What about software?

- Too complex for complete analysis:
  - Separation into non-interacting subsystems distorts the results.
  - The most important properties are emergent.

- Too organized for statistics
  - Too much underlying structure that distorts the statistics.

"Organized Complexity" (Weinberg)
Other Factors

- Large discrete state spaces
  - Continuous vs. discrete math
  - Lacks repetitive structure found in computer circuitry
  - Cannot test exhaustively

- Intangibility
  - Invisible interfaces
  - Hard to experiment with and manage
  - Transient hardware faults vs. software errors
  - Hard to diagnose problems
And One More

- No historical usage information to allow measurement, evaluation, and improvement of standard designs over time.
  - Always specially constructed.
  - Usually doing new things.
1. Students will be able to evaluate software engineering techniques and approaches.

"It is important that students bring a certain ragamuffin barefoot irreverance to their studies. They are here not to worship what is known, but to question it."

Jacob Bronowski, The Ascent of Man

"The developed theories ...have rarely been subjected to empirical testing, and so their value remains unknown. They provide zealots with opportunities to market a rash of seminars and courses and to flood the literature with papers advocating the new technologies. When the theories are subjected to testing, what little evidence has been obtained sometimes suggests that the claimed benefits, in fact, may not exist.

Vessey and Weber
Class Objectives

2. Students will be able to exercise professional judgment in selecting an approach for a particular project based on an understanding of:

   - How the present state of software practice came about
   - What was tried in the past
   - What worked and what did not work
   - Why
Required Background

Assignments

- No programming
- Reading summaries:
  Main ideas or themes
  Critical evaluation
  Any additional thoughts
- Some additional short assignments
Reading: Both classic papers and new ones

I would like to see computer science teaching set deliberately in a historical framework.... The absence of this element in their training causes people to approach every problem from first principles. They are apt to propose solutions that have been found wanting in the past. Instead of standing on the shoulders of their precursors, they try to go it alone.

Maurice Wilkes