How to Have a Bad Career

David Patterson
Google & UC Berkeley
December, 2021

Outline

- Part I How to Have Bad Career
- Part II How to Avoid a Bad Career
  - + Richard Hamming (Turing Laureate for error-detecting and error-correcting codes) video clips from “You and Your Research”
- Q&A
- 5 books to help jump start a career
- My Story: Accidental Academic (5-7 min)
- What Work(ed) for Me (5-7 min)

Bad Career Commandment #1: Be THE leading expert

- Invent a new field!
  - Make sure it’s 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*
- Stick to one topic for whole career
  - Even if technology appears to leave you behind, stand by your problem

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

- Best compliment:
  "It's so complicated, I can't understand the ideas"
- It's easier to be complicated
- Also easier to claim credit for subsequent good ideas
  - If no one understands, how can they contradict your claim?

Bad Career Commandment #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

Bad Career Commandment #4: Use the <u>Computer</u> 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 data if don’t support hunch
- Why waste time? We know this

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

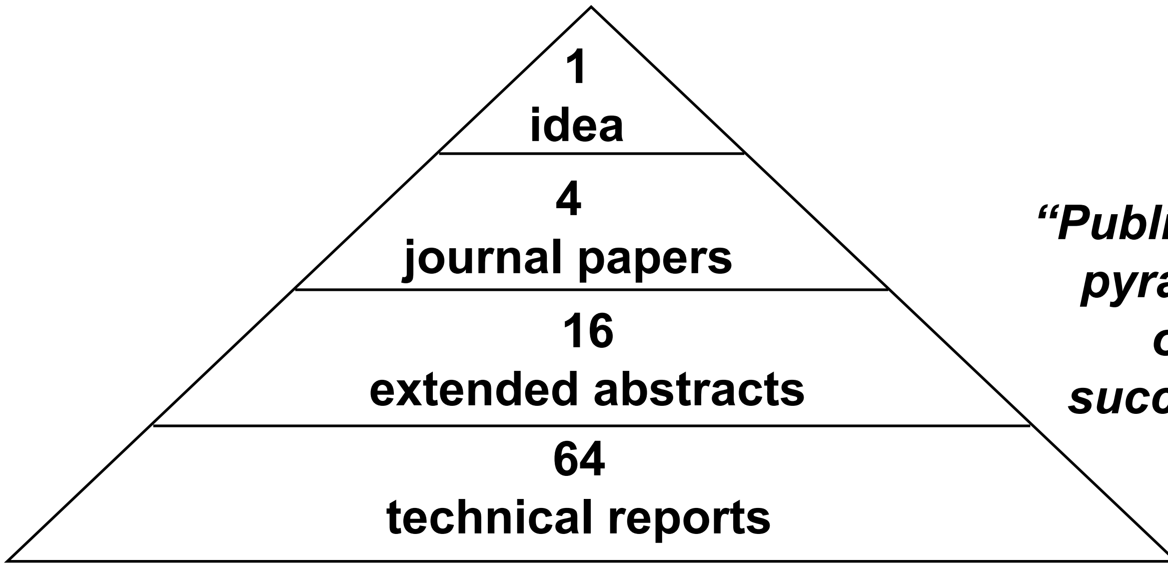
- **Always dominate conversations: Silence is ignorance**
  - **Corollary: Louder is smarter**
- **Don’t be tainted by interaction with users**

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

- **As the leading expert, your job is to publish in journals; it's *not* your job to make your ideas palatable to the ordinary engineer**
- **Going to conferences and visiting companies just uses up valuable writing time**

Bad Career Commandment #7: Publication: It's Quantity, not Quality

- **Personal Success = Length of Publication List**
  - **"The LPU (Least Publishable Unit) is Good for You"**



"Publication pyramid of success"

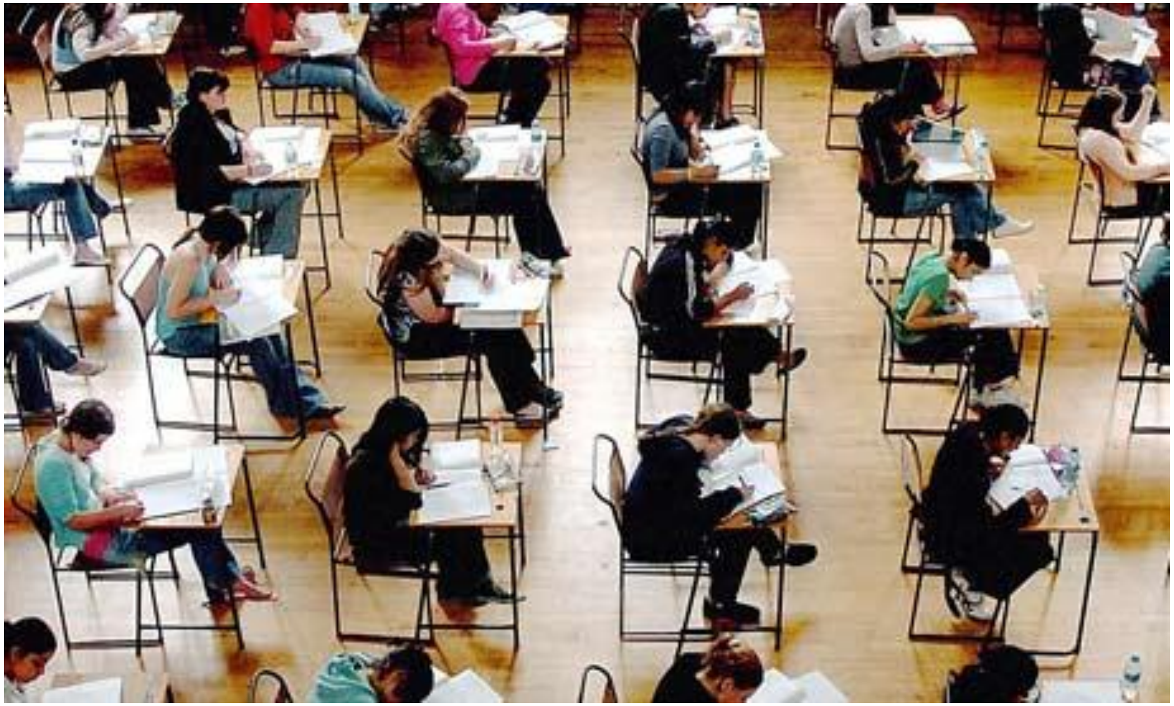
- **Legally change your name to Aaaanderson**

Bad Career Commandment #7: Publication: It's Quantity, not Quality

- How to get papers accepted by journals or conferences?
- Problem: Conferences accept just 1/N papers (N ≈ 5 - 10)
  - Rumor is reviewing papers are random decisions
- Solution: *RIP - Redundant Incremental Papers*
- To ensure publishing given random reviews with a chance 1/N, submit N RIPs
  ⇒ chances now 100%!
- Good news: After first round, have a head start for next deadlines as have N-1 rejected papers as RIPs

Bad Career Commandment #7: Publication: It's Quantity, not Quality

- But who can write N RIPs?
- RAIG: Redundant Array of Inexpensive Graduate students! ("rage")
  - N students each write 1 RIP



7 Commandments for a Bad Career

I. **Be THE leading expert**
II. **Let Complexity Be Your Guide (Confuse Thine Enemies)**
III. **Never be Proven Wrong**
IV. **Use the <u>Computer</u> Scientific Method**
V. **Don't be Distracted by Others (Avoid Feedback)**
VI. **Publishing Journal Papers IS Technology Transfer**
VII. **Publication: It's Quantity, not Quality**

Avoid Being by a Bad Career in Research

- Works for me; not the only way
- Primarily from computer systems perspective
- Goal is to have impact on society or other researchers
- 6 Steps for Research

  1) Selecting a problem
     - Richard Hamming: “Work on important problems!”*
  2) Picking a solution
  3) Running a project
  4) Finishing a project
  5) Quantitative Evaluation
  6) Transferring Technology

******The Art of doing science and engineering: Learning to learn***. CRC Press, 2003 (and corresponding video lectures)**

Hamming on value of picking important problems (He 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, 3-5 year projects
  - Long effort in fast changing field??
  - It's research, so may not succeed
  - Matches grad student "lifetimes"
  - Easier for find leader for 3-5 years
- Strive for multi-disciplinary, multiple investigator projects

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 <u>Computer</u> Scientific Method?

- No! Run experiments to discover real problems
- "Use intuition to <u>ask</u> questions, <u>not</u> to <u>answer</u> them"

(John Ousterhout)

How picked project problem and solution?

- Start meeting with faculty at least 1 year in advance to discuss ideas for an exciting vision
  - To excite grad students and nurture new faculty
- 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: Small disks for PCs, low I/O performance
  - NOW: Local Area Network Switches, Powerful Workstations
- Multiple-discipline faculty to see who wants to volunteer to form team take on new challenge
- Get feedback on potential problem and solution from outsiders whose taste you trust, and iterate on vision

(And Pick A Good Name!)

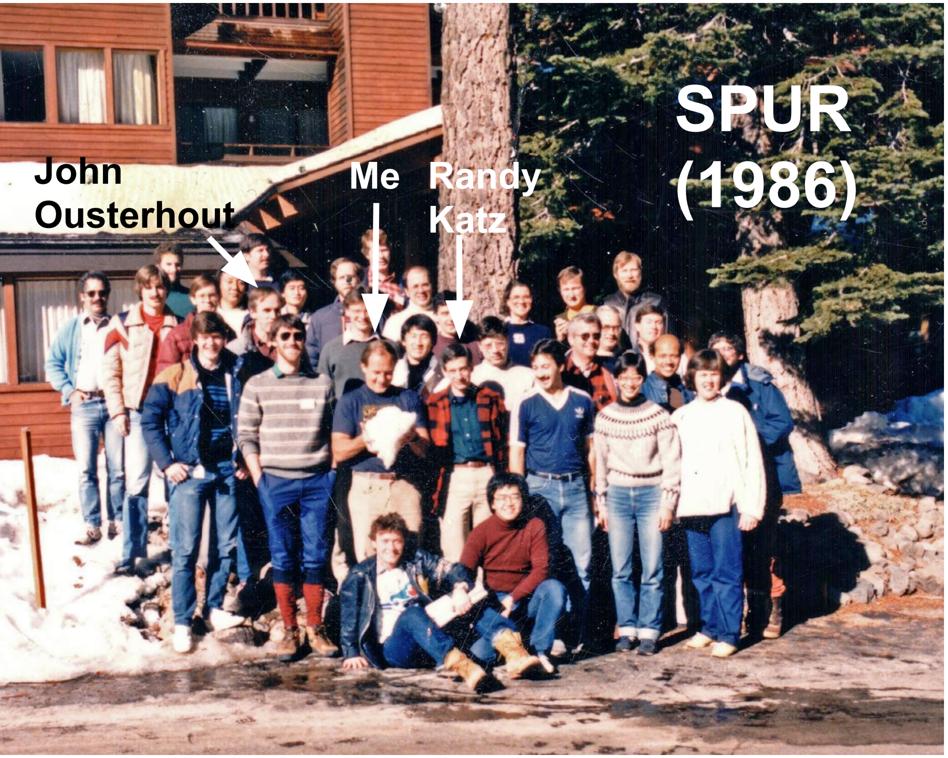
Reduced
I nstruction
Set
Computers

Redundant
Array of
I nexpensive
Disks

Network
Of
Workstations

...

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 outsiders
- Consider mid-course correction
  - Fast changing field & 5 year projects ⇒ assumptions changed
- Pick size and members of team carefully
  - 1 expert per area reduces chance of disagreement
  - Tough personalities are hard for everyone

3) Running a project

Don't be Distracted by Others?

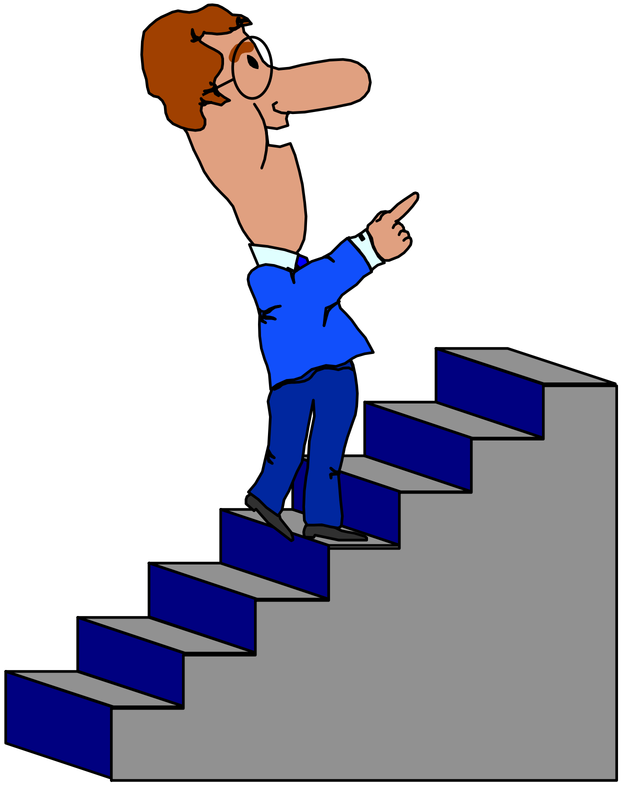
- No! Open Collaborative Laboratory
  - Avoid DSL Desert (work at home)
  - Faculty, students, staff in open space
  - Aim for Communication *and* Concentration
  - Optimized meeting rooms for discussions and phone calls
  - Kitchen with free drinks & coffee
- Accelerates research!
  - People come in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

RAD Lab first CS Open Colaboratory in in 2006

- **Hamming on importance of Open Space and Feedback**

Hamming on Doors Open vs. Door Closed at Bell Labs

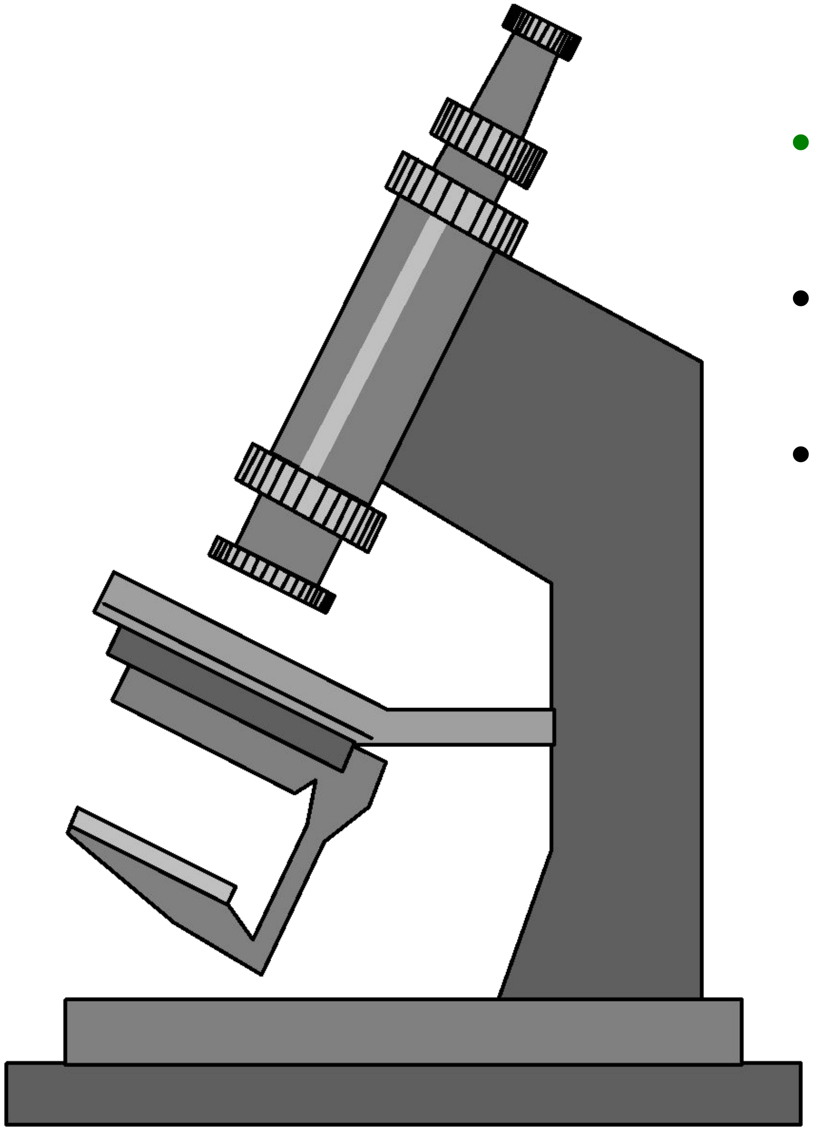
4) Finishing a project



- ***It's not how many projects you start, it's how many you finish***
  - *Successful projects* go through an unglamorous, hard phase
    - Design more fun v. making work
  - "No winners on a losing team; no losers on a winning team" (Brooks)

- Reduce the project if it's late
  - "Adding people to a late project makes it later" (Brooks)
- Finishing a project is how people acquire taste in selecting good problems, finding simple solutions
  - "You can quickly tell whether or not the authors have ever built something and made it work" (John Hennessy)

5) Evaluating Quantitatively



Never be Proven Wrong?

- **No!** If you can't be proven wrong, you can't proven right
- Report in sufficient detail for others to reproduce results
- For better or for worse, benchmarks shape a field

6) Transferring Technology

Publishing Journal Papers IS Technology Transfer?

- **No!** **Missionary work: "Sermons" first, then they read papers**
  - **Change minds with believable numbers**

My experience: product groups are reluctant to embrace change:

"The problem in this business isn't to keep people from stealing your ideas; it's making them steal your ideas!"

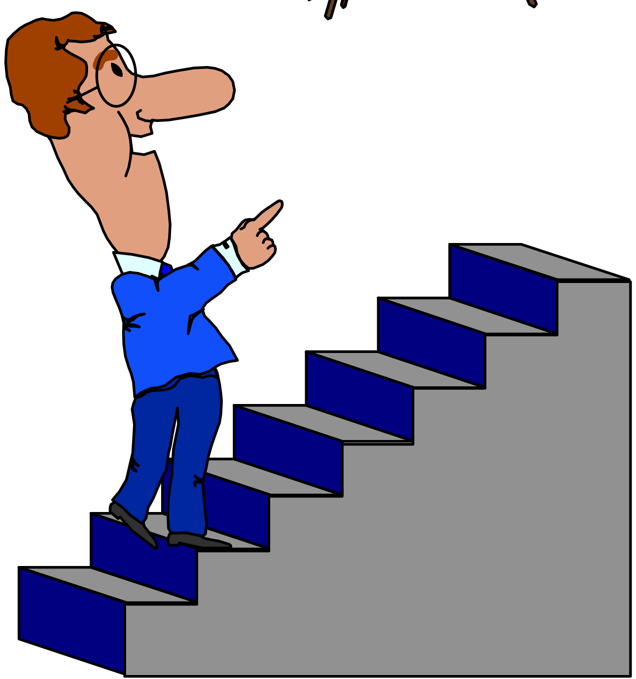
(Howard Aiken, 1950)

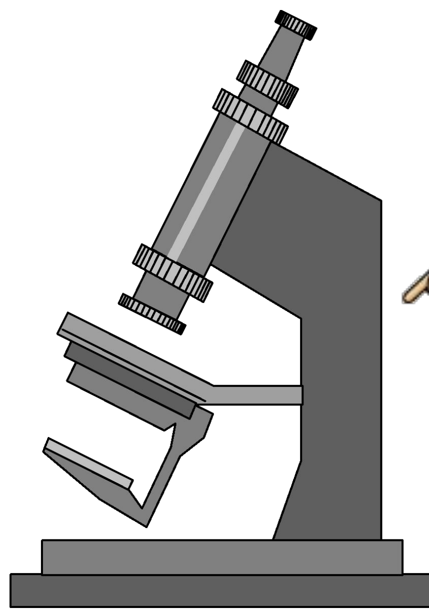
- **Need 1 bold group (often not no. 1) to take chance *and* be successful, then rest of industry must follow**
- **RISC with Sun, RAID with (EMC, …), NOW with (Inktomi, Google…)**

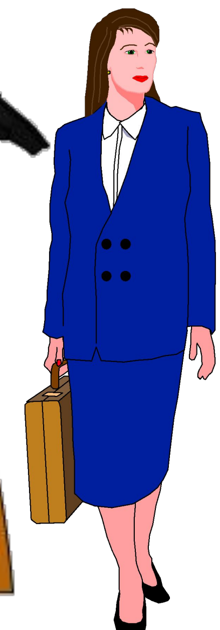
6) Transferring Technology: A Start up

- Pros
  - Everyone enjoys trying it once
  - Learn a lot
  - Personal satisfaction: seeing your product used by others
  - Personal $$$ (potentially)
  - Fame
- Cons
  - Learn about business plans, sales vs. marketing, financing, hiring, lawsuits …
  - But only 10% startups make it

Summary: Leader's Role Changes during Project







Conclusion: Alternatives to a Bad Career

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

My 13 Five-Year Projects

Years	Project Title (Impact)	Faculty Director, PIs ***(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 aka "RISC-III" (*Generational Garbage Collection*)	**Patterson, Ousterhout**	22 (1)
1985-1989	SPUR: Symbolic Processing Using RISCs aka "RISC-IV" (*Snoopy bus protocols*)	**Patterson**, Fateman, Hilfinger, **Hodges**, **Katz**, **Ousterhout**	21 (4)
1988-1992	**RAID: Redundant Array of Inexpensive Disks** *(RAID-I, RAID-II)*	**Katz**, **Ousterhout**, **Patterson**, **Stonebraker**	16 (4)
1993-1998	NOW: **Network of Workstations** *(Inktomi, Internet Clusters)*	**Culler**, **Anderson**, **Brewer**, **Patterson**	25 (4)
1997-2002	IRAM: Intelligent RAM (*Processor in Memory*)	**Patterson**, Kubiatowicz, Wawrzynek, **Yelick**	12 (2)
2001-2005	ROC: Recovery Oriented Computing Systems (*Crash-only software, Microreboot*)	**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 *(Communication Avoiding Algorithms, **RISC-V**)*	**Patterson**, Asanovic, **Demmel**, Fox, Keutzer, Kubiatowicz, Sen, **Yelick**	36
2011-2016	AMPLab: Algorithms, Machines, & People	Franklin, **Jordan**, Joseph, **Katz**, **Patterson**, **Shenker**, Stoica	34
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	31
2017-2022	RISELab: Real-time Intelligent Secure Explainable systems	Stoica, Gonzalez, Hellerstein, Popa, **Jordan**, **Patterson**, **Katz**	≈35
		31 (12 NAE of 16 total NAE in CS)	**317 (18 ACM)**

David Patterson, "How to build a bad research center," *Communications of the ACM*, 57.3 (2014): 33-36.

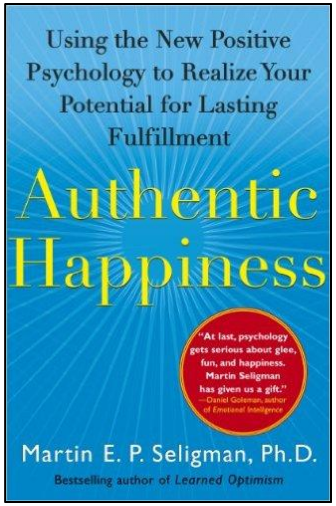
Conclusion: Alternatives to a Bad Career

- **Goal is to have impact:**
  Change way people do Computer Science & Engineering
  - **Many limited 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, *finishing* projects, & communicating results;**
  - **Taste acquired from honest feedback and completing projects**
- **It's people you work with, not the papers you produce**
  - ***Students* are the coin of the academic realm**

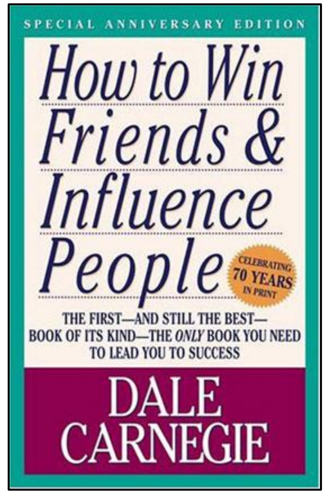
Hamming Book Last Paragraph, First Chapter

"Lastly, in a sense, this is a religious course—I am preaching the message that, with apparently only one life to live on this earth, you ought to try to make significant contributions to humanity rather than just get along through life comfortably—**that the life of trying to achieve excellence in some area is in itself a worthy goal for your life**. It has often been observed the true gain is in the struggle and not in the achievement—a life without a struggle on your part to make yourself excellent is hardly a life worth living. This, it must be observed, is an opinion and not a fact, but it is based on observing many people's lives and speculating on their total happiness rather than the moment to moment pleasures they enjoyed. Again, this opinion of their happiness must be my own interpretation as no one can know another's life. Many reports by people who have written about the "good life" agree with the above opinion. Notice I leave it to you to pick your goals of excellence, but claim only a life without such a goal is not really living but it is merely existing—in my opinion. In ancient Greece Socrates (469–399) said: The unexamined life is not worth living."

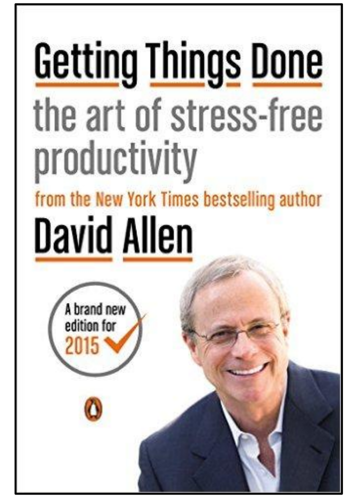
5 Slim Books to Jump-Start a New Career



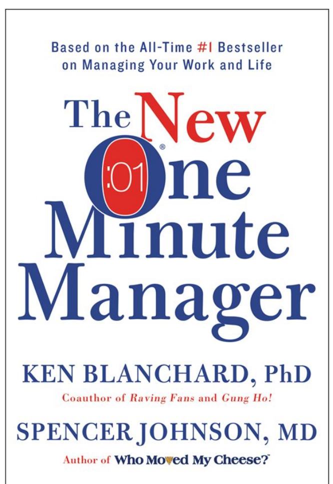
How to Be Happy
(2004, 336 pages)



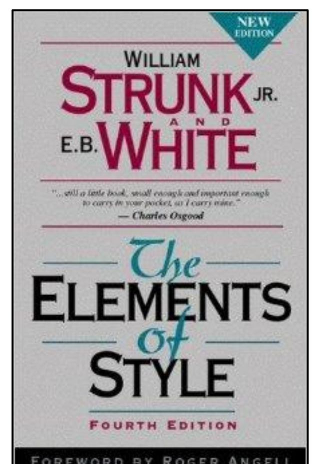
How to Lead and Persuade
(1998, 288 pages)



How to Be Productive
(2015, 317 pages)



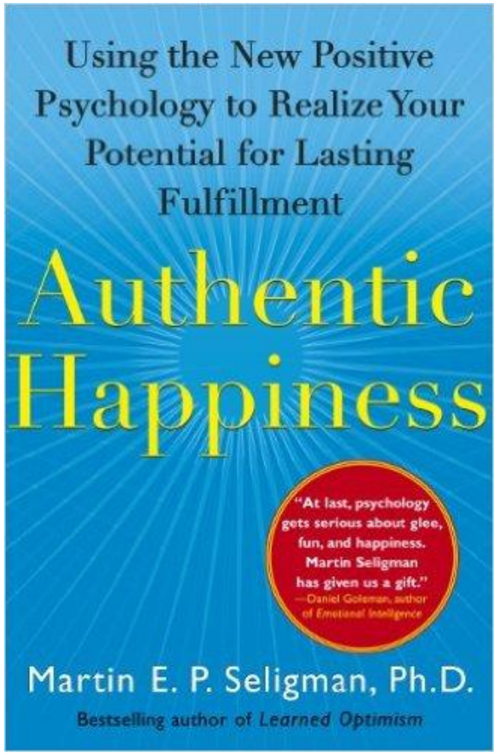
How to Manage Up and Down
(2015, 112 pages)



How to Write Well
(1999, 105 pages)

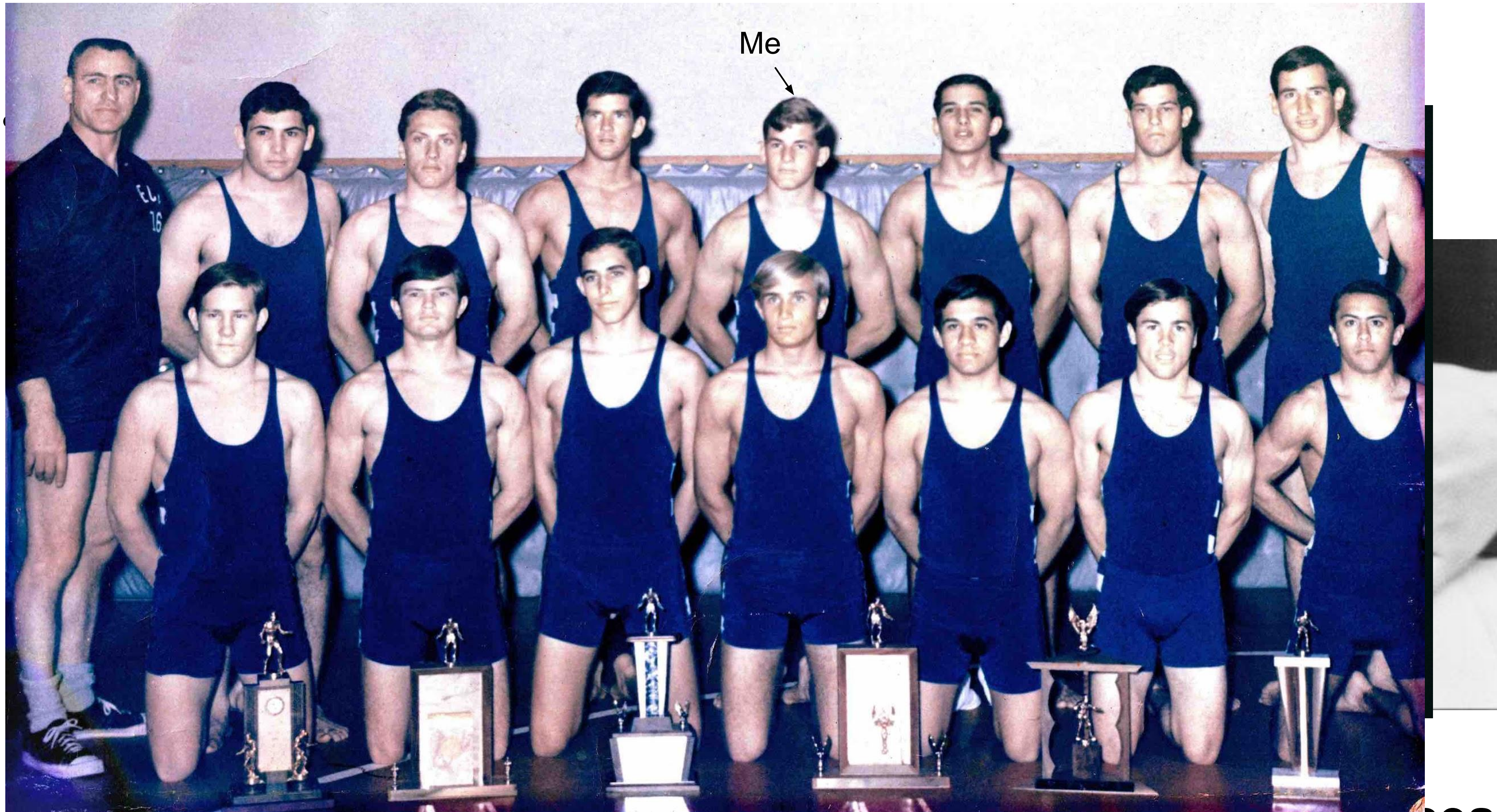
Psychologists now Study Happiness

- Have a job you love
- Family & Friends
- Have time to play
- Doing something for others
- Have a spiritual side



My Story: Accidental Academic

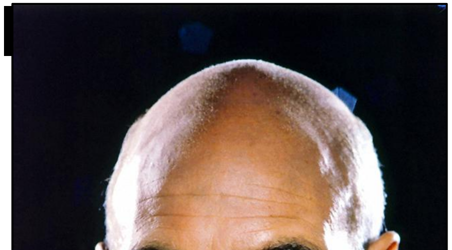
- **Accidental CS student**

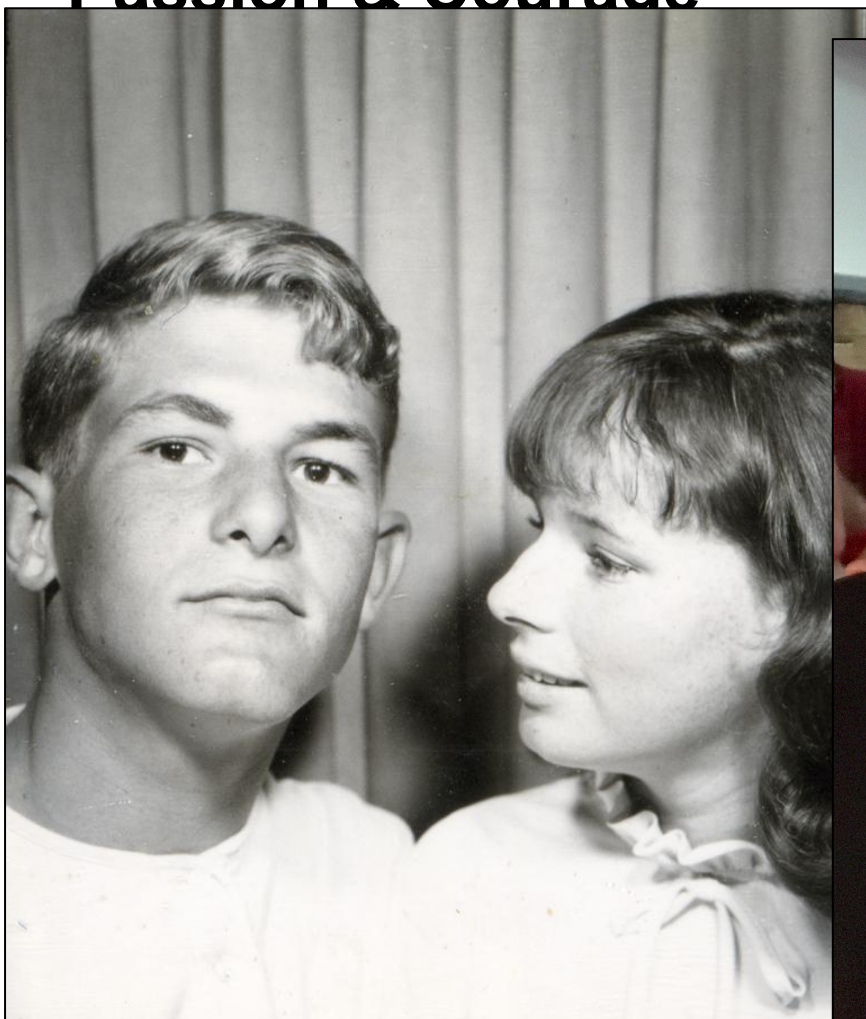


- **1st project led by Assoc. Prof too ambitious, no resources**
  - **Took leave to DEC to rethink career in 3rd year**

What Works/Worked for Me

- Maximize Happiness vs Weal
- Family First!
- Passion & Courage






9 Magic Words for a Long Relationship (for both partners)

"I ___ ___."

"You ___ ___."

"I ___ You."

9 Magic Words for a Long Relationship (for both partners)

"I Was Wrong."

"You Were Right."

"I Love You."