ECE450 – Software Engineering II
-the sequel-
Winter Spring 2007
Instructor: Jorge Aranda

Instructor

• Jorge Aranda (that is "Hor-neh Ah-ran-dah")
  – I also respond to "George" if you must
  – Coordinates:
    • Bahen Centre 5233
    • jaranda@cs.toronto.edu
    • http://www.cs.toronto.edu/~jaranda/
  – 4th year Ph.D. student in Computer Science, SW Eng. group
  – Graduated in 1999, B.Sc. in Computer Systems Engineering
  – Software developer and project manager until 2003
  – Now researching information flow and requirements engineering
    • Small-medium scale (+10 companies)
    • Large scale (IBM)
  – Movie, book, and boardgame geek

General Information

• Lectures: Mon 10am, Tues 5pm, Thurs 5pm (BA1200)

• Tutorials: Tues 6pm, when announced (GB248)
  – No tutorial this week

• Lab sessions: Fri 3-6pm, when announced (GB243)
  – No lab session this week

• Office hours: Mon 11am, or by appointment

General Information (cont)

• Course website:
  – http://www.cs.toronto.edu/~jaranda/ece450h/

• Bulletin board on CCNet:
  – http://ccnet.utoronto.ca/20071/ece450h1s/
Textbook?

- **No textbook**
- We’ll often (for 4-7 weeks) refer to:
  - Gamma et al., “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley, 1995.
- **Should I buy it?**
  - Do you see yourself as having anything to do with software design in your career?
    - If so, get it
    - Otherwise, don’t buy it just for this course – there’s tons of similar material on the web
  - We’ll start referring to it in 3-4 weeks.

How to reach me

- **Your best bet is to talk to me at the end of the lecture**
  - Attendance is mandatory!
- **Email:** Include “[ece450]” in your subject line
  - Do not expect a quick (<24hrs) response!
- **Bulletin board:** Use for stuff that benefits everyone
- **Team blog**
  - More about this later

Course Overview

- **Pragmatic view of Software Engineering**
  - Focus on structuring principles and the design and development of large, complex software systems
  - Mix of practical insights and academic research
  - No parroting!
- **Main goals**
  - Knowing how to elicit requirements for complex systems
  - Identifying proper architectural structures for their design
  - Knowing your way around other essential Software Engineering topics (testing, PMing, etc.)

List of Topics

- **Intro to Software Engineering**
- **Software Processes**
- **Software Estimation**
- **Requirements Engineering**
- **Architecture and Design**
  - Design Patterns
- **Tools**
  - Essential
  - Advanced (e.g. visualization)
- **Testing and QA**
- **Project Management**
- **Peopleware**
Marking Scheme

• Assignment 1 10% (Feb 1)
• Midterm 15% (Mar 8)
• Assignment 2 15% (Apr 5)
• Assignment 3 15% (Apr 12)
• Assignment 4 10% (Apr 12)
• Participation 5%
• Final exam 30% (TBD)

• Assigs. 1, 2, 3 are in teams. Assig. 4 is individual

About team assignments

• Form teams of 3-4 people
  – Other sizes are not allowed
  – Suggestion: Try to get 4 people
    • If someone drops out you can carry on with 3
• Teams should be formed by next Monday’s lecture
• Your ideal team:
  – People that work like you
  – Complementary skill set
  – Common expertise in a programming language

Assignment 1

• Groundwork for assignments 2 and 3
• Two elements:
  – Selection and initial exploration of two similar open source systems which you will compare in Assig. 2
  – Selection and initial exploration of an open source project to which you will contribute during the term for Assig. 3
    • (May be one of the projects for assignment 2)
• Due in less than 4 weeks!

Assignment 2

• Comparison of the design and architecture of two similar open source systems considering this course’s material
  – Why open source?
    • Not easy to get access to the architecture of proprietary software!
• Your team will give a presentation to the rest of the group in the last week of lectures
Assignment 3

• Contributing to an open source project
  – Nature and size of contribution must be approved by me

• Valid projects:
  – Older than 3 years (younger projects might be OK – ask me first)
  – At least 3 active developers contributing to it (not you guys)
  – Active mailing list

• Either full or zero marks in this assignment!!
  – Start early, patch approval processes take time
  – Don’t overreach

• But why?
  – Unfortunately, course projects never convey software complexity appropriately!

Assignment 4

• Software Engineering essay
  • Only individual assignment of the course

• Essay may be about:
  – A discussion on an issue in Software Engineering
  – A critique on a software development book

Late Policy

• Due dates are firm

• Assignments are due before 5pm on their corresponding dates

• Any submission between 5pm and midnight has 20% penalty

• Zero marks after midnight.

Participation

• Participation is encouraged and expected
  – Ask questions, answer questions, offer a different viewpoint, tell me I’m wrong, tell others they’re wrong

• Make yourself memorable (in a good way!)
  – If I can’t put a face to your name in the end of term, you’ll probably get zero participation marks

• Ground rules
  – Be respectful
  – Be open to other ideas
  – Don’t hog the mike
Your first task

• 1% of your final grade (10% of Assignment 1):
  • Send me a short bio and a picture of yourself
    – Bio should be a text file about a paragraph long
    – Make sure to add “[ece450]” to your subject line

• Don’t have a photo of yourself?
  – Send me your bio anyway
  – I’ll bring a camera tomorrow

• DUE TOMORROW BEFORE 5 P.M.!

Your second task

• Form teams!
  – Reminder: teams of 3-4 people
    • Need to be comfortable working with them

• Fill out the Team Formation sheet that will be available on the course website
  – http://www.cs.toronto.edu/~jaranda/ece450h/

• Create a team blog and include its URL in your sheet

• DUE NEXT MONDAY
  – no penalty for being late, but you’ll be making your life harder

Recap: Software Engineering I

• What was CSC444 about?

• How was CSC444 like?

• Lessons learned?

Take the survey!