20 Years
Well, actually, further back than that!
And 5000 years forward

John R. Mashey, mash at heymash dot com
Overview

• Introduction
  – *Many failures, but no surprise*
• “Small is Beautiful and Other Thoughts on Programming Strategies”
  – 1977-1983,
• Why things fail
• Hot Chips 1989
• Baggage and Surprises
The 1977 talk
The 1977 talk – software failure

DATA PROCESSING

“Advanced Projects in Data Processing” [EDP71A]

- Survey of projects
  - 18 projects total
  - 5 infeasible
  - 5 feasible, but no acceptance
  - 2 good fail-outs, no return
  - 3 partial success, return (?)
  - 3 success

- Look at 4 very successful projects
  - All < 1 year elapsed time, < 5 staff-years
  - None were accepted immediately

“We took as “successful” a project that met its requirements on schedule within the budgeted dollars and satisfied the customer. On this basis, out of 10 or 12 projects that we examined, we had one success and a whole lot of failures.”

- J. Aron (IBM)
The 1977 talk – software failure

- $ OVERRUNS (> 50%)
- CALENDAR OVERRUNS (> 60%)
- 9 CONTRACTS ($6.8M)
  - $3.2M Delivered, never used
  - $1.9M Paid, never delivered
  - $1.3M Extensive rework
  - $0.198M Used after changes
  - $0.119M Used as delivered
- PRODUCTION -> PROTOTYPE
- COST TO FIX > = COST TO DEVELOP
  GAO FGMSD-80-4, Nov. 9, 1979
  COMPUTERWORLD Sept. 29, 1980
OTHER AREAS OF R&D

“Organization of Unsuccessful R&D Projects” [JOY71A]

“In this sense [successful products] nearly every proposal considered is an eventual failure.”

- Of 400-500 ideas, 1-2 work out
- 4/5 R&D hours spent on projects that do not reach commercial success

AND OUTSIDE R&D...

Museum of Modern Art and designs of “lasting value”

1934 397 1 survived
1939 70 1 saw further development
1950 46 1 survived

“...a score of 3 successes and 510 misses is far from reassuring.”

- V. Papanek, in [PAP73A]

FAILURE IS THE NORM
1977 – failure modes … and hardware?

WAYS IN WHICH PROJECTS FAIL

- Wrong problem
- Wrong approach
- Infeasible performance
- Cannot be finished at all
- Becomes irrelevant or inappropriate when done
- Lack of user acceptance
- Leapfrogged by somebody moving faster

"Don’t look back. Something may be gaining on you."
— Satchel Paige
1977 – failure modes … and how is hardware different?

- More $$
- (Usually)
- But (lack of) software can cause failure
Hot Chips 1989

- ECL SPARC Chip Set
- Architecture of the P1 – A 250 MHz SPARC in GaAs
  
  *Mike O’Dell: Putting UNIX on Very Fast Computers or What the Speed of light Means to Your Favorite System Call, USENIX, Summer, 1990*

- Intel i860 Million Transistor 64-bit Processor
- Panel Session : Compiler Issues with Hot Chips

- Building on-the-edge hardware is hard
- Building it without having software issues in good shape … deadly
• Biggest surprise - How long it took to get 64-bit CPUs
• Next biggest - C has evolved from 1973, but … 35 years
  – And of course, FORTRAN is still around, 50+ years
• And the future, 5000 years off, from Vernor Vinge, “A Deepness in the Sky”, quoted in ACM Queue - Languages, Levels, Libraries, and Longevity

“The programs were crap…Programming went back to the beginning of time…There were programs here that had been written five thousand years ago, before Humankind ever left Earth. The wonder of it—the horror of it…these programs still worked…down at the very bottom of it was a little program that ran a counter. Second by second, the Qeng Ho counted from the instant that a human had first set foot on Old Earth’s moon. But if you looked at it still more closely… the starting instant was actually about fifteen million seconds later, the 0-second of one of Humankind’s first computer operating systems…

“We should rewrite it all,” said Pham.
“It’s been done,” said Sura.
“It’s been tried,” corrected Bret…“You and a thousand friends would have to work for a century or so to reproduce it… And guess what—even if you did, by the time you finished, you’d have your own set of inconsistencies. And you still wouldn’t be consistent with all the applications that might be needed now and then…”

“The word for all this is ‘mature programming environment.’”