M.S. in Software Engineering

Learning Outcomes

All students will be able to:

- Model the structure and behavior a software system the UML class diagrams and state diagrams.
- Design a solution to a given problem using one or more design patterns and implement the design in a programming language.
- Apply software testing and quality assurance techniques at the module level, and understand these techniques at the system and organization level.
- Understand common lifecycle processes including waterfall (linear), incremental approaches (such as Unified process), and agile approaches.
- Work collaboratively in a small team environment to develop a moderate-sized software system from conceptualization to completion, including requirements elicitation, system modeling, system design, implementation, unit and system testing, integration, source code management configuration management, and release management.
- Prepare technical documentations and make presentations on various aspects of a software development project, including the technical aspects (architecture, design, quality assurance) as well as the managerial aspects (planning, scheduling, delivery).
- Use electronic communication tools to effectively collaborate with team members in different countries with different cultural backgrounds.

Each student will choose one concentration and be assessed as indicated below.

1. **Software Development**: Students will be able to design and implement both the server and client components of a client-server application, such as a web-based application.
2. **Software Architecture**: Students will have knowledge of commonly used architectural patterns, styles, and tactics, and identify their impact upon various quality concerns such as scalability, performance, and reliability.
3. **Project Management**: Students will be able to give examples of the primary project management activities associated with each major software engineering activity including requirements elicitation, analysis and specification; analysis and design; implementation; testing; deployment; maintenance; and version control.
4. **Gaming & Entertainment Technologies**: Students will be able to apply the principles and techniques of software engineering in the architectural design, detail design, and implementation of computer games or other entertainment focused software applications.

Last Update: 2014
5.  *Entrepreneurship & Technology Leadership*: Students will be able to develop a business plan for a start-up software business to be presented to a venture capitalist.