

---

# Capítulo 20 - Manutenção de Software

Os Fatores de Qualidade de Software focalizam três aspectos importantes do Software Produto: (ISO 9126)



---

# Manutenibilidade

❖ A Manutenibilidade pode ser definida qualitativamente como a facilidade com que o software pode ser entendido, corrigido, adaptado e ou melhorado.

## ❖ Tipos de Manutenção

- MANUTENÇÃO CORRETIVA: identificar e corrigir erros
- MANUTENÇÃO ADAPTATIVA: adaptar o software ao ambiente
- MANUTENÇÃO PERFECTIVA: atender pedidos do usuário para modificar funções existentes, incluir novas funções e efetuar melhoramentos gerais
- MANUTENÇÃO PREVENTIVA: Melhorar a manutenibilidade ou confiabilidade futuras e fornecer uma base melhor para futuros melhoramentos

---

# Problemas da Manutenção

A maioria dos problemas com a manutenção do software é causada por deficiências na maneira como o software foi planejado e desenvolvido

## PROBLEMAS CLÁSSICOS

- ✓ É difícil ou impossível rastrear o processo através do qual o software foi criado. A maioria dos softwares não foram projetados para suportar alterações.
- ✓ É difícil ou impossível traçar a evolução do software através das várias versões. As alterações não são adequadamente documentadas A documentação não existe, é incompreensível ou está desatualizada.
- ✓ É muito difícil entender programas "de outras pessoas", que frequentemente não estão presentes para explicar. A dificuldade aumenta conforme o número de elementos na configuração de software aumenta.
- ✓ A manutenção não é vista como um trabalho "glamoroso" ou importante

---

# Custo de Manutenção

## ❖ Custos diretos da Manutenção

- 70 % do orçamento do software (ciclo de vida)
- Diminuição dramática na produtividade 40:1 [Boehm,79]
  - Custo do desenvolvimento : \$25,00 por linha de código
  - Custo da manutenção : \$1.000,00 por linha de código

## ❖ Outros custos Não Monetários

- Adiamento de oportunidades de desenvolvimento
- Insatisfação do cliente
- Redução da qualidade global do software
- Insatisfação do pessoal de desenvolvimento

---

# Custo de Manutenção

→ A manutenibilidade é difícil de quantificar. Pode-se determinar a manutenibilidade indiretamente considerando componentes das atividades de manutenção que podem ser medidos

## ❖ MÉTRICAS DE MANUTENIBILIDADE (Gilb, 1979)

- tempo de reconhecimento do problema
- tempo de demora administrativa
- tempo de análise do problema
- tempo de especificação da alteração
- tempo de correção ou modificação
- tempo de teste local e global
- tempo de revisão da manutenção

---

# Custo da Manutenção

→ A manutenibilidade pode ser medida indiretamente considerando medidas da estrutura do projeto e medidas da complexidade do software

## ❖ MODELO PARA ESTIMATIVA DE CUSTOS DE MANUTENÇÃO

→ Belady, 1972

$$M = P + Ke^{(c-d)}$$

Onde,

M = esforço de manutenção

P = esforço produtivo

K = constante empírica

e = número Euler (2,78...)

c = medida de complexidade atribuída a falta de bom projeto e de boa documentação

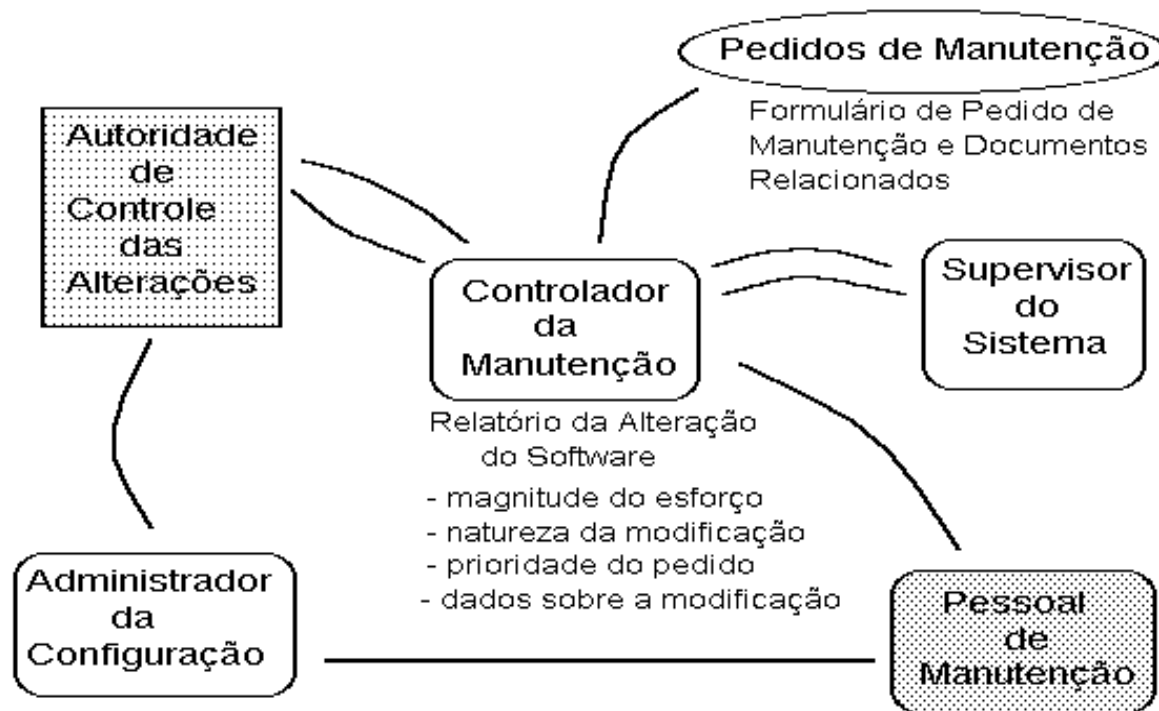
d = medida do grau de familiaridade com o software

---

# Fatores Facilitadores

- ✓ Pessoal qualificado
- ✓ Processo de software estabelecido e documentado
- ✓ Sistemas com estruturas lógicas e compreensíveis
- ✓ Processo de manutenção estabelecido e documentado
- ✓ Processo de controle de mudanças estabelecido e documentado

# Organização para a Manutenção



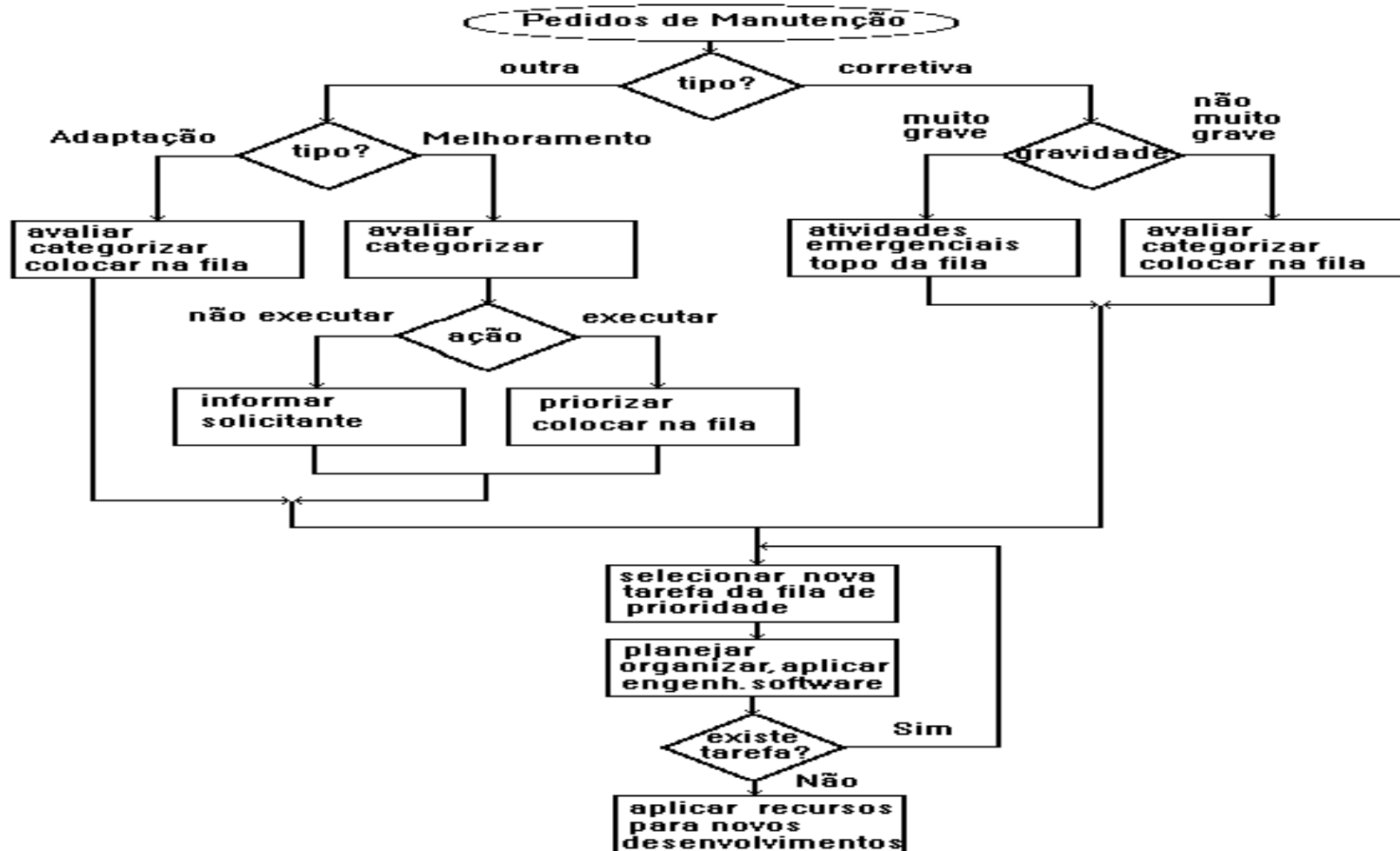


---

# Organização para a Manutenção

- ✓ Autoridade controladora de mudanças: evita que as mudanças favoreçam um solicitante e prejudiquem outros...
- ✓ Controlador de manutenção: evita que modificações sejam implementadas sem serem devidamente aprovadas...
- ✓ Supervisor de sistemas: uma vez que tem familiaridade com um grupo de sistemas, pode avaliar mais rapidamente os pedidos de manutenção

# Processo de Manutenção



---

# Registros de Manutenção

EXEMPLO DE DADOS QUE PODEM SER ARMAZENADOS  
(*Swanson*):

- ✓ identificação e descrição do programa
  - número de linhas de comando
  - linguagem de programação usada
  - data da instalação do programa
  - número de execuções do programa desde a instalação
  - número de falhas de processamento associadas ao item anterior
- ✓ identificação e descrição das alterações no programa
  - tipo de manutenção
  - número de linhas de comandos adicionadas por alteração no programa
  - datas de início e fim da manutenção
  - número de pessoas-horas despendidos na manutenção

---

# Gerenciamento da Manutenção

## ❖ Arcabouço Quantitativo (*Swanson*)

- número médio de falhas de processamento por execução do programa
- pessoas-horas despendido em cada categoria de manutenção
- número médio de pessoas-horas despendido por linha de comando adicionado ou deletado devido a manutenção
- tempo médio de processamento para um pedido de manutenção
- porcentagem de pedidos de manutenção por tipo

## ❖ Decisões:

- Tecnologia de Desenvolvimento
- Estrutura de Manutenção
- Alocação de Recursos

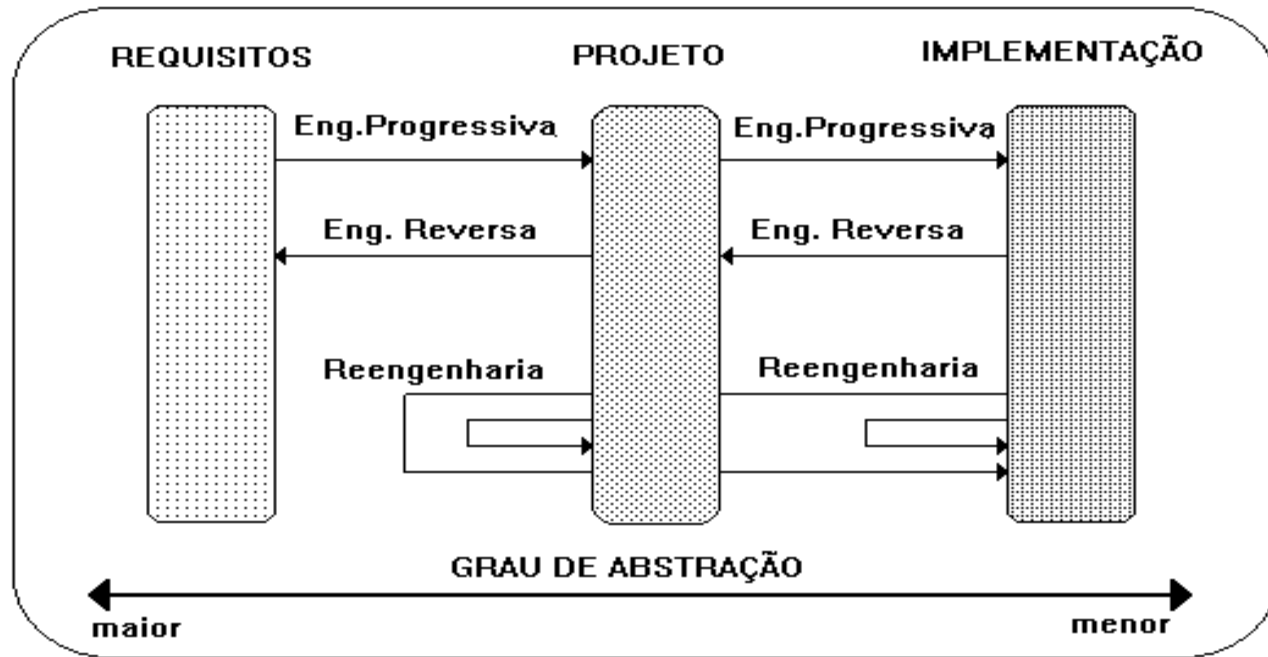
---

# Engenharia Reversa e Reengenharia

## ❖ Programas "Alienígenas"

- Programas com fluxo de controle equivalente a um "prato de espaguete", módulos muito grandes e poucas linhas de comentários significativos.
- Projeto de dados e projeto arquitetural ruins.
- Nenhuma metodologia de desenvolvimento foi aplicada.
- Nenhum outro elemento da configuração de software, além do código.
- Documentação e registro histórico das alterações incompletos.
- Nenhum membro do pessoal atual de manutenção trabalhou no desenvolvimento do programa.

# Engenharia Reversa e Reengenharia



- ❖ **ENGENHARIA REVERSA:** Processo de análise de um software, partindo-se inicialmente da implementação para um nível mais alto de abstração
- ❖ **REENGENHARIA:** Implica no exame e na alteração do software para reconstruí-lo em uma nova forma.

---

# Elementos de Eng. Reversa e Reengenharia

## DIRECIONALIDADE:

Engenharia Reversa: se a direcionalidade tem sentido único, toda informação é extraída a partir do código fonte e usada durante as atividades de manutenção.

Reengenharia : se a direcionalidade tem sentido duplo, a informação é usada para "alimentar" uma abordagem de reengenharia, que tentará reestruturar ou regenerar o programa antigo.

## GRAU DE ABSTRAÇÃO

Refere-se ao nível de generalidade das descrições. Conforme o nível de abstração aumenta, mais produtivas se tornam as informações

## INTEIREZA:

Refere-se ao nível de detalhes que é fornecido em cada nível de abstração.

## INTERATIVIDADE:

Refere-se ao grau de participação do ser humano no processo de engenharia reversa. Conforme o nível de abstração aumenta, a interatividade deve aumentar ou a inteireza será prejudicada.