

# M.S. with a Major in Software Engineering

*Candidates must satisfy a total of 30 credit hours (CH) with a minimum G.P.A. of 3.000 on a 4.000 scale.*

*All students must complete 12 credit hours (CH) of the core curriculum.*

## **CSE 7314 Software Testing and Quality Assurance**

The relationship of software testing to quality is examined with an emphasis on testing techniques and the role of testing in the validation of system requirements. Topics include module and unit testing, integration, code inspection, peer reviews, verification and validation, statistical testing methods, preventing and detecting errors, selecting and implementing project metrics, and defining test plans and strategies that map to system requirements. Testing principles, formal models of testing, performance monitoring, and measurement also are examined.

## **CSE 7315 Software Project Planning and Management**

The issues associated with the successful management of a software development project are addressed. This includes planning, scheduling, tracking, cost and size, estimating, risk management, configuration, management quality, engineering and process improvement. The course is centered on the concept of a Software Engineering Process and includes discussion of life cycle models for software development. The SEI software process capability maturity model (CMM) and other process standards are included.

## **CSE 7316 Software Requirements**

This course focuses on defining and specifying software requirements that can be used as the basis for designing and testing software. Topics include use-cases for describing system behavior, formal methods, specifying functional vs. nonfunctional requirements and the relationship of requirements to software testing.

## **CSE 7319 Software Architecture and Design**

Successful software development requires both an understanding of software design principles and a broader understanding of software architectures that provide a framework for design. The course explores the role of design in the software life cycle including different approaches to design, design tradeoffs and the use of design patterns in modeling object-oriented solutions. It also focuses on important aspects of a system's architecture including the division of functions among system modules, synchronization, asynchronous and synchronous messaging, interfaces, and the representation of shared information.

*All students must complete 9 credit hours (CH) of advanced elective courses.*

## **CSE 7111 Intellectual Property and Information Technology**

This course presents fundamentals in the nature, protection, and fair use of intellectual property. Patent, copyright, trademark, trade secret and antitrust principles are presented with an emphasis on the Internet, software, databases, and digital transmission technologies. The open source and creative commons alternatives for disseminating intellectual property are investigated. We examine the engineer, scientist, manager, and creative artist's professional and ethical responsibilities and opportunities regarding intellectual property. We will also investigate the rapid change in types and uses of intellectual property spawned by computers, digital media, ecommerce, and biotechnology.

*Prerequisites:* Graduate standing and a general understanding of software and digital information systems.

## **CSE 7340 Service-Oriented Computing**

Service-oriented computing (SOC) is the computing paradigm that utilizes services as fundamental elements for developing applications. Service providers expose capabilities through interfaces. Service-oriented architecture maps these capabilities and interfaces so they can be orchestrated into processes. Fundamental to the service model is the separation between the interface and the implementation, such that the invoker of a service need only (and should only) understand the interface; the implementation can evolve over time, without disturbing the clients of the service. Topics include Web architecture, HTTP, XML, SOAP, REST, BPEL, and developing interfaces that connect to independent services. The course will be of interest to those interested in creating and/or aggregating web services and developing user interfaces for the display of those services. *Prerequisite:* Senior or graduate standing. Programming experience is required.

## **CSE 7345 Advanced Application Programming**

Covers advanced programming techniques that span a range of programming languages and technologies. Topics include server-side application development, client graphical user interface implementation, application frameworks, design patterns, model-based development, and multithreading. The specific programming language or languages covered may vary from term to term.

*Prerequisite:* CSE 3345 or consent of instructor.

### **CSE 7349 XML and the Enterprise**

Covers conventional and state-of-the-art methods for achieving data and network security. Private key and public key encryption approaches are discussed in detail, with coverage of popular algorithms such as DES, Blowfish, and RSA. In the network security area, the course covers authentication protocols, IP security, Web security, and system-level security. *Prerequisite:* CSE 7339 or equivalent with instructor permission.

### **CSE 7359 Software Security**

As software is delivered across network and Web-based environments, security is critical to successful software deployment. This course focuses on software security issues that pertain to the network application layer in the classic OSI model. At the application network layer, issues related to encryption, validation, and authentication are handled programmatically rather than at the network level. Students work with APIs for cryptography, digital signatures, and third-party certificate authorities. The course also explores issues related to XML and Web services security by examining standards and technologies for securing data and programs across collaborative networks. *Prerequisite:* C- or better in CSE 7339.

### **CSE 8312 Software Generation and Maintenance**

Techniques for generating software and maintaining revisions to existing software will be examined. Topics include alternatives to coding, the use of program generators and very high-level languages, CASE tool, component re-use, and the role of standards in the generation of software. Also covered are issues related to maintenance as a part of software evolution, the impact of the design process on long-term software maintainability, software re-engineering, and the planning of release cycles.

### **CSE 8313 Object-Oriented Analysis and Design**

Object-oriented analysis and design is essential in developing high-quality object-oriented systems. The course will provide an overview of object-oriented analysis and design by integrating the work of Booch, Rumbaugh, Jacobson, and Wirfs-Brock. Topics will include use-case analysis, responsibility-driven design, object modeling, entity-relationship modeling, and the design notation of the Unified Modeling Language (UML). Additional topics will include object-oriented class libraries, object-oriented databases, and the Common Object Request Broker Architecture (CORBA).

### **CSE 8314 Software Metrics and Quality Engineering**

Techniques of software quality engineering with emphasis on the role of metrics are addressed. The approach is drawn from practical experience and uses many examples from industry. The psychological and behavioral aspects of quality and quality assurance are included. Metrics and quality are presented in relationship to the software process and software process maturity models. Selection of quality metrics is addressed in terms of the goal/question/metric paradigm as well as various quality models. Methods of storing data for historical purposes, analyzing, and presenting data to others are included.

### **CSE 8315 Software Acquisition Practices, Legal and Economic Issues**

Issues relating to software procurement, contract law, specification and control of product processes are examined. Topics include factors that affect cost, cost estimation, cost/benefit analysis, risk analysis, and legal implications with respect to ownership and use. Techniques and models of cost estimation are studied in detail.

### **CSE 8316 User Interface Design**

Design methodologies for user interfaces. Includes life cycles for UI development, human factors issues, prototyping, user analysis and evaluation, and design techniques. Students perform the analysis, design, and evaluation of a UI through two iterations.

### **CSE 8317 Software Reliability and Safety**

In-depth study of techniques for ensuring software reliability and safety. Topics include software reliability engineering, software safety engineering, and recent developments in those areas. Reliability concepts applied to the software domain and safety concepts applied to computer-intensive systems will be discussed. Specific techniques such as software reliability models and analysis methods, operational profiles, safety and hazard analysis using fault trees and event trees, and formal verification for safety-critical software systems will be covered.

### **CSE 8340 Advanced Topics in Software Engineering**

In-depth study of specific topics in software engineering techniques, methodologies, and issues. Topics will change from term to term and will include advanced software reliability models, software development process models, advanced object-oriented design, and cleanroom software engineering.

*All students must complete 9 credit hours (CH) of elective courses. For a full listing, please refer to the graduate catalog.*