CS 5640: SOFTWARE DEVELOPMENT

Course Description

An in-depth study of all aspects of the software development process: user requirements, specifications, design, coding, testing, maintenance, documentation, management. Use of CASE tools for analysis and design. Prerequisites: Admission to MS in CS program, or consent of department, plus CS 3240 or equivalent.

Course Syllabus

- **Overview of the software life cycle**
- **Systems analysis**
  - User and/or market requirements
  - Project specifications
  - Structured analysis
  - Decision tables or trees
- **Preliminary design**
  - User interface design and prototyping
  - Structured design
  - Design evaluation, including cohesion and coupling
- **Detailed design**
  - At least one diagraming technique for detailed design
  - Algorithm choice and efficiency issues
  - Detailed design standards
- **Implementation**
  - Language selection
  - Approaches to implementation: top-down, bottom-up, threads
  - Coding methods and guidelines
- **Testing**
  - Types: unit, integration, system, field, acceptance
  - Testing methods, including black-box and white-box
- **Software maintenance**

The student will be able not only to define and discuss the topics listed on this sheet, but will be able to apply the principles and techniques to new problems.

Experiences

It will be assumed that the student has been involved in a team project, specifically one involving the development of a software system for a user.

Student Learning Outcomes

- I can compare and contrast alternative life-cycle approaches to software development, and I can state the risks and benefits of each approach for a given project.
- I can describe factors, positive and negative, known to influence productivity of software development teams, and I can provide examples of these factors based on a retrospective analysis of my own team's project.
• I can distinguish between various forms of testing, including unit testing, integration testing, functional testing, regression testing, acceptance testing and system testing, etc.
• I am part of a team that has had productive interactions with clients/sponsors.
• I am part of a team that has communicated effectively with clients/sponsors.
• I am part of a team that has followed a user-centered design and development strategy.
• I can make selective use of a minimum subset of appropriate technical notations, including UML class diagrams.
• I have communicated effectively with other team members, work cooperatively with them, and carry their fair share of the workload.