Future Trends in Software Economics

Barry Boehm, USC
COCOMO / SCM Forum #14
October 27, 1999

Outline

• Cost and demographic trends
• Productivity and estimating accuracy trends
• Trends in estimation techniques
• Software economic analysis trends
• Conclusions
Annual Expenditure on Software Development

The Boom in User Programming

User programming
Repeat of "everyone a phone operator" trend
55M Users in US in year 2005

Application generators (0.9M)
Applications composition (0.7M)
System integration (0.7M)
Infrastructure (0.75M)
Future Software Trends

- Increased complexity
  - Everything connected
  - Opportunities for chaos (agents)
  - Systems of systems

- Decreased control of content
  - Infrastructure
  - COTS components

- Faster change
  - Time-to-market pressures
  - Marry in haste; no leisure to repent
  - Adapt or die (e-commerce)

- Fantastic opportunities
  - Personal, corporate, national, global


![Lines of Code in Service: U.S. Dept. of Defense](image-url)
Trends in Software Expansion (Bernstein, 1997)

- Expansion Factor: The rate of growth of size of code in a decade.

- Order of Magnitude Increase Every Twenty Years

Productivity and Estimation Accuracy Trends
Software Estimation Techniques

- Regression-Based: Delphi, WBS
- Moocil-Based
- Regression-Based: OLS, Robust
- Composite- Bayesian
- Learning-Oriented: Neural, Case-based
- Dynamic-Based

Software Economic Analysis Trends: Needs

- Reasoning about rapid development
- Reasoning about benefits as well as costs
- Reasoning about e-commerce/software costs and benefits
- Reasoning about COTS and reuse
- Dealing with uncertainty and risk
Software Economic Analysis Trends: Capabilities

- Static present-value techniques less valuable
  - Too many sources of change
- Dynamic present-value techniques emerging
  - multiple, time-dependent discount rates
- Utility theory and multi-objective decision analysis
  - Applications to software requirements and design
- Applications of real-options theory to software decisions
  - Investment option equivalent to portfolio of traded assets
  - Used for VLSI design
- EDSER-1, -2 workshops at ICSE 1999, 2000

Conclusions

- Software field experiencing tremendous (dizzying) rate of growth and change
- Estimation models need to continuously recalibrate and re-engineer
- Software organizations need to continuously recalibrate and re-engineer
- Need good metrics, closed-loop feedback process
Using COCOMO II to Cope With Change: I

System objectives: functionality, performance, quality

Project Parameters:
Personnel, team, sites, platform

Corporate parameters:
Tools, processes, reuse

Rescope

Yes

COCOMO II

Cost, Schedule, Risks

Using COCOMO II to Cope With Change: II

System objectives: functionality, performance, quality

Project Parameters:
Personnel, team, sites, platform

Corporate parameters:
Tools, processes, reuse

Rescope

Remote project (i.e., milestones)

Milestone data, Interim

Release, Milestones, plans, resources

Don't

Yes

Rescope

COCOMO II
Using COCOMO II to Cope With Change: III

System objectives: functionality, performance, quality

Project parameters: personnel, tools, teams, platform

Corporate parameters: tools, processes, reuse

Evaluate corporate strategies

Recalibrate or extend COCOMO II

COCOMO II calibration data

Acceptance by customer

Measures: schedule, cost, quality

Review milestones, plans, resources

End project

Using COCOMO II to Cope With Change: IV

System objectives: functionality, performance, quality

Project parameters: personnel, tools, teams, platform

Corporate parameters: tools, processes, reuse

Evaluate corporate strategies

Recalibrate or extend COCOMO II

COCOMO II calibration data

End project

End project