The Joel Test

CSE 403, Spring 2006
Software Engineering
Readings and References

• Handout
  – The Joel Test: 12 Steps to Better Code, Joel Spolsky
    http://www.joelonsoftware.com/printerFriendly/articles/fog0000000043.html
Beta Demo Reminder

• Consider how you will do your beta demos next week
  – What machinery do you need?
    • Your own computer or Prof. Alverson's?
  – Do you need network access?
  – Can you transition quickly after the previous group?
The Joel Test

- 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?
Disclaimer (Spolsky)

- These are not the only factors:
  - A great team will not help if you are working on a product no one wants
  - An incredibly talented team might produce an incredible product without these guidelines
- All else being equal, these factors indicate a disciplined team that can consistently deliver.
Source Control

Why have source control?
Source Control

Why have source control?

- Keep project in consistent state
- Track changes and enable roll-back
- Manage multiple versions
- Save data in case of disaster (disk crashing, stolen machine, etc.)
- Authoratative source for daily build
Source Control

- CVS and Subversion are free source control systems
- *For the assignment: Give us a source control dump.*
  - ('cvs log' if you are using CVS)
Source Control

What else can be managed?

- Requirements
- Design documents
- Test cases
- User documentation
- Scripts
One step build

- There should be a single script that
  - does a full checkout from scratch
  - rebuilds every line of code
  - makes the binary executable files in all version, language and #ifdef combinations
  - creates the installation package
  - creates the final media - CDROM, web site, …

- All steps are automatic and exercised regularly

- **For the assignment: Give us your build script and an output from a run of that script**
Daily Build and Smoke Test

- Build the entire product every day and run a good test suite against the new version
- See the April 13th quiz section for more details
- For the assignment: Give us a list of build versions
Use a bug database

- You cannot keep a the bug list in your head!
  - Especially with multiple developers
- When you find a bug you need to know
  - how to reproduce the bug
  - expected behavior, actual behavior
  - current owner of the bug
  - status (open, fixed)
- You should be keeping your bugs in your Bugzilla database
- For the assignment: Give us a log of bugs identified, open and resolved
Fix bugs before writing new code

Why not fix the bugs later?
Fix bugs before writing new code

Why not fix the bugs later?

- Harder to find later
- More may depend on the buggy code
- Bug may reveal fundamental problems
- Leaving all bugs to the end will make it harder to keep a good schedule
Up to date schedule

Why keep your schedule up to date?
Up to date schedule

Why keep your schedule up to date?

• Keeps expectations realistic
  – for the team, customers and business people
• Helps prevent feature creep
• What was the point of making it in the first place if you do not keep it up to date?
Have a good specification

- You all have this!
  - Now you just have to keep it up to date
- Always reevaluate
  - Are the scenarios realistic?
  - Can the goals be accomplished?
Quiet working conditions

- Minimize uncontrollable distractions
  - turn off email and IM
- Be focussed when you are working alone
- Be focussed when you are meeting
  - make good use of the time you are together, you do not have much of it!
Use the best tools money can buy

- Does not mean the most expensive tools!
- Understand
  - the tools you need
  - the tools you already have
  - what you need to be more productive
Use testers as basic part of the team

Why have separate testers?
Why have separate testers?

- Testing requires a different mind set
- Developers know their code too well
- Customers will be happier if you find the problems before they do
Have candidates write code

• Not so much for you projects, but...
• Welcome chances to write code when you interview
  – Shows you can think about problems different abstraction levels
  – Shows you know how to use basic CS tools
• My advice: Show that you can turn ideas into data structures and algorithms
  – Interview coding is not about syntax or specific function names
  – If it is, do you really want that job?
Hallway Usability Testing

Why do informal user testing?
Hallway Usability Testing

Why do informal user testing?

- Reality check
- Finds the big problems
- Quicker and cheaper than formal usability testing
- General feedback
  - Do users understand the product?
  - Do they like it?
  - What do they do that you never thought of?