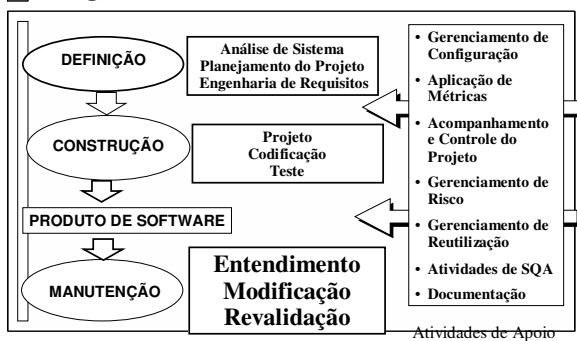


Manutenção de Software

Tópicos da Aula

- Manutenção
 - Tipos de Manutenção
 - Manutenção Estruturada x Não-Estruturada
 - Custo de Manutenção
 - Problemas Clássicos
 - Manutenibilidade
 - Tarefas de Manutenção
 - Efeitos Colaterais
- Engenharia Reversa e Reengenharia

O Processo de Engenharia de Software



Manutenção de Software

- Alterações efetuadas no software depois de sua liberação.
- As alterações ocorrem por diversas razões, as quais determinam o tipo de manutenção.

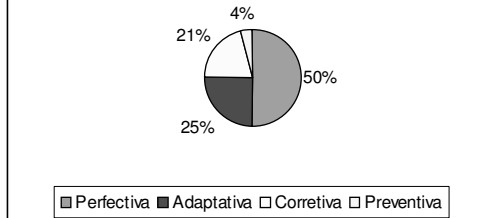
Tipos de Manutenção

- Manutenção Corretiva
 - Identificar e corrigir erros.
- Manutenção Adaptativa
 - Adaptar o software às mudanças ocorridas no ambiente.
 - Novas gerações de hardware.
 - Novos sistemas operacionais e software de apoio.
 - Atualizações e modificações em equipamentos periféricos e outros elementos de sistema.

Tipos de Manutenção

- Manutenção Perfectiva
 - Atender aos pedidos do usuário para:
 - Modificar funções existentes.
 - Incluir novas funções (novos requisitos).
 - Ampliar o escopo do software.
 - Efetuar melhoramentos gerais.
- Manutenção Preventiva
 - Melhorar a manutenibilidade ou confiabilidade futuras e fornecer uma base melhor para futuros melhoramentos.

Tipos de Manutenção



Fonte: Lientz e Swanson (1981) – Pesquisa com gerentes de 487 empresas.

A Fase de Manutenção

- Uma das fases mais problemáticas do ciclo de vida de software.
- Pode despendar mais de 70% de todo esforço de uma organização.

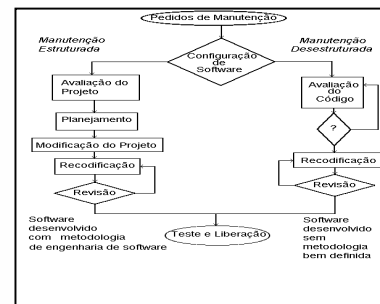
Por que é exigida tanta manutenção e ...
por que é despendido tanto esforço?

A Fase de Manutenção

- Idade média de 10 a 15 anos.
- Principal interesse: tamanho do programa e espaço de armazenamento.
- Migração para novas plataformas.
- Sistemas mal estruturados.
- Melhoramentos para atender novas necessidades.
- Nenhuma preocupação com a arquitetura global.
- Codificação, lógica e documentação ruins.

Esses sistemas devem continuar "rodando"
e as alterações são inevitáveis.

Manutenção Estruturada x Não-Estruturada



Manutenção Estruturada x Não-Estruturada

Manutenção Não Estruturada

- Código-fonte: único elemento disponível de uma configuração de software.
 - Documentação interna ruim.
- Características do software difíceis de serem verificadas e freqüentemente mal-interpretadas.
 - Estrutura de programa, estruturas de dados globais, interfaces do sistema, desempenho/restrições de projeto.
- Mudanças difíceis de serem avaliadas.
 - Testes de regressão praticamente impossíveis: não há registro de testes.

Manutenção Estruturada x Não-Estruturada

Manutenção Estruturada

- Configuração de software completa.
 - Avaliação da documentação de projeto.
 - Características estruturais, de desempenho e de interface são determinadas.
 - Impacto das modificações ou correções exigidas é avaliado.
- Uma abordagem é planejada.
- O projeto é modificado e revisado.
- Um novo código-fonte é desenvolvido e testes de regressão são executados.

Custo de Manutenção

- Custo da manutenção como uma porcentagem do orçamento do software.

	Lientz e Swanson (1980)	Arfa et al (1988-1989)
Manutenção	48.8%	44.63%
Desenvolvimento	43.3%	46.59%
Outros	7.9%	8.78%

Custo de Manutenção

- Outros custos não monetários.
 - Adiamiento de oportunidades de desenvolvimento.
 - Recursos canalizados para a atividade de manutenção.
 - Redução da qualidade global do software.
 - Mudanças que introduzem erros latentes no software mantido.
 - Insatisfação do cliente.
 - Demora das modificações.
 - Insatisfação do pessoal de manutenção.
 - Falta de motivação.

Custo de Manutenção

- Custo final da manutenção.
 - Diminuição drástica da produtividade.
 - (Boehm)
 - Custo do desenvolvimento: \$250,00 por linha de código.
 - Custo da manutenção: \$1.000,00 por linha de código.

Custo de Manutenção

- Pode ser dividido em:
 - Atividades Quebra Cabeça.
 - Tentar entender o que o software faz.
 - Tentar interpretar as estruturas de dados, as características de interface e os limites de desempenho.
 - Atividades Produtivas.
 - Analisar, avaliar, projetar, codificar e testar as modificações.

Custo de Manutenção

- Modelo de Esforço de Manutenção (Belady72)

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

- 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

Crescimento Exponencial

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.

A falta de controle e disciplina nas atividades de desenvolvimento e engenharia de software quase sempre se traduz em problemas durante a manutenção.

Problemas Clássicos

- É difícil ou impossível traçar (rastrear) a evolução do software por meio de muitas versões ou lançamentos.
 - As alterações não são adequadamente documentadas.
- É difícil ou impossível traçar (rastrear) o processo por meio do qual o software foi criado.

Problemas Clássicos

- É muito difícil entender programas “de outras pessoas”.
 - A dificuldade aumenta conforme o número de elementos na configuração de software diminui.
- As “outras pessoas” freqüentemente não estão presentes para explicar.

Problemas Clássicos

- A documentação não existe, é incompreensível ou está desatualizada.
 - A documentação deve ser compreensível e consistente com o código-fonte.
- A maioria dos softwares não foram projetados para suportar alterações.
- A manutenção não é vista como um trabalho “fascinante”.
 - Frustração.

Manutenibilidade

- A Manutenibilidade pode ser definida qualitativamente como a facilidade com que o software pode ser entendido, corrigido, adaptado e/ou melhorado.
 - ISO 9126

Manutenibilidade

- A manutenibilidade é afetada por muitos fatores:
 - Negligência no projeto, codificação e no teste.
 - Configuração de software.
 - Ambiente.
- **Impacto negativo sobre a capacidade de manter o software.**
- O software inevitavelmente sofrerá mudanças.

Manutenibilidade: Medidas Quantitativas

- A manutenibilidade de software é difícil de quantificar.
- Pode-se determinar a manutenibilidade indiretamente considerando-se os atributos da atividade de manutenção que podem ser medidos.

Manutenibilidade: Medidas Quantitativas

- Métricas de Manutenibilidade (*Gilb*)
 - Tempo de reconhecimento do problema.
 - Tempo de demora administrativa.
 - 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.
 - Tempo de recuperação total.

Métricas Orientadas ao Tempo

Manutenibilidade: Medidas Quantitativas

- A manutenibilidade também pode ser medida indiretamente considerando medidas da estrutura do projeto e medidas da complexidade do software.

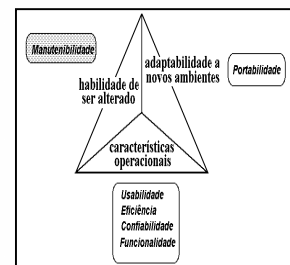
Revisões de Manutenibilidade

Revisão de Configuração

- Revisão de manutenção mais formal.
 - Conclusão da atividade de teste.
- Assegura que todos os elementos da configuração do software estejam completos, sejam compreensíveis e arquivados para ter controle das modificações.

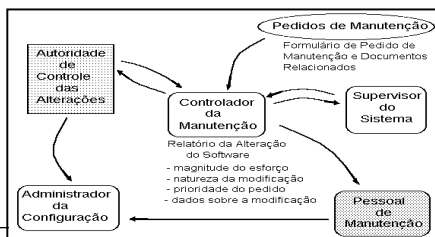
Fatores de Qualidade de Software

- Norma ISO 9126
 - Os fatores de qualidade de software focalizam três aspectos importantes do software produto.



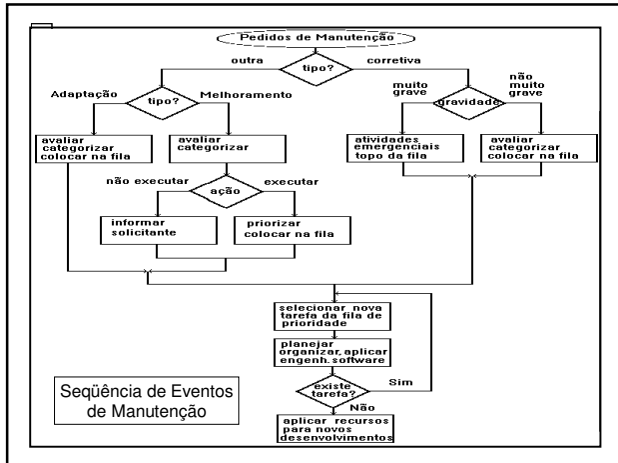
Tarefas de Manutenção

- Estabelecer uma organização para a manutenção ("de fato" ou formal).
 - Delegação de responsabilidades.



Tarefas de Manutenção

- Descrever procedimentos de avaliação e de comunicação.
- Definir seqüências padronizadas de eventos.
 - Resultado de um pedido de manutenção.



Tarefas de Manutenção

- Estabelecer procedimentos para registrar a história das atividades de manutenção.
 - Quais dados devem ser guardados.
 - Base para um banco de dados sobre manutenção.
 - Posterior avaliação.
- Exemplo (Swanson)

Tarefas de Manutenção

- 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.

Tarefas de Manutenção

- 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.

Medidas de Desempenho e Manutenção

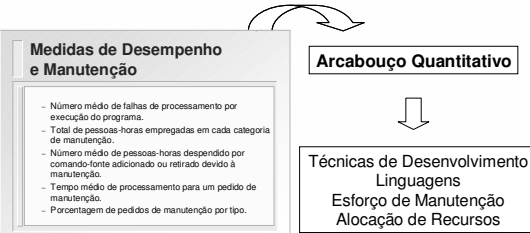
- Definir critérios de revisão e avaliação.
- Exemplo
 - Medidas de Desempenho e Manutenção.
 (Swanson)

Medidas de Desempenho e Manutenção

- Número médio de falhas de processamento por execução do programa.
- Total de pessoas-horas empregadas em cada categoria de manutenção.
- Número médio de pessoas-horas despendido por comando-fonte adicionado ou retirado devido à manutenção.
- Tempo médio de processamento para um pedido de manutenção.
- Porcentagem de pedidos de manutenção por tipo.

Tarefas de Manutenção

- Definir critérios de revisão e avaliação.



Efeitos Colaterais da Manutenção

- Erro ou outro comportamento indesejável que ocorra como resultado de uma modificação.
- Sempre que uma mudança é introduzida, o potencial de erros cresce.
- Efeitos colaterais...
 - Na codificação.
 - Nos dados.
 - Na documentação.

Manutenção de Código Alienígena

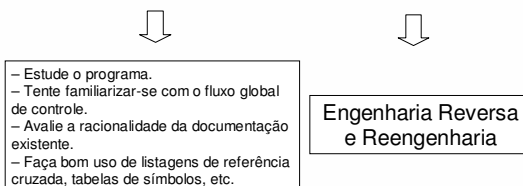
- Programas "alienígenas".
 - Programas com fluxo de controle equivalente a uma "tigela de espaguete".
 - Módulos muito grandes.
 - Poucos linhas de comentários significativos.
 - Não existe nenhum outro elemento da configuração de software, além do código.

Manutenção de Código Alienígena

- Programas "alienígenas".
 - Nenhum membro atual do pessoal técnico trabalhou no desenvolvimento do programa.
 - Nenhuma metodologia de desenvolvimento foi aplicada.
 - Projeto de dados e projeto arquitetural ruins.
 - Documentação e registro histórico das alterações incompletos e superficiais.

Manutenção de Código Alienígena

- O que pode ser feito com código "alienígena"?



Engenharia Progressiva x Engenharia Reversa

- Engenharia Progressiva
 - Processo tradicional de ES, caracterizado pelas atividades progressivas do ciclo de vida, que partem de um alto nível de abstração para um baixo nível de abstração.
- Engenharia Reversa
 - Processo inverso à Engenharia Progressiva, caracterizado pelas atividades retroativas do ciclo de vida, que partem de um baixo nível de abstração para um alto nível de abstração.

Engenharia Reversa e Reengenharia



Será possível ???

Na construção civil, ok!!
E com sistemas de software?



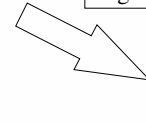
Engenharia Reversa e Reengenharia



Engenharia Reversa

Modelo Lógico

Reengenharia



Engenharia Reversa

- Processo de exame e compreensão do software existente, para recapturar ou recriar o projeto e decifrar os requisitos atualmente implementados pelo sistema, apresentando-os em um nível mais alto de abstração.
- Processo de análise de um software existente, em um esforço para representá-lo em uma abstração mais alta do que o código-fonte. (*Chikofsky e Cross, 1990*)
 - Redocumentação.
 - Recuperação de projeto.

Engenharia Reversa

- Extração de abstrações.
 - É feita uma avaliação do programa antigo, extraíndo, a partir do código fonte:
 - Uma especificação significativa do processamento.
 - A interface com o usuário.
 - As estruturas de dados ou a base de dados do programa.

Elementos da Engenharia Reversa

- Nível de Abstração
 - Conforme o nível de abstração aumenta, mais compreensíveis se tornam as informações.
 - Sofisticação da informação de projeto que pode ser extraída do código-fonte.

Elementos da Engenharia Reversa

- Completitude do Processo
 - Nível de detalhes que é fornecido em cada nível de abstração.
 - A completitude diminui à medida que o nível de abstração aumenta.

Elementos da Engenharia Reversa

■ Interatividade

- Grau de participação do ser humano no processo de Engenharia Reversa.
 - Conforme o nível de abstração aumenta, a interatividade deve aumentar ou a completitude será prejudicada.

Elementos da Engenharia Reversa

■ Direcionalidade

- Se a direcionalidade do processo tem sentido único...
 - Toda informação extraída a partir do código-fonte é usada durante as atividades de Manutenção.
- Se a direcionalidade tem sentido duplo...
 - A informação é usada para "alimentar" uma ferramenta de Reengenharia.

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
 - DFDs
 - fluxogramas
 - etc ...

Engenharia Reversa

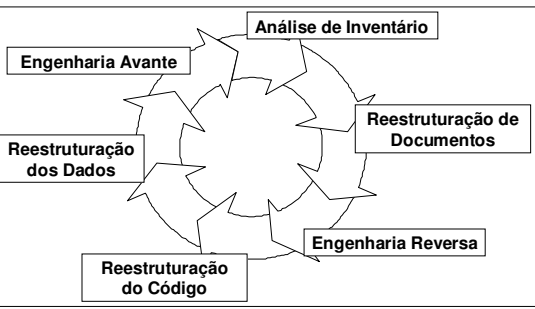
- Obtendo-se as informações necessárias para o completo entendimento do sistema...

O que fazer com essas informações?

- só para manutenção
- mesmo paradigma e mudança de linguagem
- mudança de paradigma (só modelo lógico)
- mudança de paradigma e de linguagem

Reengenharia

Reengenharia



Modelo de Processo de Reengenharia de Software

Análise de Inventário

- Inventário
 - Contém informação que fornece uma descrição detalhada de cada aplicação ativa.
 - Tamanho, idade, criticidade para o negócio.
- Avaliar cada aplicação sistematicamente, a fim de determinar quais são candidatas à reengenharia.

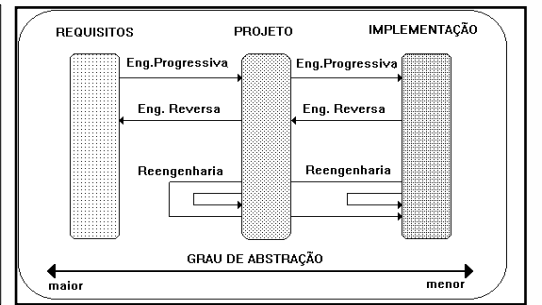
- ⇒ Sugestão: “Criar apenas a documentação necessária para melhorar o entendimento do software.”

- Desvendar os “segredos”.
- Extrair informações do projeto de dados, arquitetural e procedimental, para um programa existente.

- A arquitetura de programa é relativamente sólida, mas módulos individuais foram codificados de modo que torna difícil entendê-los, testá-los e mantê-los.
- O código-fonte é analisado usando uma ferramenta de reestruturação.
 - Violações das construções de programação estruturada são registradas.
- O código é reestruturado.
- O código reestruturado é revisado e testado para garantir que nenhuma anomalia foi introduzida.
- A documentação interna do código é atualizada.

- Uma arquitetura de dados fraca é difícil de adaptar e aperfeiçoar.
 - Viabilidade do programa a longo prazo.
 - A arquitetura de dados atual é "dissecada".
 - Os modelos de dados necessários são definidos.
 - Objetos de dados e atributos são identificados.
 - As estruturas de dados são revistas quanto à qualidade.
 - Se a estrutura de dados é fraca, os dados passam por reengenharia.
 - Modificações nos dados resultarão em modificações arquiteturais e de código.

- Transferência das abstrações, modelos lógicos e projeto do sistema para uma implementação física.
(Chikotsky e Cross, 1990)
 - Renovação ou Recomposição.
- Aplica princípios, conceitos e métodos de ES para recriar a aplicação existente.
 - Recupera informações de projeto (Engenharia Reversa), utilizando-as para alterar ou reconstituir o sistema existente a fim de melhorar sua qualidade global.
 - Reimplementa a função existente.
 - Adiciona novas funções (amplia a capacidade da aplicação antiga).



Ferramentas para a Automatização da Manutenção

- Editores de texto.
- Comparadores de arquivos.
- Compiladores/ferramentas de depuração.
- Geradores de referências cruzadas.
- Analisadores de código estáticos.
- Gerenciadores de configuração.