About the course
While the term *software engineering* covers a broad range of topics, it in general applies to the overall process of creating software solutions. It isn’t just about writing code; in fact the implementation of software is usually the smaller part of a project. In CS411 we will take a close look at the various phases of software projects: Definition; Design; Development; Delivery; Management; and Maintenance. In each phase we'll explore current methodologies and see how the pieces fit together into successful projects. Overarching topics software quality and how to achieve it by applying specific techniques in each project phase, and best practices in architecting software solutions.

After completing CS411 you should be able to understand and apply the basic concepts of software engineering to plan and execute software projects in any phase. You'll also be exposed to a variety of tools currently used in software projects in industry.

Text
Our text this semester is Sommerville’s:

*Software Engineering*, 10th edition, Pearson
ISBN 978-0-13-394303-0 (hardcover)
ASIN B015TKCR92 (Kindle)

It's available at the bookstore (and Amazon, of course), for rental on Kindle, and also is available on CourseSmart in the 9th edition, which is pretty close to the 10th edition. Off by one, as they say.

Blackboard
We use Blackboard (http://learn.bu.edu) as a repository for the slide sets for each class, copies of homework assignments, sample code, and announcements. You should be enrolled already, so that when you log on to the site you'll see the course listed.

Piazza is our tool of choice for discussions, including group project team discussions.

Programming and Prerequisites
In order to give you a chance to practice what you have learned we'll be working in small teams to design and implement a software project, specifically a web application. At this point in your life you should have some experience in programming (in any language, though my understanding is that you all are Java and Python ninjas), but you'll be using technologies and tools that you might or might not be familiar with, including client-side web page processing with JSP/ASP/PHP or Node.js, MySQL, NoSQL, and Mongo databases, back-end processing in
Python or Perl or Javascript… basically the MEAN or XAMP stack. My philosophy of teaching is that it’s important to learn theory, but even more important to put the theory into practice, so about half of our time will be spent exploring tools and techniques.

Don’t panic if you aren’t an expert in everything that we talk about…that’s the whole point of you being in class. Good software engineers are what IBM calls 'T-shaped' people…broad knowledge of lots of different things, but with the ability to dive deep when needed. It isn’t possible to build robust systems with just a narrow understanding of a few aspects of computing. Anyway, we will work together (and in teams) to figure out things like versioning strategies, proper n-tier architectures, how to normalize database tables, and so on. If you haven’t used git before, now you will.

**Tools**
Software engineers spend a lot of time learning tools and toolchains. I’ll be using resources in class that you might find useful, too, and they’re free to boot. You might need to poke for the free academic versions of these.


*Jetbrains* ([https://www.jetbrains.com](https://www.jetbrains.com)) produces high-quality IDEs for various platforms, including WebStorm (Node.js), pyCharm (Python), intelliJ IDEA (Java), PHPStorm (PHP) and RubyMine (Ruby)

You are of course welcome to use other tools that you might be comfortable with, and we’ll from time to time highlight a tool in class.

**Grades**
There are three major grading opportunities in the course. We’ll have a written exam in the first and second half, and you’ll also be working on a project in small teams that will play a significant role in your final grade. There will be several work products (specifications, test plans, and so on) that will contribute to your grade. Most weeks there will be a short quiz (5 or 6 questions) covering the previous lecture’s material. Case studies will be assigned during the semester, and you’ll be expected to write a short summary of each in addition to participating in in-class discussions.

The allocation for your final grade looks like this:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm exam</td>
<td>25%</td>
</tr>
<tr>
<td>Final exam</td>
<td>30%</td>
</tr>
<tr>
<td>Quizzes / Cases</td>
<td>15%</td>
</tr>
<tr>
<td>Team project</td>
<td>20% (all team members receive the same grade)</td>
</tr>
<tr>
<td>Team 360º review</td>
<td>10%</td>
</tr>
</tbody>
</table>

The grading scale is numerical:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Score Range</th>
<th>Letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>96-100</td>
<td></td>
</tr>
<tr>
<td>A-</td>
<td>90-95</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>80-84</td>
<td></td>
</tr>
<tr>
<td>B-</td>
<td>75-79</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>65-69</td>
<td></td>
</tr>
<tr>
<td>C-</td>
<td>60-64</td>
<td></td>
</tr>
<tr>
<td>B+</td>
<td>70-74</td>
<td></td>
</tr>
<tr>
<td>C+</td>
<td>65-69</td>
<td></td>
</tr>
</tbody>
</table>
If your course grade happens to be close to a boundary, such as an 89.5, I'll bump you up unless your overall course work for some reason doesn't justify it.

**Team Project and 360° Reviews**
Teams are formed in the first few weeks to work on a project together. The project parameters are contained in a series of documents / assignments posted on our course site. All team members will receive the same grade for their project. Broadly speaking, what I'm looking for in project work is that your team follows a design and development process in a consistent way, using tools and techniques that we discuss in class.

In addition to the team grade, each team member will receive a 360° Review grade. For the 360° Review, each team member will evaluate the participation and contribution of the other members of the team. We do this so that if only one or two team members do all of the work, the non-working members don't get a free ride in terms of the project grade.

**Contacting me and office hours**
The best way to contact me is by email. You may also IM me at perryd@bu.edu (iMessage). My office is in the Psychology building at 64 Cummington Mall, room PSY228A. Office hours will be posted on Blackboard. No appointment needed, just drop by if you have a question or want to hang out a bit. If you need to drop something off, my mail slot is in the CS office in MCS138.

**Computing**
We (and by 'we' I mean 'you') will be doing a fair amount of coding for your projects, so you will need to have access to something that you can code on. A laptop is probably the most convenient, but there are also labs with both Windows and Linux machines available for your use in the Radio Shack building. I'm sure there is some official name for that building, but it's the one with a Radio Shack at street level…the undergraduate labs are on the third floor. Most of you are seniors and should have figured this out by now, but I'm happy to point you in the right direction if you need help.

For our (and by 'our' I mean 'your') projects I recommend that you set up an Amazon Web Services virtual server using their Elastic Compute Cloud (EC2) service, and when we get to that point I'll provide advice on getting instances up and running. It's free and fun and a useful skill to learn. It's also fine to set up MEAN or XAMPP stacks on your personal machines if that's the way you prefer to work. Once the teams are assigned it makes sense to set up a git repository, too.

**Academic Conduct Code**
The University the College take cheating very seriously. Cheating and plagiarism will not be tolerated in any course. Cases will be referred to the Dean's office and may result in loss of credit for an exam or assignment or other disciplinary action.

By nature, programming is a collaborative effort, and I fully expect that you will use resources such as Google, fellow students, and our own discussion forum on Blackboard to learn the material and do your assignments. We'll discuss code and approaches in class, and I'll occasionally post sample code on Blackboard that you are welcome to use as a starting point.
However, I definitely don’t want you to simply copy entire programs that you find on the web and turn them in as your own work. If you do use more than a line or two of someone else’s code (including mine), just make a note in a comment in your program to point to where you got it.

Dr. Sullivan’s guidelines for collaboration are a good explanation of our expectations. You can read them at http://cs-people.bu.edu/dgs/courses/cs111/collaboration.html.

**Changelog**

**v.2016.SPRING.SYLLABUS.1.0**
Initial version, forked from v.2015.FALL.SYLLABUS.1.1