Andy Hudson

CIS 350

No Silver Bullet —Essence and Accident in Software Engineering – Reading Response

The main point of the article is that there is no fix-all solution, or “silver bullet,” to the inherent, or “essential,” difficulties of software engineering: complexity, conformity, changeability, and invisibility. This lack of growth in software engineering is contrasted to the rapid growth of hardware engineering, where technology is being improved upon at a rapid pace. Brooks proposes ways to improve software engineering, however, which include rapid prototyping, refining requirements, making software as adaptable as possible, and fostering great software designers.

I basically agree with every point Brooks makes, and I found most of his predictions to be correct: Ada isn’t prevalent in modern programming culture, object-oriented programming is prevalent but doesn’t make software engineering exponentially easier (which is arguable compared to a language like BASIC), artificial intelligence is laughably undeveloped in software engineering three decades later, graphical programming is still limited by the size of the PC screen, and any other additional tools are (like OO programming) helpful but not the “silver bullet.”

There is, essentially, no “silver bullet.” And he’s not only correct in this statement, but also in his proposed solutions. Code always changes, programmers are excited by progress, making better software designs earlier is preferable to fixing errors later, and the closer one gets to making a product according to the true specifications, the better. All of Brooks’ solutions have one thing in common: they focus on the improvement of the software development synthesis of human behavior and reality rather than finding what new tools or methods can make the process faster. Believing that there is one solution to software development inefficiency is a grossly simplistic worldview for an unbelievably complex topic.