

CSE 403

Lecture 23

Intellectual Property Issues in Software

Reading:

What Intellectual Property Law Should Learn from Software,
by J. Boyle

slides created by Marty Stepp, Gail Alverson, David Notkin

<http://www.cs.washington.edu/403/>

Intellectual property

- **intellectual property** (IP): Any intangible asset that consists of human knowledge and ideas.
 - patents, copyrights, trade secrets, trademarks, contracts
 - *intent*: to encourage ingenuity and protect creative ventures
 - "value" of an intellectual asset:
 - Most such assets cannot be recognized on a balance sheet when internally generated, since it is very difficult to objectively value intellectual property assets.
 - They can be included in a balance sheet if acquired, which allows more accurate valuation for the asset (the acquisition cost).

IP at Cray

- Internal Cray IP
 - Hardware management patents
 - Proprietary cluster management software
 - Apprentice2 performance tool
- External IP, with fees
 - Catamount OS, Sandia National Lab
 - C, Fortran compiler, PGI
 - Totalview parallel debugger, Etnus
 - PBS Pro batch management tool, Veritas
- External IP, without fees
 - Linux, PAPI, gcc, gdb



source: Gail Alverson

Patent

- *Protects:* inventions and innovations (processes, machines, products, phrases, algorithms...)
- *Protects against:* others making, using, selling innovation, even if they independently came up with it
- *Requirements:* novel, useful, non-obvious
- *Term:* 20 years from filing; typically must file within a year of being publicly disclosed
- *Cost:* relatively high, in time and cost



USPTO on patents



"A patent for an invention is the grant of a property right to the inventor, issued by the United States Patent and Trademark Office. ... The right conferred by the patent grant is ... 'the right to exclude others from making, using, offering for sale, or selling' the invention in the United States or 'importing' the invention into the United States."

"Utility patents may be granted to anyone who invents or discovers any new and useful process, machine, article of manufacture, or composition of matter, or any new and useful improvement thereof."

Patent "war chests"

- Companies build up collections of patents to avoid lawsuits
 - Apple's iPhone patents (e.g. vs. Samsung)
 - Google buys Motorola, largely for mobile phone patents
 - Microsoft vs. Google (Android); Microsoft / SCO vs. Linux
 - Kodak vs. digital camera companies
 - Verizon vs. Vonage (VoIP technology)
- Results of patent lawsuits:
 - may need to settle (e.g. cash payout)
 - may need to license the patents (Google pays MS \$5-10 for every Android device sold)
 - suit may be thrown out for lack of merit (SCO vs. Linux)

Patent trolls

- **patent troll:** An entity granted a patent that uses that patent largely to hold large corporations "hostage" by legal threats
 - currently account for 30% of all patent litigation
- MercExchange company holds patent on "Buy it Now" sales feature, also used by eBay
 - MercExchange sued eBay for patent infringement
- Supreme Court pushed back to lower court, but until decision, allows eBay to continue using the phrase
- Other trolls: Intellectual Ventures (\$5b!), Polaris IP (email apps), WebEx (wifi), Eolas (browser plugins), Amazon (1-click)



Patent lawsuit 1

- Court ruled that Microsoft infringed on 15 patents owned by Alcatel-Lucent relating the MP3 music format (MP3 playback) in Windows Media Player

Microsoft



Alcatel-Lucent

- Alcatel-Lucent had argued it co-developed the technology with Germany's Fraunhofer Institute (FI)
 - MS licensed through FI (for \$16M) but not also Alcatel-Lucent
- Damages: \$1.52 BILLION (Microsoft appeals)
- Bad news for Apple?
 - iTunes to iPod have dependency on MP3 technology

Patent lawsuit 2

- UW ugrad (!) has 4 patents on technology used by cell phone manufacturers in "Bluetooth" technology
 - Bluetooth is used in wireless devices; exchange info w/o wires
- Washington Research Foundation (WRF) working on behalf of UW reached licensing agreement with Broadcom
- CSR (cell phone maker) did not want to license the technology
 - WRF filed a suite against CSR for patent infringement
- April 2007, CSR settled with WRF for \$15M
 - CSR believes the suit was without merit



Other patent headlines

- Kim Dotcom wants money from Google, Twitter for 2-Factor authentication
- Microsoft's most profitable mobile operating system: Android
 - (MS collects license fees on Android devices from its patents)
- Google acquires Motorola Mobility (largely for their patents)
- Patent war looming over Google VP8, WebM video codecs
- Apple v. Samsung, Round 2
- Will patents threaten 3-D printing?
- New Sony patent blocks second-hand games on PS4 system

Trademarks

- *Protects:* any word, name, symbol, or device, or any combination thereof used to distinguish goods from others
- *Protects against:* others using the mark, likelihood of confusion and dilution
- *Excluded:* use in other industries / geographic areas
- *Requirements:* use mark in commerce or register with intent to do so in future; must maintain quality control over goods
- *Term:* 10 year renewable (no upper limit)
- Is your project's name trademarked?

<http://tess2.uspto.gov/>

USPTO on trademarks



"A trademark is a word, name, symbol, or device that is used in trade with goods to indicate the source of the goods and to distinguish them from the goods of others. A servicemark is the same as a trademark except that it identifies and distinguishes the source of a service rather than a product."

"Trademark rights may be used to prevent others from using a confusingly similar mark, but not to prevent others from making the same goods or from selling the same goods or services under a clearly different mark."

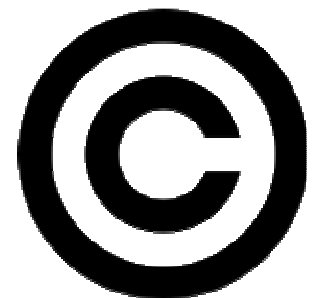
Trademark suit example

- Cisco owns the trademark "iPhone"
 - Apple started negotiations with Cisco to share the TM
 - Cisco wanted an open approach to allow Apple iPhone to be compatible with other companies' products
- Apple announced its iPhone before an agreement was made
 - Cisco sued Apple for trademark infringement
- Cisco and Apple reached a settlement (2/21/07)
 - two companies pledged to "explore opportunities for interoperability in the areas of security, and consumer and communications."
 - Other terms were confidential



Copyright

- *Protects:* expression of ideas in a medium, but not the ideas
- *Protects against:* reproduction, copy distribution, derivative work creation (NOT independent creation of same/similar work)
- *Requirements:* original work, fixed in tangible form
- *Term:* author's life + 70 years
- *Cost:* simple, no registration



USPTO on copyright



"Copyright is a form of protection provided to the authors of 'original works of authorship' including literary, dramatic, musical, artistic, and other intellectual works, published and unpublished. The 1976 Copyright Act generally gives the owner of copyright the exclusive right to reproduce the copyrighted work, to prepare derivative works, to distribute copies ..."

"The copyright protects the form of expression rather than the subject matter of the writing. For example, a description of a machine could be copyrighted, but this would only prevent others from copying the description; it would not prevent others from writing a description of their own or from making and using the machine. Copyrights are registered by the Copyright Office of the Library of Congress."

Asserting copyright

- Almost all things are copyrighted the moment they are written.
 - Can provide an explicit copyright notice or license in the work
 - e.g. email signature: "The information in this email and subsequent attachments may contain confidential information that is intended solely for the attention and use of the named addressee(s). This message or any part thereof must not be disclosed, copied, distributed or retained by any person without authorization from the sender."
 - e.g. license notice/link in a software product or source code
 - e.g. a copyright watermark logo on an image
- Copyrighting software:
 - contracts/licenses can give the user permission to use copyrighted material in various ways



Proprietary licenses

- most proprietary (\$) software is *licensed*, not sold
 - publisher grants you a "license" to use a copy of the software
 - ownership of those copies stays with the software publisher
 - reserves for the owner almost all rights
 - grants only a limited set of rights to the end-user
- Without accepting the license, the end-user may generally not use the software at all
- Are EULAs enforceable?
 - never fully challenged in court (without a settlement)



Open source / free licenses

- generally consistent with the GNU GPL and similar licenses
- owner retains ownership
- license grants all rights except the right to sell or license it on different terms ("copyleft")
 - Commercial licensing for a profit is also prohibited
- *intent*: encourage perpetuation of free use of software
- end-user may use the software without accepting the license
 - only when additional rights are desired must the end-user accept and be bound to the license terms



Fair use

- **fair use:** limitation and exception to copyright to allow duplication and usage for purposes of commentary, criticism, research, education, archiving, scholarship, etc.
 - examples: Tivo, VHS, CD ripping/burning, Xeroxing
- USPTO: "The fair use of a copyrighted work ... is not an infringement of copyright. In determining whether the use made of a work in any particular case is a fair use, the factors to be considered shall include:
 - purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;
 - nature of the copyrighted work;
 - amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
 - effect of use upon the market for or value of the copyrighted work."



Reverse engineering

- **reverse engineer:** take something apart to see how it works
 - In software, this usually means decompilation (converting from object code to source code)
- As reverse engineering is based on a copy of a piece of software, it is almost always in violation of copyright unless the right to do so is explicitly granted
 - However, if decompilation is needed to attain interoperability, US and European copyright laws permit it in some cases
- One US example allowed a company to decompile to get around a software locking mechanism for a Sega game console

Trade secret

- *Protects:* concept, idea, info, or innovation
- *Protects against:* misappropriation (NOT independent creation of same work)
- *Requirements:*
 - info not generally known or available
 - must spend reasonable effort to maintain secrecy
 - company derives some (economic) value from secrecy
- *Term:* no predefined limit
- *Cost:* no registration or examination

Trade secret examples

- Symantec said Veritas shared trade secrets and even trained Microsoft engineers as the companies began working together
 - Microsoft used trade secrets to start developing products [Vista] that directly competed with their offerings, the lawsuit said.
- Microsoft said that in 2004 it bought from Veritas the rights to the technologies in question
- other examples:
 - Mac "rumor" sites leaking secrets
 - Eric Schmidt, Sun → Google



Who owns UW projects?

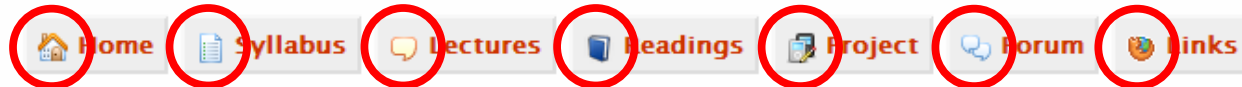
- In the US, inventors traditionally own their inventions.
 - Universities most often share monetary returns from commercialization.
- CSE commercialization committee
 - <http://www.cs.washington.edu/commercialization/>
- UW office of technology transfer
 - <http://depts.washington.edu/techtran/>
- Key question: "Who is funding the work?"

Find the IP violations!



University of Washington
Computer Science & Engineering

CSE 403: Software Engineering



Readings

If a given reading assignment has questions for you to turn in, the link to the questions will be on the [Lectures](#) page. Please check there frequently; it is your responsibility to notice and answer questions for readings that require them.

Week 1: Do at least one of the following readings:

- ~~#1: Software lifecycle models~~
 - [McConnell, *Rapid Development*, Ch. 7: Lifecycle Planning](#)
 - (Read the sections about Pure waterfall, Code-and-Fix, Spiral, Evolutionary Prototyping, Staged Delivery, and Design-to-Schedule. Also skim the table 7-1 that compares the models.)
 - (Don't have to read the "Case Study" shaded parts. Don't have to read about Modified Waterfalls, Salmon, Evolutionary Delivery, Design-to-Tools, or Commercial Off-the-Shelf Software.)
- ~~#2: Software requirements; Getting started on a software project~~
 - [Hunt/Thomas, *Pragmatic Programmer*, Ch. 7: Before the Project](#)
 - (Read only pp. 1-10 (stop at "Maintain a Glossary"), then p. 14, and pp. 17-19)