Reading 4 Response

Liam Strand

Briefly discuss one contrast Brooks draws between Software Engineering and other areas or science or engineering.

Brooks compares Software Engineering to computer hardware (what we would call Computer Engineering). He highlights how Computer Engineering has experienced dramatic, huge improvements over time, whereas Software Engineering has not. He attributes this not to Software Engineering's slowness but to Computer Engineering's speed. No other field has experienced such gains so quickly, and Software Engineering has struggled to keep up.

Brooks discusses several "Hopes for the silver", but his comments are about the state of the art in 1987. Describe one "hope for the silver" that you see today, but that Brooks did not see back then. Or, if you are a pessimist, explain why there is no hope.

Modern languages with strict type systems and memory management are one possible avenue for improving the software development experience. TypeScript, for instance, allows programmers to create extremely expressive types to represent their data, then mandate that their code adheres to the interfaces defined by those types. Rust has a similar algebraic type system, but adds a compile-time check called the "borrow checker" that verifies that the program adheres to the ownership rules of Rust (every variable has one owner and a variable may be borrowed either once mutably or many times immutably). These more stringent requirements prevent whole classes of bugs, or make them much easier to catch.