Goals of this course

Introduce industrial software development:
- formal processes/artifacts for planning, specifying, designing, implementing, and verifying
- team-based development
- life-cycle issues and "umbrella" activities

Introduce theoretical foundations underlying these activities
- E.g., test-suite adequacy criteria
- E.g., formal specification notations/techniques
- E.g., psychology of program understanding

Shaw's model of engineering evolution

Characteristics: Craft

Virtuosos and talented amateurs
Intuition and brute force
Haphazard progress
Casual transmission of knowledge
Extravagant use of available materials
Manufacture for use rather than sale

Commercial production

Skilled craftsman
Established procedure
Pragmatic refinement
Training in mechanics
Economic concern for cost and supply of materials
Manufacture for sale

Characteristics: Professional engineering

Educated professionals
Analysis and theory
Progress relies on science
Educated professional class
New applications enabled through analysis
Market segmentation by product variety
Civil Engineering

Basis in theory.
- Actually two theories:
  - Statics: composition of forces.
  - Material strength: bending of a beam.
- Theories preceded real CE by 150 years!
Underlying science emerged 1700 years after commercial production evolved!

Chemical Engineering

Basis in practice:
- Rooted in empirical observation.
- Depends on both ME and Chemistry.
- Problem of scaling laboratory results up to industrial production.
Science (unit operations)
- Chemical processes can be decomposed into a few unit processes (e.g., filtration, heat exchange, distillation).
- Pragmatic empirical science; not theoretical.

Two “pillars” of SE education

Basis in:
- production processes and process frameworks
- rigorous theories addressing design problems that attend to the various phases of these processes
This course:
- organized around first pillar
- structured so that process issues will motivate introduction of theoretical content
How “different” is this course?

Quite!
- Not a “programming course”
- Exercises aim to develop planning/estimation skills and sensitivity to real problems that attend to industrial software development
- Written communication skills will be exercised, improved, and assessed
  - Certainty in planning documents, but especially in postmortem analyses
- Team work is critical and will be assessed

Relation to other courses?

Not a design course (ala CSE 335)
- Much “higher-level” coverage of notations
- More emphasis on process than design methods
Not a capstone design experience (ala CSE 498)
- Smaller, more constrained project
- Smaller teams (2 [at most 3] members)
- I will be your customer

Ideal pre-capstone course:
CSE 335 → CSE 435 → CSE 498