

# Resenha do artigo “*No Silver Bullet: Essence and Accidents of Software Engineering*”

Famoso pesquisador da área de Engenharia de Software, Frederick P. Brooks Jr. Busca mostrar através de seu artigo “No Silver Bullet: Essence and Accidents of Software Engineering” uma análise decisiva dizendo sobre os desafios que muitos desenvolvedores encontram durante desenvolvimento de softwares. Seu artigo busca principalmente buscar e discutir sobre uma “bala de prata”, que seria a solução para que fosse capaz eliminar vários dos problemas já estruturais em toda a área de desenvolvimento.

## O que seria “bala de prata”?

Durante o artigo, Brooks já introduz o leitor sobre a uma reflexão da metáfora da “bala de prata” e comparando os problemas estruturais com lobisomens, que seriam os projetos que normalmente começam de forma inofensiva e simples, mas que conforme vai se desenvolvendo se tornam problemas enormes de prazo, orçamento e um sistema mais passivo de falhas. Tendo isto em vista, a principal busca de diversos gestores é uma solução que resolva esses problemas de uma forma rápida e realmente eficaz, aumentando a eficiência nas quais os projetos são realizados e de forma confiável. O autor, no entanto, defende que tal esperança é ilusória: não há perspectivas de avanços tecnológicos ou metodológicos capazes, por si só, de promover ganhos de uma ordem de magnitude, porém Brooks defende que não existe uma solução definitiva que satisfaça o nível de melhora que os gestores querem.

## As dificuldades essenciais do software

O artigo gira principalmente na ideia dos problemas essenciais e acidentais do software. São chamados de problemas essenciais aqueles que acontecem por motivos de alguma limitação tecnológica ou algum problema de ferramenta, que podemos resolver com diferentes linguagens, ambiente diferente ou máquinas mais potentes. No entanto, os problemas essenciais já são mais profundos no software, e não são problemas possíveis de eliminar, já que todo software já está passível de problemas. O autor identifica principais quatro propriedades fundamentais como:

- **Complexidade:** Softwares são extremamente complexos, com inúmeros estados e interações, o que dificulta algumas qualidades muito importantes como comunicação, confiabilidade e manutenção.
- **Conformidade:** Sistemas precisam se adaptar a regras e interfaces dos tipos mais variados (dependendo do escopo do projeto), isto causa um aumentando na carga de complexidade.
- **Mutabilidade:** O software está sempre sujeito a mudança, seja por novas demandas dos usuários ou pela evolução do hardware.

- **Invisibilidade:** Ao contrário de construções físicas por exemplo, o software não possui uma representação física como um prédio, isto dificulta a visualização e o raciocínio da equipe de desenvolvimento.

Tendo essas características observadas, o software se torna muito difícil de se encontrar uma simples solução para que se resolva tudo de forma direta.

## Avanços e limites das inovações

Aqui Brooks analisa diversas inovações que já foram criadas no passado e novas que possuem potencial para serem futuras soluções, como linguagens de alto nível, programação orientada a objetos, inteligência artificial por exemplo. É inevitável de dizer que essas criações ajudaram no desenvolvimento de softwares, principalmente quando entendemos o conceito de “dificuldades acidentais”, no entanto essas criações não atingem o cerne do problema.

## Estratégias promissoras

Diante de tudo o que foi dito, é comum pensar que o autor tem uma visão pessimista, porém Brooks aponta também que o uso de softwares prontos que são apenas refinados através dos requisitos e prototipagem rápida, o desenvolvimento incremental e a criação de designers grandes, podem sim produzir uma solução superior as já pensadas, mostrando um ponto de vista também esperançoso sobre o assunto.

## Considerações finais

Podemos dizer que até hoje o artigo continua sendo valioso, já que extingue a ideia de uma simples solução milagrosa na Engenharia de Software, e destaca ainda mais valores como disciplina e maturidade e a iteração como sendo os principais caminhos para o progresso da área, afirmando que mesmo a inexistência de uma mágica “bala de prata” ainda é possível enxergar um caminho de aprimoramentos no desenvolvimento. O principal objetivo do autor é retomar o foco da discussão e realinhar expectativas dos gestores que buscam por uma solução mágica, o progresso virá de esforços consistentes de todos e boas práticas.

Em resumo, o artigo “No Silver Bullet: Essence and Accidents of Software Engineering” é um grande ponto de referencia para profissionais da área a focarem em abordagens e buscas por soluções realistas, encarando os limites que o próprio software possui.