

ソフトウェア開発の手順・チームでのソフトウェア開発

樋口さぶろお

龍谷大学工学部数理情報学科

応用プログラミング☆実習 L14(2018-01-09 Tue)

最終更新: Time-stamp: "2018-01-09 Tue 07:37 JST hig"

今日の目標

- 世の中のソフトウェア開発手順を説明できる
- 世の中のチームでのソフトウェア開発方法を説明できる
- チーム内, チーム間でコミュニケーションしてソフトウェアを開発できる



<http://hig3.net>

大規模ソフトウェア開発手順 I

ウォーターフォールと言われる古典的開発手法での開発手順を zip-server/client の例で言うと、

- ① システム要件定義 (基本計画)
 - ▶ (やらなかった) この程度の正確さ, 速さ, 更新労力で住所を知りたい
- ② システム方式設計 (外部設計)
 - ▶ (教員が与えた) データはテキストファイルで. Linux で動く, インターネットで通信するサーバクライアントシステムで.
- ③ ソフトウェア要件定義 (外部設計)
 - ▶ クライアントは, メニューベースで, こういう機能を選択できるように.
- ④ ソフトウェア方式設計 (内部設計)
 - ▶ サーバとクライアントに分割, データ検索は zip.c で別モジュール.
- ⑤ ソフトウェア詳細設計 (プログラム設計)
 - ▶ モジュール分割 (→ 別チームに任せられる) を含む. その界面の定義.
 - ▶ こういうプロトコルで (Excel の設計書).

大規模ソフトウェア開発手順 II

- ▶ LSearch は何を引数にとり, 何を返す. zipdataS はどういうフォーマット. ZIP 構造体の定義. (zip.h とドキュメント)
- ⑥ ソフトウェア構築 (プログラミング)
 - ▶ サーバ/クライアントの main を書く. LSearch を書く. デバッグする.

上流工程: ユーザ側



下流工程: コンピュータ側

ウォーターフォールと言われる古典的開発手法では, 上から下へ (上流工程から下流工程へ) 1 段階ずつ進む.

この後に **テスト** とと言われる工程があるが, 下流から上流に遡るように進む.

ユーザが使い始めてから, **保守** を行う.

うまくソフトウェアを開発する方法を整備すること, は **ソフトウェア工学** という 1 個の学問領域.

ソフトウェアテスト (検証)

- システムテスト
 - ▶ 全体を結合してシステムが正しく機能するか確かめる.
- 結合テスト
 - ▶ 本物の親と子を使って, 親の外側から見て正しく機能するか確かめる.
- 単体テスト (ユニットテスト)
 - ▶ リクエストに対して LSearch を正しい引数で呼び出している
 - ▶ LSearch が正しい値を返す
 - ▶ ダミーの親 (ドライバ) で子を, ダミーの子 (スタブ) で親をテストしておく.

上流工程: ユーザ側



下流工程: コンピュータ側

満たすべき性質を, この引数を与えるとこの返り値, のリストとしてあらかじめ与えておく.

ここでいうテストとは, 学力試験や検定や試行のことではない.

呼ばれる側 (関数) の単体テストに使うドライバの例

main.c

```
1 #include <stdio.h>
2 #include "zip.h"
3 /* 略:data の準備 */
4 int main(){
5 /* 略:data の準備 */
6     printf("%d", LSearch(data, 100, 5202145));
7     return 0;
8 }
9
10 /* 開発中の LSearch をテスト*/
```

呼ぶ側 (main) の単体テストに使うスタブの例

zip.c

```
1 #include <stdio.h>
2 #include "zip.h"
3
4 int LSearch(ZIP data [], int ndata, int key){
5     int i;
6     if( ndata>0 && key<=9999999 && key>=0){
7         i=ndata-1;
8         return i;
9     } else {
10        fprintf("out_of_range.");
11        exit(1);
12    }
13 }
```

ソフトウェア開発と職種

業種 (会社): 情報, ソフトウェア, 通信, 製造, ...

職種 (人): PG, SE, 営業, ...

- 日本ではソフトウェア構築より上を担当する技術者を SE(システムエンジニア), (ソフトウェア詳細設計や) ソフトウェア構築を担当する技術者を PG(プログラマ) という職種名で呼ぶことが多い.
- SE 担当部分の中でも最上流を担当する場合, お客様のアウトな希望や困りごとを聞いて, 「こういうシステムで解決できます」などと提案することになる.

ソフトウェア開発とチームとキャリアプラン

- 案件ごとに、プロジェクトマネージャに率いられた数人から数十人のチームで行われることが多い。
- SE(... エンジニア, ... エンジニア, ...), プログラマ, テスター, デザイナ, 事務担当, ...
- プロジェクトマネジメントはそれ自身が**経営学**のひとつの研究対象。

技術者は、経験を積むとともに、下流から上流に移動していくことが多い。
プログラマ的な仕事をする SE → 下流担当 SE → 上流担当 SE → プロジェクトマネージャ

そういう仕組みの外で高収入で、尊敬されてやっていく情報技術者もいる。建築会社に対して、宮大工や彫刻家やカリスマ建築家のようなもの。システムコンサルタント、フリーランスプログラマ。

大学と就職活動でのグループワーク

就職活動(教員を含む)では、その場で集まったメンバーで、何々を作成し、何々について話し合っって結論を出せ、と言われて、採用担当者がその様子をじっと観察している、ということが起きる。

リーダー(議長)、タイムキーパー、記録係、などの役目を臨時で決めることがよく行われる。

ソフトウェア開発と仕様

ソフトウェアの製品には形がないので、簡単な指定はできない。発注者と提供者 (=お客さんとソフトウェア会社, main の担当者と LSearch の担当者) の間で、提供者の責任を明らかにしておくことが必要。詳細な仕様書が作られる。

- 「3cm×2cm×100cm の檜の角材 10 本」
- 「使いやすいポータル作って」
- 本システムは…と…の機能を有する。…は…

今回のチームプロジェクトでめざすこと

各自のチーム活動スキルの課題を発見するために、次回にチーム内で相互評価を行います。成績には無関係、匿名、(教員と本人以外に)非公表。

	NA	1	2	3	4	5
チームの作業進行	評価対象になる機会がなかった。	自分の担当部分を実行しない。	自分の担当部分を実行する。担当部分の状況について説明する。	2に加え、チーム全体の効果的な作業進行について提案する。	2,3に加え、異なる提案の間を調整する。	2,3,4の間に加え、メンバーから提案が出され、チームの全メンバーが貢献できるように配慮する。
プログラムの仕様と機能と構築についてのコミュニケーション	評価対象になる機会がなかった。	質問や説明をしない。	本人にとって疑問なことを質問する。	本人にとって疑問なことを質問する。相手から求められたときに説明をする。	質問や説明が必要なときに自分から質問や説明をする。	質問や説明が必要なときに聞き手の状況を配慮して質問や説明をする。
説明方法	評価対象になる機会がなかった。	説明しない。	不確かなまま説明する。	知っていることについて正しく説明する。	知っていることについて正しく説明する。知らないことはそれを明示して持っている範囲の情報を示す。	聞き手にあわせて言い方や例を工夫して可能な情報を提供する。

ループリック評価

お知らせ

- 樋口オフィスアワー月 3.5 – 4.5(1-502), 金 4(1-502)
- L15 の最初でも非参照のテストやります
 - ▶ ソフトウェア開発について. 予告しにくいので Learn Math Moodle の予習問題から察してね
- **基本情報技術者** プログラミングだけでない情報技術の資格. 3年生の春に合格可能. 申込期限:2018-02-19, 試験:2018-04-15(日).
<https://www.jitec.ipa.go.jp>