6 more weeks of winter? 12 more weeks of classes!
Sarah L

Advanced Programming Techniques

COS 333

Christopher Moretti

Agenda

Course Overview

Administrivia

Design Project Introduction

Agenda

Course Overview

Administrivia

Design Project Introduction

```
# ps aux
# cd ~
# ls -altr
# history
```

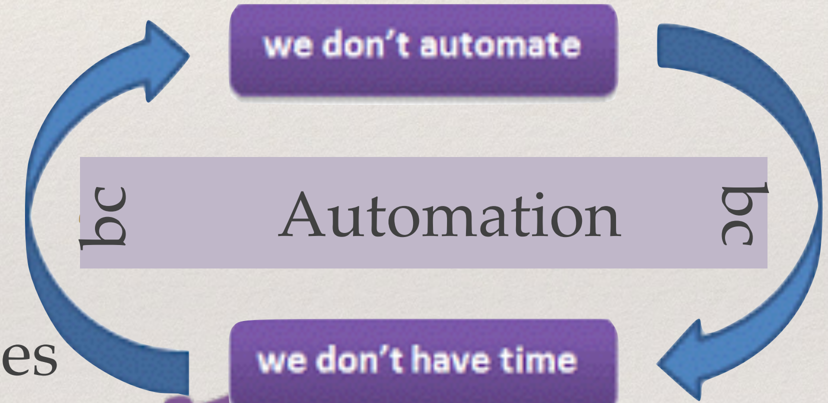
Course Overview

The 20,000 Foot View

But don’t think this is just a tech laundry-list …

Localization

Interfaces



we don’t automate

bc Automation bc

we don’t have time

Design

Standards and Specs

```
<!DOCTYPE html PUBLIC "-/
<html xmlns="http://www.w
<head>
  <meta name="TITLE" con
  <meta http-equiv="Cont
  <meta name="keywords" c
  <meta name="description
  <meta name="Author" con
  <meta name="distribution
  <meta name="copyright" c
  <meta name="content-lang
```

Reusability

Agenda

Course Overview

Administrivia

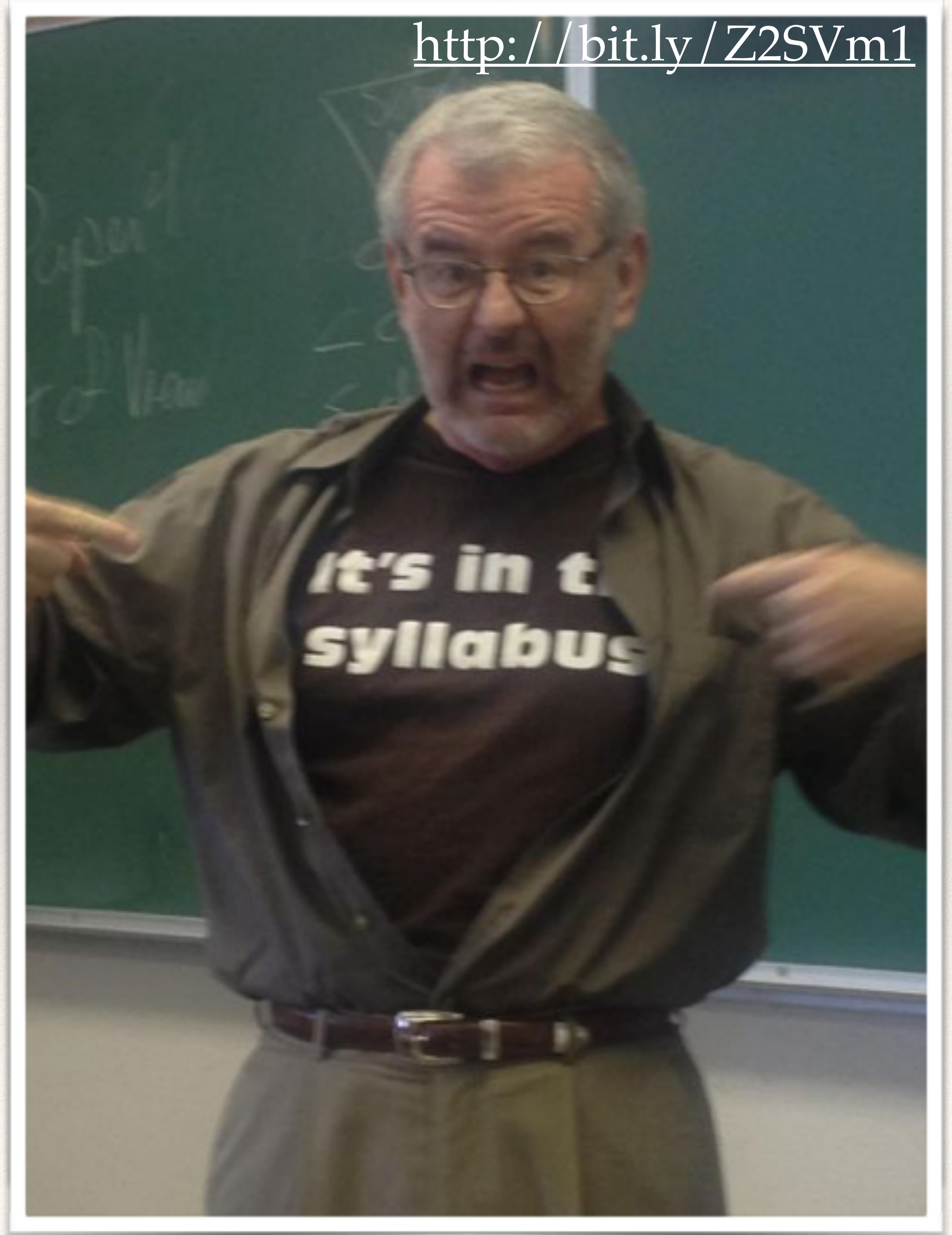
Design Project Introduction

Here's a *very* tentative outline …

week 1:	project info, regular expressions	
weeks 2-3:	scripting: AWK, Bash, Python	A1, A2
week 4:	web development	A3
week 5:	mobile device development	A4
week 6:	software engineering, project details	Design Doc
(spring break)		
week 7:	networks	A5
week 8:	databases, design patterns?	
week 9:	advanced OOP	
week 10:	Go? GUIs?	
week 11-12:	???	
reading period:		Demos!
Dean's date:		Final submission!

Does this count as a syllabus?

- Prerequisites: 217, 226
- Programming assignments:
  - 5, equally weighted: 40%
  - Out T; Due 23:59 on following F
- Design project:
  - This makes the course!
  - Deliverables combined: 60%
  - Start investigating projects, thinking about groups now!

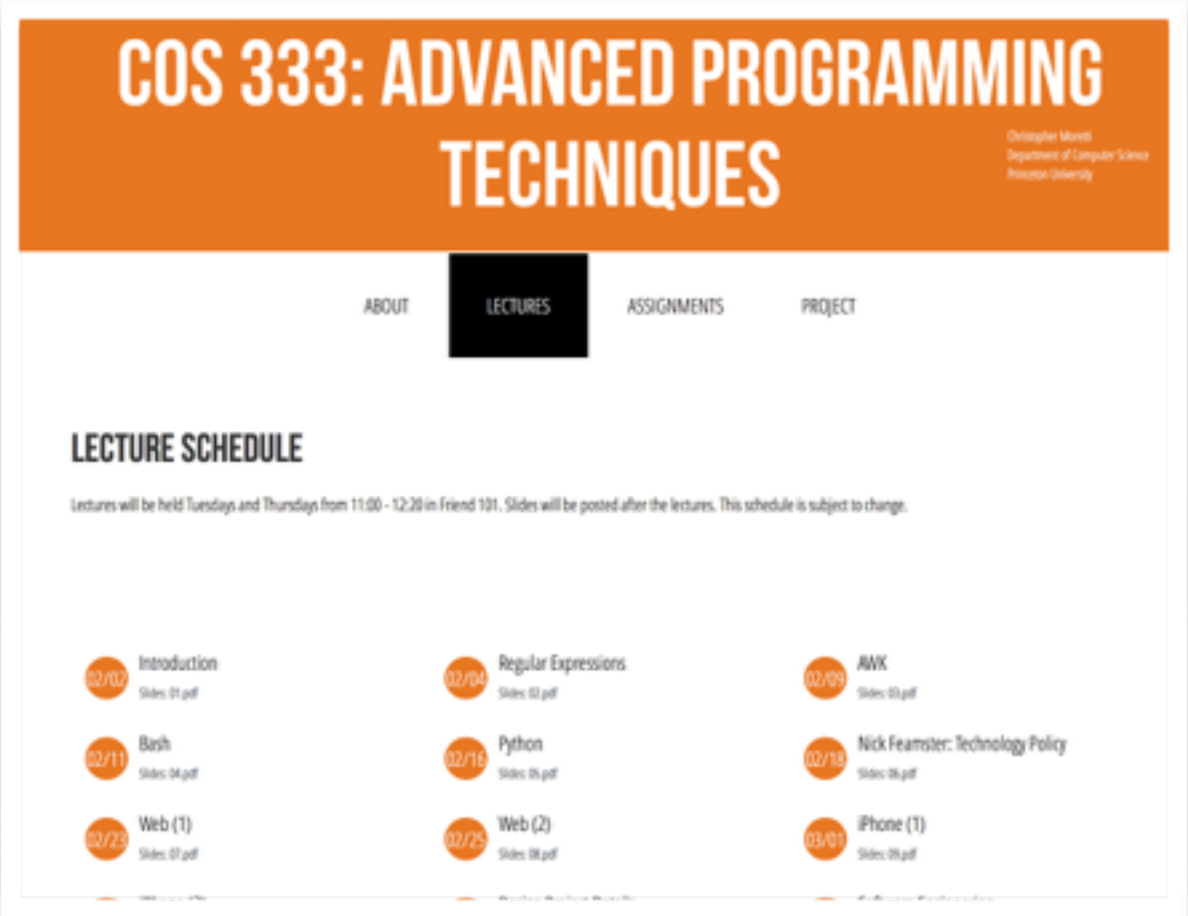


http://bit.ly/Z2SVm1

Course Staff

all images © jorge cham
DO WE HAVE TO MEMORIZE THIS?
ARE YOU GRADING ON A CURVE?
ARE YOU KIDDING ME??
DID YOU EVEN READ THE BOOK??

- Christopher Moretti
  - (cmoretti@cs)
- Mingru Bai
  - (mingrub@cs)
- Nayana Prasad Nagendra
  - (nagendra@cs)
- Nick Turner
  - (nturner@cs)
- Hansen Zhang
  - (hansenz@cs)



COS 333: ADVANCED PROGRAMMING TECHNIQUES

Christopher Moretti
Department of Computer Science
Princeton University

ABOUT | LECTURES | ASSIGNMENTS | PROJECT

LECTURE SCHEDULE

Lectures will be held Tuesdays and Thursdays from 11:00 - 12:20 in Friend 101. Slides will be posted after the lectures. This schedule is subject to change.

02/02 Introduction
Slides: 01.pdf

02/04 Regular Expressions
Slides: 02.pdf

02/09 AWK
Slides: 03.pdf

02/11 Bash
Slides: 04.pdf

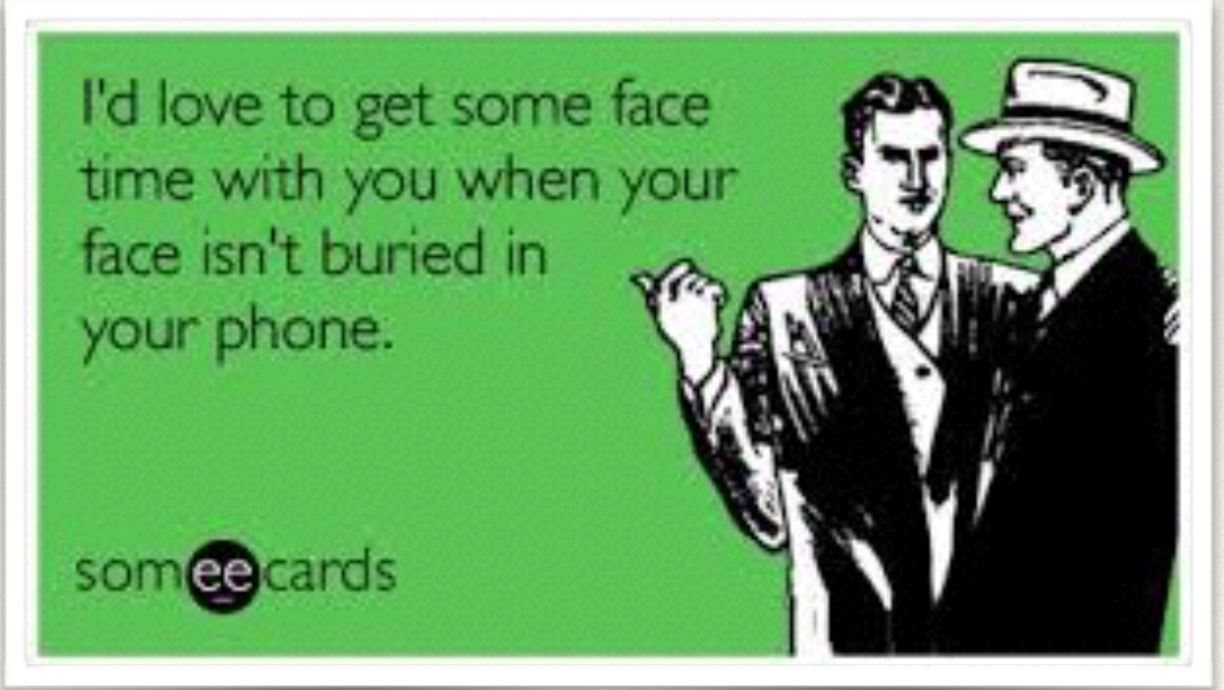
02/16 Python
Slides: 05.pdf

02/18 Nick Feamster: Technology Policy
Slides: 06.pdf

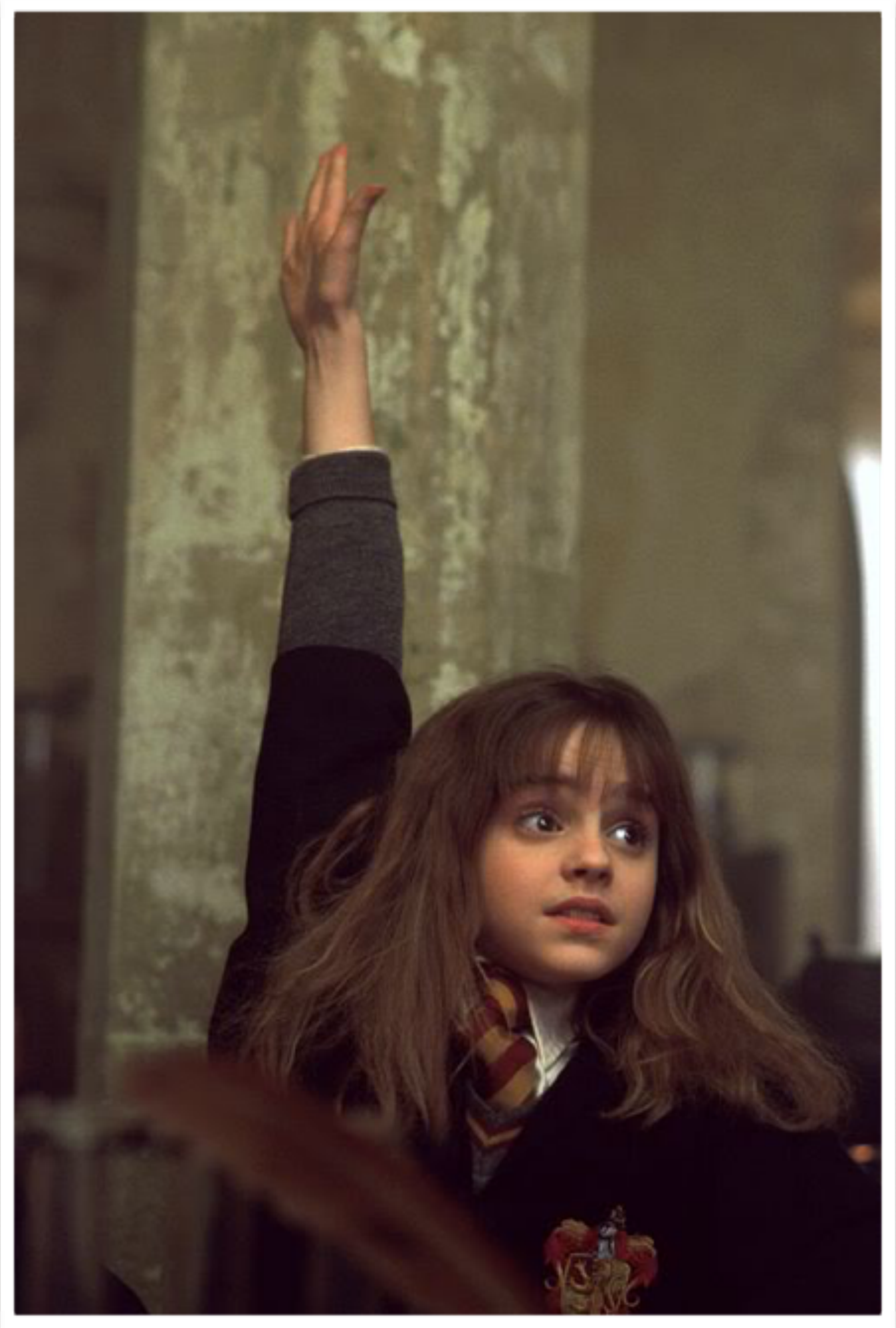
02/23 Web (1)
Slides: 07.pdf

02/25 Web (2)
Slides: 08.pdf

03/01 iPhone (1)
Slides: 09.pdf



I'd love to get some face time with you when your face isn't buried in your phone.



From a previous Course Evaluation form[*]

"Professor lacks expertise on many of the topics he lectured on (he said this); why should I trust what he teaches?"

[*] *for Brian Kernighan*[**]

[**] *you might have noticed that I'm not Brian Kernighan*

Agenda

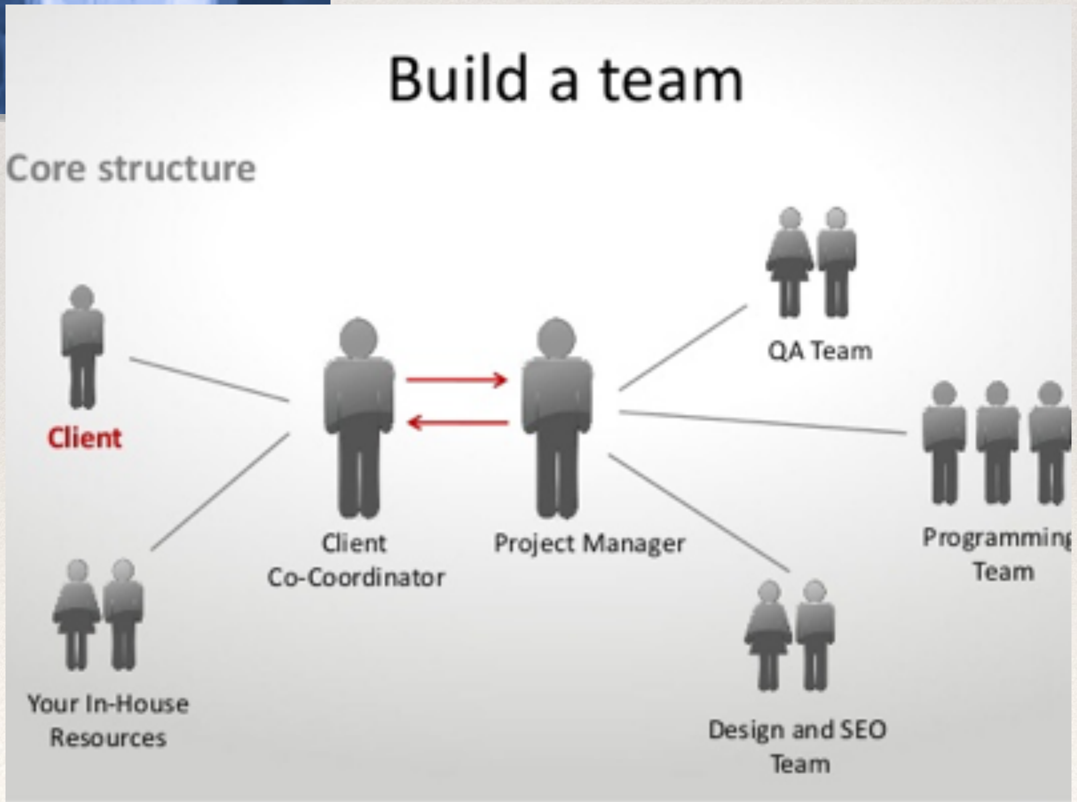
Course Overview

Administrivia

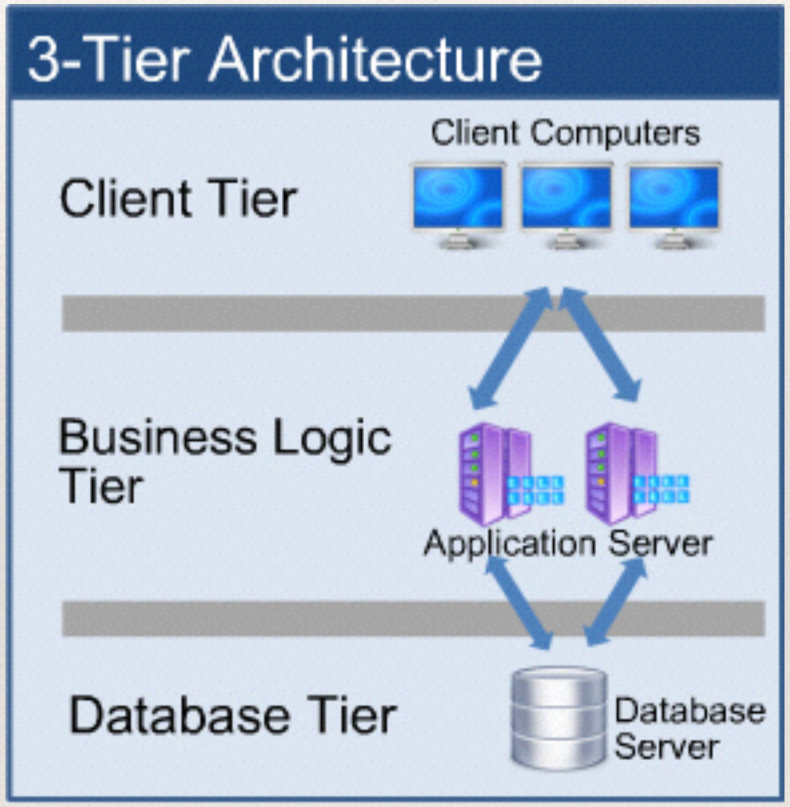
Design Project Introduction

Design Project

“a simulation of reality”



“three-tier” system for any interesting/useful application

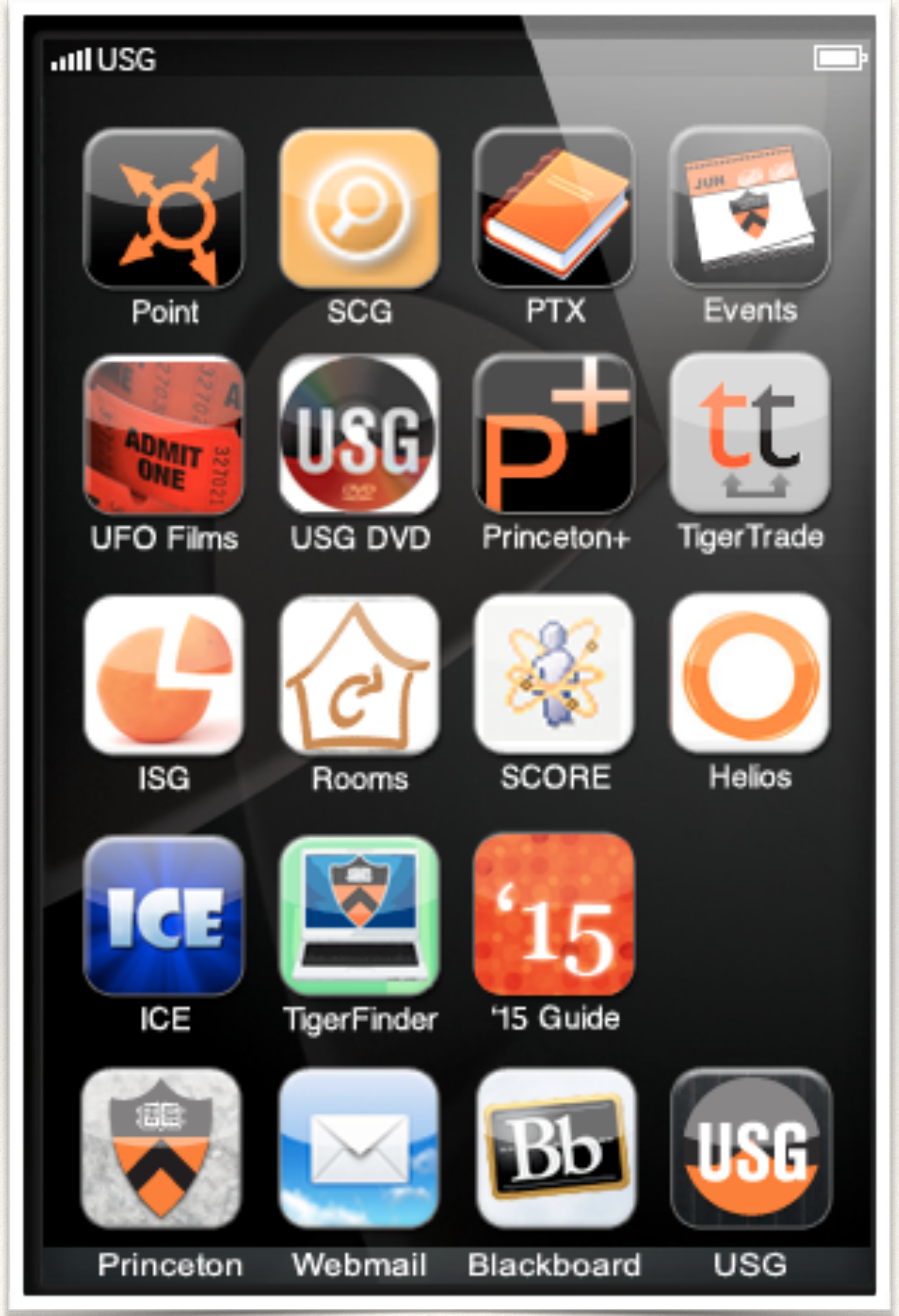


groups of 3-5

“The way to get startup ideas is not to try to think of startup ideas. It's to look for problems, preferably problems you have yourself.

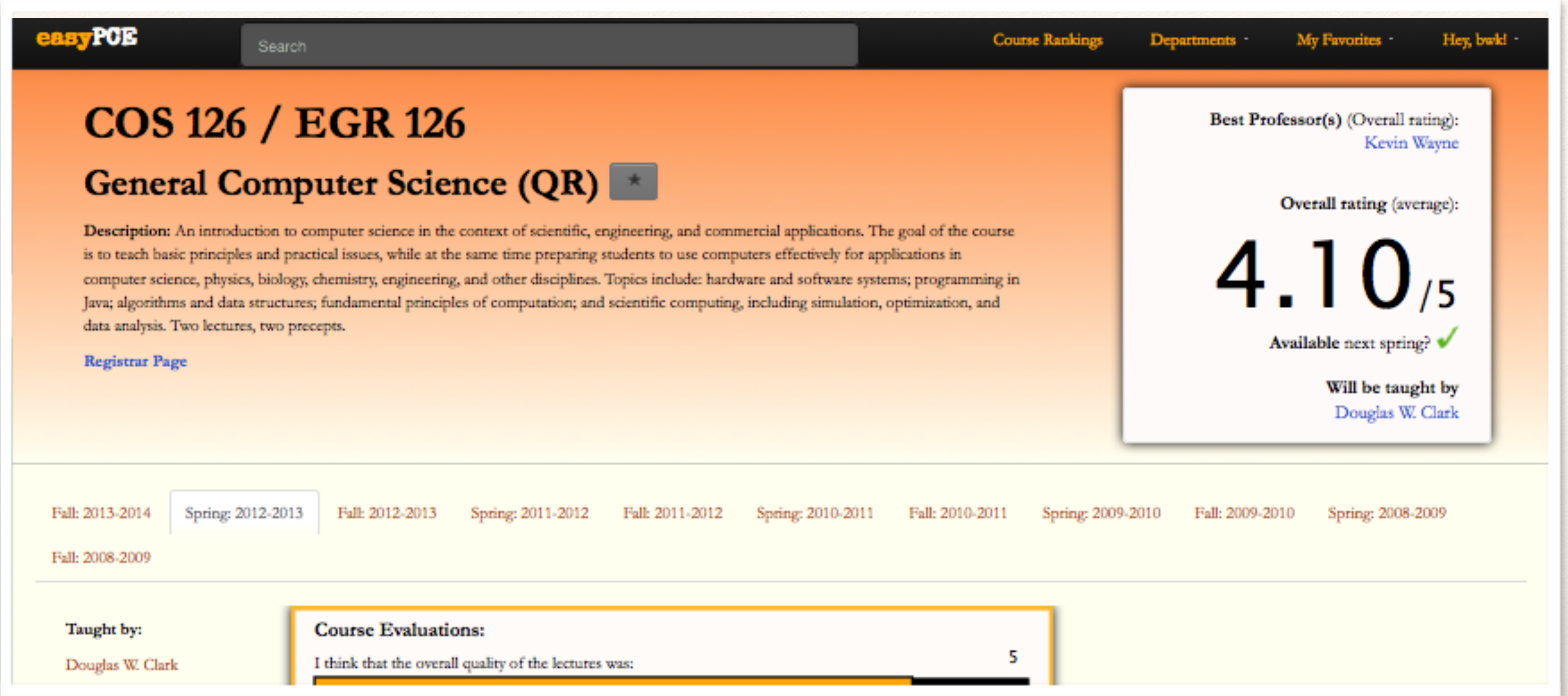
The very best startup ideas tend to have three things in common: they're something the founders themselves want, that they themselves can build, and that few others realize are worth doing.”

– Paul Graham, co-founder of Y Combinator (www.paulgraham.com)



Some familiar examples

- Point
- PTX
- Events
- Rooms
- ICE
- TigerFinder
- Pursuit of Mappiness
- EasyPCE
- PrincetonCAT
- Find-A
- Tigerbook
- ReCal/ReCal.io



COS 126 / EGR 126

General Computer Science (QR)

Description: An introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science, physics, biology, chemistry, engineering, and other disciplines. Topics include: hardware and software systems; programming in Java; algorithms and data structures; fundamental principles of computation; and scientific computing, including simulation, optimization, and data analysis. Two lectures, two precepts.

Registrar Page

Best Professor(s) (Overall rating):
Kevin Wayne

Overall rating (average):

4.10/5

Available next spring? ✓

Will be taught by
Douglas W. Clark

Fall: 2013-2014 | Spring: 2012-2013 | Fall: 2012-2013 | Spring: 2011-2012 | Fall: 2011-2012 | Spring: 2010-2011 | Fall: 2010-2011 | Spring: 2009-2010 | Fall: 2009-2010 | Spring: 2008-2009 | Fall: 2008-2009

Taught by:

Douglas W. Clark

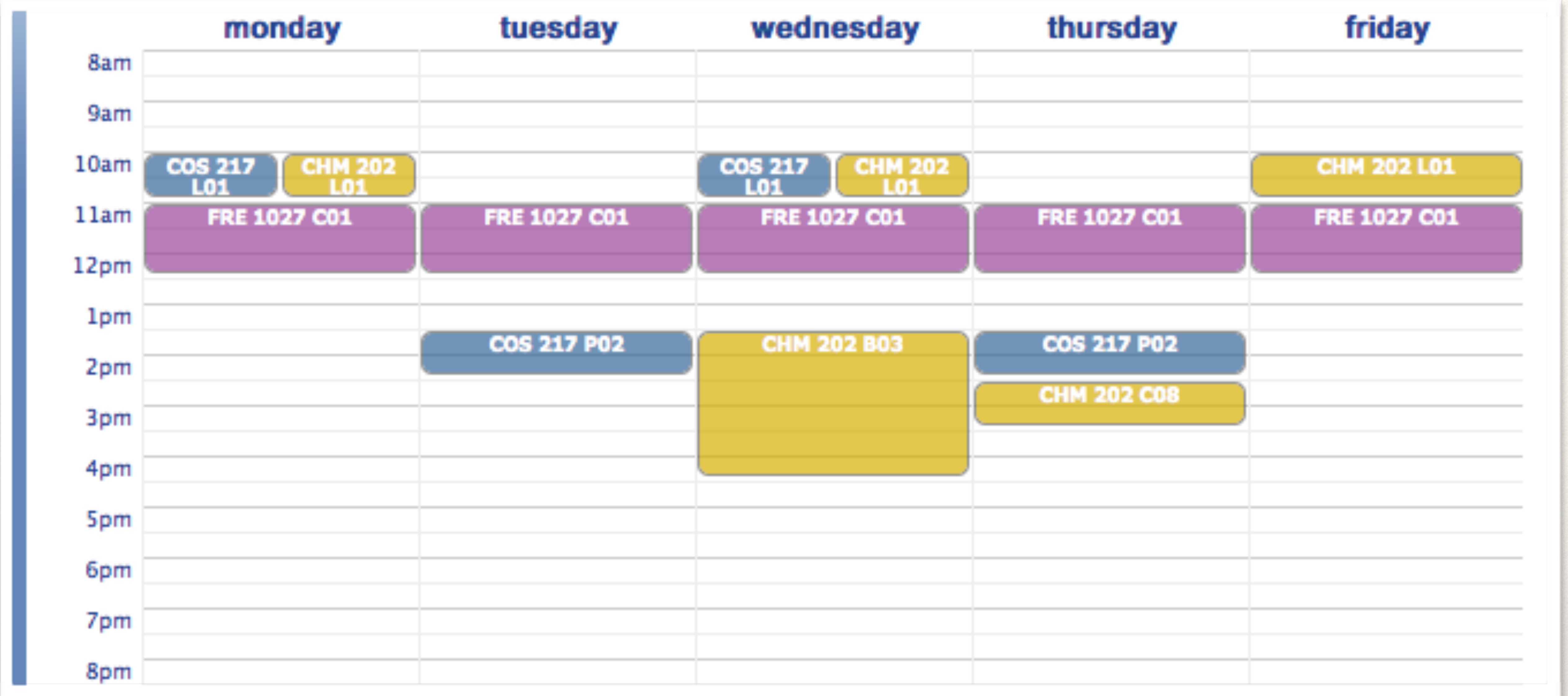
Course Evaluations:

I think that the overall quality of the lectures was: 5

Matt Haake, John Whelchel, and Yacob Yonas

easyPCE

2013: COS 333

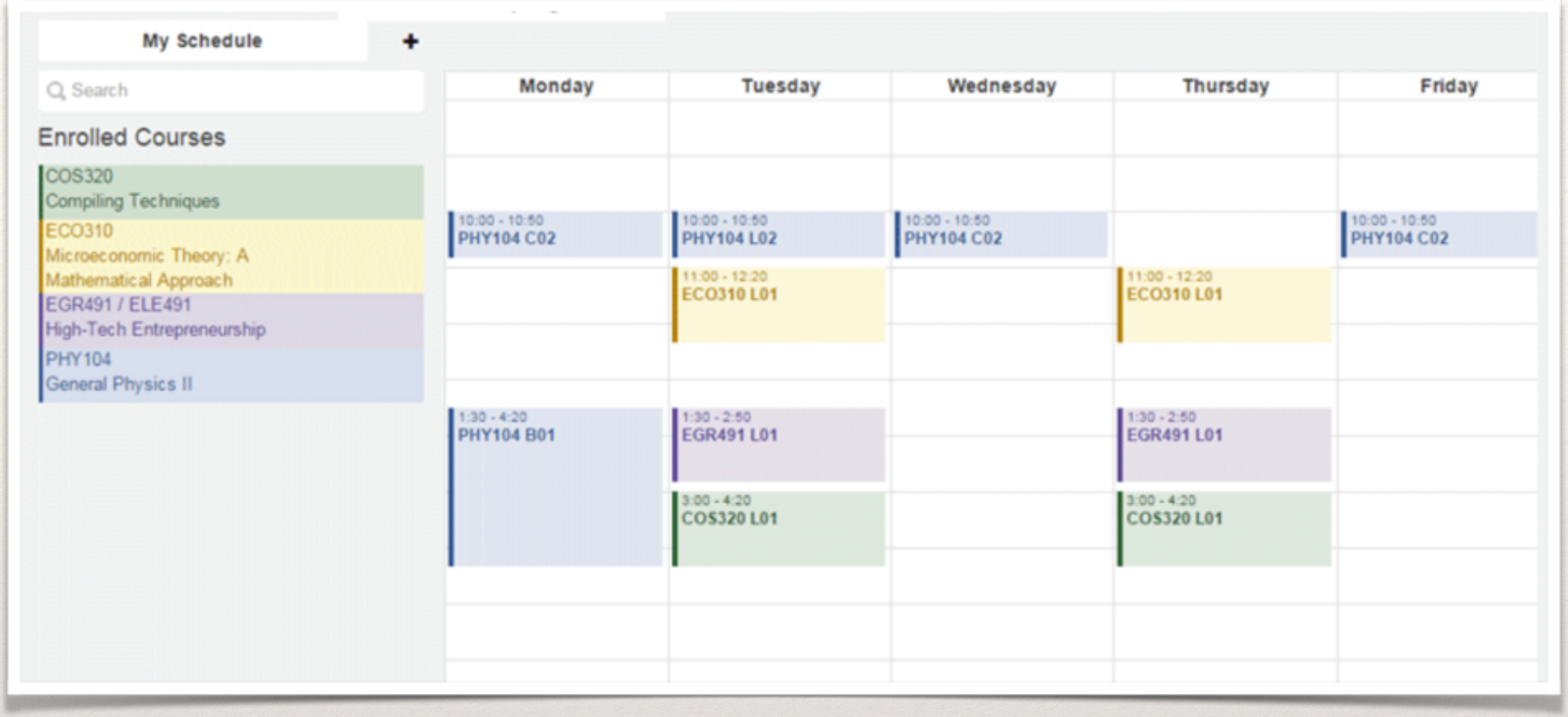


	monday	tuesday	wednesday	thursday	friday
8am					
9am					
10am	COS 217 L01 CHM 202 L01		COS 217 L01 CHM 202 L01		CHM 202 L01
11am	FRE 1027 C01	FRE 1027 C01	FRE 1027 C01	FRE 1027 C01	FRE 1027 C01
12pm					
1pm		COS 217 P02	CHM 202 B03	COS 217 P02	
2pm				CHM 202 C08	
3pm					
4pm					
5pm					
6pm					
7pm					
8pm					

*Daniel Huang, **Gyeong-Sik Choi**, Yoonju Kim, Becker Polverini, Alan Chin*

ICE: Integrated Course Engine

2008: COS 333
2010: COS IW
2012: USG
2015: decommissioning



Enrolled Courses

- COS320 Compiling Techniques
- ECO310 Microeconomic Theory: A Mathematical Approach
- EGR491 / ELE491 High-Tech Entrepreneurship
- PHY104 General Physics II

Monday	Tuesday	Wednesday	Thursday	Friday
10:00 - 10:50 PHY104 C02	10:00 - 10:50 PHY104 L02	10:00 - 10:50 PHY104 C02		10:00 - 10:50 PHY104 C02
	11:00 - 12:20 ECO310 L01		11:00 - 12:20 ECO310 L01	
1:30 - 4:20 PHY104 B01	1:30 - 2:50 EGR491 L01		1:30 - 2:50 EGR491 L01	
	3:00 - 4:20 COS320 L01		3:00 - 4:20 COS320 L01	

Travis Perlee, Naphat Sanguansin, Dyland Xue, Maxim Zaslavsky

ReCal/ReCal.io

2014: COS 333
2014: COS IW
2015 - present: widescale adoption by student body

codePost

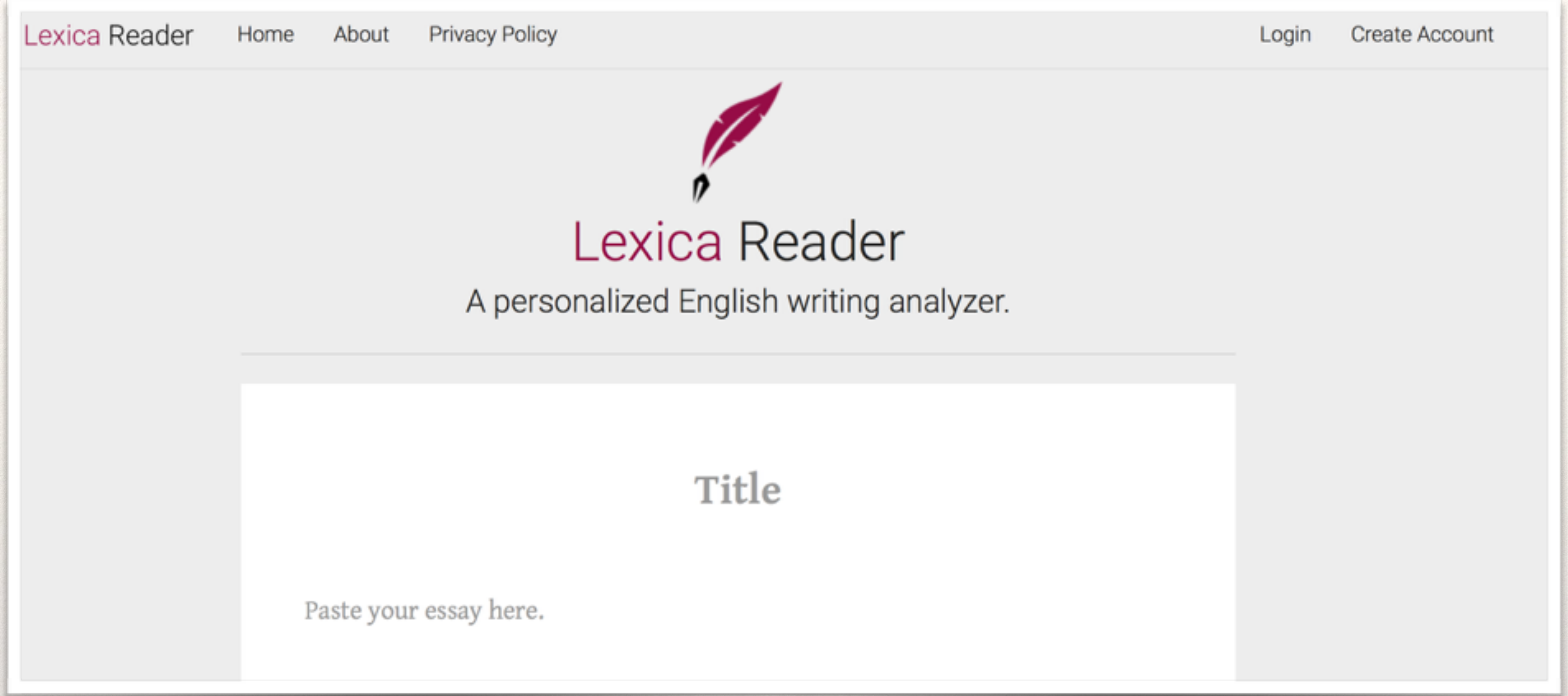
Because grading should be easy.

__Vinay Ayyala, James Evans, Richard Freling__, Alexandra Kubiak, Gabriela Leichnitz

codePost

2014: COS 333

2015 - present: adopted by COS 126 for actual use in assignment and exam grading



Lexica Reader Home About Privacy Policy Login Create Account

Lexica Reader

A personalized English writing analyzer.

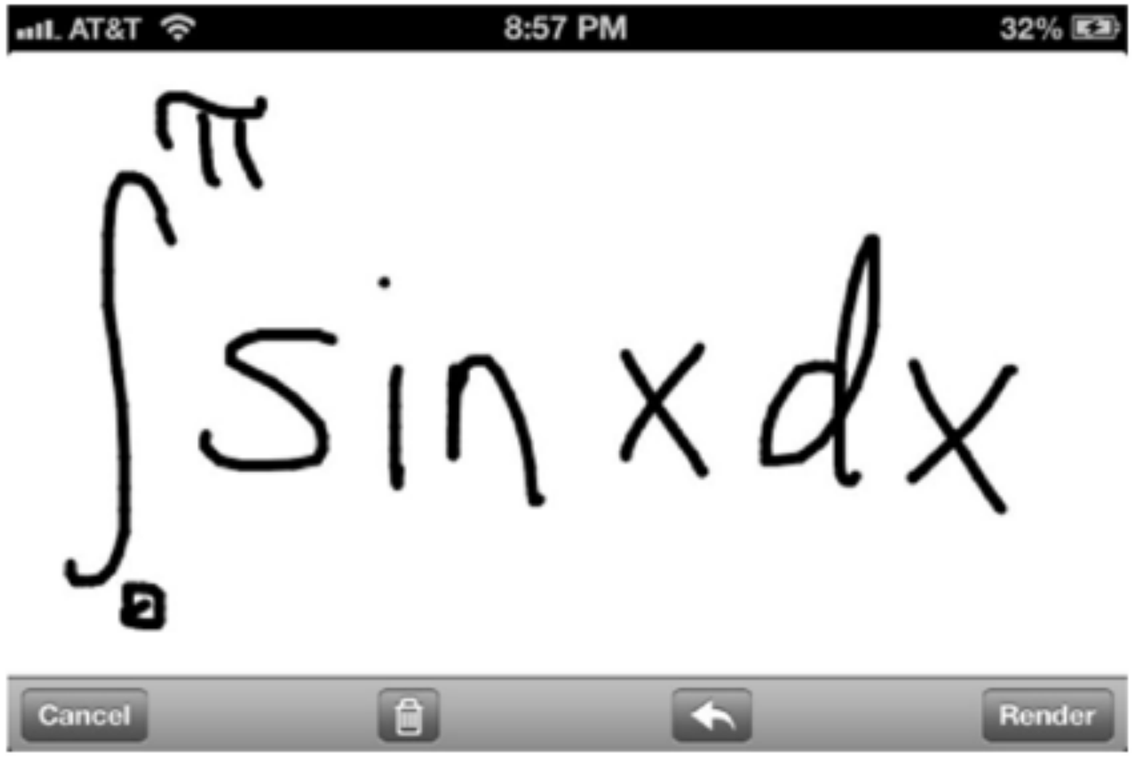
Title

Paste your essay here.

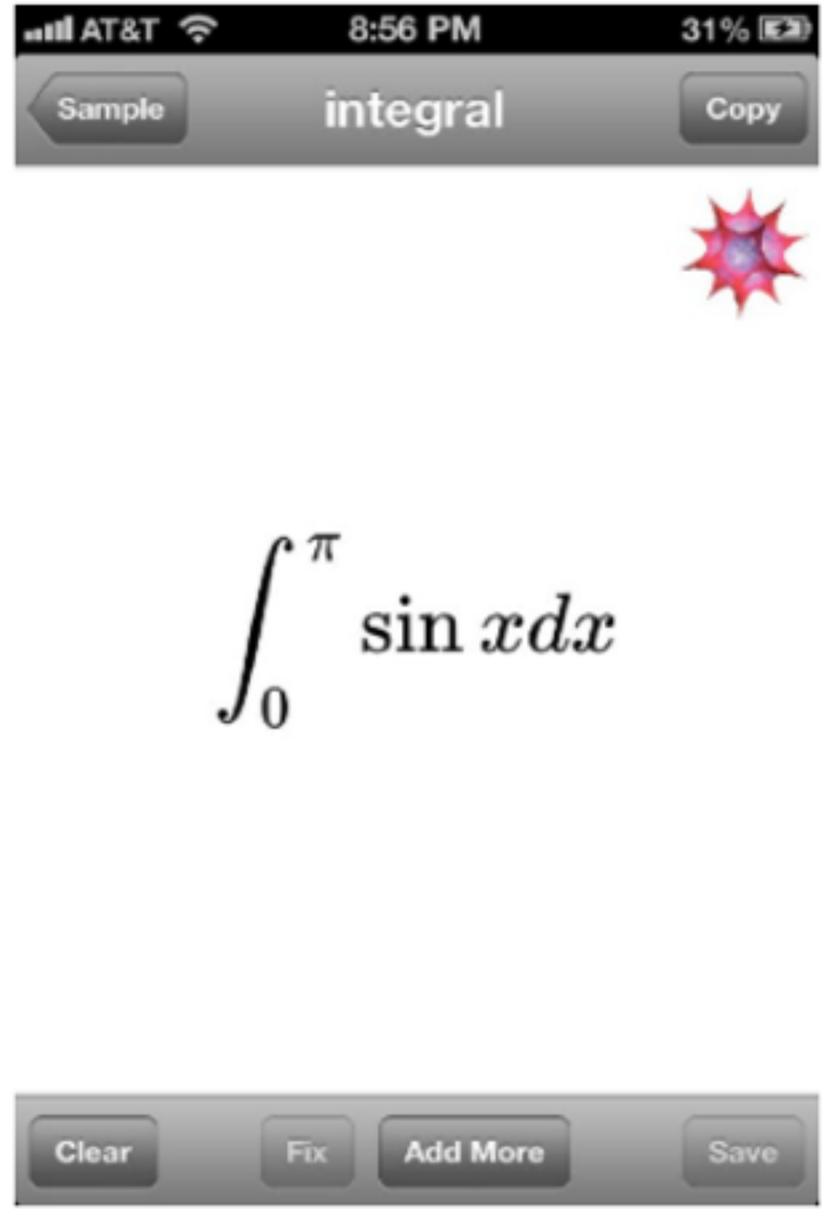
Diana Liao, Jerry Liu, Jonathan Tang

Lexica Reader

2015: COS 333



$$\int_0^\pi \sin x dx$$

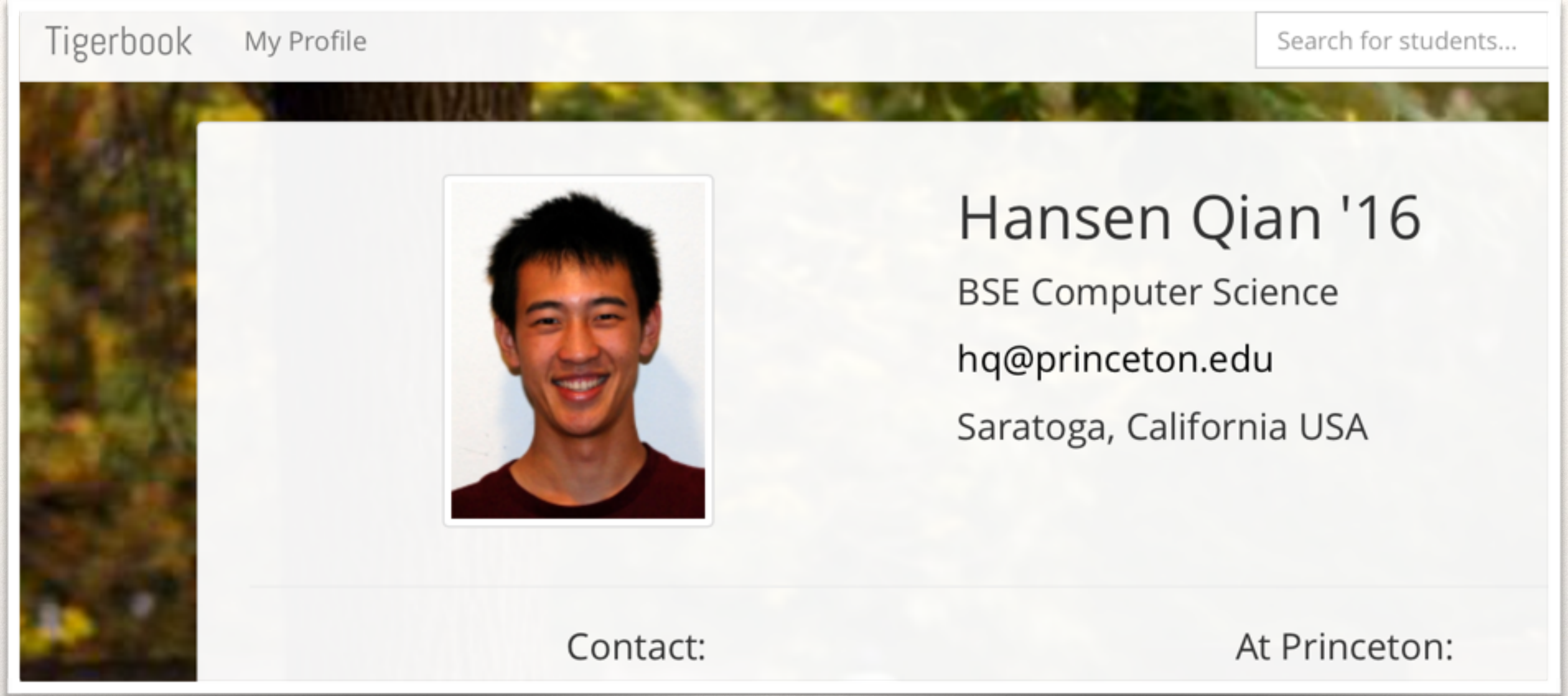


$$\int_0^\pi \sin x dx$$

Gina Triolo, Aman Sinha, Alex Baker

TeXscribe

2013: COS 333
2015: http://apple.co/1NO5ZwS
… the only(?) project to draw audible gasps, reactions of “holy $#!7”, etc. when demo’ed.



Hansen Qian '16

BSE Computer Science

hq@princeton.edu

Saratoga, California USA

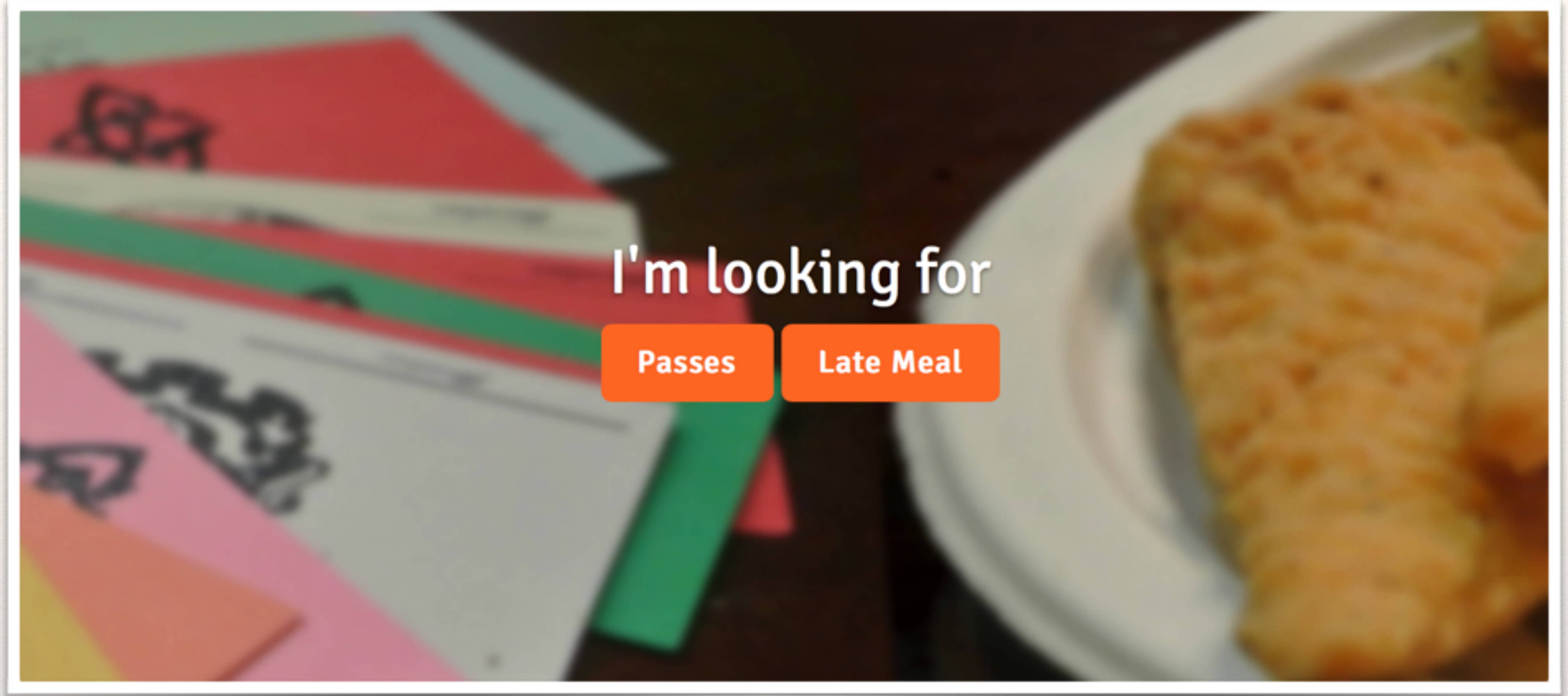
Contact:

At Princeton:

Ivo Crnkovic-Rubsamen, Hansen Qian, Rohan Sharma

Tigerbook

2014: COS 333
2014 - present: tigerbook.org
the most-used (and perhaps stalker-est?) recent project in production on campus

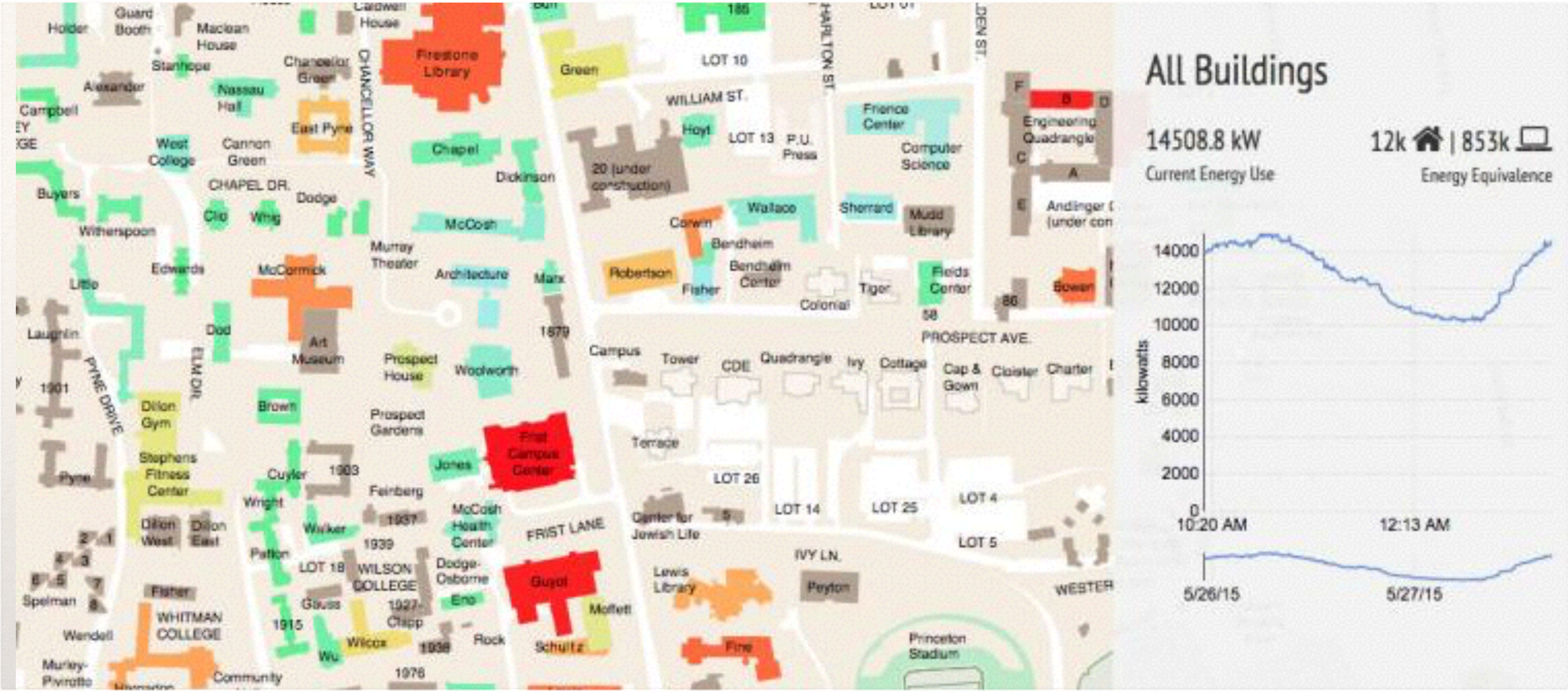


Nihar Madhavan, Utsarga Sikder, Jun Takahashi

Passes for Late Meal

2014: COS 333

… possibly the only project to get forcibly shut down by the administration? (*Prince* story: http://bit.ly/1SAnmaZ)



All Buildings

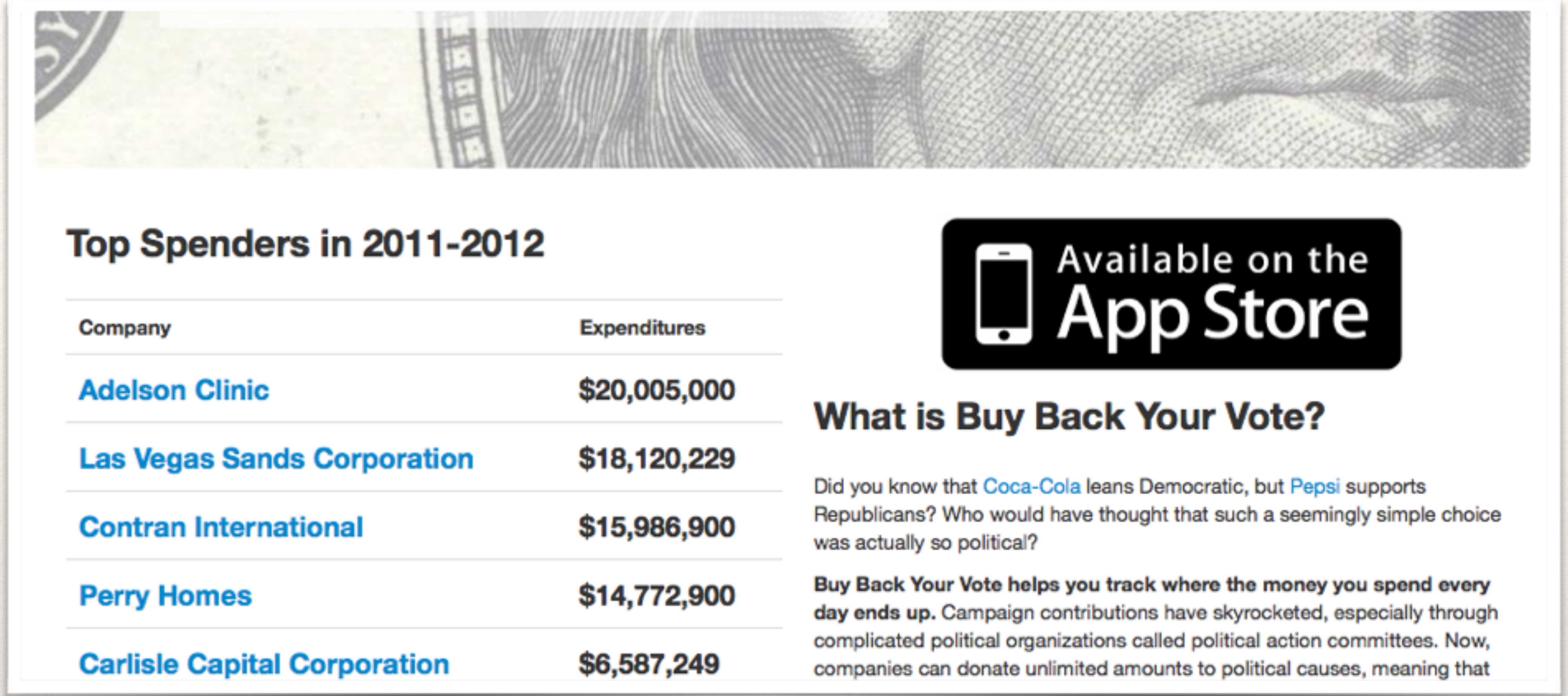
14508.8 kW
Current Energy Use

12k 🏠 | 853k 💻
Energy Equivalence

Adam Gallagher, Annie Lu, Josh Bocarsly

Princeton Energy Heatmap

2015: COS 333
2015 - present:
https://sustain.princeton.edu/lab/live-data
http://52.1.224.93



Top Spenders in 2011-2012

Company	Expenditures
Adelson Clinic	$20,005,000
Las Vegas Sands Corporation	$18,120,229
Contran International	$15,986,900
Perry Homes	$14,772,900
Carlisle Capital Corporation	$6,587,249

What is Buy Back Your Vote?

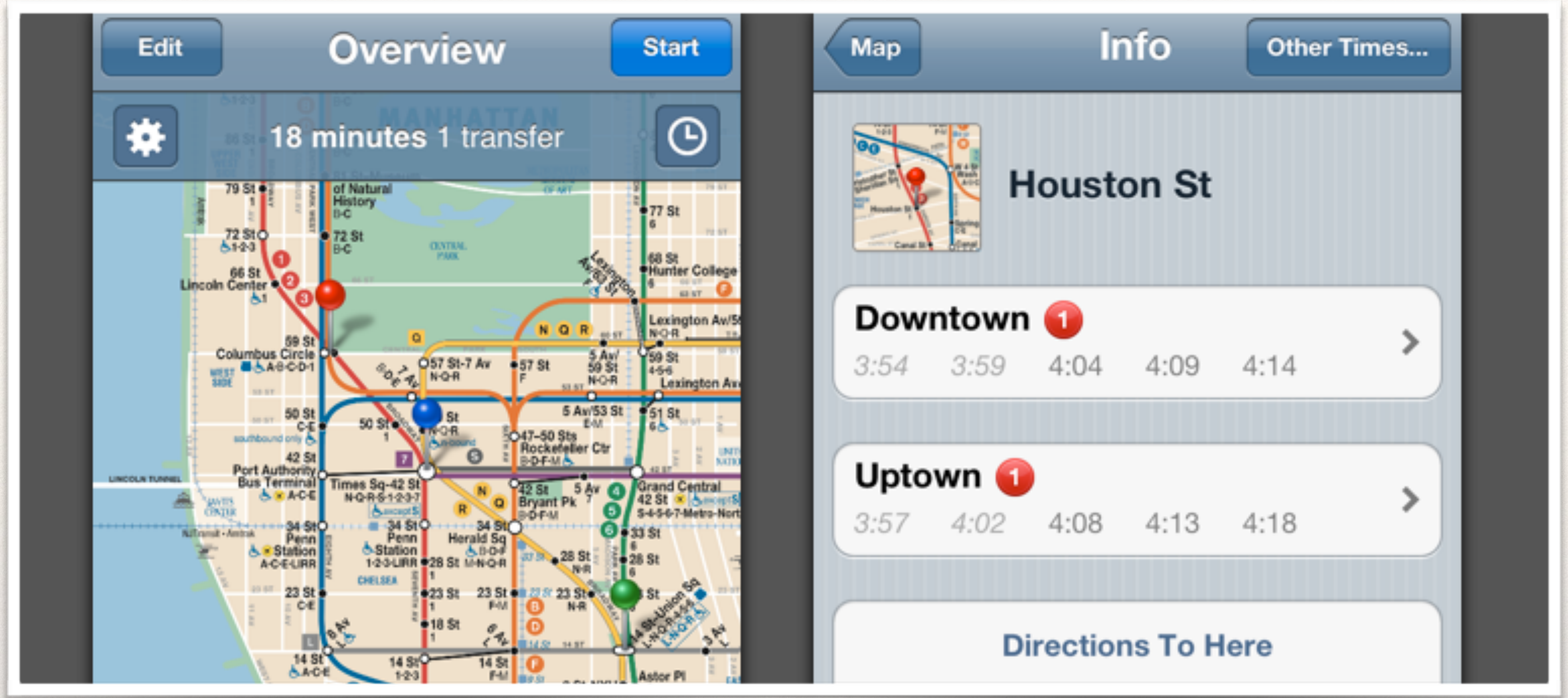
Did you know that Coca-Cola leans Democratic, but Pepsi supports Republicans? Who would have thought that such a seemingly simple choice was actually so political?

Buy Back Your Vote helps you track where the money you spend every day ends up. Campaign contributions have skyrocketed, especially through complicated political organizations called political action committees. Now, companies can donate unlimited amounts to political causes, meaning that

Tim Bauman, Matt Dolan, Margaret Fortney, Chris Kelly, Nathan Keyes

Buy Back Your Vote

2012: COS 333



Houston St

Downtown 1
3:54 3:59 4:04 4:09 4:14

Uptown 1
3:57 4:02 4:08 4:13 4:18

*Paul Cavallaro, **Adam Ernst**, Mark Limperis, Dzhelil Rufat*

iPhone Transit Maps

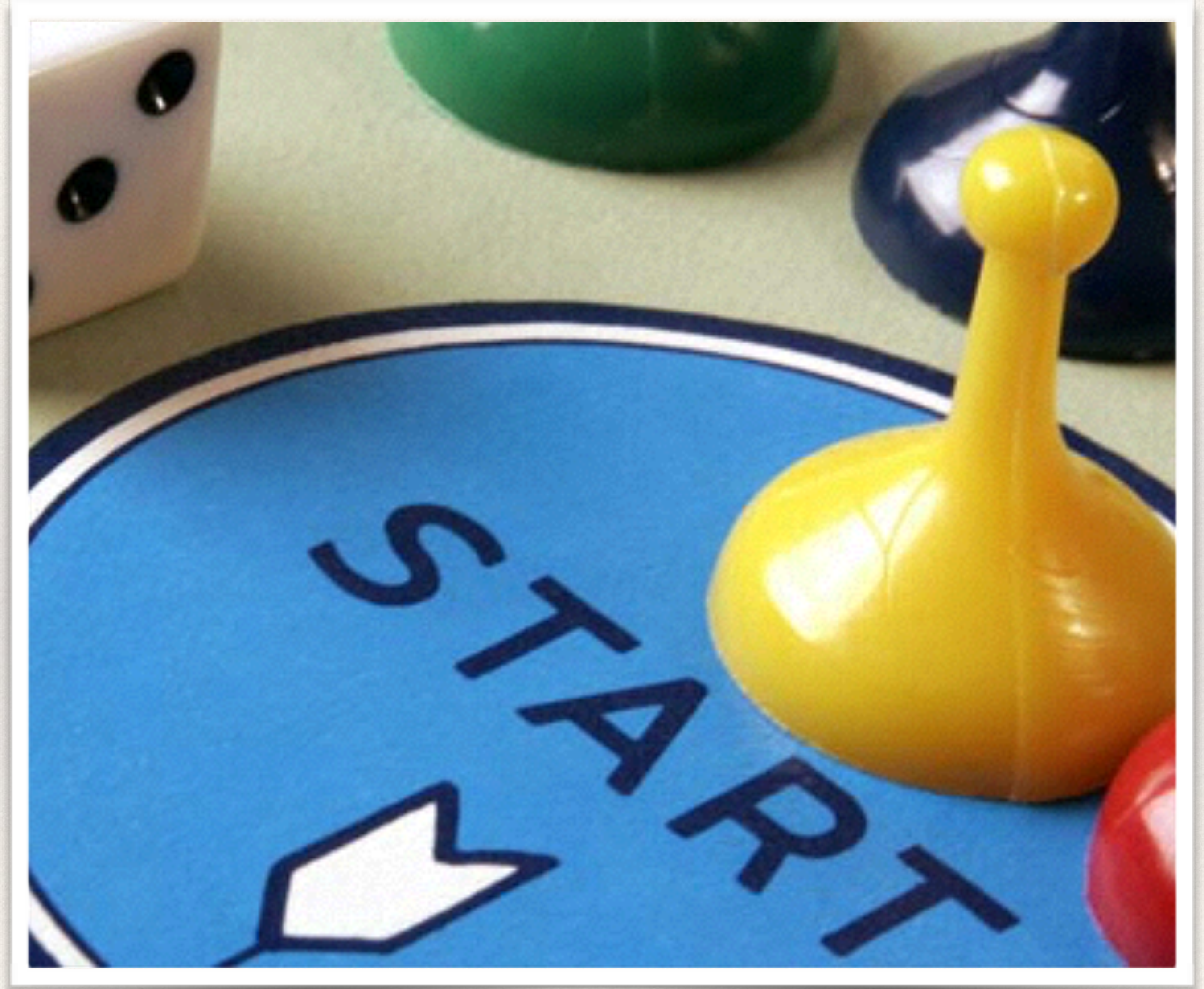
2008: COS 333
2008 - present: itrans.info
… rumor has it that it's the only COS 333 project to have paid for the rest of its creator's Princeton education.

So with all that in mind …

Think about potential projects;
start looking for teammates

- look at previous projects
- look around you, in your life
- check out client-suggested projects

By March 8, at the very latest, meet with me to be sure that idea is generally okay — come with one relatively firm idea, not several vague ones!

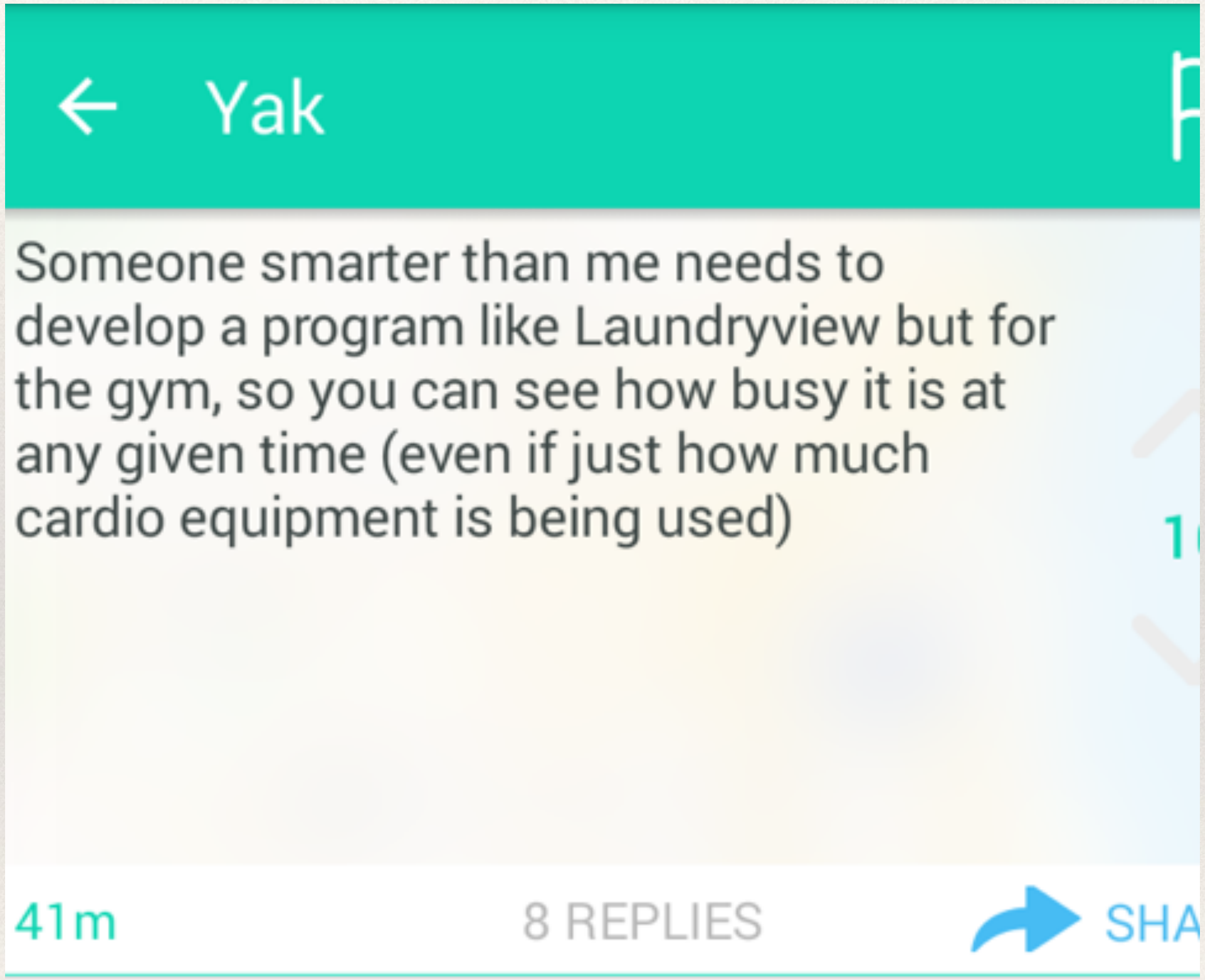


So with all that in mind …

Think about potential projects;
start looking for teammates

- look at previous projects
- **look around you, in your life**
- check out client-suggested projects

By March 8, at the very latest, meet with me to be sure that idea is generally okay — come with one relatively firm idea, not several vague ones!



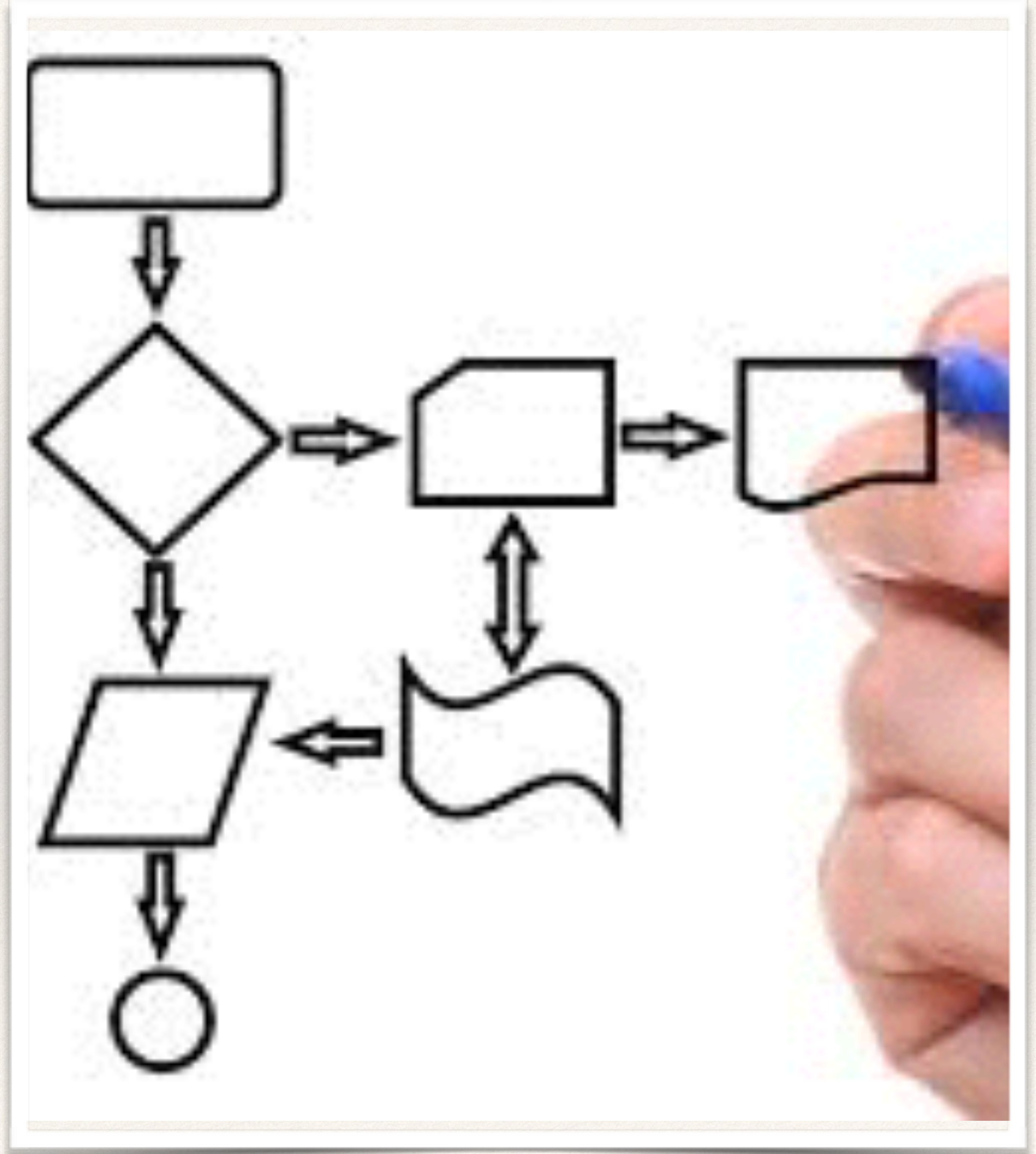
Someone smarter than me needs to develop a program like Laundryview but for the gym, so you can see how busy it is at any given time (even if just how much cardio equipment is being used)

First (real) Deliverable

- Design Document — due March 14 (M of spring break)
- Approximately 3-5 page project proposal, overviewing the goals of the project, the requirements, projected functionality, design, milestones, and project risks.
- The more thought you put into this, the smoother your project will start off!
- Even though this is due during spring break, aim to complete it earlier so that you can use the week to make real progress.

Have a *process* to make *progress*

- This is **not** a (good) process:
  - talk about project at lunch
  - hack some code together
  - test it a bit
  - fix the obvious bugs
  - repeat from the top until the semester ends

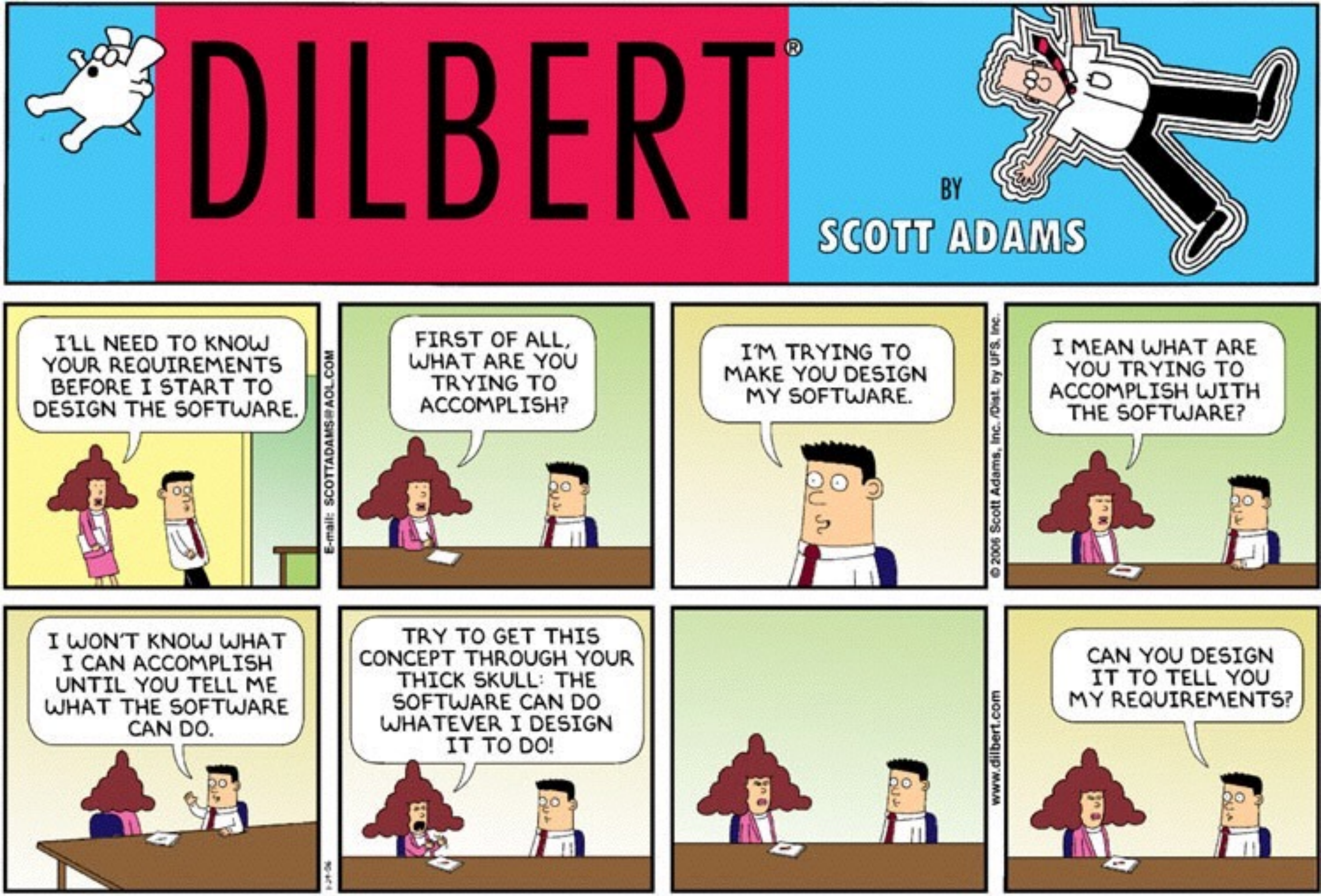


Do this from the start, and keep it up till the end:

- Keep scope and schedule in mind
- Keep a log of what you've done and what's up next on your plate
- Avoid a "big bang" project in all stages of planning
- Simplify: don't take on too big a job, don't try to do everything at once; do take "reasonable bites"
- Use source code control for **everything**.
- Test **everything**. Build automated tests from the beginning to perpetually keep your code in shape
- Remember that you have deliverables along the way!
- Remember that no battle plan survives the first encounter with the enemy.

Design Process

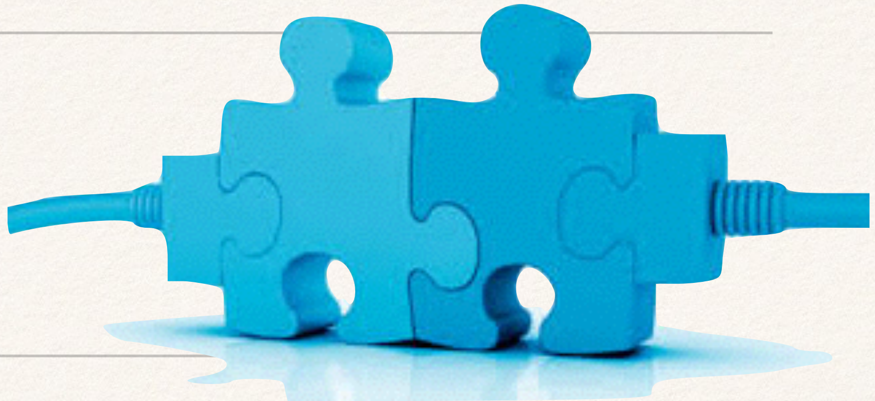
- Conceptual design: broadly, what are we doing?
  - sketches, scenarios, screenshots, storyboards
- Requirements
  - specifically, what does it do?
  - what are our options to make it work?
  - what are the constraints?
  - specify in written documentation as a “contract”


DILBERT®
BY SCOTT ADAMS
I'LL NEED TO KNOW YOUR REQUIREMENTS BEFORE I START TO DESIGN THE SOFTWARE.
E-mail: SCOTTADAMS@AOL.COM
FIRST OF ALL, WHAT ARE YOU TRYING TO ACCOMPLISH?
I'M TRYING TO MAKE YOU DESIGN MY SOFTWARE.
© 2006 Scott Adams, Inc./Dist. by UFS, Inc.
I MEAN WHAT ARE YOU TRYING TO ACCOMPLISH WITH THE SOFTWARE?
I WON'T KNOW WHAT I CAN ACCOMPLISH UNTIL YOU TELL ME WHAT THE SOFTWARE CAN DO.
TRY TO GET THIS CONCEPT THROUGH YOUR THICK SKULL: THE SOFTWARE CAN DO WHATEVER I DESIGN IT TO DO!
www.dilbert.com
CAN YOU DESIGN IT TO TELL YOU MY REQUIREMENTS?

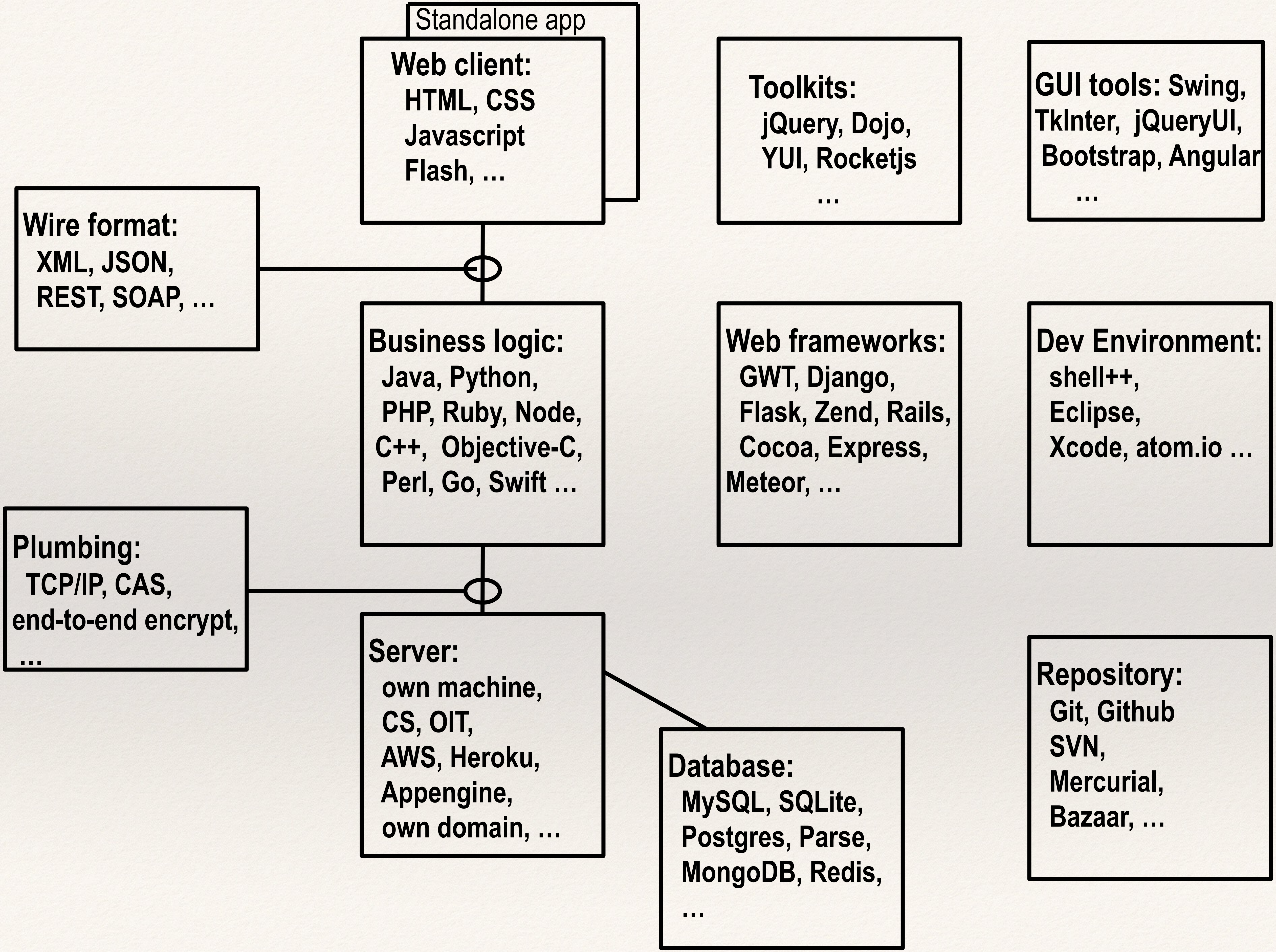
Design Process (cont'd)

- Architecture design: how are we doing this?
    - make versus buy? acquire necessary resources?
    - specifications, diagrams, prototypes, tech. details
    - define components or subsystems
    - define interface for interactions between components

Interfaces



- Contracted boundary between two parts of program
  - What data/control flow transformation occurs there?
  - Who manages resources, especially shared state?
- Hide implementation details behind interfaces
  - Encapsulation allows later design changes without ripple effects through entire system
  - Database, data representation, file format, algorithms
- "I wish we had done interfaces better" - annual comment



Standalone app

Web client:
HTML, CSS
Javascript
Flash, …

Toolkits:
jQuery, Dojo,
YUI, Rocketjs
…

GUI tools: Swing,
TkInter, jQueryUI,
Bootstrap, Angular
…

Wire format:
XML, JSON,
REST, SOAP, …

Business logic:
Java, Python,
PHP, Ruby, Node,
C++, Objective-C,
Perl, Go, Swift …

Web frameworks:
GWT, Django,
Flask, Zend, Rails,
Cocoa, Express,
Meteor, …

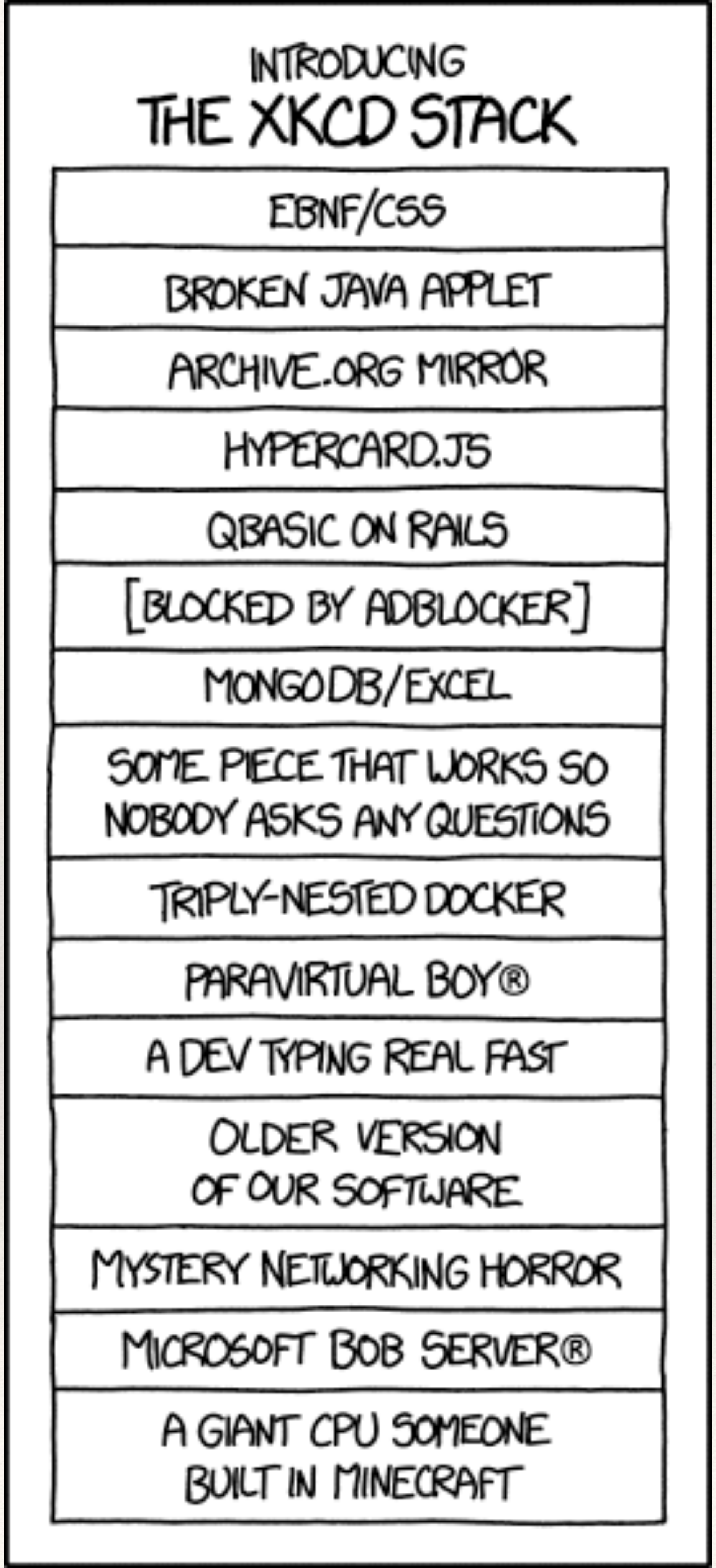
Dev Environment:
shell++,
Eclipse,
Xcode, atom.io …

Plumbing:
TCP/IP, CAS,
end-to-end encrypt,
…

Server:
own machine,
CS, OIT,
AWS, Heroku,
Appengine,
own domain, …

Database:
MySQL, SQLite,
Postgres, Parse,
MongoDB, Redis,
…

Repository:
Git, Github
SVN,
Mercurial,
Bazaar, …



This site requires Sun Java 6.0.0.1 (32-bit) or higher. You have Macromedia Java 7.3.8.1¾ (48-bit). Click here [link to java.com main page] to download an installer which will run fine but not really change anything

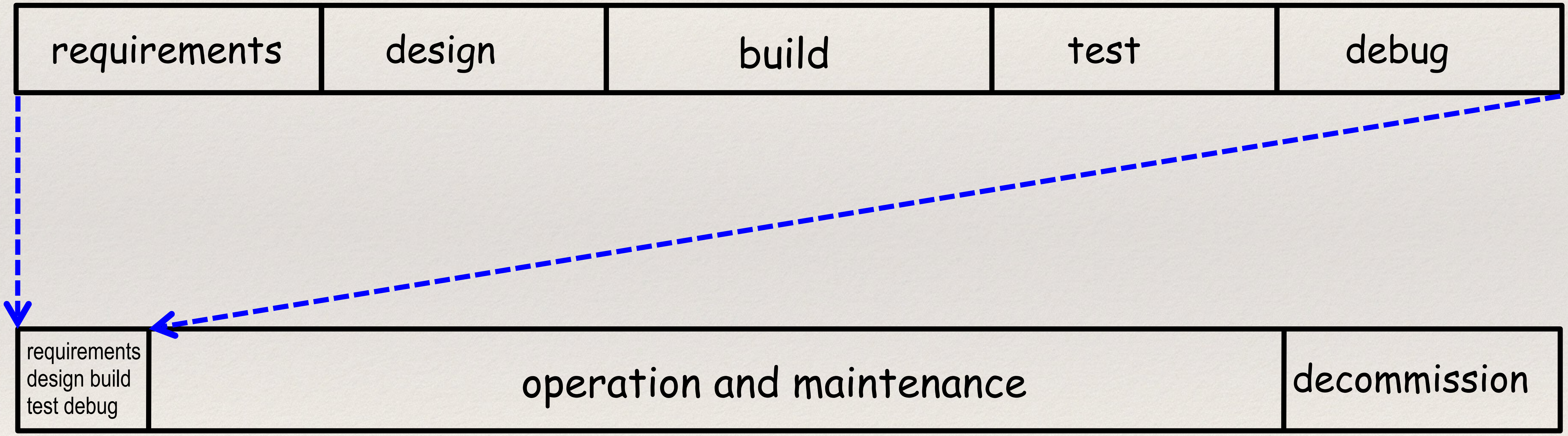
http://xkcd.com/1636/

Design Process (cont'd)

- Implementation (finally!)
  - make prototypes, establish end-to-end plumbing
  - test as you go, keeping the system perpetually working to some degree or another
  - deliver in stages, so that each does something
  - get real users to use the system as early as possible

Did somebody say “finally!”?

- The Real Software Lifecycle



requirements | design | build | test | debug

requirements design build test debug | operation and maintenance | decommission