

「ソフトウェア工学」の授業方法

片山 徹郎

(宮崎大学 工学教育研究部)

はじめに

工学部情報システム工学科のカリキュラムの専門必修科目の1つに、筆者が担当する「ソフトウェア工学」がある。この「ソフトウェア工学」の学生による授業改善アンケートの結果が総じて高い傾向にあるという報告を受けた。

そこで、この科目を担当するにあたり、筆者が準備していることと実践していることについて報告する。具体的には、「ソフトウェア工学」を実施するにあたり、シラバスでの工夫、講義での工夫、成績評価方法、レポート課題での工夫、レポートの試問の実施、などについて述べる。

1. 「ソフトウェア工学」の概略

「ソフトウェア工学」の講義の概略について述べる。「ソフトウェア工学」は、工学部情報システム工学科の専門必修科目の1つであり、3年生の前期に実施している。教科書は採用しておらず、講義のベースにしている本は、以下の2冊である。

- 「ソフトウェア工学入門改訂新版」河村一樹、近代科学社
- 「ソフトウェア開発 (改訂2版)」小泉寿男、辻秀一、オーム社

なお、講義は、板書スタイルで行っている。このため、講義中の学生は、講義内容を聞きつつ、ノートを取るようになる。

この科目では、学生を評価するために、レポート課題を2回課しており、定期試験は実施していない。レポート課題の2回は、以下を与えている。

1. 与えられた仕様に基づいて、プログラムとマニュアルを作成する。
2. 作成したプログラムの機能拡張を考えるとともに、テストデータを与えて、出力結果について検討する。

レポート課題で学生を評価しているのは、下記の2つ

の理由からである。

- ソフトウェア開発の各工程を一通り疑似体験させる。
- ソフトウェア開発における難しいとされるポイントを理解させる。

前者の理由について、具体的に述べる。情報システム工学科の学生は、1年生より、プログラミング演習を受けてきており、プログラミングの難しさや大切さは十分に理解している。ただし、このプログラミングという作業はソフトウェア開発の重要な一工程ではあるが、ソフトウェア開発全体では、その他にも、ユーザからの要求獲得や、仕様策定、テスト、保守など、さまざまな知識が必要となる。このため、それらの各工程を一通り体験できるように、レポートを実施している。

後者の理由のために、2回のレポート課題の提出後に、レポートの試問を実施している。学生の提出したレポートに対して、こちらから質問し、問題点を指摘することによって、提出レポートに自分が記述したことに責任を持たせるとともに、ソフトウェア開発における難しいとされるポイントについて、学生の理解を促す。

2. 「ソフトウェア工学」の授業改善アンケート結果

図1に、2018年度の「ソフトウェア工学」の学生による授業改善アンケートの結果を示す。どの項目も、総じて高い結果になっていることが分かる。個人的に重要視している項目は、

- 「A2: 総合的に判断して、この授業に満足しましたか」
- 「A8: 授業の内容に興味を持つことができた」
- 「A9: 授業の内容は体系的 (筋道がはっきりし、全体がよくわかる内容) だった」

の3つであり、この3つの項目にはすべて、半数以上が「4. そう思う」と回答している。

なお、この傾向は、2017年度以前のアンケート結果

「授業改善に関する学生によるアンケート」の結果報告書

■講義名：ソフトウェア工学
 ■担当教員名：片山 徹郎
 ■開講年度：2018年度前期

1: そう思わない
 2: どちらかといえばそう思わない
 3: どちらかといえばそう思う
 4: そう思う

A項: 共通質問項目	0	1	2	3	4	合計
A1: この授業の到達目標を達成できましたか	0	0	6	22	24	52
A2: 総合的に判断して、この授業に満足しましたか	0	0	1	20	31	52
A3: 授業の事前準備、授業後の学修、試験準備に要した学修時間は、1週間当たり平均何時間でしたか	0	5	16	21	10	52
A4: この授業のシラバスを、科目選択の参考や準備学修等に利用しましたか	0	4	7	23	18	52
A5: 私はこの授業に意欲的に取り組んだ	0	1	4	17	30	52
A6: 私はこの授業に関して予習、復習等の自主的な学習を行った	0	3	10	28	11	52
A7: 授業の内容はシラバスに沿って進められた	0	0	1	23	28	52
A8: 授業の内容に興味を持つことができた	0	0	2	14	36	52
A9: 授業の内容は体系的(筋道がはっきりし、全体がよくわかる内容)だった	0	0	1	11	40	52
A10: 授業の進むスピードは適切だった	0	0	0	11	41	52
A11: 授業担当者の話し方は明瞭で聞き取りやすかった	0	0	1	11	40	52
A12: この授業の「板書・OHP・パワーポイント」(実施された方法)は分かりやすく、見やすかった	0	0	3	29	20	52
A13: この授業に使われた教科書・補助教材・資料(配布プリント等)は理解しやすかった	8	0	3	21	20	52
A14:						
A15:						

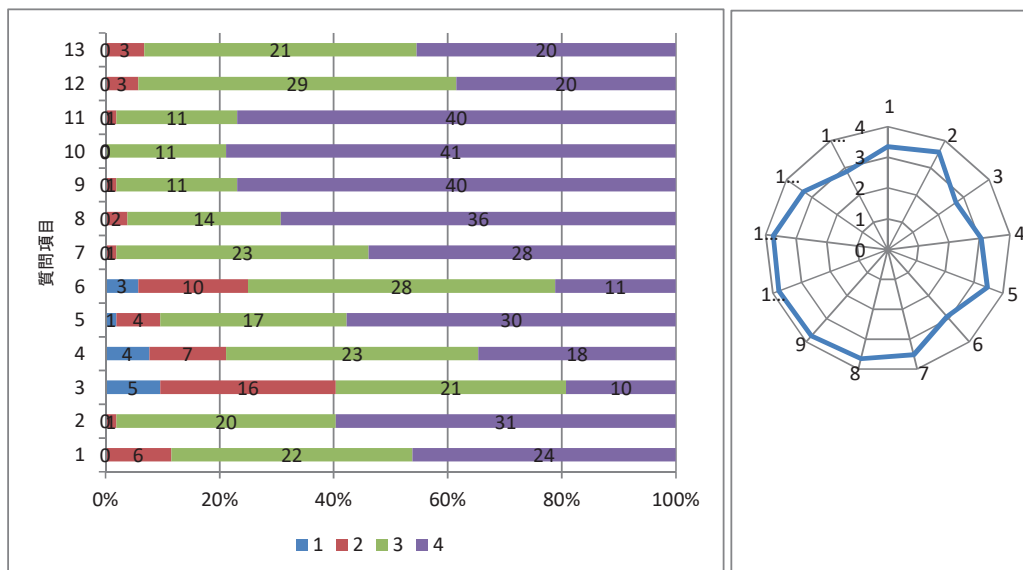


図 1 2018 年度「ソフトウェア工学」の学生による授業改善案アンケート結果

についても、おおむね変わっていない。

3. 「ソフトウェア工学」の概要と授業内容を考えるに至った理由

「ソフトウェア工学」での工夫について述べる前に、そもそも「ソフトウェア工学」とは何か、について説

明する。「ソフトウェア工学」とは、ソフトウェアの開発に、工学のアプローチをとることである。ソフトウェアは人工物なので、工学の対象物として扱うことができる。すなわち、他の土木工学や建築工学、電気電子工学、機械工学と同様に、その生産性や品質向上、コスト削減、チームによる開発などを考える研究分野である。

学習目標
身につける資質・能力
(a)ソフトウェア工学で使う用語を、正しく理解し説明することができる。 (b)ソフトウェアの特性、および、ソフトウェア開発が難しいとされる理由を理解した上で、ソフトウェア開発の際に起こりうる問題について把握する。 (c)良いソフトウェアの要点を把握し、ソフトウェアのライフサイクルや開発プロセスを理解した上で、自分がソフトウェアを開発する際に活用できる。

図 2 「ソフトウェア工学」の学習目標（シラバスより抜粋）

<ul style="list-style-type: none"> ● ソフトウェアって何？ ● NATO って何？ ● その NATO がなんでソフトウェア開発に口を出したと思う？ ● 入力が整数 N で、出力は 1～N の和。このコードを書いて。 ● 中国のソフトウェア会社に「背景は空色」で発注。何色の背景になったと思う？ ● フローチャートの欠点は？ ● $i=3; A[i++] = i;$ さて、どこに何の値が入る？ ● 2つのモジュール A, B があって、この2つをテストをしたら、A はほとんどない数のバグが出て、B はほとんどバグが出なかった。もう少しテストする時間があるとしたら、A と B どっちをテストすべき？
--

図 3 2018 年度「ソフトウェア工学」の授業で実際に学生に尋ねた質問の例

そもそも「ソフトウェア」という言葉は、「コンピュータ」が登場する以前には存在しない言葉である。そのため、「ソフトウェア工学」という言葉が生まれたのも古くはない。誕生のきっかけは、1968年に、NATO(北大西洋条約機構)が、「ソフトウェア危機 (Software Crisis)」を提唱し、1969年に世界で初めて、ソフトウェア工学に関する国際会議 ICSE (International Conference on Software Engineering) が開催されたことにある。すなわち、「ソフトウェア工学」という言葉は誕生して約 50 年しか経っていないため、学問としては若い分野である。土木工学や電気電子工学、機械工学などと比べると、明らかに後発であり、物理や数学などの学問の歴史と比べると、学問として成り立つはずがない、と言われても仕方がないくらい歴史が浅い。

つまり、「ソフトウェア工学」は、学問としては未成熟であり、現在進行中の研究分野なのである。例えば、「バグ 0 のソフトウェアの作り方」が既に存在していれば、学生に授業で教えることができるが、現実には、「PC 上のソフトが動かなくなり固まった」や「スマホのアプリが突然落ちた」という現象が身の周りで日常茶飯事に起こることから分かるように、残念ながらそのような方法論は、未だ存在しない。このような状況なので、「ソフトウェア工学」という授業の時間

枠で一体何をどう教えればいいのか、がこの授業の内容をどうすべきか考えるに至った理由である。

なお、1章で教科書は採用していない、と記述したが、その理由もここに起因する。ソフトウェアの世界は、日々新しい技術や方法論が提案されており、現在の考え方が、5年後10年後もそのまま使える保証はない。5年前10年前の教科書がその時に役に立つ保証はなく、日々新しい技術を調査する方が重要だと考えるためである。

4. 「ソフトウェア工学」での工夫

この章では、「ソフトウェア工学」で実施している工夫について述べる。

(1) シラバスでの工夫

シラバスでの工夫について述べる。図 2 に、シラバスから抜粋した学習目標を示す。学習目標は 3 つ掲げている。

(a) については、専門用語の正しい理解である。学生がソフトウェア開発関連の会社に勤めた際に、会話で使われる基本常識となる言葉について理解しておくことを目標としている。(c) については、良いソフトウェアの特徴を知り、自分が開発する際にそれらを活

成績評価方法			
評価手段	実施内容	評価比率	学習目標との対応
講義の最後に演習課題を実施	その回の講義内容	40%	(a)(b)
レポート	仕様からソフトウェアを開発する	60%	(c)

図 4 成績評価方法（シラバスより抜粋）

用できる能力である。これについても、学生が将来、職場でソフトウェアを開発する際に、良いソフトウェアだとユーザが判断するポイントを理解した上で、開発に臨んでくれることを目標としている。

(b)の文章は、熟考の末にこうした経緯がある。具体的には、「…ソフトウェア開発の際に起こりうる問題について把握する。」という箇所である。通常、例えば、数学で微分方程式を取り扱う授業であれば、学習目標には「微分方程式が解けること」と書き、その解き方を授業で取り扱い、試験では、実際に微分方程式を学生に解いてもらい、その結果で、学生の理解度を確認することになるだろう。ソフトウェア工学であれば、「良いソフトウェアが開発できること」と書きたい。しかし、3章で記述した通り、現状の技術では、確立した方法がないため、これはプロでも難しい。したがって、「なぜソフトウェア開発が難しいのか、どこにその原因があるのか。」について理解すればよい、ということにした。

(2) 講義での工夫

講義での工夫だが、講義自体は、オーソドックスに進めており、あまり工夫点はない。講義は、板書スタイルで行っている。最近では、授業にパワーポイントなどのプレゼンテーションソフトを使うことによって、限りある授業時間を有効活用する人が多いと思うが、「ソフトウェア工学」については、3章で述べたように、何を教えるべきか、で考えてしまうくらい教えるべき内容が豊富にはない。このため、板書での授業スピードで実施している。なお、2章で、この講義の授業改善アンケートの結果について述べたが、授業スピードについて、不満はほぼ出ていないことが分かる（図1のA10参照）。

講義は、ソフトウェアの開発プロセスの各工程に沿って進めている。工程は、大きく分けると5つあり、要求仕様定義、設計、実装、テスト・デバッグ、運用・保守、である。講義の内容は、各工程で使用する専門用語について述べるとともに、各工程での難しい点、重要となる考え方、失敗する理由、などについて述べ

ている。

これらに加えて、講義では、ほぼ毎回学生に1回は質問して当てるようにしており、これが唯一の工夫点かもしれない。図3に、2018年度の「ソフトウェア工学」の授業で、実際に学生に尋ねた質問のいくつかを示す。質問は、少し考えれば正解にたどり着くものや、中学や高校での知識などを問う内容であり、講義を聞いていれば回答できる、といったものがすべてではないし、成績とは全く関係ない。質問は、必ず教室の後に座っている学生から順に当てるルールで、授業の1回目に「この講義ではたまに学生さんに質問して、答えてもらいます。当てる時は、後に座っている人から順に当てていきます。」と宣言している。これを授業の1回目、2回目と続けると、学生は基本的にみな素直なので、3回目の授業くらいから、全員前の方に座るようになる。なお、年によっては、わざと教室の後に座り、「今日は質問に答えてやる」と意気込む学生達がいることもあり、これはこれで授業の雰囲気は良くなる。

(3) 成績評価方法

成績評価方法について述べる。図4に、シラバスから抜粋した成績評価方法を示す。成績評価方法は大きく2つに分けている。1つは、毎回の講義の最後に演習課題を実施している。これは、その回の授業内容について尋ねており、ソフトウェア開発の各工程での専門用語や重要なポイントの理解について評価するとともに、出席調査も兼ねている。もう1つは、2回のレポート課題であり、小規模な仕様を与えて、ソフトウェア開発の一連の流れを体験するとともに、ソフトウェア開発がなぜ難しいのかについて、理解させる。

成績評価の重みは、演習課題を40%とし、レポートを60%としている。これにより、どちらか一方の評価が極端に低い場合は、合格できないようにしている。また、学生に対して、レポートが重要であるという意識を暗黙に持たせている。

平成 27 年度 ソフトウェア工学 レポート課題 Part1

「ソフトウェアのマニュアル作成」

平成 27 年 5 月 21 日
片山 徹郎
kat@cs.miyazaki-u.ac.jp

- 以下の仕様を満足するプログラムを作成しなさい(ソースコードを提出)。
- 作成したプログラムのマニュアル(取扱説明書)を書きなさい。
- 事前に、レポート提出までのスケジュール計画(日程と実施内容)を立てること。この事前を立てた初期スケジュールと、実際の作業結果とを示し、2つを比較した上で、自分が最初に立てたスケジュールについて考察しなさい。
- 作成に当たり、自分が工夫した(時間をかけた or 力を注いだ)箇所を挙げなさい。(1 つでも複数でもよい)

※ ここから、仕様です。

- 宅配便の配送料の値段(配送料金を)計算するシステムを、作成して欲しい。
- 宅配便の配送料は荷物のサイズによって決まる。
- 荷物のサイズは、大きさと重さで決まる。
- 荷物の大きさは、箱の縦、横、高さの3辺の和とする。
 - 参考：ヤマト運輸のページ <http://www.kuronekoyamato.co.jp/takkyubin/takkyu.html#size>
- 荷物のサイズと料金は、下記の表の通り。
 - ただし、大きさ、重さのどちらか一方が超えている場合は、超えている方のサイズの扱いとする。
 - 160 サイズを超える荷物は取り扱わないとする。



サイズ	荷物の大きさ	重さ	料金
60 サイズ	60cm 以内	2kg まで	756 円
80 サイズ	80cm 以内	5kg まで	972 円
100 サイズ	100cm 以内	10kg まで	1,188 円
120 サイズ	120cm 以内	15kg まで	1,404 円
140 サイズ	140cm 以内	20kg まで	1,620 円
160 サイズ	160cm 以内	25kg まで	1,836 円

- 出力は、配送料金の表示とする。
- システムへの入力形式については、特に定めなし、つまり、荷物の大きさとして、縦、横、長さ、重さがすべて入力可能であれば、独自に定めて構わないとする。

1/2

- 例えば、下記の入力例に対する、それぞれの出力を示す。

例 1)

入力： 縦：40cm
横：30cm
高さ：20cm
重さ：7kg
出力：1,188 円

例 2)

入力： 縦：10cm
横：20cm
高さ：20cm
重さ：3kg
出力：972 円

例 3)

入力： 縦：50cm
横：60cm
高さ：30cm
重さ：10kg
出力：1,620 円

※ 以下は、注意事項です。

- 提出締切は、**6/18(木) 13:00** (「ソフトウェア工学」授業開始前) です。
- 提出は、授業の際に持参するか、締切までに A330 の郵便受けに入れるか、MAIL で提出しても構いません。
- もし、プログラムがうまく動かない場合は、作成した部分までのプログラムを提出すること。この場合のマニュアルは、「完成したプログラムがあると仮定」して、マニュアルを作成するか、または、作成した部分のみのマニュアルを作成し、提出すること。
- 2 回目の課題の提出後、レポートの試問を行なう予定です。ただし、全員試問を行なうかは未定です。
- グループでの提出を認めます。例えば、5 人で取り組んだ場合、提出するレポートは、5 名連名のもの 1 部とし、レポートには、「リーダー」「○○部分のコーディング担当」など、その役割分担について全員分記述すること。また、この場合のスケジュールの記述は、初期スケジュールを全員分書いた上で、作業結果を全員分書くこと。スケジュールに対する考察については、個人がそれぞれ書いてもいいし、グループでまとめて書いてもよい。
- <http://earth.cs.miyazaki-u.ac.jp/~kat/lecture/SE/> に、この pdf ファイルを置いておきます。

2/2

図 5 2015 年度 1 回目のレポート課題

(4) レポート課題での工夫

レポート課題について述べる。前述したように、レポートは 2 回出している。1 回目は主に、仕様書からのプログラム作成とマニュアル作成から成る。2 回目は主に、1 回目で作成したプログラムの品質と保守から成る。この構成のポイントは、2 回目のレポートは、1 回目のレポート提出後に課題として与えているが、独立した課題ではなく、連続したものになっており、2 回目の回答は、1 回目の自分の提出物に依存する点である。これは、ソフトウェアを開発し、ユーザに納品した後に起こりうる状況について、体験してもらうよう、この形をとっている。なお、これらのレポート課題は、pdf ファイルを、こちらが準備した Web サイトからダウンロードできるようにしている。

レポート課題について具体的に説明する。図 5 に、2015 年度の 1 回目のレポート課題を、図 6 に、2 回目のレポート課題を、それぞれ示す。この年度の課題は、宅急便の料金計算である。実際の宅急便の計算と同様、荷物の縦、横、高さ、重さの 4 つを入力して、その料金を出力する。この仕様を与えて、プログラム

とマニュアルを作成する。

以下、1 回目のレポート課題の各項目と課題のポイントについて述べる。

- 課題 1. 仕様を満足するプログラムを書く
簡単な仕様を与えて、依頼者であるユーザが満足するプログラムを書く。「要求分析」「仕様策定」「実装」について理解する。
- 課題 2. ユーザが使えるようマニュアルを書く
依頼されて作成したプログラムは作成者である自分ではなく、第三者であるユーザが使うものであるということを理解し、そのユーザが、プログラムを使えるために十分なマニュアルを書く。「要求分析」「実装」「運用」「ユーザ視点」について理解する。
- 課題 3. 事前にスケジュールを立て実際の結果と比較したのち考察する
ソフトウェアはその作成工程が目に見えないため、ソフトウェア開発を予定通りに進めることは難しいことを理解し、計画を立て予定通りに進んでいるかを確認する経験をさせる。「開発プ

平成 27 年度 ソフトウェア工学 レポート課題 Part2

「ソフトウェアのマニュアル作成」

平成 27 年 6 月 18 日
片山 徹郎
kat@cs.miyazaki-u.ac.jp

- 先に提出したあなたのプログラムに対して、次の 2 つの拡張をそれぞれ行なう場合、どこをどのように拡張すればよいですか？ 設計もしくはプログラミングの方針を考えて述べるとともに、その拡張にかかる時間を、それぞれ見積もりなさい。(実際に拡張する必要はありませんが、実際に拡張して報告しても構いません。ただし、その場合でも、事前に時間は見積もってください。)

① クール便をオプションとして提供する。料金は以下の通り

サイズ	料金
60 サイズ	+216 円
80 サイズ	+216 円
100 サイズ	+324 円
120 サイズ	+648 円
140 サイズ	取扱不可
160 サイズ	取扱不可

② サイズの異なる複数の荷物を持ち込んだ人に、個々の配送料金と合計の配送料金を表示する。

- 先に提出したあなたのプログラム(完成していない場合は、あると仮定したプログラム)は、以下の 14 個の入力に対して、それぞれどのような出力を作りますか？ 実際にプログラムに適用し、出力結果を示しなさい。
- その出力結果は、あなたにとって満足できるものですか？ もし不十分と感じる箇所があれば、該当箇所を明示して、何故不十分なのか、その理由についても述べなさい。
- 先に提出したマニュアルに、書いておけばよかった(つまり記述が不十分だった)と思う事項はありますか？ もしあるのならば、その内容を記述しなさい。
- 先に提出したプログラムとマニュアルをあなた自身がユーザとして使うことを想定したとして、その満足度は何パーセントくらいになりますか？ また、そのパーセントを付けた理由を述べなさい。

入力 1	入力 2	入力 3	入力 4
縦: 36cm 横: 52cm 高さ: 64cm 重さ: 22kg	縦: 14cm 横: 42cm 高さ: 23cm 重さ: 8kg	縦: 24cm 横: 33cm 高さ: 63cm 重さ: 10kg	縦: 40cm 横: 50cm 高さ: 60cm 重さ: 30kg

1/2

入力 5	入力 6	入力 7	入力 8
縦: 50cm 横: 70cm 高さ: 60cm 重さ: 19kg	縦: 10cm 横: 5cm 高さ: 5cm 重さ: 0kg	縦: 10cm 横: 0cm 高さ: 20cm 重さ: 2kg	縦: 50cm 横: 50cm 高さ: -20cm 重さ: 4kg
入力 9	入力 10	入力 11	入力 12
縦: 40.3cm 横: 42.7cm 高さ: 41.1cm 重さ: 17.6kg	縦: 30.8cm 横: 30.5cm 高さ: 30.3cm 重さ: 10.2kg	縦: 20.35cm 横: 20.35cm 高さ: 20.35cm 重さ: 1.8kg	縦: x 横: y 高さ: h 重さ: m
入力 13	入力 14		
縦: 1cm 横: 1cm 高さ: 1cm 重さ: 25 kg	FIFA(国際サッカー連盟)公式球 5 個 <参考データ> ボール 1 個の 直径: 21.5cm 重さ: 450g		

※ 以下は、注意事項です。

- 提出締切は、**7/2(木) 13:00 (「ソフトウェア工学」授業開始前)**です。
- 提出は、授業の際に持参するか、締切までに A330 の郵便受けに入れるか、MAIL で提出しても構いません。
- レポートの試問を行う予定です。
- 前回のレポートをグループで提出した人は、そのグループで提出しても、サブグループで提出しても、個別に提出しても構いません。
- <http://earth.cs.miyazaki-u.ac.jp/~kat/lecture/SE/> に、この pdf ファイルを置いておきます。

2/2

図 6 2015 年度 2 回目のレポート課題

ロセス」「見積」「納期」について理解する。

- 課題 4. 工夫したポイントを書く
開発したソフトウェアの優れた点を依頼者に説明することの重要性を理解し、ソフトウェアを作成する際に工夫したポイントを書く。「設計」「実装」「ユーザ視点」について理解する。

仕様は、例年複雑なプログラムにならないような規模に設定している。学生が一瞬でやる気をなくすことがないように考慮しており、この匙加減は毎年苦労している。また、与える仕様は、ちゃんと書いてあるように見えて、実は、曖昧で、ヌケがある仕様である。これは、ユーザからの要求は基本的に不十分であり曖昧でもあることの理解に加えて、設計や実装の工程に入る前に、ユーザとのコミュニケーションが不可欠であり、この工程をおろそかにすると、作成したソフトウェアがユーザの期待を満たさず、ソフトウェア開発が失敗に終わる可能性が高くなることの理解を期待している。

マニュアル作成は、学生がどのくらいユーザのことを考慮しているかを見るための手段としては、良い課題だと個人的には考えている。マニュアル作成を通し

て、何のためのもの作りなのか、すなわち、何のためにソフトウェアを作るのかを学生には理解してもらいたいと考えている。

また、学生が課題を投げ出さないように、次の工夫をしている(図 5 の右下部分参照)。まず、レポートの締切は、課題の難易度より少し長めの期間(約 1 か月)に設定しており、「時間がありませんでした」という状況にはならないようにしている。さらに、仮にプログラムが完成しなくても、完成途中でのプログラムと完成したプログラムがあるとして作成したマニュアルを作成すれば受け取るようにしている。これは、この課題が、プログラミングの課題ではなく、ソフトウェア開発の課題であり、プログラミングはもちろん重要な項目ではあるが、それ以外の項目(要求の曖昧性やユーザ視点のマニュアルなど)の重要性について考察することによって、プログラミングの部分を補えばよい、と考えているからである。加えて、グループでの提出も認めている。ただし、グループでの提出の場合は、各人の責任範囲を明確にすることを条件にしている。

次に、2 回目の課題について説明する。以下、2 回

- 2回目のレポート提出後、試問期間に入る
- 試問を受けなかった場合、レポートの点を0点にする
- レポートに書いてあることに対して質問する = 書いてないことを引っ張り出して尋ねたりはしない
- 期間が長いよう見えますが、私が居ない時間帯が多いために、長く設けてるだけなので、各自なるべく早く対応することを望みます
- レポート課題のファイルを置いた Web ページに、私の都合を書き込みます
- 試問は試問期間しか実施しないため、終了時刻に並んでいても打ち切る
- 試問の結果、再試問やレポートの再提出がありえるが、それらも含めて試問期間で原則打ち切る
ので、自信があまりない者は、期限ギリギリに来ないように

図 7 「ソフトウェア工学」におけるレポートの試問のルール（学生に提示）

目のレポート課題の各項目と課題のポイントについて述べる。

- 課題 1. 自分が作ったプログラムを拡張する
先に提出したプログラムに機能拡張を行う。開発したソフトウェアをユーザに納品した後、そのソフトウェアの保守が必要だということの理解と、保守作業は最初に作成したプログラムに依存するため、設計の重要性についても理解する。「保守」「設計」について理解する。
- 課題 2., 3. 自分が作ったプログラムをテストし実行結果について考察する
先に提出したプログラムを、こちらが準備したテストデータでテストする。依頼されて作成したプログラムは作成者である自分ではなく、第三者であるユーザが使うものであるということを理解した上で、そのユーザに使ってもらえるレベルまでテストすることが重要であることと、要求分析の段階で、ユーザとのコミュニケーションが不可欠な要素であるということを理解する。「テスト・デバッグ」「要求分析」「ユーザ視点」について理解する。
- 課題 4. マニュアルを補足する
マニュアルの記述不足分があれば記述する。これまで課題を進めて（特に、上記課題 2., 3.）、マニュアルに不足があると感じた場合は追加事項を記述し、ユーザがどのような使い方をするかを想定することの重要性について理解する。「要求分析」「ユーザ視点」について理解する。
- 課題 5. 自分が作ったプログラムとマニュアルをユーザとして評価する
自分が作成したプログラムとマニュアルは、本

当に第三者が使って満足できるレベルに到達しているか、また作成段階で第三者を意識した作り方をしたのか、について最後に考察してもらう。「ユーザ視点」について理解する。

この2回目の課題は、1回目の課題を学生が提出した後に課している。これは、ソフトウェアはユーザへ納品したら仕事が終わるわけではなく、その後も保守が求められる仕事は継続するということを想定している。この年は、具体的な機能拡張として、クール便への対応と複数の荷物への対応を課した。

この年準備したテストデータは、全部で14個だった。最初の方は、ソフトウェアにとって素直な入力、すなわち作成者が入力して欲しいと考える素直なデータから成っている。中盤から、入力値に小数がある、0がある。マイナスの数がある、文字がある、のように、ソフトウェアにとって、いわゆる「意地悪な入力」を準備した。このようなデータは、ユーザが悪意を持って入力する場合もあるが、うっかり誤って入力してしまう場合もあり、製品レベルのソフトウェアでは、何らかの対処が必要となる。ソフトウェアを使うユーザのことを考えて、このような対処をプログラム作成時に考慮する必要があるということと、このような入力の場合、どのようにソフトウェアが振舞うべきかを要求分析の際に考慮する必要があるということの理解のために設定している。

ちなみに、この年の入力13は、「縦1 cm, 横1 cm, 高さ1 cm, 重さ25 kg」と、地球上には存在しない物質を指定してみた。この入力に対し、2名だけが、「この入力に対して、正しい出力は何だったのか」と2回目の課題で考察していたが、残りの50名以上が、「1,836円と出力しているので正しい」と記述してい

表1 外部講師による講演

	講演タイトル	講師	所属
H22	関数プログラミング入門	山本 和彦	IJ
H23	情報産業への就職を目指す人へ	大力 修	新日鉄ソリューションズ
H24	テスト項目数を大幅に削減する「組合せテスト」の効果的活用方法	石原 一宏	バルテス
H25	要求工学概説	鈴木 三紀夫	MRT コンサルティング
H26	ソフトウェアテストにおける現場のトレンド	東 大輔	日本ノーベル
H27	コンピューター品質の未来	細川 宣啓	日本 IBM
H28	いま求められる論理的な思考とコミュニケーションの力とは?	栗田 太郎	ソニー FeliCa 事業部
H29	人工知能技術を利用したアプリケーションの開発とテスト	増田 聡	日本 IBM
H30	Smart Speaker QA	松谷 峰生	LINE Fukuoka

た(正確には、ほとんどの学生が「入力13の出力結果については問題がない」と記述)。この例に限らず、試問の際に、学生の記述に対してどこが問題なのか、を提示することで、学生には自分の記述に問題があることに気づいてもらった上で、その問題への対処についても考えてもらうようにしている。なお、4.(1)節で書いたように、この科目の教育目標は、「ソフトウェアを開発できる」ではなく、「ソフトウェア開発の際に起こりうる問題について把握する」としているため、こういう問題が起こりうるということを理解させて、合格点を出している。

(5) レポートの試問の実施

レポートの試問の実施について述べる。試問のルールを、図7に示す。これは、学生にも提示している。試問を受けなかった場合は、レポートの評点を0点にする、としており、学生は必ず試問を受けなければ、単位が取れないように設定している。レポートの試問にかかる時間は、1人(グループ)短くて15分、長い場合は、40分以上かかり、回答できない場合は、再試問も行う。このため、試問期間は、例年3週間ほどを設定している(2018年度は7/5(木)-25(水)に設定した)。この期間に私の居室に来てもらい、試問を受けてもらう。なお、期間は3週間あるのだが、私の都合、すなわち、この授業以外にも担当している授業はあるし、会議や来客、出張などで対応できない時間帯は当然ありうるので、試問を対応可能な時間帯は多いわけではない。この期間の私のスケジュールは、レポート課題をダウンロードできるWebサイトから見られるように、随時更新をしている。

学生からこの試問の時間をみると、15分から40分、これに準備(対策)時間が加わるくらいであるが、こ

ちらがこの試問にかかる時間は、1人平均25分×約60人=1,500分となり、だいたい講義1回分(90分授業×16コマ=1,440分)を使っていることになるため、実施する側の負荷は少なくはない。

試問をする目的は、以下の通りである。

- 自分が作ったものに責任を持たせる
- 学生の「分かったつもり」を問題として指摘する

学生にとって、レポート課題は、「提出してしまえば、それで終わり」と思っているところがある。しかし、一般の工業製品はほぼすべて、ユーザに納品したら、それで終わり、とはならず、運用・保守の作業が必ず発生する。特に、ソフトウェアについては、テスト工程でバグが取り切れず、納品後、ユーザが使用時にバグが発生することがよくあることと、実際にユーザが使った後にユーザから問題点を指摘され、初めてユーザの要求が分かることがよくあることから、保守にかかる作業は多い。このため、提出物について責任が続くということを自覚してもらう。

また、一般的に、学生はレポート提出した後に、合格点でない場合は、その理由を尋ねに来る学生はいるものの、合格点を出した場合は、レポートの評価について尋ねに来る学生はほとんどいない。点数が70点であれば、なぜ70点なのかを学生に指摘しない限り、学生の理解は進まない。そのため、試問を実施し、レポートに問題点があれば、それを指摘し、理解してもらうようにしている。4.(4)節でも書いたが、この科目の教育目標は、「ソフトウェアを開発できる」ではなく、「ソフトウェア開発の際に起こりうる問題について把握する」としているため、指摘した問題点を理解させて、合格点を出している。

(6) その他の工夫

その他の工夫として、外部講師による講演を1回実施している。表1に、実施した講演の、講演者と講演タイトルについて示す。

講演者は、企業の方をお願いしている。この理由は、ソフトウェア開発について教科書的に学ぶことは少ないからであって、ソフトウェア開発現場で働く人の生の声を聞くことによって、その姿勢や考え方を直接学生に学んでもらうことの方が重要だと考えているからである。加えて、学生に将来自分が働いている姿を想像して欲しいことも、理由である。

おわりに

本稿では、筆者が担当する「ソフトウェア工学」の授業方法について述べた。「ソフトウェア工学」は、現在のソフトウェアが身の回りにあふれている状況において、重要な技術であり学問であると言える。しかし、その誕生からは約50年しか経過しておらず、学問としては未だ成熟していない。

このため、他の講義とは状況が異なるので、今回述べた各種の工夫がそのまますべて他の講義に適用できることはないかもしれない。しかし、各先生方にとって、本稿が一部でも何かヒントになれば、幸いである。

文献

河村一樹, 2003, 『ソフトウェア工学入門改訂新版』, 近代科学社.

小泉寿男・辻秀一, 2015, 『ソフトウェア開発 (改訂2版)』, オーム社.

ウィキペディア, 『ソフトウェア危機』

(<https://ja.wikipedia.org/wiki/ソフトウェア危機>)

(2019年1月4日).

ICSE (International Conference on Software Engineering),

(<http://www.icse-conferences.org/>) (2019年1月4日).