Texas State 3398 Software Engineering Course

• Introduction to Software Engineering
  • examines the various phases of the software lifecycle
  • apply a selected methodology (or two) to each phase

• Intro to Software Engineering Synthesis
  • examines methodologies, formalisms and processes for describing and synthesizing software systems
    • Requirements and design notations
    • Process models

• Intro to Software Engineering Analysis
  • explores approaches to testing and analysis to improve software quality, safety, and reliability across the lifecycle
Prerequisites

- Knowledge of a higher-level programming language
  - Java
  - Ada
  - C++
  - C
- Data structures
- Some discrete mathematics
  - E.g., Graph theory, predicate logic, set theory
Synthesis Topics

• The software crisis
• The software qualities
• Software development processes
• Requirements engineering
• Software architecture
• Principles of software design
• Configuration management
Testing & Analysis Topics

- Theoretical foundations
- Manual techniques
  - Software inspection
  - Cleanroom
- Dynamic analysis
  - Assertions
  - Error seeding, mutation testing
  - Coverage criteria
  - Dependency analysis
  - Fault-based testing
  - Regression testing
  - Object oriented testing
Testing & Analysis Topics (continued)

- Static analysis
  - Symbolic execution
  - Software verification
  - Data flow analysis
  - Interprocedural analysis
- Specification based testing
Testing & Analysis Topics (continued)

- Concurrency analysis
  - Safety and liveness
  - Reachability based analysis
  - Finite state verification
- Experimental studies and s/w metrics
- Overall concerns
  - Software safety, licensing, professionalism
Reading material

• **Required:**
  - On the web

• **Suggested Readings**
  - *The Mythical Man-Month: Essays on Software Engineering* by Frederick P. Brooks Jr; Addison-Wesley Pub Co
  - *Fundamentals of Software Engineering* by Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli; Prentice Hall
Flavor of Course

- Combination of practical and theoretical approaches
  - First half of class focuses on software synthesis approaches
  - Second half of class focuses on analysis approaches

- Reading material from the literature
  - Old “classics”
  - Promising new approaches

- Discuss current industrial practices, “best” practices, “future” practices

- New insight into software quality
Reading assignment