CSC 323-001, Software Engineering, Fall 2017
MW 1 – 2:15, McKibben Room 322

Professor David Cook, Ph.D.  cookda@sfasu.edu, 303F McKibben, 468-2508

CREDIT HOURS: 3 (PREREQUISITES: CSC 241 with a C or better)


PURPOSE OF COURSE: To provide the student with knowledge of software engineering principles that can be applied to the software process.

OFFICE HOURS: M/W 11-12:30, and 2:15 – 3:15. Tuesdays from 1-3:30, and Thursdays from 12:15 – 12:45 I am also available by appointment as needed. You should make an appointment to see me outside of office hours, but if my office door is open, you are free to drop in. I do require appointments for Fridays.

REQUIRED TEXTS: The SWEBOK (Software Engineering Body of Knowledge), provided electronically. The Mythical Man Month, by Frederick Brooks (available online). You will also be provided with multiple readings (both hardcopy and electronic) from many other sources.

REQUIRED ITEMS: There will be multiple assignments, group projects, oral presentations, and written work. Assignments count for 40% of the total grade. Many assignments will be submitted either by D2L, email or a flash drive.

EDUCATIONAL OBJECTIVES:
Upon successful completion of the course, students should be able to:

1. Identify software development problems that provided the impetus for the start of software engineering.
2. Demonstrate an understanding of the different perspectives from which software is considered by users, clients, and commercial and in-house developers.
3. Describe the importance of software maintenance, and the nature of the software life cycle.
4. Describe the various software process models that have been used for software development and gain familiarity with important software development methodologies.
5. Work in a disciplined software development team demonstrating the use of COCOMO, function points, and other methods to estimate the size of a development effort.
6. Produce important artifacts of software development other than code.
7. Demonstrate an understanding of the role of software quality assurance and practice non-execution based testing.
8. Develop a prototype as a means of requirements validation.
9. Derive and use metrics for software development.
10. Use state-of-the-practice software estimation techniques.

CONTENT

<table>
<thead>
<tr>
<th>Hours</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>History of software engineering</td>
</tr>
<tr>
<td>3</td>
<td>The need for a disciplined approach</td>
</tr>
<tr>
<td>3</td>
<td>Software process models</td>
</tr>
<tr>
<td>3</td>
<td>Quality, productivity, accuracy, reliability, maintainability, reusability</td>
</tr>
<tr>
<td>3</td>
<td>The use of metrics</td>
</tr>
<tr>
<td>3</td>
<td>The role of Computer-Assisted Software Engineering (CASE)</td>
</tr>
<tr>
<td>10</td>
<td>Requirements Engineering</td>
</tr>
</tbody>
</table>

CSC 323-001, Fall 2017
Requirements definition and analysis
Feasibility study
Cost/benefits analysis
Prototyping
Tools

Design ................................................................. ................................................................. ................................................................. ................................................................. ................................................................. 12
Methodologies: structured design, functional decomposition, data-flow oriented, data-oriented, object-oriented design
Tools

Implementation and Testing ................................................................. ................................................................. ................................................................. ................................................................. ................................................................. 10
Programming environments, teams, languages, and style
Programming principles: cohesion, coupling, modularity, information hiding
Test case design, classes of tests
Quality assurance, verification, validation, reliability
Testing methods
Tools

Evolution ................................................................................................................................. 4
Operation; performance analysis and measurement
Maintenance
Reverse engineering

Exams (plus final) ................................................................................................................................. 3

TOTAL 45

REFERENCES

EXAMINATIONS: (Approximately 50% of the course grade) – from the book, lectures, and outside readings
NOTE – all exams are comprehensive. All tests will be announced at least two weeks before the test.
Test 1  approximately 10% of your final grade, late September
Test 2  approximately 10% of your final grade, late October
Test 3  approximately 10% of your final grade, late November
Final (Comprehensive)  approximately 20% of your final grade, Mon, 11 Dec, 1:00 – 3:00

NOTE:
There are no exemptions from the final examination and no changes in taking the final examination.
Check the final exam time. If the final exam time is a problem, you need to drop this course. Only the Dean can approve a change in a final exam.

ATTENDANCE:
Attendance and constructive class participation are expected and count for 10% of your grade (i.e. one letter grade). I do not teach the book. Most of the material in this class comes from lectures, my PowerPoint files, and my 34 years of experience as a software engineer. I expect you to read the book yourself. My lectures introduce material not in the book, but my lectures are based upon your prior understanding of the book coverage. I will make every effort to give you copies of my slides on D2L - but you are responsible for being in class and taking notes on the material. There will be multiple readings assigned in class to supplement the book. I will post the PowerPoint files on D2L – but not necessarily before they are used.
EDUCATIONAL OBJECTIVES: Upon completion of this course, students should be able to

1. Identify software development problems that provided the impetus for the start of software engineering.
2. Demonstrate an understanding of the different perspectives from which software is considered by users, clients, and commercial and in-house developers.
3. Describe the importance of software maintenance, and the nature of the software life cycle.
4. Describe the various software process models that have been used for software development and gain familiarity with important software development methodologies.
5. Work in a disciplined software development team demonstrating the use of COCOMO, function points, and other methods to estimate the size of a development effort.
6. Produce important artifacts of software development other than code.
7. Demonstrate an understanding of the role of software quality assurance and practice non-execution based testing.
8. Develop a prototype as a means of requirements validation.
9. Derive and use metrics for software development.
10. Use state-of-the-practice software estimation techniques.

GRADING POLICY: My typical grading policy is the standard 90 cutoff for an A, 80 for a B, etc. Grades below 60 will receive an F. I will not raise the cutoffs. I reserve the right to lower the cutoffs by curving the grades – but the 90/80/70/60 cutoffs will never be raised. Note that there are several group projects – and you will receive both an individual and group grade. In addition, you will anonymously “grade” the contributions of all of your team members – and they will be “grading” you. These peer group grades will influence your final group grade.

LATE ASSIGNMENTS: In this class, I try to make it a “real world” experience in terms of changes to assignments (often at the last minute). However, just like the “real world” – there are hard deadlines. Late assignments typically count as a 0. Since a lot of your grade depends upon group assignments – you are strongly encouraged to make sure your team turns in deliverables on time.
Syllabus Addendum

Program Learning Outcomes:
Program learning outcomes define the knowledge, skills, and abilities students are expected to demonstrate upon completion of an academic program. These learning outcomes are regularly assessed to determine student learning and to evaluate overall program effectiveness. You may access the program learning outcomes for your major and particular courses at the CS Dept website/

General Student Policies:

Academic Integrity (A-9.1)
Academic integrity is a responsibility of all university faculty and students. Faculty members promote academic integrity in multiple ways including instruction on the components of academic honesty, as well as abiding by university policy on penalties for cheating and plagiarism.

Definition of Academic Dishonesty
Academic dishonesty includes both cheating and plagiarism. Cheating includes but is not limited to (1) using or attempting to use unauthorized materials to aid in achieving a better grade on a component of a class; (2) the falsification or invention of any information, including citations, on an assigned exercise; and/or (3) helping or attempting to help another in an act of cheating or plagiarism. Plagiarism is presenting the words or ideas of another person as if they were your own. Examples of plagiarism are (1) submitting an assignment as if it were one's own work when, in fact, it is at least partly the work of another; (2) submitting a work that has been purchased or otherwise obtained from an Internet source or another source; and (3) incorporating the words or ideas of an author into one's paper without giving the author due credit. Please read the complete policy at http://www.sfasu.edu/policies/academic_integrity.asp.

Withheld Grades Semester Grades Policy (A-54)
Ordinarily, at the discretion of the instructor of record and with the approval of the academic chair/director, a grade of WH will be assigned only if the student cannot complete the course work because of unavoidable circumstances. Students must complete the work within one calendar year from the end of the semester in which they receive a WH, or the grade automatically becomes an F. If students register for the same course in future terms the WH will automatically become an F and will be counted as a repeated course for the purpose of computing the grade point average.

Students with Disabilities
To obtain disability related accommodations, alternate formats and/or auxiliary aids, students with disabilities must contact the Office of Disability Services (ODS), Human Services Building, and Room 325, 468-3004 / 468-1004 (TDD) as early as possible in the semester. Once verified, ODS will notify the course instructor and outline the accommodation and/or auxiliary aids to be provided. Failure to request services in a timely manner may delay your accommodations. For additional information, go to http://www.sfasu.edu/disabilityservices/.

Acceptable Student Behavior
Classroom behavior should not interfere with the instructor’s ability to conduct the class or the ability of other students to learn from the instructional program (see the Student Conduct Code, policy D-34.1). Unacceptable or disruptive behavior will not be tolerated. Students who disrupt the learning environment may be asked to leave class and may be subject to judicial, academic, or other penalties. This prohibition applies to all instructional forums, including electronic, classroom, labs, discussion groups, field trips, etc. The instructor shall have full discretion over what behavior is appropriate/inappropriate in the classroom. Students who do not attend class regularly or who perform poorly on class projects/exams may be referred to the Early Alert Program. This program provides students with recommendations for resources or other assistance that is available to help SFA students succeed.

Electronic Devices:
I encourage the use of electronic devices such as tablets, computers, etc. to facilitate your learning. Most of my slides will be posted on D2L prior to class – you are encouraged to download and bring them with you. Note that computers, tablets, phones, etc. are to be used to support learning in my class – not for social media updating, web browsing, texting, doing homework for other classes, etc. If it becomes obvious that you are not using your electronic devices properly and disrupting the learning of other students, I will ask you to stop. After the second warning, I will ask you to leave the class. Note that all devices must be powered off, placed in a backpack or purse, and may not be used during tests. I ask that all phones be put on vibrate during class. Also – if you spend the hour browsing on the internet or doing other homework – please give me credit for recognizing and counting it against your participation.