

ROWAN UNIVERSITY CURRICULUM PROPOSAL

10

PROPOSAL TITLE:
Principles of Software Engineering

CHECK APPROPRIATE: UNDERGRADUATE GRADUATE 3 SEMESTER HOURS

SPONSOR(S):
A. Michael Berman

DEPARTMENT/TELEPHONE # Computer Science / 4743

CHECK ONE: COURSE MINOR PROGRAM CONCENTRATION SPECIALIZATION
 ACHIEVEMENT CERTIFICATE CERTIFICATION PROGRAM MAJOR PROGRAM

Step #1 (Department)	Step #2 (Receipt)	Step #3 (School)
<input checked="" type="checkbox"/> Approved (Date) <u>10-24-97</u> <input type="checkbox"/> Not Approved (Date) <u>[Signature]</u> Dept. Curriculum Chr. <u>Oct. 24 1997</u> Reviewed (Date) <u>[Signature]</u> Dept. Chr.	<u>SCC# 97-98-98</u> <u>10-24-97</u> Date Received Senate <u>[Signature]</u> Senate Curriculum Chr.	Reviewed Date: <u>11-17-97</u> <input checked="" type="checkbox"/> Recommend to Approved <input type="checkbox"/> Recommend NOT to Approve Forward for Open Hearing: <input type="checkbox"/> WITHOUT Reservations <input type="checkbox"/> WITH Reservations: Comments: <u>[Signature]</u> School Committee Chr.

Step #4 (Academic Dean): Recommended NOT Recommended Conditionally Recommended (See Comments)

Comments:

Dean Signature/Date: [Signature]

Step #5 (Senate Curriculum Committee): Open Hearing Date: 9-20-98 Approved by Curriculum Committee Date: _____

Returned to Sponsor(s) for the following reason:

Step #6 (Senate) Date announced/voted on at Senate: 9-24-98 If voted on: Approved NOT Approved

Date forwarded to Executive Vice President/Provost: 3-31-98

Senate Curriculum Committee chair Signature/Date: [Signature] 5/31/98

Rowan University
Department of Computer Science

Course Proposal

Principles of Software Engineering

1. Details

- a. Course Title: Principles of Software Engineering
- b. Sponsor: A. Michael Berman, Computer Science Department
- c. Credit Hours: 3
- d. Course Level: Junior/Senior
- e. Curricular Effect: Required course for Computer Science majors; possible elective for Engineering or MIS students.
- f. Prerequisites: Data Structures and Algorithms (0704.222) or Digital II, ^{0909.242} Public Speaking (1506.202), Probability and Statistics (~~1701.131~~) - ~~calculus~~ ^{1702.360}
- g. Suggested Time, Implementation: Two sections per year
- h. Resources: Faculty, equipment, and library resources are adequate

2. Rationale

Software Engineering is the application of Computer Science, Engineering, Mathematics, and Business principles to develop high-quality software that meets user's specifications. By applying rigorous standards and formal tools to the software development process, Software Engineering causes software development to become something closer to an Engineering discipline. Since nearly all Computer Science graduates become Software Engineers, it is essential that we teach how to apply their Computer Science training, using modern principles and techniques. In addition, since many Engineers serve as members of software development teams, the College of Engineering may add this course as an elective. Because modern software is created using teams, students will work in teams on mini-projects that will expose them to several phases of software development.

3. Essence of the course

a. Objectives in relation to student outcome

Students will be able to

- describe the overall process of software development – the “Software Lifecycle”;
- describe in detail each phase of software development;
- create the relevant documentation (“deliverables”) for each phase;
- create test plans and perform software testing procedures;
- use one or more Software Engineering tools;

- understand the ethical and legal context of Software Engineering;
- apply engineering design principles (such as modularity, separation of concerns, etc.) to the development of software;
- conduct code and design walkthroughs.

b. Topic outline

Introduction to Software Engineering
The Software Process and its Problems
Software Life-Cycle Models
Tools for Software Engineering
Principles of Software Testing
Capturing User Requirements
Developing Specifications
Planning and Estimating Software Projects
Software Design and Architectures
Implementation and Integration
Software Maintenance
Software Process Improvement

c. Evaluation and grading procedure of students

Students will be evaluated based on team mini-projects, one or more in-term examinations, and a final examination.

d. Course evaluation

This course will be evaluated by the department curriculum committee.

4. Results of consultation

I consulted with Dr. John Schmalzel, program director for the College of Engineering's EE major, and with Dr. Robert Fleming, chair of the Management Department in the College of Business Administration. The results are attached.

I have also attached a letter of support from an alumna of our department, Kathy Meehan, who currently works as a Software Engineer for Data Transformation Corporation in Blackwood, New Jersey.

5. Catalog Description

0707.321 Principles of Software Engineering

(Prerequisites: 0704.222 Data Structures and Algorithms or Digital II, 1506.202 Public Speaking, 1701.131 Probability and Statistics)

This course is an introduction to the discipline of Software Engineering. Students will explore the major phases of the Software Lifecycle, including analysis, specification, design, implementation and testing. Techniques for creating documentation and using software development tools will be presented. Students will gain experience in these areas by working in teams on mini-projects.

Subject: Course Proposal

Date: Wed, 22 Oct 1997 16:08:01 EST

From: "JOHN SCHMALZEL" <schmalzel@rowan.edu>

Organization: Rowan University

To: BERMAN@rowan.edu

CC: SHREEK@rowan.edu, RAVI@rowan.edu, CHIN@rowan.edu

<<< Message autoforwarded from BERMAN >>>

Mike,

Thanks for your efforts on the Software Engineering Course!

Our students need this (required) for the E&CE program. They will have Digital II, which is additional software (C, Assembly) based in a microprocessor environment.

I look forward to working with you on this course!

jls
John L. Schmalzel
College of Engineering
Rowan University
201 Mullica Hill Road
Glassboro, NJ 08028
Ph: 609.256.4629
Fx: 609.256.4950



Management & Management Information Systems

TO: A. Michael Berman
Computer Science Department

From: Robert S. Fleming
Chair, Management & MIS Department

Date: October 23, 1997

Subject: Principles of Software Engineering

I have reviewed the "Principles of Software Engineering" course proposal prepared by your department. On behalf of the Management & MIS Department, I commend you on the quality of this proposal.

One of the most interesting and valuable courses that I took in my graduate studies in computer and information science was a course in software engineering. I have always thought that such a course would be extremely important in the preparation of our Computer Science students. The proposed course would also be valuable for some of our Management Information Systems students.

It is my pleasure to offer the enthusiastic support of our department for this proposal. Should I be able to provide further support for this proposal, please feel free to contact me.

October 22, 1997

Dr. Mike Berman
Rowan University
Computer Science Department
Glassboro, NJ

Dear Dr. Berman:

I would like to emphasize how important I feel it is for a course in Software Engineering to be added to the current Computer Science curriculum at Rowan University. I, myself, am a Rowan graduate (class of 89) who majored in Computer Science and Mathematics. While I felt technically prepared for a career in computer science when I left Rowan, I feel there were many topics necessary for a career in software development that were not covered in my undergraduate studies. I went on to obtain a master's degree in Software Engineering from Monmouth University and I found that almost none of the software engineering subject matter was repetitive of anything I had seen as an undergraduate at Rowan. I also found that the subject matter covered in my master's studies would have been most beneficial before I began my career as a software engineer.

Students leaving Rowan with a degree in computer science will most likely become titled software engineers. Most computer science careers now involve working as a part of a software development team on a large software product. There are several software engineering topics that I feel are essential to anyone beginning a career in the software industry.

For example, I feel that the software development life cycle is a very important concept for new graduates to understand. This involves the evolution of a software product from user requirements to specifications to a design phase and eventually coding, testing and integration phases. Without this concept, engineers may not understand how the product on which they work is developed. Also, at each phase of the development cycle there are different types of documents with which the students should be familiar. Among these are requirements documents, specifications, and design documents. They will be expected to review existing documents as a basis for developing their own software units and will also be expected to develop parts of these documents themselves. Therefore, it is important that they understand the contents and purposes for these documents.

I also feel it is important for graduates to understand the nature and qualities of software products themselves. For example, qualities to design into software should be reliability, robustness, expandability, reusability and interoperability. Also, performance issues should be understood. Until I began my career, I had never considered the issues of software performance or maintainability – two essential issues for the success of any software project.

In addition to the qualities of the software itself is the importance of the software development processes themselves. In order for software engineering to become as reliable and visible as other engineering disciplines, it is necessary for those involved in the process to use formal, rigid, repeatable processes when developing the software product. This may involve design and code reviews, specific testing procedures, configuration management and other tools that allow the developer to trace the original requirements back to the end system. Currently, software engineering companies are working to be rated on a level of one to five by the Software Engineering Institute based on their development processes. It would be most beneficial to understand those processes before entering a computer science career.

There are also some software engineering design principles not discussed in current undergraduate studies. These include issues like separation of concerns, modularity, and abstraction – all vital in designing a successful piece of software.

There are other issues related to the management and existence of software projects that would be helpful for anyone planning a long-term career in the software industry. Skills such as project planning, estimation of software cost and source lines of code, and resource planning are necessary for anyone in the industry. Again, these issues are not discussed in current undergraduate courses.

Finally, there are tools used in most software development processes to which the students should have been exposed. These are referred to as Computer Aided Software Engineering (CASE) tools. These include tools such as debuggers, path testing tools and configuration management tools. The student should have some level of exposure to these concepts and how they are used on a software development project.

While I have mentioned many topics that could be covered in a software engineering course, there are many others that I have not mentioned that would be beneficial for students. I also feel that a hands-on practicum would be most beneficial in exposing the students to the different phases of the software development life cycle as well as providing an opportunity to work in a simulated team environment.

Again, I feel the computer science curriculum at Rowan University fully prepares students for the technical aspects of a career in software development. However, I feel that the addition of the above topics is necessary in order for Rowan students to be productive, competitive member of the software development community.

Sincerely,

A handwritten signature in black ink, appearing to read 'Kathleen Meehan', written in a cursive style.

Kathleen Meehan