ACS 560
Software Engineering

Fall 2013

Instructor: Dr. Zesheng Chen

Class information

Lecture:

- Instructor: Dr. Zesheng Chen
- Office hours: T 8:45 – 9:45 pm
- Class time and location: T 6:00 – 8:45 pm KT250

- Email: zchen@engr.ipfw.edu
- Website: http://engr.ipfw.edu/~zchen/course/ACS560_13F.htm

Course Goal

- Survey the engineering aspects of software system design
- Give an introduction to software engineering using an object-oriented approach
- Emphasize on both theoretical and practical design techniques/patterns of software engineering

My Goals

- Become a better programmer
- Find a better job in software development
- Get a great sense of accomplishment when you can design a novel application and get a software system to work

Course Topics

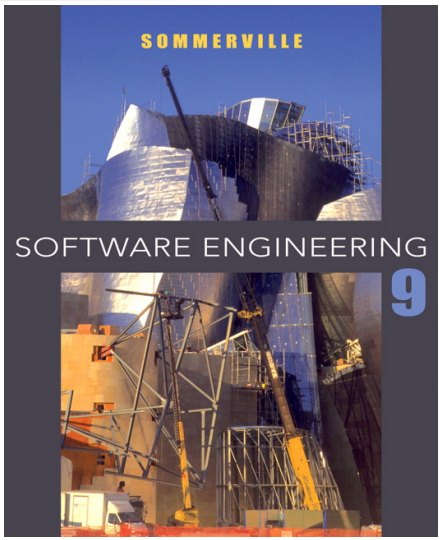
- Theoretical parts:
  - Software engineering ethics
  - Software process models
    - Agile software development
  - Requirements engineering
  - System modeling
  - Architectural design
  - Design and implementation
  - Software testing
  - Software evolution

Course Topics (Cont.)

- Practical parts:
  - Version control system (GIT)
  - UML
  - Socket programming
  - Design patterns
  - Refactoring
  - Debugging
  - Programming principles
  - Database
  - Linux command line

Textbook

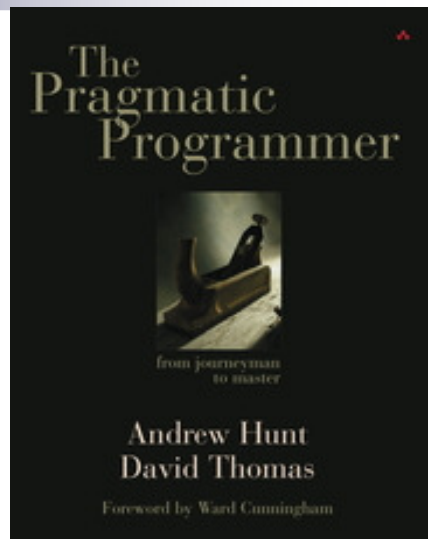
- Ian Sommerville, "*Software Engineering*", 9th edition, Addison Wesley (2011).



http://www.softwareengineering-9.com/

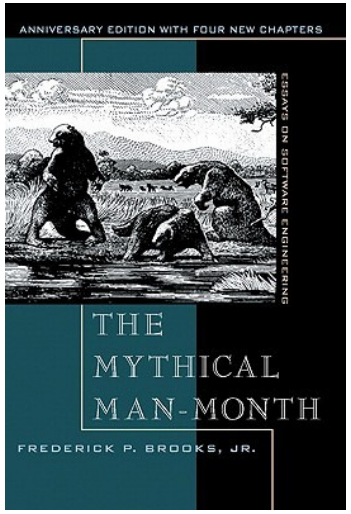
Textbook

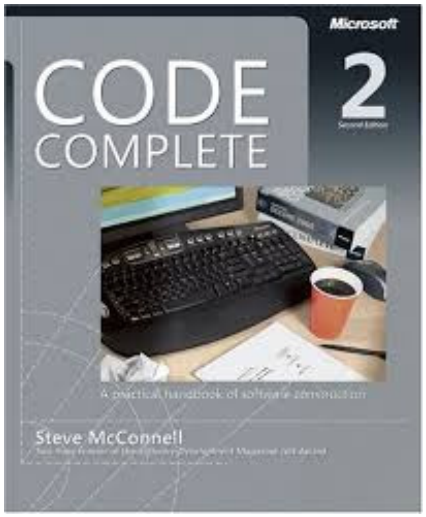
- Andrew Hunt and David Thomas, "*The Pragmatic Programmer Engineering*", Addison Wesley (2000).



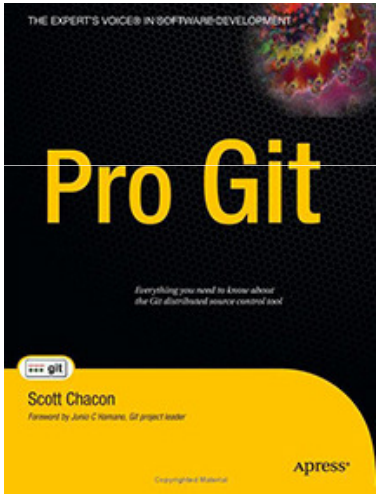
http://pragprog.com/the-pragmatic-programmer

Reference Books

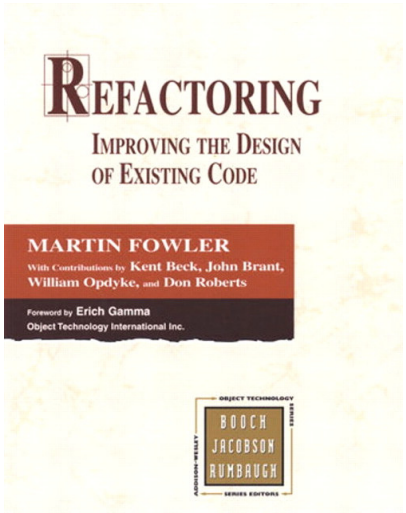




Reference Books



Reference Books



Grading

- Homework: 30%
  - Usually once every week, due one week after assignment before the class
- Take-home midterm exam: 30%
- Group Course project: 30%
- Class participation and quizzes: 10%

Grading (more)

- A 93 – 100
- A- 90 – 92
- B+ 87 – 89
- B 83 – 86
- B- 80 – 82
- C+ 77 – 79
- C 73 - 76
- C- 70 – 72
- D 60 – 69
- F Below 60

Homework

- Programming (short programs)
- Different languages (HTML/C++/Java/…)
- Questions and answers
- UML design
- System design
- Refactoring code

Midterm Exam

- Take home
- Problems similar to homework problems
- Allow to use Internet/books
- **Not** allow to discuss with each other
- Tentatively on Dec. 3

Group Course Project

- Two students in a group
- Practice Extreme Programming (XP)
- Semester-long project
  - Define the application
  - Write down requirements
  - Use UML to design the architecture/system
  - Program
  - Test
  - Demo

Group Course Project (Cont.)

- **You will choose the project topic**
- But here are some constrains/requirements
  - Client/Server architecture
  - Client and server use different operation systems
    - E.g.: One in Linux and the other in Mac OS X
  - Client and server use different programming languages
    - E.g.: One uses JavaScript and the other uses C++
  - Server connects to database

Group Course Project (Cont.)

- Examples:
  - Library system
  - Movie DVD rent system
  - Online text chatting system
  - Online gaming system
  - IP phone system
  - Online video chatting system
  - Mobile device application (Android/iPhone)

Learn software engineering through group course project!

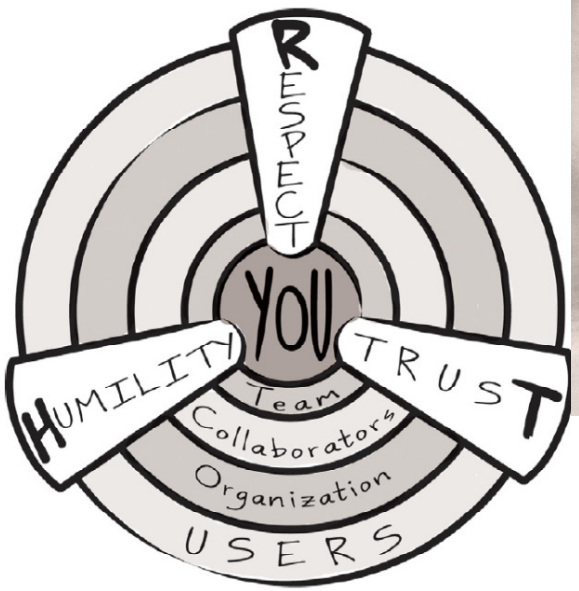
Group Course Project (Cont.)

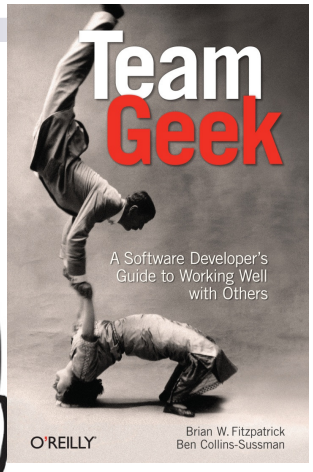
- Course project milestones (tentatively):
  - (Sept. 10) Define the application
  - (Sept. 24) Collect the requirements *
  - (Oct. 8) System architecture (UML) *
  - (Oct. 22) Define main classes
  - (Nov. 12) First coding phase done *
  - (Nov. 26) Second coding phase done *
  - (Dec. 10) Demo *

*** : Need to give the presentation in class**

Group Course Project (Cont.)

- Find your partner this week
- Start thinking possible course projects
  - Write down the list
  - Compare them
  - Decide which one you will work on
- Expect each group to meet weekly
  - Discuss the progress
  - Review each other's code (XP)





Class Policies

- No late homework
- Homework done individually
- Academy Integrity Code in IPFW student manual
  - Cheating
  - Plagiarism
- Missed exam regulations
- Class participation
- Late arrival

Learning Methodology

- Two learning methods:
  - Learning by instruction
  - Learning by discovery
- Software development
  - Ability to find the answer quickly
  - Ability to learn how to find the answer quickly

Leaning ability is very important!!!

Acknowledgements

- Thank Prof. Ken Rodham for sharing with me his class materials

- One last point
  - Communication

Have fun!!!