

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

1

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

3

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

4

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

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.”)

6

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.**

7

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**

8

Overall characteristics of successful teams

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

9

Outline

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

10

Individuals and Group Dynamics

- Jung – psychological foundation
- Myers-Briggs – modern theory
- Keirsey etc. – broad dissemination www.keirsey.com
- 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

11

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

12

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

13

Understanding the Dimensions - 3

- **Thinking (T) vs. Feeling (F)**
 - Do you relate to the world:
 - Through rational models? (T)
 - Through interpersonal dynamics? (F)
- **Simple test:**
 - You have to tell a team member that they're being dropped from the team – you are concerned about:
 - How they will feel
 - How the rest of the team will pick up the work

14

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**

15

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**

16

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``**

17

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``.**

18

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!”**

20

Outline

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

21

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**

22

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**

23

Legal Protection of Software

David Baumer
NCSU, Spring 2001
BUS 504

24

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

25

Legal Protection of Software

- As judicial enthusiasm for extensive use of CR law to protect software waned patent law became increasingly attractive
 - Initially there was judicial resistance to using patent law to protect software but more recently patents have been allowed for software that enhances business methods
 - Trade secret law has always been available to software developers and continues to be a mainstay of legal protection of software in the initial development phases

26

Copyright Protection of Software Overview

- In 1976 Congress amended the Copyright Act to include software
 - Software is considered a literary work
 - The Copyright Act protects the expression of an idea
 - Cannot be used to protect an idea, procedure, process, system, method of operation, concept, principle, or discovery
 - Copyright (CR) law is not supposed to create monopolies
 - Creating monopolies is the province of patent law

27

Copyright Protection: Extent

- What cannot be protected by CR law?
 - Ideas, procedures, methods of operations, discoveries, functionality
 - Note that a CR can be given for work that does not significantly advance the state of the art
 - CR law is not supposed to give the CR holder a monopoly based on the CR
 - CR law cannot be used to protect an industry standard

28

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**

29

What can be Protected?

- **Object code and source code can be protected**
 - **Again as long as the code is not an industry standard or necessary to accomplish a functional task**
 - **Object code and source code are the *literal* portions of the program**
- **CR protection can be obtained without having the entire program (source code or object code) be on display at the CR Office**
 - **Only the first and last 25 pages of the entire program need be displayed**

30

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)

31

What can be protected

- SSOs can be protected as long as they are not industry standards or basically functional
 - Most SSOs are one or the other
 - Also an SSO cannot be protected if it is the only way to accomplish a task
- Suppose a CR owner discovers a work that is similar but not identical to his own
 - In some cases the courts have used the look and feel test. If the allegedly infringing work has a similar look and feel as the CR'ed program it is an infringement according to some early cases.

32

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

33

Patentability of Software

- Patents are awarded to those who invent new and useful
 - Processes, machines, manufactures, or compositions of matter
 - that are not obvious to those sophisticated in the industry
- Patents are not awarded to those who make scientific discoveries, mathematical formulae, or laws of nature
- Most software programs that receive a patent or are part of a patent claim are process patents

34

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?

35

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**

36

1996 PTO Guidelines on Patents and Software

- **Note that PTO Patenting Guidelines do not bind the courts, but the courts often follow them**
 - **The 1996 PTO Guidelines represent a significant expansion for the PTO acceptance of software patents and yet**
 - **The courts have gone even further than the 1996 PTO Guidelines**
- **(PTO: Patent and Trademark Office)**

37

1996 PTO Guidelines on Patents and Software

- **No longer begin with an inquiry as to whether the software just solves an equation**
 - **The software could simply solve an equation or algorithm, but it must have a real world value**
- **If a patent application composed mainly or entirely of software is**
 - **approved, the PTO must specify whether the patent is process, manufacture, machine, or composition of matter**
 - **not approved then the PTO must state why the invention is not statutory-- just a mathematical algorithm etc.**

38

1996 PTO Guidelines on Patents and Software

- **No patent will be issued for material that is merely descriptive such as data structure, but**
- **A patent may be issued if the software program activates a data structure in such a way as to create functionality--see the *State Street* case.**
- **If a software program is part of a physical product (a machine or manufacture) the ordinary rules applies to the patent application**

39

Recent Software-Related Patents

- **In *State Street* the software calculated values for investors in a mutual fund**
 - **In the past courts would have rejected the patent claim in *State Street* as simply a business method**
 - **Amazon.com has a half a dozen software patents related to ordering a book online**
 - **Some of the patents granted to Amazon.com for its ordering software appear obvious to the author**

40

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

41

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)

42

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.**

43

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.**

44

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**

45

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**

46

Employee Relationships

- **At common law employees have a fiduciary obligation to employers**
 - **Fiduciary obligations extend beyond the employment relationship**
 - **There are fiduciary obligations not to disclose trade secrets even after the employee leaves the employer**
 - **High-tech firm typically augment common law obligations with *employment contracts* that explicitly discuss trade secret issues**

47

Employment Relationships

- **Again, employers in high-tech firms make explicit these relationships**
 - **The employment contract contains assignment clauses that transfer all rights to innovations to employers**
 - **Most states have legislation that limits the ability of employers to enforce assignment clauses if the creation**
 - **Was not created on company time with company facilities or**
 - **The innovation does not relate to the company's business**

48

Trade Secret Protection

- **Note that the courts do not favor clauses that effectively make ex-employees 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**

49

Remedies

- **Going to court to protect a secret is often an oxymoron**
 - **Court proceedings are public**
 - **Most courts are sensitive to the protection of trade secrets but if the trade secrets are the basis for the suit**
 - **juries must know what the trade secrets are in order to determine damages**
 - **Many times the most effective remedy by the pl. is to obtain an injunction or temporary restraining order (TRO)**
 - **To get a temporary restraining order, the pl. must show that it is likely to win on the merits and that irreparable harm may occur if the actions of the def. are not stopped.**

50

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**

51

End of Baumer's presentation

52

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 disclosure)
- 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

54