

# AULA 7: Implementação de melhorias em projetos de software (manutenção, reengenharia e engenharia reversa)

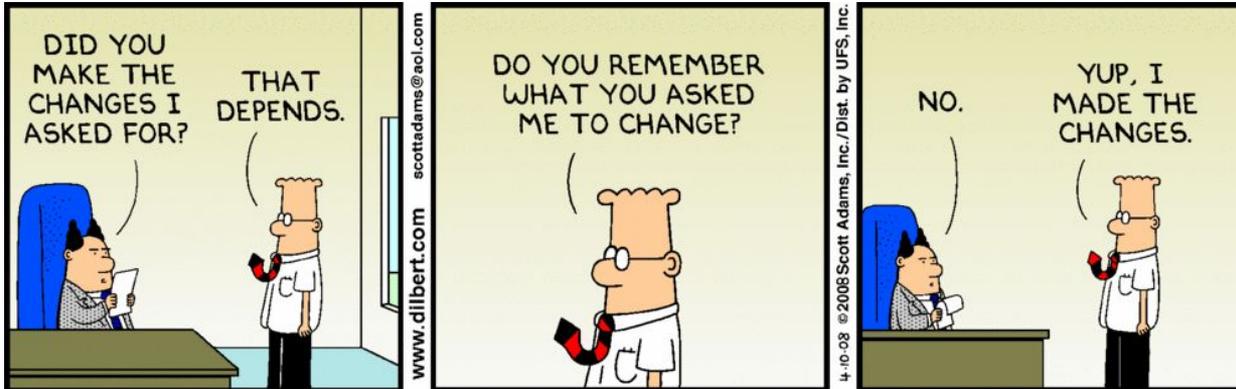
---

Professora: **Rosana T. Vaccare Braga**



# Manutenção do software

- O propósito do processo manutenção do sistema e software é modificar o produto de sistema/software depois de liberado.
- Necessidade de corrigir um defeito, melhorar a usabilidade, adicionar uma nova funcionalidade ou adaptar o software a um ambiente diferente.



# Manutenção do software

- **As alterações ocorrem por diversas razões.**
- **As razões para as alterações determinam a categoria de manutenção.**
  - **Manutenção Corretiva**
  - **Manutenção Adaptativa**
  - **Manutenção Perfectiva**
  - **Manutenção Preventiva**

# Iceberg da manutenção

- **Muitos problemas e custos não são visíveis ou conhecidos**



# Custo de manutenção

- ○ **Custo de manutenção pode ser dividido em:**
  - **Tentar entender o que o software faz**
  - **Interpretar as estruturas de dados, as características de interface e limites de desempenho**
  - **Analisar, avaliar, projetar, codificar e testar as modificações**
  - **Atividades produtivas**

# Custo de manutenção

- **Modelo de esforço de manutenção:**

$$M = p + ke^{(c-d)}$$

onde:

**$M$**  = Esforço total

**$p$**  = Esforço produtivo

**$k$**  = Constante empírica

**$c$**  = Medida de complexidade atribuída à falta de bom projeto e documentação

**$d$**  = Medida do grau de familiaridade com o software

# Problemas de manutenção

- **Modelo de esforço de manutenção:**
  - **É difícil ou impossível traçar a evolução do software através das várias versões se as alterações não forem adequadamente documentadas (isso tem influência na **variável c**)**
  - **É difícil ou impossível traçar o processo através do qual o software foi criado.**

# Problemas de manutenção

- **Modelo de esforço de manutenção (variável d):**
  - É muito difícil entender programas "de outras pessoas". A dificuldade aumenta conforme o número de elementos na configuração de software diminui.
  - "As outras pessoas" frequentemente não estão presentes para explicar.

# Problemas de manutenção

- **Modelo de esforço de manutenção:**
  - **A documentação não existe, é incompreensível ou está desatualizada (variável c).**
  - **A maioria dos softwares não foram projetados para suportar alterações.**
  - **A manutenção não é vista como um trabalho glamoroso.**

# Manutenibilidade

- **Pode ser definida qualitativamente como a facilidade com que o software pode ser entendido, corrigido, adaptado e ou melhorado**

# Manutenibilidade

- É **afetada** por muitos fatores:
  - **Cuidado inadequado com o projeto, codificação e teste**
  - **Configuração de software ruim**
  - **Disponibilidade de pessoal qualificado de software**

# Manutenibilidade

- **É afetada** por muitos fatores:
  - **Facilidade de manusear o sistema**
  - **Uso de linguagens de programação padronizadas**
  - **Uso de sistema operacional conhecido**
  - **Estruturas padronizadas de documentação**

# Manutenibilidade

- É **afetada** por muitos fatores:
    - Disponibilidade da pessoa ou grupo que desenvolveu o software
    - O planejamento para manutenibilidade
    - (fator mais importante que afeta a manutenibilidade)
- 

## Manutenibilidade: **Medidas Quantitativas**

- **Difícil** de quantificar
- **Pode-se determinar a manutenibilidade indiretamente considerando-se atributos das atividades de manutenção que podem ser medidos**

# Manutenibilidade: **Medidas Quantitativas**

- **Métricas de Manutenibilidade (Gilb)**
  - **tempo de reconhecimento do problema**
  - **tempo de demora administrativa (aprovar)**
  - **tempo de coleta de ferramentas de manutenção**
  - **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**

# Revisões de Manutenibilidade: Etapa de revisão de requisitos

- **Observar:**
  - **áreas de melhoramentos futuros**
  - **aspectos de portabilidade do software**
  - **interfaces que poderiam impactar a manutenção**

# Revisões de Manutenibilidade: Etapa de revisão de projeto

- **Avaliar:**
  - projeto arquitetural
  - projeto procedimental
  - projeto de interfaces e
  - projeto de dados quanto à facilidade de manutenção e a qualidade global
- **Dar ênfase:**
  - ao estilo
  - à documentação interna

# Fatores de Qualidade de Software

- (Norma ISO 9126/25010)
- Os fatores de qualidade de software focalizam três aspectos importantes do software:

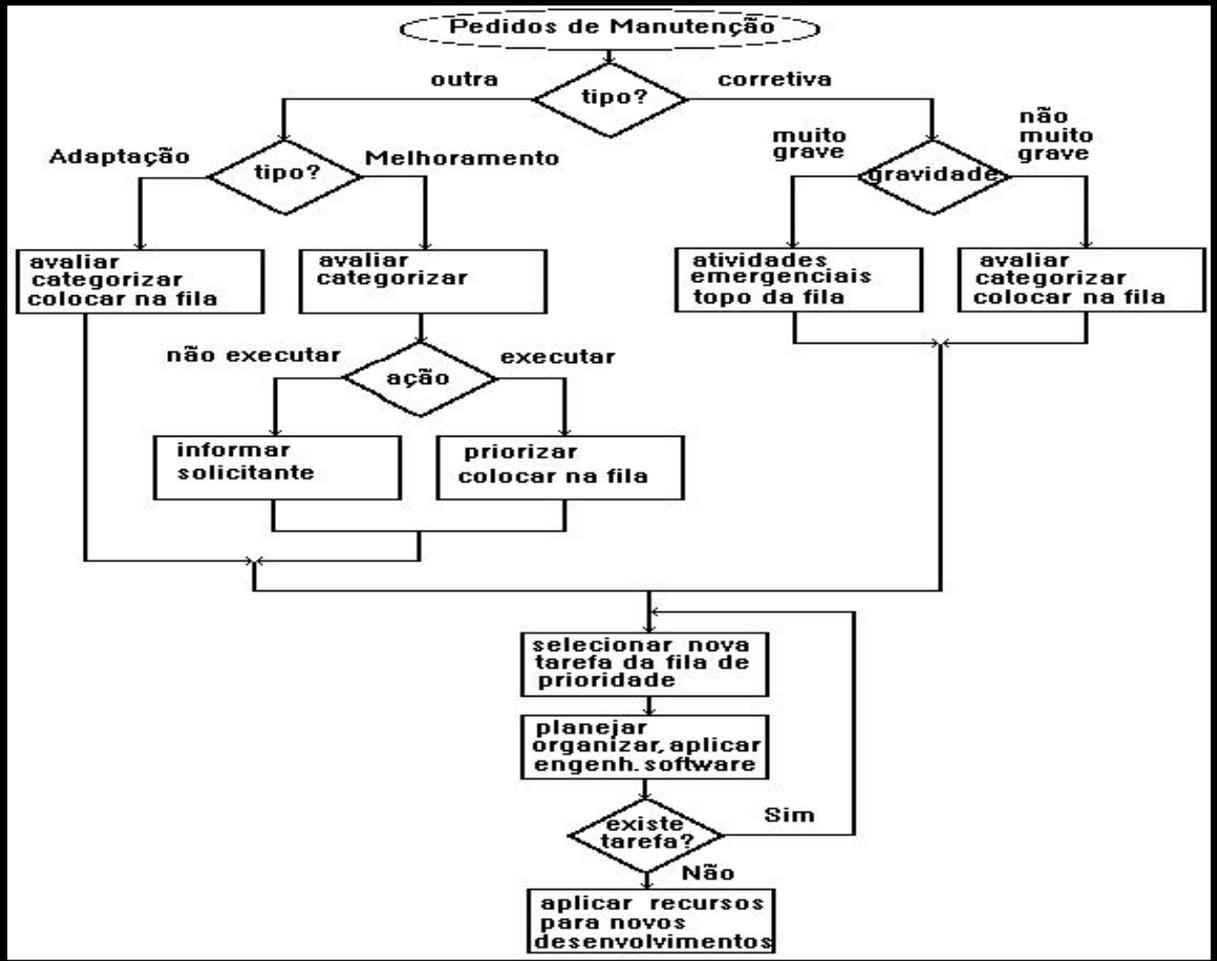




# Tarefas de Manutenção

- 2) Descrever procedimentos de avaliação e de comunicação**
- 3) Definir sequências padronizadas de eventos (para os pedidos de manutenção)**

# Manutenção do software



# Tarefas de Manutenção

**4) Estabelecer procedimentos para registrar a **história** das atividades de manutenção**

# Manutenção do software

## Exemplo de Dados que podem ser Armazenados (Swanson)

- identificação do programa
- número de comandos fonte
- 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

# Manutenção do software

## Exemplo de Dados que podem ser Armazenados (Swanson)

- nível e identificação da alteração no programa
- número de comandos fonte adicionados por alteração no programa
- número de pessoas-horas despendidos na manutenção
- identificação do pedido de manutenção
- tipo de manutenção
- datas de início e fim da manutenção

# Exemplo de ferramenta de controle de alterações: Redmine

Visão geral Atividade Scrum2B Tasks Sprints Run charts **Tarefas** Nova tarefa Gantt Calendário Notícias Documentos W

## Funcionalidade #3373

Atualizar Tempo de trabalho Observar Copiar Excluir

Portal #3435: Atualização TCCWEB 2023 - Sprint3

« Anterior | 49/125 | Próximo »

### Alteração na inclusão do projeto

Adicionado por Artur Sampaio 3 meses atrás. Atualizado 30 dias atrás.

Situação:	Em andamento	Início:	06/02/2023
Prioridade:	Normal	Data prevista:	
Atribuído para:	Artur Sampaio	% Terminado:	<div style="width: 0%;"></div> 0%
Categoria:	-	Tempo gasto:	-
Versão:	-		

### Descrição Responder

3.11  
Um campo adicional deve informar o tipo específico de projeto: 1- projeto de graduação/formatura/empreendedor orientado por docente; 2-projeto em universidade fora do Brasil orientado por docente; 3- estágio regulamentar presencial/home-office em empresa sediada no Brasil; 4-estágio presencial em empresa fora do Brasil; 5- estágio home-office em empresa sediada fora do Brasil; 6- emprego como CLT ou PJ/prestador de serviço no Brasil; e 7-emprego em empresa fora do Brasil. Para os tipos 1 e 2, deve-se indicar um orientador (que tem o papel de docente na base de usuários) e opcionalmente um coorientador. O orientador pode também ser um doutorando ou pós-doutorando, mas nesse caso há necessidade de ter um coorientador que seja docente do ICMC e que receba a notificação da inscrição e autorize o projeto também. Para os casos 3 a 7, deve-se indicar um supervisor (que é alguém externo ao sistema e ligado a uma empresa das empresas). Para o caso 3, deve-se indicar um supervisor acadêmico (docente)

### Subtarefas

Funcionalidade #3413: Possibilitar que doutorando ou pósdoc co-orientar projeto Nova

### Tarefas relacionadas

### Histórico Revisões associadas

Visão geral Atividade Scrum2B Tasks Sprints Run charts **Tarefas** Nova tarefa

## Tempo gasto

Tarefa  Funcionalidade #3373: Alteração na inclusão do projeto

Data \*

Horas \*

Comentário

Atividade \* --- Seleccione ---

Hora início --- Seleccione ---

Min. início

Hora fim

Min. fim

- Design
- Desenvolvimento
- Reunião
- Manutenção
- Implementação
- Documentação
- Atendimento ao Usuário
- Colocação em Produção
- Preparação de Ambiente
- Teste
- Análise

# Tarefas de Manutenção

## 5) Definir critérios de revisão e avaliação

- **Envolve a criação de um conjunto de padrões e regras que serão usados para avaliar a qualidade do software existente. Isso pode incluir a avaliação de código-fonte, documentação, testes e outros artefatos relacionados ao software**

# Engenharia Reversa e Reengenharia

## Cenário usual para manutenção

- **Sistemas sem documentação**
- **Dificuldade de manutenção**
- **Erros gerando outros erros**
- **Código duplicado**

# Engenharia Reversa e Reengenharia

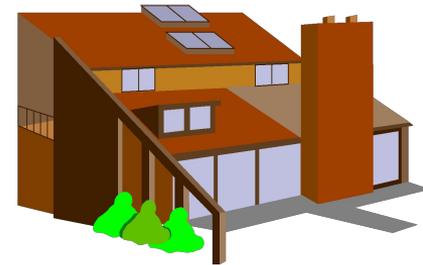
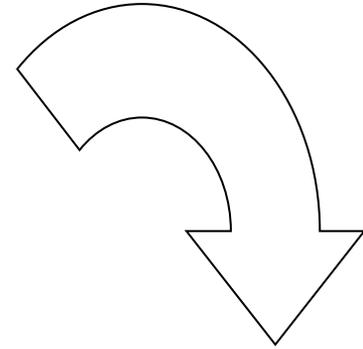
## Necessidade de Manutenção no Software

- **O quê fazer?**
- **Quem poderá me ajudar ?**
- **Cadê o programador?**
- **O quê será que ele quis fazer aqui?**



# Engenharia Reversa e Reengenharia

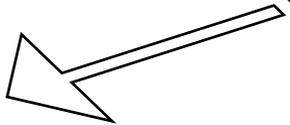
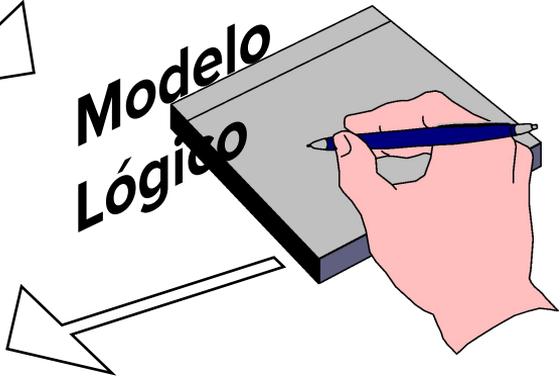
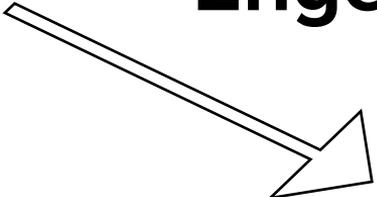
- **Será possível?**
- **Na construção civil, ok**
- **E com sistemas de software?**



# Engenharia Reversa e Reengenharia



**Engenharia Reversa**



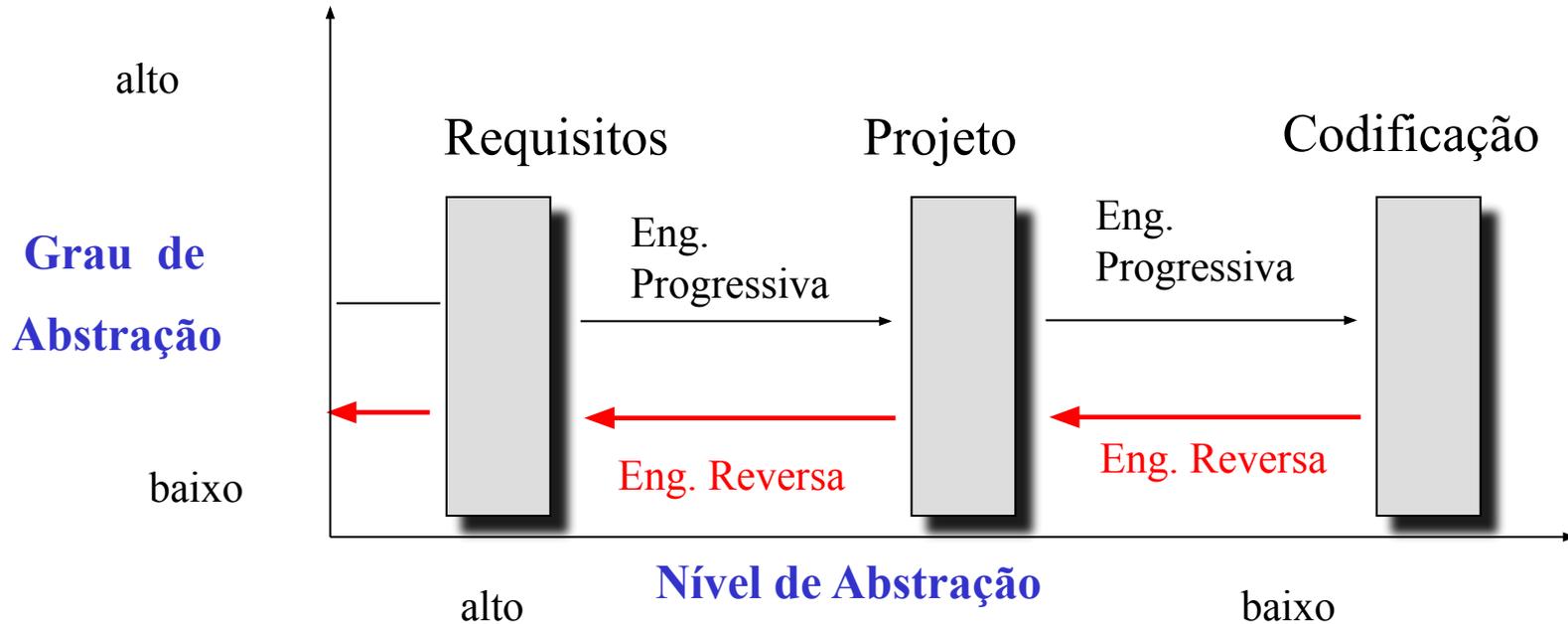
**Reengenharia**

# Engenharia Reversa

- O termo “Engenharia Reversa” tem sua origem na análise de hardware, pois é comum a prática de decifrar projetos de produtos finalizados com intuito de duplicá-los.
- O conceito de Engenharia Reversa de Software é similar. Porém, tradicionalmente o objetivo dessa engenharia reversa é obter apenas um entendimento do sistema

# Abstrações do Software

- Engenharia Progressiva x Reversa



# Engenharia Reversa

- **Quais os documentos utilizados para realizar engenharia reversa?**
  - **Código fonte**
  - **Informações de usuários e/ou analista**
  - **Documentação existente: manual de usuário, manual de sistema, casos de uso, modelos/diagramas UML, fluxogramas...**

# Reengenharia de Software

- **Reengenharia de Software é qualquer atividade que:**
  - **(1) Melhore o entendimento do software**
  - **(2) Prepare ou melhore o software em si, aumentando sua manutenção, seu reuso e sua extensão**

**Chikofsky e Cross definem reengenharia:**

***“o exame e a alteração de um sistema para reconstituí-lo de uma nova forma, seguida pela sua implementação”***