The Software Problem

“In 2000, total sales of software reached approximately $180 billion, supported by a large workforce encompassing 697,000 software engineers and 585,000 computer programmers.”

- Scale
- The cost of change
- Users as bugs
- Evolution yields complexity and bugs
- Software engineering matters
Scale

The Cost of Change

“Studies have found that reworking defective requirements, design, and code typically consumes 40 to 50 percent of the total cost of software development (Jones 1986). As a rule of thumb, every hour you spend on defect prevention will reduce your repair time from three to ten hours. In the worst case, reworking a software requirements problem once the software is in operation typically costs 50 to 200 times what it would take to rework the problem in the requirements stage (Boehm and Papaccio 1988). It’s easy to understand why. A 1-sentence requirement can expand into 5 pages of design diagrams, then into 500 lines of code, 15 pages of user documentation, and a few dozen test cases. It’s cheaper to correct an error in that 1-sentence requirement at requirements time than it is after design, code, user documentation, and test cases have been written to it.”

Steve McConnell, Software Quality at Top Speed, Software Development, August 1996
Users as Bugs

Boehm, SE as it is, ICSE’79
Evolution yields Complexity/Bugs

Scale, Bugs, Evolution

It’s a wonder software works at all.
And it’s so cheap, too.
What’s up with that?
RAYTHEON HAS SAVED $17.2 million in software costs since 1988, when its equipment division began using rigorous development processes that doubled its programmers’ productivity and helped them to avoid making expensive mistakes.

SOURCE: Raytheon
The Changing Face of Software

• Applications
  - Web 2.0, Mobile 2.0, …
  - Ubiquitous computing
  - Developing world

• Methodologies
  - Open Source
  - Agile (XP, Scrum)

• Technologies
  - Web services, javascript, AJAX, JQuery, …
  - Programming environments (Eclipse), AOP
  - Component-based, Model-driven software development

Do we rewrite the rules, or just reinterpret them?
Goals of the Course

• Develop common sense about software engineering
• Secondarily, take some ideas and skills into your own practice
• Improve reading papers critically
• Improve discussing technical ideas
• Conversant in issues – think and talk like a software engineer(ing researcher)
My Promise

Authentic practice
A minimum of busy work
Your Promise

Come prepared every day
Rest of Today

• Structure of course
• Grading
• How to read and discuss papers
• Project
• Questions (at any time)