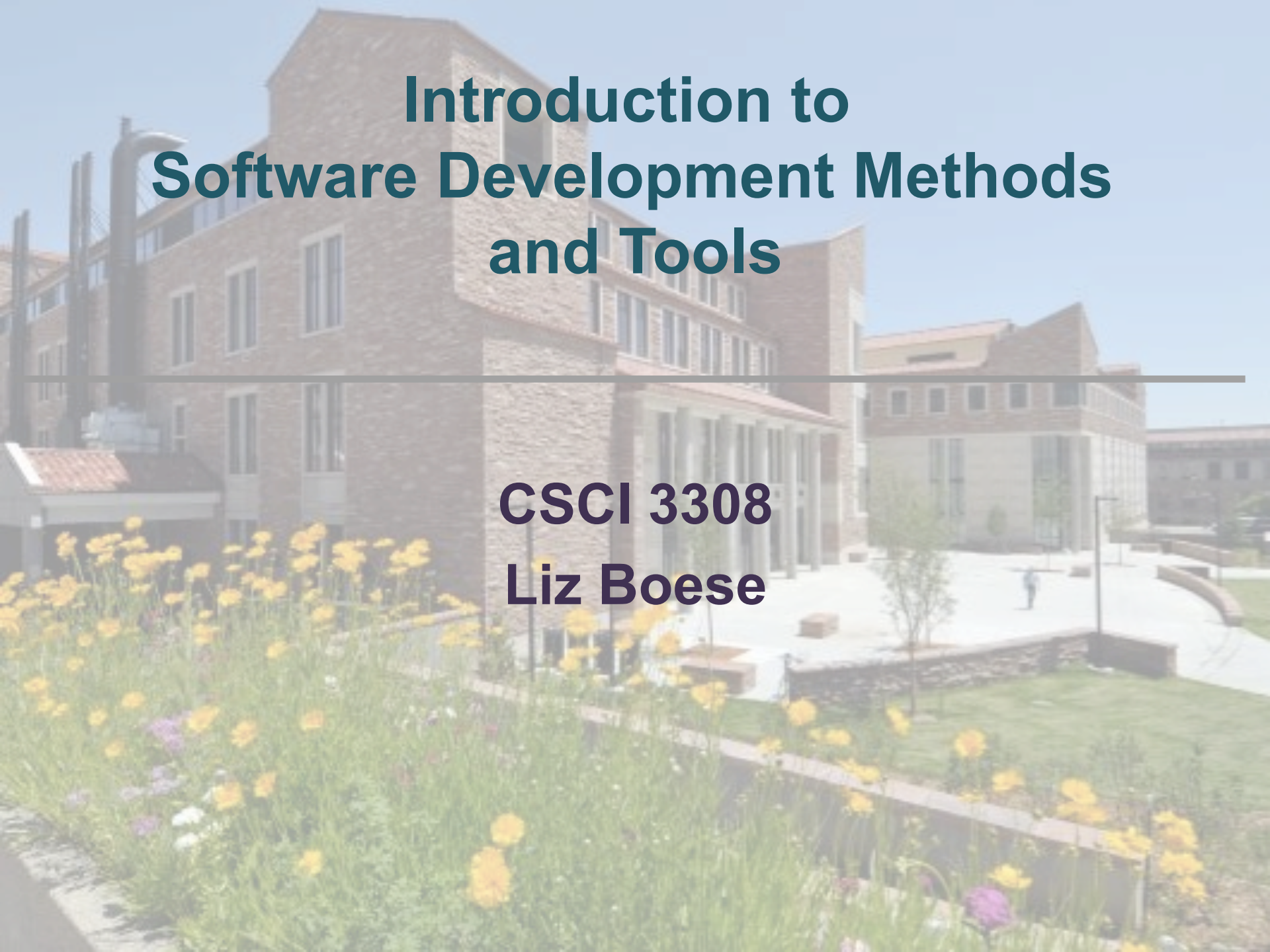


Introduction to Software Development Methods and Tools

CSCI 3308

Liz Boese

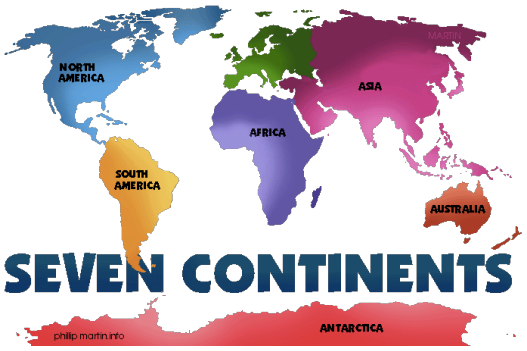
Top 5 Reasons to Take this Course

- Be a more efficient programmer
  - Learn what tools/methods are available and when to use them
- Be a more effective programmer
  - Create code that does what it is supposed the way its supposed to do it
- Be better at working with others
  - Learn to use collaborative tools and conventions
- Create more understandable code
  - Learn how to organize (and re-organize) your code

US

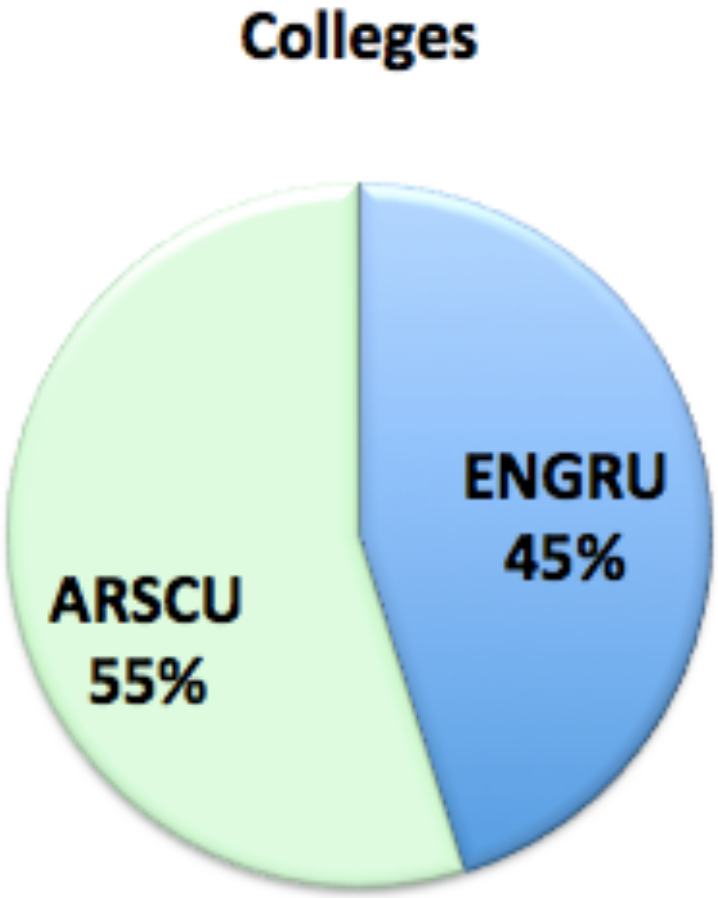
Introductions

- Liz Boese
  - Office: ECOT 733
  - Office Hours in Moodle
  - Background:
    - » Management Consulting
    - » Teaching CS
    - » Start-ups
    - » Author
    - » Speaker
- TAs
  - See website
- LAs
  - Available in CSEL, …

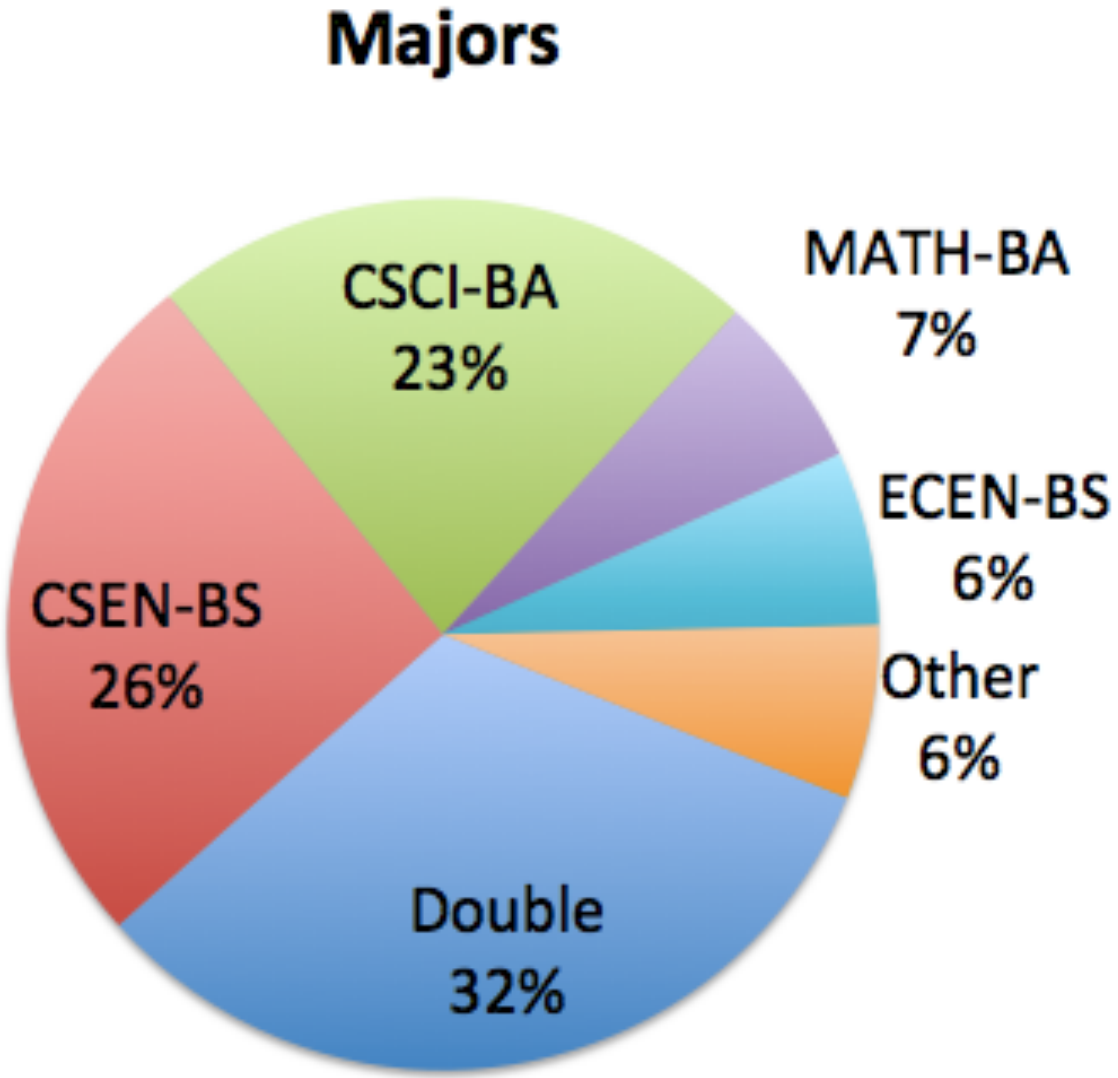


Who are you and Why are You Here?

YOU



YOU



COURSE MATERIALS

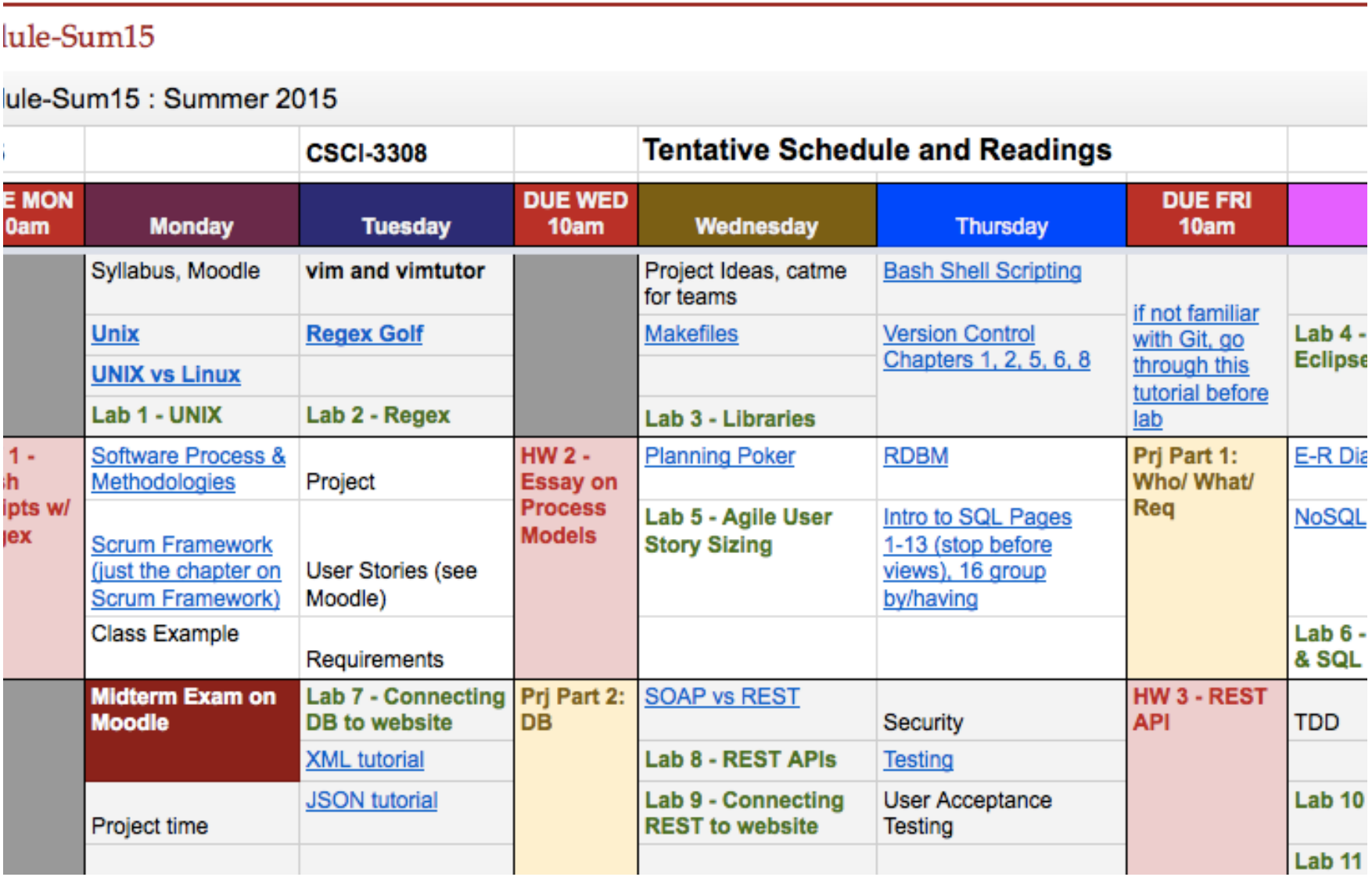
Moodle

- Find most stuff here
  https://sites.google.com/a/colorado.edu/csci-3308

- **Login to Google:** with your colorado.edu account

I found that the authentication doesn't work on chrome if you have multiple google accounts setup unless you go to the bottom of the page, click "sign in" and switch it to your school google account (even though I authenticated with the school account, it apparently defaults to viewing the page with your primary chrome account).

Schedule



lule-Sum15

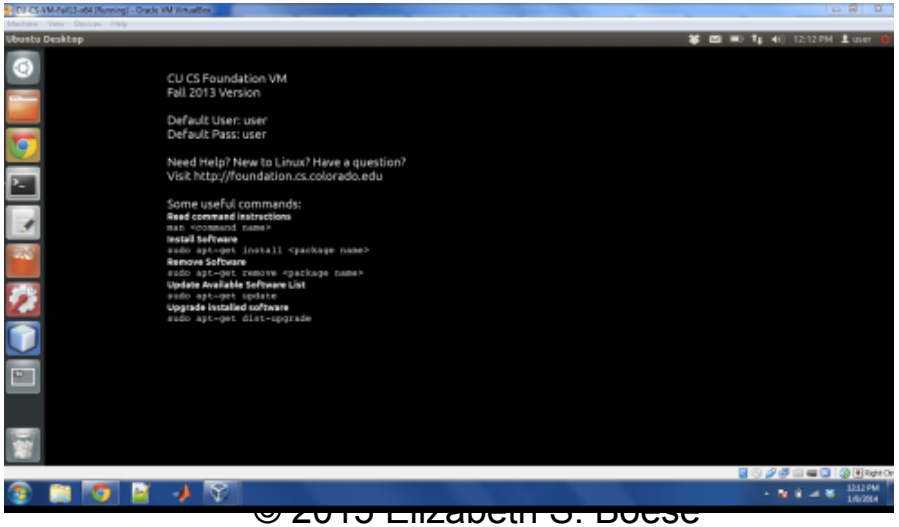
lule-Sum15 : Summer 2015

		CSCI-3308		Tentative Schedule and Readings			
E MON 0am	Monday	Tuesday	DUE WED 10am	Wednesday	Thursday	DUE FRI 10am	
	Syllabus, Moodle	vim and vimtutor		Project Ideas, catme for teams	Bash Shell Scripting	if not familiar with Git, go through this tutorial before lab	
	Unix	Regex Golf		Makefiles	Version Control Chapters 1, 2, 5, 6, 8		Lab 4 - Eclipse
	UNIX vs Linux						
	Lab 1 - UNIX	Lab 2 - Regex		Lab 3 - Libraries			
1 - h ipts w/ ex	Software Process & Methodologies	Project	HW 2 - Essay on Process Models	Planning Poker	RDBM	Prj Part 1: Who/ What/ Req	E-R Dia
	Scrum Framework (just the chapter on Scrum Framework)	User Stories (see Moodle)		Lab 5 - Agile User Story Sizing	Intro to SQL Pages 1-13 (stop before views), 16 group by/having		NoSQL
	Class Example	Requirements					Lab 6 - & SQL
	Midterm Exam on Moodle	Lab 7 - Connecting DB to website	Prj Part 2: DB	SOAP vs REST	Security	HW 3 - REST API	TDD
		XML tutorial		Lab 8 - REST APIs	Testing		
	Project time	JSON tutorial		Lab 9 - Connecting REST to website	User Acceptance Testing		Lab 10
							Lab 11

Virtual Machine (VM)

- https://foundation.cs.colorado.edu/sde/vm-obtain.html
- Connect with your DropBox account
- Help sessions available to set up your computer
- Get the latest version of the VM!
- Get the 3308 course setup for your VM

```
sudo apt-get install cu-cs-csci-3308
```



Dropbox

- Use Dropbox from the VM!
  - Easy to transfer files between the VM and your computer
  - If anything goes wrong with your VM or your computer, your files are on the interwebs!

Course Overview

- Intro to Technologies
  - Learn background
  - Apply methods
  - Apply tools
- Programming
  - Semester long programming project of your choice
- Explore
  - Apply methods and tools to your project program
  - Will work with tools in class so bring your laptop

Pragmatics

- Course Web Site : https://sites.google.com/a/colorado.edu/csci-3308
  Required Readings are on website (see schedule)

- Grades
- Project
- Homeworks
- Labs
- Calendar
- Review Syllabus

Safari Bookshelf available through Norlin Library

Safari Bookshelf through ACM (Join as a student for $42/year

Lectures

- Bring your computer to class!
  - Exercises
- Read BEFORE you come to class
  - Book and/or assigned readings
  - Slides
- Lecture will highlight points in the slides and cover some of the more difficult material in more detail, exercises…

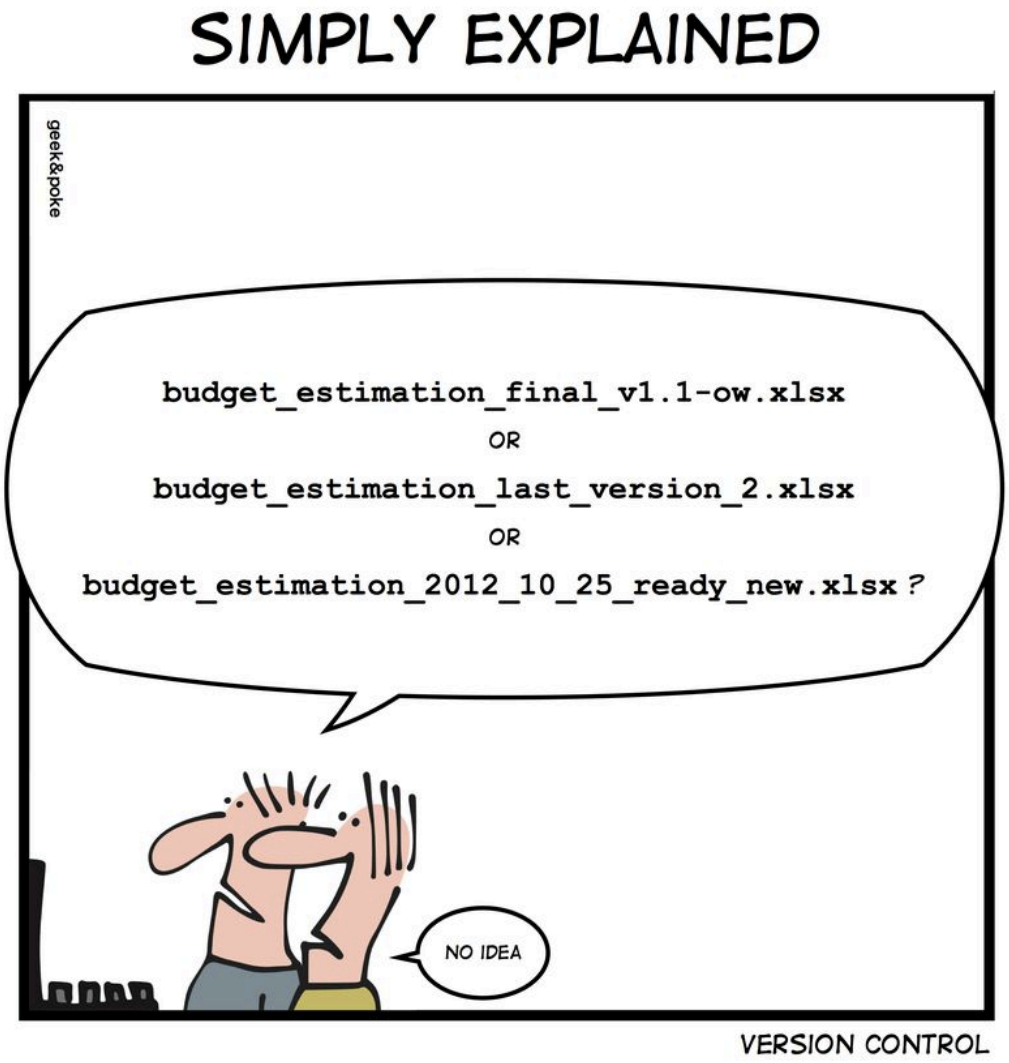
TOPICS

Types of Methods/Tools

- Project Management
- Code Improvement
- Testing
- Continuous Improvement
- Integrated Development Environments
- Deployment Environments

Project Management

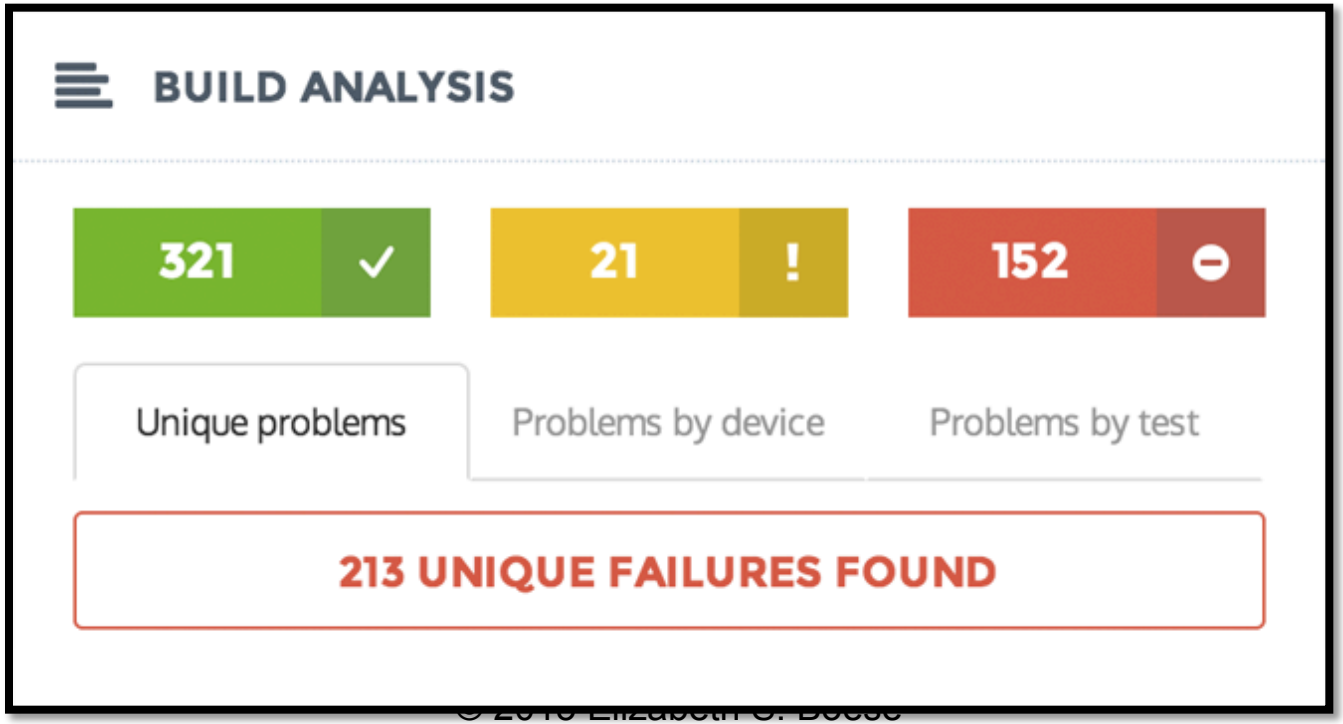
- **Version Control** (a.k.a. source code control) and Configuration Management
  - Keep track of what you are doing
  - Automated protection
  - Support for team interactions
  - Examples: RCS, CVS, SVN, GIT



Idea from Jen Simmons and John Albin Wilkins during episode #40 of "Web Ahead" about Git: http://5by5.tv/webahead/40

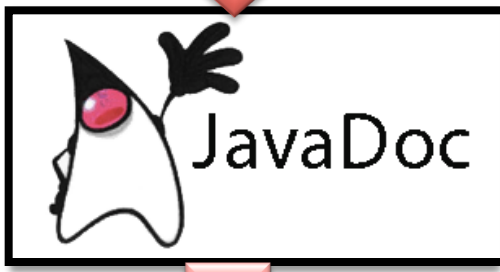
Project Management

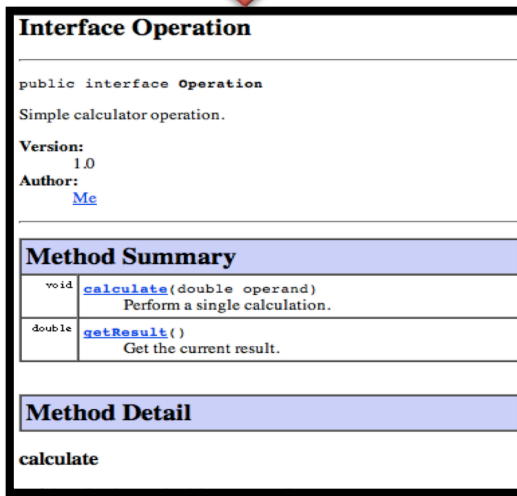
- Automated Build Tools
  - Once you get beyond a single file (which you should do) you need help rebuilding your executable
  - Can do many other tasks for you
  - Examples: Make, Ant



Project Management (cont)

- Automated Documentation
  - Document as you code
  - Generates overview documentation from program structure
  - When run as part of the automated build process, keeps your documentation in sync with the code
  - Extracts documentation from code
    - » Special comments
    - » Structural features based on code elements such as #include
  - Examples: Javadoc, Doxygen





Code Improvement Tools: Checkers

- Static vs. Dynamic
  - Static – applied to source code
  - Dynamic – applied to executing code
- Performance Analysis
  - Which kind do you think? Static or Dynamic…
  - Help identify hotspots for CPU, network, memory usage
- Correctness Analysis
  - Analyze data and control flow looking for probable bugs

Code Improvement: Debugging

- There are strategies that make for more efficient debugging
- Debugging tools are embedded in every integrated development environment
- Don’t use print statements!

Continuous Improvement

- Refactoring
  - Strategic code improvement without changing what code does
- Reengineering
  - Create new system that does same thing as original for some age related reason:
    - » To use new language
    - » For new platform
    - » To take advantage of new technologies
- We'll focus on refactoring and cover reengineering as time permits

Why would you do this?

Why would you **NOT** do this?

Do companies do this?

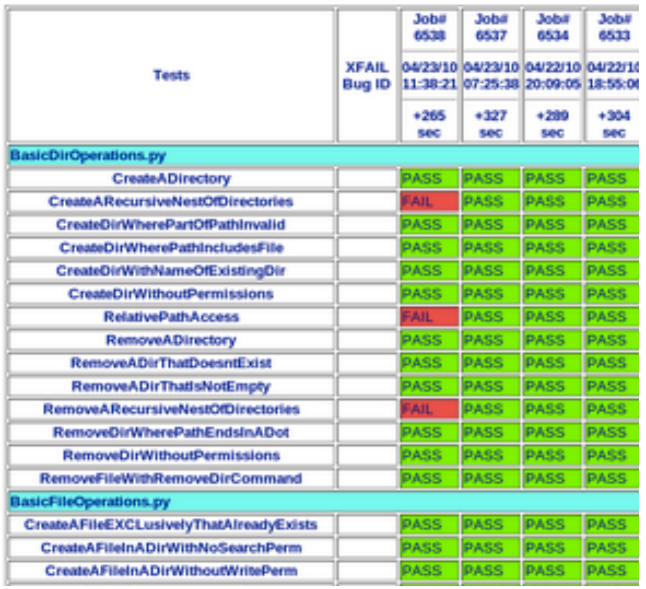
Case Study

- Reengineering Example
  - Company hosts their software and leases to other businesses
  - Started with 

  - Eventually got to a point where they disliked the restrictions of Zend and morphed the code to be totally custom.
  - Each business had their own requirements that required extra custom development, leading to a mess *(lots of #ifdefine COMPANYNAME. take 4448 to learn how to fix)*
  - Became extremely difficult to release a new version of the base product to all client instances.

Testing Methods/Tools

- Structural vs. Functional (white box vs black box)
  - Functional tests desired behavior
  - Structural tests based on code structure with goal of exercising code on all paths
  - A combination of both is most effective in catching bugs
- Unit Testing
  - Applied by programmer as coding progresses
  - Supported by tools that are frequently embedded in development environments
- Integration Testing
  - Making sure the parts work together as planned



Tests	XFAIL Bug ID	Job# 6538 04/23/10 11:38:21 +265 sec	Job# 6537 04/23/10 07:25:38 +327 sec	Job# 6534 04/22/10 20:09:05 +289 sec	Job# 6533 04/22/10 18:55:06 +304 sec
BasicDirOperations.py					
CreateADirectory		PASS	PASS	PASS	PASS
CreateARecursiveNestOfDirectories		FAIL	PASS	PASS	PASS
CreateDirWherePartOfPathInvalid		PASS	PASS	PASS	PASS
CreateDirWherePathIncludesFile		PASS	PASS	PASS	PASS
CreateDirWithNameOfExistingDir		PASS	PASS	PASS	PASS
CreateDirWithoutPermissions		PASS	PASS	PASS	PASS
RelativePathAccess		FAIL	PASS	PASS	PASS
RemoveADirectory		PASS	PASS	PASS	PASS
RemoveADirThatDoesntExist		PASS	PASS	PASS	PASS
RemoveADirThatIsNotEmpty		PASS	PASS	PASS	PASS
RemoveARecursiveNestOfDirectories		FAIL	PASS	PASS	PASS
RemoveDirWherePathEndsInADot		PASS	PASS	PASS	PASS
RemoveDirWithoutPermissions		PASS	PASS	PASS	PASS
RemoveFileWithRemoveDirCommand		PASS	PASS	PASS	PASS
BasicFileOperations.py					
CreateAFileEXCLusivelyThatAlreadyExists		PASS	PASS	PASS	PASS
CreateAFileInADirWithNoSearchPerm		PASS	PASS	PASS	PASS
CreateAFileInADirWithoutWritePerm		PASS	PASS	PASS	PASS

Integrated Development Environments

- IDEs have been developed to
  - Support team work
  - Improve developer efficiency
  - Support hierarchical development
- IDEs usually include tools from each of the areas
- Examples: Microsoft Visual Studio, Eclipse, Cocoa Touch for iOS, sublime