Updated for Spring 2016.

CS314: Software Engineering

James M. Bieman
Computer Science Department
Colorado State University

Introductions

- Instructor 1. Prof. James (Jim) Bieman
  - Phone: (970) 491-7096.
  - Office: Room 462, CS Building
  - Office Hours.TBA.
  - Email: *Bieman@ColoState.EDU*
- Instructor 2. Prof. Geri Georg
  - Office: Room 360 CS Building
  - Email: Geri.Georg@colostate.edu
- Teaching Assistant. Saksham Manchanda
  - Email: saksham.manchanda@gmail.com
  - Phone: 970-491-TBA Lab Hours: TBA

  cs314@cs.colostate.edu goes to all three of us
- Lectures: Mon/Wed/Fri 10:00 – 10:50AM.

Focus: Methods to Develop Commercial-scale Software.

- Large size systems: may be millions of lines long.
- Large development teams.
- Large or highly specialized user communities.

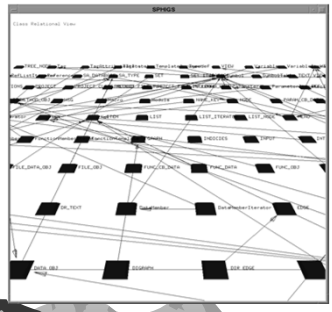
Large-scale software development

➔ Large-scale problems

Static connections in a small system

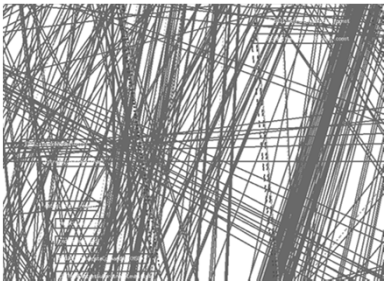
CIV
•60 Classes
• 10 KLOC

•150 Classes with 3rd party classes.



Detailed Call-Graph of CIV

Shows 3% of the entire call graph.



CS314 Objectives

You will learn to

- use software models (in UML) to
  - analyze software requirements,
  - express designs, and
  - guide implementations;
- apply systematic code testing techniques;
- use well-known software design patterns.

Objectives (continued)

You will learn to use software development tools for:

- program development,
- configuration management (revision control),
- software modeling,
- testing.

Development Tools

You must use the following software development tools:

- Git at GitHub
- GitHub collaboration tools: issue tracker, pull request review system,
- Eclipse, GitHub integration with eclipse
- JUnit, code coverage Eclipse plugins.

Objectives (cont.):

You will learn

- a variety of software development processes, the software lifecycle, and their implications;
- software safety challenges;
- ethical issues in software engineering.

Development Process

We will use Agile methods for all programming assignments

- Usually 1-2 week iterations, teams of 3
- Your team will practice Agile techniques for setting requirements and acceptance criteria for each iteration, estimating resources, and improving estimations.

A Software Safety Challenge
The Google Driverless Car

Prerequisites

- CS253 and all the prerequisites for CS253 including:
  - Discrete structures (CS166, or CS160-CS161) and
  - Data structures (CS200).
- You should know how to program using object-oriented programming languages and structures.
- Programming in CS314 will be (mostly) done in Java.

Course Materials

- Required Text: Software Engineering custom text from the CSU bookstore.
- Course Notes.
- Supplemental material as needed.

Software Development Tools

- Eclipse
- Testing tools:
  - Junit 4: installed on Eclipse. You may need to add a link and put it on the "build path".
  - Emma and/or EclEmma (Eclipse plugin.) test coverage tools.
- Revision control:
  - Git or Github
- Modeling tools.

Additional Materials

- Articles from technical journals, magazines, and web sites.
- More Reference Texts:
  - R. Pressman, *Software Engineering: A Practitioner's Approach, 7th Ed.*, McGraw-Hill, 2010.
  - *Applying UML and Patterns-An Intro to OOA/D and Iterative Development* , Craig Larman, Prentice Hall, 3rd edition, 2005.
  - Ian Sommerville, *Software Engineering*, Addison-Wesley, 9th Ed., 2011.
  - S. L. Pfleeger & J. Atlee, *Software Engineering: Theory & Practice*, 3rd Ed. , Prentice Hall, 2006.

Learning Activities

Learn vocabulary and practice through:

- Lectures,
- Reading: text and other references,
- Exams,
- Quizzes
- Assignments,
- In-class design studios.

Exams

- Expect 2 exams:
  - Exam 1 during Week 7.
  - Exam 2 during Week 14.
- Closed book and closed notes.

  Exception: you may bring one 8.5 x 11 inch sheet of paper with your notes to the exam. You must put your name on the sheet and turn it in with your exam.

Quizzes

- Quizzes are based primarily on reading material.
- Complete reading assignments before class.
- Quizzes will either be done
  - On-line via Canvas, or
  - In-class at the start of a class-period; they take approximately 10 minutes.
- Expect one (or more) quizzes per week.
- Quiz 0: Prerequisites Parts A, B, and C are posted on Canvas.

Assignments

- Work in teams of 3 on all assignments.
  Except Assignment 0, which is posted.
- Planned assignments involve unit testing, programming, design, modeling, system testing, design patterns, requirements, and software safety.

Design Studios

Provide practice for many of the techniques required in the assignments.

1. Pre-design studio work – do on your own.
2. Quiz covering pre-class work due before the design studio class period.
3. In-class activity where results are used in your next assignment.
4. Post-class reflection – do on your own.

Web Support

- Calendar, Bulletin Board, Discussions.
- Password protection: view grades.
- Most assignments are submitted via Canvas.
- Check the class Canvas bulletin board daily for important announcements.
- Assignments are posted to Canvas
- Key links:
  - Weekly Guide
  - Class Session Blog
  - Syllabus

Course Rules

- Keep up.
  - Attend lectures.
  - Be on time.
  - Start assignments early.
- Behave professionally.
  - Put cell phones on silent.
  - Leave classroom to answer if necessary.
  - Be polite on discussion boards and email.
  - Respect others.
- Laptops in back row or rows.

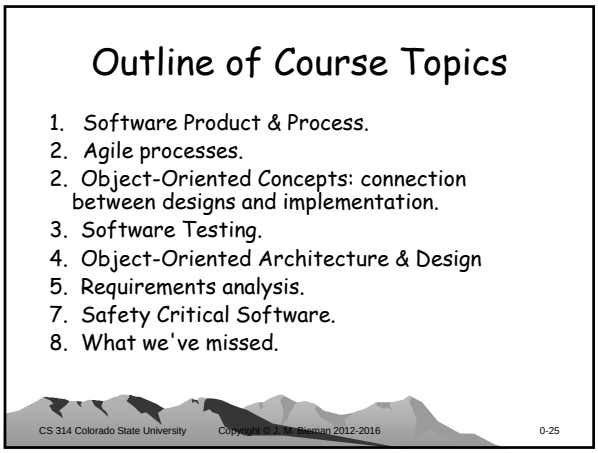
Grading

Approximate breakdown:

- Assignments: 40% of grade.
- Exam 1: 20%.
- Exam 2: 20%
- Quizzes: 10% of grade
- Design Studios: 10% of grade

Policies

- All work done on time
  - 48 hour grace for assignments with a 10% penalty.
  - Exceptions only for unforeseeable circumstances.
- Re-grade Policy
  - The GTA grades assignments, and is the first contact for grading concerns. Contact the instructor if you still disagree.
  - Bring up grading concerns promptly (within 5 days).


Outline of Course Topics
1. Software Product & Process.
2. Agile processes.
2. Object-Oriented Concepts: connection between designs and implementation.
3. Software Testing.
4. Object-Oriented Architecture & Design
5. Requirements analysis.
7. Safety Critical Software.
8. What we've missed.
CS 314 Colorado State University
Copyright © J. M. Bieman 2012-2016
0-25