

63. [Sebesta, 2000] Qual é a sua opinião a respeito do argumento segundo o qual linguagens demasiadamente complexas são muito perigosas de usar e que, portanto, devemos manter todas as linguagens pequenas e simples?

Se uma linguagem de programação for demasiadamente complexa, ou seja, possuindo um grande número de diferentes construções, alguns programadores podem não estar familiarizados com todas as construções disponibilizadas, o que pode levar ao erro inadequado de alguns recursos e ao desuso de outros que podem ser mais elegantes ou mais eficientes (ou ambos) do que aqueles usados. Pode, até mesmo, ser possível usar recursos desconhecidos acidentalmente, com resultados bizarros.

Portanto, linguagens de programação pequenas e mais simples, possuindo um pequeno número de construções primitivas e um conjunto consistente de regras para combiná-las é muito melhor do que, simplesmente, ter um grande número de primitivas. Por exemplo, a sintaxe da linguagem de programação C é uma complicada mistura de palavras na língua inglesa e álgebra, enquanto que a sintaxe da linguagem de programação LISP é um modelo de simplicidade, com o código de programa e os dados tendo exatamente a mesma forma: listas entre parênteses.

Uma crítica interessante em relação a complexidade de linguagens de programação foi proferida por Edsger Dijkstra, em 1972 na sua *Turing Award Lecture*, que fez uma das mais fortes críticas à complexidade da linguagem de programação PL/I: “Eu absolutamente não consigo ver como podemos manter nossos programas de crescimento firmemente dentro de nossa compreensão intelectual quando, por sua clara ‘barroquice’, a linguagem de programação – nossa ferramenta básica, imaginem! – já escapa de nosso controle intelectual”.