Welcome to
CS 410 –
Introduction to
Software Engineering

Fall 2016
Instructor: Marc Pomplun

Main Instructor – Marc Pomplun
Office: S-3-171
Lab: S-3-135
Office Hours: Tuesdays and Thursdays 3:15 – 4:00pm and 5:15 – 6:00pm
Phone: 287-6443 (office)
E-Mail: marc@cs.umb.edu
Website: http://www.cs.umb.edu/~marc/cs410/

Additional Instructors

Lere Williams
Co-founder, Engineering at Beanstock, Nigeria (https://www.beanstock.ag/)

Nam Chu Hoai
Fullstack Developer, Wellframe, Germany (https://www.wellframe.com/)

Srinath Vaddepally
Chief Executive Officer, RistCall, India (http://www.ristcall.com/)

The Visual Attention Lab

The new EyeLink-2K System

Example: Distribution of Visual Attention
Modeling of Brain Functions

Computer Vision:

Human-Computer Interfaces:

Now back to CS 410:

Textbook (available at the UMB Bookstore):


Course materials on the Web:
http://www.cs.umb.edu/~marc/cs410/
(contains all kinds of course information and also my slides in PPT and PDF formats, updated after each session)

Mailing List

Please use the ‘apply’ command on the UNIX system to register for our mailing list (CS410, section 1).
I expect everyone to be on the list, because I will use it to make announcements.
Also, I would like to encourage you to use the list for discussion. If you have a question that you think is important for many students in the course, please send it to the list at cs410-1@cs.umb.edu, and I will respond to the list.
Send all other questions to me. For these questions, I will send my reply only to you.
Your Evaluation

- 3 sets of exercises each set 3.33% 10%
  (only individual submissions allowed)
- software project 40%
  (groups of 3 or 4 students)
- midterm (1.5 hours) 20%
- final exam (2.5 hours) 30%

Grading

For the assignments, exams and your course grade, the following scheme will be used to convert percentages into letter grades:

- \[ \geq 95\% : A \]
- \[ \geq 86\% : B+ \]
- \[ \geq 74\% : C+ \]
- \[ \geq 62\% : D+ \]
- \[ \geq 90\% : A- \]
- \[ \geq 82\% : B \]
- \[ \geq 70\% : C \]
- \[ \geq 56\% : D \]
- \[ \geq 78\% : B- \]
- \[ \geq 66\% : C- \]
- \[ \geq 50\% : D- \]
- \[ < 50\% : F \]

Complaints about Grading

If you think that the grading of your assignment or exam was unfair,
- write down your complaint (handwriting is OK),
- attach it to the assignment or exam,
- and give it to me or put it in my mailbox.
I will re-grade the whole exam/assignment and return it to you in class.

Software Engineering

- 1968: Conference on "software crisis".
- Delivery of software was sometimes years late.
- Its cost was often much higher than predicted.
- Many programs were unreliable.
- Maintenance of software tended to be difficult.
- The software often poorly performed the task for which it was designed.

⇒ The term 'software engineering' was coined.

Questions about Software Engineering

Q: What is software engineering?
A: Software engineering is an engineering discipline which is concerned with all aspects of software production, for example, software specification, development, validation and evolution.

Q: What is the difference between software engineering and computer science?
A: Computer science is concerned with theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.

Problems: Complexity and Change

Complexity:
- Software systems can include a huge number of functions and components.
- Many participants with usually different backgrounds participate in the development of software systems.
- Often no single person can understand the whole system.
- Sometimes systems become so hard to understand that they are never finished: 'vaporware'.
Problems: Complexity and Change

Change:

• Requirements are updated when errors are discovered and when developers get a better understanding of the application.
• Long-term projects involve high staff-turnaround.
• Often, important technological changes occur during the development of a software system.
• The client’s needs may change during the development process.

⇒ It is impossible to specify a static set of requirements.