

コラボレティブ開発特論

産業技術大学院大学
中鉢 欣秀

2016 年度

コラボレイティブ開発特論

産業技術大学院大学
中鉢 欣秀

第 1 章 モダンなソフトウェア開発の道具達

コラボレイティブ開発特論

- 第 1 章 モダンなソフトウェア開発の道具達
 - 連絡事項
 - 授業の全体像
 - 授業の方法
 - 「学びの共同体」
 - モダンなソフトウェア開発とは
 - 準備作業

連絡事項 (1)

資料等の入手先

- ▶ GitHub の下記リポジトリにまとめておきます
 - ▶ https://github.com/ychubachi/collaborative_development
- ▶ 資料は随時 update するので、適宜、最新版をダウンロードしてください

Twitter のハッシュタグ

- ▶ Twitter ハッシュタグ -> #enpit_aiit

連絡事項 (2)

仮想環境 (Vagrant)

- ▶ 各自の PC に仮想環境をインストールしておいてください
 - ▶ PC を持っていない方には貸出も可能です (数量限定)
- ▶ インストール方法については下記を参照
 - ▶ enpit/preparation.org at master · ychubachi/enpit

コラボレイティブ開発特論

- 第1章 モダンなソフトウェア開発の道具達
 - 連絡事項
 - 授業の全体像
 - 授業の方法
 - 「学びの共同体」
 - モダンなソフトウェア開発とは
 - 準備作業

学習目標と目的

目標

- ▶ ビジネスアプリケーションを構築するための基礎力
- ▶ 分散型 PBL を実施する上で必要となる知識やツールの使い方
- ▶ これら活用するための自己組織的なチームワーク

目的

- ▶ 分散ソフトウェア開発のための道具を学ぶ
 - ▶ 開発環境 (Ruby), VCS とリモートリポジトリ (GitHub)

前提知識と到達目標

前提とする知識

- ▶ 情報系の学部レベルで基礎的な知識を持っていること

最低到達目標

- ▶ 授業で取り上げる各種ツールの基本的な使い方を身につける

上位到達目標

- ▶ 授業で取り上げる各種ツールの高度な使い方に習熟する.

授業の形態

対面授業

- ▶ 担当教員による講義・演習

個人演習

- ▶ 個人によるソフトウェア開発

グループ演習

- ▶ グループによるソフトウェア開発

コラボレイティブ開発特論

- 第1章 モダンなソフトウェア開発の道具達
 - 連絡事項
 - 授業の全体像
 - 授業の方法
 - 「学びの共同体」
 - モダンなソフトウェア開発とは
 - 準備作業

講義 ・ 演習 ・ 課題

講義

- ▶ ツールの説明
- ▶ ツールの使い方

演習

- ▶ 個人でツールを使えるようになる
- ▶ グループでツールを使えるようになる

成績評価

課題

- ▶ 個人でソフトウェアを作る
- ▶ グループでソフトウェアを作る

評価の方法

- ▶ 課題提出と実技試験

評価の観点

- ▶ 分散 PBL で役に立つ知識が習得できたかどうか

コラボレイティブ開発特論

- 第1章 モダンなソフトウェア開発の道具達
 - 連絡事項
 - 授業の全体像
 - 授業の方法
 - 「学びの共同体」
 - モダンなソフトウェア開発とは
 - 準備作業

共に学び、共に教える「場」

- ▶ この教室に集うメンバーは学びの共同体である
- ▶ 他人に教えることは、優れた学びの方法の一つ
- ▶ 困った時には助けを求める

自己紹介

- ▶ 教員
 - ▶ 自己紹介
- ▶ 履修者
 - ▶ 名前（可能であれば所属も）
 - ▶ どんな仕事をしているか（あるいは大学で学んだこと）
 - ▶ この授業を履修した動機

コラボレイティブ開発特論

- 第1章 モダンなソフトウェア開発の道具達
 - 連絡事項
 - 授業の全体像
 - 授業の方法
 - 「学びの共同体」
 - モダンなソフトウェア開発とは
 - 準備作業

ソフトウェア開発のための方法・言語・道具

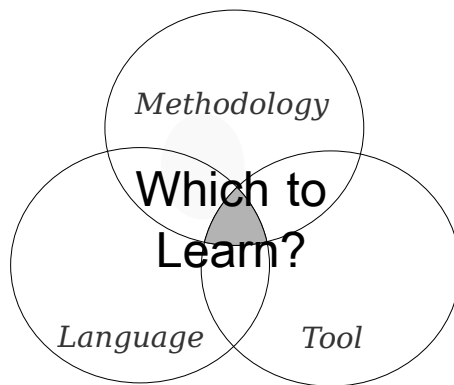


Figure: The Framework-Language-Tool framework.

授業で取り上げる範囲

取り上げること

- ▶ 方法を支えるための道具
- ▶ 良い道具には設計概念として方法論が組み込まれている
- ▶ 道具はプログラミング言語を問わない

取り扱わないこと

- ▶ 方法論そのものについてはアジャイル開発特論で学ぶ
- ▶ 言語の備えるエコシステムについては必要な範囲で学ぶ

Scrum するための道具

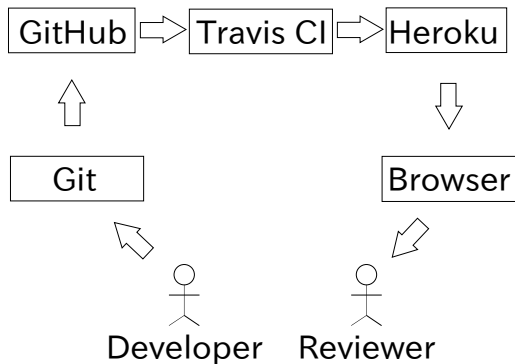


Figure: The modern tools for Scrum developments.

モダンな開発環境の全体像

仮想化技術 (Virtualization)

- ▶ Windows や Mac で Linux 上での Web アプリケーション開発を学ぶことができる
- ▶ Heroku や Travis CI 等のクラウドでの実行や検査環境として用いられている

ソーシャルコーディング (Social Coding)

- ▶ Linux のソースコードの VCS として用いられている Git を学ぶ
- ▶ Git は GitHub と連携することで OSS 型のチーム開発ができる

enPiT 仮想化環境

インストール済みの言語と道具

- ▶ エディタ (Emacs/Vim)
- ▶ Ruby の実行環境
- ▶ GitHub, Heroku, Travis CI と連携するための各種コマンド (github-connect.sh, hub, heroku, travis)
- ▶ PostgreSQL のクライアント・サーバーと DB
- ▶ 各種設定ファイル (.bash_profile, .gemrc, .gitconfig)
- ▶ その他

コラボレイティブ開発特論

- 第1章 モダンなソフトウェア開発の道具達
 - 連絡事項
 - 授業の全体像
 - 授業の方法
 - 「学びの共同体」
 - モダンなソフトウェア開発とは
 - 準備作業

クラウドのアカウント作成

GitHub

- ▶ [Join GitHub · GitHub]

Heroku

- ▶ [Heroku - Sign up]

Travis CI

- ▶ [Travis CI]
 - ▶ Travis CI は、GitHub のアカウントでログインできる

enPiT 仮想化環境のアップデート

作業内容

- ▶ enPiT 仮想化環境（vagrant の box）を更新しておく

コマンド

```
1 cd ~/enpit  
2 vagrant destroy  
3 vagrant box update
```


Port Forward の設定

説明

- ▶ Guest OS で実行するサーバに，Host OS から Web ブラウザでアクセスできるようにしておく
- ▶ 任意のエディタで Vagrantfile を変更

変更前

```
# config.vm.network "forwarded_port", gu
```

変更後

```
config.vm.network "forwarded_port", gues  
config.vm.network "forwarded_port", gues
```

enPiT 仮想化環境にログイン

作業内容

- ▶ 前の操作に引き続き、仮想化環境に SSH 接続する

コマンド

```
1 vagrant up  
2 vagrant ssh
```

github-connect スクリプト

URL

- ▶ [github-connect.sh]

git config を代行

- ▶ GitHub にログインし，名前と email を読み込んで git に設定

SSH の鍵生成と登録

- ▶ SSH 鍵を作成し，公開鍵を GitHub に登録してくれる

github-connect.sh の実行

作業内容

- ▶ スクリプトを起動し、設定を行う
- ▶ GitHub のログイン名とパスワードを聞かれるので、入力する
- ▶ rsa key pair のパスフレーズは入力しなくて構わない

コマンド

```
github-connect.sh
```

Git と GitHub の設定確認

Git の設定確認

```
git config --list
```

GitHub の設定確認

- ▶ ブラウザで GitHub の SSH Key ページを開く