CSCE 4430: Software Engineering - Spring 2013

Instructor: Paul Tarau, Associate Professor - see my home page for classroom and office hours.

E-mail: tarau@cs.unt.edu
WWW: http://www.cs.unt.edu/~tarau
Address: Department of Computer Science, University of North Texas, P.O. Box 311366, Denton, Texas 76203, USA
Phone: Tel: +1-940-565-2806, +1-940-565-2767
Fax: +1-940-565-2799

Teaching Assistant. Srujan Kotikela See his home page for contact info and office hours.

First exam: TBD

Final exam: TBD – please complete exit surveys

Description and Objectives:

An ACM/IEEE 2013 curricula based course, focusing on principles and best practices for specification and implementation of quality software systems, including the use of current design, refactoring and testing tools, team, risk and project management, requirement analysis and formal methods.

Topics include requirements and specifications development,

Syllabus (L1..Ln) indicate number of the lecture

- Introduction to software process models (waterfall, incremental, agile) L1,L2
- Team Management L3
  - Roles and responsibilities in a software team
  - Team conflict resolution
  - Team organization and decision-making
- Risk Management L4
  - The role of risk in the life cycle
  - Risk categories: security, safety, market, financial, technology, people, quality, structure and process
  - Risks associated with virtual teams (communication, perception, structure)
• Project Management L5
  • Scheduling and tracking
  • Project management tools
  • Cost/benefit analysis
• Requirements Engineering (prototyping, specification, validation, tracking) L6
• Software Design and Implementation L7,L8
  • Paradigms, principles
  • Separation of concerns and information hiding
  • Coupling and cohesion, re-use
  • Object / function / aspect / service oriented designs
  • Relationships between requirements and designs: transformation of models, design of contracts, invariants
• Software Verification and Validation L9
  • Testing fundamentals
    ▪ Unit, integration, validation, and system testing
    ▪ Test plan creation and test case generation
    ▪ Black-box and white-box testing techniques
    ▪ Regression testing
    ▪ Usability testing
  • Test-driven development
• Software Evolution L10,L11
  • Characteristics of maintainable software
  • Refactoring and Reengineering
  • Software reuse
• Formal Methods L12,L13
  • Assertions, pre-conditions and post-conditions
  • Executable specifications
  • Tools
• Software Reliability L14
  • System reliability and failure behavior
  • Fault lifecycle concepts and techniques
  • Software fault tolerance techniques and models
  • Measurement-based analysis of software reliability
• Software engineering as knowledge engineering L15

Prerequisite: 3110 Data Structures

Recommended free online information sources and Kindle books:

• Extreme Programming
• Software Engineering
• Requirements analysis
• Formal Methods
• Unified Modeling Language
• Entity-relationship model
• Scala Tutorials
• Kindle books: **Rapid Development** [Clean Code](#) [The Mythical Man-Month](#) [Code Complete](#)

• 2 Individual Exams: 60%
• Team Project and Assignments (groups of 2-3): 40%

**Resources:**

• Scala
• The [Eclipse](#) Open Software Development Platform
• Java interpreter/compiler, on CSCE network, PC or Mac (java + javac)
• [Programmers Competency Matrix](#)
• [The Interface Hall of Shame](#)
• An unconventional view about how learning works
• A view on asking good questions in public forums

**Outcomes:**

1. Elicit and document requirements for a software project.
2. Being able to chose and evaluate software process models (waterfall, incremental, agile).
3. Conduct software testing and usability testing.
4. Participate in peer reviews such as code inspections.
5. Communicate software product and process results in oral and written form.