

# Introdução à Engenharia de Software

Everson Santos Araujo  
everson@everson.com.br

## Crise do Software

- *Anos 60 surgimento de novos Sistemas Operacionais e equipamentos*
- *Necessidade de criar grandes sistemas de Software*
- *Surgimento da Engenharia de Software*

## Custos

- *O custo de desenvolvimento de Software não diminuem, como os custos de equipamentos*
- *Desenvolvimento é geralmente realizado para fins específicos, sem respeitar estimativas de tempo e valor*

## Software

- *Conjunto de instruções organizadas com a finalidade de desempenhar funções específicas*
- *Para Engenharia: Produto a ser vendido*
- *Um produto de software é sistematicamente destinado a ser utilizado por terceiros, com formações e experiências diferentes*

## Produto Software

- *Deve ser concebido e desenvolvido como resultado de um trabalho minucioso de engenharia*
- *Software não se desgasta*
- *Em sua maioria, são feitos sob medida para uma empresa ou problema específico*

## Problemas

- *Porque é necessário tanto tempo para criar um software?*
- *Porque um software custa tanto?*
- *Porque todo software possui erros desconhecidos?*
- *Porque não se consegue determinar a porcentagem de trabalho realizado e a realizar durante a criação de um Software?*

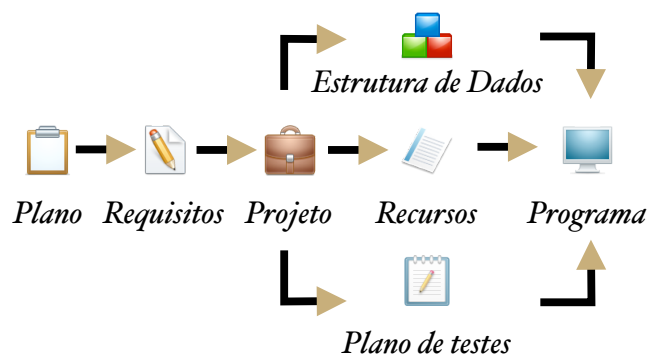
## Engenharia de SW

- *“O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais.” (Fritz Bauer)*

## Objetivos

- *Utilizar métodos, técnicas e ferramentas para melhorar a qualidade do software e aumentar a produtividade e satisfação profissional de produtores de software e clientes*

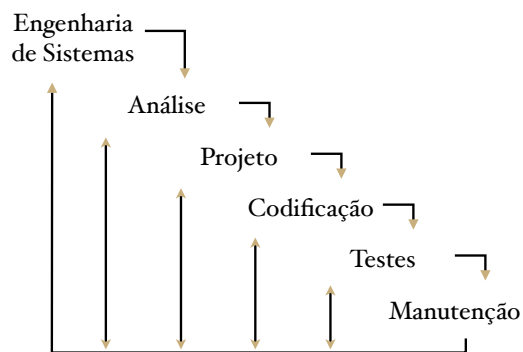
# Componentes de SW



# Modelos

- “Um modelo de desenvolvimento corresponde a uma representação abstrata do processo de desenvolvimento que vai, em geral, definir como as etapas relativas ao desenvolvimento do software serão conduzidas e interrelacionadas para atingir o objetivo do desenvolvimento que é a obtenção de um produto de software de alta qualidade a um custo relativamente baixo.” (MAZZOLA, Vitorio Bruno)

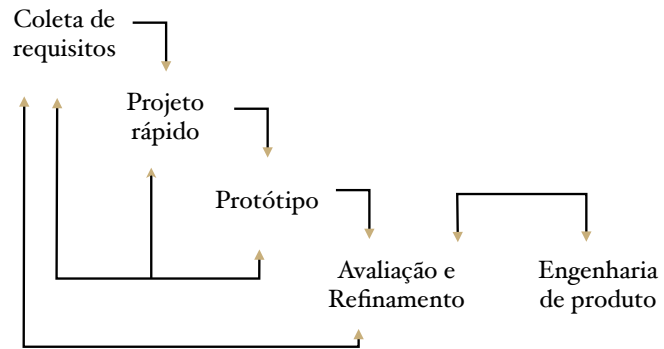
# Clássico



# Clássico

- Na prática, projetos não seguem o fluxo sequencial
- Acomodações de incertezas no início do projeto se tornam difíceis de gerenciar nas etapas seguintes
- Só existe algo funcional nos últimos estágios do projeto

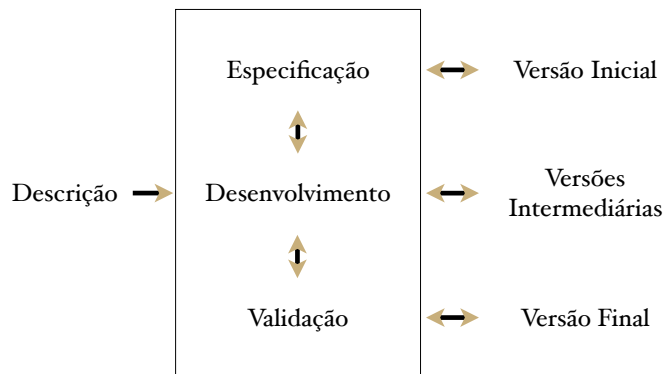
# Protótipo



# Protótipo

- *Permite perceber aspectos visíveis para o usuário*
- *Permite adequar o protótipo às necessidades do usuário*
- *O protótipo pode ser descartado ou utilizado*
- *Decisões e soluções improvisadas acabam por chegar ao produto final*

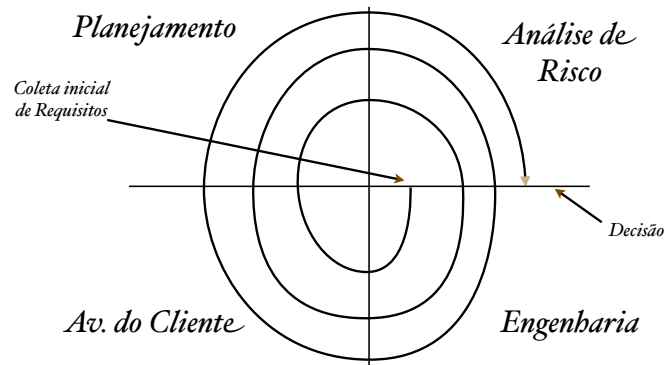
# Evolucionário



# Evolucionário

- *Incorporação de diferente partes e criação de diferentes versões*
- *Inclui protótipos*
- *Permite desenvolvimento exploratório*

# Espiral



# Espiral

- *Paradigma mais realístico*
- *Metamodelo*
- *Incorpora análise de riscos*
- *Permite prototipação variada*
- *Precisa de maior controle e gerenciamento para evitar perda de prazos*