



# **Análise e Desenvolvimento de Sistemas**

Prof. Vinícius Alves Hax



# Na aula anterior

- Gerência de projetos



# Na aula de hoje

- Introdução à qualidade de software



# Qualidade de software



O que é qualidade?

Qualidade é subjetivo?

















Mas voltando pra informática ...

# Linguagens de programação exóticas

- Brainf\*ck

```
++++++[>++++[>+>+++>+>+>+<<<←]>+>+>→>+[<]←]>>.>---.+++  
++++..++++.>>.<.<..++  
+.------.------.>>+.>++.
```

- Whitespace



# 4 languages ...

- Ruby

```
(1..4).each { |n| puts n }
```

- Javascript

```
for (let n = 1; n <= 4; n++)  
{  
    console.log(n);  
}
```

- C

```
#include <stdio.h>  
  
int main() {  
    for (int n = 1; n <= 4; n++) {  
        printf("%d\n", n);  
    }  
}
```

- Python

```
for n in range(1, 5):  
    print(n)
```



# Critérios


- Comparar requer critérios
- Alguns critérios:
  - Intuitividade
  - Otimização
  - Concisão
  - Legibilidade



**Tabela 24.1** Atributos de qualidade de software

Segurança	Compreensibilidade	Portabilidade
Proteção	Testabilidade	Usabilidade
Confiabilidade	Adaptabilidade	Reusabilidade
Resiliência	Modularidade	Eficiência
Robustez	Complexidade	Capacidade de aprendizado

Fonte: Sommerville (2011)



# Engenharia de software



# Aquedutos Romano





# Engenharia de software

- Diferente de outros ramos da engenharia, a engenharia de software é muito mais recente
- Nos anos 60 os computadores comerciais se tornaram mais populares e com isso surgiram softwares mais complexos
  - Necessidade de organizar melhor os softwares

# Exemplo: Código da Apollo 11

- Feito em Assembly
- Hardware
  - 72 KB de memória ROM e 4KB de memória RAM
  - Processador com 2 MHz
- O código do módulo lunar tem aproximadamente 145 000 linhas de código





# Critérios mais objetivos

- Quantidade de bugs
- Tempo para resolver os bugs
- Impacto dos bugs
- Quantidade de características implementadas
- Performance na execução
  - Frames por segundo, tempo (de execução, carregamento, etc), uso de memória

# Critérios mais objetivos

- Equipe de teste
  - Quantidade de bugs
  - Tempo para resolver os bugs
  - Impacto dos bugs
- Equipe de desenvolvimento / projeto
  - Quantidade de características implementadas
  - Performance na execução
    - Frames por segundo, tempo (de execução, carregamento, etc), uso de memória, etc

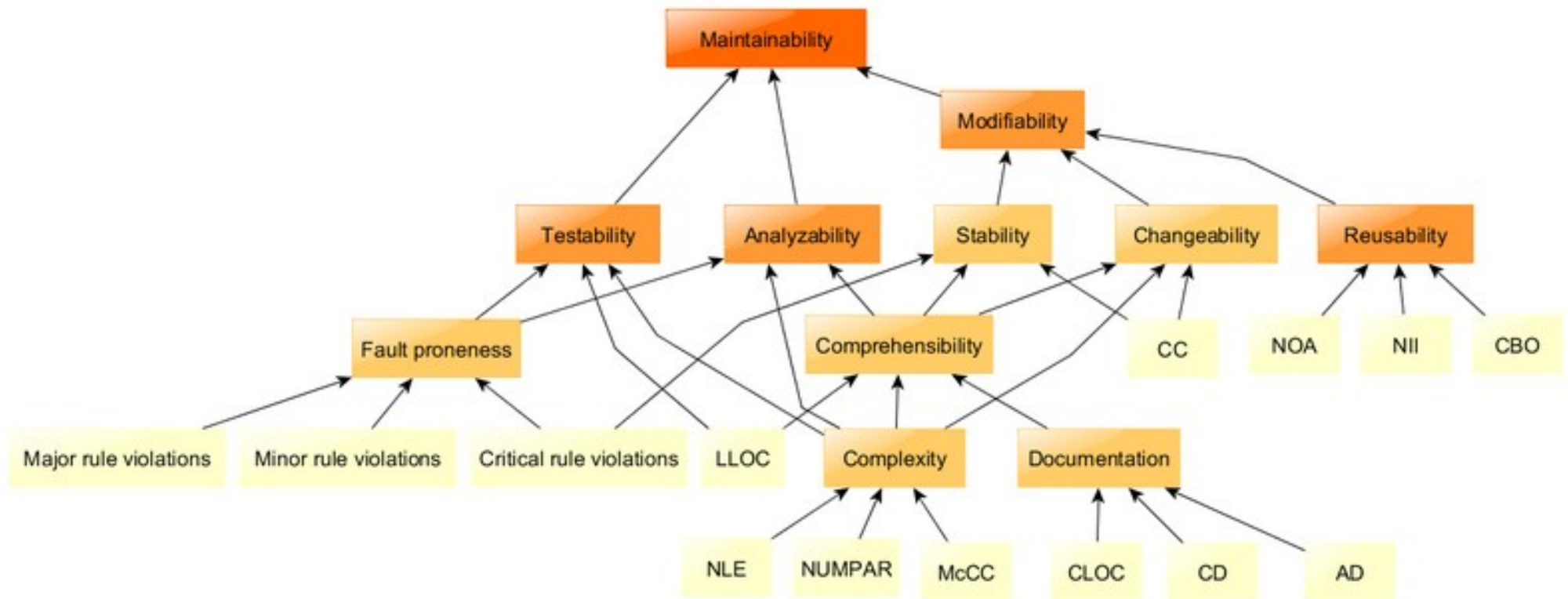


# Padrões de qualidade de software

- Existem inúmeros padrões, ferramentas, métricas, etc. Ex:
  - ISO 25010
  - CMMI

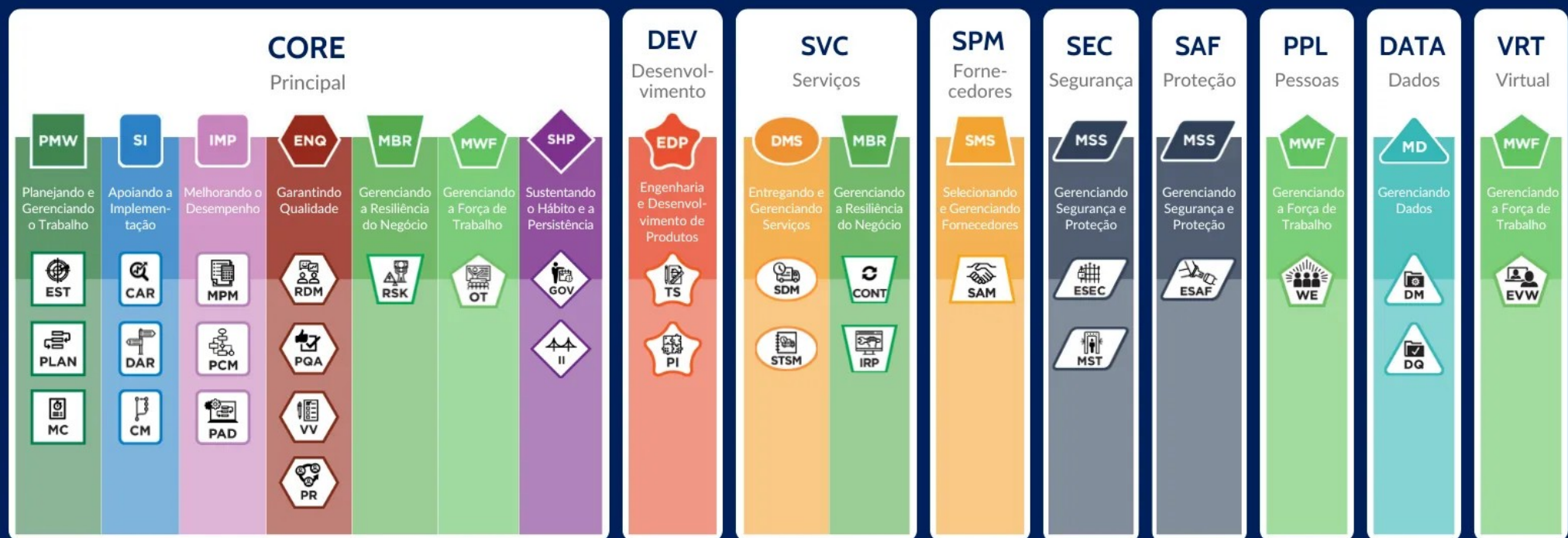


# ISO 25010



# Capability Maturity Model Integration (CMMI)

## Áreas de Prática do CMMI



# CMMI (2)

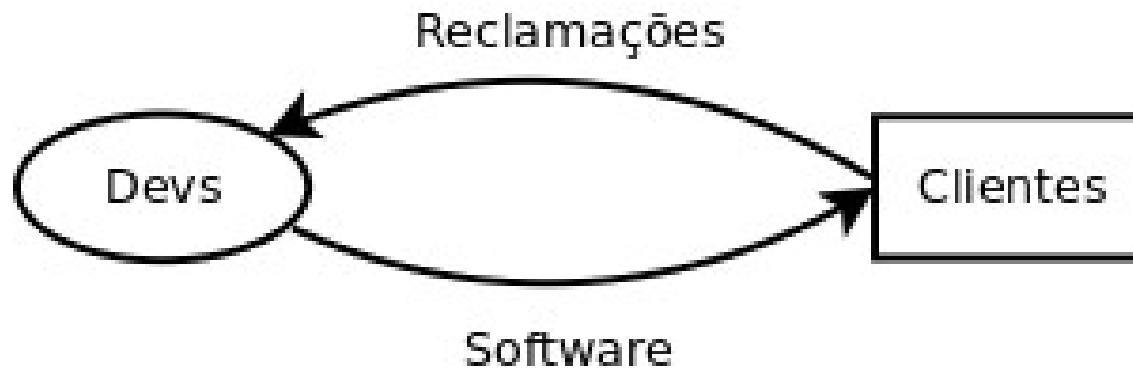
## Níveis de Maturidade CMMI





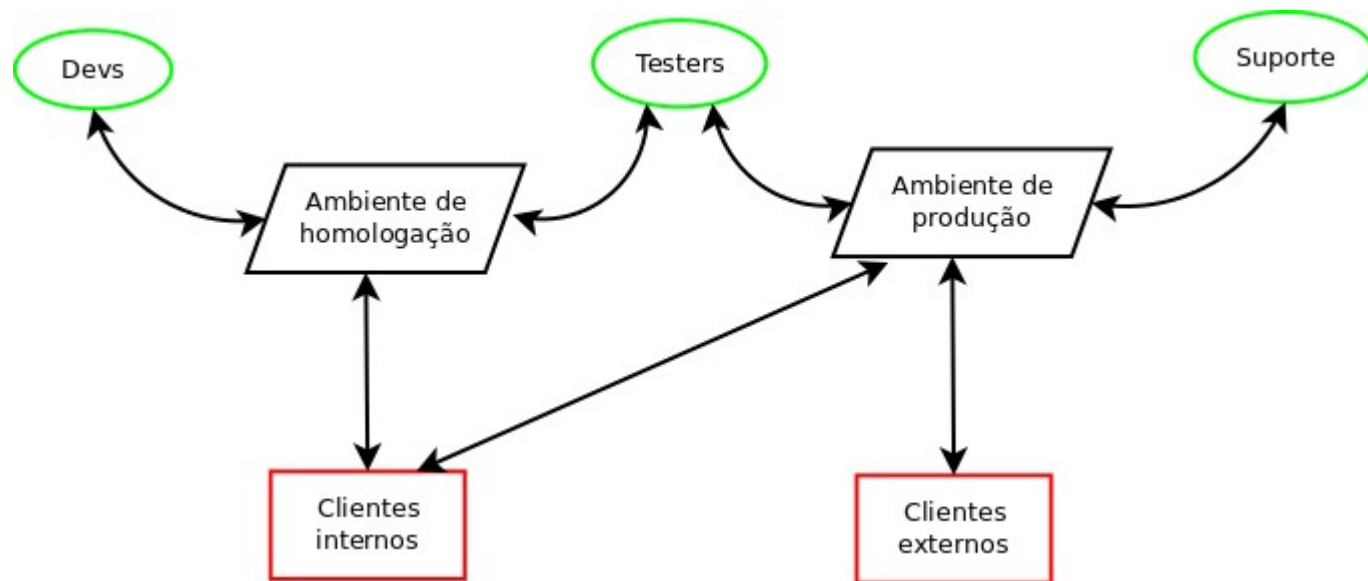
# Ambientes de desenvolvimento

Equipes/projetos menores



# Ambientes de desenvolvimento

Equipes/projetos maiores



# Referências

- Academia de Atenas, Noite estrelada e Abaporu Fonte: Wikipédia
- Linguagens ‘esotéricas’  
[https://en.wikipedia.org/wiki/Esoteric\\_programming\\_language](https://en.wikipedia.org/wiki/Esoteric_programming_language)
- Aquedutos romanos  
[https://pt.wikipedia.org/wiki/Aqueduto\\_romano](https://pt.wikipedia.org/wiki/Aqueduto_romano)
- “Quantas linhas de código tinha o software da Apollo 11?”  
<https://chatgpt.com> Resposta gerada em 12/12/2024
- ISO 25010  
[https://en.wikipedia.org/wiki/ISO/IEC\\_9126#Developments](https://en.wikipedia.org/wiki/ISO/IEC_9126#Developments)
- CMMI <https://promovesolucoes.com/cmmi-o-que-e-e-como-usar/>