Outline

• Early software economics experiences
  – General Dynamics, Rand: 1955-1973

• Large-scale industrial applications
  – TRW, 1973-1989

• Conclusions
Early Software Economics Experiences

• 1955: First day in the software field

• 1956-68: Blissful ignorance: cost-plus R&D

• 1966-68: Economics of interactive graphics

• 1968-70: Rand Computer Systems Analysis Group

• 1970-71: Air Force CCIP-85 study
My First Day As A Programmer
- General Dynamics, 1955

- Manager
  “We’re paying that computer $600/hour.
  We’re paying you $2/hour.
  We want you to act accordingly.”

- Positive habits
defensive programming, desk checking, manual execution

- Negative habits
  overconcern with hardware, saving microseconds
Economics of Interactive Graphics, 1960’s

- Research applications on DARPA-purchased IBM 360/50
  - Rocketry, medicine, networking

- Used enthusiastically by Rand engineers, analysts

- No outside interest at $50/hour
  - Maybe I should learn more about economics
Systems Analysis at Rand

- Major defense cost-effectiveness analyses
  - Force structure; operational strategies

- Established Domestic Studies Division, 1967

- Very strong economics, applied math staff
  - At least 2 Nobel economists
  - Originated much of linear and dynamic programming, game theory
Rand Computer Systems Analysis
- Experiences, 1968-70

- Computer slowed down transaction processing
  - No cost-benefit analysis, just computerizing

- Overrun, delayed fire dispatching system
  - Gold-plated geolocation system

- Huge expense for archaic file system
  - Overconcern with sunk cost

- Insecure urban computer centers
  - No risk analysis
CCIP -85 Study: Air Force, 1970-71

- Forecasted 1985 AF command-control needs
- Recommended information processing research program
- Key needs not in MIPS and displays, but in software
- Major recommendations
  - Structured methods
  - Requirements/exercise technology
  - Software certification technology
Air Force Computing Cost Trends

% of total costs

Year

Hardware

Software

1955  1970  1985
Risk of Minimizing Hardware Cost

Experience

Folklore
Career Crossroads

• Successfully exited software career
  - Thought the action was in systems analysis
• Found software a systems analysis challenge
• Joined TRW to try to systematize software practice
  - Economics-driven processes
  - Risk Management and spiral model
  - Cost estimation and COCOMO
Economics-Driven Processes

- Early concept validation
  - Prototyping
- Thorough specifications
- Planning and control
- Continuous improvement
- Risk management
Software Risk Management

• Use Risk Exposure for management priorities
  \[ RE = \text{Prob}(\text{Loss}) \times \text{Size}(\text{Loss}) \]
• Basis for spiral model decisions
• Helps answer “how much is enough?”
  - Testing, prototyping, COTS evaluation, etc.
Example RE Profile: Time to Ship

- Loss due to unacceptable dependability
- Loss due to market shade erosion

![Graph showing the relationship between time to ship and the number of defects, rivals, and their levels of dependability and market shade erosion.](image)

- Many defects: high P(L)
- Critical defects: high S(L)
- Many rivals: high P(L)
- Strong rivals: high S(L)
- Few rivals: low P(L)
- Weak rivals: low S(L)
- Few defects: low P(L)
- Minor defects: low S(L)
Example RE Profile: Time to Ship

- Sum of Risk Exposures

\[ RE = P(L) \times S(L) \]

- Many defects: high \( P(L) \)
  - Critical defects: high \( S(L) \)
- Few rivals: low \( P(L) \)
  - Weak rivals: low \( S(L) \)
- Many rivals: high \( P(L) \)
  - Strong rivals: high \( S(L) \)
- Few defects: low \( P(L) \)
  - Minor defects: low \( S(L) \)

Time to Ship (amount of testing)
Constructive Cost Model (COCOMO)

- Motivated by TRW proposal commitments
- Drew on good previous work
  - SDC, Aron, Wolverton, Doty, Walston-Felix
- Published in *SW Engr. Economics*, 1981
Software Estimation: The Receding Horizon

COTS: Commercial Off-the-Shelf
RAD: Rapid Application Development

Relative Productivity
Estimation Error
Unprecedented Preceded Component-based COTS RAD Systems of Systems
A B C D

Time, Domain Understanding
Conclusions

- The software field exists because processed information has value

- Understanding and working with information-value effects is in our enlightened self interest

- Be prepared to adapt to continual change in value propositions