



**...And Very Few Lead Bullets, Either**

**Barry Boehm, USC**  
**University of Massachusetts**  
**Computer Science Convocation**  
**September 14, 2000**



# Outline

- **Silver bullets and lead bullets**
- **Information technology trends**
  - **The dwindling lead-bullet niche**
- **Underlying world-view trends**
- **Implications for software engineering and computer science (SE & CS)**
- **Implications for SE & CS professionals**



# The Silver Bullet Fantasy

(Brooks, 1986)

- **Even if we can't define what the “software problem” werewolf is,**
- **And we've seen that lead, bronze, iron, and steel bullets can't kill it,**
- **We're sure there's a silver bullet that can**

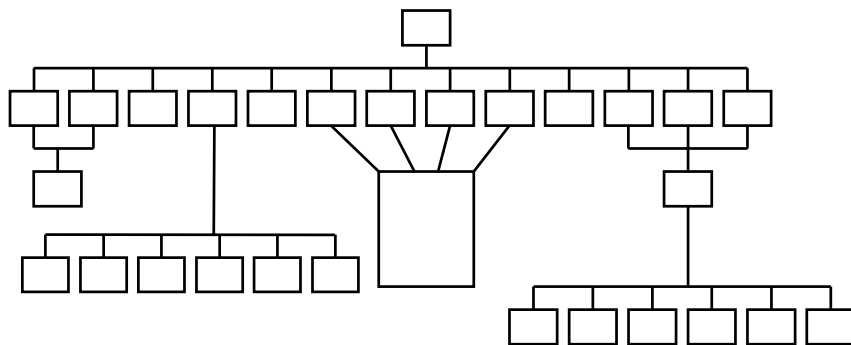
**\*\*\*\*\***

- **Fallacy: We can fix things we understand**
  - **“Accidental” software problems**
- **But we can't fix things we don't understand**
  - **“Essential” software problems**

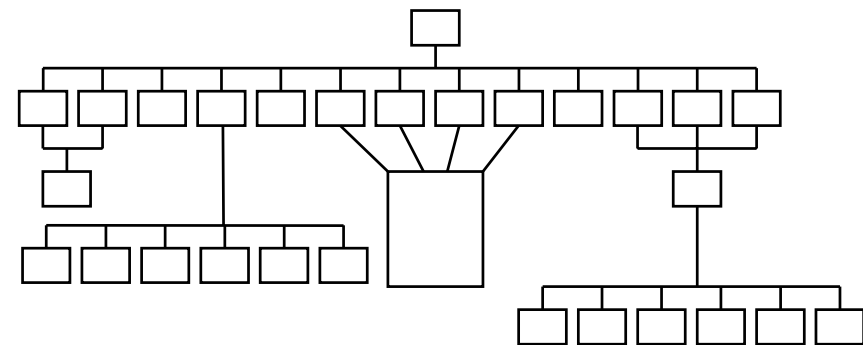
# Example: Conway's Law and its Converse

- **Conway's Law (Datamation, 1968), extended**

**The structure of a  
computer program ...**



**reflects the structure  
of the organizations  
that build and use it**





# Converse of Conway's Law

- **We will learn how to build perfectly functioning software**
- **As soon as we learn how to build perfectly functioning organizations**



# The Lead Bullet Expectation

- **If a lead bullet can kill a software-problem wolf this year,**
- **It will be able to do it next year too**
- **Counterexamples**
  - **Waterfall model of the software process**
  - **Pre-WYSIWYG word processing architecture**
  - **Pre-Web book sales management applications**
- **Key drivers: technology, economics, humanization**



# Information Technology Trends

## Traditional Development

- Standalone systems
- Stable requirements
- Rqts. determine capabilities
- Control over evolution
- Enough time to keep stable
- Repeatability-oriented process, maturity models

## Current/Future Trends

- Everything connected
- Rapid requirements change
- COTS capabilities determine rqts.
- No control over COTS evolution
- Ever-decreasing cycle times
- Adaptive process models



# Lead Bullets with Dwindling Niches

- **Complete, consistent, traceable, testable requirements**
- **Static domain architectures and enterprise architectures**
- **Formal methods**
- **Contract models of software management**
- **COTS- and value-insensitive object-oriented methods**



# Core Niches for Current Lead Bullets

- Still very important

- **High-assurance,**
- **Single-processor,**
- **Autonomous control systems**



# Outline

- **Silver bullets and lead bullets**
- **Information technology trends**
  - **The dwindling lead-bullet niche**
- ➔ • **Underlying world-view trends**
- **Implications for software engineering and computer science (SE & CS)**
- **Implications for SE & CS professionals**



# Underlying World-View Trends

- **Universalism → Situationalism**
  - Stephen Toulmin, Cosmopolis, U. Chicago Press, 1990
- **Reductionism → Emergence**
  - Stuart Kauffman, At Home in the Universe, Oxford U. Press, 1995.
  - W. Brian Arthur, “Increasing Returns and the New World of Business,” Harvard Business Review, Jul/Aug 1996, pp. 100-109.
  - James Highsmith, Adaptive Software Development, Dorset House, 1999.
- **Software Focus → System Focus**
  - John Thorp, The Information Paradox, M<sup>c</sup>Graw Hill, 1998.



# Cosmopolis: The Erosion of Modernist Philosophy

- **Dominant since 17<sup>th</sup> century**
- **Formal, reductionist**
  - Apply laws of cosmos to human polis
- **Focus on written vs. oral; universal vs. particular; general vs. local; timeless vs. timely**
  - one-size-fits-all (lead bullet) solutions
- **Strong influence on focus of computer science**
- **Weak in dealing with human behavior, rapid change**



# Reductionism vs. Emergence

- **Order is not imposed on complex adaptive systems; it emerges (Kauffman)**
- **Knowledge-based industries have increasing vs. decreasing returns (Arthur)**
  - Network effects, up-front costs, customer groove-in
  - Adaptation succeeds better than optimization
- **Adaptive model best fits future software projects (Highsmith)**
  - Balance of discipline and flexibility

# The Information Paradox (Thorp)

- **No correlation between companies' IT investments and their market performance**



- **Field of Dreams**
  - **Build the (field; software) and the great (players; profits) will come**
- **Need to integrate software and systems initiatives**



# Outline

- **Silver bullets and lead bullets**
- **Information technology trends**
  - **The dwindling lead-bullet niche**
- **Underlying world-view trends**
- ➔ • **Implications for software engineering and computer science (SE & CS)**
- **Implications for SE & CS professionals**



# Implications for SE & CS

- **Broaden involvement with behavioral sciences**
  - Not “What can be computed” (ACM)
  - But “The scientific investigation of phenomena involving computers” (Newell, Perlis, & Simon, Science, 1967)
- **Broaden SE scope to include participation in system definition**
- **Balance discipline and flexibility**
  - Situation-driven model generators
  - Risk-driven “how much is enough?”
  - Stakeholder win-win plus accountability

# Balancing Discipline and Flexibility



- **Billie Jean King at UMass Commencement**



# The “Separation of Concerns” Legacy

**“The notion of ‘user’ cannot be precisely defined, and therefore has no place in CS or SE.”**

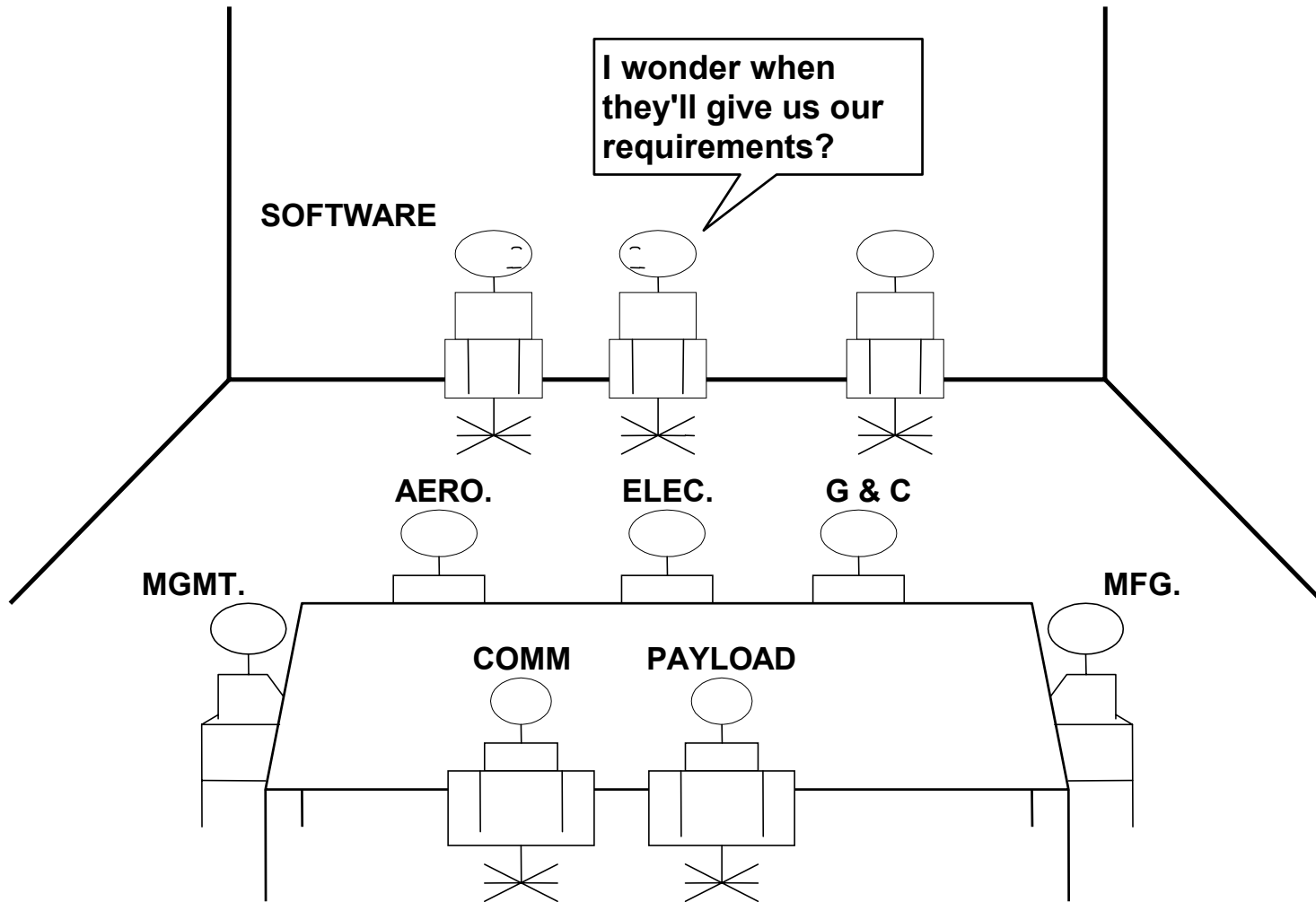
**– Edsger Dijkstra, ICSE 4, 1979**

**“Analysis and allocation of the system requirements is not the responsibility of the SE group but is a prerequisite for their work.”**

**– Mark Paulk et al., SEI Software CMM v.1.1, 1993**



# Resulting Project Social Structure

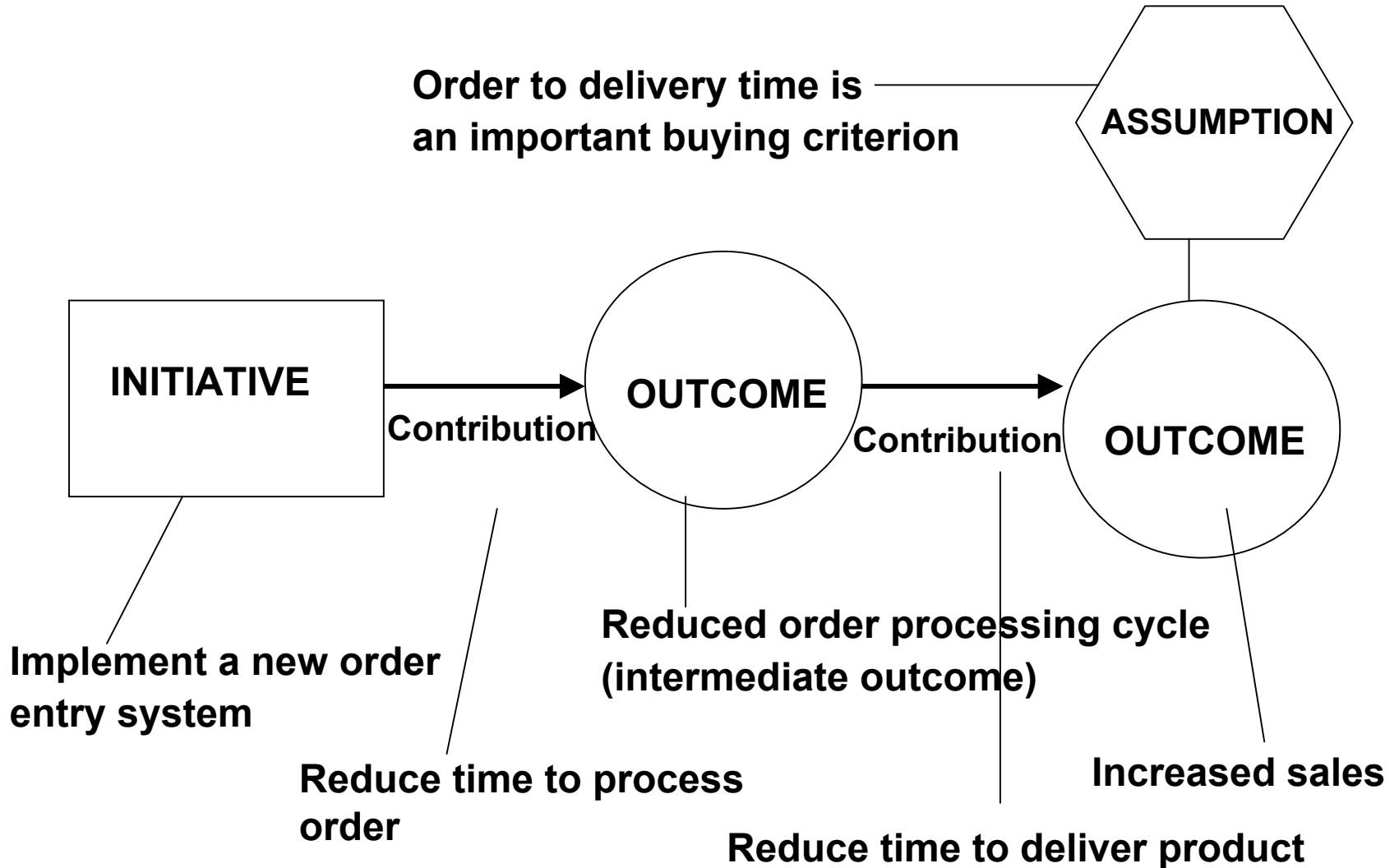




# Responsibility for System Definition

- **Every success-critical stakeholder has knowledge of factors that will thwart a project's success**
  - Software developers, users, customers, ...
- **It is irresponsible to underparticipate in system definition**
  - Govt./TRW large system: 1 second response time
- **It is irresponsible to overparticipate in system definition**
  - Scientific American: focus on part with software solution

# DMR/BRA Results Chain





# How Much Is Enough: Risk-Driven Specifications

- If it's risky not to specify precisely, Do
  - Hardware-software interface for safety-critical-system
- If it's risky to specify precisely, Don't
  - GUI layout
  - COTS behavior



# Implications for SE & CS

- **Broaden involvement with behavioral sciences**
  - Not “What can be computed” (ACM)
  - But “The scientific investigation of phenomena involving computers (Newell, Perlis, & Simon, Science, 1967)
- **Broaden SE scope to include participation in system definition**
- **Balance discipline and flexibility**
  - Situation-driven model generators
  - Risk-driven “how much is enough?”
  - Stakeholder win-win plus accountability



# Implications for SE & CS Professionals

- **Prepare to live in a world of rapid change, and to:**
  - Phase out many “lead bullet” solutions
  - Build more adaptable solutions
  - Help create more adaptable solution communities
- **Reach out to software customers and users**
  - They need your expertise (and vice versa)
  - Helps you understand the sources of IT value
  - Think win-win, with accountability
- **Avoid one-size-fits-all solutions**
  - Adapt solutions to changing value situations
  - Look for balanced rather than extreme solutions
  - Use risk considerations to determine “how much is enough”
- **Read Brooks, Toulmin, Kauffman, Arthur, Highsmith, and Thorp!**



# References

**Frederick P. Brooks, The Mythical Man Month (2<sup>nd</sup> ed.),  
Addison Wesley, 1996**

**Stephen Toulmin, Cosmopolis, U. Chicago Press, 1990.**

**Stuart Kauffman, At Home in the Universe,  
Oxford U. Press, 1995.**

**W. Brian Arthur, “Increasing Returns and the New World of  
Business,”**

**Harvard Business Review, Jul/Aug 1996, pp. 100-109.**

**James Highsmith, Adaptive Software Development,  
Dorset House, 1999.**

**John Thorp, The Information Paradox,  
M<sup>c</sup>Graw Hill, 1998.**