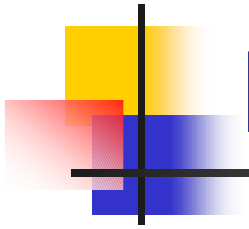
How to be a computer systems graduate student

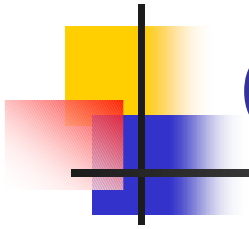
Richard Martin

10/2000



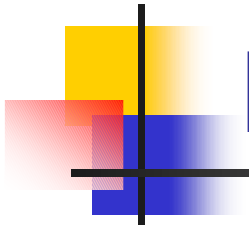
Motivation

- Not many resources to learn how to be a successful grad student
  - Easy to get lost
- This talk: a broad picture for success
  - Give food for thought on how to pursue grad school career



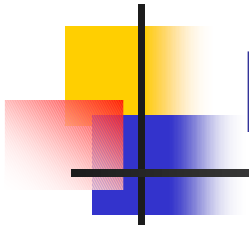
Outline

- Know Thyself
  - take ownership of your degree
- Picking a topic
- The grind: making it happen
  - Investigate
  - Explore
  - Evaluate
  - Getting resources
- Communicate your results



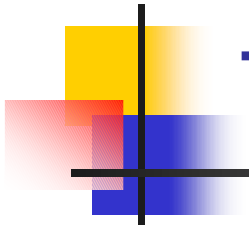
Know thyself

- Answer: why are YOU getting a PhD?
- Prerequisite to a research career
  - University/Industrial/Government labs
- Personal development
  - learn to write, speak, critical thinking
- Learn the “Art” of computer systems design, analysis and evaluation
  - Why is a system better? More enduring?



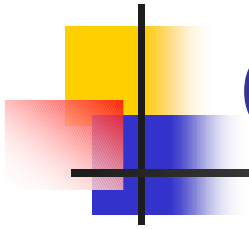
Bad reasons

- Nothing better to do
- F-1
- Job ticket
- Grad student lifestyle



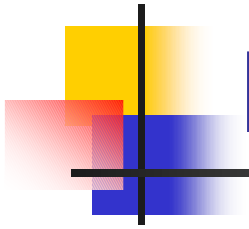
Take ownership

- No one is responsible for getting your degree but you.
  - Many obstacles
    - lack of space, equipment, advisor's time
- System researchers must work with others
  - Advisor, staff, other students
  - Output of focused group >> lone wolf
  - BUT pick a work style and lab culture that fits you



Outline

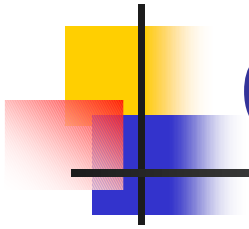
- Know Thyself
  - take ownership of your degree
- Picking a topic
- The grind: making it happen
  - Investigate
  - Explore
  - Evaluate
  - Getting resources
- Communicate your results



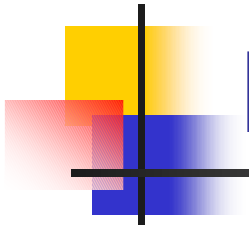
Picking a topic

- Always have a current research topic, ”proposition” or idea
- Start small
  - Ask advisor, senior students for ideas
- Later, come up with own
  - Part of a PhD is acquiring “taste”
  - Differentiate what’s cool from junk

The misnomer ---- Computer Science

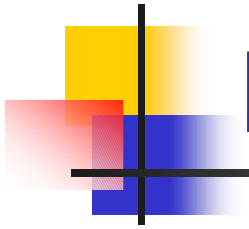


- Science (results research)
  - Evaluation and quantification of existing phenomena
  - modeling how the world works
- Engineering (idea research)
  - Building useful tools
- Systems research is a mix of both!
  - Know how your topic fits into ideas, results, or both



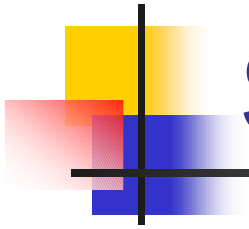
Results driving ideas

- Result: programs exhibit locality
  - ➔ Idea: cache
- Result: most programs are simple
  - ➔ Idea: RISC
- ➔ Result: Traffic is self-similar
  - ➔ Idea: ?



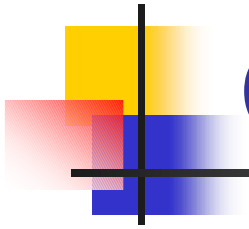
Picking a topic

- Have some religion about your idea or result
  - Will greatly help when the going gets tough
- Don’t worry about people stealing your ideas
  - Feedback from sharing >> cost of theft
  - Getting people interested in your idea will be much harder than getting them to “steal” it.
- Most ideas are dead ends, few endure
  - learn to discard bad ideas quickly
  - learn to recognize a great ideas



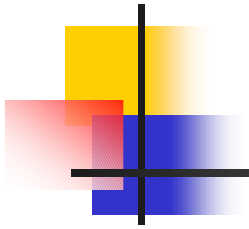
Stay on top of events

- Follow conferences
  - sosp, sigmod, isca, asplos, sigcomm ...
- Read trade rags
  - infoworld, slashdot.com, news.com, techweb.com, wired ...
- Learn who are the opinion leaders in your field, know what they are doing
  - But question the established order too ...



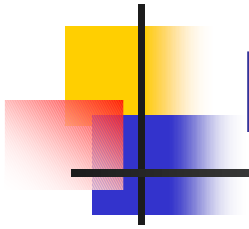
Outline

- Know Thyself
  - take ownership of your degree
- Picking a topic
- The grind: making it happen
  - Investigate
  - Explore
  - Evaluate
  - Getting resources
- Communicate your results



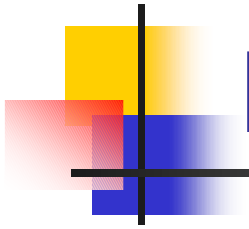
Investigate

- Has your idea been done before?
  - know what are in classic papers
- Why is your idea “better”?
- Why will your result be important?
- Who will care?
- Final impact if you’re successful?



Explore your idea

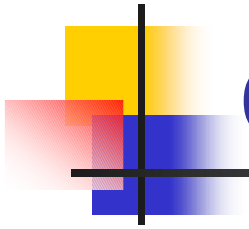
- 3 Approaches to systems research:
  - Build a prototype
    - To do right is very hard
    - Ultimate validation
  - Build a simulation
    - Not as hard but is it credible?
  - Build a measurement apparatus
- A sufficient analysis background is critical to all 3 approaches



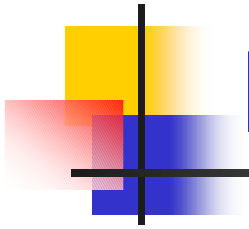
Evaluate your idea

- How is your idea better? Result novel?
- Measure it
  - Latency, throughout, fault tolerance
  - space (still an issue?)
  - usability, manageability (new!)
- Judgment on artistic merit
  - Is your result or idea exciting?
    E.g. Cray-1, Unix, Risc, Fortran, self-similarity

Scientific method vs computer scientific method

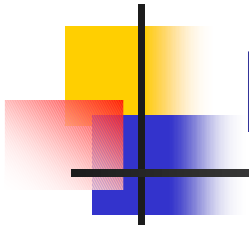


- Scientific method
  - Control 1 parameter at a time, observe results
- Computer scientific method
  - Change everything
  - If data doesn't fit your intuition, throw it away!



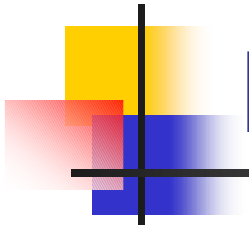
No magic

- What if it doesn't work?
- no magic, everything can be figured out
- Form a hypothesis
  - Cross-check with other evidence
  - test with a simple experiment
- Find who's done it before or built it and ask them



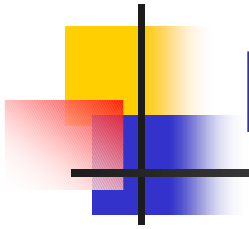
Newer evaluation points

- How will your idea mesh with the installed base?
  - Huge deployment costs?
- What are the switching costs over the current or obvious solutions?
  - Is your idea 10x better today?
- How will predictable technology advances impact your idea?
  - Will your idea be 10x better in 5 years?



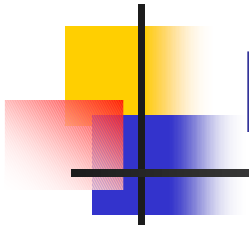
Resources

- So you have a great idea?
- Need resources!
  - Human cycles
    - Funding for you!
  - Space
  - Machine cycles
    - Equipment



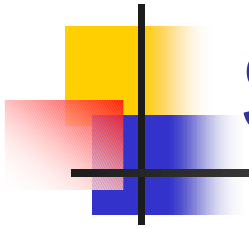
Funding

- Teaching Assistantship (T.A.)
  - Allow you to try grad. school
  - Ready to move on to ...
- Graduate Assistantship (G.A.)
  - also Research Assistant (R.A)
- Internships



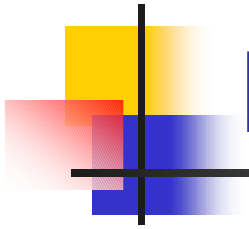
Funding Sources

- Multi-year grants
  - Defense Advanced Research Projects Agency (DARPA)
  - National Science Foundation(NSF)
- Year-to-year
  - Corporate (Cisco, IBM, Siemens, Intel ...)
  - USENIX (student applied)
  - Rutgers/NJ



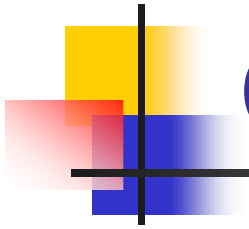
Space

- Lack of space a problem in many CS departments
- Find an Advisor
  - Hill 429 your home?
  - You'll get more disk space too!
    - 5 MB on Paul?



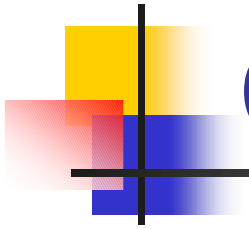
Equipment

- Find an Advisor
  - Scrounge for 90 Mhz mac?
- Ask your Advisor
  - If you have good reasons, advisor can work to make things appear
  - Corallary: you have to deliver!



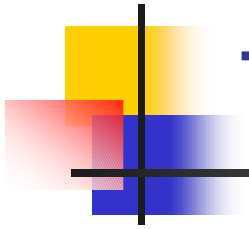
Outline

- Know Thyself
  - take ownership of your degree
- Picking a topic
- The grind: making it happen
  - Investigate
  - Explore
  - Evaluate
  - Getting resources
- Communicate your results



Communicate your ideas

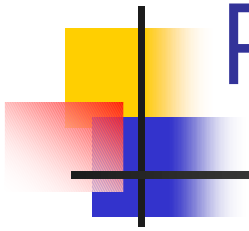
- Clear communication defines the top students from the average
  - Critically important since dawn of science
  - What's a brilliant result if unknown for 100 years?
- Talks
  - posters
- Papers
  - "If you didn't write it down, it didn't happen"
  - "Publish or perish"?
- Software releases



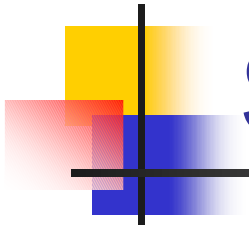
Talks: verbal communication

- “Ad” for your work
  - Good talk: people spread the word
  - A bad talk diverts people’s attention
- Practice, practice, practice
  - Use video, mirror
  - get feedback from advisor, other students, visit
  - go to talks, use good ones as models
- A poster is really a mini-talk
  - 10 minute one-on-one

Papers: written communication

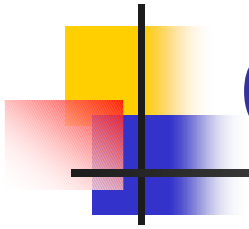


- 3 kinds:
  - Conference, Journal, Tech. Reports,
  - Class projects are good practice
- Start as soon as you're far enough along to communicate results!
  - Ask advisor for model papers
  - Get feedback on your paper



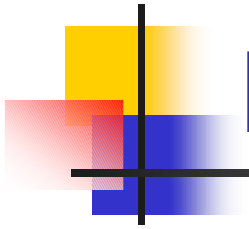
Software releases

- Software use can be the ultimate test of an idea
  - vote with their mice
  - E.g. Unix, X, mosaic,Tcl/Tk, Magic
- Pros:
  - feedback, fame, personal satisfaction
- Cons:
  - Support, documentation, upgrades, users, fame



Conclusion

- You can do systems research
  - Take control of your degree
- Tenacity is key
- Pick a topic you believe in and stick with it.



Further reading

How to Be a Good Graduate Student

by Marie desJardins

http://www.cs.indiana.edu/how.2b/how.2b.html

So long, and thanks for the Ph.D.!

by Ronald T. Azuma

http://www.cs.unc.edu/~azuma/hitch4.html