SOFTWARE ENGINEERING CAPSTONE COURSE

• Course Description
  • Required two-semester course
  • Semester 1: Requirements engineering (must pass with C or better)
  • Semester 2: Design and implementation
  • Team based; complex project defined by an actual customer
  • Learn-Practice-Apply-Improve cycle

• Audience: Seniors
CURRICULUM DEVELOPMENT

• Learning outcomes
  • Documented and included in syllabus
    • Knowledge and comprehension
    • Application and analysis
    • Synthesis and evaluation
  • Every other year: attainment of student outcomes and learning outcome examined; recommendations are made to instructors and curriculum committee
  • Every 4 years: major review

• Course Repository
  • Resources
  • Templates for documents using IEEE standards
BALANCE OF RESEARCH AND TEACHING

- High-Assurance Transformation System (HATS) GUI
  - Client: Sandia National Laboratories
- Scene and Countermeasures Integration for Munition Interaction with Targets (SCIMITAR)
  - Client: ARL
- Saturn Rings project
  - Client: NASA
- Gravity Data Repository and Processing System (GDRP)
  - Client: U.S. Geological Survey; Pan American Center for Earth and Environmental Studies (PACES)
- Sensor Data Property Specification tool
  - Client: Environmental scientist with Cyber-ShARE Center
PROFESSIONAL DEVELOPMENT CONNECTED WITH TEACHING

- Certification
  - Secondary education
  - IEEE-CS CSDP
- Cooperative learning methodologies for team building
  - Teaching professional and team skills
  - Individual accountability
  - Student-professor reflections
- Communities of practice (Lave & Wenger, 1991)
  - Learners develop the skills, knowledge, and expertise of the group through supported immersion
- Attendance, publication and presentation at education conferences
TEACHING TIPS

- Engage the students
- Help students ask good questions
- Challenge their knowledge
- Provide timely, constructive critique
- Scope your expectations

It’s better for students to learn essential concepts deeply rather than many concepts superficially