Outline

- Questions
- Scheduling informal discussions for project milestone #1 deliverables
- Homework #2
- Next milestone deliverables
- Prototyping

Questions First

- On class?
- On project?
- On homework?
- On material we’ve discussed?
- Other?

CSE403 Section 4:

Prototyping

Bonus: Common Mistakes to Avoid

Valentin Razmov, CSE403, Spr'05

How Detailed Should Designs Be?

Table 1-1: Design Formality and Level of Detail Needed

<table>
<thead>
<tr>
<th>Factor</th>
<th>Level of Detail Needed</th>
<th>Design Formality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design construction team has deep experience in application area</td>
<td>Low Detail</td>
<td>Low Formality</td>
</tr>
<tr>
<td>Design/construction team has deep experience but is inexperienced in the application area</td>
<td>Low-Medium Detail</td>
<td>Low-Medium Formality</td>
</tr>
<tr>
<td>Design/construction team has moderate to high technical competence</td>
<td>Medium Detail</td>
<td>Medium Formality</td>
</tr>
<tr>
<td>Application is complex</td>
<td>High Detail</td>
<td>High Formality</td>
</tr>
<tr>
<td>Application is complex and well tested</td>
<td>Medium Detail</td>
<td>Medium Formality</td>
</tr>
<tr>
<td>Project is small</td>
<td>Low Detail</td>
<td>Low Formality</td>
</tr>
<tr>
<td>Project is large</td>
<td>Medium Detail</td>
<td>Medium Formality</td>
</tr>
<tr>
<td>Software is expected to be released within several months or years</td>
<td>Low Detail</td>
<td>Low Formality</td>
</tr>
<tr>
<td>Software is expected to be a long time before release</td>
<td>Medium Detail</td>
<td>Medium Formality</td>
</tr>
</tbody>
</table>

Next Milestone – Preliminary Release

Deliverables:
- Application sources and binaries
- Latest spec & design documents
- Release notes
- Instructions on how to run a (small) demo of your app
- Known issues
- Automated (unit and acceptance) tests
- Up-to-date schedule

We highly recommend:
- Have a testing framework established
- Albeit with few tests present
- Start working on an installer
- It will be required in the final release

Mistakes Students in Previous SwEng Classes Have Made (1)

Scheduling and prioritizing-related:
- Not exploring all unknowns (risks) early on to create a realistic schedule
- Not maintaining an up-to-date schedule with all remaining tasks and how they map to the resources (time, people) in the team
- Not leaving enough “safety net” time before major releases in case something unexpected happens
  - It often does happen in the most inopportune moments.
Mistakes Students in Previous SwEng Classes Have Made (2)

Scheduling and prioritizing-related:
- Underestimating the challenges of a new development environment
- Overly relying on similarities to known environments
- Leaving too few resources (people) for a critical task that can’t be delayed
- Spending time on “cool” features that are not central to the needs of the users while delaying the development of promised features
  - A real project is not about what developers enjoy doing, it's about what brings value to customers.  
  - Hopefully, the two are similar, but if not, the latter should take precedence.

Mistakes Students in Previous SwEng Classes Have Made (3)

Communication-related:
- Failing to submit (for the preliminary release and even for the final release!) key required components
  - Missing documentation, tests, etc.
- Submitting code without clear instructions about how to run it if one starts from scratch
  - Most customers aren’t as tech-savvy as you are!
  - Customers aren’t intimately familiar with your project and your way of doing things

Prototyping

Why is prototyping useful?

What is it useful for?

Prototyping: Common Terminology

- Horizontal prototype
- Vertical prototype
- Throwaway prototype
- Evolutionary prototype

<table>
<thead>
<tr>
<th>Throwaway</th>
<th>Evolutionary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td></td>
</tr>
<tr>
<td>Vertical</td>
<td></td>
</tr>
</tbody>
</table>

Risks of Prototyping

The biggest risk I see with prototyping is ..............................................................

because ....................................................................................................................

Risks of Prototyping

Managing customer expectations
- “Oh, it’s almost done, so we’ll have a final version next week, won’t we?”

Suggestions:
- Be very clear what the purpose of the prototype is.
- Don’t show fancy GUIs.
- Be careful not to promise too high performance.
One-minute Feedback

- What one or two ideas discussed today captured your attention and thinking the most?

- List any ideas/concepts that you would like to hear more about. Be specific.