

No Silver Bullet: Essence and Accident in
Software Engineering

What is a silver bullet in myth?

What is a silver bullet in software?

“There is no single development, in either technology or management technique, which by itself promises even one order of magnitude improvement in productivity, in reliability, in simplicity.”

Why might we expect a silver bullet in software?

What are accidental difficulties in software?
What are essential difficulties in software?

“I believe the hard part of building software to be the specification, design and testing of this conceptual construct, not the labor of representing it and testing the fidelity of the representation.”

“We still make syntax errors, to be sure; but they are fuzz compared to the conceptual errors in most systems.”

What are the four inherent properties of essential difficulties in software?

Complexity
Conformity
Changeability
Invisibility

Complexity

“...a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements.”

“...descriptions of a software entity that abstract away its complexity often abstract away its essence. Mathematics and physical sciences made great strides...by constructing simplified models... It does not work when the complexities are the essence.”

Conformity

“The physicist labors on; however, in a firm faith that there are unifying principles to be found...”

“No such faith comforts the software engineer. Much of the complexity he must master is arbitrary complexity, forced without rhyme or reason by the human institutions and systems to which his interfaces must conform.”

Changeability

“In short, the software product is embedded in a cultural matrix of applications, users, laws, and machine vehicles. These all change continually, and their changes inexorably force change upon the software product.”

Invisibility

“The reality of software is not inherently embedded in space. Hence it has not ready geometric representation in that way that land has maps...”

“In spite of progress in restricting and simplifying the structure of software, they remain inherently unvisualizable.”

What are some of the software technologies that have solved only accidental difficulties?

High-level languages
Unified programming environments
Artificial intelligence
Expert systems
“Automatic” programming
Programming verification

What are the four promising attacks on
the conceptual essence?

Buy versus build
Requirements refinement and rapid
prototyping
Incremental development-- grow, not
build software
Great designers