HOURS: Lectures MWF 9-9:50am NL121 3 credit hours

PROFESSOR: Dr. Stefan Brandle

OFFICE AND CONTACT INFORMATION: NS 008C, x84685, sbrandle@css.taylor.edu
Office Hours: MWF 11am-noon, 1-2pm; T 9-10am. You may also come at other times. I will normally be in my office if I am not in class. Making an appointment will help ensure that I am available at a particular time, but you should also feel free to just drop in.

CLASS WEB PAGE: http://www.css.taylor.edu/~sbrandle/340

Catalog Description: A study of the concepts, procedures, and tools of large system software project development, including project estimation and management, software technical metrics, configuration management, and software testing. Concepts of software engineering are introduced using the development of a large software system as an instructional illustration. The project is designed and its development managed using the methods and techniques examined in the course.

COURSE OUTCOMES: As a result of taking this course the learner will

- Be able to identify and communicate the current challenges of developing quality software.
- Know the current success/failure rates of software projects and the basic contributing factors for success or failure.
- Understand what Software Engineering is and how it differs from coding.
- Be able to identify and understand the use of common Software Engineering practices that are necessary for engineer quality software.
- Be able to evaluate a process framework and identify how effectively it addresses each of the activities identified in the generic process framework.
- Be able to communicate the differences between prescriptive and agile processes (methodologies) and be able to think critically about how to choose the appropriate process for a given project.
- Be able to design a physical environment that is conducive to developing quality software, taking into account physical factors that hinder developer productivity.
- Be able to identify the characteristics of poorly designed software, and be able to implement basic design principles that address each of those issues.
- Be able to identify the different types of testing, know when they should be used and how they should be conducted.
- Be able to write automated unit tests against a software application including the use of Test Driven Development techniques.
- Understand the benefits of automated acceptance tests and be able to implement code using automated acceptance testing techniques.
- Understand how product metrics contribute to the quality of software, the challenges of identifying good product metrics and know basic product metrics that are commonly used.
- Understand the importance of configuration management of software products and what the components are necessary for of a complete configuration management package.
- Understand and identify the need for and the techniques used to estimate software projects.
- Be able to articulate the challenges of estimation and how to deal with the "political" issues that surround estimation efforts.
- Understand the techniques used to schedule projects that use either prescriptive or agile processes.
- Understand the multiple facets of project management and have a basic understanding of the role of a project manager.
- Understand how project metrics are used to increase probability of success for a project and how to design a metric to measure a specific aspect of a project.
- Understand risk management, and know techniques for identifying, assessing and mitigating risks.
- Be able to identify quality management practices used in Software Engineering.
- Understand the importance of focusing effort and resources on creating a people friendly work culture, and will know what factors should be addressed to create such a culture.
- Be able to discuss the factors that contribute to growing productive teams.

TEXTS: "Peopleware: Productive Projects and Teams" 2nd ed., Tom DeMarco and Timothy Lister (required)
"Software Engineering: A Practitioners Approach" 6th ed., Roger S. Pressman (required)

OTHER MATERIALS: The course will provide software tools for assigned projects through use the CSS department servers. There will also be online materials.

PERFORMANCE ASSESSMENT:

Exams (classroom)	= 35%
Class presentations	= 10%
Quizzes	= 10%
"What is Software Engineering" report	= 15%
Projects	= 30%

	100%

GRADING: Exams will attempt to cover the more conceptual material of the course, while the assignments will evaluate practical application. Quizzes will be used to provide feedback about how well you understand the material.

The semester grade will be assigned based on the following scale.

		100-93	A	>= 90	A-
>= 87	B+	>= 83	B	>= 80	B-
>= 77	C+	>= 73	C	>= 70	C-
>= 67	D+	>= 63	C	>= 60	D-
< 60	F	Incomplete	I		

ATTENDANCE: Attendance is expected and can alter the course grade. Situations such as illness or serious illness/death of a family member are valid reasons for an absence to be excused. Prior notification of excused absences is expected if possible. Documentation may be required. It is the student's responsibility to verify that the instructor has been notified of excused absences for any reason, including official university functions. The due date of an assignment will not be automatically extended as a result of an excused absence.

FINAL EXAM: The final exam will be similar to the other exams, but will be longer and comprehensive. However, the final exam will emphasize material covered since the previous exam.

QUIZZES: There will be an online quiz most weeks. The quiz will cover the assigned reading and homework for the week. The quiz is your chance to determine whether you're understood the class material.

ASSESSING THE COURSE AND INSTRUCTOR: In order to improve this course, I will give you the opportunity to assess my teaching and the course content during the term. Both structured and free-form evaluation will used. Please feel free at any time, however, to let me know how you think things are going and/or how they might be improved.

CHEATING: The standard departmental cheating policy is attached and your adherence to it will be expected.

COURSE SCHEDULE: See web page for details

EVALUATION WEEK – Final exam is on Wednesday, December 12, 8:00am-10:00am

DEPARTMENTAL CHEATING POLICY
TAYLOR UNIVERSITY COMPUTING AND SYSTEM SCIENCES

GUIDELINES AND DEFINITIONS

CSS faculty will assume the honesty of students as indicated by their assent to the Taylor University Life Together Covenant and will not, unless circumstances indicate otherwise, aggressively search for instances of cheating. However, the CSS faculty will not condone cheating. When cheating is suspected, reasonable action will take place to establish what actually occurred. It is impossible to provide a definition of cheating that is unambiguous and applies to all possible situations. Each case of suspected cheating must be carefully analyzed on its own merit.

Two of the guidelines that will be used to determine whether cheating has occurred are:

- Program plagiarism will be suspected if an assignment requiring independent development results in multiple solutions so similar they could easily be produced by mechanical translation. However, it is recognized that some assignments will have narrow enough parameters that very similar solutions are prescribed by the nature of the assignment.
- Cheating will be suspected if a student who was to complete an assignment independently cannot explain the details of the solution or the techniques used to generate the solution.

Extremely serious incidents of cheating will be labeled ***flagrant*** and will require more severe penalties.

While it is not possible to exhaustively list or categorize all possible situations, some examples may be useful and are listed below. Individual instructors may state requirements for certain assignments that will supercede these examples.

EXAMPLES OF CHEATING

- Submitting another's work as your own (with or without their knowledge)
- Allowing someone else to submit your work as their own
- More than one student working on an assignment and submitting multiple copies represented (implicitly or explicitly) as individual work
- Using a solution developed by a student in another term
- Discussion of the general approach to a problem solution that involves viewing another student's code
- Use of a solution obtained from the internet or other external source

EXAMPLES OF FLAGRANT CHEATING

- Stealing an examination or solution from an instructor or student assistant
- Obtaining a solution from another student without their knowledge in a manner that involves deceit, making false statements after the act, or a breach of system security
- Submitting another's work that is completely duplicated

*EXAMPLES OF **NOT** CHEATING*

- Submitting work done alone or with assistance from course staff
- Submitting one assignment for a group of students if group work is permitted or required
- Obtaining or giving assistance in the use of computer systems or tools needed for completion of an assignment
- High-level discussion of course material to better understand the problem and potential solutions
- Discussion of assignments to ascertain the requirements (NOTE: It is best to consult the instructor or student assistants assigned to the course rather than other students in that it is easy for such questions to degenerate into implementation details.)

ASSESSMENT AND PROCEDURES

The following general procedures will be followed when an instructor suspects that cheating may have occurred.

- A standard form will be filled out for each student suspected of cheating. This form will include the name of the student being investigated and the names of other students potentially involved. Since levels of involvement will be different for each student, a separate form will be used for each student.
- If it is determined that no cheating has occurred or if it is not possible to make a definitive judgement, the student(s) names will be removed from the form, but it will be kept on file.
- If it is determined that cheating has occurred, the form will be kept on file in order that repeated offenses may be detected. The student's name will be removed from the file after the student graduates.
- At least one additional CSS faculty member will be consulted. Reasons for this requirement include:
  - fairness to the student so that a single faculty member will not make an error in judgment
  - protection for the faculty so that a student may not charge an individual faculty member with bias
- The assessment of the situation may involve multiple faculty members and/or discussion at a department meeting. The assessment will often involve assistance from CSS technical staff and may involve assistance from Information Services staff.
- At least one meeting between the instructor and the student will be held. The consulted faculty member may be included in this meeting. The instructor will maintain notes of the meeting(s).
- The instructor in the course will maintain a file of material gathered in the investigation. If it is determined that no cheating occurred, this file will be destroyed at the discretion of the instructor. If it is determined that cheating has occurred, the file will be maintained at least as long as the student's name is retained on the form filed on the incident.

PENALTY

- If it is determined that a first, non-flagrant cheating offense has occurred, the penalty will be at least as severe as that for not submitting the work in question. The exact penalty will be determined by the instructor in consultation with the faculty consultant involved in determination of guilt. The CSS Department Chair or the entire department faculty may be consulted.
- The penalty may be more severe including failure in the course. An example of conditions warranting a more severe penalty is the failure of the student to acknowledge guilt in the face of hard evidence.
- Repeated or flagrant offenses will result in failure in the course.
- All investigations resulting in a determination of cheating will be reported to all CSS faculty and technical staff.
- All investigations resulting in a determination of cheating will be reported to Academic Affairs. If the student disagrees with the assessment, the student's viewpoint will be represented.
- All cases of repeated or flagrant offenses (or any case resulting in failure in the course) will also be reported to Student Development.
- If a cheating offense involves violation of computer use policy (sharing of a password, etc.), that policy may also include additional penalties. Such penalties may include restriction on use of lab facilities that could have additional effect on the grade in the course or in any courses requiring the use of computer facilities.

Revised January, 2001