

Recitation 1

Brandon Rozek

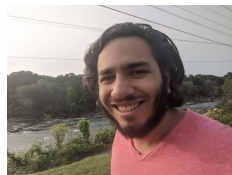
Rensselaer Polytechnic Institute, Troy, NY, USA

January 2022

About Me

Contact Information:

- Office Hours: T 4-6pm / F 12-2pm
- Personal Email: roze kb@rpi.edu
- Staff Email: psoftstaff@cs.lists.rpi.edu



About Recitations

- Recitations are a newly added feature to this class.
 - You have the opportunity to drive the discussion.
 - Feel free to interrupt me at any moment via voice or chat.
-
- Currently thinking about going over some initial parts of the homework
 - As well as mixing in some stories working on software with other people.

Software requires maintenance

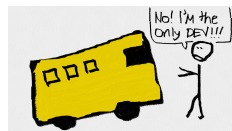
- The world changes often! New feature requests, updates to operating systems that break your application
- Therefore for any software to exist long-term, changes need to be made.

How expensive are those changes?

- Is the code documented?
- Does it follow a specification?
- Follow best practices like modularity?

Bus Factor

- The bus factor is a measurement of risk if a certain team member "got hit by a bus"
- Now it doesn't have to necessarily be death, but projects members may come and go due to resignations, reallocation of time, etc.



In this course, we hope to teach you techniques and methods which can help you spread the work among your team members.

- Setting up Eclipse
- Git Version Control
- JavaDoc for elegant documentation
- JUnit for testing

Eclipse Setup

Git Version Control

- Decentralized tool for checkpointing code
- Controversial: A method of having *backups*
- Allow for multiple people to work on the same code base, especially with *branches*.

Common Git Commands

- git clone: Copies an existing repository
- git pull: In an existing repository, fetch and merge updates
- git add: Adds a file to *staging*
- git commit: Creates a new snapshot based on whats in staging
- git push: Pushes the local copy to a remote site

- Unit testing is a way to boost confidence that your application follows a specification
- Test driven development is the philosophy that you should write tests first and then write your software in order to pass these tests
- Often a good way to split up the development

Any Questions?