Programming = Mathematics + Murphy's Law (E. Dijkstra)

Rafael Castro

rafaelcgs10@gmail.com

Karina Roggia

karina.roggia@udesc.br

04/09/2019

Mini-bibliografia

Quem foi Edsger Dijkstra?

- Nome Completo: Edsger Wybe Dijkstra.
- Nacionalidade: Holandês.
- Longevidade: 1930 - 2002.

Alguns trabalhos de Edsger Dijkstra

- Algoritmos de grafos: Prim e Dijkstra.
- Sistema Operacional THE (BATCH, camadas de abstração e multitasking).
- Variável de semáforo.
- Trabalhou no primeiro compilador da linguagem ALGOL 60.
- Introduziu o conceito de pilha em programas recursivos.
- Pai da programação estruturada - Uso de subrotinas, laços de repetição etc.
- Defensor do uso de metodologias matemáticas na programação - Verificação formal.
- Escreveu diversos textos (EWDs).

Prêmios que Dijkstra ganhou

1. Fellow da British Computer Society em 1971.
2. Membro da Royal Netherlands Academy of Arts and Sciences em 1971.
3. ACM Turing Award em 1972.
4. American Academy of Arts and Sciences em 1975.
5. Fellow da Association for Computing Machinery 1994.

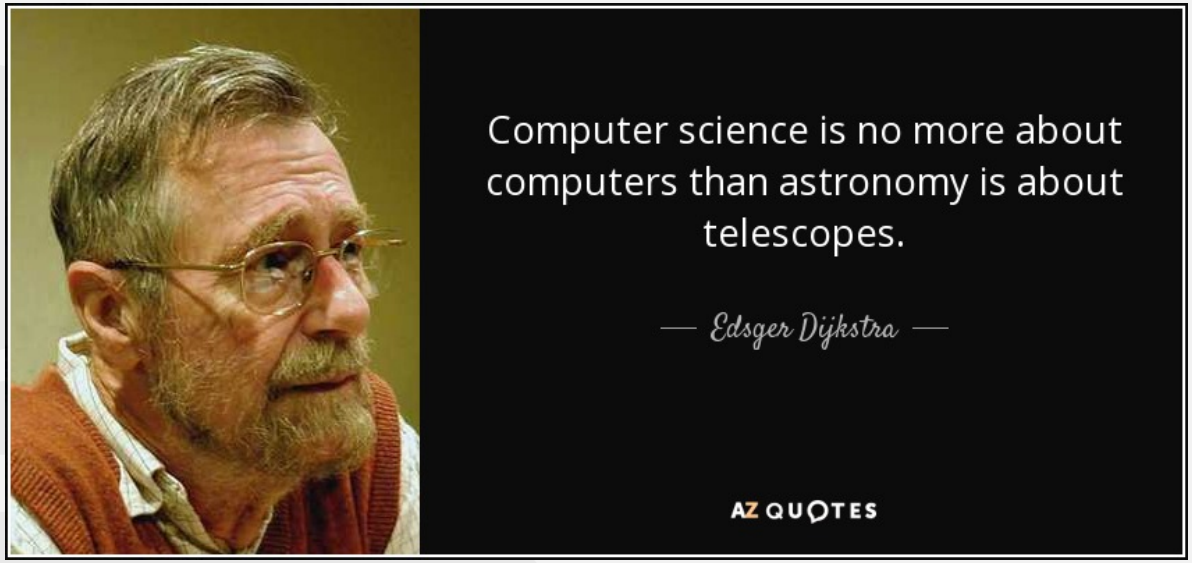
Frases de Dijkstra

Divisão

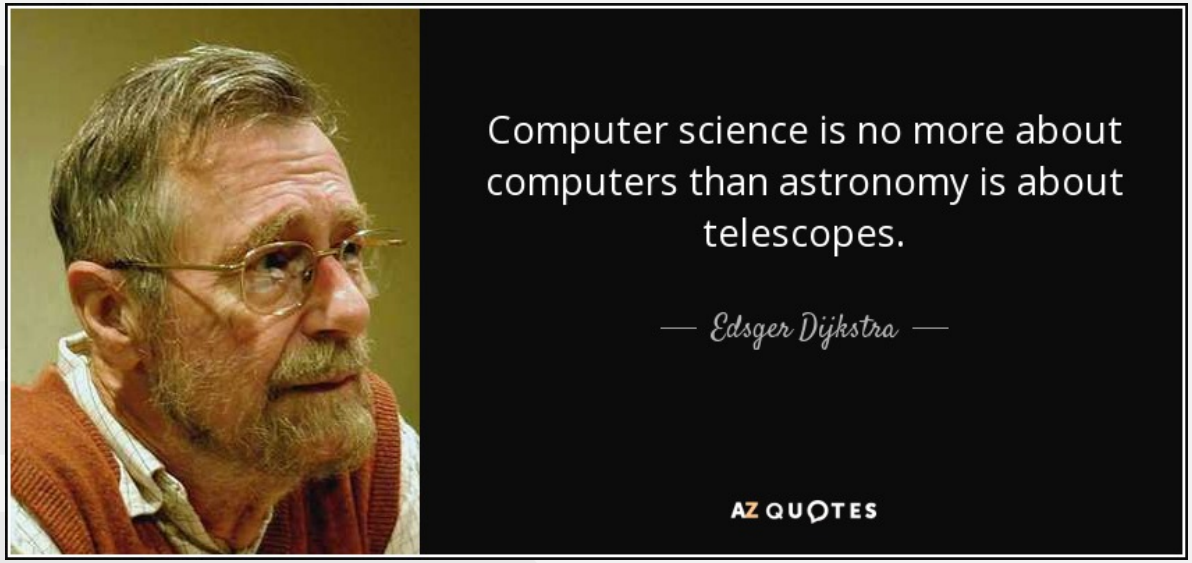
- Frases que todos citam, mas que ninguém sabe o contexto
- Frases falando mal de Engenharia de Software
- Frases falando mal do Ensino de C. da Computação
- Frases falando mal de linguagens de programação
- Frases em defesa dos métodos formais

Frases que todos citam, mas que ninguém sabe o contexto

Ciência da Computação é sobre ...

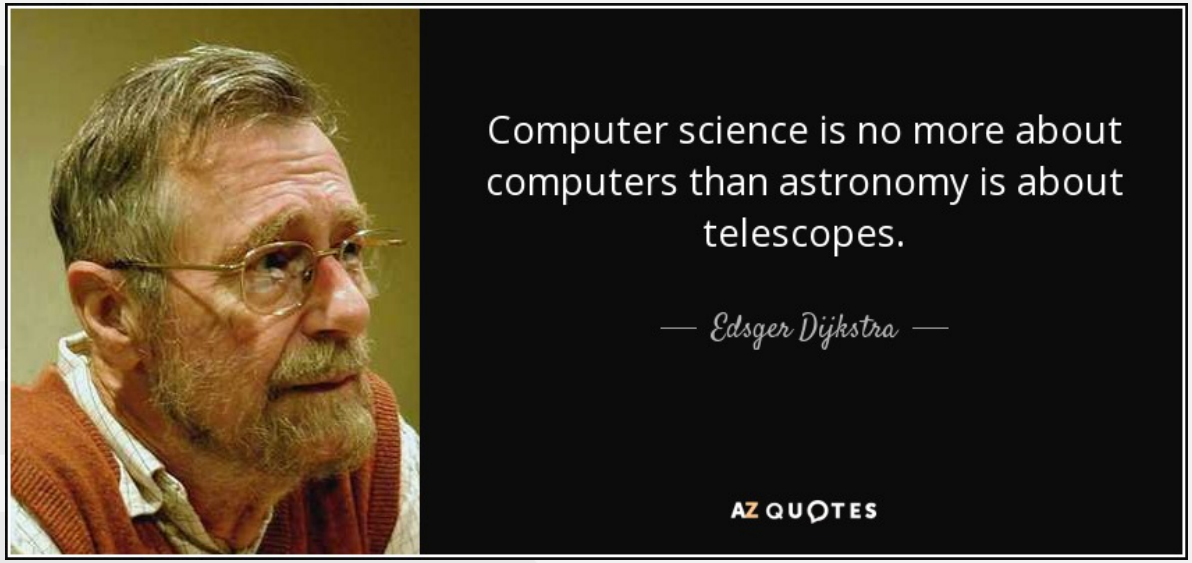


Ciência da Computação é sobre ...



Não é do Dijkstra!

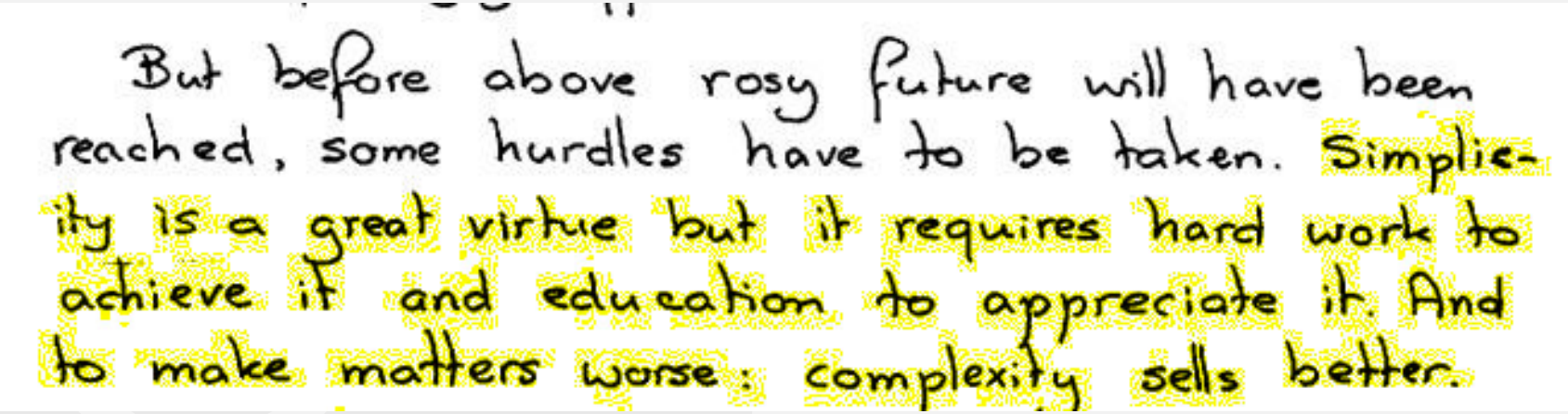
Ciência da Computação é sobre ...



Não é do Dijkstra!

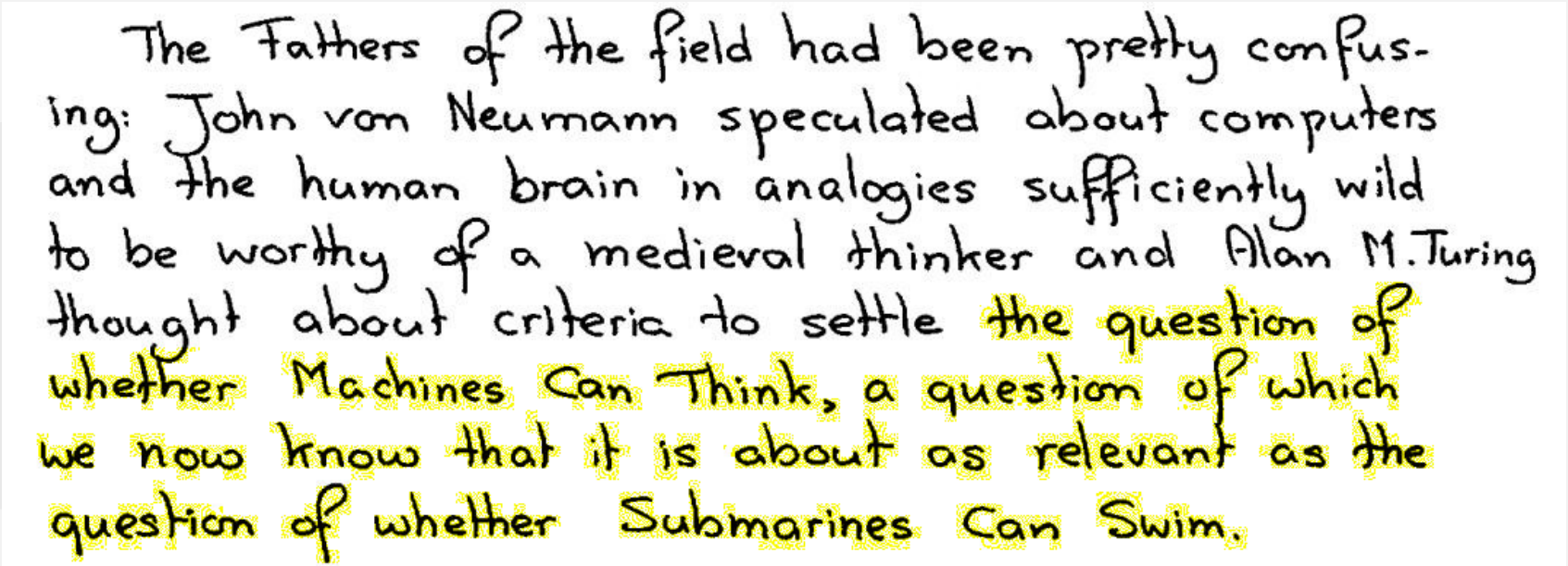
Michael R. Fellows, Ian Parberry (1993) "SIGACT trying to get children excited about CS"

Simplicidade é uma grande virtude...


But before above rosy future will have been reached, some hurdles have to be taken. Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better.

- On the nature of computing science - EWD 896 - 10 de Agosto de 1984
- O texto questiona a natureza da Ciência da Computação.
- Contexto: "a computação vai curar todos os males do mundo" - a busca pela Pedra Filosofal.
- Como a computação se afirma como uma disciplina acadêmica? Através da matemática.
- Se é simples, não é relevante: rejeitaram um artigo do Dijkstra porque a solução era muito simples.

A questão sobre se computadores podem pensar...



- The threats to computing science - EWD 898 - 16 de Novembro de 1984
- O texto discute sobre as ameaças para Ciência da Computação
- Mesmos os pais da C. da Computação estavam bem confusos.
- Qual era o foco da C. da Computação?
  e.g. conseguir computadores que funcionam.

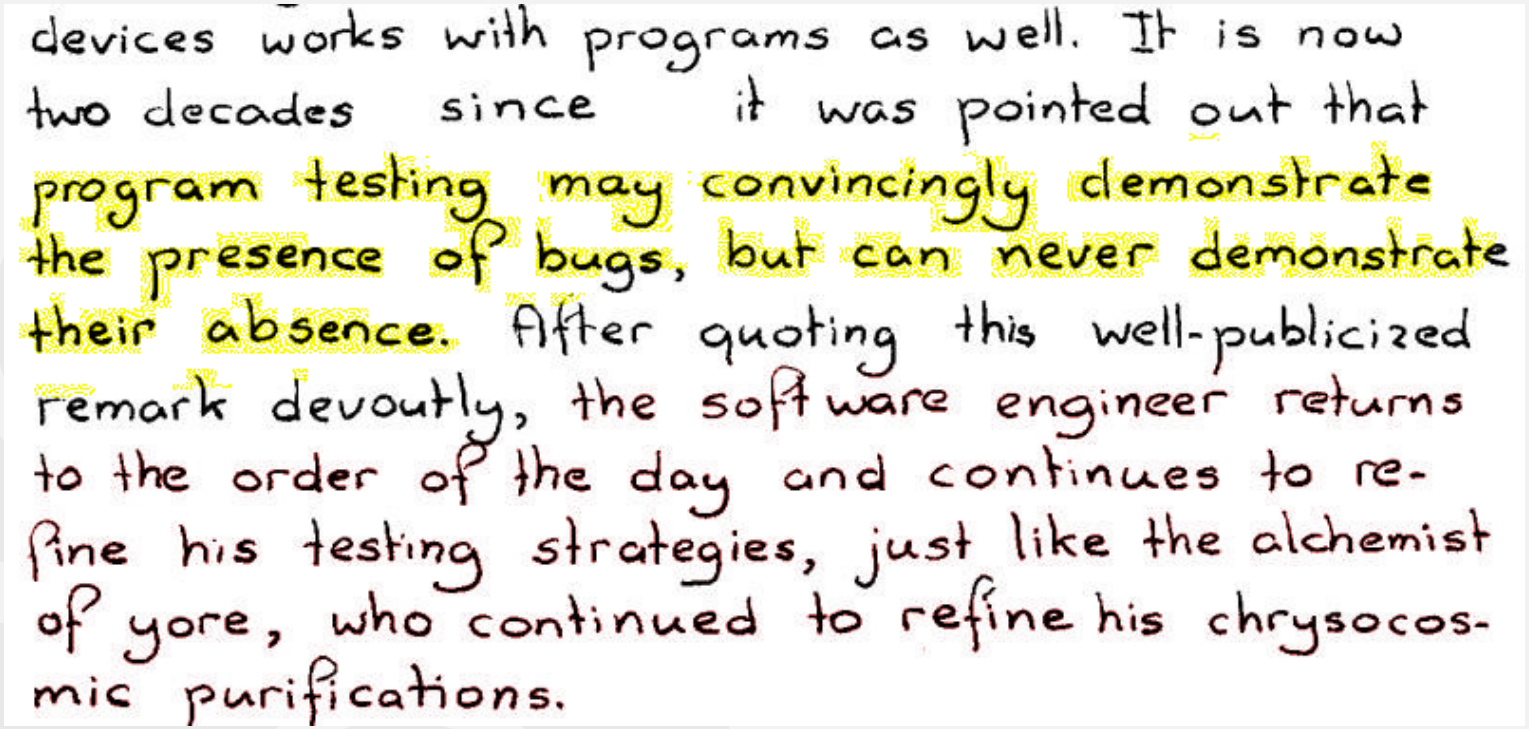
Testes mostram a presença de bugs...

Dijkstra: Testing shows the presence, not the absence of bugs.

- Software Engineering Techniques, April 1970 - Report on a conference sponsored by the NATO Science Committee

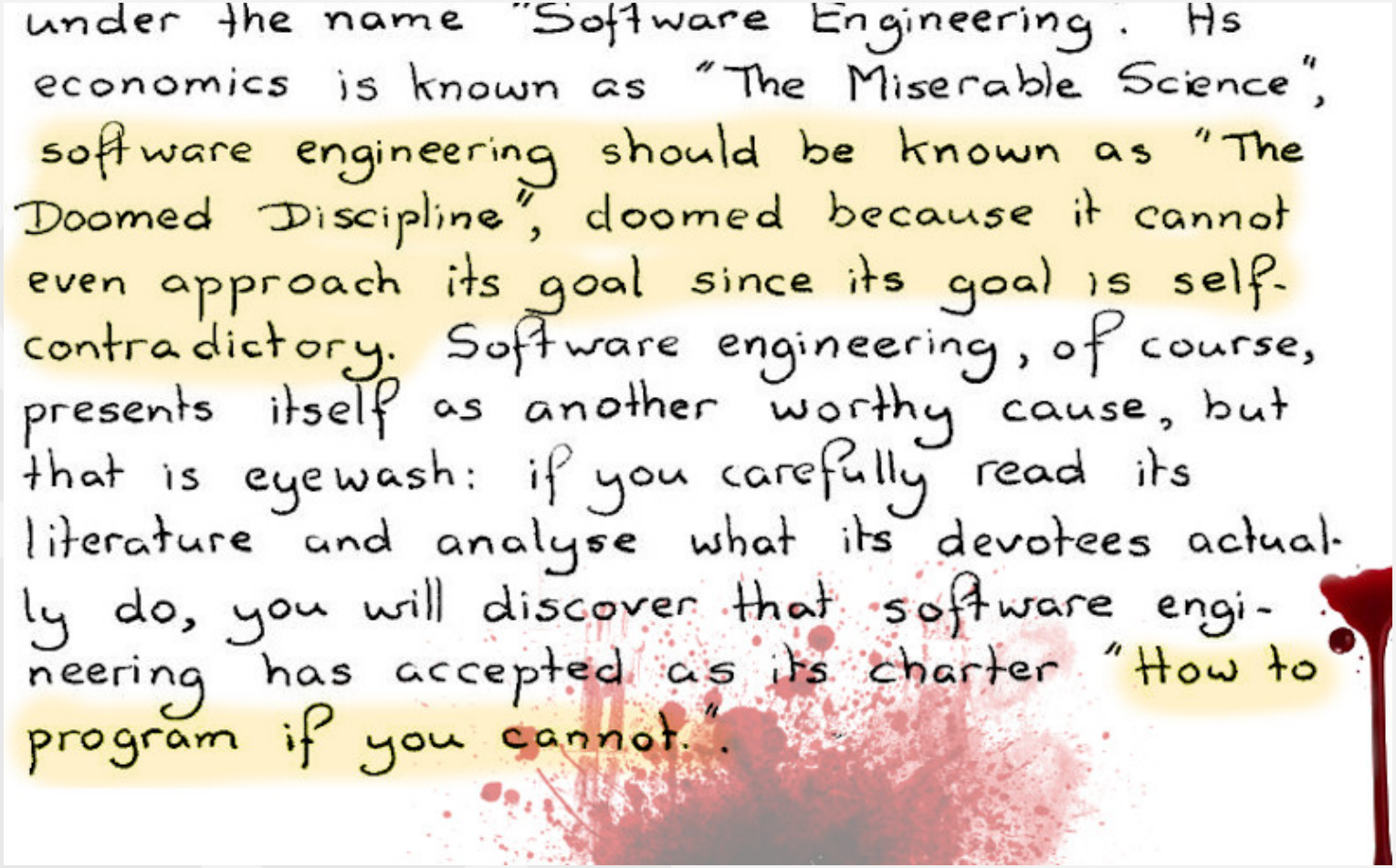
Frases falando mal de Engenharia de Software

Testes mostram a presença de bugs...


devices works with programs as well. It is now two decades since it was pointed out that program testing may convincingly demonstrate the presence of bugs, but can never demonstrate their absence. After quoting this well-publicized remark devoutly, the software engineer returns to the order of the day and continues to refine his testing strategies, just like the alchemist of yore, who continued to refine his chrysocosmic purifications.

- On the cruelty of really teaching computing science - EWD 1036 - 2 de Dezembro de 1988
- Programas não funcionam como máquinas.
  Programação não é uma engenharia.
- Computação é uma novidade radical.

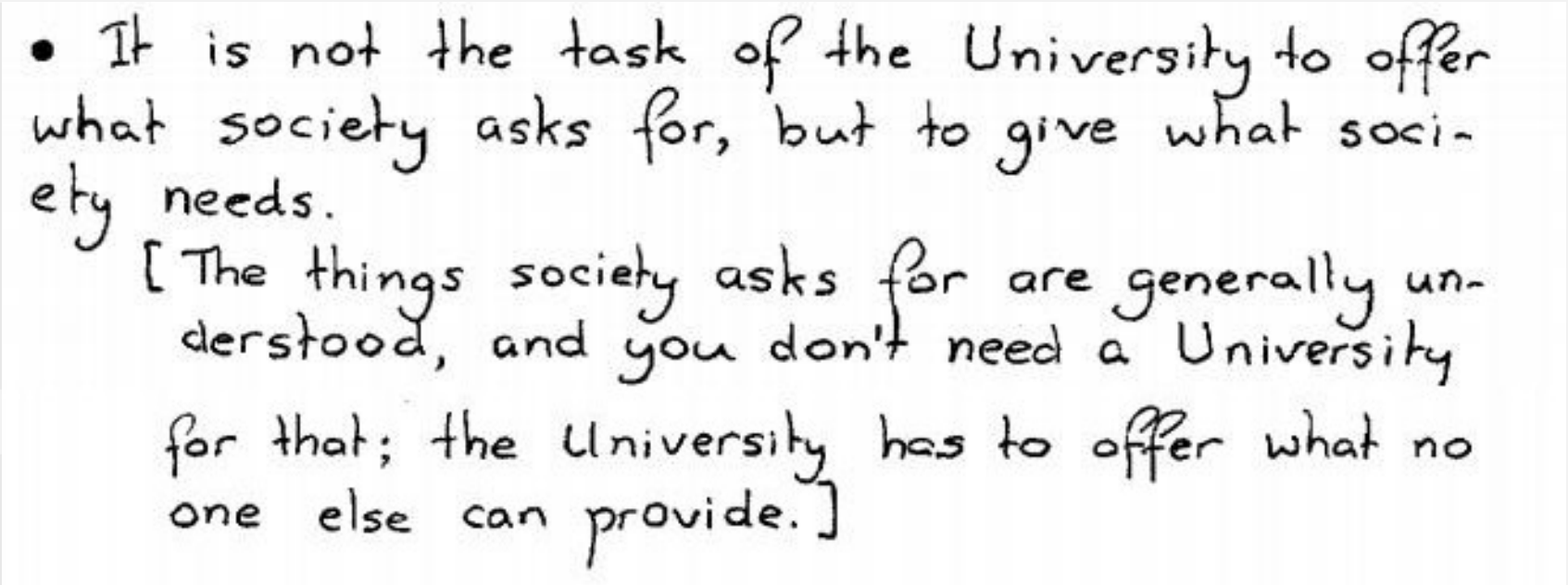
A disciplina condenada


under the name "Software Engineering". Hs economics is known as "The Miserable Science", software engineering should be known as "The Doomed Discipline", doomed because it cannot even approach its goal since its goal is self-contradictory. Software engineering, of course, presents itself as another worthy cause, but that is eyewash: if you carefully read its literature and analyse what its devotees actually do, you will discover that software engineering has accepted as its charter "How to program if you cannot.".

- On the cruelty of really teaching computing science - EWD 1036 - 2 de Dezembro de 1988

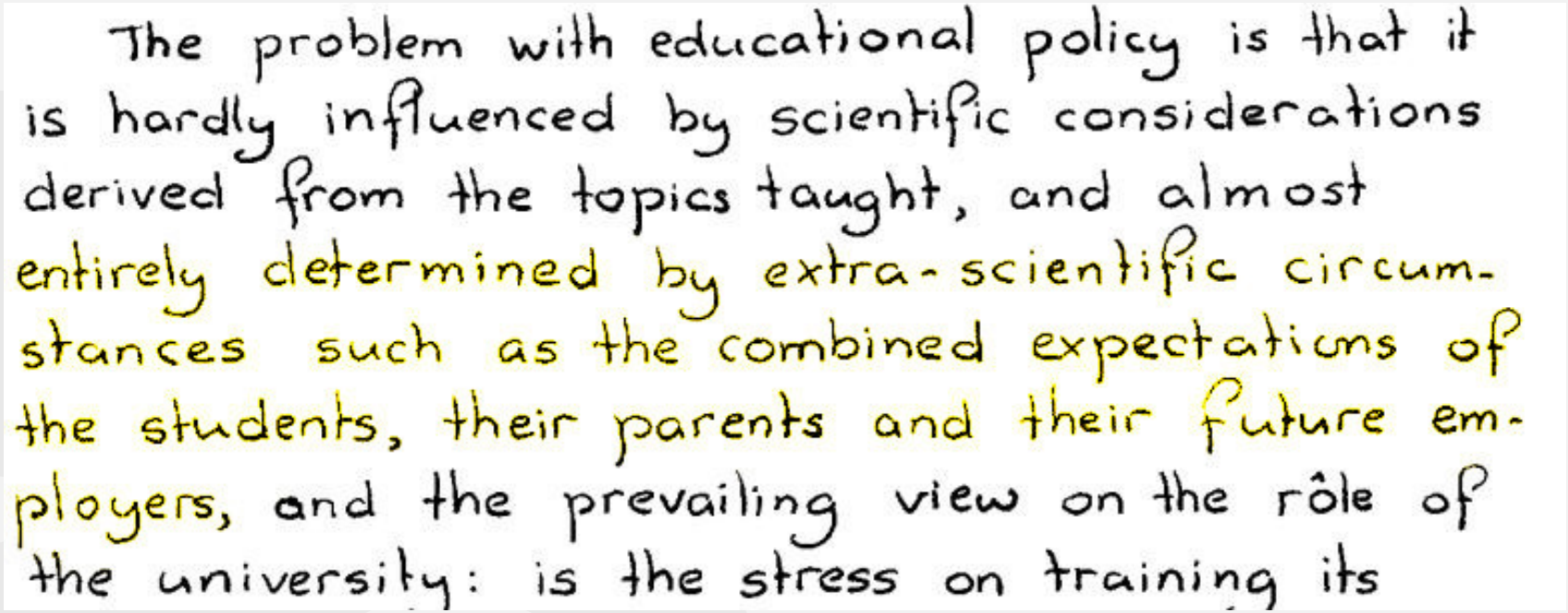
Frases falando mal do Ensino de C. da Computação

O objetivo da Universidade



- Answers to questions from students of Software Engineering - EWD 1305 - 28 de Novembro de 2000

Expectativas erradas


The problem with educational policy is that it is hardly influenced by scientific considerations derived from the topics taught, and almost entirely determined by extra-scientific circumstances such as the combined expectations of the students, their parents and their future employers, and the prevailing view on the rôle of the university: is the stress on training its

- On the cruelty of really teaching computing science - EWD 1036 - 2 de Dezembro de 1988
- O expectativas são contra realmente ensinar C. da Computação.
- Cada setor quer que C. da Computação seja uma coisa.

Frases falando mal de linguagens de programação

Dijkstra sobre Basic e Cobol

It is practically impossible to teach good programming to students that have had a prior exposure to BASIC: as potential programmers they are mentally mutilated beyond hope of regeneration.

The use of COBOL cripples the mind; its teaching should, therefore, be regarded as a criminal offence.

- EWD 498
- Ferramentas que utilizamos influência a nossa forma de pensar.

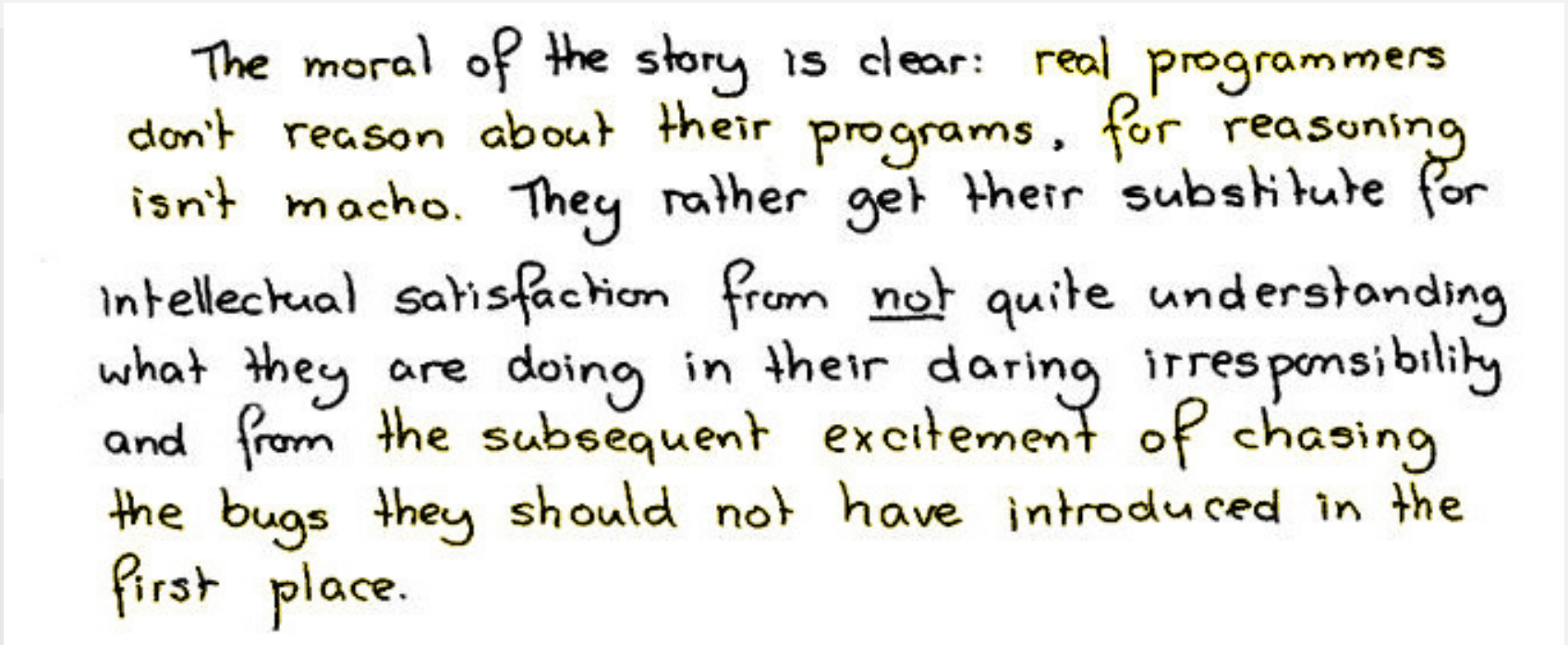
Dijkstra sobre Lisp

The third project I would not like to leave unmentioned is LISP, a fascinating enterprise of a completely different nature. With a few very basic principles at its foundation, it has shown a remarkable stability. Besides that, LISP has been the carrier for a considerable number of in a sense our most sophisticated computer applications. LISP has jokingly been described as "the most intelligent way to misuse a computer". I think that description a great compliment because it transmits the full flavour of liberation: it has assisted a number of our most gifted fellow humans in thinking previously impossible thoughts.

- The Humble Programmer - EWD 340 - 1972
- Lisp é a primeira linguagem do paradigma funcional. Criada por John McCarthy em 1958.

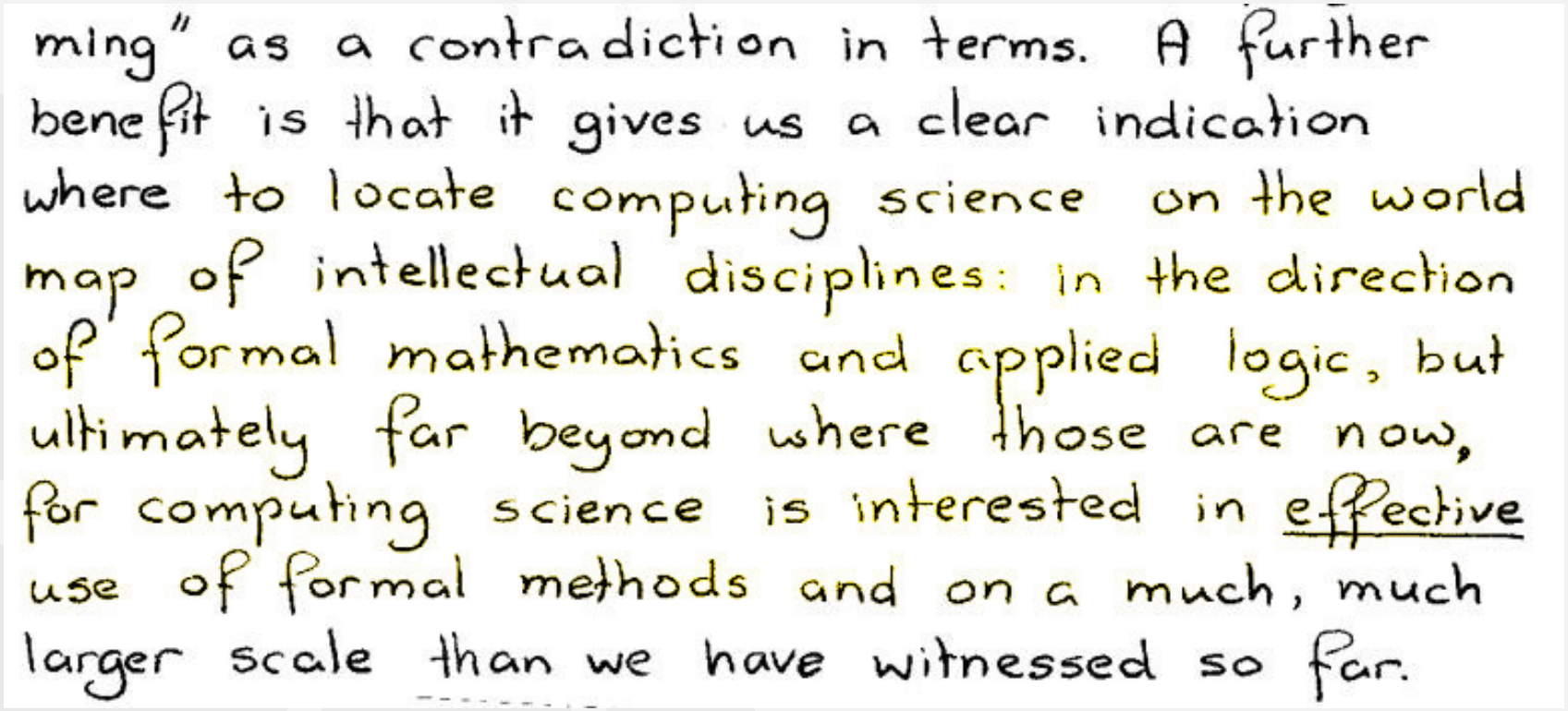
Frases em defesa dos métodos formais

Programadores de verdade não provam


The moral of the story is clear: real programmers don't reason about their programs, for reasoning isn't macho. They rather get their substitute for intellectual satisfaction from not quite understanding what they are doing in their daring irresponsibility and from the subsequent excitement of chasing the bugs they should not have introduced in the first place.

- Real mathematicians don't prove - EWD1012 - 24 de Janeira de 1988

Onde está a C. da Computação?


ming" as a contradiction in terms. A further benefit is that it gives us a clear indication where to locate computing science on the world map of intellectual disciplines: in the direction of formal mathematics and applied logic, but ultimately far beyond where those are now, for computing science is interested in effective use of formal methods and on a much, much larger scale than we have witnessed so far.

- On the cruelty of really teaching computing science - EWD 1036 - 2 de Dezembro de 1988
- Defesa que programação **é** elaborar fórmulas matemáticas.

Programming = Mathematics + Murphy's Law

There is no doubt that programming, viewed as a mathematical activity, has to be a very (if not totally) formal one. Mathematical theorems intended for human usage are appealed to in a modest frequency and, moreover, by mathematicians, who are wise enough not to appeal to a theorem in a situation in which it obviously does not hold. A programmer's theorem that a given repetition maintains a given invariant, however, is appealed to in high frequency ~thanks to mechanization easily up to 1000 Herz~ and, moreover, these appeals take place without a wise mathematician as witness. In this sense, Programming = Mathematics + Murphy's Law. (The latter states "Anything that can go wrong will go wrong.".)

- What Computing Science is about - EWD 1008 - 25 de Maio de 1987
- Dado a frequência das repetições, se pode dar errado, então vai.

Conclusão

Conclusão