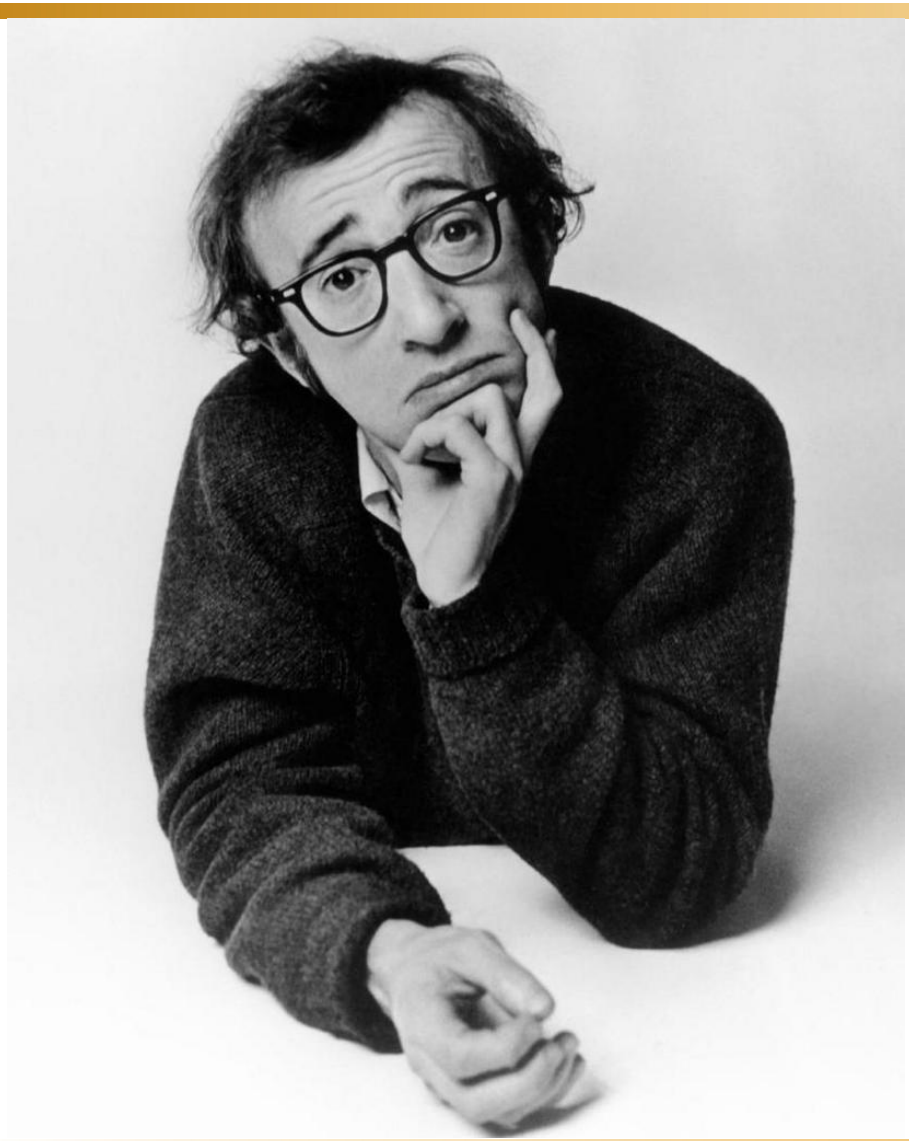
Mastering the Fine Art of the Pivot

Todd Austin, University of Michigan

Perspectives on Failure



If you're not failing every now and again, it's a sign you're not doing anything very innovative.

Rule-Breaking Research

- A *rule-breaking approach* to research is effective and engaging
  - Find a venerable rule and break it however possible
  - E.g., DIVA – designs must be bug-free to launch
  - E.g., CDI – programs need indirect jumps to implement calls/returns/dll's
- The "rules" create artificial barriers that hide good ideas
  - You will often find yourself on very fertile ground
- You will more fully engage your community
  - One half will think your crazy idea will never work
  - One half will be intrigued (with your crazy idea)
- A rule-breaking approach is more prone to failure
  - Often the outcome of the work is a deeper understanding of why the rule you broke should never be broken
  - Even when the research is successful, it can be very difficult to publish

Working Through Failures

- Step #1: ***the failure***
  - Essentially, the hypothesis that you have been working toward is incorrect
- Step #2: ***the enlightenment***
  - Assess why your hypothesis is incorrect
- Step #3: ***the pivot***
  - Piv·ot /ˈpivət/ - turn on or as if on a swivel
  - Explore how you can use this new understanding to reach your goal
  - Or, ***reach a new goal*** with this new understanding

The Downside of Failure

- Let's face it, failure really sucks
  - Try to stay focused on the bigger picture and eventual possible payoffs
  - Be resilient, and be ready to pivot
  - Most of my top-cited papers are in MICRO, rejected out of ISCA
- It is more difficult to publish negative results
  - Not true for all research communities, e.g., life sciences
  - WDDD and NOPE are welcome additions to computer architecture
- There is a fundamental tension between high-risk failure-prone research and PhD students
  - PhD students need to make steady progress toward a PhD degree
  - Often, I will "skunk work" my new ideas with undergraduate students

Case #1: Discovering Runahead

- ***The failure***: Attempting to show that simulators that don't model mispeculation are ***overestimating*** performance
  - These simulators ***underestimate*** performance
- ***The enlightenment***: Executing the mispeculation path (runahead) provides significant performance advantages as it warms up the caches
- ***The pivot***: Years later, using this same idea to develop the DIVA runtime checker, which uses runahead ideas to keep the checker processor very simple

Case #2: Cache-Conscious Data Placement

- ***The failure***: Attempting to improve D-cache performance by reordering data in memory for better spatial/temporal locality
  - Extremely difficult for the compiler to improve upon natural data layout
- ***The enlightenment***: Programmers co-place logically related data, leading to significant spatial/temporal locality
  - Randomly ordered data reduces cache performance by 20-30%
- ***The pivot***: Reimagine the placement algorithm to only make highly reliable placement improvements over natural layout
  - Eventually leading to a 24% improvement in cache performance, and my first ASPLOS paper!

Additional Advise for Graduate Student

- Getting papers published
  - Important tip: ***assume your paper will never be read***
  - Corollary: write your paper so that the i) abstract, ii) intro, iii) figures and well-detailed captions, and iv) conclusion ***transmit the key ideas***
  - Approach meshes well with the highly demanding review process

- Getting the word out is critical to an idea's success
  - Be an evangelist for your project
  - Name your project so the community can talk about it

Questions or Comments?