DIRECTIONS: Provide one form with original approval signatures in lines 1 - 4 for each proposed action. Keep this form to one-page in length. Forms and appropriate attachments should be submitted to the Office of Academic Affairs, who will assign a Log Number to each proposal. Additional copies may be required at a later time.

DATE SUBMITTED 4/28/03

COLLEGE/SCHOOL A. J. Clark School of Engineering

DEPARTMENT/PROGRAM GCEN

PROPOSED ACTION (A separate form for each) ADD ___ DELETE ___ CHANGE ___

DESCRIPTION (Provide a succinct account of the proposed action. Additional detail may be provided in an attachment. Provide old and new sample programs for curriculum changes.)

Add a new option under the existing Graduate Certificate in Engineering (GCEN) Program.

JUSTIFICATION/REASONS/RESOURCES (Explain the reason for the proposed action. Identify the source of new resources that may be required. Attach additional material if needed.)

See attached.

APPROVAL SIGNATURES

1. Department Committee Chair [Signature] 4/28/03
2. Department Chair [Signature] 4/28/03
3. College/School PCC Chair [Signature] 10/28/03
4. Dean [Signature] 3/06/03
5. Dean of the Graduate School (if required) [Signature] 11/10/03
6. Chair, Senate PCC [Signature] 11/26/03
7. Chair of Senate [Signature] 11/26/03
8. Vice President for Academic Affairs & Provost [Signature] 11/26/03

VPAAP Rev. 2/2/98

PCC-AY03-06-PM
November 21, 2003

MEMORANDUM

TO: Nariman Farvardin
Dean, A. James Clark School of Engineering

FROM: Victor Korenman
Associate Provost for Academic Planning and Programs

SUBJECT: Proposal to Add a New Track in Software Engineering to the Post Baccalaureate Certificate Program in Engineering (PCC Log No. 03015)

At its meeting on November 21, 2003, the Senate Committee on Programs, Curricula, and Courses approved your proposal to add a new track in Software Engineering to the Post Baccalaureate Certificate Program in Engineering. A copy of the approved proposal is enclosed.

The change is effective immediately. The School should ensure that this change is appropriately reflected in all university documentation, and that all advisors are informed.

VK: sfm
Enclosure

Cc: Dr. Sylvester Gates, Chair, Senate PCC
Dr. Mary Giles, University Senate
Ms. Barbara Hope, Data Administration
Ms. Trudy Lindsey, Graduate Studies
Dr. Gary Pettmer, A. James Clark School of Engineering
Ms. Anne Turkos, Archives
Mr. Frank Valines, Student Financial Aid
Dr. Linda Yokoi, Records & Registrations
As software has become more complex, an increased interest in proper software engineering practices has developed. This demand can be seen in all levels of the field, including government, industry, and academic environments. The software engineer is seen as the most competent of technical professionals, while also appreciating the management and business aspects of software development.

The A. James Clark School of Engineering has recognized both this demand and the importance of software competency in its graduates, and is proposing a Software Engineering option under the Graduate Certificate in Engineering program to satisfy them. The intent of this option is to provide the education and training opportunities for individuals to move into the field and to offer advanced training for individuals currently in software development.

The Software Engineering option will be offered live at the College Park campus and via Instructional Television to remote sites around the State of Maryland. This option is proposed to start in the Fall 2003 semester, with an initial set of courses as outlined below. In the future, these courses could also form the core of a Software Engineering option under the Professional Master of Engineering program. The list will be augmented with selections from Systems Engineering, Computer Science, and current topics in software engineering.

Prerequisites: BS in Electrical and Computer Engineering or Computer Science and competency in one programming language. CMSC 435 (Software Engineering) can be substituted for ENPM 808G.

A request for a permanent number for ENPM 808G has been made. It will become ENPM 611 effective Fall 2003. ENPM 808U and ENPM 808S are planned to be offered in the Spring 2004 and Fall 2004 semester respectively. After they have been taught once, requests will be made to assign permanent numbers to them, namely, ENPM 612 and ENPM 613.
Course Listings:

**ENPM 808G (ENPM 611) Software Engineering** - This course covers software engineering concepts, methods, and practices important to both the theorist and the practitioner. The entire range of responsibilities expected of a software engineer are presented. The fundamental areas of requirements development, software design, programming languages, and testing are covered extensively. Sessions on supporting areas such as systems engineering, project management, and software estimation are also included.

**ENPM 808U System & Software Requirements** - This course focuses on the theoretical and practical aspects of requirements development. Students will recognize the place of requirements, how to work with users, requirements methods and techniques, the various requirements types, how to set requirements development schedules, requirements evolution, how to model and prototype requirements, how to evaluate and manage risk in requirements, techniques to test requirements, how to manage the requirements process, and how to write an effective requirements document.

**ENPM 608 Software Design & Implementation** - This course covers software design concepts and practices within the field important to both the practitioner and the theorist. Architectural and detailed designs are included for batch, client/server, and real-time systems. Design considerations for structured, object-oriented, and Web-based systems are covered. Design of databases, user interfaces, forms, and reports are also included. Implementation issues that affect the design, including error handling, performance, and inter-process communication, are presented.

**ENPM 808S Software Testing & Maintenance** - This course covers aspects of software development after coding is completed. Students will understand the various levels of testing, techniques for creating test data, how to manage test cases and scenarios, testing strategies and methods, testing batch, client/server, real-time, and Internet systems, and the development of an effective test plan. Software maintenance will include the creation of easily maintained software; preventative maintenance, corrective maintenance, and enhancements; configuration management practices; and assuring quality in software maintenance.
Course Description:

This course focuses on the theoretical and practical aspects of requirements development. Students will recognize the place of requirements, how to work with users, requirements methods and techniques, the various requirements types, how to set requirements development schedules, requirements evolution, how to model and prototype requirements, how to evaluate and manage risk in requirements, techniques to test requirements, how to manage the requirements process, and how to write an effective requirements document.

Course Objectives (this course should enable you to):

1. Analyze requirements for relevancy and practicality, and to recognize incomplete and inconsistent requirements.
2. Construct requirements scenarios and prototypes.
3. Develop requirements validation processes.
4. Manage requirements development in conjunction with other project activities.
5. Use the latest approaches to requirements development, such as the Unified Modeling Language.
6. Recognize the differences between non-functional and functional requirements.
7. Write an effective Requirements Document.

Required Text:

Course Description:

This course covers aspects of software development after coding is completed. Students will understand the various levels of testing, techniques for creating test data, how to manage test cases and scenarios, testing strategies and methods, testing batch, client/server, real-time, and Internet systems, and the development of an effective test plan. Software maintenance will include the creation of easily maintained software; preventative maintenance, corrective maintenance, and enhancements; configuration management practices; and assuring quality in software maintenance.

Course Objectives (this course should enable you to):

1. Understand the software verification and validation processes.
2. Recognize when independent verification and validation is needed.
3. Separate verification efforts to test requirements, design, and code.
4. Develop test scenarios and test cases.
5. Write an effective Test Plan and Test Procedures.
6. Recognize software developments that are easy and hard to maintain.
7. Develop maintenance management plans for various kinds of systems.
8. Maintain systems without source code, and systems for which the source compiler no longer exists.

Required Text:

TBD