
UNIP – Universidade Paulista
Campus Tatuapé – SP
Ciência da Computação
Engenharia de Software

Mudanças de Software

Prof. Marcelo Nogueira

Mudanças em software

- Gerir os processos de sistema em mudança de software

Objetivos

- Compreender três diferentes estratégias para fazer alterações em sistema de software
 - Manutenção de software
 - Evolução de arquitetura
 - Reengenharia de software
- Compreender os princípios de manutenção de software e por que ele tem uma manutenção tão dispendiosa
- Compreender como os sistemas legados podem ser transformados em sistemas cliente-servidor, a fim de prolongar sua duração e fazer uso efetivo de hardware moderno

Tópicos

- Dinâmica da evolução de programas
- Manutenção de software
- Evolução de arquitetura

Mudanças em software

- Mudança em software é inevitável
 - Novos requisitos emergem quando o software é utilizado
 - Mudanças no ambiente de negócios
 - Erros devem ser reparados
 - Novos equipamentos devem ser acomodados
 - A performance ou a confiabilidade podem ter que ser melhorada
- Um dos principais problemas para as empresas é implementar e gerir a mudança para seus sistemas legados

Estratégias para as mudanças em software

- Manutenção de software
 - As mudanças são feitas em resposta a requisitos modificados, mas a estrutura fundamental do software permanece estável
- Transformação de arquitetura
 - A arquitetura do sistema é alterada de uma arquitetura centralizada para uma arquitetura cliente-servidor
- Reengenharia de software
 - Nenhuma funcionalidade nova é adicionada ao sistema. Em vez disso, o sistema é modificado (reestruturado e reorganizado), a fim de tornar mais fácil sua compreensão e alteração
- Essas estratégias podem ser aplicadas juntas ou separadamente

Dinâmica da evolução de programas

- A dinâmica de evolução de programas é o estudo da mudança no sistema.
- Após um grande estudo, Lehman e Belady propuseram que houvesse um conjunto de "leis" referentes a mudanças nos sistemas
- São hipóteses ao invés de leis. Elas são aplicáveis aos grandes sistemas desenvolvidos por grandes empresas. Talvez menos aplicável em outros casos

Leis de Lehman

Lei	Descrição
Mudança contínua	Um programa utilizado em um ambiente do mundo real necessariamente tem de ser modificado ou se tornará de maneira progressiva menos útil nesse ambiente.
Aumento da complexidade	À medida que um programa em evolução se modifica, sua estrutura tende a se tornar mais complexa. Recursos extras precisam ser dedicados a preservar e simplificar a estrutura.
Evolução de um programa de porte grande	A evolução do programa é um processo auto-regulador. Os atributos do sistema, como tamanho, tempo entre releases e número de erros relatados, são aproximadamente invariáveis para cada release do sistema.
Estabilidade organizacional	Durante o tempo de duração de um programa, sua taxa de desenvolvimento é aproximadamente constante e independente dos recursos dedicados ao desenvolvimento do sistema.
Conservação da familiaridade	Durante o tempo de duração de um sistema, as mudanças incrementais em cada release são aproximadamente constantes.

Aplicabilidade das leis de Lehman

- Isto ainda não foi estabelecido
- Em geral, são aplicáveis para grandes sistemas adaptados desenvolvidos por grandes empresas
- Não é claro como devem ser modificadas para:
 - Pacotes de software;
 - Sistemas que incorporam um número significativo de componentes;
 - As pequenas organizações;
 - Os sistemas de médio porte.

Manutenção de software

- A manutenção de software é o processo geral de modificação de um sistema depois que ele foi colocado em uso
- As modificações podem ser:
 - Simples - destinadas a corrigir erros de código
 - Mais extensas - a fim de corrigir os erros de projeto
 - Significativas - com a finalidade de corrigir erros de especificação ou acomodar novos requisitos
- A manutenção de software não envolve, normalmente, mudanças maiores na arquitetura do sistema
- As mudanças são implementadas pela alteração dos componentes de sistema já existentes e adicionando-se novos componentes ao sistema

A manutenção é inevitável

- Os requisitos de sistema são suscetíveis de mudança, enquanto o sistema está sendo desenvolvido, porque o ambiente está mudando.
- Quando um sistema é instalado em um ambiente, o ambiente é alterado e, por conseguinte, os requisitos do sistema são alterados.
- Os sistemas devem ser mantidos, pois, eles devem permanecer úteis ao ambiente.

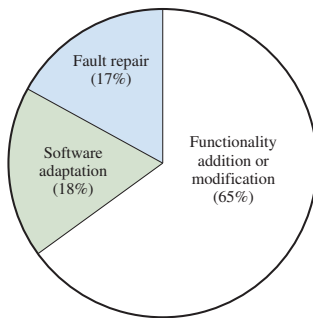
Tipos de manutenção

- Manutenção para reparar os defeitos no software
 - A correção de erros de codificação é um processo barato; os erros de projeto são mais dispendiosos e os erros de requisitos são os mais dispendiosos de corrigir
- Manutenção para adaptar o software a um ambiente operacional diferente
 - Esse tipo de manutenção é necessária quando algum aspecto do ambiente de sistema é modificado, como o hardware, o SO, etc.
- Manutenção para fazer acréscimos à funcionalidade do sistema ou modificá-la
 - Esse tipo de manutenção é necessária quando os requisitos de sistema são modificados, em resposta a mudanças organizacionais ou de negócios

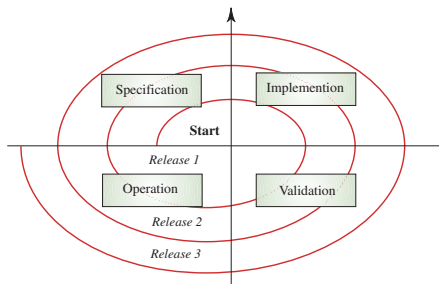
Outros tipos de manutenção

- **Manutenção corretiva**
 - É universalmente utilizada para se referir à manutenção para o reparo de defeitos
- **Manutenção adaptativa**
 - Algumas vezes, significa adaptação a um novo ambiente e, outras vezes, significa adaptar o software a novos requisitos
- **Manutenção evolutiva**
 - Às vezes, significa aperfeiçoar o software implementando novos requisitos e, em outros casos, se refere a manter a funcionalidade do sistema, melhorando sua estrutura e seu desempenho

Distribuição de esforço de manutenção



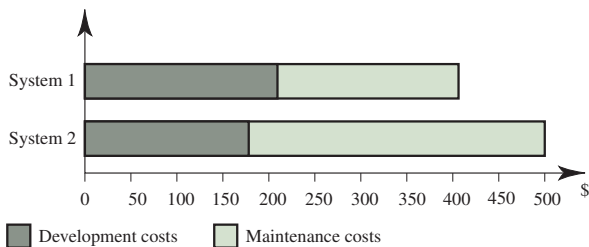
Modelo espiral de manutenção



Os custos de manutenção

- Os custos de manutenção do sistema representam uma grande proporção do orçamento da maioria das organizações que utilizam sistemas de software
- Na década de 80, constatou-se que as grandes organizações dedicavam pelo menos 50 por cento do total de seu esforço de programação para a evolução dos sistemas existentes
- Investir esforço ao projetar e implementar um sistema para reduzir os custos de manutenção é uma opção eficaz, em termos de custos.
- Boas técnicas de engenharia de software, como a especificação precisa, o uso do desenvolvimento orientada a objetos e do gerenciamento de configuração, contribuem para a redução dos custos de manutenção

Custos de desenvolvimento e de manutenção



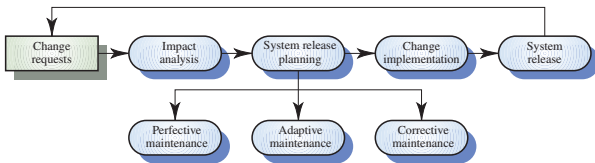
Fatores de custos de manutenção

- Estabilidade da equipe
 - Depois de um sistema ter sido entregue, é normal a equipe se dispersar para trabalhar em novos projetos. Os custos de manutenção seriam reduzidos se a mesma equipe estiver envolvida com a manutenção por algum tempo
- Responsabilidade contratual
 - O contrato para fazer a manutenção de um sistema, geralmente, é separado do contrato de desenvolvimento, e pode ser concedido a uma empresa diferente, não existindo nenhum incentivo para escrever o software de maneira que seja fácil de ser modificado
- Habilidade da equipe
 - O pessoal da manutenção, freqüentemente, tem pouca experiência e não está familiarizado com o domínio da aplicação
- Idade e estrutura do programa
 - À medida que os programas envelhecem, suas estruturas tendem a se tornar mais difíceis de ser entendidas e modificadas

Sistema evolucionários

- Em vez de desenvolver sistemas, mantê-los até que seja impossível realizar sua manutenção e, então substituí-los, temos que adotar a noção de sistemas evolucionários.
- Sistemas evolucionários são aqueles projetados para evoluírem e serem modificados em respostas a novas exigências

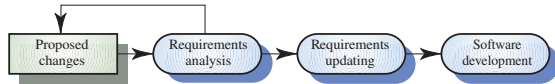
Processo de manutenção



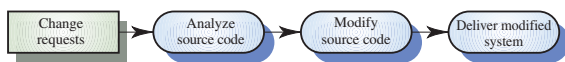
Pedidos de modificações

- Os pedidos de modificações são pedidos para mudança de sistema de usuários, clientes ou gestão
- Em princípio, todos os pedidos de modificações devem ser analisados cuidadosamente como parte do processo de manutenção e, em seguida, implementados
- Na prática, os pedidos de modificações devem ser implementados com urgência, por três razões:
 - Defeito de um sistema que deve ser reparado
 - Mudanças ambientais
 - Mudanças nos negócios que não foram previstas

Implementação de mudanças



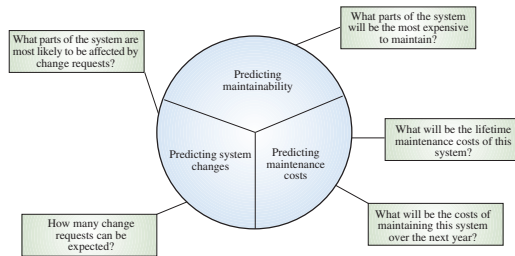
Reparos de emergência



Previsão de manutenção

- A previsão de manutenção está preocupada em avaliar quais as partes do sistema pode causar problemas e têm altos custos de manutenção
 - Se uma mudança no sistema deve ser aceita ou não, depende da facilidade de manutenção dos componentes de sistema afetados pela mudança
 - A implementação de mudanças no sistema tende a degradar a sua estrutura e reduzir sua facilidade de manutenção
 - Os custos de manutenção dependem da quantidade de mudanças e os custos de implementação dependem da facilidade de manutenção

Previsão de manutenção



Previsão da manutenção

- A previsão do número de pedidos de alteração para um sistema requer uma compreensão da relação entre o sistema e seu ambiente externo
- Sistemas fortemente acoplados exigem alterações sempre que o ambiente é alterado
- Fatores que influenciam esta relação são:
 - O número e a complexidade das interfaces de sistema
 - O número de requisitos de sistema inerentemente voláteis
 - Os processos de negócios em que o sistema é utilizado

Medida de complexidade

- Previsão de manutenção pode ser feita através da avaliação da complexidade dos componentes de sistema
- Estudos tem constatado que, quanto mais complexo for um sistema ou componente, mais dispendiosa será sua manutenção
- A complexidade depende:
 - Complexidade das estruturas de controle
 - Complexidade das estruturas de dados
 - Tamanho de procedimento e módulo

Métricas de processo

- Métricas de processo podem ser úteis para avaliar a facilidade de manutenção
 - Número de pedidos de manutenção corretiva
 - Tempo médio requerido para a análise de impacto
 - Tempo médio gasto para implementar um pedido de modificação
 - Número de pedidos de modificação importantes
- Se algum ou todos estes está a aumentar, isto pode indicar um declínio na manutenção

Evolução de arquitetura

- Existe a necessidade de modificar sistema centralizado de mainframe para sistemas distribuídos cliente-servidor
- Fatores para essa mudança
 - Custos de hardware. Servidores são mais baratos que mainframes
 - Expectativas quanto à interface com o usuário. Usuários preferem interfaces gráficas com usuário
 - Acesso distribuído aos sistemas. Usuários podem acessar o sistema a partir de diferentes localizações e por diferentes tipos de equipamentos

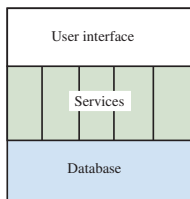
Fatores de distribuição

Fator	Descrição
Importância para os negócios	Os retornos de investimento na distribuição de um sistema legado dependem de sua importância para os negócios e por quanto tempo ele continuará sendo importante. Se a distribuição fornecer apoio mais eficiente para os processos de negócios estáveis, então será mais provável que essa seja uma estratégia de evolução eficaz em termos de custos.
Idade do sistema	Quanto mais antigo for o sistema o sistema, mais difícil será modificar sua arquitetura, porque as modificações anteriores terão degradado a estrutura do sistema.
Estrutura do sistema	Quanto mais modular for o sistema, mais fácil será modificar sua arquitetura. Se a lógica da aplicação, o gerenciamento de dados e a interface com o usuário do sistema estiverem estreitamente acoplados, será difícil separar as funções para realizar a migração.
Políticas de suprimento de hardware	A distribuição da aplicação pode ser necessária, se houver uma política da empresa de substituir computadores mainframes dispendiosas por servidores mais baratos.

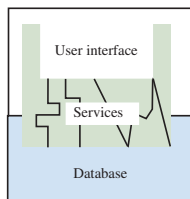
Estrutura de sistemas legados

- Ideal, para distribuição, que os sistemas legados tivessem uma estrutura com a interface com o usuário, com os serviços fornecidos pelo sistema e pelo banco de dados claramente separados
- Na prática, diferentes partes do serviço são implementadas em diferentes componentes; a interface com o usuário e o código do serviço estão integrados nos mesmos componentes

Estruturas de sistemas legados

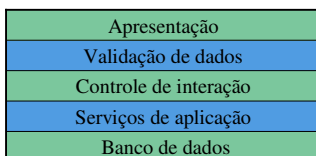


Ideal model for distribution

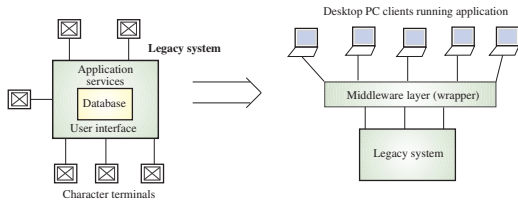


Real legacy systems

Modelo de distribuição em camadas



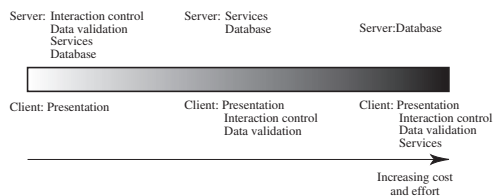
Distribuição do sistema legado



Opções de distribuição

- Quanto mais é distribuído do servidor para o cliente, maior será o custo da evolução de arquitetura
- Na opção mais simples, o computador cliente se ocupa somente com a apresentação da interface com o usuário e todas as outras funções são mantidas no servidor
- Na opção mais radical, o servidor somente gerencia os dados do sistema, e todas as outras funções são distribuídas para o cliente

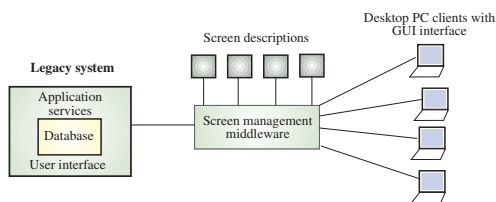
Espectro das opções de distribuição



Distribuição da interface com o usuário

- A distribuição da interface com o usuário tira proveito do poder local de processamento disponível em PCs, a fim de fornecer uma interface gráfica mais amigável aos usuários do sistema
- Se o sistema legado for estruturado de maneira que os serviços de interface com o usuário estejam claramente identificados, então o sistema legado poderá ser modificado para implementar a distribuição da interface com o usuário
- Caso contrário, o middleware de gerenciamento de tela pode traduzir as interfaces de texto para interfaces gráficas

Distribuição de interface do usuário



Estratégias de migração UI

Estratégia	Vantagens	Desvantagens
Implementação com a utilização do sistema de gerenciamento de janelas	Acesso a todas as funções de UI; assim, não há nenhuma restrição ao projeto de UI. Melhor desempenho de UI.	Dependente de plataforma. Pode ser mais difícil atingir a consistência da interface.
Implementação com a utilização de um Web browser	Custos de treinamento mais baixos, devido à familiaridade do usuário com a Web. Mais fácil de atingir a consistência da interface.	Desempenho de UI potencialmente menor. O projeto de interface fica limitado pelos recursos fornecidos por Web browsers.

Pontos-chave

- Entre as estratégias de modificação em software estão a manutenção de software, a evolução da arquitetura e a reengenharia de software
- Leis de Lehman são uma série de relacionamentos invariantes que afetam a evolução de um sistema de software
- Existem três tipos de manutenção que são:
 - Manutenção para reparar os defeitos no software
 - Manutenção para adaptar o software a um ambiente operacional diferente
 - Manutenção para adicionar ou modificar a funcionalidade do sistema

Pontos-chave

- O custo da mudança de software geralmente excede os custos de desenvolvimento de software
- Os altos custos de manutenção são devidos à falta de estabilidade de pessoal, aos contratos de desenvolvimento que não encorajam a produção de códigos de fácil manutenção, à falta da capacidade requerida para manter um sistema e a uma estrutura de sistema que se degradou, devido à idade e às regulares modificações no sistema
- A evolução da arquitetura envolve modificar a arquitetura de um sistema, a partir de uma arquitetura centralizada para uma arquitetura distribuída
- Uma estratégia comum da evolução de arquitetura para sistemas legados é encapsular o sistema legado como um servidor e implementar uma interface com o usuário distribuída, que acessa a funcionalidade do sistema por meio de um middleware

Bibliografia

- SOMMERVILLE, Ian, Engenharia de Software. 6 ed. São Paulo: Addison Wesley, 2003.
- <http://www.comp.lancs.ac.uk/computing/resources/IanS/SE6/Slides/PPT/ch27.ppt>
