MTAT.03.295
Agile Software Development
Lecture 1: Introduction

Luciano García-Bañuelos
University of Tartu
Course Objective

The objective of this course is to introduce an agile method for software development that takes advantage of the principles of cloud computing and software as a service.
A bit of history about the course
Connection with edX CS 169.1x

• edX is an initiative backed by several large universities (e.g., MIT, Harvard, Berkley) promoting the use of massive online open courses (MOOCs). The initiative includes:
  ◦ An open source project with all the code supporting edX
  ◦ A web site providing already a large number of courses

• Our course is a direct clone of Berkeley’s CS 169.1x as offered by edX
  ◦ We have a small private online course (SPOC) for our class
  ◦ You will be using edX materials, autograders, etc.
Approach

Take two SaaS projects from conception to deployment

- Mimic the interaction with non-technical customers
- Backend: Ruby on Rails
- Frontend: Modern frontend CSS frameworks, Javascript and AngularJS
- Project tracking, Testing, Continuous integration, Deployment to the cloud
Agile method – Sneak preview

- Talk to “Customer”
- User stories (BDD)
- Unit test (TDD)
- Velocity measurement
- Deployment (Cloud)

Legacy

Design patterns
Agile method – Sneak preview

Talk to “Customer”

User stories (BDD)

Unit test (TDD)

Velocity measurement

Deployment (Cloud)

Lo-Fi UI/Storyboards
Balsamiq

Points, velocity,
Pivotal tracker

Continuous Integration
Jenkins/
Organization of the Course

- Lecture/Practice (Tuesdays/Thursdays)
- SPOC Lectures (Released every Tuesday)
  - First 7 weeks
  - Probably, some additional lectures from Berkeley’s CS 169.2x
- Project (weeks 9-15)
  - Presentations on 13 December
  - Report due on 20 December
- See details on the course pages:
  - https://courses.cs.ut.ee/2016/asd
Grading

• **SPOC:** 20 points
  ◦ 6 Homework assignments and 4 quizzes

• **Project:** 20 points
  ◦ 12 points for the product
    ◦ Soundness of design and architectural choices
    ◦ Scope of the solution (how much was implemented?)
    ◦ Functional correctness/validation (does it work?)
  ◦ 2 evidence of use of the agile method
  ◦ 3 points for written documentation
  ◦ 3 points for presentation

• **Exam:** 60 points

• **Participation:** up to 10 bonus points
Important Dates

• Project
  ◦ Presentation  13.12

• Exams
  ◦ 5.01.2017
  ◦ 10.01.2017
Hands on with Ruby
def factorial(n):
    if (n < 0):
        raise "Undefined"
    end
    if (n == 0):
        return 1
    else:
        return n * factorial(n - 1)
    end
end

puts factorial(10)
puts factorial(3)
puts factorial(-1)
def factorial(n):
    if n < 0:
        raise "Undefined"
    elif n == 0:
        return 1
    else:
        return n * factorial(n - 1)
end

• Parenthesis can sometimes be omitted
• return is not required (a function returns the result of the last executed statement)