Pragmatic Programmer Tip: Care about Your Craft

Why spend your life developing software unless you care about doing it well?
Course Staff

• Gail Alverson

• Yuriy Brun

• Hao Lu (TA)

• Bilge Soren (TA)
Cray Inc. – Supercomputer performance for scientific applications

3D Seismic Earthquake models – PSC
Combustion Models – SNL

Understanding climate change sometimes requires the world’s most powerful open science computer.

With more than 100 trillion calculations per second, it’s not a problem.

http://computing.ornl.gov/cesi

CSE 403, Winter 2011, Alverson
Red Storm system

- Massively parallel processing supercomputer system used for analysis and stewardship of nuclear weapons - for Sandia National Lab $93M – 10k cores to 30k cores
From RS to XT3 to XT6

- Full delivery of Red Storm was 3 ½+ years, but got something to the customer in 3. It was a sprint the whole way, and the team felt it.
- Software effort was much more complex than expected; rearchitected multiple components along the way.

Red Storm grew into a product line, the XT series. Software has vastly evolved over its lifetime, as has the Hardware.
Fastest supercomputer

Jaguar (XT4/5 system, ~250,000 cores) supercomputer claws its way to #1 (11/09-11/10)
### Making SW is hard – Pitfalls to avoid

<table>
<thead>
<tr>
<th>People</th>
<th>Process</th>
<th>Product</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Undermined motivation</td>
<td>• Overly optimistic schedules</td>
<td>• Requirements gold-plating</td>
<td>• Silver-bullet syndrome</td>
</tr>
<tr>
<td>• Weak personnel</td>
<td>• Insufficient risk management</td>
<td>• Feature creep</td>
<td>• Overestimated savings from new tools or methods</td>
</tr>
<tr>
<td>• Uncontrolled problem employees</td>
<td>• Contractor failure</td>
<td>• Developer gold-plating</td>
<td>• Switching tools in the middle of a project</td>
</tr>
<tr>
<td>• Heroics</td>
<td>• Insufficient planning</td>
<td>• Push-me, pull-me negotiation</td>
<td>• Lack of automated source-code control</td>
</tr>
<tr>
<td>• Adding people to a late software project</td>
<td>• Abandonment of planning under pressure</td>
<td>• Research-oriented development</td>
<td></td>
</tr>
<tr>
<td>• Noisy, crowded offices</td>
<td>• Wasted time during the &quot;fuzzy front end&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Friction between developers and customers</td>
<td>• Shortchanged upstream activities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Unrealistic expectations</td>
<td>• Inadequate design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Lack of effective project sponsorship</td>
<td>• Shortchanged quality assurance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Lack of stakeholder buy-in</td>
<td>• Insufficient management controls</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Lack of user input</td>
<td>• Premature or overly frequent convergence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Politics placed over substance</td>
<td>• Omitting necessary tasks from estimates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Wishful thinking</td>
<td>• Planning to catch up later</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Code-like-hell programming</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Today’s outline

- Course overview
- Assignment 1 – Product proposals

Week readings

- Readings will start next week
So what IS software engineering?

What are your thoughts?

- How to organize projects and work with a team
- Developing features based on customer need
- Evaluating the competition
- Responding to customer evolving needs in an iterative development process
- Testing and making sure the deliverable is what the customer wanted
So what IS software engineering?

Software engineering involves:

1. Processes necessary to turn a concept into a robust deliverable that can evolve over time
2. Working with limited time and resources
3. Satisfying a customer
4. Managing risk
5. Teamwork and communication
What is a software project?

Projects are a balance of three dimensions, with the goal of producing a successful deliverable.

- **Features**
- **Time**
- **Resources ($$$)**

CSE 403, Winter 2011, Alverson
A typical 403 week

1. **Class sessions** to discuss best practices
2. **Sections** to dig deeper and/or discuss pragmatics and tools; also use to meet with your team
3. **Readings and assignments** to reinforce the information
4. **Group project** to enable you to have direct experience with the material we’re covering
   - You’ll meet *technical challenges* given the larger project
   - You’ll meet *social challenges* given the team effort
Readings and summaries

Each week:
- Will have a set of required readings associated with it
- Will have a reading summary of one topic assigned

Reading summary assignments – generally:
1. Paragraph containing the main idea/main points
2. Paragraph about how this material relates to 403
3. Paragraph of your own analysis of the paper

More definition will come with first assignment.
The Project

- You make product proposals - **this Thurs/Fri**
  - And then vote on which products to “fund”
- You’re divided into project teams of 6-8 students
  - We choose the teams, to mimic the real world
  - Larger teams, larger projects, like industry
- You develop your deliverable in stages
  - Reflects modern methodologies for effective software project development
  - From requirement development through delivery
- Another team will act as your customer
  - Ultimately, a project will be successful only if it satisfies its customer
Project culture

- This is a real project
  - We expect you to work to build a real system
  - To be used by real people

- Take responsibility
  - Take initiative
  - Find and solve problems yourselves
  - Coding is only part of the job
  - Good planning and design, hitting your market, and working well with your team, are all needed for success
Project timeline

Jan 6, 2011
Project Proposals

Jan 21, 2011
System Reqs

Feb 7, 2011
Alpha Release

Feb 28, 2011
Customer Exposure

Feb 4, 2011
System Design

Feb 21, 2011
Beta Release

Mar 11, 2011
Final Release

Jan 3, 2011
Today

* Subject to change
SuiteRates project example

Add an Expense

<table>
<thead>
<tr>
<th>Name</th>
<th>Amount</th>
<th>Recurring?</th>
<th>Notify?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Costco</td>
<td>$100</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>no</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Due date: 1/2/07

Weight (%) Amount
You 10 10
Tom 5 5
Joe 85 85
Nancy 5 5
Marla 5 5
Tammy 85 85
George 50 50
Chad 85 85
Mary

post expense
Introduction

Collaboration, literally, consists of working together with one or more other people.

Although the word collaboration is widely used in many varying contexts such as education, science, art, and business, very little research has been carried out to determine the properties of this process. With the relatively recent advent of computer mediated communication (CMC), the nature of collaboration is coming under more intensive scrutiny. As software designers, facilitators and theorists from many diverse fields strive to create more useful and effective collaborative environments and methods, more light is thrown on this ubiquitous and taken-for-granted practice. However, what light is being cast is still fairly refracted into the diverse fields in which the research is being carried out. Perhaps more collaboration into the nature of collaboration will be required to answer such questions as:

- How does collaboration differ from cooperation? (note: these definitions are generally more or less equivalent.)
- What qualifies as a collaboration? Is it simply any collaboration in the same way that a work of art is when two artists collaborate face-to-face and for that matter, does a family, city, nation or species qualify?
- What are the defining principals or elements of this process?

Currently there exists no uniforming general theory of collaboration.

Etymology

Dating from 1871, collaboration is a back-formation from collaborator (1822), from the French collaborateur, ultimately from Latin collaborare, past participle of collaborare ("to work with"), itself derived from com- ("with") and laborare ("to work").

Nuances

"Collaborate" implies "to work together on a project". When individuals work together as in an academic setting, "collaborate" includes the "to be jointly accredited" for the work completed. When individuals and organizations work together, or organizations with other organizations, nuances include "usually but not necessarily willingly" and "with another organization with which one is not normally connected".

Client Browser

Client 1

Client 2

railpad | Technology Overview
Lessons from past students

- Foundation of the success of our team was communication.
- Team communication and cooperation are all-important.
- Working together (physically) was good.
- Well-run and consistently scheduled meetings help a project a lot.
- We often underestimated tasks. If we had spent more time analyzing each task and breaking it down into more manageable chunks our estimated completion times would have been more accurate.
- Get things done early; don’t cram at the end.
More lessons

- We learned (through some pain) to ensure to do small, frequent updates and commits. Failing to do this results in merges that can be a nightmare.
- Thoroughly testing your code and ensuring that your code passes all current tests before submitting is very helpful.
- Need better upfront testing design.
- Remember you can cut features (triple constraint).
- It's important not to underestimate the difficulty of learning new programming languages, frameworks and tools.
Assignment 1 - Proposals

Your chance to turn a great idea into a product!

- Constraints
  - Client/server networked architecture
  - Installable/runnable on lab machines
  - Not a game
  - Achievable by 6-7 engineers, in 8-9 weeks
Assignment 1 - Proposals

- Prepare a 3 slide, 3 min pitch in teams of 2-3
  - Vision
  - Software architecture
  - Challenges and risks

- Turn in Thurs 1/6 by noon and present on 1/6, 1/7

- Vote on Friday 1/7 by 11pm
  - Rank your choices
  - Option to choose a buddy (reciprocal)
## Back to the overview … grading

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>Proposal (3)</td>
</tr>
<tr>
<td></td>
<td>Requirements (6)</td>
</tr>
<tr>
<td></td>
<td>Software Design and Planning (10)</td>
</tr>
<tr>
<td></td>
<td>Alpha Release (6)</td>
</tr>
<tr>
<td></td>
<td>Beta Release (15)</td>
</tr>
<tr>
<td></td>
<td>Customer feedback on Beta (5)</td>
</tr>
<tr>
<td></td>
<td>Final Release (15)</td>
</tr>
<tr>
<td>15%</td>
<td>Reading summaries and related assignments</td>
</tr>
<tr>
<td>10%</td>
<td>First midterm</td>
</tr>
<tr>
<td>10%</td>
<td>Second midterm</td>
</tr>
<tr>
<td>5%</td>
<td>Class/lecture participation</td>
</tr>
</tbody>
</table>

Your scores on project work may be adjusted, based on your contribution. Peer evaluations will occur several times in the quarter. Your % will be mapped to a 4.0 scale for your final grade.
Class website

www.cs.washington.edu/education/courses/403/11wi/index.html

The “Calendar” link will be especially useful to you
Goals of 403 (WIFM)

- Be exposed to some of the best software development practices in use today

- Learn how to more effectively collaborate with others toward a common goal

- Understand how software is produced – from conception to shipping and subsequent maintenance

- Develop skills to articulate your ideas and progress

- Understand the issues and tradeoffs involved in making decisions as software engineers and project managers
Questions?