



UNIVERSIDADE  
FEDERAL  
DO PARÁ

# Engenharia de Software

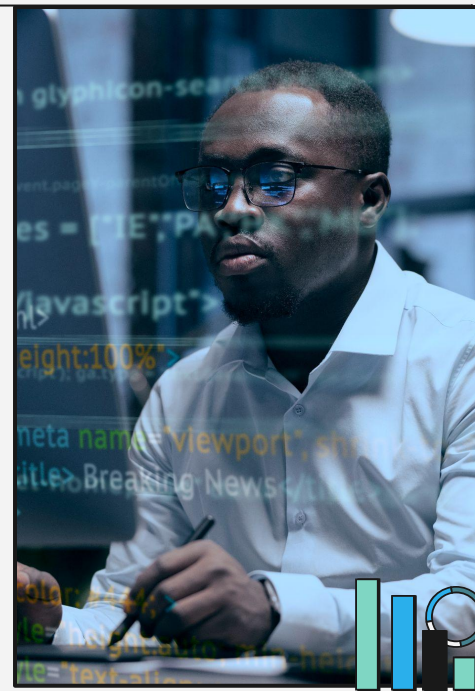
Prof. Allan Veras  
Faculdade de Computação - FACOMP



# Contexto e História

Historicamente, a área surgiu no final da década de 60 do século passado.

- Nas duas décadas anteriores, os primeiros computadores modernos foram projetados e começaram a ser usados principalmente para resolução de problemas científicos.
- Neste contexto - **software não era uma preocupação central.**





# Conferência da OTAN : 1968

O objetivo da conferência era chamar a atenção para um problema crucial do uso de computadores, o chamado software.

- Apontou-se a necessidade de que software fosse construído com **base em princípios práticos e teóricos**, tal como ocorre em ramos tradicionais e bem estabelecidos da **Engenharia**



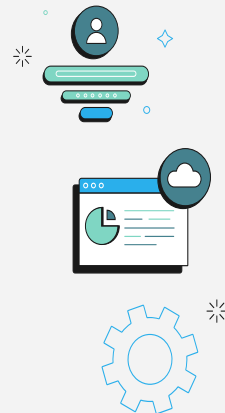


Segundo Frederick Brook, prêmio Turing em Computação (1999) e um dos pioneiros da área de Engenharia de Software, existem dois tipos de dificuldades em desenvolvimento de software: **dificuldades essenciais** e **dificuldades acidentais**.

# As dificuldades essenciais

## Complexidade

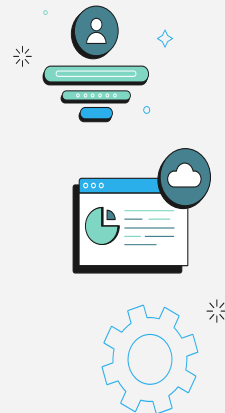
Dentre as construções que o homem se propõe a realizar, software é uma das mais desafiadoras e mais complexas que existe. Mesmo construções de engenharia tradicional, como um satélite, uma usina nuclear ou um foguete, são cada vez mais dependentes de software.



# As dificuldades essenciais

## Conformidade

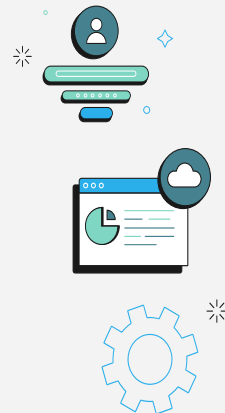
Pela sua natureza software tem que se adaptar ao seu ambiente, que muda a todo momento no mundo moderno. Por exemplo, se as leis para recolhimento de impostos mudam, normalmente espera-se que os sistemas sejam rapidamente adaptados à nova legislação. ***Brooks comenta que isso não ocorre, por exemplo, na Física, pois as leis da natureza não mudam de acordo com os caprichos dos homens***



# As dificuldades essenciais

## Facilidade de mudanças

Consiste na necessidade de evoluir sempre, incorporando novas funcionalidades. Na verdade, quanto mais bem sucedido for um sistema de software, mais demanda por mudanças ele recebe.



# As dificuldades essenciais

## Invisibilidade

Devido à sua natureza abstrata, é difícil visualizar o tamanho e consequentemente estimar o esforço de construir um sistema de software.

