CSC555 - SOFTWARE ENGINEERING
FALL 2006

Instructor: Richard G. Epstein
Anderson Hall 324 / 610-436-3231
E-mail: epstein@wcupa.edu
Web: www.cs.wcupa.edu/~epstein
Office hours: to be announced and posted on the Web

Texts: (Both required):
Software Engineering Coursepack (Estimated Cost: $10.00)

Required work: Your grade will depend upon the following factors:
Class participation: 5%
Submitted discussion questions: 10%
Team-oriented software process
document and in-class presentation: 40%
Individual research project: 25%
Final exam: 20%

Description of course:
This course has four fundamental goals:
(a) To introduce students to significant software processes in the software industry
(b) To introduce students to current research and "hot topics" in software engineering
(c) To get students to work on teams
(d) To devote some attention to the impact of the increased emphasis on computer security upon software engineering.

The software processes that we shall explore are:
(a) Personal Software Process (PSP)
(b) eXtreme Programming (XP)

1 I will know the exact cost by the time the semester begins.
2 There are really two grades involved here: 20 points for the written project and 20 points for the in-class presentation.
(c) agile processes (AG)
(d) Capability Maturity Model (CMM)

In recent years, eXtreme Programming (and other "agile methodologies")
have been a "hot topic" in software engineering. So, we want to devote quite
a bit of attention to this particular software process. However, we also
want to address the issue of whether eXtreme Programming is consistent
with engineering SECURE SOFTWARE.

One interesting option for your individual research papers would be to
address this issue by trying to accommodate security concerns into the
eXtreme Programming methodology (or, more generally, into an agile process
framework) or to otherwise discuss how consistent different processes are
with the goal of producing secure software.

Let me just say a few things about the Watts Humphrey book and the
Personal Software Process. I am assuming that most of you have not been
exposed to a formal software process. I want to give you some idea as to
what is involved with such a process without devoting most of the course to
learning ONE SOPHISTICATED PROCESS. Thus, we are going to spend just
a little bit of time (two weeks) studying a THE SIMPLEST FORMAL
PROCESS that I know, the Personal Software Process. Watts Humphrey’s
book does an excellent job of communicating what a formal process is about,
especially in terms of written processes, scripts, and forms to be filled out.
However, the Personal Software Process has had limited success in real
industrial settings, as we shall discover in some of our assigned readings.
Nonetheless, I still feel that this simple book is an excellent way to
introduce the idea of a formal process. (Another argument for covering PSP
is that training is PSP is a prerequisite for training in TSP, which is gaining
acceptance in the industrial sector.)

You will be receiving separate handouts relating to the team project. The
basic goal of the team project is to create a software process for a pretend
company using the ideas introduced in the course and other ideas that you
yourself have investigated. About a month before the end of the course
(mid-November) you will submit your pretend company’s documented
software process as a written document. At the end of the semester, each
team will present a presentation about their process as an in-class
presentation.
In addition, each student is to do an individual research project. I am somewhat flexible if a student comes up with an innovative idea for this project. The standard individual research project would be a research paper that investigates some aspect of this course which you find especially interesting. Other possibilities include personal experience reports on software tools (such as tools for testing and static analysis) and software processes in industry. Again, I am open to suggestions for your individual research projects. We will devote some time at the end of the semester to in-class presentations regarding your individual research projects and what you have discovered in your research.

There will be many reading assignments from the coursepack. For each reading assignment from the coursepack you are to submit discussion questions relating to the articles. The reason for this is that I do not want any students to fall back and depend upon other students to participate in our discussions of the assigned articles. The quality of your submitted discussion questions will almost entirely determine your “class participation” grade. Of course, if you make really fine comments in class, that will also help your “class participation” grade.

**Tentative Schedule**

<table>
<thead>
<tr>
<th>Week of Tuesday ...</th>
<th>Tuesday:</th>
<th>Thursday:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug 29</td>
<td>Intro</td>
<td>No Silver Bullet</td>
</tr>
<tr>
<td>Sept 5</td>
<td>PSP³</td>
<td>PSP</td>
</tr>
<tr>
<td>12</td>
<td>PSP</td>
<td>Team meetings</td>
</tr>
<tr>
<td>19</td>
<td>XP⁴</td>
<td>XP</td>
</tr>
<tr>
<td>26</td>
<td>XP</td>
<td>AG⁵</td>
</tr>
<tr>
<td>Oct 3</td>
<td>AG</td>
<td>CMM⁶</td>
</tr>
<tr>
<td>10</td>
<td>CMM</td>
<td>FLEX⁷</td>
</tr>
</tbody>
</table>

³ Personal Software Process  
⁴ eXtreme Programming  
⁵ Agile Processes  
⁶ Capability Maturity Model  
⁷ Flexibility
<table>
<thead>
<tr>
<th></th>
<th>fall break</th>
<th>SF	extsuperscript{8}</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>SF</td>
<td>PR	extsuperscript{9}</td>
</tr>
<tr>
<td>31</td>
<td>CC	extsuperscript{10}</td>
<td>CO	extsuperscript{11}</td>
</tr>
<tr>
<td>Nov 7</td>
<td>QA	extsuperscript{12}</td>
<td>Security</td>
</tr>
<tr>
<td>14</td>
<td>Security</td>
<td>Security</td>
</tr>
<tr>
<td>21</td>
<td>Security</td>
<td>Thanksgiving</td>
</tr>
<tr>
<td>28</td>
<td>Team presentations</td>
<td>Team presentations</td>
</tr>
<tr>
<td>Dec 5</td>
<td>Team presentations</td>
<td>Team presentations</td>
</tr>
<tr>
<td>12</td>
<td>finals week</td>
<td>finals week</td>
</tr>
</tbody>
</table>

	extsuperscript{8} Software failures
	extsuperscript{9} Professional responsibilities
	extsuperscript{10} Corporate cultures
	extsuperscript{11} Communications
	extsuperscript{12} Quality Assurance