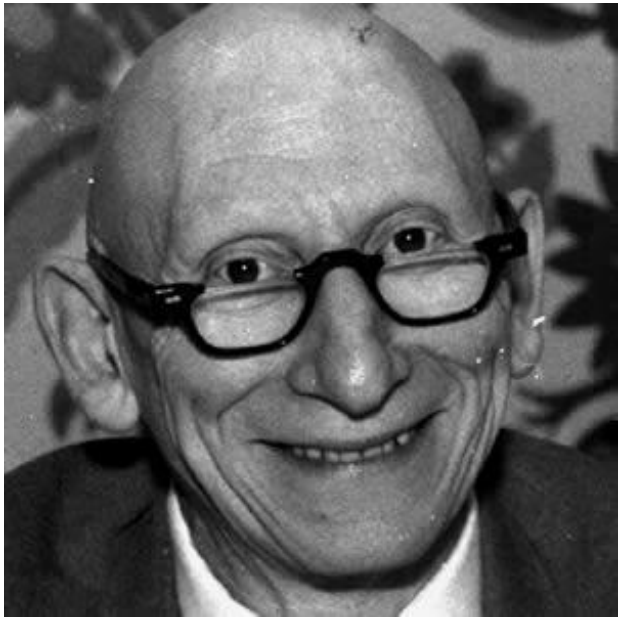
Keep the Fun in the House



We began to feel as if we really were responsible for the successful error-free perfect use of these machines.

I don’t think we are.

I think we’re responsible for stretching them setting them off in new directions and keeping fun in the house.

What’s in your hands I think and hope is intelligence: the ability to see the machine as more than when you were first led up to it, that you can make it more.

Alan J.Perlis

ACM A.M. Turing Award Winner 1966

Strategic Thinking for Researchers

Andy Gordon

@AndrewDGordon

MSR PhD Summer School, July 1–4, 2014

Based in part on the MSRC Impact Workshop 2010,
co-organised by Thore Graepel, with material by Bob Williamson

Strategic Thinking for Researchers

- You can write great papers, give great talks – what next?
- This talk: long-term goals and strategies
- If research is your life, learn strategy as well as execution
- No claim of originality
- No rigorous theory; no empirical evaluation; no usability study
- There is no one correct strategy, but there are commonalities
- I don’t even claim anyone follows all this advice all of the time
- I don’t feel especially qualified, except…

Background: Impact Workshop 2010

Goal

To gather as a community to consider the question:

How to have impact, through research?

We hope that the afternoon will:

- Make us stop and think how our work can change the world.
- Get all of us talking about how to do that.
- Generate tips for how we can set bigger goals, be more effective.

Organising it was so much fun it almost didn't feel like work…


MSR Speed
Dating Society
5.5.11

MSR Speed Dating Society 5.5.11

We believe social links lay foundations for eventual transfer of expertise and for serendipitous collaborations between groups, but the immediate goal is simply to spend 90 minutes surprising each other!

- **24 attendees: each had one minute "surprise on a slide"**
- **8 "speed dates" 5 mins each**
- **Followed by wine and nibbles**

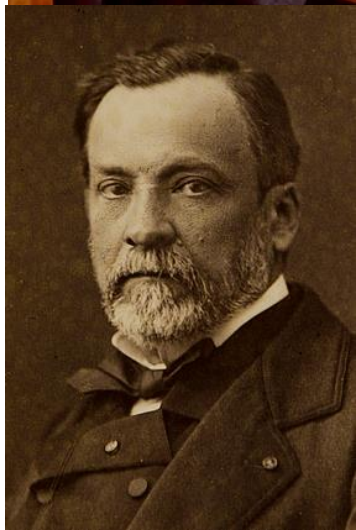
Know what you are trying to do

The most important single aspect of software development is to be clear about what you are trying to build.

– Bjarne Stroustrup

Research is what I'm doing when I don't know what I'm doing.

- Wernher Von Braun



- But you can know what you are *trying* to do
- A job interview question (2 parts):
  - What's the most important problem in your field?
  - What are *you* working on?
- Serendipity is real, *but it is not an excuse for not knowing what you are trying to do*
  - *Chance favours the prepared mind*

You've got to work on important problems

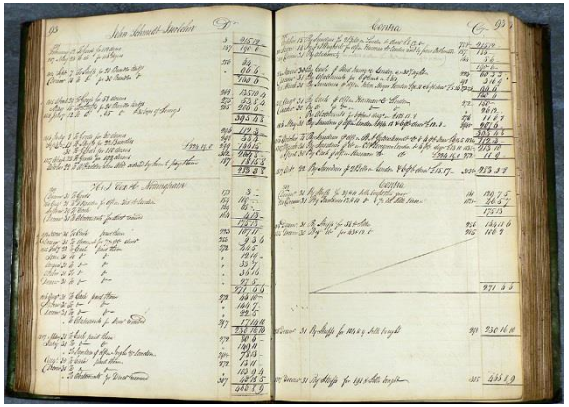
- I committed 10% of my time trying to understand the bigger problems in the field, what was important.
- If I really believe the action is over there, why do I march in this direction? So I changed something I did and I marched in the direction I thought was important. It's that easy.

Richard Hamming, “You and Your Research”

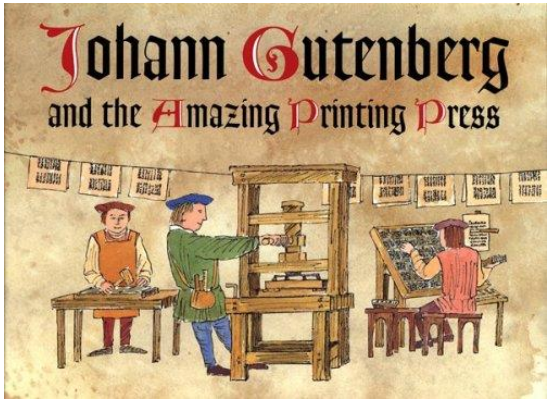
- Corollary: don’t work on unimportant problems
- Be opportunistic, but don’t do “peanut butter strategy”

Recycle and Re-Combine Ideas

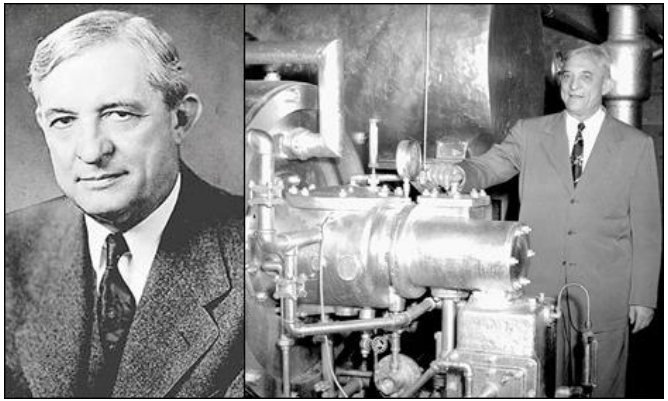
"Big new ideas more often result from recycling and combining old ideas than from eureka moments."



Collective effort



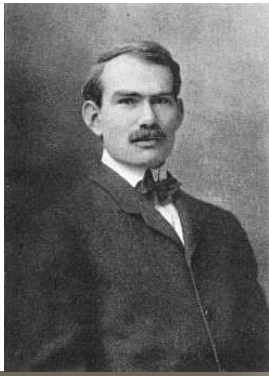
Combinatorial

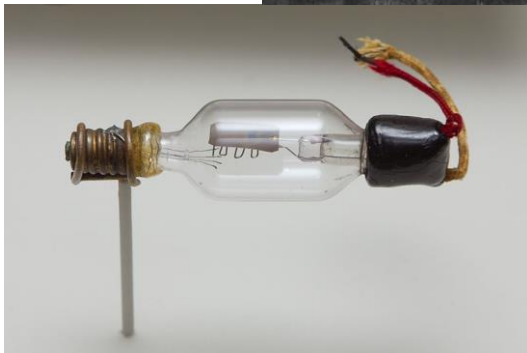


Sheer individual insight

Explore the Adjacent Possible

- Liquid networks – coffeehouses, cities, internet connect ideas
- The slow hunch – scientists nurse connections for years
- Serendipity – LSD, Teflon, Viagra
- Error – Lee de Forest and the Audion





http://www.ted.com/talks/steven_johnson_where_good_ideas_come_from.html

Go All In

- Don Syme led successful MSRC tech transfer
  - .NET Generics (research 1999-2003, “all in” 2002-2004)
  - F# 2.0-3.0 (research 1999-2007, “all in” 2007 on)
  - Required deep trust, total dedication
  - Show respect at all times, win respect by going deep
  - Spot problems, fix them – apply research at the base (language, runtime)
  - Get your hands dirty, treat industry problems as real and serious
  - Align with a shared vision, eg, .NET
- Consider going “all in” with a product team partnership; lots of upside, lots of risks, a guaranteed wild ride

Use the Heilmeier catechism

- **What are you trying to do? [GOAL]**
  Articulate your objectives using absolutely no jargon.
- How is it done today,
  and what are the limits of current practice?
- What's new in your approach,
  and why do you think it will be successful?
- **Who cares?**
- **If you're successful, what difference will it make? [IMPACT]**
- What are the risks and the payoffs?
- How much will it cost?
- How long will it take?
- **What are the midterm and final exams to check [REVIEW]**
  (plans, *not* guarantees)

- **What are you trying to do? (Articulate your objectives using absolutely no jargon.)**
Allow benefits of typed functional programming on the .NET platform.
- **How is it done today, and what are the limits of current practice?**
C# 1.0, with no generics, no functions, no type inference.
- **What's new in your approach and why do you think it will be successful?**
Simple succinct syntax as opposed to C#; the CLR supports multiple languages.
- **Who cares?**
Hmm, maybe not folks making websites, but wizards in the financial industry?
- **If you're successful, what difference will it make?**
Further attract and lockin financial/technical institutions to .NET
- **What are the risks and the payoffs?**
Risks: little support from product groups to start with? Vehicle to transfer research ideas.
- **How much will it cost?**
Don Syme will have to go all in, but one person can drive this?
- **How long will it take?**
Basic system less than a year.
- **What are the midterm and final "exams" to check for success**
Basic system, compiling itself; free download, get customer using it.

Seek criticism

- Better to seek than have imposed!
- Proposals
  - Force you to decide what you want to do
  - Force you to articulate the path to impact
  - Criticism can make a better project
- Reviews
  - Easy to fool yourself; harder to fool peers
  - Some failures are good; and plans never survive contact with reality anyway
  - Any researcher who has never failed is not pushing the envelope… is not living on the edge
  - Researchers *a priori* defensive; *a posteriori* grateful

The fundamental problem is to make the mistakes as fast as possible

If all your researchers' projects succeed, you have failed!

Don’t be seduced by proxies

- Numbers of papers
- Memberships of program committees
- Citations
- Downloads

- Most people can find measures by which they look good
- *It is harder to fix the proxy up front!*
- Do less, but do it well

Work in Collaboration

- Individual versus group work
  - “Only do what only you can do”
  - Your team needs pigs not chickens

- Within a discipline versus interdisciplinary
  - “One half the world thinks the other daft”
  - “Subjects are conveniences for administrators”
  - But wait till you get your PhD
  - And no more than one specialist per discipline

Is Computing an Experimental Science?

Robin Milner

Text of the Inaugural Lecture of the Laboratory for Foundations of Computer Science[1]

We are beginning an ambitious programme of research into the Foundations of Computer Science. This isn't to say that we are beginning to study the theory of computation; this has been going on fruitfully for many years. Nevertheless the particular programme which we have put forward is a new kind of exercise. What makes it new is a central commitment to a double thesis: that the design of computing systems can only properly succeed if it is well grounded in theory, and that the important concepts in a theory can only emerge through protracted exposure to application.

[1]This lecture, marking the inauguration of the Laboratory, was given on 17th January 1986 in the James Clerk Maxwell Building of the University of Edinburgh. The lecture was given by Professor Milner in his capacity as Director of the Laboratory before an audience of about 250 invited guests from U.K. Industrial, Academic and Governmental organisations

Test Theory in Practice

"Just as a physicist or chemist tests and refines his theories by carefully controlled experiment, so it should be with us.
I believe this analogy is close, and that the word "experiment" is also correct for Computer Science.
However, for a physical scientist an experiment will reinforce (or undermine) his conceptual grasp of what is true;
for a Computer Scientist, the experiment will reinforce (or undermine) his conceptual grasp of how to design."

Milner's theories started out trying to do something, tested in practice; eg LCF, CCS
LCF -> ML -> Ocaml, F#, Haskell…

Research like a Lean Startup?

- An organization designed to create new products and services under conditions of extreme uncertainty
  - Learning is the essential unit of progress for startups
  - Easy to kid yourself about what customers want.
  - Validated learning backed by empirical data from real customers

- “Minimum viable product”
  - Version enabling build-measure-learn loop with minimum effort
  - Lacks many features that will be essential later
  - Need to measure it’s impact – put it in front of customers
  - Examples: Video; Concierge; Wizard of Oz

Work with the System

- Exploit resources at your disposal
  - Funding bodies, but also “cognitive surplus”
    eg the open source community, specialist interest groups
- If you fight the system, pick your battles carefully
  - Hamming: Which do you want to be? The person who changes the system or the person who does first-class science?
- Invite yourself places
  - Boston -> Newhaven -> DC -> Chicago -> Calgary -> Vancouver -> Berkeley -> New Jersey

Maintain Balance

- Research jobs are amazing, but demanding, anxious-making
  - Too much to do? Doing well? Nobody tells you what to do

Get out of the lab

When it started out it was an awful lot of fun - keep the fun in the house

Identify your principles: work, family, community

http://www.cra.org/uploads/documents/events/cmw/2010/CMW2010.Fisher.pdf

Five Regrets of the Dying

- I wish I'd had the courage to live a life true to myself, not the life others expected of me.
- I wish I hadn't worked so hard.
- I wish I'd had the courage to express my feelings
- I wish I had stayed in touch with my friends
- I wish that I had let myself be happier

http://www.hospicepatients.org/five-regrets-of-the-dying-bronnie-ware.html

12 Resolutions for Grad Students

- Map out the year
- Improve productivity
- Embrace the uncomfortable
- Upgrade your tools
- Stay healthy
- Update your CV and web site
- Network
- Say thanks
- Volunteer for a talk
- Practice writing
- Check with your committee
- Keep an eye on the job market

http://matt.might.net/

Your Cambridge Homework!

- Manage email time better
  - Delete/delegate/do/defer
  - Switch off notifications
  - No email at weekends
- You choose your boundary

- Organise a speed dating society
  - Forge new connections
  - Learn how to plan/lead a meeting
  - Take a professional initiative
- You cross a boundary

- Know what you are trying to do
- You've got to work on important problems
- Go all in
- Use the Heilmeier Catechism
- Know tactics for creativity
- Seek criticism
- Don't be seduced by proxies
- Research like a lean startup
- Work with the system
- Maintain balance and keep the fun in the house
- Don't send email at weekends
- Found a new branch of the Research Speed Dating Society

- Finish things and give them to people!

Discussion

Resources

- Richard Hamming (Bell Labs, 1986): *You and Your Research* http://www.paulgraham.com/hamming.html
- David Patterson (UCB, 2010): *How to Have a Bad Career in Research/Academia* http://www.cra.org/uploads/documents/events/cmw/2010/CMW2010.Patterson.pdf
- Kathleen Fisher (AT&T Labs, 2010): *Finding Balance* http://www.cra.org/uploads/documents/events/cmw/2010/CMW2010.Fisher.pdf