


Outline

- n (Mis)communicating expectations
- n Schedule of class-related deliverables
- n Specific final release deliverables
- n Your questions


- n Intellectual activity: anticipating changes, redesigning
- n How do you know if your a product is "good enough"?

- n Handout: "When You're Tested" by Jon Bach




Schedule of Class-Related Deliverables

- n **Sun, May 29 @ 10pm:** essay #2 responses due
- n **Wed, Jun 01 @ noon:** final release, presentation, and design critique due
- n **Wed, Jun 01 @ 1:30pm-2:30pm:** in-class demos and presentations by Teams E, D, C
- n **Wed, Jun 01 @ 10pm:** peer reviews #3 due
- n **Fri, Jun 03 @ 1:30pm-2:30pm:** in-class demos and presentations by Teams B, A
- n **Fri, Jun 03 @ 10pm:** essay #3 responses & usefulness of peer reviews #3 due
- n **Mon, Jun 06 @ 2:30pm-4:30pm:** in-class final exam
- n **Mon, Jun 06 @ 10pm:** final questionnaire due




Next Milestone – Final Release Deliverables

- n **Installation packages**
 - That include all of the items below
- n **Application sources and binaries**
 - Separate distributions (installation packages) for customers and developers
 - One-step build – from compiling all sources to creating installation packages
- n **User & technical documentation (separate)**
 - User doc: What does my mom need to know (and do) in order to run this product?
 - Technical doc: What does a support team need to know in order to work on ver2?
- n **Release notes**
 - Known issues with associated severities & priorities
 - Include a link to your bug tracking system's tasks/tickets that reflect those issues
 - Specify where your current CVS repository is
 - Instructions on running the installer and your app are now part of the user doc.
- n **Latest test plan**
- n **Automated unit and acceptance tests**
 - Test coverage?
- n **Up-to-date schedule**
 - Things that have been accomplished (of those that were planned)
 - Features (of those initially planned) that are now pushed to ver2 or abandoned
 - How much would each such feature cost (in terms of dev effort)?



Your Questions

- n On class?
- n On project?
- n On homework?
- n On material we've discussed?
- n Other?




CSE403
Section 9:

**Intellectual Activity:
Anticipating Changes; Redesigning**

How Do You Know If Your Product Is "Good Enough"?

Valentin Razmov, CSE403, Sp'05



Anticipating Design Changes

- n What are your biggest risks now?

- n What types of changes of requirements are you anticipating (in your designs) and, conversely, what kinds of changes would require significant redesign?

Anticipating Design Changes (cont.)

- What specific changes to your design would be needed to sell Gizmoball internationally?
- How can you (proactively) anticipate such changes early on in your designs?

Lessons from Redesigning a System

- Case: the peer review system losing its anonymity
- Lessons I was reminded after fixing it (that I should have known)
 - Listen carefully to your customers!
 - Don't assume what existing library functions do – verify it!
 - Do *regression tests* even if the added feature doesn't seem to have anything to do with the tested functionality!
 - Human-readable intermediate formats help
 - One selling point of XML, as compared to binary formats
 - If you're not sure how long it'll take to fix, assume 3x as long
 - "Good judgement comes from experience. Experience comes from bad judgement."

How Do You Determine if Your Product is "Good Enough"?

Criteria for "Good Enough"

(James Bach, <http://www.satisfice.com/articles.shtml>)

- It has sufficient benefits.
- It has no critical problems.
- The benefits sufficiently outweigh the problems.
- In the present situation, and all things considered, further improvement would be more harmful than helpful.
- Key questions to ask when doing an evaluation:
 - Good enough for whom?
 - Good enough for what?

Is GizmoBall "Good Enough"?

- Does it have sufficient benefits?
- Does it have no critical problems?
- Do the benefits sufficiently outweigh the problems?
- In the present situation, and all things considered, would further improvement be more harmful than helpful?
- So, is your product "good enough" for the final release?

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.