簡単な紹介を行い、ワークショップで使う語句(施主・プロジェクトなど学生が知らない言葉)について説明する。

このワークショップでは発注者を「施主」と呼んでいる。多くの学生にとって「発注」は自分の体験から遠い言葉である。学生に「高いお金をかけて、自分が依頼し、多くの人に関わるもの」をイメージしてもらうには、「マイホーム」などの建物の例を用いるのが適切と考え、「施主」という言葉を用いている。

3.2. ワークショップの準備

次に、お絵かきプログラム開発演習の実施方法を学生に解説する。授業でお絵かきプログラム開発演習を行う場合、効率よく(60分で)進めるために、演習の説明書を作成した付録の**図9**に示す。

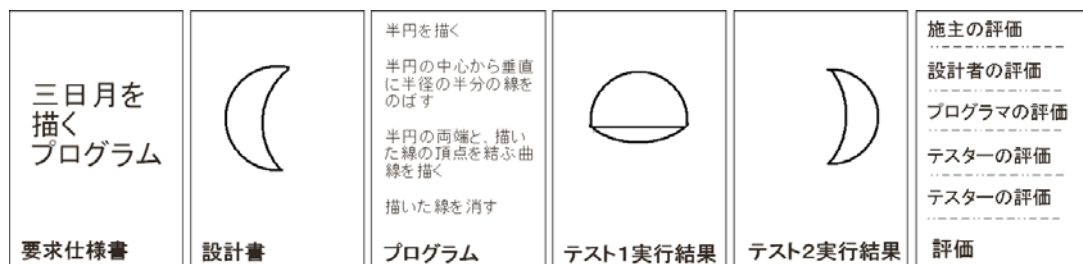


図3 プロジェクトの成果物例

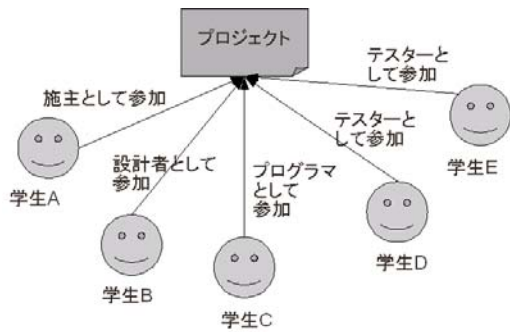


図4 プロジェクトとメンバーの関係

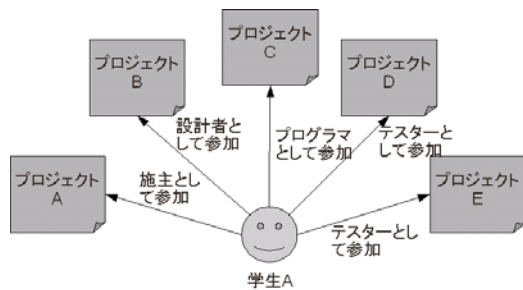


図5 メンバーとプロジェクトの関係

この説明書に従うと、1プロジェクトには5名が参加し(図4)、1人は5プロジェクトにそれぞれ違った立場で参加することになる(図5)。

この方法で演習を行うことにより、全員がすべての役割を体験することができ、グループ内で複数のプロジェクトの実施されるため、立場の違いやプロジェクトの比較などの議論ができる。

手順は説明書にすべて書かれている。説明書を学生に5分ほどで読ませ、分からないところがあったら質問をするように指示する。

3.3. ワークショップ

お絵かきプログラム開発演習を行う。演習の実施方法は「ワークショップ」の章で述べたとおりである。演習時間のあいだは、指導者は説明書に書かれたタイムテーブルにしたがって時間管理を行う。

指導者は議論の時間になったら、各グループを巡回し、グループごとの面白い失敗や、成功について考察をするように促す。

3.4. 全体討論

議論の時間終了後に、クラス全体で失敗や考察、成功例などについて共有する全体討論の時間を設ける。ここで、ソフトウェア開発

メロンを描く プログラム	
大きな円を描く。 横が縦の2倍長いTの字を描く。 円の中に細かい線をいっぱい描く。	
要求	設計
プログラム	実行結果

図6 プロジェクトAの成果物

で頻出する問題(要求があいまい・設計が非現実的である・プログラムのバグ・目的が共有されていないなど)を取り上げ、実際の事例を紹介する。

最後に、ソフトウェア開発のプロセスについてももう一度紹介し、ブランコの絵を見せる。そして、ブランコの絵のような問題が実際のソフトウェア開発プロジェクトでもおきていることを知らせる。

4. プロジェクトの例

お絵かきプログラム開発プロジェクトは、成功させることが目的なのではなく、成功・失敗を体験し、その要因を討論することに意義がある。ここでは、実際に行われたプロジェクトの中から、いくつかの例を取りあげて、その経緯とメンバーの考察を紹介する。

プロジェクトA: メロンが量りに

プロジェクトAの結果を図6に示す。このプロジェクトでは、施主は「メロン」を発注している。設計者はマスクメロンを設計している。プログラマーはこの絵を「円と、T字と、細かい線でできている図形」と分析して、プログラムとして記述している。

このプログラムをテスターが実行したところ、1人のテスターは「細かい線」の意味が分からず、独自の解釈で「はかり」だと考え、「はかり」を描いている。もう一人のテスターは、メロンを描くことができていた。

このプロジェクトの評価として、テスターは「メロンならば、メロンだとプログラムに書いてほしかった」と評価している。プログ



図7 プロジェクトBの成果物



図8 プロジェクトCの成果物

ラマーはこれに対して「メロンと書けば、すぐに分かったはずだった」と反省している。このような失敗例を用いて、プログラミングの授業で、コメントに目的を書くことの意義についての講義を円滑に進めることができる。

プロジェクトB: 美しい日本

このプロジェクトの結果を図7に示す。このプロジェクトでは、要求が「美しい日本」という抽象的なものが設定されている。設計者はその要求に困ってしまい、制限時間内で設計を済ませるために「日本列島を輝かせる」という設計を行っている。設計を受けとったプログラマーは、これを「日本列島と、小さな島と、いくつかの十字型のもの」と解釈してプログラムを書いている。

そのプログラムの実行結果は、日本列島の周りに、小さな丸い島と十字型が配置された絵になっている。

このプロジェクトの評価として、施主は「美しい日本とは、紅葉や富士山、海など、旅行パンフレットに出てくるような美しい日本の景色のことだった」とコメントし、結果に満足していないと評価している。これに対して、設計者は「そんな要求だとは分からなかった、分かったとしても、5分でその絵を描くことはできない」と反論している。こうしたあいまいな要求による問題は、実際のソフトウェア開発現場でも起きている。

プロジェクトC: かわいくないキャラクター

プロジェクトの結果を図8に示す。このプロジェクトでは、施主は「かわいいキャラク

ター」を要求している。しかしながら、設計者の画力不足により、すでに設計の時点で、キャラクターはかわいいとはいえないものになっている。プログラマーは正確にキャラクターを描画するプログラムを記述しており、プログラムの実行結果は設計どおりになっている。

施主は「ぜんぜんかわいくない」と、この結果を不満足と評価している。しかしながら、要求記述という観点からは、「かわいい」というあいまいな要求をした発注者に問題があるだろう。

このように、単純な言葉遣いのミスや、背景知識の違いにより、意図と異なるものになるプロジェクトが多い。これは受講者には意外であり、結果の評価は大いに盛り上がり、指定された時間内に議論が終わらないことも多く、全体討議では積極的に面白い失敗例が発表される。

5. 参加者の感想の分析

我々はこの演習を、高校生・文系大学学部生・理系大学学部生・社会人を対象に行っている。本章ではこれらの参加者の感想を「ソフトウェア開発プロセスへの理解の深まり」「アイスブレイキングの効果」「プログラムに対する理解の深まり」の観点から分析して述べる。

5.1. ソフトウェア開発プロセスへの理解の深まり

- 相手に伝わらないので要求仕様書など

はしっかり書かないといけないということがわかりました。

- 仕事としてのプログラミングの流れと
いうものがわかったような気がした。
- 要求の具体性が大切。(施主の要求があ
いまいで困った設計者)
- 施主は実行結果に満足したが、テスター
は何を描いているかわからなかった。目
的が共有されていなかった。
- 設計者が施主の発注とは異なる設計を
した。
- テスターがプログラムから(自分の解釈
で)トトロだと読み取って実行してくれ
た。人間でなかったらできない。
- 要求と実装力のバランスが大切である。

このように、ソフトウェア開発プロジェクトで起こるコミュニケーションの問題や、ソフトウェア開発プロセスの全体像を体験することができたことがわかる。

5.2. アイスブレイキングの効果

- なによりも一番印象的だった授業は、第
1回目の授業です。「論理思考とプログ
ラミング」なんていう授業名からして、
相当エグイのだろうなと思っていまし
た。そもそもパソコン自体に詳しくない
し、プログラミングなんてもってのほか
だと私は思っていたので、ただ必修だ
から取るという感じでいました。そこ
に待っていたのが、何やら楽しいゲー
ム。絵を描いたり、出来上がってきた
絵を見て爆笑したりと、こんなにも
楽しいものなのかと思いました。導入
としてはすばらしい授業だったと思
います。(プログラミング初心者の文
系学生)
- 初めて会う人ばかりで、皆パソコン
も詳しく自分がついていけないのか
不安ではあったが、初めに行ったお
絵かきプログラム開発演習を通して
一部の人達と交流し仲良くなること
ができたと思う。(プログラミング
合宿参加の高校生)
- お絵かきプログラムは楽しかった。

このように、プログラミングに対する恐怖感が払拭され、クラスメイトと仲良くなるなどの効果があることがわかる。

5.3. プログラムに対する理解の深まり

- 成功したのは、プログラムが細かく書
いてあるか、とても分かり易いかのど
ちらか二つだった。プログラムはき
つちり書かないと上手くいかないこ
とがよくわかった。
- 人に読んでもらうプログラムは、情
報量が多くても少なくてもダメ。
- 図や絵を日本語で表すのが難しい。
- プログラムが詳しいと絵を描きやす
い(作るものが何かをプログラムに書
いてもらうとよい)
- プログラムにミスがあり、テスター
2人の実行結果が異なった。
- 大きいものから小さいものへプロ
グラムを書くと、テスターに分かり
やすい。このように、プログラムは考
えて書くものであることや、目的を
書くこと(コメントを書くこと)の必
要性を理解したことがわかる。

6. 今後の課題

筆者らは2005年から、お絵かきプログラム開発演習を、大学生、社会人、高校生に対して実施してきた。2007年春に現在の方法を開発し、60分で実施可能になった。しかしながらその後の実践によって、まだいくつかの問題点があることが分かっている。学生への指示や、学生のレベルに応じた議論の進め方などである。

お絵かきプログラム開発演習の実践結果とマ教材は

<http://www.crew.sfc.keio.ac.jp/projects/2007/DrawingProject/>

にある。ワークショップの洗練のために、意見をいただければ幸いである。

7. おわりに

現在のプログラミング教育では、プログラミング言語の文法や、コンピュータサイエンスについてのみ教えている。そのため、自分と、コンピュータやプログラムがどのようなかわりを持っているのか、何のためにプログラミングを学ぶのかがあやふやなまま、授業を受けることになる。

筆頭著者は2年前、プログラミングにあま

り良い印象を持っておらず、学習する動機も得られていなかった。そんなときに、プログラミング授業の導入ワークショップとして、このワークショップを体験し、「これから習うプログラミングは、システムやソフトとつながっているのだ！」と、強い印象を受けたことを覚えている。それまで、プログラムはコンピュータを操るためだけに学ぶものだと思っていたが、このワークショップを通じて、ソフトウェア開発という「ものづくり」の一環としてプログラミングを学ぶ理由を感じることができた。同時に、「人と協調作業をするためには、意図と手順を正確に表さなければならない」という頭では分かっていたが体験したことがなかったことを体験でき、新鮮な驚きがあった。

学生の立場から見て、これはとても面白いワークショップであったし、ワークショップを TA として実施する側になった今は、手軽に出来て、学生の面白い失敗プロジェクトを見て、学生の議論に参加する楽しさがある。手軽に出来るワークショップの一例として、ぜひ、授業に取り入れてみていただきたい。

参考文献

- 有澤誠 (1988) ソフトウェア工学. 岩波書店
- 大岩元 (2007) 情報教育と 21 世紀の教育. SSS2007 論文集
- Tim Bell, Mike Fellows, Ian Witten , 兼宗進 訳 (2007) コンピュータを使わない情報教育 アンブラグドコンピュータサイエンス. イーテキスト研究所
- 佐藤雅彦, 村上泉, 山本晃士, 菅俊一, 石川将也, 佐藤匠 (2004) コンピュータサイエンスにおける教育原理. 平成 14 年度ハイテク・リサーチ・センター報告書 pp.488-501
- UNESCO .ICT Curriculum for School /Program of Teacher Development. (2002)
<http://unesdoc.unesco.org/images/0012/001295/129538e.pdf>
- University of London Computer Center Newsletter, No.53, March 1973

付録

問題：『魅力的な絵を、期限内に、正確に、だれにでも描けるような日本語プログラムを作成しなさい。』

演習の手順：

学生ひとりひとりが、施主・設計者・プログラマー・テスター（プログラムの実行者）のすべての役割を体験し、プログラムを作る。書かれた設計書・プログラム・実行結果は最後に施主に返され、施主が評価を行う。

タイムテーブル	
<p>施主になる時間（2分） ひとりひとりが、<u>要求仕様を①の要求仕様書に書きます</u>。時間がきたら、①を隣の人にわたします。</p>	<p>① 要求仕様書とか。</p>
<p>設計者になる時間（5分） もらった①をもとに、②の設計仕様書に<u>設計を描きます</u>。絵を必ず描くこと。時間が来たら、②を隣の人にわたします。</p>	<p>② ①をもとに設計を描く。</p>
<p>プログラマーになる時間（10分） もらった②をもとに、③を描くプログラムを日本語で書きます。絵や図は使ってはいけません。日本語のみです。時間がきたら、③を隣の人にわたします。</p>	<p>③ ②をもとにプログラムとか。</p>
<p>プログラムの実行テストをする時間（5分） ③のプログラムをもとに④のテスト結果用紙に<u>絵を描きます</u>。時間がきたら、③のプログラムを隣の人にわたします。</p>	<p>④ ③をもとに (1) 実行し絵とか。</p>
<p>プログラムの実行テストをする時間（5分） ③のプログラムをもとに④のテスト結果用紙に<u>絵を描きます</u>。時間がきたら、③のプログラムと④の絵を施主に渡します。</p>	<p>④ ③をもとに (2) 実行し絵とか。</p>
<p>評価の時間（3分×5） 施主が、自分が施主になっている紙をすべて集め、評価用紙をかさねてクリップで留めます。書き込みます。3分たったら、紙の束を隣の人に渡します。これを5回繰り返します</p>	<p>施主の手に全部を1度おさえる。評価する。</p>
<p>ディスカッションの時間（10分） 結果についてディスカッションを行います</p>	

図9 演習説明書