Pior é melhor

Origem: Wikipédia, a enciclopédia livre.

Pior é melhor ou **estilo Nova Jersey** é o nome de uma técnica de desenvolvimento de software ou filosofia de design na qual a simplicidade na interface e na implementação é mais importante que qualquer outra propriedade do sistema (incluindo correção, consistência e completude).

Índice

Origem

O expert em Lisp Richard P. Gabriel criou o conceito em 1989 e apresentou-o em *Lisp: good news, bad news, how to win big* (Lisp: boas notícias, más notícias, como triunfar). Uma seção do artigo intitulada O surgimento de "*Pior é melhor*", foi amplamente difundida no início de 1991.

Descrição

- **Simplicidade**: o design deve ser simples tanto na implementação quanto na interface. No entanto, a simplicidade da implementação é mais importante que a da interface. **Simplicidade é a propriedade mais importante no design**.
- **Correção**: o design deve ser correto em todos os aspectos observáveis, mas **ser simples é ligeiramente melhor que ser correto**.
- **Consistência**: **o design não deve ser exageradamente inconsistent**e, mas em alguns casos a consistência pode ser sacrificada pela simplicidade. Para evitar inconsistências ou complexidade na implementação é preferível eliminar as partes do design que lidam com circunstâncias pouco comuns.
- **Completude**: o design deve incluir tantas situações importantes quanto for prático. Todos os casos que são razoavelmente esperados devem ser incluídos. **A completude pode ser sacrificada em favor de qualquer outra qualidade**, e de fato deve ser sacrificada sempre que arrisque a simplicidade da implementação. A consistência pode ser sacrificada para conseguir completude se simplicidade for conservada. Em especial, a consistência da interface é de pouco valor.

Contraste com a abordagem *MIT*

Além disso, Gabriel contrasta esta filosofia com a chamada *abordagem MIT* (também conhecida como *A Coisa Certa*), e afirma que *Pior é melhor* produz software mais bem sucedido. Sendo o programa inicial basicamente bom, fica mais fácil adaptá-lo para novas máquinas e situações, sua implementação inicial tomará muito menos tempo e esforço, e seu uso se difundirá muito antes. Uma vez distribuído sofrerá pressão para que sua funcionalidade seja melhorada, mas os usuários já terão sido condicionados a aceitar menos do que *A Coisa Certa*. "Portanto, o software pior-é-melhor ganhará aceitação primeiro, depois condicionará a seus usuários a esperar menos, e posteriormente será melhorado a um ponto que seja quase A Coisa Certa. Em termos concretos, ainda que em 1987 os compiladores de Lisp fossem tão bons quanto os de C, havia muito mais experts que queriam melhorar os compiladores de C do que os que queriam melhorar os compiladores de Lisp."

Outras ideias intimamente relacionadas são importantes na filosofia de design Unix e no movimento de código aberto.

O ensaio de Gabriel foi uma resposta à conferência "Mais é menos", um ataque ao design de software inchado. Nessa época, Unix e a linguagem de programação C tinham superado a Lisp como o ambiente de desenvolvimento dominante na comunidade de investigação em ciências computacionais, e as relações entre os laboratórios Bell e as comunidades de inteligência artificial do MIT eram altamente competitivas.

Bibliografia

- Lisp: Good News, Bad News, How to Win Big (http://web.archive.org/web/http://www.ai.mit.edu/docs/articles/good-news/good-news.html), Richard P. Gabriel.

Ligações externas

- Worse is Better (http://dreamsongs.com/WorseIsBetter.html), Richard P. Gabriel.
- The Rise of "Worse is Better" (http://www.jwz.org/doc/worse-is-better.html), Richard P. Gabriel.

Veja também

- Programador