

# 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

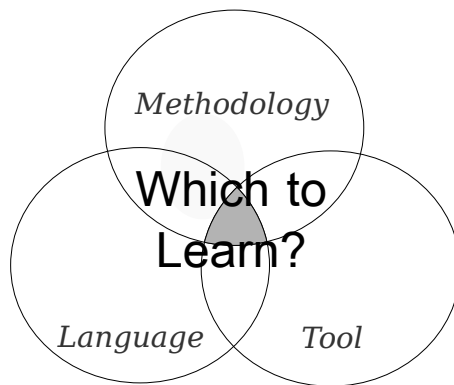
産業技術大学院大学  
中鉢 欣秀

## 第2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2章 モダンなソフトウェア開発の道具達
  - モダンなソフトウェア開発とは
    - 仮想環境の準備
    - クライドの準備

# 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

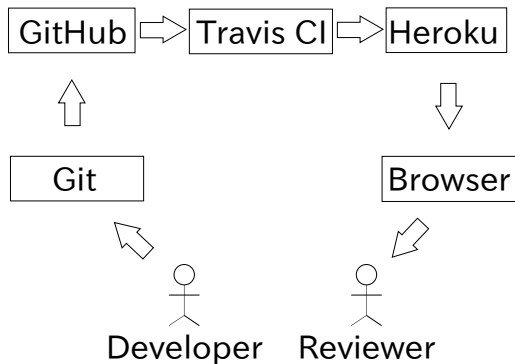
## 取り上げること

- ▶ 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2章 モダンなソフトウェア開発の道具達
  - モダンなソフトウェア開発とは
  - 仮想環境の準備
  - クラウドの準備

# 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
```

# コラボレイティブ開発特論

- 第2章 モダンなソフトウェア開発の道具達
  - モダンなソフトウェア開発とは
  - 仮想環境の準備
  - クライドの準備

# GitHub/Heroku のアカウントを作成

## 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

# 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