COMPUTER SCIENCE 5744

SOFTWARE DESIGN AND QUALITY

(ADP TITLE: SOFTWARE DES & QUAL)

I. CATALOG DESCRIPTION:

5744: SOFTWARE DESIGN AND QUALITY

This course focuses on critical aspects of the software lifecycle that have significant influence on the overall quality of the software system including techniques and approaches to software design, quantitative measurement and assessment of the system during implementation, testing, and maintenance, and the role of verification and validation in assuring software quality. Pre: 5704. (3H, 3C).

II. LEARNING OBJECTIVES:

Upon successful completion of 5744, a student will be able to:

- Read and write basic UML design notation
- Review and critique software designs for quality
- Collect and interpret OO code metrics
- Plan and manage integration and system tests
- Apply some design patterns
- Explain software process and project metrics
- Assess software quality through metrics
- Discuss verification and validation activities throughout the software lifecycle
- Discuss methods for managing traceability

III. JUSTIFICATION:

This course takes students who have already been introduced to the basic concepts of software engineering, and pursues more advanced techniques in software design and project management that are aimed at producing high-quality results. It is the first course in the software development module of the Graduate Program in Information Technology. In addition, it offers additional advanced study in a high-demand topic area for students in the CSA program.

CS 5744 is centered around group learning activities that promote student interaction, discussion, and critical analysis of the techniques and sample systems presented in the course material. As a result, it encourages active learning through these in-class and out-of-class group activities. Group activities include analysis and presentation of relevant published articles, software design reviews, design critiques, and redesigns that students must perform. Often, such group activities require some out-of-class work organized and managed by small groups of students, followed by a presentation of the results to the

entire class for further discussion. While much of the preparatory work for these activities can be done asynchronously, the final step in most group activities involves class participation. The design of the course materials for 5744 aims to maximize asynchronous, self-paced delivery of lecture and reference material so that in-class time can be devoted to class activities that require simultaneous participation by many students.

IV. PREREQUISITES AND COREQUISITES:

This course requires a basic understanding of software engineering concepts and terminology. Specifically, students should have some familiarity with a variety of lifecycle models, techniques for managing software projects, V&V and testing practices, object-oriented and structured design methods, and an introductory background on metrics. These requirements are met by 5704 and its prerequisites.

V. TEXTS AND SPECIAL TEACHING AIDS:

Required text to be chosen from:

Mary Shaw and David Garlan. SOFTWARE ARCHITECTURE: PERSPECTIVES ON AN EMERGING DISCIPLINE. Prentice-Hall, 1996. xxi, 242.

Edward Kit. SOFTWARE TESTING IN THE REAL WORLD: IMPROVING THE PROCESS. Addison Wesley Longman, Reading, MA, 1995. xiv, 252.

Len Bass, Paul Clements, and Rick Kazman. SOFTWARE ARCHITECTURE IN PRACTICE. Addison Wesley Longman, Reading, MA, 1998. xxiii, 452.

Special teaching aids: The course presumes all students have access to a personal computer and to the internet. A commercial-quality CASE tool available free for educational use may be used by students. Supplementary reading materials may include contemporary journal articles and technical papers.

VI. SYLLABUS:

	Topic	Percent of Course
1.	Design for Quality	35
	a. Current Design Notation(s)	
	b. Software Architecture	
	c. Designing for Usability	
	d. Advanced Design Techniques	
2.	Quality Assessment and Improvement	25
	a. Verification and Validation	
	b. Quality Assurance	
	b. Applying Software Metrics	
	c. Planning/Managing Testing	
3.	Concept Integration, Case Studies	20
4.	Concept Application, Student Projects	20
	Total	100

VII. OLD SYLLABUS:

N/A

VIII. CORE CURRICULUM GUIDELINES:

N/A