Coming together is a beginning. Keeping together is progress. Working together is success.

Henry Ford
Timeline

- Apr 3, 2008: Project Proposals
- Apr 18, 2008: System Req
- Apr 30, 2008: Zero Feature
- May 19, 2008: Customer Exposure
- Jun 5, 2008: Final Release
- Apr 28, 2008: System Design
- May 12, 2008: Beta Release
- Jun 4, 2008: Today
Timeline up close

1. Final release demos in class Thurs and Fri

   *Feel free to send your project url to the class!*

2. Final Team Member Contribution Assessment
due Fri by midnight

3. Final exam Monday June 9th @ 8:30am
   in this room
The Final Exam – June 9, 8:30am

- Final will be in class (BAG260), open notes, open book (no laptop, cellphones, or similar electronic devices)

- Material will be based on lectures, readings, (see 403 Calendar), guest speakers, and experiences from the project

- Questions will largely be hands-on, small to medium answer, for example:
  - create a software design for a customer idea xyz
  - recognize or suggest a technique (pattern, heuristic, ...) for a problem
  - pinpoint an error in a design, a UML diagram, a use case, a UI, a ...
  - identify the best IP strategy to use for situation abc

- Practice questions – see old 403 finals and midterms on the web, jeopardy game
Whirlwind tour through 403

- Software development lifecycles
  - Jessan, Google lifecycle
  - Mark, Microsoft lifecycle, lessons
  - Wayne, Startup lifecycle
  - Cathedral, Bazaar
- Requirements/specifications (Use cases, feature lists, good/bad UI, customers)
- Architecture (Views, UML class/sequence)
- Working in teams, team structures
- Design patterns, principles, heuristics
- Quality, testing, bug tools/reports
  - John, Microsoft test lead
- Refactoring
  - Valentin, Microsoft PM
- Estimation
- Intellectual Property
Intellectual Property
Common forms of IP

- Patents
- Copyrights
- Trade secrets
- Trademarks
- Contracts/Licenses
Contracts/Licenses

- You can make anything work if you have agreement by all parties involved

- Protections, exclusions, requirements, terms, and costs must all be *explicitly* defined as part of the contract

- Examples:
  - License agreements
  - Vendor agreements
  - Non-disclosure agreements
  - Employee contracts
Free Software Foundation

The Free Software Foundation (FSF), established in 1985, is dedicated to promoting computer users' rights to use, study, copy, modify, and redistribute computer programs.

FSF promotes the freedom to

1. run the program, for any purpose
2. study how the program works, and adapt it to your needs
3. redistribute copies so you can help your neighbor
4. improve the program, and release your improvements to the public, so that the whole community benefits
Software licensed under the GPL is open source that must be made available with the product release.

GPL requires that all code that is “affected” by GPL code must also be distributed under the GPL. “Affected” is loosely equated to, part of the same functional unit.

poisonous license, copyleft license
LGPL

- LGPL is referred to as the “Lesser” GPL or “Library” GPL
- Applications may link to LGPL code without having to take on the LGPL license
There are lots of other licenses

- Mozilla Public License (bugzilla) – type of copyleft
- MIT License (ruby on rails) - permissive FOSS
- BSD License (Trac, PostgreSQL) - permissive FOSS
- MySQL – GPL or MySQL commercial license ($)
Microsoft takes on the free world

Microsoft alleges that FOSS infringes on 235 of its patents. It wants royalties from distributors and users.

- the Linux kernel violates 42 Microsoft patents
- the Linux user interface, design elements infringe on 65 patents
- OpenOffice.org infringes on 45
- another 83 are infringed on in other FOSS programs

"What's fair is fair," Ballmer told *Fortune*. "We live in a world where we honor, and support the honoring of, intellectual property."


CSE 403, Spring 2008, Alverson
And the FOSS response is?

- FOSS legal strategist, is “uncowed”
  - The action is in tight qualitative analysis of individual situations
  - Patents can be invalidated on numerous grounds
  - Others can be invented around
  - Supreme Court stated in April that patents have been issued too readily for the past two decades and lots are probably invalid (!)
  - Corporate patrons and allies

Introducing, GPLv3!
What’s your opinion?

“Patents and the open-source movement get along awkwardly at best. Patent law gives proprietary, exclusive rights to patent holders, but open-source programming is built on the idea of free sharing.”

Is FOSS (Free and Open Source Software) the best way to foster the advancement of science?

Isn’t it reasonable for a company to patent technology to attain/keep competitive advantage?
The Cathedral and the Bazaar

Your thoughts
Traits of “bazaar” style of development

- Rapid release cycle
- Involve customers early
- Treat customers as co-developers
- No centralized direction?
- Many eyes on source
- Collaborative?
Pro bazaar  Or con?

• ESR seems to have hit the nail on the head...open source projects are becoming more and more relevant and companies are moving more to the bazaar style of development.

• The bazaar style sounds awesome. However, although open source projects have become fairly common, victory over cathedrals is clearly not imminent. For example, no successful game has been developed using this bazaar tactic.
Opinions on quality?

- I’m skeptical that the bazaar model can consistently produce results as good as Raymond suggests....

- I don’t believe the amount of quality time put into bazaar style apps could ever compare to that of cathedral style. Sure, you can get more eyes on the product with early public releases, but what quality can you expect of hackers coding in their free time?
Opinions on economics?

If the bazaar style is superior, what happens to the for-profit world of Google and Microsoft?

If feels like an idealistic dream to believe that all software will be written collaboratively and altruistically by balling bazaars. How do all these open source developers support themselves without getting paid? How will you and I survive?
Contrast

• Mark Zbikowski – Microsoft system development
• Wayne Yamamoto – Startup Web development
Where is this important?

- Honest estimates
- Quality, quality, quality
- Have a customer and understand your customer, not just what they say, but what they really want
- Short release cycles
- Solid shippable product
- Don’t corrupt data
Where is this important?

- Getting something out early, even if it’s has flaws
- A good regression suite
- Face to face communication
- Good choice of tools
- Developers must take on a number of roles
- Constant refactoring