

# ENGENHARIA de software



## CICLO DE VIDA DE SOFTWARE

**IMPORTANTE!**

- Ciclo de vida: fases de um produto de software de quando é concebido até não estar mais disponível para uso:



## ASPECTOS GERAIS

- Aplicação de uma abordagem sistemática, disciplinada e quantificável de desenvolvimento, operação e manutenção de software.
- Aborda questões técnicas e não-técnicas
- Meta = Desenvolver software com um bom custo-benefício
- Camadas: Ferramentas, Métodos, Processos com foco na qualidade

## PRINCÍPIOS

- Formalidade (Deve-se seguir passos definidos)
- Abstração (Considerar só pontos principais, desconsiderando os detalhes)
- Decomposição (Divide-se o problema em partes)
- Generalização (Resolução genérica, para poder ser aproveitada em casos semelhantes)
- Flexibilização (Permite a alteração do software sem causar problemas a sua execução)

## OUTRAS VERSÕES



# ENGENHARIA de software

=PROCESSOS DE SOFTWARE=

## PRINCIPAIS MODELOS

⚠ ATENÇÃO!

### MODELOS SEQUENCIAIS



### MODELOS ITERATIVOS

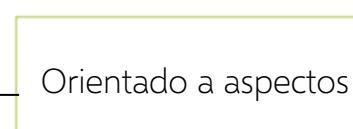
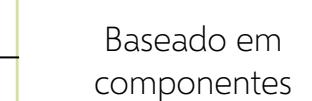


(*Rapid Application Development*)

(*Rational Unified Process*)

(*Test Driven Development / Acceptance Test Driven Development*)

### DEMAIS MODELOS



Exploratória/  
Evolucionária

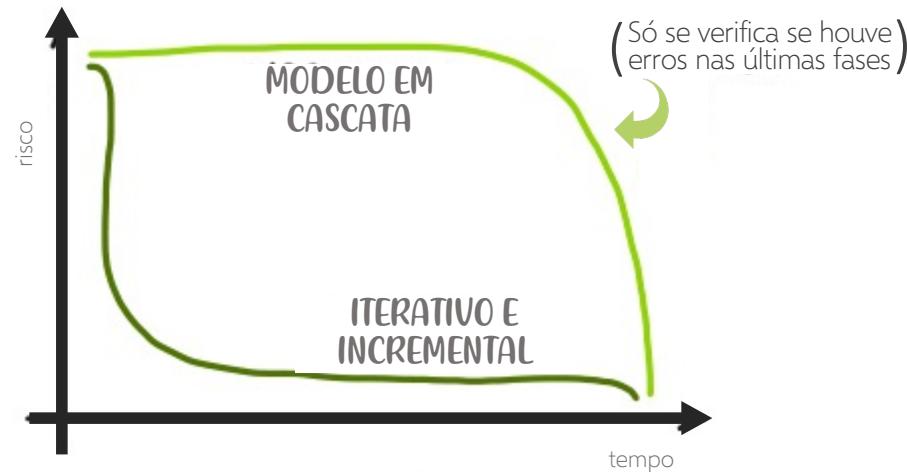
Throw-away/  
Descartável

# ENGENHARIA de software

= MODELO EM CASCATA =

## ASPECTOS GERAIS

- Modelo clássico, sequencial, linear, tradicional, *waterfall*, rígido e monolítico
  - Encadeamento simples entre as fases: uma fase só se inicia após o término e aprovação da fase anterior.
- Desvantagem: há um atraso na redução de riscos



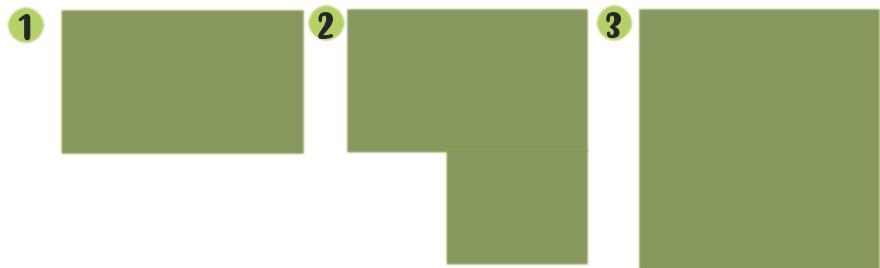
- Deve ser **utilizado** (preferencialmente) quando:
  - Os requisitos forem bem compreendidos
  - Houver pouca probabilidade de mudanças radicais

## MODELOS ITERATIVOS E INCREMENTAIS

- **Vantagem:** O cliente pode receber/ avaliar a entrega do produto mais cedo (Ainda no início do desenvolvimento do software)

### Modelo incremental

- Produz **BUILDS** (Partes do software)



### Modelo iterativo:

- Produz **RELEASES** (Versões constantemente melhoradas)

