1
The Tar Pit

The Tar Pit

- The Programming Systems Product
- The Joys of the Craft
- The Woes of the Craft

Tar pit analogue

- *Great beasts in tar pits* versus *Large-system programming*
- "The fiercer the struggle, the more entangling the tar"
- "Few have met goals, schedules and budgets"
- "The accumulation of interacting factors brings slower and slower motion"
- "Everyone seems to be surprised about the stickiness of the problem"

Software's chronic crisis

- From Scientific American September 1994
- Denver Airport automated baggage handling
- 2x size of Manhattan; 10x width of Heathrow
- 4,000 robotic carts; 56 bar-code scanners; 400 radio receivers; 5,000 electric eyes
- 100 networked computers
- 20 airlines

What if software is late?

- Denver Airport baggage handling software
- $193M to BAE Automated Systems
- Scheduled for delivery October 31, 1993
- Delayed to December, then March, then May
- By June, the airport's credit rating was "junk"
- Airpost lost $1.1M per day in costs and interest
- Planners gave up on predicting delivery

The Programming Systems Product

- Individual programmers and small teams routinely exceed industry averages in code output
- The difference lies in what is being produced
- Differentiate between:
  - *Program*: Run by the author in the same context as it was developed
  - *Programming Systems Product*: Used by others in a different context

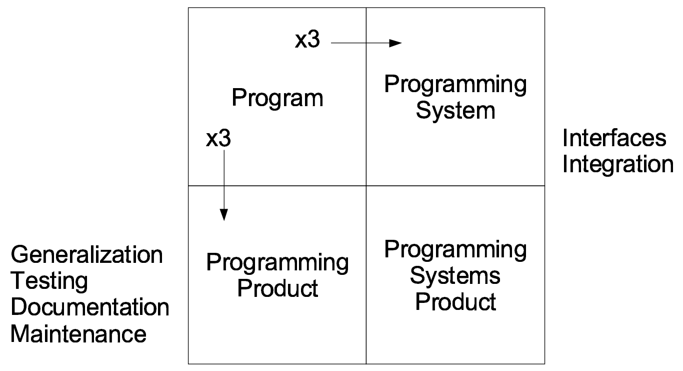
From program to product

- Run, tested, repaired and extended by anybody
  - i.e., well documented externally and internally
- Applicable to full possible range of inputs
  - i.e., generalized across all anticipated uses
- Throughly tested
  - i.e., sufficiently reliable to assume full liability
- Cost(*Product*) ≈ 3 x Cost(*Program*)

From program to system

- Contributes to the solution of a larger task
  - i.e., interfaces with current and future systems
- Performs well with limited resources
  - i.e., uses limited compute and network budget
- Behaves predictably with interacting systems
  - i.e., robust to complexity and errors elsewhere
- Cost(*System*) ≈ 3 x Cost(*Program*)

From program to systems product



The Joys of the Craft

- Making things
- Making things that are useful to other people
- Making things from interlocking components
- Learning from new experiences
- Working in a purely intellectual medium

The Woes of the Craft

- Adjusting to perfection in expression
- Dependance on other's imperfect programs
- Painstaking labour
- Slow convergence to completion
- Persistant threat of obsolescence