

RESENHA - NO SILVER BULLET

O texto de Frederick P. Brooks Jr. busca comparar logo de início os projetos de software com monstros da ficção, os famosos lobisomens. Isso ocorre pois quando os projetos de software são mal gerenciados e mal executados, ocorrem diversos problemas que são aterrorizantes como: produtos não entregues até o prazo combinado, que não fazem o que deveriam e gastos além do mensurável. O artigo foca no porquê, mesmo após muitas décadas, ainda assim não se encontra uma bala de prata - solução confiável, simples e objetiva - para esses lobisomens.

Ademais, o artigo estabelece as dificuldades por trás de se encontrar soluções, e comparando com o rápido avanço do hardware de dois em dois anos (Lei de Moore) e como isso não se aplica aos softwares e a maioria das tecnologias humanas. Em segundo lugar, observa-se que o autor faz uma divisão entre as dificuldades enfrentadas pela tecnologia de software, as quais são as dificuldades de essência e as dificuldades acidentes. Após isso, ele deixa explícito que a essência está relacionada com as dificuldades inerentes à natureza do software (complexidade, conformidade, mutabilidade e invisibilidade), e acidentais, aquelas dificuldades que hoje acompanham sua produção, mas não são inerentes. Além disso, são enumerados avanços na tecnologia de software que atacam dificuldades acidentais, como Linguagens de Alto Nível, as quais são altamente confiáveis, simples e produtivas de se utilizarem e que encapsulam toda a complexidade de um programa, Time-Sharing, o qual preserva o imediatismo e melhora no tempo de resposta dos programas, e os Ambientes de Programação Unificados, que possibilitou o desenvolvimento de diversas ferramentas e bibliotecas integradas.

Para Brooks, a verdadeira esperança de progresso não reside em uma única tecnologia, mas em abordagens que atacam a essência do problema. Ele propõe três estratégias principais, na primeira ele afirma: "A solução mais radical possível para construir software é não construí-lo de forma alguma.", pois a quantidade de fornecedores oferecendo melhores aplicações aumentam gradativamente. Logo, ele considera mais barato comprar software a produzir um do zero. Já na segunda, é dito que a parte mais difícil e crítica do processo é a especificação dos requisitos, porque os clientes raramente sabem o que realmente querem. A prototipagem rápida permite um processo iterativo, onde um protótipo funcional é criado para que o cliente possa testar e refinar a especificação antes que o produto final seja construído. Por último, Brooks também defende a metáfora de "cultivar" o software, em vez de "construí-lo". A abordagem incremental, começando com um sistema mínimo e adicionando funcionalidades gradualmente, ajuda a gerenciar a complexidade e a manter o sistema funcionando em todas as etapas, aumentando o moral da equipe e a qualidade do produto.