

**The Master of Software Engineering (MSE) Degree  
Department of Computing and Information Sciences  
College of Engineering  
Kansas State University  
Fall Semester 1995**

### **The Software Engineering Discipline**

The discipline of software engineering covers the application of engineering principles to the building of computer software. The field covers the theories, tools and methods for systematic representation, design, verification, development, production, validation, and maintenance of software products including programs, prototypes, documentation, user interfaces, training, and evaluation. Software engineering is applicable not only to computer systems software; the techniques of software engineering offer benefit for software developed for all disciplines.

Engineering is the "practical application of scientific knowledge." The application of knowledge about software is made practical through the use of common techniques, components, tools, and methods of management. Specifically, like other branches of engineering, software engineering:

- uses formal models and methods of computing to develop formal requirements of and specifications for application domain software;
- uses established techniques of design to establish the structure of software before it is actually programmed;
- uses established techniques for verification (formal analysis of correctness of properties) of system design;
- uses established techniques for validation (systematic measurement and analysis of properties) of systems implementation;
- emphasizes the building of large software systems by integration of standardized components;
- uses numerous software tools to provide assistance in all activities of software development;
- studies the processes of software development as the basis for systematic management.

### **Admission to the MSE Program**

The admission requirements for the proposed MSE program are:

1. a baccalaureate degree in computer science, computer engineering, a mathematical science, or a related engineering or science area, with an undergraduate grade point average of 3.0 or above (junior/senior years); applicants with degrees in other areas must have mathematical maturity characteristic of students who have taken courses in symbolic logic (KSU - CIS 301) or discrete mathematics (KSU - MATH 510).
2. (undergraduate) courses in programming, data structures, and algorithms (e.g. KSU - CIS 200, 203, 300 and 500), and software engineering (e.g. CIS 540 or equivalent practical software engineering experience).

If the prerequisite courses mentioned above have not been taken, several additional courses may be required to improve the student's preparation for graduate study.

### **Program of Study for the KSU Master of Software Engineering Degree**

The Program of Study in the proposed MSE program will consist of 33 semester credits that must include the following:

1. CIS 740 and CIS 771
2. two courses from CIS 644, CIS 725, CIS 746, CIS 748, and CIS 764.
3. one course from CIS 826, CIS 841, CIS 842, and CIS 864.
4. two courses from an application area such as:  
Parallel and Distributed Systems, Operating Systems and Realtime systems, Database Engineering, Knowledge-based Systems, Graphics, or specialty areas from Electrical Engineering, Computer Engineering, Industrial Engineering, Mechanical Engineering, Nuclear Engineering, Chemical Engineering and other areas by special arrangement.
5. CIS 895 MSE Project (six credits)
6. six credits of technical electives (computer science or application area courses).

Each student will specialize in an application area. It is expected that each student will do his/her project in an area related to that application area and that one supervisory committee member will have expertise in the application area. Each student will produce a "software portfolio" which will contain a representation of the student's most important software expertise. The electives will allow the MSE students the opportunity to strengthen their overall computer science skills or to gain some expertise in related areas.

### **Core Courses in the MSE Program**

✓ -CIS 644 Object Oriented Software Development (new course)

This is a graduate course in object-oriented design and development. It covers use of different OOD methodologies, structuring of objects, and implementation in an object-oriented language. Prereq: CIS 500.

✓ -CIS 725 Advanced Computer Networks

This is a graduate level course focusing on network protocols. It covers network algorithms -routing and congestion control; protocol engineering - protocol decomposition, specification and verification, and synthesis; protocols for high speed networks and parallel implementations, and light-weight protocols. Prereq: Graduate Standing.

✓ -CIS 740 Software Engineering

This is the graduate-level introduction to the area of software engineering. It covers software lifecycle models, requirements, design, software measurement, cost estimation, testing, reliability, maintenance, and re-engineering. Prereq: CIS 540

-CIS 746 Software Measurement

This course introduces the theory of software measurement and covers the practical application to software management. It includes measurement theory, model-order-

mapping, measure validation, and measures throughout the life cycle. Prereq: CIS 740.

-CIS 748 Software Management

This is an advanced presentation of topics related to the management of software development. It includes topics of planning, process models, TQM and SQA, estimation, organization, and configuration management. Prereq: CIS 740.

-CIS 764 Database Design (new course)

This course covers the logical, physical, and user interface design of databases. It includes the study and use of data models, query languages, and the use of tools for the design and development of databases and applications in systems with graphical user interfaces. Prereq: Graduate Standing.

-CIS 771 Software Specification

This course introduces the formal specifications of program components with emphasis on the composition of well-crafted specifications and the formal verification of formal specifications. Prereq: Knowledge of Symbolic Logic.

-CIS 826 Protocol Engineering (new course)

This course covers the basic concepts of protocol design, specification languages and formal description techniques, safety and liveness properties, protocol validation, protocol synthesis, protocol translation, implementation, and conformance testing. Prereq: CIS 725.

-CIS 841 Software Validation (new course number, material previously taught under CIS 840)

This course is an advanced presentation of verification and validation of software. It includes the theory of testing and reliability as well as practical application to the problems of validation. Prereq: CIS 740.

-CIS 842 Specification and Verification of Reactive Systems (new course number, material previously taught under CIS 840).

This course emphasizes the composition and verification of well-crafted specifications based on temporal logic. Prereq: CIS 771.

-CIS 864 Data Engineering (new course)

This course covers advanced topics in database design and maintenance, including performance monitoring, database tuning, dictionaries, selection of file and index structures, query optimization, recovery, and security. Distributed data systems and interoperability will also be covered. Prereq: CIS 761 or 764.

-CIS 895 MSE Project (new course)

This is the capstone course that takes students through the process of developing a project. Includes reviews and walkthroughs of the requirements, design and implementation. This will be a major portion of each student's "software portfolio", a collection of representative samples of a student's software engineering expertise. Prereq: CIS 740 and CIS 771.

## **Participating Computing & Information Sciences Faculty**

- Jan Chomicki, Assistant Professor, PhD Rutgers - Data Engineering, Deductive Data Bases, and Logic Programming.
- David Gustafson, Associate Professor, PhD Wisconsin - Software Engineering, Metrics, Verification and Validation and Program Development Environments.
- William Hankley, Professor, PhD Ohio State - Software Specification, Temporal Logic, and Graphics User Interfaces.
- Rodney Howell, Associate Professor, PhD Texas at Austin - Analysis of Algorithms and Real-time Scheduling.
- Masaaki Mizuno, Associate Professor, PhD Iowa State - Distributed Systems and Operating Systems.
- K. Ravindran, Assistant Professor, PhD British Columbia - Distributed Systems, High-Speed Packet Networks.
- David Schmidt, Professor of CIS, PhD Kansas State - Programming Languages and Object-Oriented Programming
- Gurdip Singh, Assistant Professor, PhD State University of New York - Protocol Engineering and Distributed Systems.
- Allen Stoughton, Associate Professor of CIS, PhD Edinburgh (Scotland) - Programming Language Semantics
- Elizabeth Unger, Professor, PhD Kansas - Data Engineering, Database design, and Data integrity and security.
- Virgil Wallentine, Professor, PhD Iowa State - Parallel Programming, Object-Oriented Concurrent Programming, and Knowledge-based Software Engineering.
- Maria Zamfir-Bleyberg, Associate Professor of CIS, PhD UCLA - Concurrent Systems, Algebraic Specifications, Artificial Intelligence Applications.

## **Faculty Participating in Software Engineering Applications Areas**

- Satish Chandra, Associate Professor of EECE, PhD Auburn, Computer Architecture and Computer Vision
- Dwight Day, Assistant Professor of EECE, PhD Oklahoma State, Image Processing and Computer Vision Applications
- John Devore, Associate Professor of EECE, PhD Kansas State, Graphics Systems and Embedded Systems
- Eddie Fowler, Professor of EECE, PhD Oklahoma State, Expert Systems, Fuzzy Logic, and Neural Networks
- Rodney Fox, Assistant Professor of EECE, PhD Kansas State, Reactive Flow Modeling, Stochastic Processes, Non-linear Computational Models and Numerical Methods
- Brian Harms, Professor EECE, PhD Kansas State, Signal Processing and Communications
- William Hudson, Assistant Professor of EECE, PhD New Mexico State, Neural Networks, Embedded Systems, and Hardware Verification
- Donald Lenhert, Professor EECE, PhD University of New Mexico, Microcomputer Applications, Embedded Systems and Testing of Digital Systems
- Ken Shultis, Professor of Nuclear Engineering, PhD Michigan, Numerical Algorithms for Fluid Flow, Neutron Diffusion, and Heat Transfer
- Daniel Swenson, Professor of Mechanical Engineering, PhD Cornell, Computational Modeling

Other engineering and science faculty will be added as new software applications area are developed.