Introduction to Software Development Principles and Practices (SWPP)

M1522.000100

Chung-Kil Hur

(Credit: Byung-Gon Chun)

All You Ever Wanted to Know about How to Build Large-Scale Software ☺

SWPP, CSE, SNU

Who am I?

- Prof. Chung-Kil (Gil) Hur [허충길]
  - Education: KAIST (B.S.), U of Cambridge (Ph.D.)
  - Software Foundations Lab
    http://sf.snu.ac.kr
  - Research Topics
    - Software Verification
    - Low-level Language Semantics (C/C++/LLVM/Rust)
    - Concurrency Models
  - Our collaborators
    - [UK] U of Cambridge, Microsoft Research Cambridge
    - [Germany] Max Planck Institute for Software Systems
    - [France] INRIA
    - [USA] Princeton, UPenn, Utah, State U of New York at Oswego, Google, IBM, Mozilla, Azul Systems.
  - Publications
    - 8 top conference papers (last 4 years at SNU).
      PLDI(4),POPL(2),ICFP(1),AAAI(1)

Teaching Staff

- Instructor: Chung-Kil Hur
  - Email: gil.hur@sf.snu.ac.kr
  - Office: Bldg. 301, Rm. 407
  - Office hours: Anytime by appointment

- TAs
  - Sanghoon Park
  - Juneyoung Lee
  - Dongyeon Shin
  - Yonghyun Kim
  - Email at swpp@sf.snu.ac.kr

- Course Web
  https://github.com/snu-sf-class/swpp201701

Goals for Today

- What is this course about?

- How does this class operate?

- Interactive is important!
  - Ask questions!

This Course is About

- Principles + Practices
  of building large-scale software systems

- An hands-on course on large-scale software systems: project-oriented
  - This semester’s theme is web services

This Course is About

- Building large software systems that actually work is hard. This course covers techniques for dealing with the complexity of software systems

- We will focus on the technology of software development principles and software engineering for the individual and small team
    - Specifications, principles of design and software architecture, testing, abstraction, modularity, design patterns, software development process, etc.

This Course is About

- The students are expected to apply the principles to systems in practice by working on semester-long group projects on web services

- You can think that each team is creating its own startup. The students applies software engineering principles to build their software products.

Class Components [Tentative: the percentage can change]

Class participation	5%
Warm up practice (Debugging, Code Refactoring)	20%
Milestone 1 (Specific features + Presentation)	35%
Milestone 2 (Your own features + Poster)	40%

Course Materials

- **There is no required textbook in this class.**

- If you want to read more about the topics covered in the class, I recommend to read the following books.
  - "Engineering Software as a Service: An Agile Approach Using Cloud Computing", by Armando Fox and David Patterson
  - “Software Engineering. A Practitioner’s Approach (6th ed.) ”, by Roger Pressman
  - "Code Complete”, by Steve McConnell
  - "Design Patterns: Elements of Reusable Object-Oriented Software”, by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
  - “Extreme Software Engineering. A Hands-On Approach”, by Daniel H. Steinberg, Daniel W. Palmer
  - “Structure and Interpretation of Computer Programs (SICP) (2nd ed.)”, by Harold Abelson, Gerald Jay Sussman

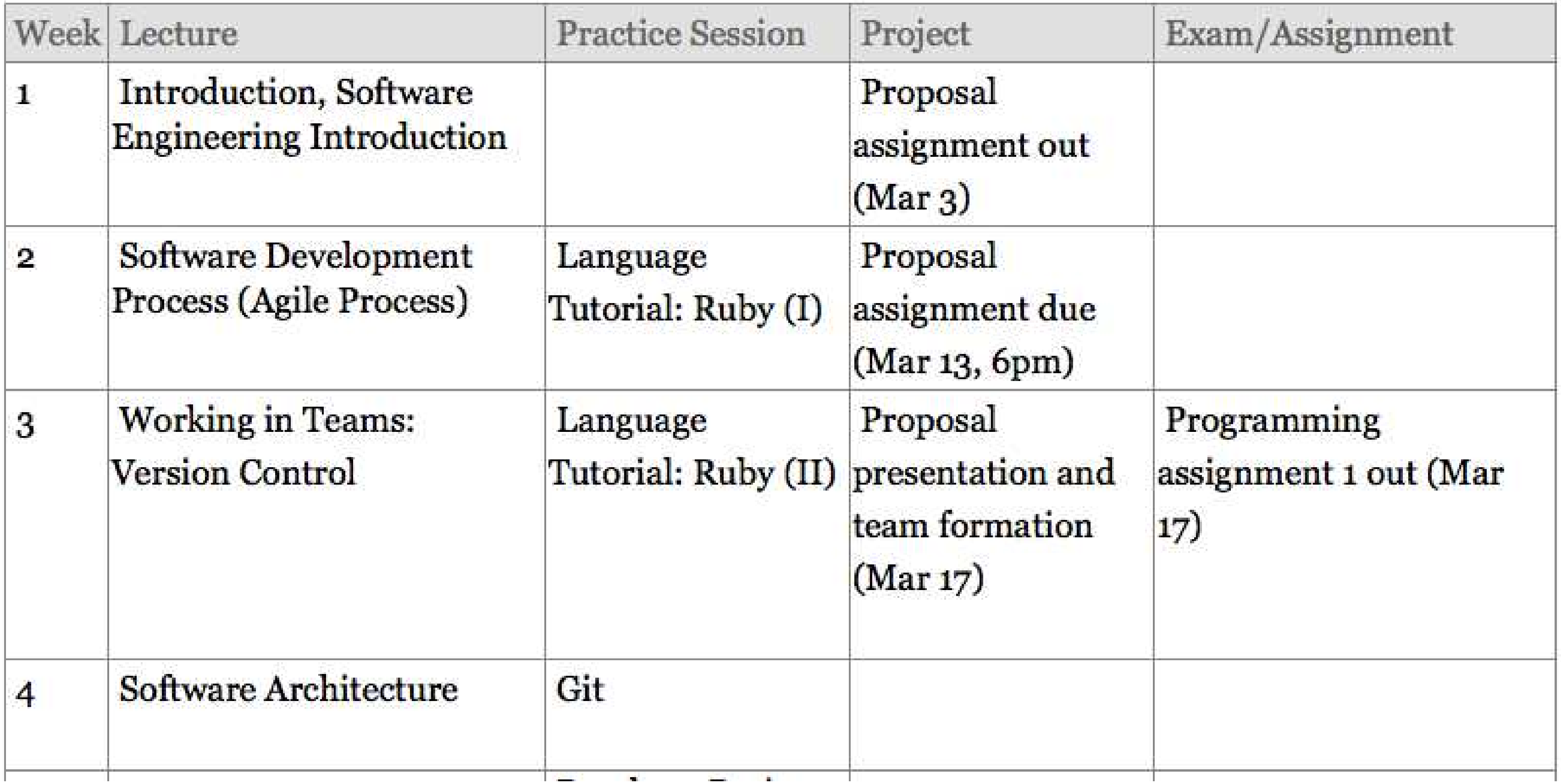
  …

Course Structure

- Lecture – Tue, Thu 5:00-6:00 PM
  - Project presentations
- Practice session – Tue 7:00-9:00 PM
  - Project presentations
  - Step-by-step guidance on software development principles
- Don’t miss practice sessions: lectures and practice sessions go hand in hand

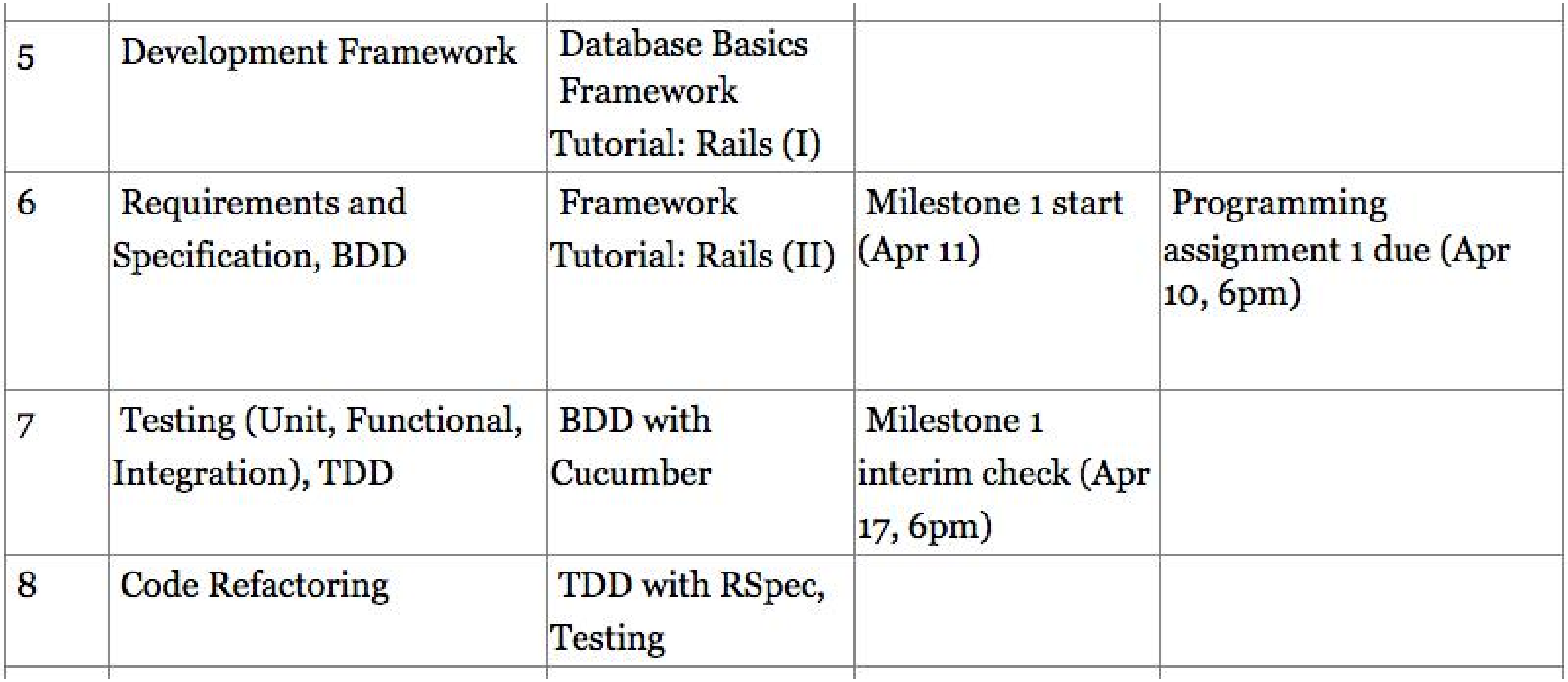
Syllabus (tentative)

(https://sites.google.com/site/snuswppspr2015/)



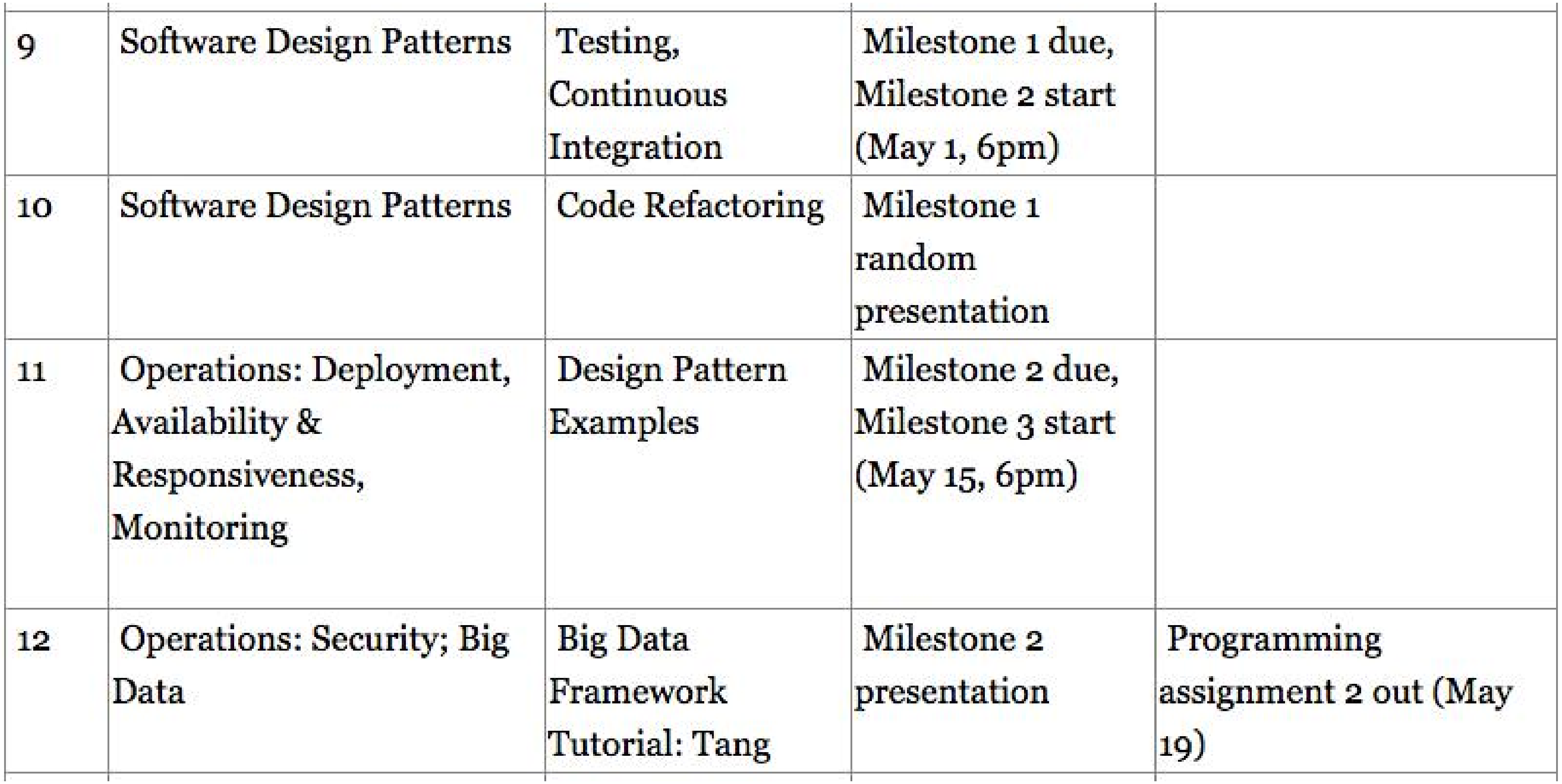
Week	Lecture	Practice Session	Project	Exam/Assignment
1	Introduction, Software Engineering Introduction		Proposal assignment out (Mar 3)	
2	Software Development Process (Agile Process)	Language Tutorial: Ruby (I)	Proposal assignment due (Mar 13, 6pm)	
3	Working in Teams: Version Control	Language Tutorial: Ruby (II)	Proposal presentation and team formation (Mar 17)	Programming assignment 1 out (Mar 17)
4	Software Architecture	Git		

Syllabus (tentative)



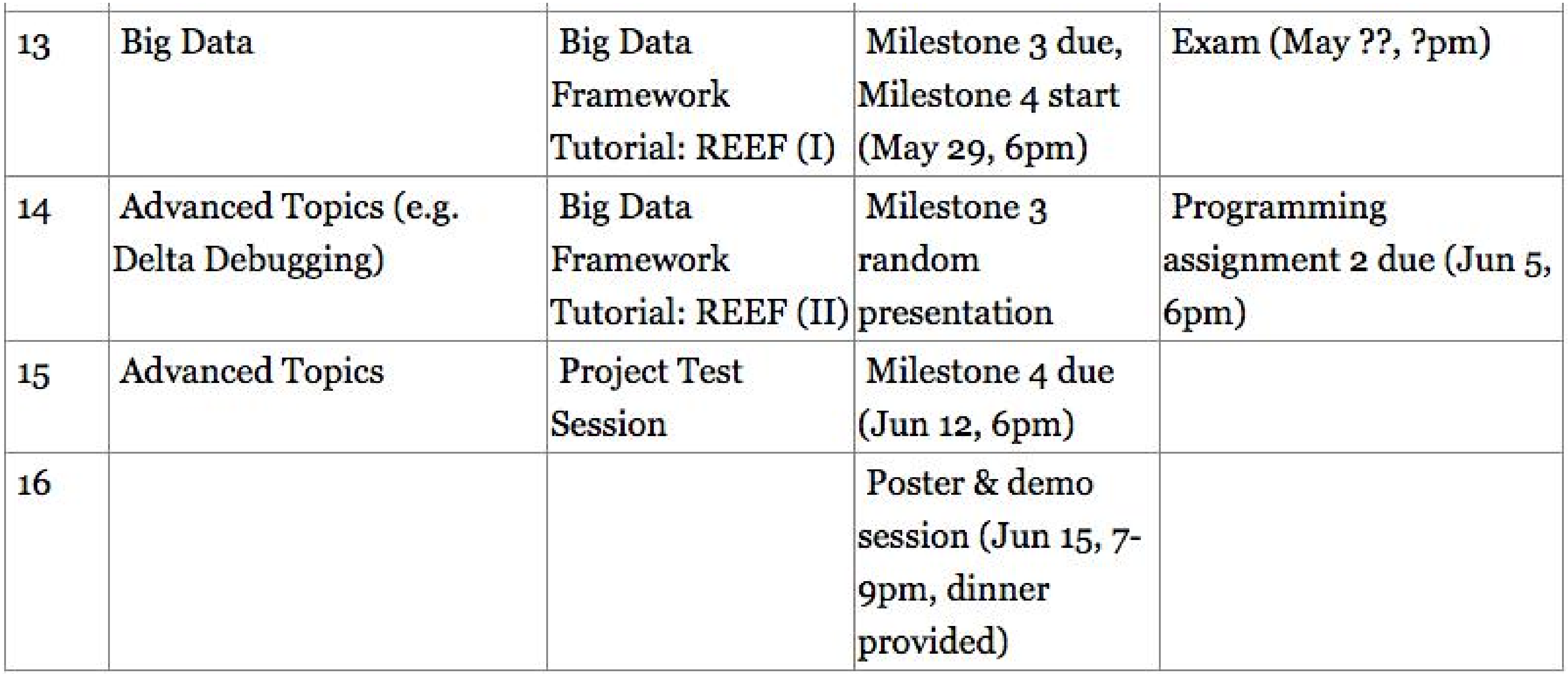
5	Development Framework	Database Basics Framework Tutorial: Rails (I)		
6	Requirements and Specification, BDD	Framework Tutorial: Rails (II)	Milestone 1 start (Apr 11)	Programming assignment 1 due (Apr 10, 6pm)
7	Testing (Unit, Functional, Integration), TDD	BDD with Cucumber	Milestone 1 interim check (Apr 17, 6pm)	
8	Code Refactoring	TDD with RSpec, Testing		

Syllabus (tentative)



9	Software Design Patterns	Testing, Continuous Integration	Milestone 1 due, Milestone 2 start (May 1, 6pm)	
10	Software Design Patterns	Code Refactoring	Milestone 1 random presentation	
11	Operations: Deployment, Availability & Responsiveness, Monitoring	Design Pattern Examples	Milestone 2 due, Milestone 3 start (May 15, 6pm)	
12	Operations: Security; Big Data	Big Data Framework Tutorial: Tang	Milestone 2 presentation	Programming assignment 2 out (May 19)

Syllabus (tentative)



13	Big Data	Big Data Framework Tutorial: REEF (I)	Milestone 3 due, Milestone 4 start (May 29, 6pm)	Exam (May ??, ?pm)
14	Advanced Topics (e.g. Delta Debugging)	Big Data Framework Tutorial: REEF (II)	Milestone 3 random presentation	Programming assignment 2 due (Jun 5, 6pm)
15	Advanced Topics	Project Test Session	Milestone 4 due (Jun 12, 6pm)	
16			Poster & demo session (Jun 15, 7-9pm, dinner provided)	

Finale: Poster & Demo Session

Main Project

- This semester’s theme – web services
  - Facebook

- ~~Build “new” services you’ve dreamed about while learning software development principles and practices~~
  - ~~Amazon, Ebay, Paypal, Uber, Airbnb, Facebook, Twitter, LinkedIn, …~~

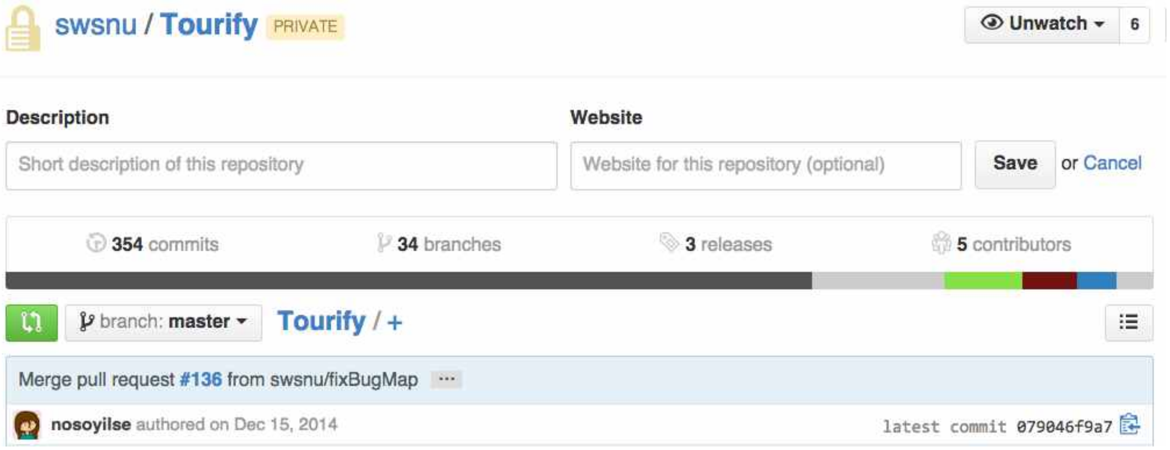
Main Project

- Group: a team of 4 students (no exception)
  - W.h.p. 2-student group(s) fail to finish
- Start forming teams this week!

- Development environment
  - Backend: Python, Django
  - Frontend: HTML5/Javascript, React with Redux

Main Project

- Agile software development process
- Git for version control
- Github for project management
  - Milestones
  - Issues
  - Pull requests
  - Code review



Merge pull request #136 from swsnu/fixBugMap …

nosoyilse authored on Dec 15, 2014

latest commit 079046f9a7

- Testing infra – unit tests/integration tests

Main Project

- There is a project proposal assignment today.
- Thanks to students who took WEC (including TAs), we have sample documents we can share ☺

Project Schedule

- Project Proposal - 1-2 page project description (individual assignment, due 6pm March 13 (Fri), 2015)
- Project Bidding – 1) proposals selected by the staff, 2) select final project topics by bidding who want to work on the projects
- Project requirements and specification
- <u>Project Steps</u>
  - Milestone 1 – project design doc
  - Milestone 2
  - Milestone 3
  - Testing another team’s project
  - Milestone 4 – final report
  - Final poster and demo presentation

Back-end

- The back-end of your project will run on a server in the Amazon EC2 or on Heroku. It is likely that it will use a database (Mongo, MySQL, etc.) and a model-view-controller architecture.

Front-end

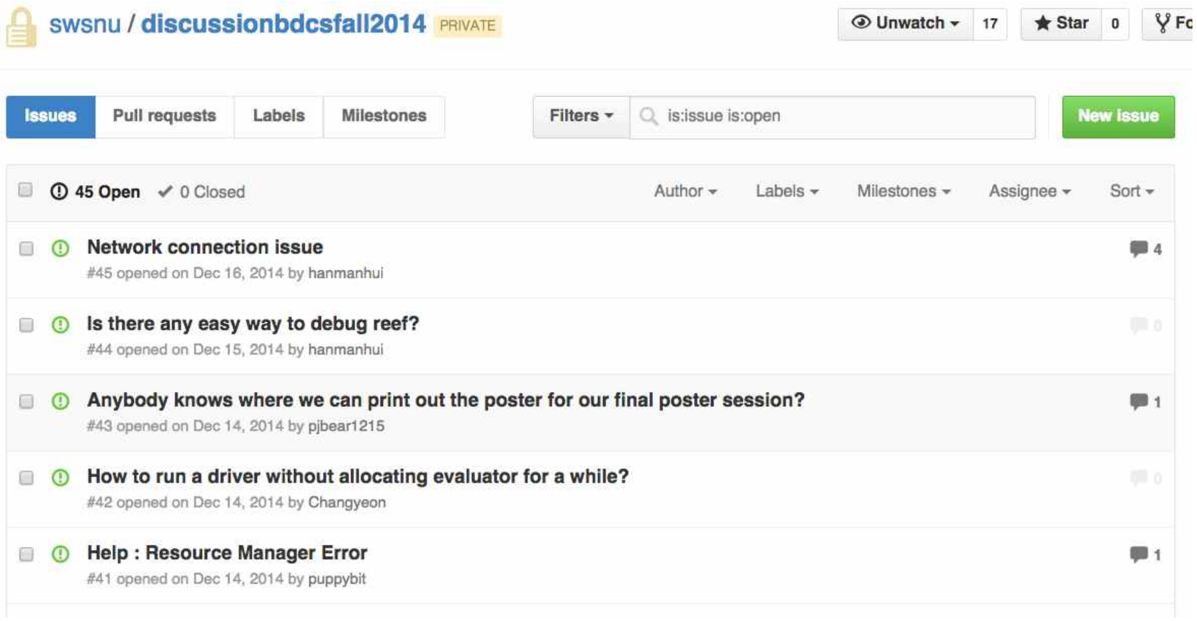
- The front-end for your project will run in a browser. HTML5 and Javascript are your friends.

Warm-up Individual Projects

- Warmup Project1 – Frontend using Javascript, Backend using RoR

Getting Help

- Class ETL for slides and announcement
- Course web site:
  - https://sites.google.com/site/snuswppspr2015/
- Github issues for discussion
- Office hours



Network connection issue 4
#45 opened on Dec 16, 2014 by hanmanhui

Is there any easy way to debug reef? 0
#44 opened on Dec 15, 2014 by hanmanhui

Anybody knows where we can print out the poster for our final poster session? 1
#43 opened on Dec 14, 2014 by pjbear1215

How to run a driver without allocating evaluator for a while? 0
#42 opened on Dec 14, 2014 by Changyeon

Help : Resource Manager Error 1
#41 opened on Dec 14, 2014 by puppybit

Timeliness

- Hard deadlines

- Catastrophic events
  - Major illness, death in family, …
  - Consult your academic advisor to come up with a plan to get back on track
  - Consult with me about this class

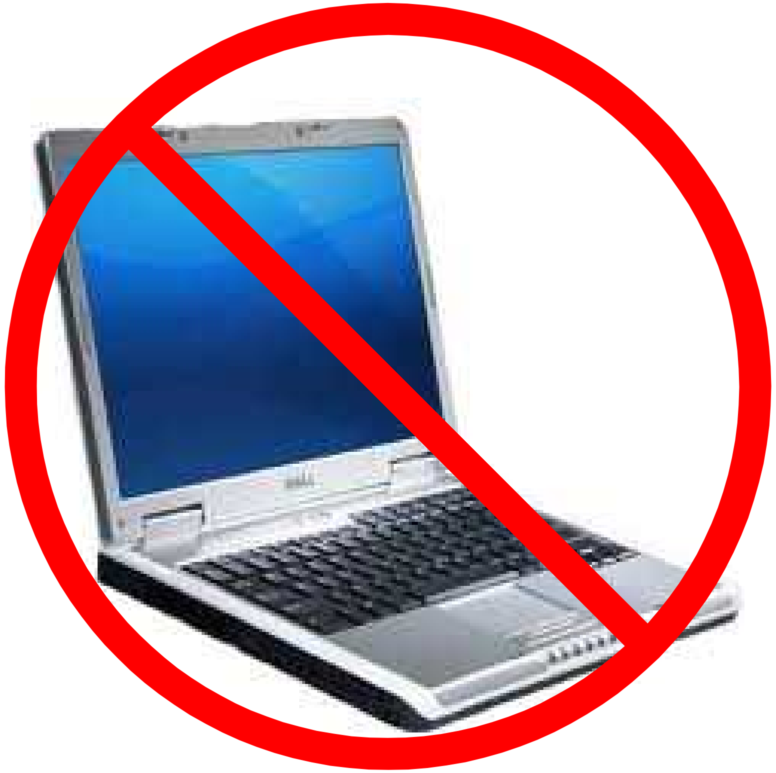
Cheating

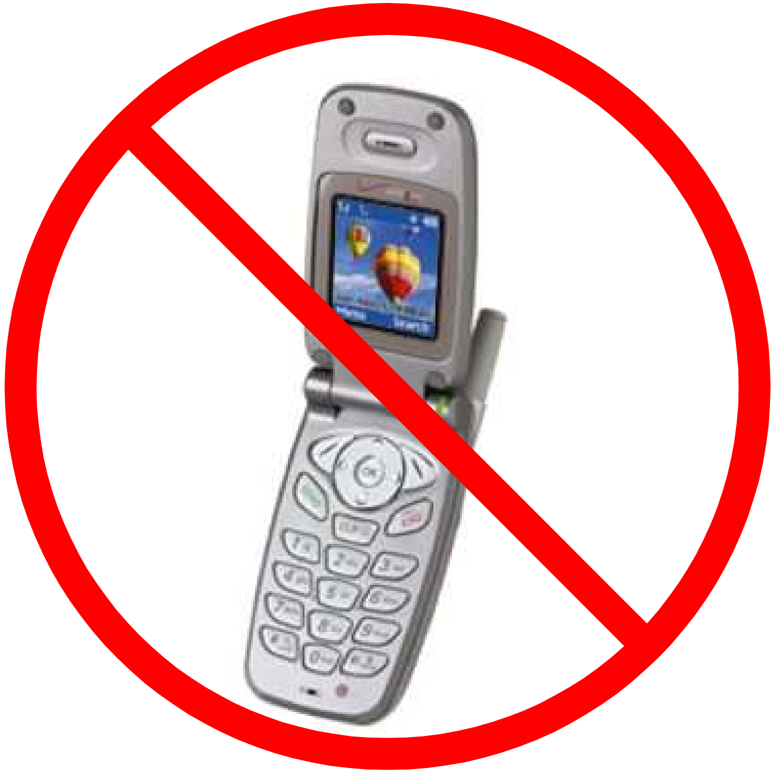
- What is cheating?
  - Sharing code: by copying, retyping, looking at, or supplying a file
  - Coaching: helping your friend to write a programming assignment, line by line
  - Copying code from pervious course or from elsewhere in the Internet
  - Especially, be careful about copying code since we may <u>open</u> your project code! Be alert about code licenses.
- Penalty for cheating
  - Removal from course with failing grade

Other Rules of the Lecture Hall

- **Laptops: closed**
  - **Exceptions: when we actually try things together during the class**
- **Electronic communications: forbidden**
  - **No email, SMS, instant messaging (Kakao, etc.), calls, etc.**
- Presence in lectures
  - We will check attendance
- No recordings of ANY KIND: no video, audio recording allowed
- **No auditing allowed (except (future) course staffs)**

The Rules
(and we really mean it!)





Welcome!

We will have lots of fun!