# How to Have a Bad Career in Research/Academia Pre-PhD and Post-PhD (& How to Give a Bad Talk)

**David Patterson**

**UC Berkeley**

**November 18, 2015**

`https://goo.gl/ChRlHT`

## Acknowledgments & Related Work


- **Many of these ideas came from (inspired by?) Tom Anderson, David Culler, Al Davis, Ken Goldberg, John Hennessy, Steve Johnson, John Ousterhout, Randy Katz, Bob Sproull, Carlo Séquin, Bill Tetzlaff, …**
- **Studs Terkel, *Working: People talk about what they do all day and how they feel about what they do*. (1974) The New Press.**
- **"How to Give a Bad Talk" (1983), http://www.cs.berkeley.edu/~pattrsn/talks/BadTalk.pdf**
- **"How to Have a Bad Career" (1994), Keynote address, Operating Systems Design and Implementation Conf.**
- **Richard Hamming, "You and Your Research" (1995), www.youtube.com/watch?v=a1zDuOPkMSw**
- **Ivan Sutherland, "Technology and Courage" (1996).**
- **"How the RAD Lab space came to be" (2007), https://radlab.cs.berkeley.edu/wiki/space/history**
- **"Your Students are Your Legacy" (2009) *Communications of the ACM* 52.3: 30-33.**
- **"How to Build a Bad Research Center" (2014) *Communications of the ACM* 57.3: 33-36.**

# Outline

- Part I How to Have Bad Grad Student Career, and How to Avoid One
- Q&A
- Part II How to Have Bad Research Career
- Part III How to Avoid a Bad Research Career
  - Richard Hamming (Turing Award for error-detecting and error-correcting codes) video clips from "You and Your Research" (1995)
- Q&A
- My Story: Accidental Academic (3 min)
- What Works for Me (3 min)

# Part I: Commandments on to Have a Bad Graduate Career


## I. Concentrate on getting good grades

- Postpone research involvement: might lower GPA
- Aim for PhD class valedictorian!

## Alternative: Maintain reasonable grades

- No employer cares about GPA
  - » Sorry, no valedictorian
- Only once I gave below B in grad course
- 3 prelim courses only real grades that count
- What matters: Letters of recommendation
  - » From 3-4 faculty & external PhDs who have known you for 5+ years

# Part I: Commandments on to Have a Bad Graduate Career

II. **Concentrate on graduating as fast as possible**
  - Winner is first in class to PhD
    - » Only care PhD & GPA, not what you know
  - Don't spend a summer in industry: takes longer
    - » How could industry experience help with topic?
    - » Or letters of reference?
  - Don't work on large projects: takes longer
    - » Have to talk to others, have to learn different areas
  - Don't do a systems PhD: takes longer

- **Alternative: Your last chance to learn (mostly outside classroom)**
  - Considered newly "minted" when finish PhD
    - » No youth credit post PhD
  - Judged on year of PhD vs. year of birth
  - To person in 40s or 50s, 27 ≈ 29

# Part I: Commandments on to Have a Bad Graduate Career

## III. Don't go to conferences

- It costs money and takes time
- You'll have plenty of time to learn the field after graduating

- **Alternative: Chance to see firsthand what the field is like, where its going**
  - Talk to people in the field in the halls as well as go to talks
  - If your advisor won't pay, then pay it yourself
    - » Prof. Landay paid his own way to conferences while grad student
    - » There are student rates, can share a room

# Part I: Commandments on to Have a Bad Graduate Career

## IV. Don't trust your advisor

- Advisor is only interested in his or her own career, not yours
- Advisor may try to give you work to do, which uses up your time, ⇒ could interfere with GPA & delay graduation

- **Alternative: Try trusting your advisor**
  - Primary attraction of campus vs. research lab is grad students
  - Grad students reward for academic career
    - » Faculty career is judged by success of students
  - Why not try taking advice of UC Berkeley Prof?

# 5 Writing Commandments for a Bad Career

I. Thou shalt not define terms, nor explain anything

II. Thou shalt replace “I will build” with “has been built”

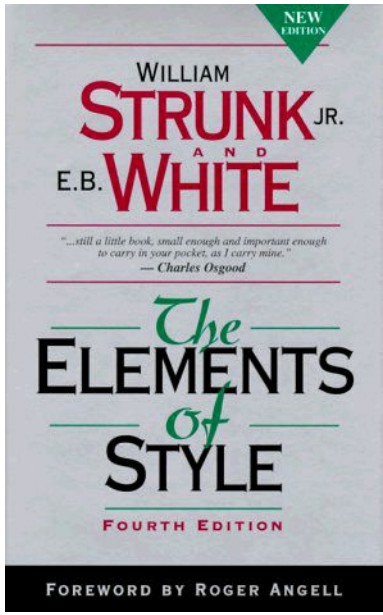
III. Thou shalt not mention drawbacks to your approach

IV. Thou shalt not reference any papers

V. Thou shalt publish before implementing

# Alternatives to Bad Papers

- Do opposite of Bad Paper commandments
  - Define terms, distinguish “will do” vs. “have done”,
  - Mention drawbacks, real performance, reference other papers.
  - Find related work via Google scholar…
- First read Strunk and White, then follow these steps;
  1. 1-page paper outline, with tentative page budget/section
  2. Paragraph map
     - » 1 topic phrase/sentence per paragraph, hand drawn figures w. captions (white board & photo)
  3. (Re)Write draft
     - » Long captions/figure can contain details ~ Scientific American
     - » Uses Tables to contain facts that make prose dreary
  4. Read aloud
  5. Grammar check
     - » Pearson Writer ($15/year for academics) or
     - » MS Word - select “technical” for writing style
  6. Get feedback from friends and critics on draft; go to 3.
- `www.cs.berkeley.edu/~pattrsn/talks/writingtips.html`



# 10 Talk Commandments for a Bad Career

I. **Thou shalt not be neat**
II. **Thou shalt not waste space**
III. **Thou shalt not covet brevity**
IV. **Thou shalt cover thy naked slides**
V. **Thou shalt not print large**
VI. **Thou shalt not use color**
VII. **Thou shalt not illustrate**
VIII. **Thou shalt not make eye contact**
IX. **Thou shalt not skip slides in a long talk**
X. **Thou shalt not practice**

# Following all the commandments in Powerpoint!

- We describe the philosophy and design of the control flow machine, and present the results of detailed simulations of the performance of a single processing element. Each factor is compared with the measured performance of an advanced von Neumann computer running equivalent code. It is shown that the control flow processor compares favorably in the program.

- We present a denotational semantics for a logic program to construct a control flow for the logic program. The control flow is defined as an algebraic manipulator of idempotent substitutions and it virtually reflects the resolution deductions. We also present a bottom-up compilation of medium grain clusters from a fine grain control flow graph. We compare the basic block and the dependence sets algorithms that partition control flow graphs into clusters.

- A hierarchical macro-control-flow computation allows them to exploit the coarse grain parallelism inside a macrotask, such as a subroutine or a loop, hierarchically. We use a hierarchical definition of macrotasks, a parallelism extraction scheme among macrotasks defined inside an upper layer macrotask, and a scheduling scheme which assigns hierarchical macrotasks on hierarchical clusters.

- We apply a parallel simulation scheme to a real problem: the simulation of a control flow architecture, and we compare the performance of this simulator with that of a sequential one. Moreover, we investigate the effect of modeling the application on the performance of the simulator. Our study indicates that parallel simulation can reduce the execution time significantly if appropriate modeling is used.

- We have demonstrated that to achieve the best execution time for a control flow program, the number of nodes within the system and the type of mapping scheme used are particularly important. In addition, we observe that a large number of subsystem nodes allows more actors to be fired concurrently, but the communication overhead in passing control tokens to their destination nodes causes the overall execution time to increase substantially.

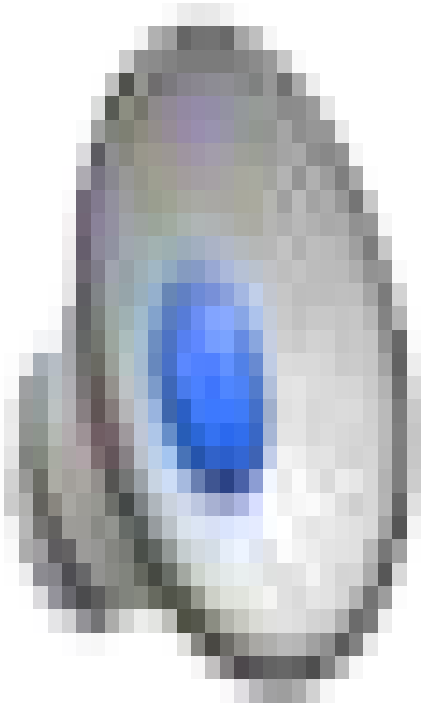
- The relationship between the mapping scheme employed and locality effect in a program are discussed.

- Medium grain execution can benefit from a higher output bandwidth of a processor and finally, a simple superscalar processor with an issue rate of ten is sufficient to exploit the internal parallelism of a cluster. Although the technique does not exhaustively detect all possible errors, it detects nontrivial errors with a worst-case complexity quadratic to the system size. It can be automated and applied to systems with arbitrary loops and nondeterminism.

# Alternatives to Bad Talks

- Do opposite of Bad Talk commandments
- Allocate 2 minutes per slide, leave time for questions
- Don’t over animate
- Do dry runs with friends/critics for feedback,
  - including tough audience questions
- Record a practice talk (video)
  - Don’t memorize speech, but have notes ready
- IBM: “Giving a first class ‘job talk’ is the single most important part of an interview trip. Having someone know that you can give an excellent talk before hand greatly increases the chances of an invitation. That means giving great conference talks.”

# Richard Hamming on Importance of Communication



# Part I: Alternatives to a Bad Graduate Career

- **Advice from a very successful "student"; Remzi Arpaci (now Wisconsin Professor)**
  - **Why do you think you did so well?**
  - **Remzi: Advice you gave me first week I arrived**
  - **What did I say?**
  - **Remzi: 3 observations, still good advice**

1. **"Swim or Sink"**
   - **"Success is determined by me (student) primarily"**
   - **Faculty will set up opportunity, but its up to me leverage it**
- **"Read/learn on your own"**
  - **"Related to 1), I think you told me this as you handed me a stack of about 20 papers"**
- **"Teach your advisor"**
  - **"I really liked this concept; go out and learn about something and then teach the professor"**
  - **Fast moving field, don't expect Prof to be at forefront everywhere**

# Outline for Bad Research Career (Post-PhD)

- **Part III: 6 Commandments for a Bad Research Career**
  I. **Be THE leading expert**
  II. **Let Complexity Be Your Guide (Confuse Thine Enemies)**
  III. **Never be Proven Wrong**
  IV. **Use the Computer Scientific Method**
  V. **Don't be Distracted by Others (Avoid Feedback)**
  VI. **Publishing Journal Papers IS Technology Transfer**
- **Part IV: Advice on Alternatives to a Bad Research Career**

# Bad Career Move #1: Be THE leading expert

- **Invent a new field!**
  - **Make sure its slightly different**
- **Be the real Lone Ranger: Don't work with others**
  - **No ambiguity in credit**
  - **Adopt the Prima Donna personality**
    - » ***Prima Donna: a very temperamental person with an inflated view of their own talent or importance***
- **Research Horizons**
  - **Never define success**
  - **Avoid Payoffs of less than 20 years**
  - **Stick to one topic for whole career**
  - **Even if technology appears to leave you behind, stand by your problem**

# Bad Career Move #2: Let Complexity Be Your Guide (Confuse Thine Enemies)

- **Best compliment:**
  **"Its so complicated, I can't understand the ideas"**
- **Easier to claim credit for subsequent good ideas**
  - **If no one understands, how can they contradict your claim?**
- **It's easier to be complicated**
  - **Also: to publish it must be different; N+1st incremental change**
- **If it were not unsimple then how could distinguished colleagues in departments around the world be positively appreciative of both your extraordinary intellectual grasp of the nuances of issues as well as the depth of your contribution?**

# Bad Career Move #3: Never be Proven Wrong

- Avoid Implementing
- Avoid Quantitative Experiments
  - If you've got good intuition, who needs experiments?
  - Why give grist for critics' mill?
  - Plus, it takes too long to measure
- Avoid Benchmarks
- Projects whose payoff is ≥ 20 years gives you 19 safe years

# Bad Career Move #4:
# Use the Computer Scientific Method

**Obsolete Scientific Method**

- Hypothesis
- Sequence of experiments
- Change 1 parameter/exp.
- Prove/Disprove Hypothesis
- Document for others to reproduce results

**Computer Scientific Method**

- Hunch
- 1 experiment & change all parameters
- Discard if doesn't support hunch
- Why waste time? We know this

# Bad Career Move #5:
# Don’t be Distracted by Others (Avoid Feedback)

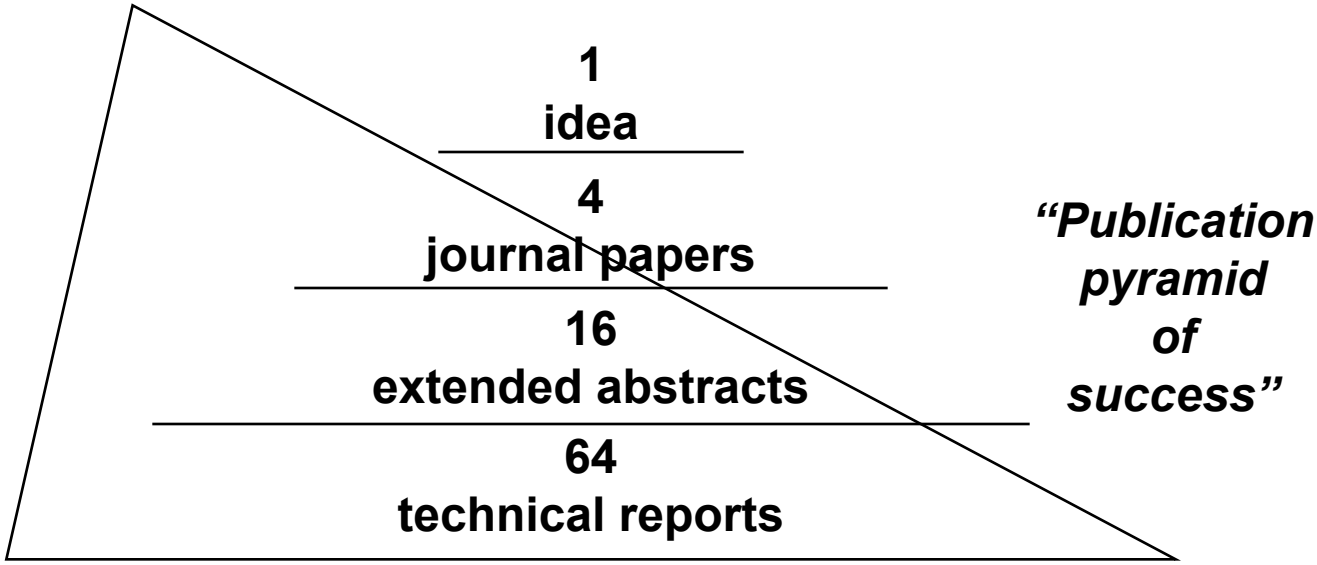
- Always dominate conversations: Silence is ignorance
  - Corollary: Louder is smarter
- Don’t read
- Don’t be tainted by interaction with users, industry
- Reviews
  - If it's simple and obvious in retrospect ⇒ Reject
  - Quantitative results don't matter if they just show you what you already know ⇒ Reject
  - Everything else ⇒ Reject

# Bad Career Move #6:
# Publishing Journal Papers IS Technology Transfer

- **As the leading scientist, your job is to publish in journals; its *not* your job to make you the ideas palatable to the ordinary engineer**
- **Going to conferences and visiting companies just uses up valuable research time**
  - **Travel time, having to interact with others, serve on program committees, ...**

# Bad Career Move #7: Writing Tactics for a Bad Career

- **Papers: It's Quantity, not Quality**
  - **Personal Success = Length of Publication List**
  - **"The LPU (Least Publishable Unit) is Good for You"**




- **Student productivity = number of papers**
  - **Never ask students to implement: reduces papers**
  - **Number of students: big is beautiful**
- **Legally change your name to Aaaanderson**

# Outline

- Part I: Key Advice for a Bad Career, Pre Ph.D.
- Part II: Key Advice for a Bad Career, Post Ph.D.
- Topics covered in Parts III, Alternatives to a Bad Career
  - Selecting a Problem
  - Picking a Solution
  - Performing the Research
  - Evaluating the Results
  - Communicating Results
  - Transferring Technology

# One Alternative Strategy to a Bad Career

- **Caveats:**
  - **From a project leader's point of view**
  - **Works for me; not the only way**
  - **Primarily from researcher, computer systems perspective**
- **Goal is to have impact:**
  ***Change way people do Computer Science & Engineering***
  - **Academics have bad benchmarks: number published papers**
  - **Richard Hamming: work on important problems!**
- **6 Steps**

  **1) Selecting a problem**

  **2) Picking a solution**

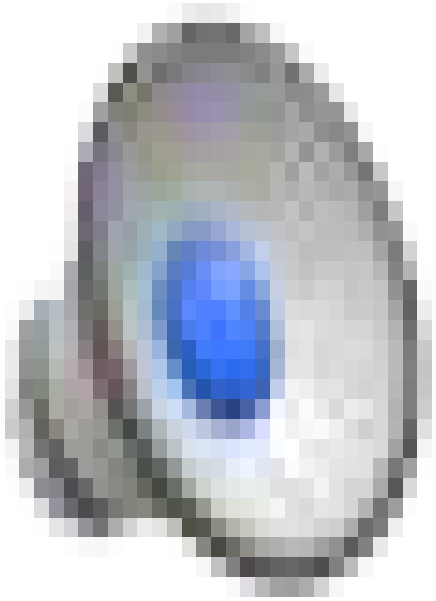
  **3) Running a project**

  **4) Finishing a project**

  **5) Quantitative Evaluation**

  **6) Transferring Technology**

# Hamming started having lunch with chemists at Bell Labs (after physicists got prizes and left or were promoted)



# 1) Selecting a Problem

## Invent a new field & stick to it?

- **No!** Do "Real Stuff": solve problem that <u>others</u> think Is important
  - Positive Impact on CS&E
- **No!** Use separate, short projects
  - Always takes longer than expected
  - Matches student lifetimes
  - Long effort in fast changing field???
  - Learning: Number of projects vs. calendar time
  - If going to fail, better to know soon
- Strive for multi-disciplinary, multiple investigator projects
- Match the strengths and weaknesses of local environment
- Make sure you are excited enough to work on it for 5 years
  - Prototypes help

## My first project

- Multiprocessor project with 3 hardware faculty ("Xtree")
- 1977: Design our own instruction set, microprocessor, interconnection topology, routing, boards, systems, operating system
- Unblemished Experience:
  - none in VLSI
  - none in microprocessors
  - none in networking
  - none in operating systems
- Unblemished Resources:
  - No staff
  - No dedicated computer (used shared department PDP-11/70)
  - No CAD tools
  - No applications
  - No funding
- Results: 2 journal papers, 12 conference papers, 20 TRs
- Impact?

# 2) Picking a solution

## Let Complexity Be Your Guide?

- **No!** Keep things simple unless a very good reason not to
  - Pick innovation points carefully, and be compatible everywhere else (spend intelligence beans carefully)
  - Best results are obvious in retrospect "Anyone could have thought of that"
- Complexity cost is in longer design, construction, test, and debug
  - Fast changing field + delays ⇒ less impressive results

## Use the Computer Scientific Method?

- **No!** Run experiments to discover real problems
- Use intuition to ask questions, not to answer them (Ousterhout)

## (And Pick A Good Name!)

**Reduced**
**Instruction**
**Set**
**Computers**

**Redundant**
**Array of**
**Inexpensive**
**Disks**

**Network**
**Of**
**Workstations**

**…**

# How do we pick project problem and solution?

- Start meeting with faculty at least 1 year in advance to discuss ideas
- Track interesting technology trends over next 5-10 years, to see if some new opportunity
  - RISC: VLSI Design, Moore's Law, 32-bit microprocessor
  - RAID: 5.25" disks for PCs, low I/O performance
  - NOW: Local Area Network Switches, Powerful Workstations
- Team of multidisciplinary faculty to see if want to volunteer to take on new challenge
- Get feedback on potential problem and solution from outsiders whose taste you trust, and iterate on vision
  - Industry unlikely to compete with our project, so safer

# 3) Running a project


## Avoid Feedback?

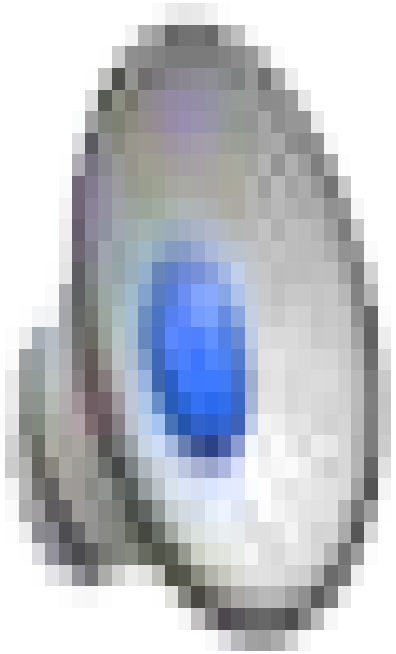
- **No!** Periodic Project Reviews with Outsiders
  - Twice a year: 3-day retreat
  - faculty, students, staff + guests
  - Key piece is feedback at end
  - Helps create deadlines, team spirit
  - Give students chance to give many talks / posters to interact with others industry
- Consider mid-course correction
  - Fast changing field & 5 year projects ⇒ assumptions changed
- Pick size and members of team carefully
  - Tough personalities are hard for everyone
  - 1 expert per area reduces chance of disagreement

# 3) Running a project

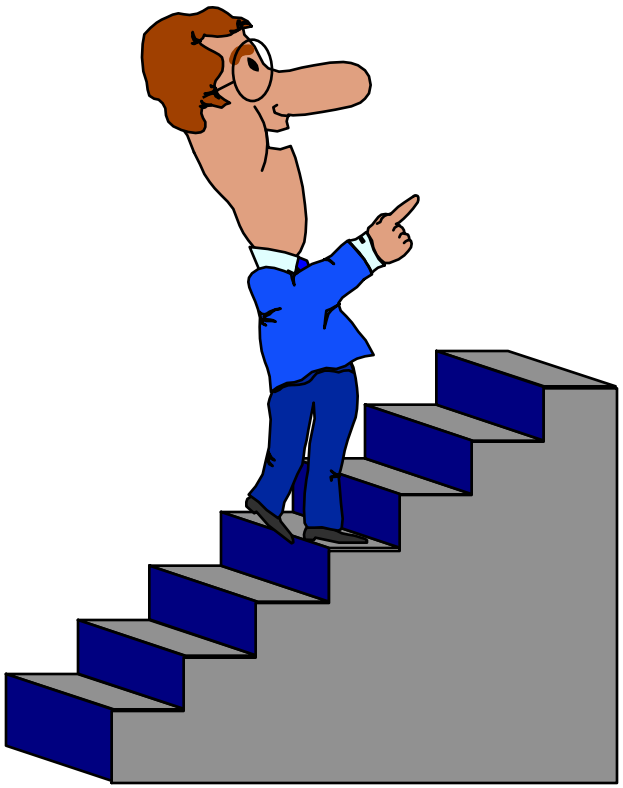
## Don't be Distracted by Others?

- **No!** Open Collaborative Laboratory
  - Avoid DSL Desert (work at home)
  - Faculty, students, staff in open space
  - Aim for Communication *and* Concentration
  - Optimized meeting rooms for discussions and phone calls
  - Kitchen, free drinks & coffee
- Accelerates research!
  - People come in more
  - Leads to spontaneous meetings
  - Improves 0 to 60 MPH time of new grad students
- Hamming on importance of Open Space and Feedback

# Hamming on Doors Open vs. Door Closed at Bell Labs

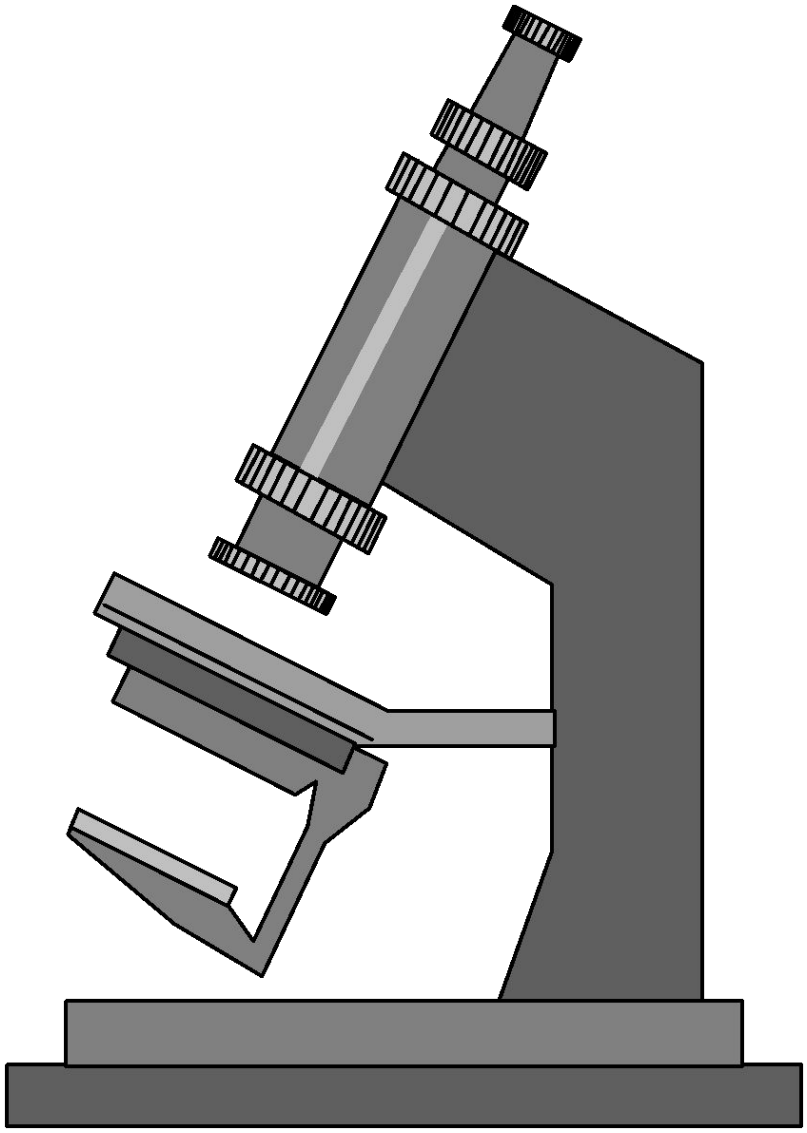


# 4) Finishing a project



- ***People count projects you finish, not the ones you start***
- ***Successful projects*** **go through an unglamorous, hard phase**
- **Design is more fun than making it work**
  - **"No winners on a losing team; no losers on a winning team."**
  - **"You can quickly tell whether or not the authors have ever built something and made it work."**
- **Reduce the project if its late**
  - **"Adding people to a late project makes it later."**
- **Finishing a project is how people acquire taste in selecting good problems, finding simple solutions**

# 5) Evaluating Quantitatively



## Never be Proven Wrong?

- **No!** If you can't be proven wrong, then you can't prove you're right
- Report in sufficient detail for others to reproduce results
  - can't convince others if they can't get same results
- For better or for worse, benchmarks shape a field
- Good ones accelerate progress
  - good target for development
- Bad benchmarks hurt progress
  - help real users vs. help sales?

# 6) Transferring Technology

## Publishing Journal Papers IS Technology Transfer?

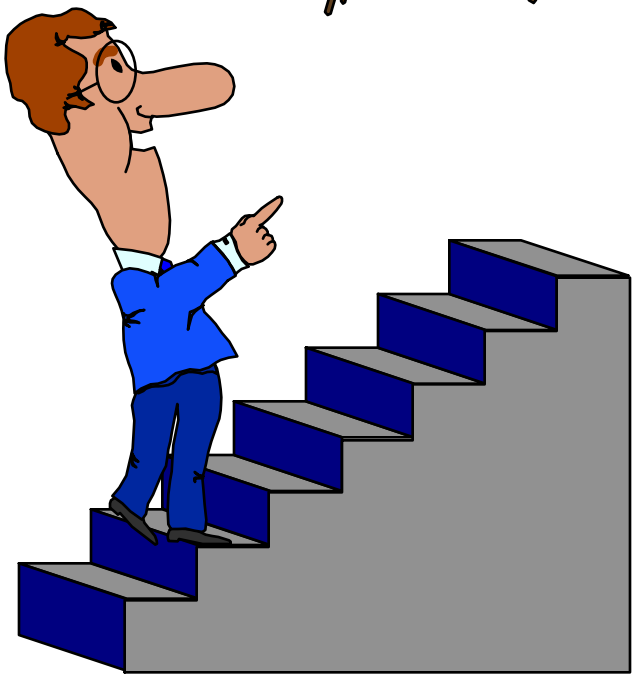
- **No!** Missionary work: "Sermons" first, then they read papers
  - Selecting problem is key: "Real stuff"
    - » Ideally, more interest as time passes
    - » Change minds with believable numbers
    - » Prima Donnas interfere with transfer
- My experience: industry is reluctant to embrace change
  - Howard Aiken, circa 1950:

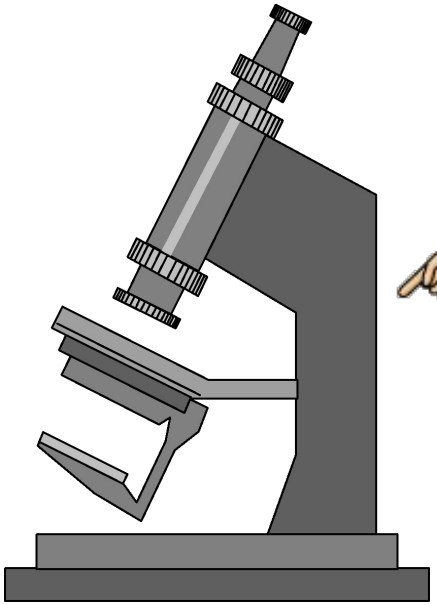
    *"The problem in this business isn't to keep people from stealing your ideas; its **making** them steal your ideas!"*
  - Need 1 bold company (often not no. 1) to take chance ***and*** be successful
    - » RISC with Sun, RAID with (EMC, …), NOW with (Inktomi, Google…)
  - Then rest of industry must follow

# 6) Transferring Technology: A Start up

- **Pros**
  - **Everyone enjoys trying once**
  - **Learn a lot**
  - **Personal satisfaction: seeing your product used by others**
  - **Personal $$$ (potentially)**
  - **Fame**
- **Cons**
  - **Learn about business plans, sales vs. marketing, financing, personnel benefits, hiring, lawsuits …**
  - **Spend time doing above vs. research/development**
  - **Only 10% of startups really make it**

# Summary: Leader's Role Changes during Project



# Richard Hamming's Advice: "You and Your Research" (AKA "You and Your Engineering Career")

- *(YouTube 1995 talk: www.youtube.com/watch?v=a1zDuOPkMSw)*
- Try a life of doing something significant, since only get 1 life to live
  - If you what you are working on is not important or not likely to be important, why are you working on it?
- Luck? "Luck favors the prepared mind" Pasteur
- Courage/Confidence: "I ain't scared of nothing" Shannon
- Work Hard: "Genius is 99% perspiration, 1% inspiration" Edison
- Open doors (vs. closed offices): short term vs. long term benefit
  - Door closed people worked on slightly wrong problem (e.g. IAS)
- Limited resources can help: can use creatively to lead to original solutions
- Selling the work: not only publish paper, but people must read it
  - as much work spent on polish and presentation as on the work itself

# "Technology And Courage" Ivan Sutherland, 1996 (search on citeseer for PDF)

- **Courage: to perceive risk and proceed in spite of it**
  - **Technology: risks to reputation and pride**
  - **Research: high probability that an attempt will fail**
  - **If inadequate courage, work up courage, reduce risk, reduce perception of risk, or don't do it**

1. **External Encouragement (rewards and punishment)**
   - **Deadlines, groups of people, mentors, seminars, tenure, taking / teaching classes, starting companies, stock**
2. **Self Encouragement**
   - **Getting started: warm-up project, break into tasks and do 1st one**
   - **To continue: refuse to let urgent drive out the important**
3. **Rewards**
   - **Thrill of discovery, following curiosity, beauty, simplicity**
   - **Personal joy of playing with technology**

# Conclusion: Alternatives to a Bad Career

- **Goal is to have impact:**
  ***Change way people do Computer Science & Engineering***
  - **Many 5 year projects gives more chances for impact**

# My 12 Five-Year Projects

| *Years* | *Project Title (Impact)* | *Faculty **(NAE in Bold)*** | *Students (ACM fellows)* |
|---|---|---|---|
| 1977-1981 | X-Tree: A Tree-Structured Multiprocessor | Despain, **Patterson**, Sequin | 12 (2) |
| 1980-1984 | Reduced Instruction Set Computer (RISC-I, RISC-II) | **Patterson, Ousterhout**, Sequin | 17 (1) |
| 1983-1986 | SOAR: Smalltalk On A RISC<br>aka "RISC-III" | **Patterson, Ousterhout** | 22 (1) |
| 1985-1989 | SPUR: Symbolic Processing Using RISCs<br>aka "RISC-IV" | **Patterson**, Fateman, Hilfinger, **Hodges**, **Katz**, **Ousterhout** | 21 (4) |
| 1988-1992 | Redundant Array of Inexpensive Disks (RAID) | **Katz**, **Ousterhout**, **Patterson**, **Stonebraker** | 16 (4) |
| 1993-1998 | NOW: Network of Workstations (Internet Clusters) | **Culler**, Anderson, **Brewer**, **Patterson** | 25 (2) |
| 1997-2002 | IRAM: Intelligent RAM | **Patterson**, Kubiatowicz, Wawrzynek, Yelick | 12 |
| 2001-2005 | ROC: Recovery Oriented Computing Systems | **Patterson**, Fox | 11 |
| 2005-2011 | RAD Lab: Reliable Adaptive Distributed Computing Lab (Spark, Mesos) | **Patterson**, Fox, **Jordan**, Joseph, **Katz**, **Shenker**, Stoica | 45 |
| 2007-2013 | Par Lab: Parallel Computing Lab<br>(Communication Avoiding Algorithms, RISC-V) | **Patterson**, Asanovic, **Demmel**, Fox, Keutzer, Kubiatowicz, Sen, Yelick | 36 |
| 2011-2016 | AMP Lab: Algorithms, Machines, & People | Franklin, **Jordan**, Joseph, **Katz**, **Patterson**, **Shenker**, Stoica | 40 |
| 2012-2017 | ASPIRE Lab: Algorithms and Specializers for Provably optimal Implementations with Resilience and Efficiency | Asanovic, Alon, Bachrach, **Demmel**, Fox, Keutzer, Nikolic, **Patterson**, Sen, Wawrzynek | 40 |

**27 (10 NAE, 15 total in CS) 297 (14 ACM)**

# Conclusion: Alternatives to a Bad Career
## https://goo.gl/ChRlHT

- **Goal is to have impact:**
  ***Change way people do Computer Science & Engineering***
  - **Many 5 year projects gives more chances for impact**
- **Do "Real Stuff": make sure you are solving a problem that others think is important**
- **Key is getting good feedback and listening to it**
- **Taste is critical in selecting research problems, solutions, experiments, and communicating results;**
  - **Taste acquired from feedback and completing projects**
- **Faculty real legacy is people you produce, not papers:**
  - **Expected from reading 1974 book *Working: People talk about what they do all day and how they feel about what they do***
  - **Create environments that develop PhDs of whom proud**
- ***Students* are the coin of the academic realm**

# My Story: Accidental Academic

- 1st college graduate in family; no CS/grad school plan
  - Wrestler, Math major in high school and college
- Accidental UCLA PhD student
  - New UCLA PhD (Jean-Loup Baer) took pity on undergrad
- Wife + 2 sons in Married Students Housing while grad student
  - Lost RA-ship after ≈4 years because grant ended
  - Part time at Hughes Aircraft Company ≈3 more years
- Accidental Berkeley Professor
  - Wife forced me to call UC Berkeley CS Chair to check on application
- 1st project as Assistant Prof with an Associate Prof too ambitious & no resources
  - Took leave to DEC to rethink career in 3rd year
- Tenure not easy (Conference vs. journal, RISC too recent)
- Still get papers rejected by jerks on Program Committee

# What Works for Me

- Maximize Personal Happiness vs. Personal Wealth
- Family First!
- Passion, Optimism, & Courage
  - Swing for the fences vs. Bunt for singles
  - "Friends may come and go, but enemies accumulate"
- Winning as Team vs. Winning as Individual
  - "No losers on a winning team, no winners on a losing team"
- Seek Out Honest Feedback & Learn From It
  - Reliable Danger Sign: "I'm smartest person in the room"
- One (Big) Thing at a Time
  - It's not how many projects you start;
    It's how many you finish
- Have Fun: Work Hard, Play Hard