290T: The Business of Software: Teams Creating Barriers and IP

> Professor Kurt Keutzer Fall 2003 EECS keutzer@eecs.berkeley.edu

IP material from David Baumer, NCSU

Class News

- Suggestion for schedule:
- Wrap up on 11/24
- Project presentations:
 - 12/1 and 12/3 4:30 6:00PM
 - What do people think?

2

Outline

- Reflections on teams, team dyanamics and software
 - Qualities of successful (and unsuccessful teams)
 - Interpersonal team dynamics
- Creating barriers to entry for software, and intellectual property (IP) protection

3

4

People and Teams in SW

- The human factor people and team organization is always important in a business, but why is it especially true in a software business?
- · People and teams embody the entire value of the company
 - "Our intellectual property gets in the car and drives home at night. How can we ensure that they come back again tomorrow?"
- What about the software itself?
- It is often useless without those who wrote it/or understand it
- What about the patents?
- Patents are hard to defend without inventors

Key SW Team Members

- · Key elements of a software product team
- Product leader the engine
 - spends 24 hours a day working to make project successful
- Architect the chassis
 - understands the current problem formulation and how the software is structured to solve it
- Marketeer the steering wheel
 - customer champion understands the customer's problem
- Visionary the map reader
 - · knows where you've been and where you are going
- (Backstop) anchor, ballast, lifeboat
 - · answers any question, codes the team out of almost any bind
- ``Find the right people and fire them [like a rocket]." G. Tate, CEO Rambus
- If you get the right people in the right positions the rest will take care of itself.

5

6

Product Team Reviews Showed

- As Chief Scientist and later CTO, reviewed 25 core/product teams:
- 100% correlation between team presence of team roles (Leader, Architect, Marketeer, Visionary, Backstop) and product success
- The roles were
 - Understood
 - Staffed
 - Recognized
 - · team members respected each other's roles
 - Members rarely competed for roles (E.g. Never heard: ``Well really, I'm running the product team even though Buford is project leader.'')

On the other hand

- Unsuccessful products and their teams tended to lack all (except perhaps 1 role):
- Product leader ``I just got assigned to this job.''
- Architect ``I just don't understand why anyone would structure a piece of software this way.''
- Visionary ``We're just trying to make it through this release then we'll have time to think about the future.''
- Marketeer ``I'm working hard to make my career, eh, I mean the product oh, I mean the customer, successful."
- Insult to the injury: Team members spend inordinate amount of time worrying about other people *not* doing their job.

8

Back to successful teams

- Why is it that successful teams staffed all jobs and unsuccessful teams staffed none?
 - Product leader's dedication
 - · Architect's grasp of the software product
 - · Visionary's captivating vision
 - Marketeer's customer awareness and advocacy
 - Backstop's solidity
- · Each one of these individuals skills draws the others

Overall characteristics of successful teams

- Clear roles and responsibilities
- Mutual respect
- Mutual accountability
- Genuine admiration

9

Outline

- Reflections on teams, team dyanamics and software
 - Qualities of successful (and unsuccessful teams)
 - Interpersonal team dynamics
- Creating barriers to entry for software, and intellectual property (IP) protection

Individuals and Group Dynamics

- Jung psychological foundation
- Myers-Briggs modern theory
- Keirsey etc. broad dissemination <u>www.keirsey.com</u>
- Last ``free'' site: http://www.humanmetrics.com/
- The key dimensions
 - Introversion (I) vs. Extroversion (E)
 - Intuitive (N) vs. Sensing (S)
 - Thinking (T) vs. Feeling (P)
 - Perceptive (P) vs. Judging (J)
- There's no right and wrong and you can't really change yourself anyway!
- Learn the strengths and weakness of your character type

12

Understanding the Dimensions - 1

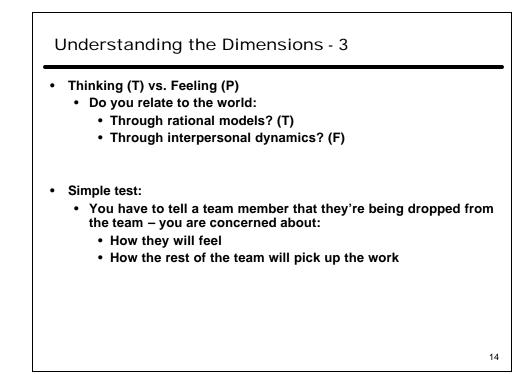
- Introversion (I) vs. Extroversion (E)
 - Are you energized by:
 - Being alone? (I)
 - Being with others? (E)
- Easy test- you're on the road (at a conference, campus offsite activity, whatever) at 10PM you're:
 - At the bar (or in the jacuzzi) chatting away
 - In your room watching television

Understanding the Dimensions - 2

- Intuitive (N) vs. Sensing (S)
 - Do you learn and organize information:
 - Top down? (N)
 - Bottom up? (S)

• Easy test:

- Asked: ``How does the software work?''
- You:
 - · Jump into the details of the software modules
 - Draw a module-block diagram description of the whole system



Understanding the Dimensions - 4

- Perceptive (P) vs. Judging (J)
 - In decision making do you:
 - Like keeping as many options open as long as possible? (P)
 - Like to nail things down as soon as possible? (J)

• Simple test:

- When planning a vacation you like to:
 - · Have a detailed itinerary for every day
 - Keep things open so you can be spontaneous

Why is this important ? - 0

- Our natural inclination is to believe that how we are is the best:
 - ENTJ hard driving project leader
 - ISFP sensitive team member working for the good of all
- · And to be different is to be inferior
- We must understand differences in how other people view the world if we are to work successfully with them
- We do need to work successfully with people who are very different from ourselves –not just to make the team harmonious but to cover our own weaknesses

I's and E's

- Highly extroverted people are energized by interpersonal interactions
- Highly introverted people are depleted by interpersonal interactions
- Each type needs the other, but often fundamentally misunderstands the other
- I's think that E's talk to much
- E's think I's are anti-social
- Typical challenging scenario: E boss to I employee
 - ``Rick, I know I was going to give you the afternoon to work on that, but I decided it would be better if the group got together and brainstormed a solution'''

N's and S's

- Highly intuitive people view the world top down
- Highly sensing people view the world bottom up
- Each type needs the other, but often fundamentally misunderstands the other
- N's think S's are a bit slow and plodders
- S's think that N's are a bit flighty
- But
 - N's will give S's a product roadmap
 - S's will give N's reliable products
- Challenging scenario: High-N boss to High-S employee
 - ``Rick, can you give me simple top-down assessment of the time to build that new product I'm having lunch with the CEO today''.

T's and F's

- Highly Thinking people view the world as a series of rational decisions to be optimized
- Highly Feeling people view the world as a series of interpersonal interactions
- Each type needs the other, but often fundamentally misunderstands the other
- T's think F's are too sentimental
- F's think that T's are a bit cold
- Challenging scenario: High T boss to High F middle-manager
 - ``Jim, I'm going to leave it to you to communicate my decision about lay-offs in your group. After all, they're you're people and you know them better than I do.''

19

P's and J's

- Highly perceptive people are exhilarated by the sense of opportunity that is associated with deferred decisions and feel confined and de-motivated when too many things are nailed down
- Highly judging people are in a state of anxiety when too many decisions are deferred and feel its hard to operate in uncertainty.
- P's think J's are boring and rigid
- J's think that P's are indecisive and waffling
- But
 - P's will give J's a broad range of possibilities
 - J's will give P's products, quarterly revenues etc. on-time
- Challenging scenario: High-P boss to High-J employee:
- Charlie: ``Steve, I like to keep a lot of balls up in the air. It gives me a greater sense of opportunity.''
- Steve: ``Charlie, I like to keep the balls on my desk where I can keep an eye on them!"

Outline

- Reflections on teams, team dyanamics and software
 - Qualities of successful (and unsuccessful teams)
 - Interpersonal team dynamics
- Creating barriers to entry for software, and intellectual property (IP) protection

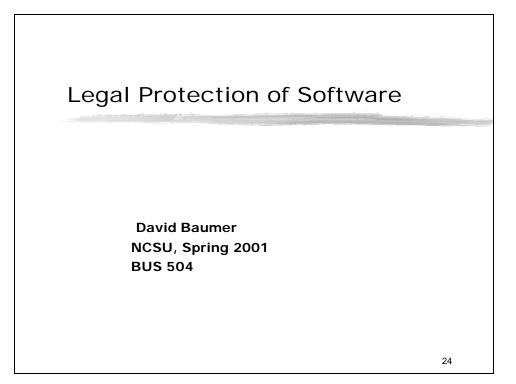
Software's virtues are its vices

- Low start-up and development costs means
 - Easy to get a business venture started but ...
 - Harder to create a barrier to entry from a competitor
- I'd like you to take a few minutes now, in your project groups to identify ways that you would create barriers to entry for your projects/products

Barriers to entry

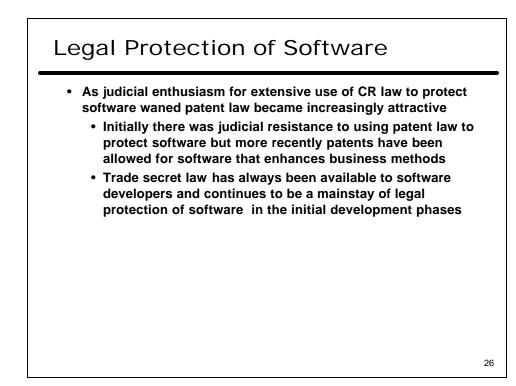
These are partially overlapping ("not orthogonal in nerdspeak")

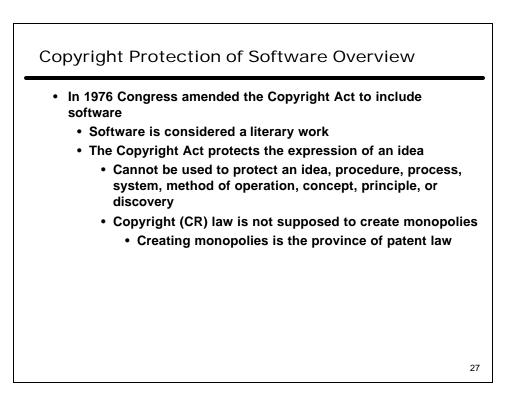
- Lock on the shelf space
 - Unique relationship with distributors that eliminates competitors
 - Challenge: ruthlessly competitive distributors
- Own a key link in the user flow
 - Own the source, define the language e.g. Matlab
 - Own the target (e.g. target hardware)
 - Challenge: trends toward standardization
- Dominant market share
 - Especially true for software
 - Low incremental cost so dominant player amortizes the development cost the most
 - Challenge: evolving markets, innovators dilemma
- Technical lock exclusive access to technologies/individuals that are essential to success of your endeavor

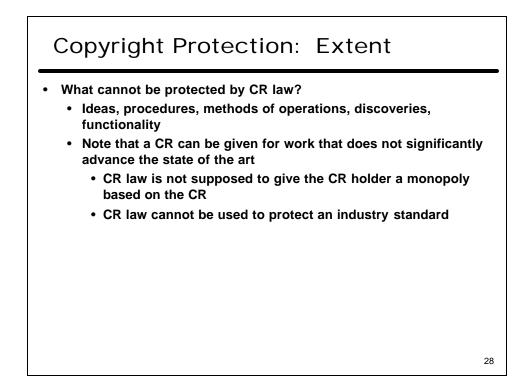


Legal Protection of Software

- Originally it was anticipated that CR (copyright) law would provide the protection for software
 - In 1976 CR Act revised to include software protection
 - From 1976 to 1986 CR protection of software expanded with the "look and feel" test
 - Protection was provided for the look and feel of software programs
 - Gradually courts began to cut back on use of look and feel test and thus CR protection of software has receded

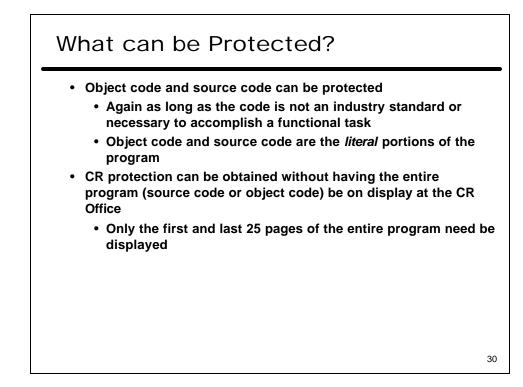






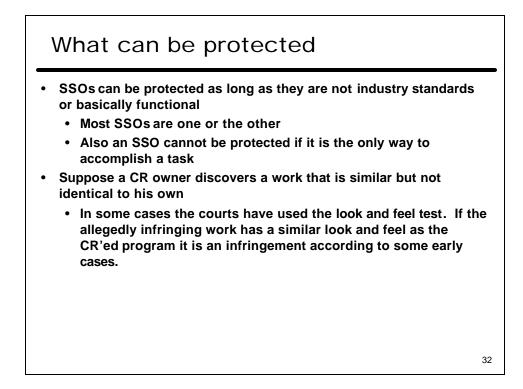
Copyright Law: Extent

- CR law cannot protect:
 - material that is in the public domain
 - Once in the public domain it cannot be plucked out by use of CR
 - material that has previously been licensed or assigned to someone else for their use during the duration of the lease



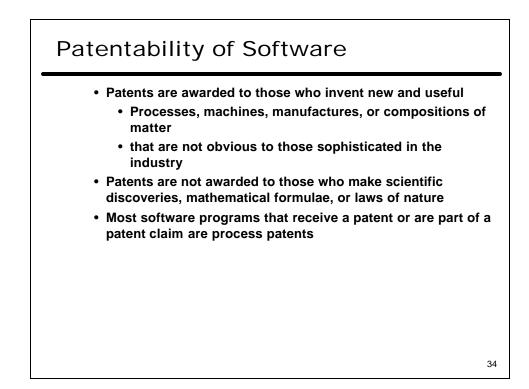
What Can Be Protected

- Screen displays and user interfaces are part of the *nonliteral* portions of the program
 - What appears on the computer screen can be protected under CR law
 - In addition the *structure*, *sequence*, and *organization* (SSO) of a computer program can be protected
 - Most programs make use of icons so the user interface is graphical (GUI)



CR Protection for Software

- It is possible to classify CR protection for software as broad:
 - Here we are applying the "look and feel" test--the legal protection extends even beyond what appears on the screen to the "look and feel" of the program
- Suppose the court says that the software program is only entitled to "thin" protection:
 - What the court means is that the CR owner is only protected against copying of the code (source and object) with regard to material that is protectible

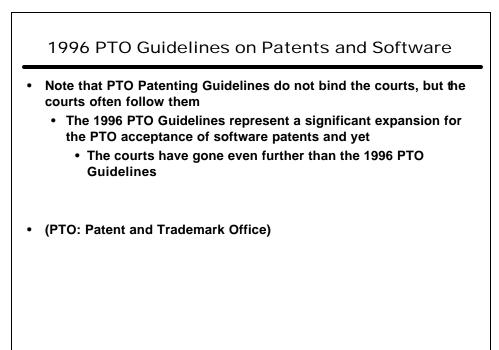


Patentability of Software

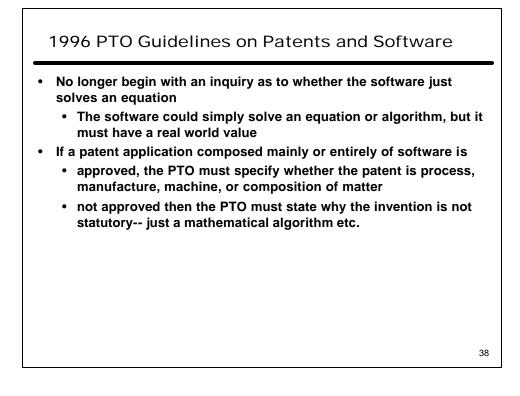
- If software does nothing but solve a mathematical algorithm, it is not patentable
 - Gottschalk v. Benson
- If software is part of a process, it is patentable
 - Diamond v. Diehr
- Following *Diehr* the courts devised a two-part test:
 - Did the application contain software?
 - Did the application have contain more than just software solutions to algorithms?

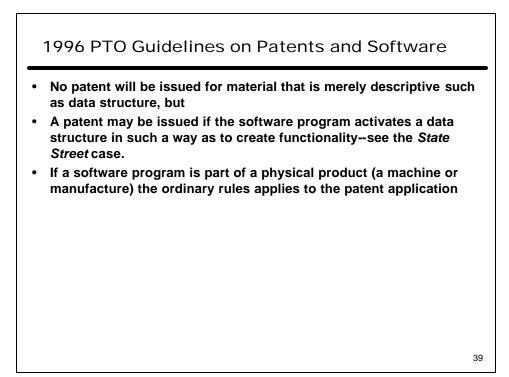
Patentability of Software

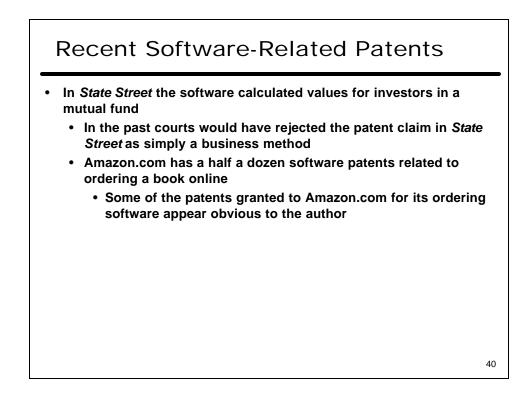
- As time when on, the judicial reception for software-related patents increased:
 - Software that mathematically transformed heart palpitations into values of mean square errors received a patent
 - Arrhythmia v. Corazonix
 - A process whereby software transformed waveforms from an oscilloscope into pixels on a TV screen was patentable
 - In re Alappat











Trade Secrets

- Trade Secrets are [were?] the most commonly used method of protecting software secrets
 - A commonly used definition is that a trade secret is information (broadly defined) that
 - Derives economic value by not being known by business rivals and
 - Is the subject of reasonable efforts to keep secret
 - Obviously, the development of software makes extensive use of trade secrets's

Trade Secrets

- It is a tort for a firm or an individual to wrongfully appropriate a trade secret
 - Defendants in trade secret litigation are called misappropriators
 - Trade secret litigation basically involves an investigation as to whether the def. wrongfully obtained trade secret information
 - It is not a tort to reverse engineer a product of a rival and discover the secrets of how to make the product
 - It is a tort for a firm to obtain trade secrets by making use of wire taps or inducing former employees to breach non disclosure agreements (NDAs)

Trade Secrets

- Juries are required to determine whether the appropriation of a trade secret was wrongful
 - If firm obtains trade secrets from a former employee of another firm the standard is
 - Whether the firm knew or should have known that the information involved a breach of the ex-employee's fiduciary duty or a nondisclosure agreement.

Security Measures

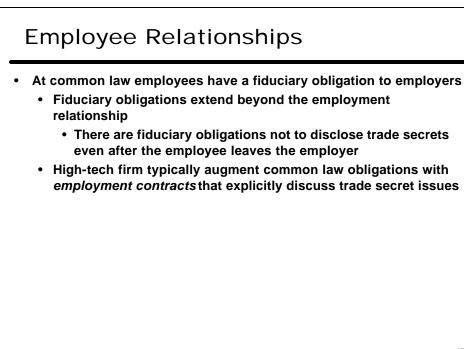
- It is not a tort if a firm obtains trade secrets through lawful means
 - The owner of a trade secret must use reasonable means to protect the trade secret
 - In the case of software protection, what is reasonable security may involve very high-tech and exotic measures
 - The courts take into account the resources of the firm in determining whether reasonable security measures were undertaken to protect trade secrets.

Reasonable Security Measures

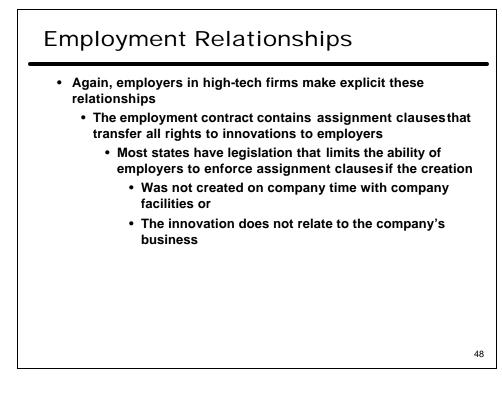
- Reasonable Security Measures include:
 - Control of physical access through badges, logs, guards, and locks
 - Computer systems often contain trade secrets.
 - · Controlling access to firm computers is crucial
 - Physical access should be regulated
 - For computer secrets, passwords, firewalls, and cryptography are standard and essential parts of a security system

Reasonable Security Measures

- · The effectiveness of security measures depends on
 - · The commitment of management and employees
 - Employees with a grudge can of course create an enormous amount of havoc

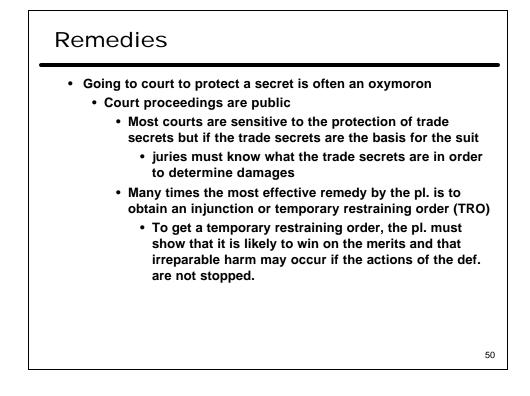






Trade Secret Protection

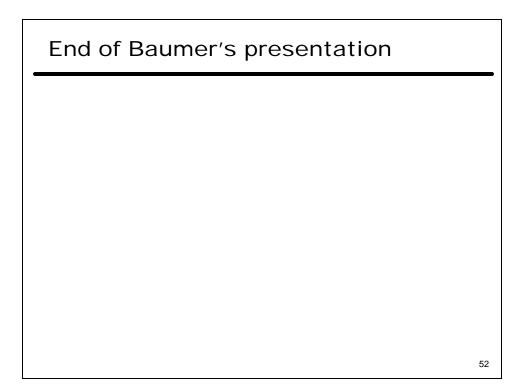
- Note that the courts do not favor clauses that effectively make exemployees unemployable
 - General skills acquired by an employee on the job are can be used in subsequent jobs by the employee
 - An employer cannot simply declare a skill developed by an employee a trade secret and prevent the employee from using the skills



Trade Secret Remedies

- Damages must be proved by the plaintiff with reasonable certainty
 - Lost profits on lost sales
 - Plaintiff must show that it would have made the sales but for the trade secret misappropriation
 - Alternatively, the pl. can obtain reasonable royalties on sales made by the def. due to the trade secrets.
 - Under the Uniform Trade Secret Act (ratified by most states) damages for misappropriation of trade secrets are doubled
 - In many states misappropriation of trade secrets are considered an unfair trade practice and thus subject to a tripling of damages





KK's Use of Intellectual Property Law

- · Copyright appears to be virtually useless
- Trade secrets
 - · Challenged by California's "at will" employment
 - Courts favor freedom of the employee
 - Problems with enforcement (company vs. individual and public dis closure)
 - Patents more and more commonly the first choice for IP protection
- · Requisite of a software business plan
- Strategies:
 - · Build a portfolio to keep competitors from arising
 - · Build a portfolio to give teeth to battle with competition
 - Build a portfolio to swap with gorillas lest they crush you
- Two meta strategies:
 - Build a BIG portfolio principally aimed at swapping with other companies and little intent to prosecute
 - · Build a SMALL portfolio with full intention to prosecute your patents
- Challenges:
 - Time (top engineers in your company) and money to build portfolio
 - Time (top engineers in your company) and money to swap portfolios
 - Time (top engineers in your company) and money to prosecute a patent

53

Summary

- · Understand team roles and staff them
 - Leader
 - Architect
 - Marketeer
 - Visionary
 - Backstop
- Understand and appreciate differences in human types
- · Understand means to constructing barriers to entry