

# Software Engineering for Data Scientists

## *Class Project Part 1 – “Dating”*

David Beck<sup>1,2</sup>, Joseph Hellerstein<sup>1,3</sup>, Jake VanderPlas<sup>1,4</sup>

Jay Garlapati<sup>3</sup>

<sup>1</sup>eScience Institute

<sup>2</sup>Chemical Engineering

<sup>3</sup>Computer Science Engineering

<sup>4</sup>Astronomy

The University of Washington

April 20, 2017



# Objectives For Today

- Introduce expectations for the project
- Initiate team formation – “dating”



# Class project overview

- Collaborative software engineering experience
  - Teams of 3 to 4 with 4 being optimal
    - Prefer *teams with diversity*
  - Develop project in Git w/ GitHub
    - Not Google docs or Dropbox



# Class project overview

- Collaborative software engineering experience
  - Design (use cases, component specification)
  - Documentation (how to, docstrings)
  - Style (PEP8, pylint)
  - Coding, testing & milestones
  - Standup & code reviews

<http://uwseds.github.io>



# Project Type 1:

## *Answer “Research” Questions*

- Problem statement: Answer two to three questions of business or scientific relevance
  - Use a Jupyter notebook and supporting python files
- Example
  - [Climate Police](#): Analyze effects of pollution on the planet.



## Capstone Project Type 2: *Create Reusable Data*

- Problem statement: Create data repository with tools (e.g., search, visualization, analytics)
- Example
  - [Car2Know](#): Provide car rental data to users of Car2Go (e.g., for planning trips)

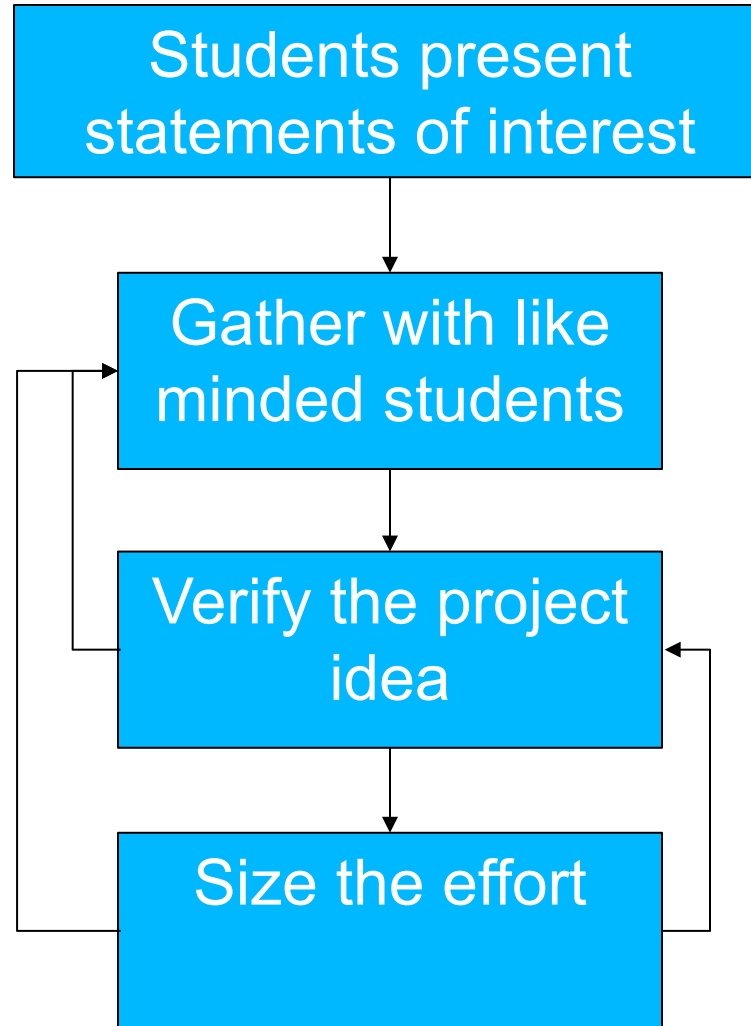


## Project Type 3: *Create a Tool*

- Problem statement: Solve a problem common to many users
  - Don't re-invent the wheel
- Example
  - [BioReactor Data Logging](#) – Monitor and publish data from BioReactor experiments



# Getting Started





# Student Summary

- Topics of interest
- Data you have access to NOW
  - How much you've used the data
  - Code you have to access the data
  - How clean the data are

**Do this in 1 minute!**



# Verify the Project Idea

- Is there an unmet need (i.e. no code already exists)?
- Clarity about the project type?
- Consensus on the problem being solved.
- Do you have data that can solve the problem?



## More on the Data

- At least two non-trivial data sets
- Data need to be combined, joined, merged, etc. to answer the scientific questions
- Have access to the data NOW!



# Some Public Data

- <http://drugbank.ca>
- <http://toxnet.nlm.nih.gov>
- <https://data.seattle.gov/Transportation/Traffic-Flow-Counts/7svg-ds5z>
- <https://www.divvybikes.com/data>
- [http://www.nyc.gov/html/tlc/html/about/trip\\_record\\_data.shtml](http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml)
- <https://www.kaggle.com>
- [Pronto bike data](#)
- [American Fact Finder Data](#)
- [European union data](#) (World bank)
- [Russian federation data](#) (World bank)
- [China data](#) (World bank)



# Some Third Party Tools

- What third party tools can / might you leverage?
  - Sci Kit Learn
    - <http://scikit-learn.org/stable/>
  - Lasagne
    - <http://lasagne.readthedocs.org/en/latest/>
  - Bokeh
    - <http://bokeh.pydata.org/en/latest/>



# Grading Rubric

- Design (use cases, component specification)
- Documentation (how to, docstrings)
- Style (PEP8, pylint)
- Coding, testing & milestones
- Standup
- Project presentation



# Data! Data! Data!

- At least two non-trivial data sets
- Data need to be combined, joined, merged, etc.

## Think about your data NOW!

