CS446
Software Engineering

Richard Bartels

University of Waterloo
Faculty of Mathematics
Department of Computer Science

September 11, 1996

CS446
Software Engineering
The Software Crisis

Richard Bartels

University of Waterloo
Faculty of Mathematics
Department of Computer Science

September 11, 1996

Outline

- Random Statistics
- Graphical Overview
- Horrible Examples
- Pertinent Comments
- Engineering Evolution Paradigm

Reference:

Scientific American, September, 1994, pp. 86–95

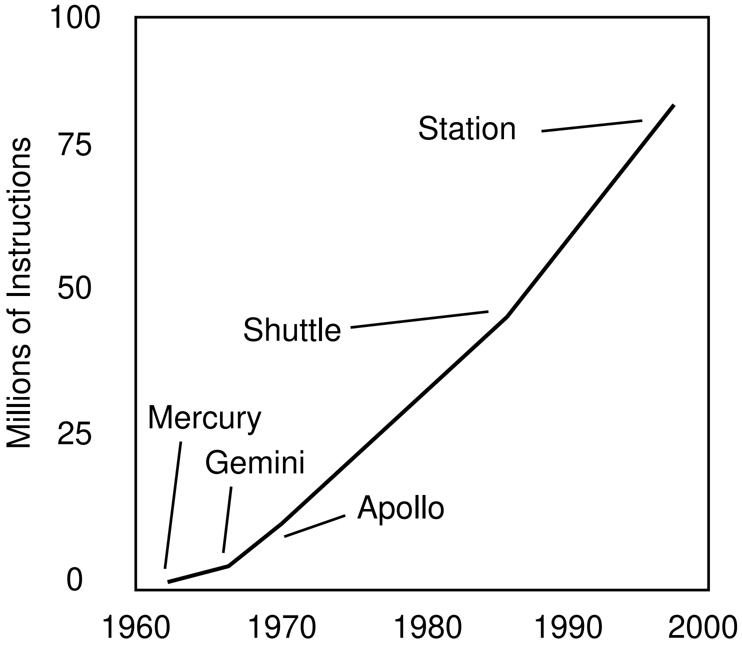
Random Statistics

Large-scale software systems:

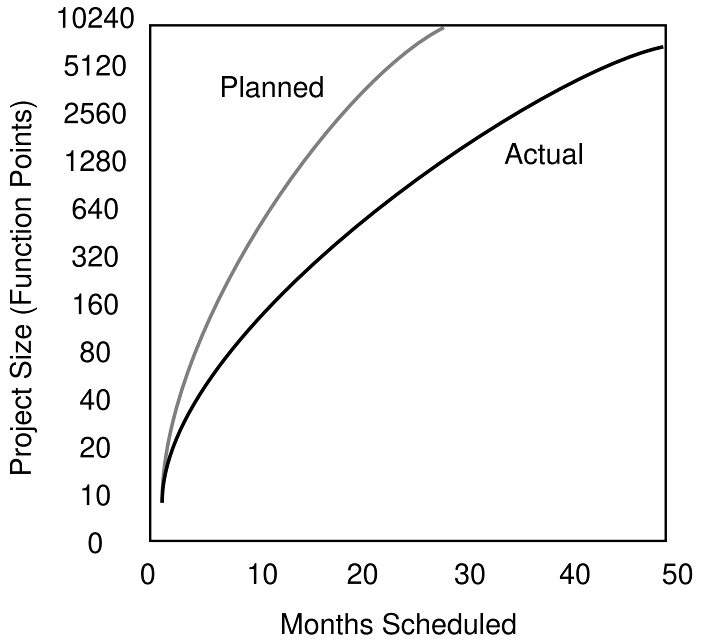
- For every 6 new systems, 2 others are cancelled
- The average project overshoots its schedule by 50%
- 3/4 of all larger systems are "failures"
  - Either do not work as intended
  - Or are never used
- Nearly every system is hand crafted
- Research takes an average of 18 years to become practice
- Code in consumer products is doubling every two years
  - The power train in a GM car runs over 30,000 lines of code
- In a study of 24 projects of leading software companies
  - 55% cost more than expected
  - 68% overran schedules
  - 88% needed significant redesign
- The cost of well-managed projects is
  - Around US$100 per line for consumer code
  - Around US$500 per line for mission-critical code

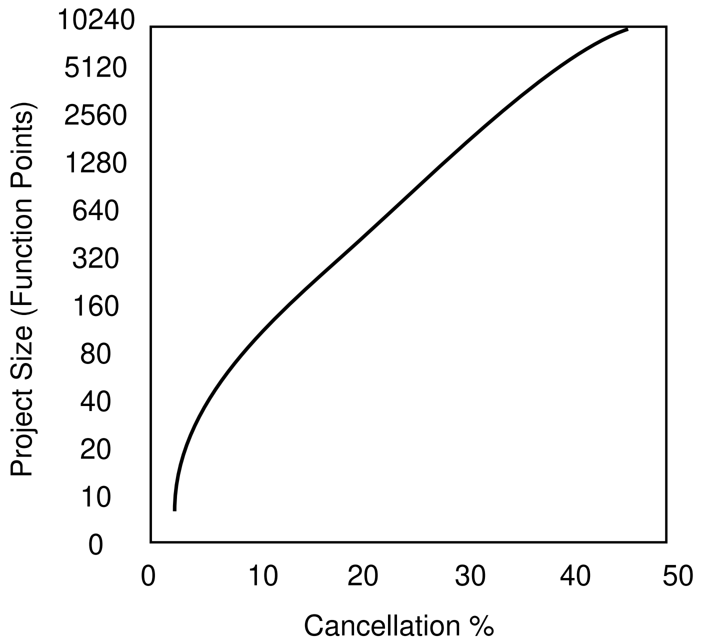
(and it's even more expensive for the failures)

Code Growth at NASA



Schedule Overruns and Cancellations





Horrible Examples

Denver Airport Baggage Handling System

- Original cost estimate US$193 million
- Scheduled for December, 1993
- As of June, 1994
  - No completion in sight
  - US$1.1 million per day continuing losses

California DMV System

- Begun in 1987
- Projected cost explodes to 6.5 times estimate
- Early delivery prediction revised to end of centry
- Abandoned after 7 years and US$44.3 million

Revision of Air Traffic Control System

- Costing up to US$900 per line of code
- Average line of code is rewritten at least once
- Parts already finished not meeting reliablilty requirements
- 5 years late already
- A billion dollars over budget

Pertinent Comments

- Software production is like the production of goods before the industrial revolution, when there was a nonspecialized approach to manufacturing goods that involved very little interchangeability and a maximum of hand crafshmanship.
- We have a hand-to-mouth, every-programmer-builds-everything-from-scratch, preindustrial approach.
- The majority of software firms are at a level at which they have no formal process, no measurements of what they do, no way of assessing which of their practices are successful, and no way of knowing when they are on the wrong track or off the track altogether.
- When software is embedded in light switches, you have to get it right the first time, because you're not going to have a chance to update it.
- You can't build skyscrapers using carpenters.
- There are no silver bullets.

Engineering Evolution Paradigm

Early Stages

- Virtuosos and talented amateurs
- Design by intuition and brute force
- Knowledge transmitted slowly and casually
- Extravagant use of materials
- Manufacture for use rather than sale

Middle Stages

- Skilled craftspeople
- Established procedures
- Pragmatic refinement
- Training and guilds
- Concern for cost and supply of materials
- Manufacture for sale

Advanced Stages

- Educated, certified professionals
- Analysis and theory
- Standardization and controls
- Progress based on science
- Market segmentation based on variety
- Specialization