



Lawyers, Guns, and Money

(the title of a 1978 song by Warren Zevon)

Vivek Pai

1



Goals of this Lecture

- Help you learn how to:
 - Understand some aspects of law and business as it applies to software development
 - Promote a broader understanding of what life may be like in various business settings
- And thereby:
 - Keep you partially awake for the last lecture
 - Cover a topic that gets relatively little attention outside of specialized courses
- Disclaimer: I am *not* a lawyer, and I am *not* providing legal advice

2

Legal Concepts



- Copyright
- Patents
- Trade Secrets
- Derivative works
- Licenses
- Trademarks

3

Copyright



- From <http://www.copyright.gov/help/faq/faq-general.html>
 - Copyright is a form of protection grounded in the U.S. Constitution and granted by law for original works of authorship fixed in a tangible medium of expression. Copyright covers both published and unpublished works.
- Basically,
 - Anything you write/draw/etc is “automatically” protected from being copied, and you can opt to register your copyright to help enforce it
- Applies regardless of originality, unless you’re copying something else, translating, or mechanically reproducing it
- Protects you from someone literally copying the exact code, file, etc., but you still have to pursue it

4

Copyright Examples



- **You write a program in C and compile it**
 - The source file in C is protected by copyright
 - The binary file, mechanically produced from the source, is also protected by copyright
 - You can sell the copyrighted binary by itself
- **You draw an image**
 - The image is protected by copyright
 - Resized, resampled, recolored versions of the image are also protected by that same copyright

5

Copyright Ownership



- **Who owns the copyright?**
 - If working for yourself, you do
 - If working for someone else, they likely do
 - Even if done on your own time, you may have signed away your rights as part of an employment contract
- **What if you are a student?**
 - Undergrads typically regarded as individuals
 - Grad students typically regarded as employees
 - Undergrads working on internships, grants, etc., also considered employees
- **International issues**
 - Copyrights largely governed by Berne Convention
 - Covers 165 countries with largely uniform protection
 - Length of coverage varies, but typically 50+ years

6

Patent



- From <http://www.uspto.gov/patents/index.jsp>
 - A patent is an intellectual property right granted by the Government of the United States of America to an inventor “to exclude others from making, using, offering for sale, or selling the invention throughout the United States or importing the invention into the United States” for a limited time in exchange for public disclosure of the invention when the patent is granted.
- Basically,
 - You disclose how something new/original works in detail, in exchange for protection against others duplicating it for some time (e.g., 7-14 years)
- Applies only for things that are new and not “obvious”
- Protects you from people copying your invention, at the cost of telling everyone how it works

7

Patent Examples



- You design a new algorithm and implement it
 - E.g., like the RSA crypto algorithm for public keys
- You design a new protocol and implement it
 - E.g., like a remote-screen protocol for Windows
- You design a new shape for a product – design patent
- Patent is **not** automatic
 - In US, can apply up to 12 months from first public disclosure
 - In rest of world, no patenting after public disclosure
 - In large # of countries, can file provisional patent that “holds” the patent filing for a year
 - After provisional, must file patent in each country to be protected
 - Patent costs can be very large

8

Patents vs Copyrights



- **Scope**
 - Patents protect general idea, regardless of implementation
 - Copyrights protect specific implementation & direct derivatives
- **Novelty**
 - Patents have to be for new, non-obvious things
 - Copyrights can be on almost anything
- **Duration**
 - Patents are short-lived due to wider scope
 - Copyrights are long-lived
- **Prosecution**
 - Copyright infringement is usually fairly clear
 - Software patents are often a matter of interpretation
 - Molecular (i.e., drug) patents are typically much clearer

9

Trade Secret



- **Another mechanism for protecting something of value**
- **Basically**
 - Do not disclose it
 - Restrict access only to “need to know” employees
 - Clearly identify it as a trade secret
 - Make it clear in employment contract
- **Examples**
 - The Colonel’s secret recipe of herbs and spices
 - The formula for Coca-Cola
- **Recourses for dissemination**
 - Sue employee who violates trade secret
 - Use legal mechanisms regarding theft to suppress disclosure

10

Derivative Works



- **Main idea: works that derive from copyrighted works inherit those copyrights**
 - Pro: provides protection for extending work, compiling it into another form, aggregating it with other works, etc.
 - Con: typically cannot copyright derivative work separately to extend copyright duration, unless significant new material
- **Why it matters to developers**
 - Who owns derivative works in a contract?
 - Note: adding a .o file into a library (.so, .a) is typically a derivative work

11

License



- **The actual text that gives you permission to do something with a copyrighted/patented work**
 - The “click-through” license on many software packages
 - The “end user license agreement” on boxed software
 - The text accompanying/following the copyright notice on many source files
- **The license is a mini legal contract**
 - It can contain whatever the licensor wants it to say
 - The licensee usually has some limited set of options
 - Accept license
 - Reject license, then no right to use/run software

12

Sample Licenses



- **Single-user commercial license**
 - You can install this software binary on one machine and use it only on that machine
- **Commercial site license**
 - You can install this software on 100-500 machines owned by some organization
- **Permissive open source license**
 - You can use this software and source code in any way that you want as long as this copyright notice stays attached to it
- **Non-permissive free software license**
 - You can do anything you want with this software and source as long as you give the source code along with any binary you distribute

13

Trademarks



- The protected use of names & images for a certain market
- Gives perpetual exclusive use of name/image in that market
- Owner must aggressively pursue infringement to retain trademark
- Examples



14

Open Source vs Free Software



- **Free Software**
 - Coined by the Free Software Foundation
 - Best exemplified by the Gnu Public License (GPL)
 - Goal: allow end user to see, modify, and use all source code
- **Open Source Software**
 - Coined by the Open Source Initiative
 - Best exemplified by Apache Web Server, various Unix variants
 - Goal: encourage access to source code, but not necessarily dictate restrictions on what end user can or can't do

15

Free Software (GPL)



- Free derives from freedom, not necessarily free-of-charge
- License allows selling software & associated services
- Source code must be provided
 - Both for the GPL-licensed code & anything linked with it
 - No other restrictions can be added
 - E.g., cannot tell customers that they cannot release source
- How to monetize
 - Can sell the software (with source)
 - Can sell consulting/customization services
 - Can sell update services

16

How This Affects You



- You desire to avoid re-inventing wheel
- You need a component that does X
- You search for X, find it, and use it
- What happens next?

17

Using Licensed Components



- High-order rule: must follow license
- In permissively-licensed code
 - Typically, don't remove copyrights
 - Don't claim that portion of the code as your own
 - Can release source or not, as you choose
- In GPL-licensed code
 - Must release source for any `_linked_` code
 - What is linking?
 - Typically, linking of `.o` files or libraries
 - GNU libc licensed under "lesser GPL" – linking OK
 - Who gets its?
 - Anyone who get binary from you

18

Loophole Case Study #1



- GPL code used to build network service
 - Client connection to GPL-based code via HTTP
 - Client sends request, gets response
 - Client never receives binary – only run on company-owned servers
- Technically, fully compliant with GPL (version 2)
- Violated the spirit, but not the letter, of GPL

- Response: Affero GPL, which extended source release requirements to cover network-based services

19

Loophole Case Study #2



- Device manufacturer uses Linux (GPLv2)
 - Provides all source code
 - All code can be compiled to produce binaries
 - Vendor-custom hardware checks for signed binaries
 - Signed binary has checksum function that produces a value stored with binary
 - Refuses to boot unsigned binaries
- Result: end user cannot boot modified system
 - Fully compliant with letter, not spirit of GPL

- Response: GPLv3 requires release of all signing tools, keys

20

Loophole Case Study #3



- Red Hat Linux wanted to dominate Linux market
 - Would periodically release updates of Red Hat Linux
 - Charged for packaged software & update service
- However, each release was immediately followed by other vendors selling the same CDs at low cost
 - Legal since all of Red Hat's system was GPL
- Red Hat employed trademark protection
 - CDs contained Red Hat trademarks, including name & logo
 - Argued that consumers would be confused by third party sources
- Response: none, really
 - CentOS formed to release RedHat releases w/o RedHat trademarks
 - Sometimes large delay between RedHat release & CentOS release

21

Loophole Case Study #4



- Popular database software released as GPL
- Vendor did not accept outside code contributions
 - Vendor owned copyright to all code
- Software also available as non-GPL commercial license
- GPL version spurred adoption, market share, commercial version paid the bills
- Result: vendor acquired for \$1B
- Response: users worried about stagnation of GPL version
 - Nobody can revoke GPL version
 - New features do not have to appear simultaneously (if ever) in GPL version

22

Guns



- **Hired guns**
 - People who perform a task for money
 - Typically lose the ownership of the output
 - Called “work product” in the US
 - Sometimes “moral rights” preserved, esp in Europe
- **As a hired gun, you have few rights**
 - No control over the fate of the code
 - Ownership automatically assigned to employer
- **You do have some responsibilities**
 - Cannot violate licensing rules
 - Potentially liable for such violations, to owner & employer

23

Money



- **How do you make money?**
 - Consult
 - Work for a large company
 - Start a business
 - Work for a startup
 - Other
- **Common issues**
 - Salary
 - Profits, revenues
 - Stock & stock options
 - Ownership

24

Consultants



- Consulting can be fairly popular in booms
 - Companies want task-specific employment
 - Want to be able to easily hire & let go
 - Willing to pay more for specific skills & flexibility
- Consultants are typically
 - Independent contractors with no benefits – health, retirement, etc
 - Fully responsible for own taxes – no automatic withholding
 - Responsible for “employer’s share” of payroll taxes
 - Allowed to self-fund retirement accounts (with higher caps)
- Pros: higher hourly rate in many cases, even after taxes
- Cons: less stable employment, loneliness

25

Stock



- Stock – public companies are owned in fractions
- These fractions are shares of stock in the company
- Company is “worth” current share price * # of shares
 - Also known as “market capitalization” or “market cap”
 - E.g., Apple is a \$544B company by market cap, sales of \$144B
- Share price fluctuates, tied to expected future of company
- Some stocks pay owners some amount per share, called dividend
- Profit can be made by selling at higher share price, or receiving dividends, or both

26

Stock Options



- Companies want to align employee's interest with company's future
- Assumption is positive future for company tied to rising share price
- Giving employee stock is taxed like income (since employee can just sell stock)
- Instead, give employee a stock option
 - The right to purchase a certain number of shares at a given price
 - The given price called a "strike price"
 - Worth something if current stock price above stock price
 - Worth nothing if stock price lower than strike price
 - # of shares available may increase over time, known as vesting

27

Restricted Stock Units



- Actual grants of stock
 - But become yours over time (vesting)
- Taxable typically as soon as it vests (vs exercise)
- Keeps employees motivated when share price fluctuates
- Costs the company more, but easier accounting
- Becoming much more popular than before

28

Employees – Large Firm



- **Salaried employees**
 - Receive fixed salaries (no overtime) as professionals
 - Receive health insurance & retirement benefits
 - Sometimes get stock options
 - Stock options likely to be very small % of company
 - Sometimes get profit sharing or other bonuses
- **Pros: stability, unlikely to fail massively**
- **Cons: less direct remuneration from actions**

29

Startup Economics



- **Founders – may have salary, outright stock (not options)**
 - Founders typically split original stock in company
 - Some portion of stock set aside for employee pool
 - Founders (and other employees) get ownership diluted as more stock is issued to investors
 - Founders own what are called “common shares”
- **Employees – salaried, with stock options**
 - Stock options may be “relatively” large % of company
 - First employee often 1-3% of company
- **Investors**
 - Typically angel funding for <\$1M, venture capitalists for larger
 - Hold a “higher class” of stock that gets paid first, called preferred
 - May also hold debt, which is foremost a loan (and paid first)

30

How a Small Startup Works



- Two friends form a company, 50% each
 - Work for six months, build viable prototype
 - Release prototype, see market traction, need to grow
 - Approach investors
- Investor evaluates company, puts in money
 - May say that the current company is “worth” \$2M
 - Invests \$3M to grow the company
 - Post-money company now worth \$5M (\$2M + \$3M)
 - Investor owns 60% of company
 - Investor may dictate that future employees receive 20% of shares
- Original owners now have $100\% - 60\% - 20\% = 20\%$ of shares

31

If Company Gets Acquired



- Acquired for \$3M
 - Investor gets back all \$3M, weeps
- Acquired for more than \$3M
 - Investor may have “preference multiple”
 - If 2x or 3x, first 6M or 9M go to investor
 - If investor has participation, then after payback, investor’s shares become common shares
- Scenario: assume 1x preference + participation, \$8M exit
 - Investor first gets \$3M back
 - Now, \$5M left for all shares
 - Investor gets 60% of this, another \$3M
 - Owners get 20% of this, for \$1M
 - Employees purchase their stock at strike price, get difference between that and remaining \$1M

32

Why Take Outside Money



- **Much easier to grow**
 - Profits small, elusive when starting
 - Organic growth can be very slow
 - First-mover often signs up more customers
 - Market leader more likely to be acquired
- **Less upside, less downside**
 - In event of failure, investors lose, founders walk
 - In event of success, investors gain, founders gain less
 - Smaller % gain may beat larger % of small profit
- **Investors have vested interest in success**
 - Investors want to see 3x or more returns
 - Will work to make that happen if possible
 - If you owe \$1M, it's your problem. If you owe \$100M, it's their problem

33

Course Wrap-up



- **Final projects, final exam**
 - Yes, end of semester is stressful
 - Please don't do anything hasty
- **Dean's Date is fixed**
 - University deadline
 - Talk to us & res college dean early if many problems
- **Final exam is *closed-book, closed-notes***
 - Format comparable to midterm (except closed-everything)
 - 3 hours provided
 - Unlikely to have time pressure

34