

A Project Oriented Course on Software Engineering

By Sallie Henry

TR 90-23

A PROJECT ORIENTED COURSE
ON SOFTWARE ENGINEERING

Dr. Sallie Henry
Department of Computer Science
University of Wisconsin - LaCrosse
LaCrosse, Wisconsin 54601

I. Introduction

An undergraduate course in Software Engineering has been offered at the University of Wisconsin-LaCrosse for the past three years. The intent of the course is to present a broad overview of most of the areas of Software Engineering. Most Software Engineering courses offered will also cover the general aspects of Software Engineering. However, UW-LaCrosse students participate in a group project where they apply these Software Engineering concepts. Each team of students design a system, make a system design presentation, "hire" programmers, integrate the system modules, and document their system. The administration of group projects in general is complicated; however the administration of this type of project with the large amount of interaction among the students is worse. This paper makes an attempt to outline this Software Engineering course with particular attention on the administration of the project.

Section II describes the type of students taking the course, and a general description of the topics covered in the course. In section III the project itself is discussed with attention focused on the need for a project, what is expected of the students and the administration of the project. Section IV describes how parts of the course have been automated to relieve some of the problems of administration. Finally, the last section of the paper summarizes the impact of this course on the students and their employers.

II. General Explanation of Topics Covered

This is a beginning course in Software Engineering. The prerequisites are

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

two courses in programming and Data Structures. It is recommended that the student take Software Engineering immediately after Data Structures and before the more difficult upper divisional courses. Due to the amount of interaction among the students, the course becomes very time consuming. Since the course is an elective and as stated above requires a large time commitment, the students taking the course are highly motivated in Computer Science.

This is a very broad course in Software Engineering. Several topics are covered, but few are covered in depth.

The following issues are discussed:

1. General issues in Software Engineering
2. Software Reliability
3. Design Languages
4. Software Design Methodologies
5. Testing Methodologies
6. System and Code Walkthroughs
7. Proof of Correctness
8. Software Quality Complexity Metrics

General Schedule For Software Engineering Course

- Week#1: General introduction to Software Engineering
Essays on Software Engineering (BR75)
Software Life Cycle (MY76)
- Week#2: Selection of teams (See Section IV)
Discussion of Design Methodologies
Jackson's Data Structure approach to Design (BE79)
Parnas' Information Hiding (PA76A) and "uses" hierarchy (PA76B)
Yourdan's Composite Design (BE79, YO75)
Myers' Levels of Cohesion (MY76)
- Week#3: Discussion of Design Languages
PSL/PSA (BE79)
RDL (Requirements Design Language-Sperry Univac)
ADLIF (See Section IV)
- Week#4: White box testing methodologies (MY79)
Approval of group projects
- Week#5: Black box testing methodologies (MY79, MC83)
System Walkthroughs
Code Inspections