Outline

- Feedback on informal team discussions after milestone #2
- Your questions
- Clarifications on Test Plan deliverables
- Intellectual activity: the Joel test
- Intellectual activity: design changes, anticipating changes

Your Questions

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

Next Milestone – Test Plan

Deliverables:

- Test plan
  - A deliverable to your team, not to the customer
    - Make it useful for you; I’ll simply look to see that you’ve done the work
  - Describes what you want to test and how you will do it
    - Must match the tests you will create (are creating)
  - Follows a disciplined methodology (e.g., SFDPO)
    - Some types of tests may not fit nicely, so invent your own categories

Issues to consider:

- Do you have tests in each of the categories?
- Do your proposed tests cover all use cases?
- Which tests are critical / high priority?

Test Planning using the SFDPO Methodology

From http://www.satisfice.com/articles/sfdpo.htm

- **Structure** (what the product is):
  - What files does it have? Do I know anything about how it was built? Is it one program or many? What physical material comes with it? Can I test it module by module?
- **Function** (what the product does):
  - What are its functions? What kind of error handling does it do? What kind of user interface does it have? Does it do anything that is not visible to the user? How does it interface with the operating system?
- **Data** (what it processes):
  - What kinds of input does it process? What does its output look like? What kinds of modes or states can it be in? Does it come packaged with preset data? Is any of its input sensitive to timing or sequencing?
- **Platform** (what it depends upon):
  - What operating systems does it run on? Does the environment have to be configured in any special way? Does it depend on third-party components?
- **Operations** (how it will be used):
  - Who will use it? Where and how will they use it? What will they use it for? Are there certain things that users are more likely to do? Is there user data we could get to help make the tests more realistic?

CSE403 Section 8:

Intellectual Activity:

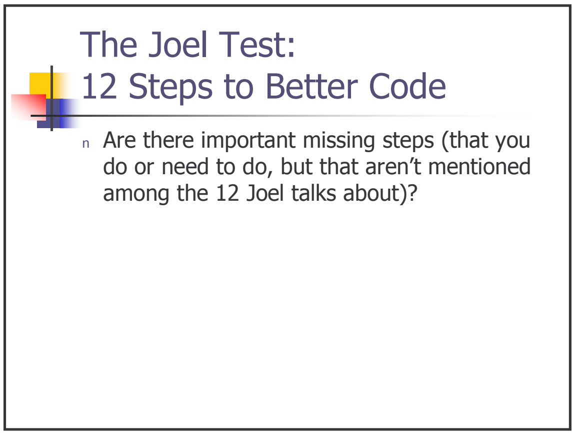
**The Joel Test;
Anticipating Design Changes,
Redesigning**

Valentin Razmov, CSE403, Sp'05

The Joel Test: 12 Steps to Better Code

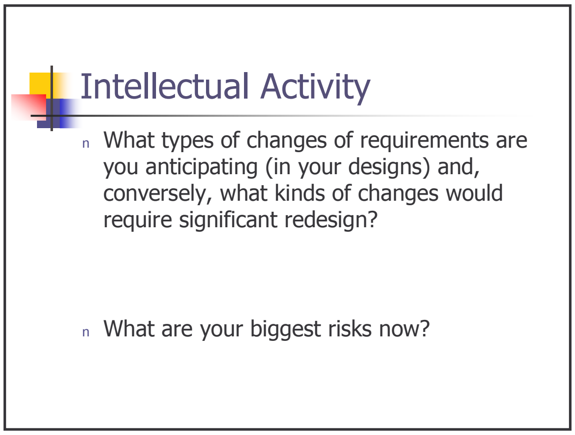
- Which of the 12 steps is your team following?

Do you use source control? ....
Can you make a build in one step? ....
Do you make daily builds? ....
Do you have a bug database? ....
Do you fix bugs before writing new code? ....
Do you have an up-to-date schedule? ....
Do you have a spec? ....
Do programmers have quiet working conditions? ....
Do you use the best tools money can buy? ....
Do you have testers? ....
Do new candidates write code during their interview? ....
Do you do hallway usability testing? ....



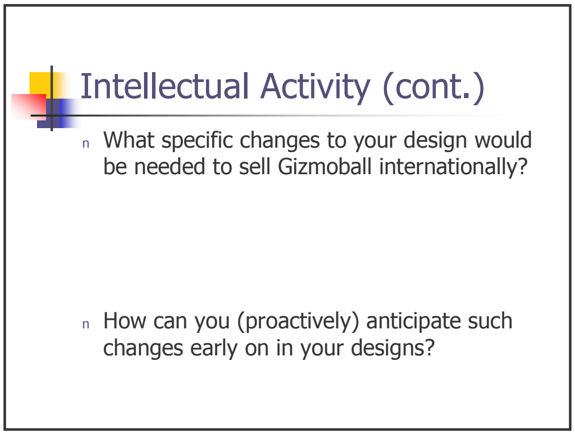
The Joel Test: 12 Steps to Better Code

- Are there important missing steps (that you do or need to do, but that aren't mentioned among the 12 Joel talks about)?



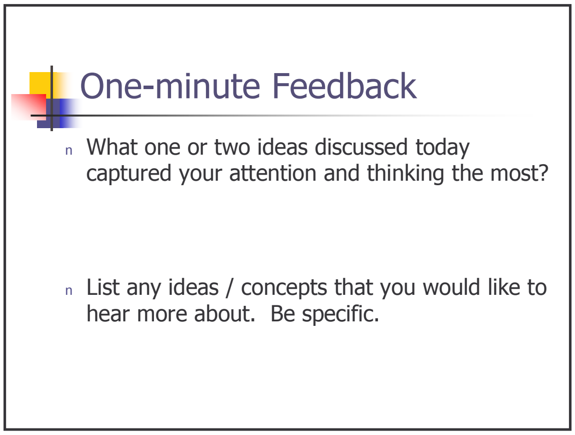
Intellectual Activity

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



Intellectual Activity (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?



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.