DAY 4 NOTES

Today we'll go over chapter 2 slides.

Here are the topics the students found regarding cutting edge computing and software engineering.

Mark – 1. ULTRA-Large scale computing systems 2. Bacterial computing. 3. Proof-carrying code
Russell – 1. Real-time systems (jasdeep-2) 2. Testing → quality 3. Lifecycle → security

Chapter 2 – Modeling the process and life cycle (Pfleeger’s slides)

Software Development (Dr. Saad sounds doubtful about these being in the proper order... See Chapter 2, Slide 8 for the book’s steps)

2) $Money$?!
3) Requirements
4) Design
5) Implementation
6) Test – Internal/External
7) Deploy
8) Maintain and back to step 1.

Percentage of time spent

<table>
<thead>
<tr>
<th>CASE: Computer-Aided Software Engineering</th>
<th>Available</th>
<th>Not Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>3) Analysis (after What &amp; Money)</td>
<td>19%</td>
<td>7%</td>
</tr>
<tr>
<td>4) Design</td>
<td>49%</td>
<td>36%</td>
</tr>
<tr>
<td>5) Coding</td>
<td>32%</td>
<td>57%</td>
</tr>
</tbody>
</table>

Only 27% of companies surveyed said they used case tools. Case tools help you figure out how you're going to build the software. Case tools allow you to spend more time with analysis & design, and less time coding. Why? Plan things out early on so you don't stumble around while coding.
Process – A series of steps involving activities, constraints, and resources that produce an intended output of some kind. Involves a set of tools and techniques.

Process Characteristics... See slide 5

1. Clarity & Integrity
2. Completeness (Your document should be clear in capturing what you want the software to do)

The importance of process (slide 6)

Reasons for modeling a process (slide 7)

Software Life Cycle (slide 8)

Software Development Process Models (slide 9)

Waterfall Model (just see the slides...)

What is prototyping? - Requirements analysis, system design, program design...Stuff you do before writing any code. Work with the customer.

V Model (see slides) – Same steps, but different arrangement.

Prototyping model (see slides) - “Just do it”... Build&revise until you run out of time or money.

Operational Specification Model – Most of the time is spent in Transformed Specification.

+Behavior-driven development

Transformational Model – Relies on formalism.


Spiral model – BUDGET! :)

Agile methods & Extreme – [next class]

A few high-level observations. You have a process with pros&cons. Each box has an input & output to its process.

What do we mean by FORMAL METHODS – logic, grammar, predicate logic, etc... First order, second order. Can you describe your software using logical and mathematical formulation. You can then prove things using math.

FOR NEXT CLASS, READ CHAPTER 2 & GO THROUGH CHAPTER’S SLIDES AHEAD OF TIME.

DR. SAAD – PROVIDE PAPER ON FORMAL METHODS.

Bring 1-3 papers for each of the models and have pointers.

Waterfall Jasdeep

V BEN (and show the iteration of the V)
<table>
<thead>
<tr>
<th>Method</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prototyping</td>
<td>Ivan</td>
</tr>
<tr>
<td>Operational</td>
<td>Javier</td>
</tr>
<tr>
<td>Transformational</td>
<td>Russ</td>
</tr>
<tr>
<td>Phased</td>
<td>Brian</td>
</tr>
<tr>
<td>Spiral</td>
<td>Mark</td>
</tr>
<tr>
<td>Agile</td>
<td>Jeff &amp; David</td>
</tr>
<tr>
<td>?Michael?</td>
<td></td>
</tr>
</tbody>
</table>