

My best MOOC ever - Experience with the Berkeley SaaS course

Ignacio Reguero 4 April 2014

What is in a MOOC

- Massive Open Online Courses
- Platforms such as
 - edX.org
 - Coursera.org
- Tons of (mostly) university level courses
 - Not necessarily about computing
- Typically lectures combined with weekly homework + quizzes.

The MOOC that I did

- Software as a Service
- CS169.1x and CS169.2x from EdX.org
- Based on book By Armando Fox and David Patterson
 - Influenced by ACM discussions on improved curriculum for Software Engineering
 - Overview of Software Engineering with emphasis on Agile methods.
- Explains SaaS, SOA, Cloud Computing, Three Tiered architecture, MVC, REST...
- With examples and assignments based on Ruby on Rails as well as a bit of JavaScript (the good parts).

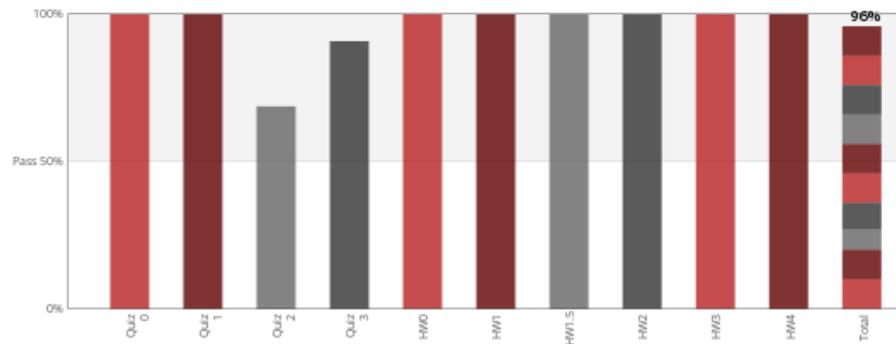
Nice Tools for the MOOC Assignments

- Ubuntu Linux VM with the course environment
 - To run with VirtualBox or Amazon EC2, Openstack,
- GitHub
- Heroku
 - Cool PaaS for Rails Apps
- Contacts with teachers, other students and TAs through
 - EdX platform, Chat
 - Encouraged to do Pair programming with Google Hangouts

MOOC Logistics

- Cost effective
 - The MOOC itself is free
 - I paid 50\$ for ID verification
 - I paid for the book.
- Quite relevant for my work objective
- Got support from my management
 - But did it on evenings + week ends.
- Got a Nice Certificate
 - Automagically send to my Linkedin account
 - But no grade on it
 - Got snapshot of Web page with scores and made PDF

Course Progress for Student 'reguero' (ignacio.reguero@cern.ch)



Overview

[Welcome to CS169.1X](#)

No problem scores in this section

[edX Tutorial \(3/3\) 100%](#)

Practice Scores: 1/1 1/1 1/1

Introduction to Software Engineering

[Overview \(21:21\) \(1/1\) 100%](#)

Practice Scores: 1/1

[1.1: Introduction to Software Engineering\(5:59\)](#)

No problem scores in this section

[1.2: Software as a Service \(6:39\) \(1/1\) 100%](#)

Practice Scores: 1/1

[1.3: Service Oriented Architecture \(7:48\) \(1/1\) 100%](#)

Practice Scores: 1/1

[1.4: Cloud Computing\(9:32\) \(1/1\) 100%](#)

Practice Scores: 1/1

[1.5: Beautiful Code vs. Legacy Code \(2:55\) \(1/1\) 100%](#)

Practice Scores: 1/1

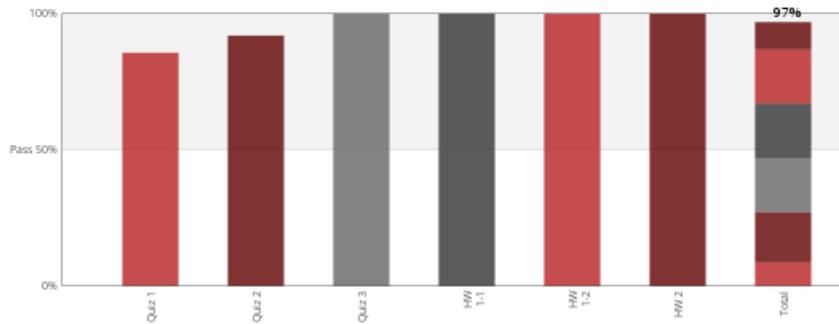
[1.6: Software Quality Assurance \(5:18\) \(1/1\) 100%](#)

Practice Scores: 1/1

[1.7: Productivity: Conciseness, Synthesis, Reuse, and Tools \(8:45\) \(1/1\) 100%](#)

Practice Scores: 1/1

Course Progress for Student 'reguero' (ignacio.reguero@cern.ch)



Advanced Rails

Public Resources

No problem scores in this section

[5.1: DRyIng Out MVC \(17:40\) \(1/1\) 100%](#)

Practice Scores: 1/1

[5.2: Single Sign-On and Third-Party Authentication \(8:58\) \(1/1\) 100%](#)

Practice Scores: 1/1

[5.3: Single Sign-On Example \(13:58\)](#)

No problem scores in this section

[5.3 cont: Associations & Foreign Keys \(8:32\) \(1/1\) 100%](#)

Practice Scores: 1/1

[5.3 cont: ActiveRecord Association Support \(8:41\) \(1/1\) 100%](#)

Practice Scores: 1/1

[5.3 addendum: Associations: mechanics \(7:40\) \(1/1\) 100%](#)

Practice Scores: 1/1

[5.4: Through-Associations \(14:08\) \(1/1\) 100%](#)

Practice Scores: 1/1

[5.5: RESTful Routes for Associations \(16:38\) \(1/1\) 100%](#)

Practice Scores: 1/1

[5.6: DRyIng Out Queries with Reusable Scopes \(11:03\) \(1/1\) 100%](#)

Practice Scores: 1/1

[5.7-5.9: Associations Wrap-Up \(10:55\) \(1/1\) 100%](#)

Practice Scores: 1/1

[QUIZ 1 \(6/7\) 86%](#)

Quiz 1 due Nov 17, 2013 at 23:59 UTC

Problem Scores: 1/1 1/1 1/1 1/1 1/1 1/1 0/1

Quick Conclusion

- The material was state of the art
- The combination of lectures with programming assignments and quizzes gives a different insight.
- It required a bit of personal investment but it was much better than any course that I've done in a very long time.
- New run of this MOOC coming next week.

If Time: Some Impacting Concepts from the MOOC



Agile Methods: best for SaaS

- Methodology in which change is the norm rather than the exception
 - 2001 Agile Manifesto
- Emphasis on working with others in small teams:
 - Pair Programming
 - SCRUM

Testing as Foundation of Agile

- Behaviour Driven Design:
 - User Stories define the behaviour of the app.
 - Tool: Cucumber (with Capybara)
- Test Driven Development:
 - Before writing the code, define executable tests for the interfaces that you are going to implement
 - Tool: Rspec
- Run these tests all the time
 - Continuous Integration tools

Ruby: rocks

- Everything is an object
 - And every operation is a method call
 - Method chains facilitated
- Duck Typing
- Reflection and metaprogramming facilities
 - `method_missing`
- Blocks:
 - Lambda expressions that carry scope around with them -> closures
- Yield:
 - allow to trivially implement operators

Software as a Service (SaaS)

- Delivering SW and data as a service through the Internet via a thin program such as a Web browser instead of distributing application binaries.
- Single copy of program in the cloud VS Shrink-wrap sw with millions of copies of the program
- Examples are
 - Search, Social Networking, Watching videos
- Frameworks that facilitate SaaS Development
 - Python/Django, Java/EJB, Ruby/Rails

Service Oriented Architecture (SOA)

- SOA is when components of an application act as interoperable services that can be used independently and combined in other applications.
- 2002 e-mail from Amazon CEO
- SaaS is a special case of SOA