Web アプリケーション開発による
プログラミング教育カリキュラム

環境情報学部 4 年　青山　希

概要

情報技術の発展に伴い、ソフトウェアの需要は増え続けている。そのため社会ではどのようなソフトウェアがひとに使ってもらえるのかを理解し、新たなものを創り出すための考え方をもった人材が求められている。

本当に人の役に立つソフトウェアを作るための考え方や新たなものを創り出すための考え方は、誰かに使ってもらうためのソフトウェアをつくっていく中でしか身につかない。社会が求める人材を育成するためには、そのようなより実践的なソフトウェアの開発を行うカリキュラムが必要である。

慶応義塾大学大岩研究室では、そのような社会が求める人材を育成するため、Web アプリケーションの開発を題材としたカリキュラムを開発してきた。カリキュラムはグループ開発、UML によるコミュニケーション、他人の要求の実装など、多くの概念を取り入れ、より実践な形でソフトウェア開発の過程を学ぶものである。以下は 2002 年度研究プロジェクト科目において実施されたカリキュラムとその評価である。

1. はじめに

1.1. 背景

現在、情報技術の発展にともない、ソフトウェアによって解決できる問題の範囲が広がり、ソフトウェアに対する需要は増える一方である。一方で、本当に使いやすく、便利なソフトウェアをつくるには、ソフトウェアを開発する側にも、ソフトウェアを導入して業務を改善しようと考える側にも、ソフトウェアに関する理解と、新たなものを創り出すための考え方を持った人材が必要であり、そのような人材は常に不足しがちである。

1.2. 大学に求められる役割

そのような社会が求める人材を育成することは大学に求められる役割である。現在の大学におけるプログラミング教育は、プログラミングの初歩的を学ぶ中で論理的な考え方を養うことを目的としている。しかし、そこで扱われるプログラムは文法や考え方などを学ぶためのプログラムであり、実際に誰かに使ってもらうためのものではない。

新たなものを創り出すための考え方や、本当に人の役に立つソフトウェアを作るための考え方は、誰かに使ってもらうためのソフトウェアをつくっていくなかでしか身につかない。社会が求める人材を育成するためには、そのようなより実践的なソフトウェアの開発を行うカリキュラムが必要である。

2. カリキュラムのコンセプト

2.1. 人を幸せにするソフトウェア

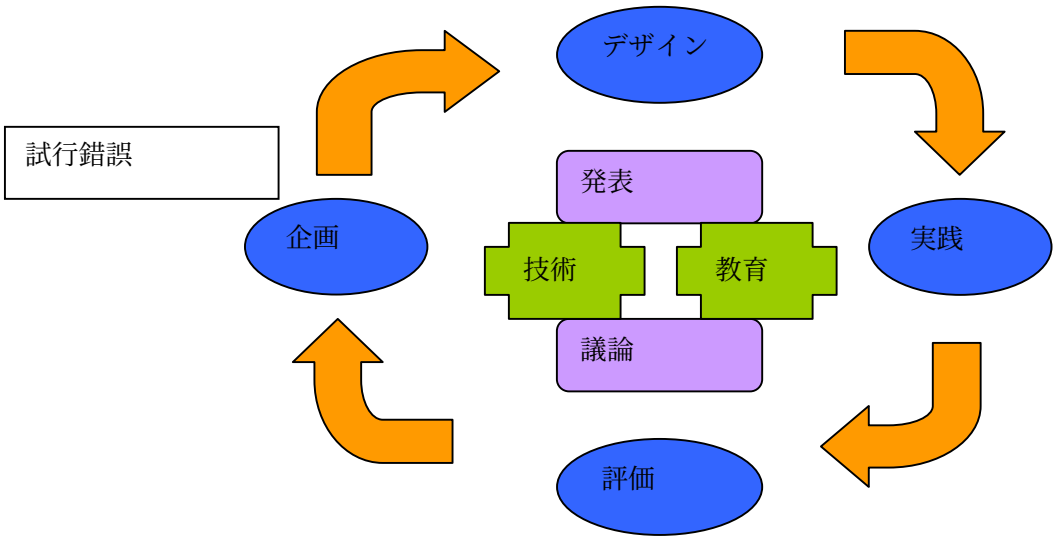
カリキュラムでは、人がそのソフトウェアを使ってなにか一つでも目的を達成することができるような、人が使ってくれるソフトウェアをつくることを目的とする。持続的に人に使ってもらえるソフトウェアをつくるには、

- 使いやすいユーザインターフェース
- ユーザの目的を達成する機能
- 保守管理のしやすさ
- ユーザがストレスを感じない実行速度

など多くの要素が必要である。カリキュラムではこのようなソフトウェアを“ 人を幸せにするソフトウェア ”と定義し、どうすれば人を幸せにするソフトウェアをつくることができるかについて考えてもらう。

2.2. ものを作っていく過程

新たなものをつくっていく課程には、以下の図のような企画・デザイン・実践・評価の一連の流れを、試行錯誤を繰り返しながら反復させていくことが必要である。またそれら一連の流れの中で、常に必要になるのは発表と議論であり、同時に、技術や教育といった要素も求められるようになる。このようなものづくりの過程は、ソフトウェアの開発に特化したものではなく、何か新しいものを作るときには必要なものである。カリキュラムでは授業の中でこの反復を実際にやってもらい。ものづくりに必要な考え方を養ってもらう。



3. カリキュラムの実装

3.1. 実施形態

慶應義塾大学大岩研究室では、上に述べられたコンセプトに則ってカリキュラムを作成した。カリキュラムは 2002 年度春・秋学期の研究プロジェクト科目として実施された。

3.2. 題材

カリキュラムでは春・秋学期を通して、Web アプリケーションの開発を題材として、人を幸せにするソフトウェアと、その開発の過程について履修者に学んでもらった。Web アプリケーションを題材とした理由を以下に挙げる。

- 実装が比較的容易で、それほど技術がなくても社会で実際に使われているものに近いものが作れる
- 企画次第で人の役に立つソフトウェアをつくることができる
- 見た目が派手なものをつくりやすいので受講者の興味をひきつけることができる
- 受講者が実装したソフトウェアを人に見せたり、使ってもらったりしてフィードバックを得ることが容易である

3.3. カリキュラムの基本的な進め方

授業は基本的に春・秋学期共に、1 学期かけて受講者が企画した一つのソフトウェアを開発していく形で行われた。授業は、受講者それぞれの企画に対する議論の時間と、開発を進めていく上で必要な技術的なトピックスを学ぶ時間の 2 部で構成された。受講者は毎週、企画、分析、設計、開発のそれぞれの段階での進捗報告を行い、それに対して問題点などを全員で議論する。また、技術的なフォローの時間では、その時々のテーマに応じた講義を受け、最後に実習を行った。

3.4. 実施内容

3.4.1. 春学期

テーマ	Web アプリケーションの開発
前提知識	Java の基本的な文法が書ける アルゴリズムを自分で考えて組むことができる メソッドを使うことができる クラス、オブジェクトの概念が理解できている
目標	人を幸せにするという観点で、ソフトウェアを開発することができる Web アプリケーションの基礎が理解できる
習得すべき技術や考え方	HTTP プロトコルによる通信の仕組み Servlet を使ったサーバーサイドのプログラミング リレーショナルデータベースの基礎
評価の方法	開発した Web アプリケーション レポート

(1)**進め方**

人を幸せにするソフトウェアをつくるためには、まずソフトウェアでなにができるのかを理解する必要がある。春学期のカリキュラムでは、まず自分たちがこれからつくろうとしている Web アプリケーションとはどのようなもので、どのようなことができるのかを、Web アプリケーションの作り方を学びながら感じてもらう。

具体的には受講者にそれぞれ自分たちで考えた、Web アプリケーションの企画をそれぞれ人を幸せにするということを考えながら実装していく。春学期の段階では、まず何ができるのかを理解し、実装できる程度の技術的な足並みをそろえることを主眼とする。分析や設計などのソフトウェア開発の進め方や、人を幸せにするソフトウェアに必要な、保守管理のできるソースコードや、実行速度などについては言及しない。

(2)**具体的なカリキュラム**

第 1 回

□企画に関する議論・議論

- なし

□必要な技術や考え方の講義

- 人を幸せにするソフトウェアとは何かについて受講者に考えてもらい、どのような考え方が必要かを感じる。

第 2 回

□企画に関する発表・議論

- なし

□必要な技術や考え方の講義

- Web アプリケーションとは何か、具体例をあげ、どのようなものが人を幸せにするソフトウェアとなりうるのかを理解する。
- 今後企画を立て、実装していくに当たりどんなものをどんなソフトウェアを作ることができるのかを感じる。

第 3 回

□企画に関する発表・議論

- 社会に存在する Web アプリケーションの実例を各自調査してきて発表する。
  受講者人を幸せにする Web アプリケーションのイメージがかたまっているかを確認する。

□必要な技術や考え方の講義

- Web アプリケーションの開発に必要な HTTP プロトコルによる通信のメカニズムについて学ぶ。

第 4 回

□企画に関する発表・議論

- なし

□必要な技術や考え方の講義

- Servlet によるサーバサードプログラミングの初歩について
- アプリケーションサーバのインストールと簡単なテストプログラムを実行し、Servlet を使ったプログラミングに慣れる。

第 5 回

□企画に関する発表・議論

- なし

□必要な技術や考え方の講義

- アプリケーションサーバの構成と、その概念について。Web アプリケーションとはどの

ような概念で構成されているのか、HTTP プロトコルについて復習しながら学ぶ。

第 6 回
□企画に関する発表・議論
- なし

□必要な技術や考え方の講義
- HTML フォームを使った Web アプリケーションの作り方を学ぶ。HTTP プロトコルを使って入力データを送受信するプログラムが書けるようになる。

第 7 回
□企画に関する発表・議論
- なし

□必要な技術や考え方の講義
- ストリームの概念を学び、ファイルの入出力を使ったプログラムが書けるようになる。
- ここまでで掲示板のような基本的な Web アプリケーションを実装できるようになる。

第 8 回
□企画に関する発表・議論
- なし

□必要な技術や考え方の講義
- セッション管理の考え方を学びセッションを使った Web アプリケーションが作れるようになる。

第 9 回
□企画に関する発表・議論
- 企画発表会。これまでの議論や講義などを踏まえ、それぞれが考えた人を幸せにする Web アプリケーションについての発表を行う。

□必要な技術や考え方の講義
- なし

第 10 回
□企画に関する発表・議論
- 進捗報告。実装の進み具合やそこで起きた疑問などについて発表し、議論する。

□必要な技術や考え方の講義
- データベース入門。データベースとはどのようなものなのかを学び、基本的な SQL が使えるようになる。

第 11 回
□企画に関する発表・議論
- 企画した Web アプリケーションの画面遷移図を作成し、発表する。

□必要な技術や考え方の講義
- JDBC とインターフェースについて。プログラムはデータベースをどのように利用するのかを学び。インターフェースによる抽象化の概念に触れる。

第 12 回
□企画に関する発表・議論
- 進捗報告

□必要な技術や考え方の講義
- リレーショナルデータベースの考え方について。データベースを利用したプログラムを書くのに必要なテーブルなどの概念について学ぶ

第 13 回
□企画に関する発表・議論
- 進捗報告

□必要な技術や考え方の講義
- なし

第 14 回
□企画に関する発表・議論
- 最終発表会。それぞれが 1 学期かけて作ってきた Web アプリケーションを発表する。

□必要な技術や考え方の講義
- なし

3.4.2. **秋学期**

テーマ	Web アプリケーションのグループ開発
前提知識	・春学期に学習した内容
目標	➢ 企画、分析、設計、実装というプロセスを反復するという考え方を理解できる ➢ 適切なコミュニケーションの手段を使ってグループでの開発ができる
習得すべき技術や考え方	➢ ユースケースの概念 ➢ UML(ユースケース図、クラス図、シーケンス図、コラボレーション図、ステートチャート図) ➢ 継承の概念 ➢ 外部のライブラリの利用 ➢ ファイル共有などのグループ開発のための概念 ➢ MVC モデルアーキテクチャ
評価の方法	➢ 開発した Web アプリケーション ➢ UML テスト ➢ 相互評価シート ➢ レポート

(1)進め方

秋学期は、春学期に身に付けた Web アプリケーション開発のための基本的な技術や、概念を利用して、より実践に近い形で、開発を行う。より実践に近い形で、開発のプロセスや、開発に必要なコミュニケーションを学ぶために秋学期では以下の二つの新たな概念を導入した。

★ グループワークによる開発

UML を使ったコミュニケーションや、わかりやすいソースコードは、人を幸せにするソフトウェアを開発するためには非常に重要なことである。しかし、一人で開発をしていてはそのようなことの大切さは実感することができない。そのようなことの必要性を実感し、より実践的な開発を体験するために秋学期のカリキュラムではグループでの開発を取り入れた。

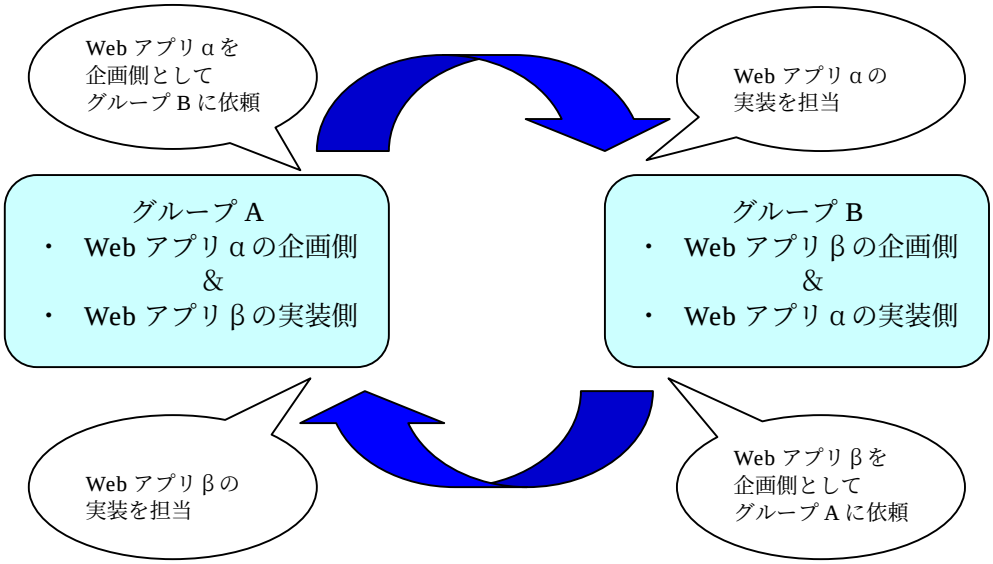
★ Web 教材の予習＆確認テストによる UML の学習

企画・設計・実装の各段階でのコミュニケーションのための記法として UML を学習する。しかし、UML をつかったコミュニケーションを身に付けるには、実際の開発

の中での経験が必要不可欠である。そこで、授業では Web 教材の予習と、確認テストを行うのみとし、それ以外は履修者に開発のアウトプットとして発表させ、そのレビューを行う。

★ ソースコードレビュー
　春学期のカリキュラムでは、まず Web アプリケーションとはどのようなものかを理解することを主眼とし、保守管理や要求の追加に耐えられるような、設計やソースコードについては、ほとんど言及してこなかった。しかし、実際に使われるソフトウェアを作るには、そのようなことは必要不可欠である。秋学期のカリキュラムでは、ソースコードのレビューを行い、読みやすく、保守管理のしやすいソースコードについても学んでもらう。

★ 企画・実装を他のグループとのインタラクションを行う
　グループはそれぞれ 1 つの Web アプリを企画し、その実装を他のグループに依頼する。同時に、グループは他のグループが企画した Web アプリの実装を担当する



図のようなやり方で開発を行っていくにあたり、それぞれの班の役割は以下のように定義した。

■ **企画班**
- ✧ **企画そのものに対する決定権を持つ**
- ✧ **ユースケースのたたき台を作る**
- ✧ **ユースケースに対する議論をする**
- ✧ **実装班が作成する概念モデルに対するフィードバックを行う**
- ✧ **実装班が提出するプロトタイプに対するフィードバックを行う**
- ✧ **最終発表にむけて適宜インタビュー**

■ **実装班**
- ✧ **企画そのものにはアドバイスや交渉はできるが、決定権はない**
- ✧ **UC に対する議論をする**
- ✧ **概念モデルの作成&手直し**
- ✧ **クリティカルな機能に対するプロトタイプを実装**
- ✧ **最終発表にむけて適宜インタビュー**
- ✧ **最終発表として企画の実装を行う**

(2)**具体的な授業計画**

第 1 回

- 授業の進め方に対する説明
- 次回に行われる UML テストに対する説明を行う

第 2 回
□企画に関する発表・議論
- 全員がひとつづつ企画を発表する
- 企画の中からいくつかの企画を選びその企画を担当するグループを作る

□必要な技術や考え方の講義
- ユースケースに関する確認テストとテストの解説を行う

第 3 回
□企画に関する発表・議論
- 前回決めた企画に対してグループ内で議論を重ね、企画内容をかためてそれぞれ発表を行う
- 企画に対してその実装を担当するグループを決める
- 決定した企画＆実装のグループ間で第 5 回に行われるユースケース発表に向けて議論をそれぞれ議論を進めてもらう

□必要な技術や考え方の講義
- クラス図と概念モデルに関する確認テストとテストの解説を行う

第 4 回
□企画に関する発表・議論
- なし

□必要な技術や考え方の講義
- 動的モデルに関する確認テストとテストの解説を行う

第 5 回
□企画に関する発表・議論
- ユースケース発表会。企画、実装のそれぞれの立場からどのような議論が行われたかを発表し、全員で疑問点について議論を行う。

□必要な技術や考え方の講義
- なし

第 6 回
□企画に関する発表・議論
- 概念モデル発表と今週行われた議論についての発表を行う。ユースケースに基づいた概念モデルがどのような議論を経て作成されたかを発表し、疑問点について全員で議論する

□必要な技術や考え方の講義
- 継承についての講義を行う。今まで何気なく使ってきた継承という概念についての知識を固め、どのような場面で継承を活用するべきかについて考える。

第 7 回

□企画に関する発表・議論

- 前回に引き続き概念モデル発表と今週行われた議論についての発表を行う。前回の議論や指摘を受けどのような議論をチーム内で重ね、概念モデルを変更したかについて発表をする

□必要な技術や考え方の講義

- 汎用的なリストについての講義。汎用的なリストの使い方、作り方を学び、外部に用意された Java の API がどのような概念で構成されているのかを理解する。

第 8 回

□企画に関する発表・議論

- 第 10 回に予定されているプロトタイプ発表に向けて、実装側にプロトタイプの指針を発表してもらう。
- 発表に対して全員で疑問点などを指摘し、議論する。
- プロトタイプは企画側を安心させるためにシステムにとってクリティカルな機能を実装する必要があることを議論の中で確認する。

□必要な技術や考え方の講義

- グループ開発におけるファイル共有の考え方を学ぶ
- CVS が使えるようになる

第 9 回

□企画に関する発表・議論

- 前回に引き続きプロトタイプの指針を発表し、どのような議論が行われているのかを発表する。
- 発表に対して全員で疑問点などを指摘し、議論する。

□必要な技術や考え方の講義

- クラス設計におけるモデルの分離について。
- モデルを分離する意義について理解し、実際に応用できるようになる

第 10 回

□企画に関する発表・議論

- プロトタイプの発表
- 発表に対して全員で疑問点を指摘し、議論する

□必要な技術や考え方の講義

- なし

第 11 回

□企画に関する発表・議論

- それぞれの企画についてソースコードのレビューを行う。プロトタイプ時点のソースコードについてレビューを行い、改善の方針を提示する

□必要な技術や考え方の講義

- クラス設計におけるビューの分離について
- ビューを分離する意義について理解し、実際に応用できるようになる

第 12 回

□企画に関する発表・議論

- 前回のソースコードレビューをうけどのような議論がなされ、どのような変更を行ったのかについて発表する

□必要な技術や考え方の講義

- なし

第 13 回
□企画に関する発表・議論
- 最終発表に向けての進捗報告を行う
- 疑問点や困っている技術的な問題などがあれば質問をする

□必要な技術や考え方の講義
- なし

第 14 回
□企画に関する発表・議論
- 最終発表

□必要な技術や考え方の講義
- なし

---

4. カリキュラムの評価

4.1. アンケートの実施

カリキュラムの評価を行うにあたり、受講者にカリキュラムについてのアンケートを行った。質問の内容は以下のようなものである。

□カリキュラム全体について 設問 1 この研究会を通じてソフトウェアに対するイメージはどのように変化しましたか？ 変化がなければなかったと書いてください 設問 2 この研究会では、企画、設計、開発、評価というものをつくっていく過程を学んで いきましたが、みなさんの中でもの作りに対する姿勢はどのように変化しましたか？
□春学期について 設問 3 授業（サブゼミ、講義）の難易度はどうでしたか？ 設問 4. サブゼミでわからなかった点、ほかにやってほしかったことなどあったら 書いてください 設問 5 研究会を通じてどのようなことが身についたと思いますか? また、どのようなことが身につかなかったと思いますか？ 設問 6 なにか要望、感想などあったら自由に書いてください
□秋学期について 設問 7 授業の難易度はどうでしたか？ 設問 8 サブゼミについてわからない点、ほかにやってほしかったことなどあったら 書いてください

設問 9
　研究会のなかでどのようなことが身についたと思いますか？
　またどのようなことが身につかなかった と思いますか？
設問 10
　今学期は自分の企画ではなく、人がたてた企画を実装してもらいました。
　それについてはどう思いますか？
設問 11
　UML の講義を予習とテストの解説だけにして、あとは実際に使ってもらう形で
　学んでもらいましたが、使っていく中でわからなかった点などあったら書いてください
設問 12
　感想、要望などあったら自由に書いてください

4.2. 考察

4.2.1. 「人を幸せにするソフトウェア」について

今回のカリキュラムでは、開発の過程で「人を幸せにするソフトウェア」という観点でソフトウェアを分析、設計実装をしていくように授業の中で言ってきた。アンケートの結果や受講者に話を聞いた中でも、彼らが今までプログラミングの入門教育の中でつくってきた“ 動く ”ことを一番の目的としたプログラムでは人に使ってもらうことはできず、人に使ってもらい、人を幸せにするソフトウェアを作るためには多くのことに注意していかなければならないということはある程度受講者に浸透したように思う。

しかし、今年度の反省点として、受講者が「人を幸せにするソフトウェア」の具体的なイメージをつかむことが出来なかったことが挙げられる。受講者のアンケートを見ると「必要なのは設計である」というような意見が多く見られる。人を幸せにするソフトウェアをつくるために、設計が重要であることは確かである。しかし、最も大切なのは、「ユーザの目的を達成するため」、「保守管理をしやすくするため」「機能の追加を容易にするため」といった設計を行う目的である。

今後カリキュラムを改善していく上では、どのようなソフトウェアが人を幸せにするソフトウェアなのかという、具体的な手本を提示し、受講者が目指す具体的な目標を設定することが必要である。

4.2.2. 「ものづくりの過程」について

カリキュラムでは前述のような企画・分析・設計・実装の一連の流れを反復させていくことがものづくりの過程において非常に重要であるという考え方を受講者に養ってもらうことを目的のひとつとした。受講者のアンケートを見てみると、“ ソフトウェア開発＝実装(コーディング) ”というような考え方から企画から設計、実装という一連の流れで開発が成り立っているという考えに変わってきたという多く見られる。これは受講者の意識の中に、企画や、設計といった新たな概念を植え付けることができたという点では評価できる。

しかし、それだけではカリキュラムの目的とする考え方は、まだ完全に伝わっているとは言い難い。まず第一に受講者が「分析」というフェーズを具体的にイメージすることが出来なかったという点が挙げられる。アンケートを見ても分析という言葉はまったく出てこない。設計の前段階としてシステムに関連する概念を分析することは、カリキュラムの中でも繰り返し行ってきた。しかし、その段階でやるべきことの定義や、動機付けが不十分であったために、受講者が具体的なイメージをもつことができなかったのではないかと思われる。この点は今後検討すべき課題である。

もうひとつの問題点は、受講者の中で、ここまでで述べたプロセスは反復させていくものであるという認識をもたせることが出来なかった点である。企画から実装までの開発の流れの後、評価を行い、それをもとに次のイテレーションに生かすということを、カリキュラムの中で実際に

受講者に体験させることができなかったことが、原因であると思われる。来年度のカリキュラムでは、実装が一通り終わった後、新たな要求を追加するなど、反復という概念を導入することが必要である。

4.2.3. 技術的な知識のフォロー

カリキュラムの中では技術的なフォローのために、春学期は主に Web アプリ作成のための技術や考え方、秋学期は UML や設計についての概念を扱った。難易度については今回のカリキュラムのレベルが受講者のレベルに適したものであったことがアンケートの結果からも、見て取ることができる。

今回のカリキュラムでは、基本的に技術的な知識は最低限にとどめ、受講者が企画を実装するために必要な最低限の事柄を取り扱った。これは使わない技術や考え方を教えても、その場限りの表面的な理解にとどまり、実際に使えるようにならないためである。しかし逆に、今年度のカリキュラムでは、その場で使う技術や知識だけを履修者が学び、その背景にあるソフトウェアの本質的な考えかたについて十分理解しているとはいえない。その点はサブゼミに対する意見や感想が、Java についての知識や、Web アプリを実装するための知識についての意見しか述べられていない点からも明らかである。今後、カリキュラムを改善していくにあたり、講義や実習を背景にある考え方をより深く学ぶことができるようなものに変えていく必要がある。

4.2.4. UML

秋学期のカリキュラムでは、分析、設計のための標準的な記法である UML 用いた設計・分析の手法をカリキュラムの中に取り入れた。UML は記法だけを学んでも実際に UML を使って設計や実装の議論ができるようにはならない。そのため今回のカリキュラムでは、UML の学習を Web による予習用の教材と、知識の確認テストだけにとどめ、あとは発表や議論の場でその使い方をレビューしていく方法をとった。このような方法をとることで履修者はグループワークの中で試行錯誤しながら、UML 用いた設計・分析の手法を身につけていくことができた。

4.2.5. グループ開発

コミュニケーションの必要性を理解し、UML の実践ができたという点で、グループ開発を導入した狙いはある程度達成できた。アンケートではグループ開発を行ううえでの障害や、マネージメントの重要性についての意見や感想を述べた履修者も多く、より実践的な開発を履修者が体験することができた。

しかし、グループ開発において履修者それぞれが一定の仕事を担当し、全員がカリキュラムを有意義なものとするには、それぞれが企画、分析、実装、それぞれの面である程度の力をもっていることが前提条件である。その点で春学期にまず一人で開発ができるようになってからグループ開発を導入するという今回の進め方は適当であったと思われる。

4.2.6. 人の企画を実装する

人の企画を実装するという新たな試みには、履修者のアンケートを見てもある程度好意的な意見が多い。開発におけるコミュニケーションを学ぶという点で有効な方式であったと思われる。

その中でも注目したいのは、自分で立てた企画と違って、要求がはっきりとあるからいい意味で妥協ができないという意見である。これは、実際の社会におけるソフトウェア開発では当然のことである。しかし、一般的なプログラミング教育の自分の立てた企画や、与えられた課題を実装する実習では体験することができない。いい意味で妥協ができないために、新たな技術を導入する必要がでたり、ソフトウェアの質が求められたりすることは、人を幸せにするソフトウェアとはどのようなものかであるかを学ぶという、カリキュラムのコンセプトを達成するために非常に重要なことである。

4.2.7. ソースコードレビュー

わかりやすいソースコードを書くことは保守管理や、機能拡張がしやすいソフトウェアを作るために最も重要なことのひとつである。そのようなグループの中だけで開発を行っていると見過ごされがちなことをソースコードのレビューを行うことでよい方向に向かわせることができた。

また、ソースコードをレビューすることで履修者が抱える実装上の勘違いや、悪い癖などを指摘することができ、非常に有用であったと思われる。

5. 終わりに

このカリキュラムのコンセプトである人を幸せにするソフトウェアと、ものづくりの過程についての考え方は、今後どれだけ情報技術が進歩しても変わることのない普遍的なものである。このカリキュラムはその普遍的な考え方を、Web アプリケーションといういわば流行の技術を題材として教えるものである。

普遍的な考え方や、情報技術の基礎は、履修者にとっては非常につまらないものである。また、その重要性も、プログラミングの初心者には理解することができない。そのため、それだけをとって履修者に教えようとしてもモチベーションをあげることは非常に難しい。そのような問題を解決するため、カリキュラムでは Web アプリケーションという履修者にとって身近で、魅力的な題材を用いて、その作り方を学ぶ中で普遍的な考え方や情報技術の基礎が身につくことを心がけた。

今回のカリキュラムによって Web アプリケーションや、UML など流行の技術にあたる部分については、履修者はこちらが想定した以上の力を身につけることができた。しかし、本当に教えたい普遍的な考え方や情報技術の基礎的な力を履修者が十分に身につけることができたかというと、評価の章にもあるとおりそれは十分なものとは言えない。

今後カリキュラムを改善していくに当たり最大の課題は流行の技術を学ぶ中でいかにしてその背景にある普遍的な考え方や、情報技術の基礎を身につけさせるかということである。それができることで初めてこのカリキュラムの目的である、人の役に立つソフトウェアとはどのようなものか理解し、さらにそれを新たに生み出すための過程を身につけた人材を育成することができるのである。

6. 付録

(1)アンケート

□カリキュラム全体について

設問 1

この研究会を通じてソフトウェアに対するイメージはどのように変化しましたか？
変化がなければなかったと書いてください

- ✓ 考えてみましたが、あまり変わったような気はしませんでした。
- ✓ 複数のメンバーで仕事を分担してソフトウェアを作っていくと、意思疎通をしているつもりでも自分の制作した部分以外は
- ✓ 軽い理解にとどまる場合が多い。このようなことから、大人数で作ったソフトウェアにバグは多いとうなずけるような気が する。
- ✓ ユーザビリティについて深く考えるようになった。ユーザの要求がソフトウェアに具現するまでの過程がイメージできるよ うになったことで、ソフトウェア開発の全体像が見えるようになった。
- ✓ もう少し企画班との連携がうまくいけば理想的なものができるかもしれない。
- ✓ プログラミングの中心はコードを書くことではなく、設計をすることだというのがよく分かった。
- ✓ 企画する側が見えてきた。
- ✓ 前がとってもいいかげんに作っていたので、モデルがとっても大切なことを学びました。そしてキレイに動くと感動。
- ✓ 特に・・・・。
- ✓ なかった

設問 2

この研究会では、企画、設計、開発、評価というものをつくっていく過程を学んでいきましたが、みなさんの中でもの作りに対する姿勢はどのように変化しましたか？

- ✓ 今までは開発が一番大事だと思っていましたが、実は設計が大事で、設計がどうしようもないと開発も適当になってしまうことが良く分かりました。逆に設計に時間をかけ、良い設計ができれば、開発もスムーズに進むことがわかりました。
- ✓ 技術を使ってもの作りを行うが、技術に振り回されてはいけない。知っている技術のみでもの作りをする、という姿勢から、もの作りをするためには新しい技術を使うことも怠ってはいけない！という姿勢に変わりつつある。
- ✓ 以前は、技術的な問題以外でソフトウェア開発になんとなく行き詰まったとき、次に何をすればよいか分からないときがあった。今は、「人を幸せにするものを作る」という作業の全体が見えたことで、問題点の所在に当たりをつけられるようになった。
- ✓ 特に設計が楽しいことがわかりました。実装では、設計でわからなかった部分が出てくることもあり、まだまだ設計が不十分であったと思いました。
- ✓ 方法論の重要さが分かってきた。今までは経験則に頼りすぎていたが、きちんとした方法論を学ぶことでよりよいもの作 りができるというのが実感できた。
- ✓ 設計、企画がその後に及ぼす影響が大きいことに気づいた。
- ✓ やっぱり議論は大切だなと。一人でやるだけのもの作りが多かったのでみんなと考えるのは貴重でした。
- ✓ 企画と設計が思いのほか重要なことがわかった。
- ✓ 役に立つものをオリジナルの考え方でつくれることは有意義であると思う。

□春学期について

設問３
授業（サブゼミ、講義）の難易度はどうでしたか？

- ✓ 難しすぎてついていけなかった 0 人 （0%）
- ✓ 難しいがなんとかついていけた 2 人 （22%）
- ✓ ちょうどよい 6 人 （67%）
- ✓ やさしかったが得るものはあった 1 人 （11%）
- ✓ やさしすぎた 0 人 （0%）

設問 ４.
サブゼミでわからなかった点、ほかにやってほしかったことなどあったら
書いてください

- ✓ サーブレットだけをやるのではなく、JSP もやってほしかったと思います。
- ✓ 特に記憶にないです。
- ✓ 実際にどのような点でつまずいたり難しいかなども教えて欲しかった。
- ✓ 秋学期ではなく、春学期のうちにコレクションなどを教えてもらうと後々楽だった気がする。また、コード記述規則の例などを教えてもらって、きれいなコードを書く習慣を早めに身に付けられればよかったとも思う。
- ✓ static とかそういう概念

設問 5
研究会を通じてどのようなことが身についたと思いますか?
また、どのようなことが身につかなかったと思いますか？

- ✓ サーブレットというものが存在することが分かりました。
- ✓ サーブレットの実装方法を一通り学ぶことができました。まだ、モデル、コントロール、ビューの実装の仕方などまでは身につけることができなかった。
- ✓ ある程度の規模のソフトウエアを動かすためには、どういったことに気をつけるべきかを身をもって経験できた。
- ✓ 幸せにするというコンセプトとサーバサイド Java 技術の概要。詳細技術。
- ✓ Servlet の作り方を学びました。自分にしてみればこれだけでかなり進歩。
- ✓ グループで共同作業することが難しく、マネジメント力がついたかも

設問 6
なにか要望、感想などあったら自由に書いてください

- ✓ １期目だったので、全体的に楽すぎたような気がしなくもありません。１期目でももう少しだけ大変にしても良いと思います。
- ✓ 最初に設計の仕方などを教えて欲しかったです。それからなら、もう少しうまくコードが書けた気がします。
- ✓ 新規履修者は知識レベルにばらつきがあるので、研究会そのものはある程度のレベルを要求する内容で行い､別途知 識の不足している人のフォローをする､というかたちだと、全体のレベルアップが図れていいのではないかと思う。
- ✓ 1 期目の人は、一人仕事だったので、もうちょっと他の人とコミュニケーションをとれるような企画がほしいです。
- ✓ いろいろな技術を使うテーマを設定して組むとみんなの知識がひろがる

□秋学期について

設問 7
授業の難易度はどうでしたか？

- ✓ むずかしすぎた 0 人 （0%）
- ✓ むずかしいがどうにかついていける 3 人 （33%）
- ✓ ちょうどいい 5 人 （56%）
- ✓ やさしいが得るものはあった 1 人 （11%）
- ✓ やさしすぎた 0 人 （0%）

設問 8
サブゼミについてわからない点、ほかにやってほしかったことなどあったら
書いてください

- ✓ ありません。
- ✓ 「グループワークのリーダーとしてなにをすべきか」を知りたかった。
- ✓ サーブレットや JSP 以外のもの、アプリケーションなどによる WEB アプリの製作などがしてみたいと思いました。
- ✓ 学期頭にその学期のサブゼミで取り扱う内容を一通り触れてもらうと、その学期でどういった知識が身につくかの見当がついてよかったと思う。
- ✓ やることはそんなに難しいことじゃないと思うんですが、もろもろの理由で動かないことが多くて、ちょっとせつないですね。

設問 9
研究会のなかでどのようなことが身についたと思いますか？
またどのようなことが身につかなかった と思いますか？

- ✓ 概念モデルとクラス図を書く意味がわかり、そして如何に綺麗に書くかが大事なのか分かりました。また概念モデルとク
- ✓ ラス図の書き方が分かってきたと思います。
- ✓ 適度に距離をおいたメンバーでグループワークをすることにおける、障害、問題点などがわかった。
- ✓ 設計や分析などの勉強はかなりよくできたと思う。もう少し実装についてのサブゼミを充実して欲しかった。
- ✓ チームでソフトウエアを開発することの大変さを学んだ。また、設計時にコミュニケーションを図る道具としての UML の有用性が理解できた。
- ✓ モデルの考え方基礎、グループワークの難しさ。コードの実装力
- ✓ UML を使ってみんなと意思疎通を図りつつアプリを作ること。はじめは UML と実際のコードがまったく結びつかなかったの ですが、みんなで取り組んでいくうちに、「ああ、なるほどな」と思いました。
- ✓ 設計の大切さが身にしみた。
- ✓ より深くオブジェクトを使えるようになった。周辺の技術もだんだん応用できるようになった

設問 10
今学期は自分の企画ではなく、人がたてた企画を実装してもらいました。
それについてはどう思いますか？

- ✓ 人が立てた計画を実装することで、人と自分のそのアプリに対する考え方の違いが分かり、よりよい WEB アプリを作ることができるようになったと思います。
- ✓ 将来企業に入ったら、自分の企画を実装する機会は少なく、人の企画・提案を実装する機会が多いと思うので、よいと 思います。自分の企画を実装する、となると、実装の難

しい点を少し丸め込んで終わらせることも可能といえば可能です が､自分以外のクライアントの声を聞いて実装する場合、ごまかしがきかないので技術面の問題に立ち向かわなくては ならない場面が発生し、勉強にもなると思います。
- ✓ コミュニケーションについては深く体感できた。自分の企画のほうが作り甲斐があると思う。
- ✓ おもしろい方法だと思うが、途中から実装が主になってしまったので、進め方を考えたほうが良いかもしれない。
- ✓ 自分で立てた企画と違って､要求がはっきりとあるからいい意味で妥協ができないので､ソフトウエア開発の訓練として はこの形式がいいと思う。
- ✓ 相手の意思疎通が開発に加わりおもしろい。
- ✓ はじめは「興味ない…」と思っていましたが､やってみるとそれはそれで楽しめました。視野が広がっていいと思います。
- ✓ 大変いいことだと思う。
- ✓ 非常に面白いと思った。実際の社会に近い感じだと思う。ただ企画が重要なので質の良い企画を出すことが重要

設問 11
UML の講義を予習とテストの解説だけにして、あとは実際に使ってもらう形で
学んでもらいましたが、使っていく中でわからなかった点などあったら書いてください

- ✓ 習ったことを実際に使う場面になって､本当にここには習ったことを使うべきなのかが､確信を持てず、苦労しました。
- ✓ 特に無し
- ✓ 実際に進めていく中で学べたので特にテストは必要なかったかもしれない。
- ✓ プログラミングと同じで、実際に書いてみないとなかなか深い理解にたどり着かないので、概論をさっと流してあとは実践という形式はいいと思う。
- ✓ 予習のあと、疑問点を抱えたままテストをするのはちょっと辛かったです。でも他にどうしたらよいのかあまりいい案が浮かびませんが。
- ✓ 細かい使い方、についてわからなかった XOR とか。

設問 12
感想、要望などあったら自由に書いてください

- ✓ 今回の研究会は大変楽しかったです。
- ✓ 電源不足に泣いた。電源が簡単にたくさん確保できる部屋で開講したい。
- ✓ とても勉強になった。
- ✓ ある程度履修回数が多い人は、他の研究室やプロジェクトから企画を請け負って実装するという形に出来ると、より一層
- ✓ いい勉強になるのではないかと思った。
- ✓ 共通テーマ(簡単なもの)と自由テーマの両面でやると全員が全員のやってることに興味を持つと思う