403 Projects
Spring 2006
Outline

- LCO Retrospective
- Survey results
- Selected projects
- Selected teams
LCO Retrospective: what worked well?

What should folks continue doing in their pitches?

What you saw and liked:
• Examples to give a concrete idea of a project and vision
• Pictures, diagrams
• Less text on slides focus – focus audience attention on speaker
• Not seeming over rehearsed
• Enthusiasm
What I saw and liked:
• Good motivation of the problem
• Good comparison with alternatives
• Good scoping
  • Features now, later
  • Goals, non goals
• Good diagrams (pictures say 1000 words)
  • Mock up GUI solutions
  • Software architecture
What should people think about when preparing their LCA pitches?

What you saw and thought could be improved:

• Write notes/triggers in powerpoint, not full prose
• Don’t read the slides – embellish the slides
• Emphasize key ideas
• For short presentations, choose wisely what you cover – you don’t have time to cover everything
LCO Retrospective – what could be improved?

What should people think about when preparing their LCA pitches?

What I saw and thought could be improved (a process aspect):

• Testing and documentation being left until the end
  • Early testing can drive good code
  • Early documentation can drive good interfaces
• (Erika) Integration being left until the end
  • Integrate early and often!

Pragmatic Programmer Tip: Design to Test - Start thinking about testing before you write a line of code.
So... how did the survey turn out?

- Great response – thank you!

- **Compelling concept** - all over the board, with Collaborative Dynamic Art project taking the lead

- **Compelling design** – all over the board, with leaders (in no special order): TestFiles, Railpad, Collaborative Dynamic Art, Grid computing
Before we move onto the projects...

- We had to narrow the playing field to 6, in order to get reasonable sized teams.

- All the projects presented great ideas – you’re a talented lot!

- Teams may have some original members.
  Teams may have some new members.
  All members are created equal!*

*with respect to having a voice in the project
How were the projects chosen?

- Order the projects by greatest sum(1st, 2nd, 3rd choice) preferences
  
  Select the top 6
Onto the results …

- Get a Life
- Collaborative Dynamic Art Project
- General Purpose Grid Computing
- Collaborative Imaging Grid
- Railpad
- TestFiles

CSE 403, Spring 2006
How were team members assigned?

- Painfully 😊
- Note that some 1^{st}, 2^{nd}, 3^{rd}, ok, choices made were no longer available (ARG – prefer nots)
- Satisfy all 1^{st} choices
  - Satisfy all 2^{nd} choices
  - Satisfy all 3^{rd} choices
- Handful of people left – satisfy with “ok” choice if one available
Questions?
Life Cycle Architecture

I need a description of your project and its projected cost.

That's impossible.

The project uncertainty principle says that if you understand a project, you won't know its cost, and vice versa.

You just made that up.

That doesn't make it wrong.
Readings

- “Anchoring the Software Process”, p 1-10, Barry Boehm
- “Software Project Survival Guide”, McConnell, Chapter 10, Architecture
Outline

- What is an LCA review?
- Deep dive into the LCA components
- Assignment #3, LCA review of your project
What is an LCA review?

- The culmination of the *detailed* planning and design phase of your project

- An elaboration of the LCO review - addresses the same five elements
  - More details and decisions are expected, and fewer open options
  - Changes are anticipated as you’re refining the deliverable
  - Major risks are resolved (fully or with a management plan)

- Requires approval by stakeholders to move forward
Elements of an LCA

1. Operational Concepts
   What is it – high level?

2. System Requirements
   What is it – low level?

3. System and software architecture
   How – technically?

4. Lifecycle plan
   How – resources, time?

5. Feasibility Rationale
   Are the risks managed?

SOUND FAMILIAR?
1. Operational Concepts

- Provide *precise* system objectives and scope
  - User community?
  - Environment this program works in?
  - Major benefits?
  - Establish what the system *does and does not do*

Now is the time for all the stakeholders to recognize what they are and are not getting
2. System Requirements/Interface

- Consider the interfaces from the point of view of all stakeholders
- Specify *all* features of the system
  - well defined now or can be defined later with low risk
  - capabilities, interfaces, appearance
  - include all out-of-band functions - support, admin, update
  - include performance and reliability of particular functions
  - specifics of security requirements
- Include what will NOT be provided

Techniques: use cases, prototyping, release feature lists
3. System and software architecture

An architecture bridges the gap between specifications and code

- Detail major software components, their interfaces, and how they interact, including GUI’s
- Make some decisions - you are headed for action! If compelling alternatives, argue why you made your selection
- Identify specific 3rd party software that will be used in your product (consider licensing aspects)
- Identify evolutionary paths - Where do you anticipate change or upgrades? Can you support it?

Notations: state/dataflow diagrams, sequence diagrams, class diagrams, UML, architectural views
4. Lifecycle plan

Provide a detailed project development plan
  - Define major milestones
  - Define smaller milestones
  - Fill in tasks and map resources to them
  - Define roles and responsibilities (team structure)

Tools: Microsoft Project is commonly used in industry. Excel is another good option – reasonable for 403.
5. Feasibility Rationale

- Provide assurance of consistency among the elements
- Identify top remaining risks and a risk management plan

<table>
<thead>
<tr>
<th>Risk</th>
<th>Probability of occurring</th>
<th>Impact if it occurs</th>
<th>Mitigation plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cannot license 3rd party tool</td>
<td>40%</td>
<td>9/10</td>
<td>By April 30th, have backup vendor in place</td>
</tr>
</tbody>
</table>
So, how does this translate to the assignment?

Overview presentation = Operational concept
Specification document = System requirements
Architecture document = System & SW Arch
Team structure, schedule, task assignments and risks = Lifecycle plan, feasibility rationale
Test (think about data sets!) and documentation plan = … a missing feature of the LCA…

Assignment 3 – LCA Review – Now available on the class web
Due Wed April 19th, 10pm
One final note

These are living documents!

Can you think of reasons why?