INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

Análise e Desenvolvimento de Sistemas

Instituto Federal da Bahia

Análise e Desenvolvimento de Sistemas

INF022 – Tópicos Avançados

Evolução de Software

Prof. Dr. Renato L. Novais
renato@ifba.edu.br

 Engenharia de Software, 8ª. edição. Capítulo 21

Objetivos

- Explicar por que as mudanças são inevitáveis para que os sistemas de software permanecam úteis
- Discutir a manutenção de software e os fatores de custo de manutenção
- Descrever os processos envolvidos em evolução de software

Tópicos abordados

- Dinâmica da evolução de programas
- Manutenção de software
- Processo de evolução
- Evolução de sistemas legados

Mudança de software

- Mudança de software é inevitável
  - Novos requisitos surgem quando o software é usado;
  - O ambiente de negócio muda;
  - Erros devem ser reparados;
  - Novos computadores e equipamentos são adicionados ao sistema;
  - O desempenho ou a confiabilidade do sistema deve ser melhorada.
- Um problema-chave para as organizações é a implementação e o gerenciamento de mudanças em seus sistemas.

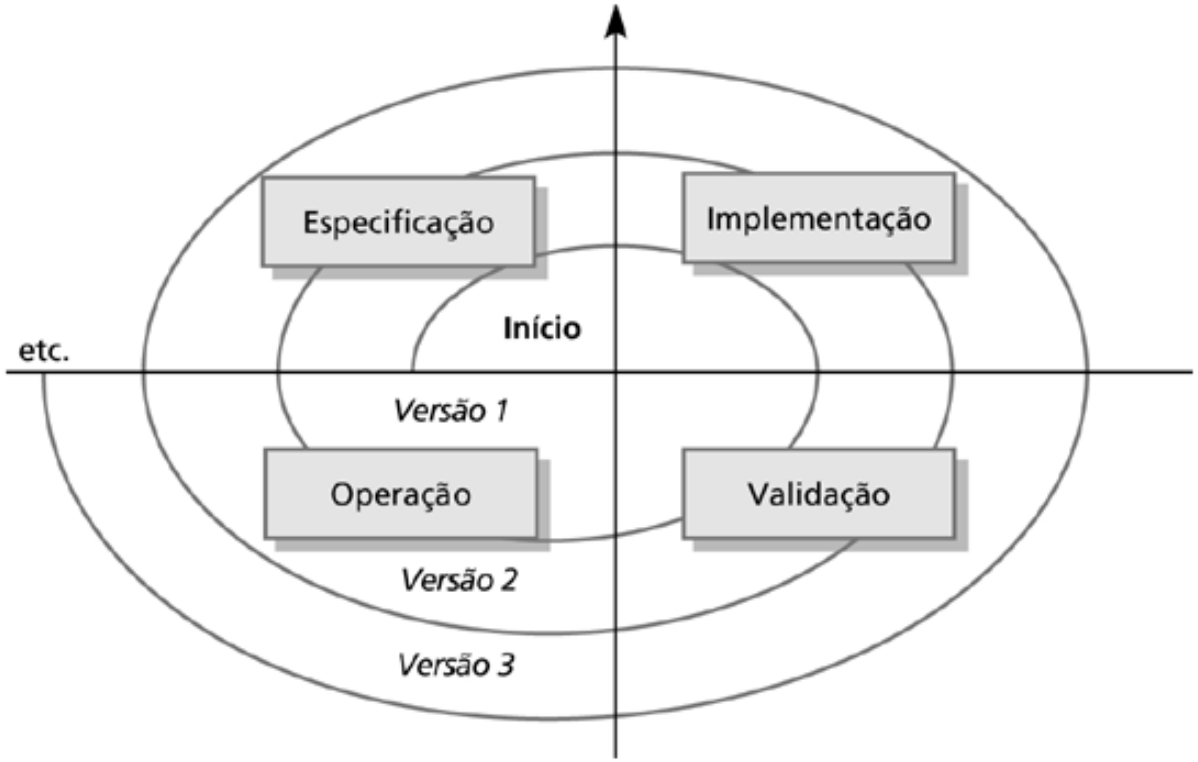
Importância da evolução

- As organizações fazem grandes investimentos em seus sistemas de software – eles são ativos críticos de negócios.
- Para manter o valor desses ativos de negócio, eles devem ser mudados e atualizados.
- A maior parte do orçamento de software nas grandes organizações é voltada para evolução do software existente ao invés do desenvolvimento de um novo software.

Modelo espiral de evolução

Figura 21.1

Modelo em espiral de desenvolvimento e evolução.



Dinâmica da evolução de programas

- Dinâmica de evolução de programas é o estudo dos processos de mudança de sistema.
- Após muitos estudos empíricos, Lehman e Belady propuseram que havia uma série de 'leis' que se aplicavam a todos os sistemas quando eles evoluiam.
- Existem observações consideráveis ao invés de leis. Elas são aplicáveis a sistemas de grande porte desenvolvidos por grandes organizações. Talvez menos aplicáveis em outros casos.

Leis de Lehman

Tabela 21.1 Leis de Lehman

Lei	Descrição
Mudança contínua	Um programa usado em um ambiente real deve mudar necessariamente ou tornar-se progressivamente menos útil.
Complexidade crescente	À medida que um programa muda, sua estrutura tende a se tornar mais complexa. Recursos extras devem ser dedicados para preservar e simplificar a estrutura.
Evolução de programa de grande porte	A evolução de programa é um processo auto-regulável. Atributos de sistemas como tamanho, tempo entre versões e número de erros reportados é quase invariável em cada versão de sistema.
Estabilidade organizacional	Durante o ciclo de vida de um programa, sua taxa de desenvolvimento é quase constante e independente de recursos dedicados ao desenvolvimento do sistema.
Conservação de familiaridade	Durante o ciclo de vida de um sistema, mudanças incrementais em cada versão são quase constantes.
Crescimento contínuo	A funcionalidade oferecida pelos sistemas deve aumentar continuamente para manter a satisfação do usuário.
Qualidade em declínio	A qualidade dos sistemas entrará em declínio a menos que eles sejam adaptados a mudanças em seus ambientes operacionais.
Sistema de feedback	Os processos de evolução incorporam sistemas de feedback com vários agentes e loops e você deve tratá-los como sistemas de feedback para conseguir aprimoramentos significativos de produto.

Aplicabilidade das leis de Lehman

- As leis de Lehman parecem ser, geralmente, aplicáveis a sistemas customizados de grande porte desenvolvidos por grandes organizações
  - Confirmado em mais recente trabalho de Lehman sobre o projeto FEAST (veja leitura adicional no Website do livro)
- Não está claro como elas devem ser modificadas para
  - Produtos de software de prateleira (commertial of-the-shelf - COTS);
  - Sistemas que incorporam um número significativo de componentes COTS;
  - Pequenas organizações;
  - Sistemas de tamanho médio.

Manutenção de software

- É a modificação de um programa após ter sido colocado em uso.
- A manutenção normalmente não envolve mudanças consideráveis na arquitetura do sistema.
- As mudanças são implementadas pela modificação de componentes existentes e pela adição de novos componentes ao sistema.

A manutenção é inevitável

- Os requisitos de sistema podem mudar enquanto o sistema está sendo desenvolvido porque o ambiente está mudando. Portanto, um sistema entregue não atenderá a esses requisitos!
- Os sistemas estão fortemente acoplados ao seu ambiente. Quando um sistema é instalado em um ambiente, ele muda esse ambiente e, portanto, muda os requisitos de sistema.
- Portanto, os sistemas DEVEM ser mantidos se forem úteis em um ambiente.

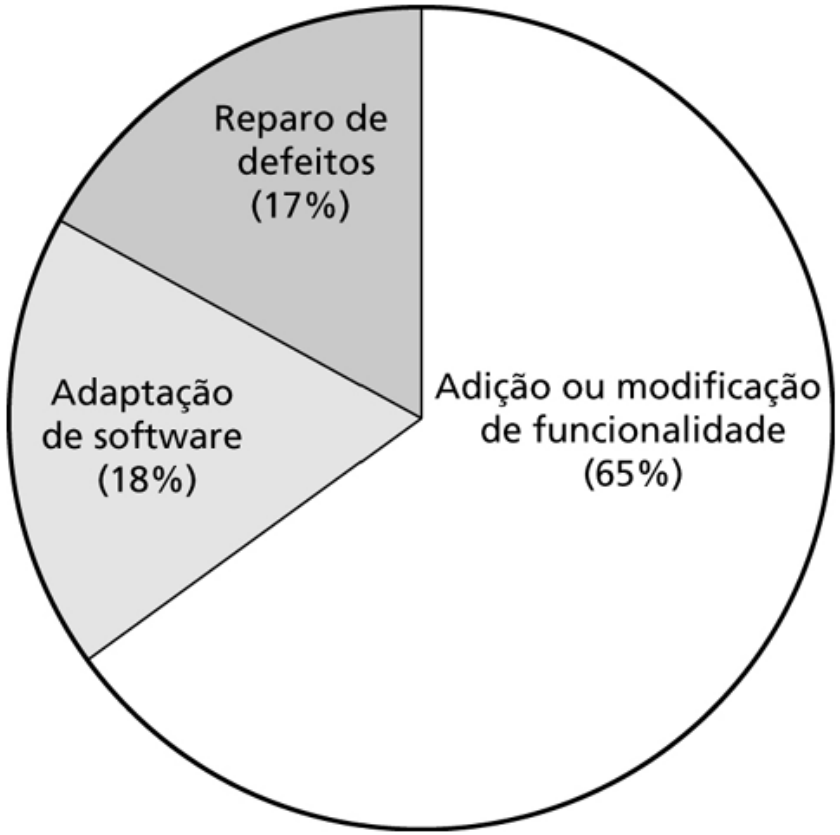
Tipos de manutenção

- Manutenção para reparar defeitos de software
  - Mudança em um sistema para corrigir deficiências de maneira a atender aos seus requisitos.
- Manutenção para adaptar o software a um ambiente operacional diferente
  - Mudança de um sistema de tal maneira que ele opere em um ambiente diferente (computador, OS, etc.) à partir de sua implementação inicial.
- Manutenção para adicionar funcionalidade ao sistema ou modificá-lo
  - Modificação do sistema para satisfazer a novos requisitos.

Distribuição de esforços de manutenção

Figura 21.2

Distribuição de esforços de manutenção.



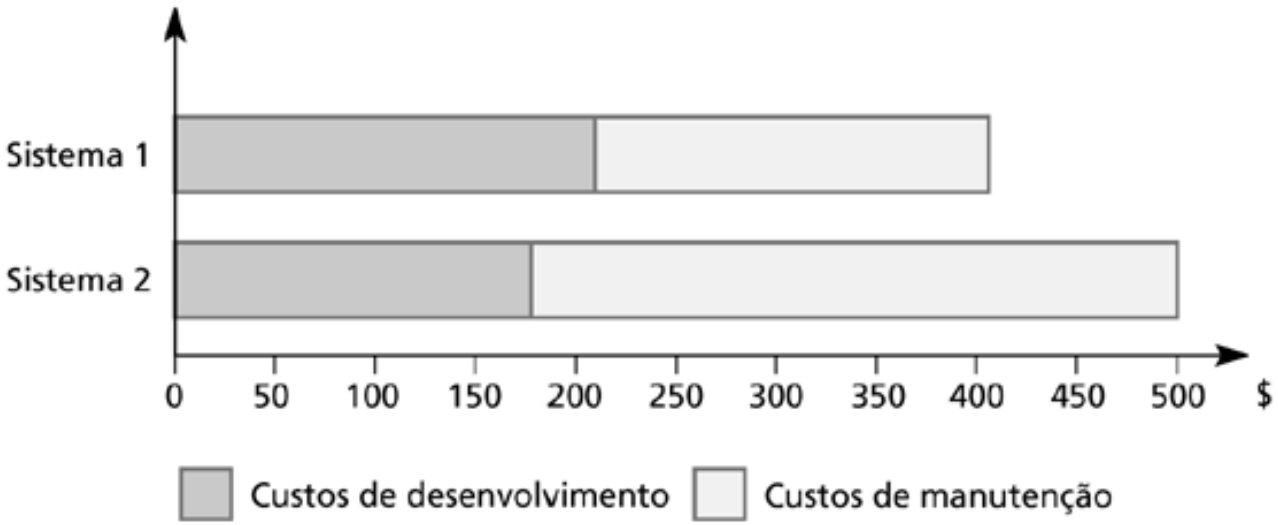
Custos de manutenção

- Geralmente, são maiores que os custos de desenvolvimento (de 2 a 100 vezes, dependendo da aplicação).
- São afetados por fatores técnicos e não técnicos.
- Aumentam conforme o software é mantido. A manutenção corrompe a estrutura do software, tornando a manutenção posterior mais difícil.
- Software em envelhecimento pode ter altos custos de suporte (por exemplo, linguagens antigas, compiladores, etc.).

Custos de desenvolvimento/manutenção

Figura 21.3

Custos de desenvolvimento e de manutenção.



Fatores de custo de manutenção

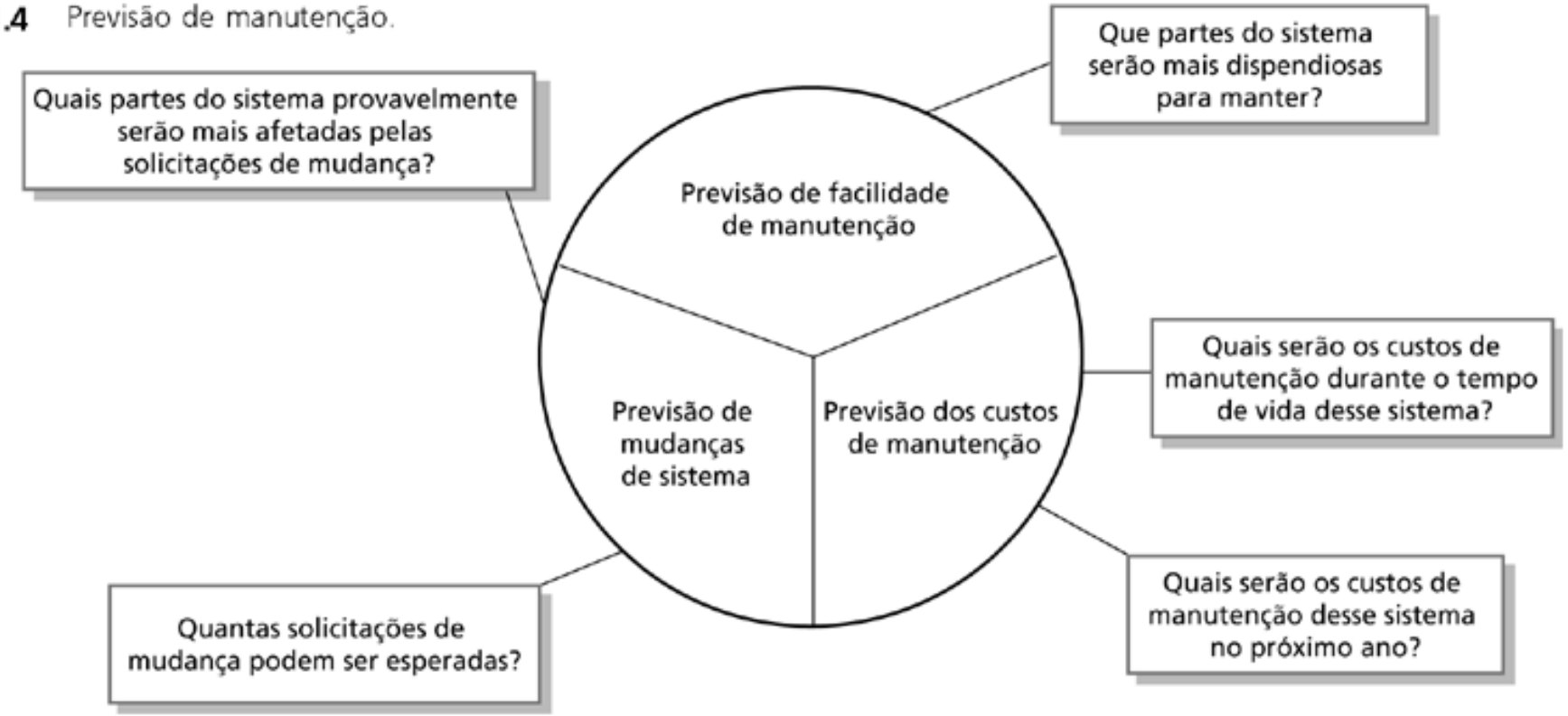
- Estabilidade da equipe
  - Os custos de manutenção são reduzidos se o mesmo pessoal estiver envolvido por algum tempo.
- Responsabilidade contratual
  - Os desenvolvedores de um sistema podem não ter responsabiidade contratual pela manutenção, portanto, não há incentivo para projetar para mudanças futuras.
- Habilidade do pessoal
  - O pessoal da manutenção geralmente é inexperiente e tem conhecimento limitado de domínio.
- Idade e estrutura do programa
  - À medida que os programas envelhecem, sua estrutura é degradada e se torna mais difícil de ser compreendida e modificada.

Previsão de manutenção

- A previsão de manutenção está relacionada à avaliação de quais partes do sistema podem causar problemas e ter altos custos de manutenção
  - A aceitação de mudança depende da facilidade de manutenção dos componentes afetados por ela;
  - A implementação de mudanças degrada o sistema e reduz a sua facilidade de manutenção;
  - Os custos de manutenção dependem do número de mudanças, e os custos de mudança dependem da facilidade de manutenção.

Previsão de manutenção

Figura 21.4 Previsão de manutenção.



Previsão de manutenção

- A previsão do número de mudanças requer o entendimento dos relacionamentos entre um sistema e seu ambiente.
- Sistemas fortemente acoplados requerem mudanças sempre que o ambiente é mudado.
- Fatores que influenciam esse relacionamento são
  - O número e a complexidade das interfaces de sistema;
  - O número de requisitos de sistema inerentemente voláteis;
  - Os processos de negócio nos quais o sistema é usado.

Métricas de complexidade

- As previsões de facilidade de manuteção podem ser feitas pela avaliação da complexidade dos componentes de sistema.

- Estudos mostraram que a maior parte do esforço de manutenção é despendida sobre um número relativamente pequeno de componentes de sistema.

- A complexidade depende
  - Da complexidade das estruturas de controle;
  - Da complexidade das estruturas de dados;
  - Do objeto, do método (procedimento) e do tamanho do módulo.

Métricas de processo

- As medições de processo podem ser usadas para avaliar a facilidade de manutenção
  - Número de solicitações para manutenção corretiva;
  - Tempo médio necessário para análise de impacto;
  - Tempo médio para implementar uma solicitação de mudança;
  - Número de solicitações de mudança pendentes.
- Se qualquer uma ou todas essas estão aumentando, isso pode indicar um declínio na facilidade de manutenção.

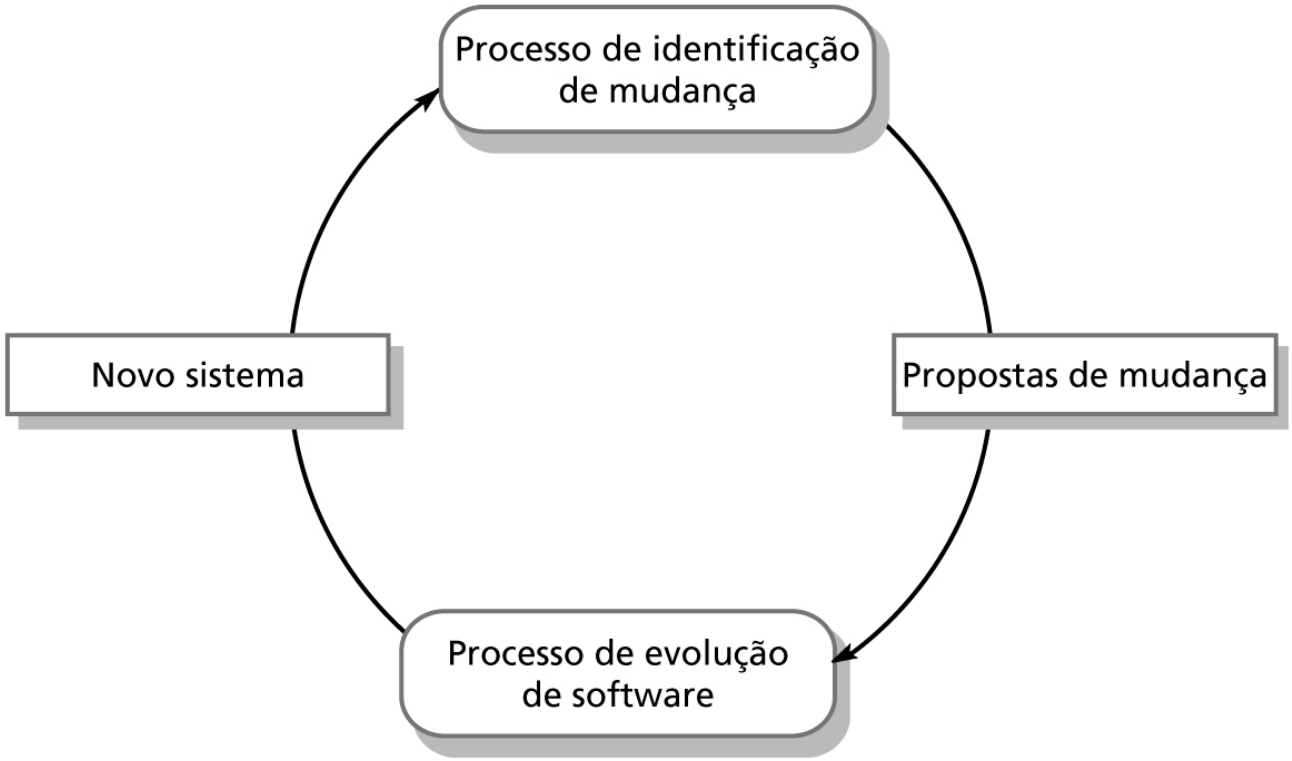
Processos de evolução

- Os processos de evolução dependem
  - Do tipo de software que está sendo mantido;
  - Dos processos de desenvolvimento usados;
  - Das habilidades e das experiências do pessoal envolvido.
- Propostas para mudança são os direcionadores para a evolução do sistema.
- Identificação de mudança e evolução continuam ao longo do ciclo de vida do sistema.

Identificação de mudança e evolução

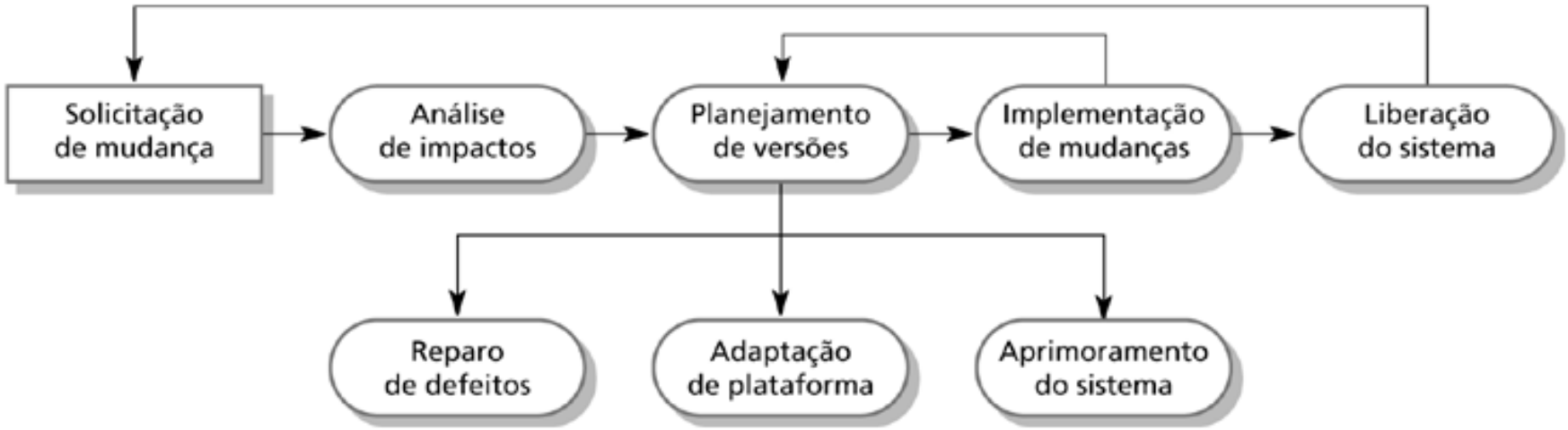
Figura 21.5

Processos de identificação de mudança e evolução.



O processo de evolução de sistema

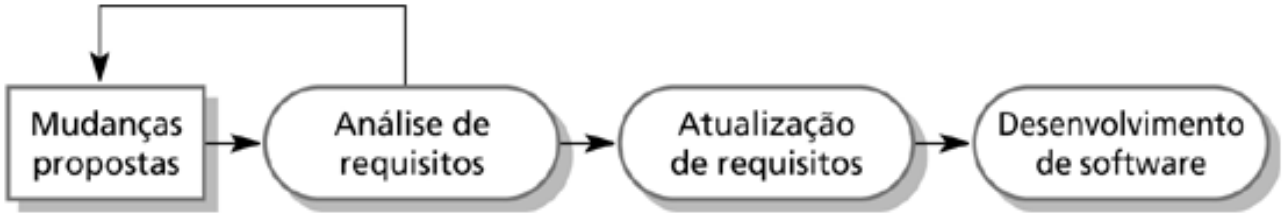
Figura 21.6 Processo de evolução de sistema.



Implementação de mudanças

Figura 21.7

Implementação de mudança.



Solicitações de mudança urgentes

- Mudanças urgentes podem ter de ser implementadas sem passar por todos os estágios do processo de desenvolvimento de software
  - Se um defeito sério de sistema tem de ser reparado;
  - Se mudanças no ambiente do sistema (por exemplo, atualização do OS) têm efeitos inesperados;
  - Se existem mudanças de negócio que necessitam de uma resposta muito rápida (por exemplo, o release de um produto concorrente)

Reparo de emergência

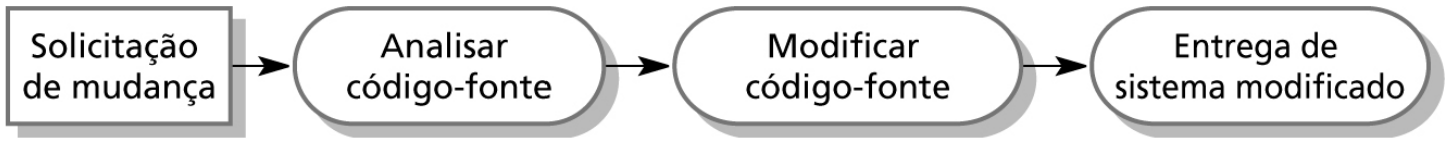


Figura 21.8

Processo de reparo de emergência.

Reengenharia de sistema

- É a reestruturação ou reescrita de parte ou de todo um sistema legado sem mudança na sua funcionalidade.
- É aplicável onde alguns subsistemas de um sistema de grande porte necessitam de manutenção frequente.
- A reengenharia envolve a adição de esforço para torná-los mais fáceis de manter. O sistema pode ser reestruturado e redocumentado.

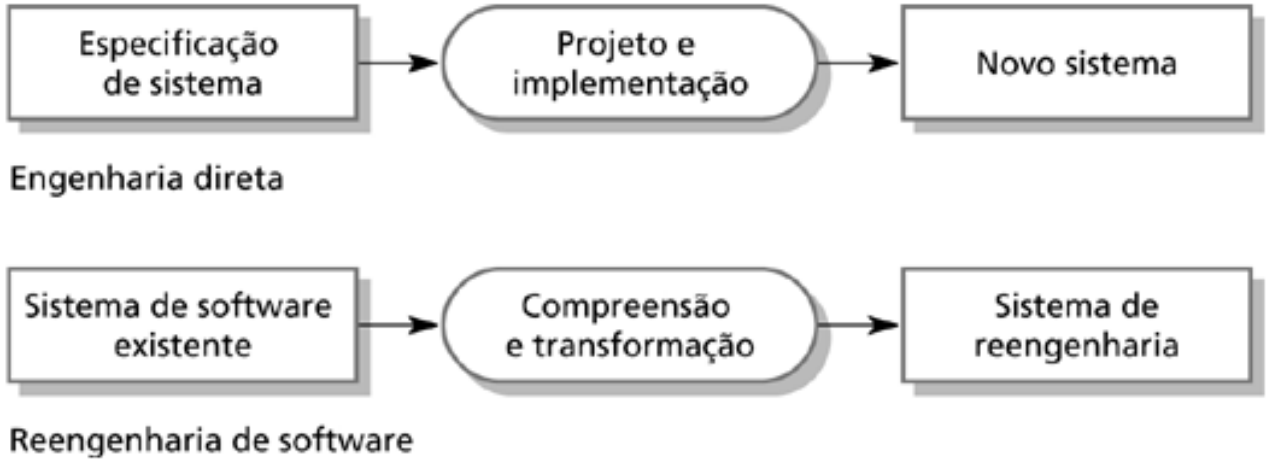
Vantagens da reengenharia

- Risco reduzido
  - Existe um alto risco no redesenvolvimento de software. Pode haver problemas de desenvolvimento, de pessoal e problemas de especificação.
- Custo reduzido
  - O custo de reengenharia é, frequentemente, menos significativo que os custos de desenvolvimento de um novo software.

Engenharia direta e reengenharia

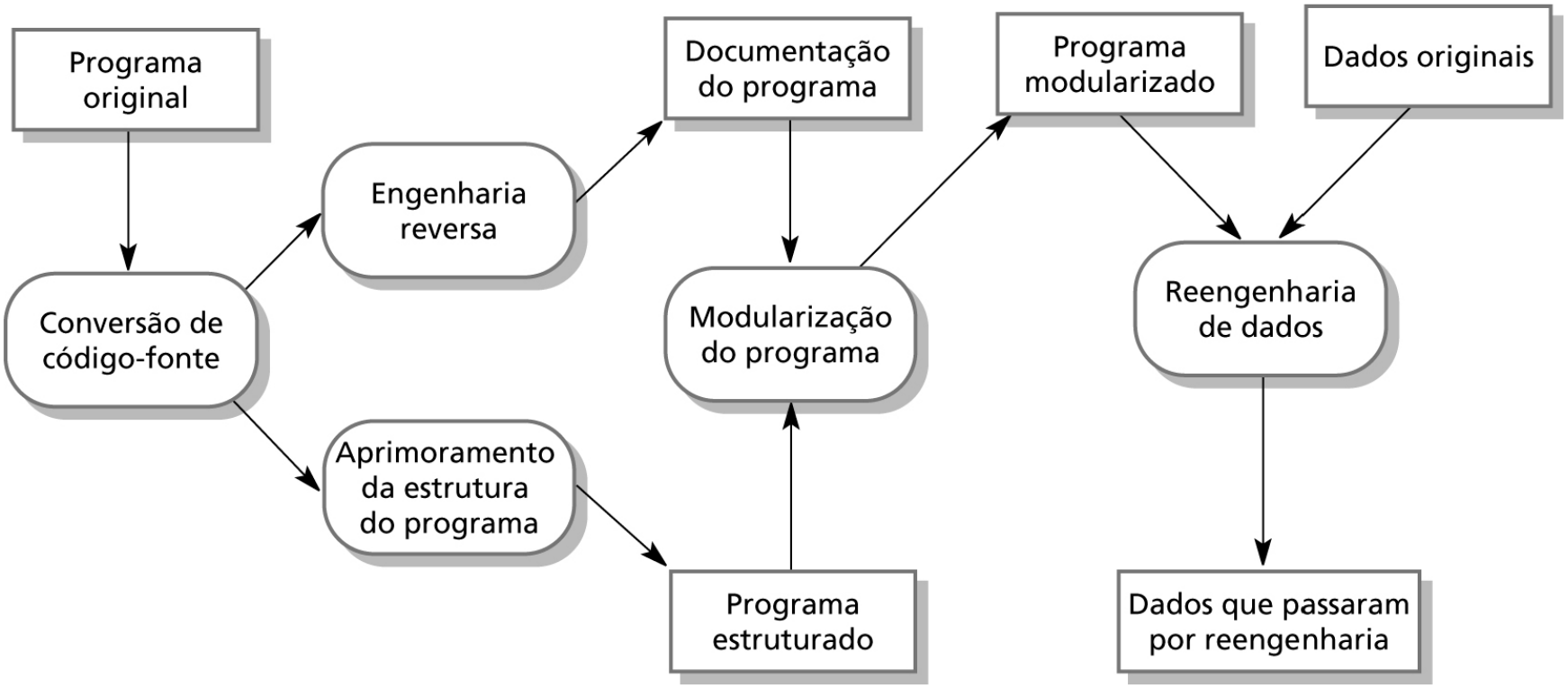
Figura 21.9

Engenharia direta e reengenharia.



O processo de reengenharia

Figura 21.10 Processo de reengenharia.



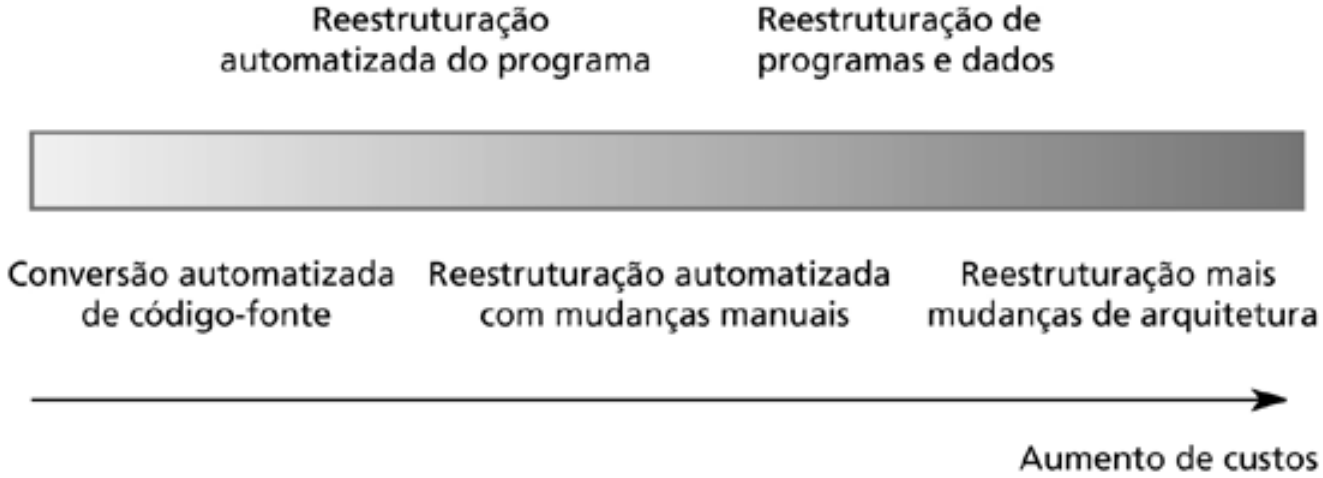
Atividades do processo de reengenharia

- Conversão de código-fonte
  - Converter o código para uma nova linguagem.
- Engenharia reversa
  - Analisar o programa para compreendê-lo.
- Aprimoramento da estrutura de programa
  - Analisar e modificar a estrutura para facilidade de entendimento.
- Modularização de programa
  - Reorganizar a estrutura do programa.
- Reengenharia de dados
  - Limpar e reestruturar os dados do sistema.

Abordagens de reengenharia

Figura 21.11

Abordagens de reengenharia.



Fatores do custo de reengenharia

- A qualidade do software que deve passar pela reengenharia.
- O apoio de ferramentas disponíveis para reengenharia.
- Extensão da conversão de dados.
- A disponibilidade do pessoal especializado
  - Isso pode ser um problema com sistemas antigos baseados em tecnologia que não são mais amplamente usadas.

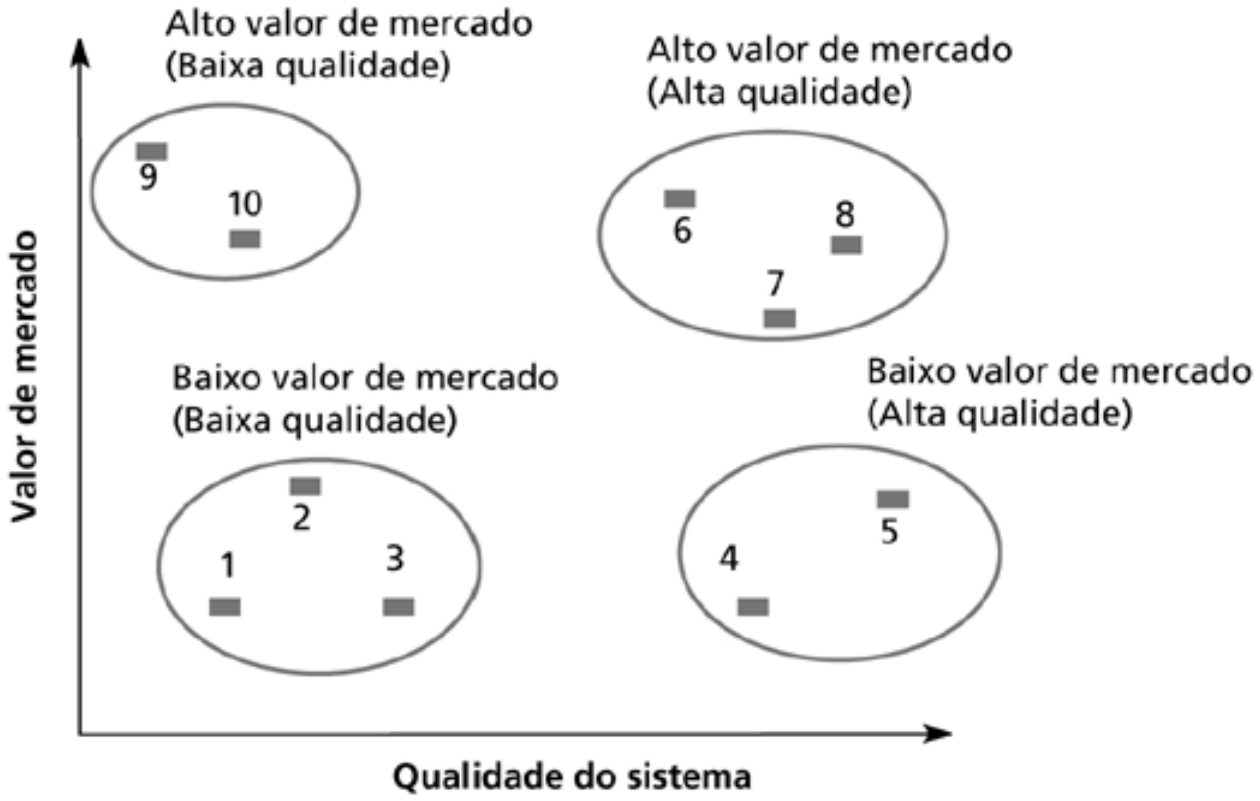
Evolução de sistemas legados

- Organizações que contam com sistemas legados devem escolher uma estratégia para a evolução desses sistemas
  - Descartar o sistema completamente e modificar os procesos de negócio de maneira que ele não seja mais necessário;
  - Deixar o sistema sem alterações e continuar com a manutenção regular;
  - Reengenharia do sistema para aprimorar sua facilidade de manuteção;
  - Substituir todo ou parte do sistema por um novo sistema.
- A estratégia escolhida depende da qualidade do sistema e de seu valor de negócio.

Qualidade de sistema e valor de negócio

Figura 21.12

Avaliação de sistemas legados.



Categorias de sistemas legados

- Baixa qualidade, baixo valor de mercado
  - Esses sistemas devem ser descartados.
- Baixa qualidade, alto valor de mercado
  - Esses sistemas contribuem de maneira importante para a empresa, mas sua manutenção é dispendiosa. Devem sofrer reengenharia ou ser substituídos caso um sistema adequado esteja disponível.
- Alta qualidade, baixo valor de mercado
  - Substituir com COTS, abandonar completamente ou manter.
- Alta qualidade, alto valor de mercado
  - Esses sistemas devem continuar em operação usando manutenção normal de sistema.

Avaliação de valor de negócio

- A avaliação deve levar em conta pontos de vista diferentes
  - Usuários finais do sistema;
  - Clientes do negócio;
  - Gerentes de linha;
  - Gerentes de TI;
  - Gerentes sênior.
- Entrevistar stakeholders diferentes e comparar resultados.

Avaliação da qualidade de sistema

- Avaliação do processo de negócio
  - Quão bem o processo de negócio apóia as metas atuais?
- Avaliação de ambiente
  - Quão efetivo é o ambiente do sistema e quão dispendiosa é sua manutenção?
- Avaliação de aplicação
  - Qual é a qualidade do sistema de aplicação de software?

Avaliação do processo de negócio

- Usar uma abordagem orientada a pontos de vista e procurar respostas dos stakeholders do sistema
  - Existe um modelo de processo definido: Ele é seguido?
  - Diferentes partes da organização usam processos diferentes para a mesma função?
  - Como o processo foi adaptado?
  - Quais são os relacionamentos com os outros processos de negócio? São necessários?
  - O processo é efetivamente apoiado pelo software de aplicação legado?

- Exemplo – um sistema de pedidos de viagem pode ter um baixo valor de negócio por causa do uso amplamente difundido de pedidos baseados em Web.

Avaliação de ambiente

Tabela 21.2 Fatores usados na avaliação de ambientes

Fator	Questões
Estabilidade do fornecedor	O fornecedor ainda existe? O fornecedor é financeiramente estável e é provável que continue a existir? Se o fornecedor não estiver mais atuando no mercado, alguém mais mantém os sistemas?
Taxa de falhas	O hardware tem alta taxa de falhas relatadas? O software de apoio 'trava' ou força o reinício do sistema?
Idade	Qual é a idade do hardware e do software? Quanto mais antigos o hardware e o software de apoio, mais obsoletos eles serão. Eles podem ainda funcionar corretamente, mas pode haver benefícios econômicos e de mercado em migrar para sistemas mais modernos.
Desempenho	O desempenho do sistema é adequado? Os problemas de desempenho têm efeito significativo sobre os usuários do sistema?
Requisitos de apoio	Qual apoio local é exigido pelo hardware e pelo software? Se houver altos custos associados a esse apoio, pode valer a pena considerar a substituição do sistema.
Custos de manutenção	Quais são os custos de manutenção de hardware e de licenças de software de apoio? Hardwares mais antigos podem ter custos de manutenção maiores do que os de sistemas modernos. O software de apoio pode ter altos custos anuais de licença.
Interoperabilidade	Existem problemas de interface do sistema com outros sistemas? Os compiladores podem, por exemplo, ser usados com versões atuais do sistema operacional? É necessária a emulação de hardware?

Avaliação de aplicação

Tabela 21.3 Fatores usados na avaliação de aplicações

Fator	Questões
Facilidade de compreensão	Qual é a dificuldade para compreender o código-fonte do sistema atual? Qual é a complexidade das estruturas de controle usadas? As variáveis têm nomes significativos que refletem suas funções?
Documentação	Qual documentação de sistema está disponível? A documentação é completa, consistente e atualizada?
Dados	Existe um modelo de dados explicito para o sistema? Até que ponto os dados estão duplicados nos arquivos? Os dados usados pelo sistema estão atualizados e consistentes?
Desempenho	O desempenho do sistema é adequado? Os problemas de desempenho têm efeito significativo sobre os usuários do sistema?
Linguagem de programação	Os compiladores modernos estão disponíveis para a linguagem de programação usada para desenvolver o sistema? A linguagem de programação ainda é usada no desenvolvimento do sistema?
Gerenciamento de configuração	Todas as versões de todas as partes do sistema são gerenciadas por um sistema de gerenciamento de configurações? Existe uma descrição explícita das versões de componentes usadas no sistema atual?
Dados de teste	Existem dados de teste para o sistema? Existe um registro de testes de regressão realizados quando novas características forem acrescentadas ao sistema?
Habilidades de pessoal	Existem pessoas disponíveis que tenham as habilidades para manter a aplicação? Existe somente um número limitado de pessoas que compreendem o sistema?

5/28/15

Medição de sistema

- Você pode coletar dados quantitativos para fazer uma avaliação da qualidade do sistema de aplicação
  - O número de solicitações de mudança no sistema;
  - O número de interfaces diferentes de usuário usados pelo sistema;
  - O volume de dados usados pelo sistema.

Pontos-chave

- Desenvolvimento e evolução de software devem ser um processo iterativo único.
- As leis de Lehman descrevem uma série de percepções de evolução de sistemas.
- Três tipos de manutenção são: correção de defeitos, modificação de software para um novo ambiente e implementação de novos requisitos.
- Para sistemas sob encomenda, os custos de manutenção geralmente excedem os custos de desenvolvimento.

Pontos-chave

- O processo de evolução é dirigido por solicitações de mudanças a partir dos stakeholders de sistema.
- A reengenharia de software está relacionado à reestruturação e redocumentação de software para torná-lo mais fácil de mudar.
- O valor de negócio de um sistema legado e sua qualidade deve determinar a estratégia de evolução que será usada.

Referências

- ©Ian Sommerville 2006 **Engenharia de Software, 8ª. edição. Capítulo 21**