Lecture 24: Intellectual Property Issues (Part II)

Outline

- Why Intellectual Property (IP) Protection?
- Different Types of IP Protection
  - Patents
  - Copyrights
  - Trade secrets
  - Trademarks
  - Licenses and Contracts
- Discussion questions

Resources

- Lecture from csep590tu "Information Technology and Public Policy" (autumn 2004: 09/30)
  - http://www.cs.washington.edu/education/courses/csep590/04au/lectures/
- Lectures from cse590so "Society and Technology" seminar (spring 2005)
- SBE workshop as part of the UW Business Plan Competition program (winter 2004)

Motivation behind Intellectual Property Protection (reminder)

- **What:** Protecting intangible assets
- **Why:** To foster creativity and encourage (technological) progress
- **How:** By providing temporary monopoly as an incentive for creators to do intellectual work for a living
  - Must be balanced against need to not stifle (shut out) competition completely and for all times

Patents (reminder)

- Protect: inventions (processes, machines, products, models, improvements, etc.)
  - Protected against: others making, using, selling invention, *even if they independently came up with the same invention*
- Excluded: natural laws and phenomena, abstract ideas
- Requirements: novel, useful, non-obvious; must file patent application (generally) before public disclosure
- Term: 20 years from filing
- Cost: relatively high, in both time and money
- Problems:
  - Patent officers are paid by number of issued patents.
  - Full disclosure is not enforced.
  - Overreaching patents effectively lead to monopolies.
  - Gradual expansion of what is patentable

Copyrights

- Protect: expression of ideas, *not* the ideas themselves
  - Protected against: reproduction, copy distribution, derivative work creation (*not independent creation* of the same or similar work), public performance and/or display
- Excluded: facts, data
- Requirements: original work, fixed in tangible form
- Term: author's life + 70 years
- Cost: simple, no registration
- Problems:
  - Laws subject to change under pressure from industry.
    - E.g.: Term length, cost, definition of "fair use" (reverse engineering), "first sale" doctrine, etc.

Trade Secrets

- Protect: "... any formula, pattern, design, device, or compilation of information that … gives [a business] an advantage over competitors who do not know or use it."
  - Protected against: misappropriation
- Excluded: general knowledge, skill, or experience
- Requirements: info not generally known or available, derives economic value from secrecy, must spend reasonable effort to maintain secrecy
- Term: no predefined limit
- Cost: no registration or examination
- Problems:
  - Once lost, the secret can't be regained.

Trademarks

- Protect: "any word, name, symbol, or device, or any combination thereof" used to distinguish certain goods from others
  - Protected against: others using the mark, likelihood of confusion and dilution
- Excluded: use in other industries / geographic areas
- Requirements: use the mark in commerce or register with intent to use in future, must maintain quality control over goods
- Term: 10 year renewable (no upper limit)
- Cost: ?

Contracts

- Protections, exclusions, requirements, terms, and costs must all be *explicitly* defined as part of the contract.
- Examples:
  - License agreements
    - For software, standard agreements are GPL, BSD, etc.
    - For media, Creative Commons is emerging as an alternative to the default 'All Rights Reserved'.
  - Non-disclosure agreements
  - Employee contracts
    - Including non-competition agreements, pay compensation, etc.
  - Ownership allocations

Example: IP Issues with the Use of Third-Party Software

Important questions to explore early:

- Do you have the right to use the third-party software?
- Is it important to protect some IP you are adding?
- Do the IP rights of the third-party software allow you to do this?

Example: Seemingly Easy Questions…

What do you think?

- Who owns the idea that your team has been developing?
- Is posting an email message from someone else a violation of copyright?

"The Devil Is in the Details"

Advice: Know and understand the basics but consult with a lawyer for the details.

Lessons from the History of Software Development (Part II)

Outline

- Is Software Different?
- Trends from the History of Software Development
  - Sophistication of skills (of developers and users)
  - Propagation of good development practices

Next time:

- Size of projects and products
- Criticality of getting it right

References

- "Professional Software Development", by Steve McConnell
- "Crossing the Chasm", by Geoffrey Moore

Is Software Different? (from Other Engineering Disciplines)

Arguments in favor:

- Testing the quality of software is harder
  - The Halting Problem presents a fundamental limitation in the extent to which software quality can be evaluated
    - Most properties of software (that we care about) are unverifiable
  - Unlike bridges and buildings where everything can be tested using known procedures
- Much higher rate of failure
  - May also have to do with the immaturity of the discipline
- Lower barrier to entry
- Customers have a greater role
- Customer expectations: for quality, delivery timeline, etc.
- Frantic rate of technological change
- Software is easier to copy

Is Software Different? (from Other Engineering Disciplines)

Arguments against:

- Software isn't "soft".
  - Contrary to popular perception, change cannot be "*easily* accommodated"
    - Yet requirements do change.
  - In reality, even though change is *possible* in principle, accommodating change often forces a rewriting of major parts of the software.
- Software developers still need to plan, execute, test, and sell their products. Same lifecycle.
- The discipline is still in its infancy.

Is Software Different? (from Other Engineering Disciplines)

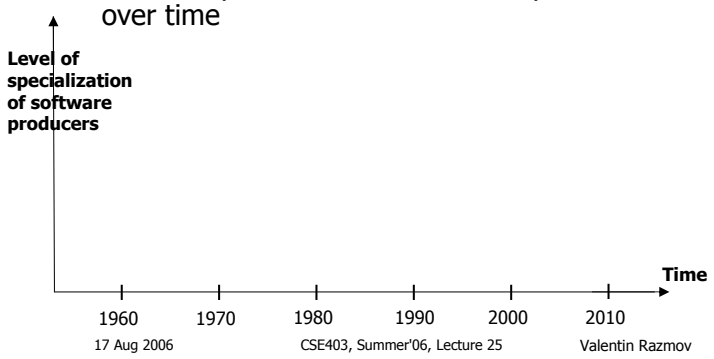
More questions to consider:

- Is software less reliable?
- Does it break differently?
- Is the environment of use of software different?
- Is the culture of software development different?
- and more…

Lessons from the History: Software Producers' Skills

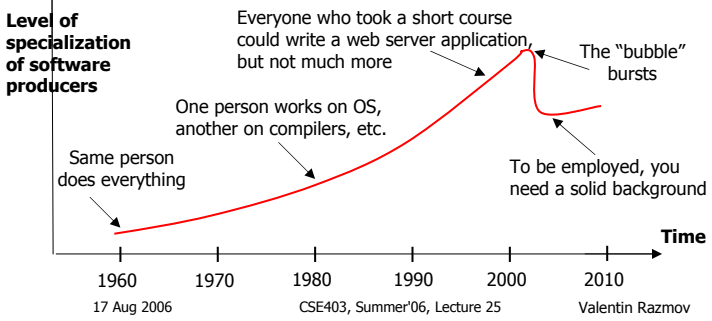
Student Submission

- Level of specialization of software producers over time



Lessons from the History: Software Producers' Skills

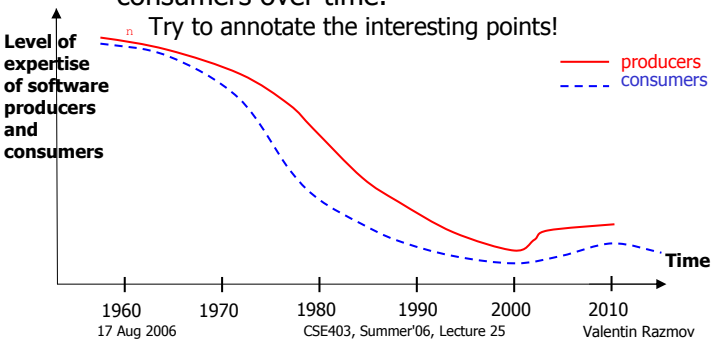
- Level of specialization of software producers over time



Lessons from the History: Sophistication of Stakeholders

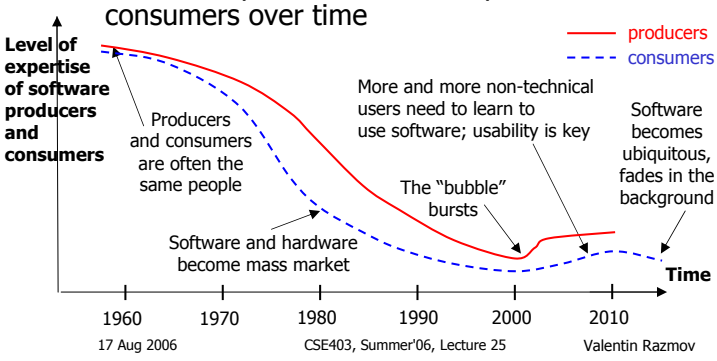
Student Submission

- Level of expertise of software producers and consumers over time.
  - Try to annotate the interesting points!



Lessons from the History: Sophistication of Stakeholders

- Level of expertise of software producers and consumers over time

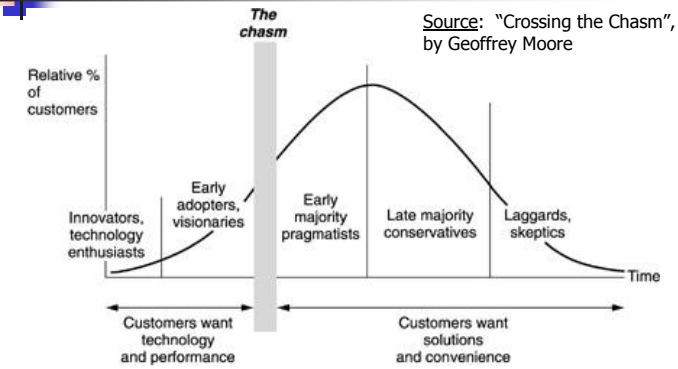


Lessons from the History: Software 'Gold Rushes' (reminder)

- The software 'Gold Rush' fever periods
  - Goal: being first-to-market in an unclaimed segment
  - Typical environment: two guys in a garage
  - High-risk projects, potentially high pay-off
  - Code-and-fix development, very informal processes
  - Customers are tech savvy, willing to forgive bugs
- The in-between (post-'Gold Rush') periods
  - Goal: sustained, productive competition with others
  - Typical environment: larger teams, formal processes
  - Lower-risk, likely lower but more predictable pay-off
  - Careful, quality-driven development with an emphasis on quality (reliability, interoperability, usability, etc.)
  - Different customer base: demands reliability

Lessons from the History: Propagation of Good Practices

Source: "Crossing the Chasm", by Geoffrey Moore



Student Submission

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 know more about. Be specific.