Lecture 10: Incremental Releases

Valentin Razmov

Outline

- Why Incremental Releases?
- Deliverables for Zero-Feature, Alpha/Beta, and Final Releases
- "The Joel Test: 12 Steps to Better Code"

Resources

- *Rapid Development*, by Steve McConnell
- *The Joel Test: 12 Steps to Better Code*, by Joel Spolsky, http://www.joelonsoftware.com/printerFriendly/articles/fog0000000043.html
- Lectures from Winter 2004, by David Socha
- Discussions with James Bullock

Why Incremental Releases?

- To reduce the risks
  - Manage customer expectations
  - Have something shippable / shipped at all times
  - Allow the development team to quickly identify and correct/recover from early failures
- Closely related to evolutionary prototyping, evolutionary delivery, staged delivery, and design-to-schedule

Deliverables: Zero-Feature Release

- **Build process, installation process, code repository, automated testing framework, bug tracking system**
  - Maybe no tests yet and no tickets in the bug tracking system
- **Installation package**
  - Includes all of the items below
- **Demo of one-step build and component communication**
  - Checks out all sources from repository, compiles and builds binaries, packages them along with all existing documentation and automated tests, and places the result on a known site ready for downloading
  - Shows that your main components identified in the design can successfully communicate (be integrated) with each other
- **Latest specification, design, and test plan documents**
  - Keep them short! Consider what is / isn't important for customers / devs.
- **Up-to-date schedule**
  - Includes what has been done and what remains to be done
- **Release notes**
  - Detailed instructions on how to run the demo
  - Known issues with prioritization

Deliverables: Alpha/Beta Releases

- **Installation package**
- **Application sources and binaries**
  - One-step build for all sources
- **Latest spec & design documents**
  - Keep it short! Consider what is/isn't important for customers/devs.
- **Release notes**
  - Detailed instructions on how to run a (small) demo of your app
  - Known issues with prioritization, expressed in a bug tracking system
- **Up-to-date test plan**
- **Automated tests (unit and acceptance)**
- **Up-to-date schedule**
  - Including what has been done and what remains to be done

Issues to consider:

- Who is your audience – customers or developers? What do they expect from this release? What defines success for them?

Deliverables: Final Release

- **Installation packages**
  - That include all of the items below
- **Application sources and binaries**
  - Separate distributions (installation packages) for customers and developers
  - One-step build – from compiling all sources to creating installation packages
- **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?
- **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.
- **Latest test plan**
- **Automated tests (unit and acceptance)**
  - Test coverage would be a very welcome addition.
- **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)?

The Joel Test: 12 Steps to Better Code

- Which of the 12 steps will your team follow?

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)?