

UNIVERSIDADE FEDERAL DO ACRE
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ENGENHARIA DE SOFTWARE II

Daricélio Moreira Soares



Evolução de Software

Mudança de Software

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

Mudança de Software

- Mudança de software é inevitável
 - Novos requisitos emergem quando o SW é usado
 - O ambiente do negócio muda
 - Erros devem ser reparados
 - Novo equipamento deve ser incorporado
 - O desempenho ou confiabilidade podem ser melhorados
- Um problema chave para as organizações é implementar e gerir mudança para os seus sistemas legados

Estratégias de mudança de software

- Manutenção de software
 - Mudanças são feitas em resposta a requisitos modificados, mas a estrutura fundamental do SW é estável
- Transformação arquitetural
 - A arquitetura do sistema é modificada. Ex: centralizada para uma distribuída
- Reengenharia de software
 - Nenhuma funcionalidade é adicionada ao sistema, mas ele é re-estruturado e re-organizado para facilitar mudanças futuras
- Estas estratégias podem ser aplicadas conjunta ou separadamente

Dinâmica de evolução de programas

- A dinâmica de evolução de programas é o estudo dos processos de mudança do sistema
- Depois de um estudo empírico, propuseram que há um número de leis que são aplicadas a todos os sistemas enquanto eles evoluem
- Na realidade há observações sensatas que são aplicadas a sistemas de larga escala desenvolvidos por grandes organizações

Leis de Lehman

- Mudança contínua – Um programa num ambiente real necessariamente muda ou torna-se progressivamente menos útil
- Aumento da complexidade – Com a evolução do programa sua estrutura tende a ficar mais complexa
- Evolução de programas de larga escala – A evolução de um programa é um processo auto-regulável
- Estabilidade organizacional – mudanças de recursos tem poucos efeitos na evolução do sistema
- Conservação da familiaridade - a mudança incremental é constante
- Crescimento contínuo - Todo software deve ter o conteúdo funcional continuamente ampliado durante o ciclo de vida para manter a satisfação dos seus usuários.

Aplicabilidade das leis de Lehman

- São aplicadas a sistemas de larga escala desenvolvidos por grandes organizações
- Não está claro como podem ser adaptadas para
 - Pequenas organizações
 - Sistemas de médio porte

Manutenção de software

- É modificar um programa depois que ele foi colocado em uso
- A manutenção normalmente **não** envolve grandes mudanças na arquitetura do sistema
- As mudanças são implementadas modificando os componentes existentes e adicionando novos componentes ao sistema
- A manutenção é inevitável
 - Os requisitos do sistema mudam enquanto o sistema está sendo desenvolvido porque o ambiente muda
 - Se os sistemas estão fortemente acoplados com seu ambiente, quando um sistema é instalado, ele muda o ambiente e portanto muda os requisitos do sistema
 - Os sistemas devem ser mantidos se quer que eles permaneçam úteis num ambiente

Distribuição do esforço de manutenção



Evolução



Tipos Evolução

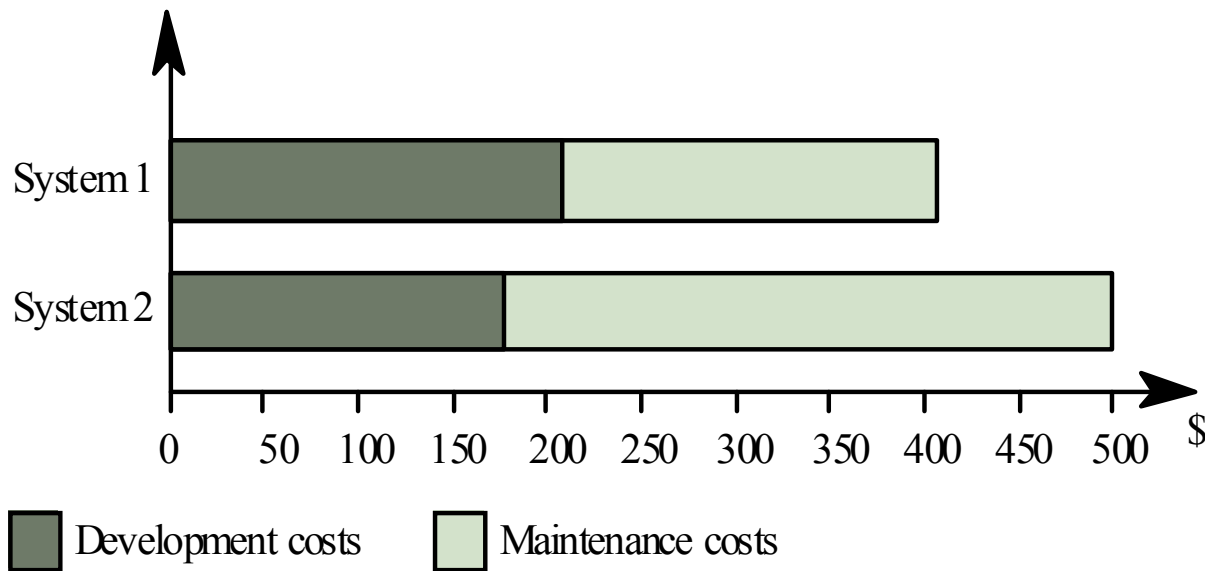
- Manutenção emergencial
 - Não programada
 - Mantém temporariamente o sistema funcionando
 - Necessita uma manutenção corretiva posterior
- Manutenção corretiva
 - Reativa
 - Corrige problemas reportados
 - Faz o software voltar a atender aos requisitos
- Manutenção preventiva
 - Pró-ativa
 - Corrige problemas latentes

Tipos Evolução

- Manutenção adaptativa
 - Mantém o software usável após mudanças no ambiente
- Manutenção perfectiva
 - Provê melhorias para o usuário
 - Melhora atributos de qualidade do software

Custos de manutenção

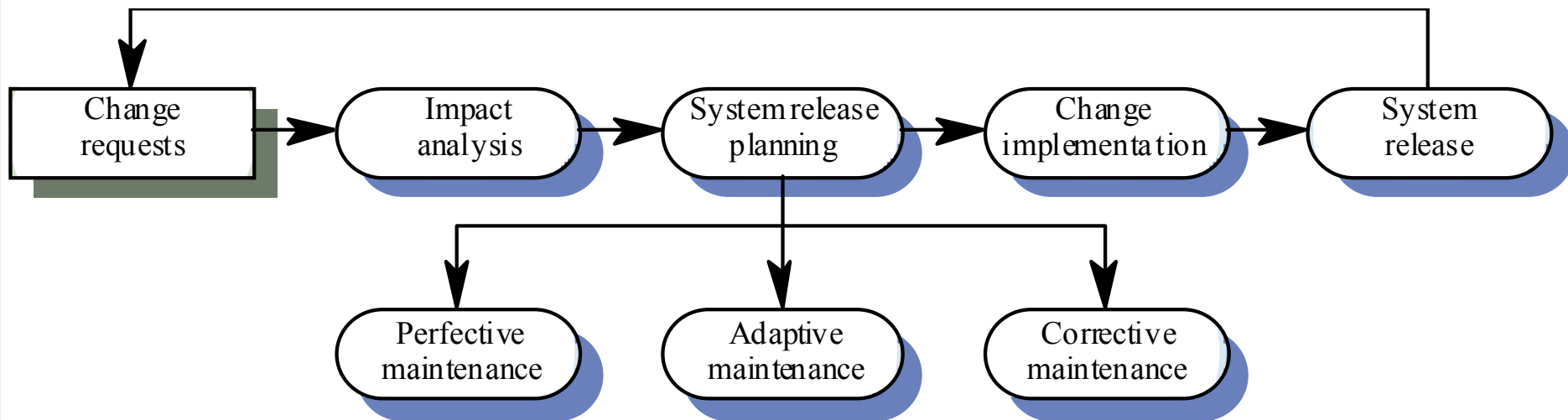
- Usualmente maiores do que os custos de desenvolvimento (de duas a cem vezes, dependendo da aplicação)
- Afetados por fatores técnicos e não técnicos
- Aumenta enquanto o SW é mantido. A manutenção corrompe a estrutura do SW, tornando-a mais difícil. SW antigos podem ter custos de manutenção altos



Manutenção

- Fatores de custos
 - Estabilidade da equipe - Custos de manutenção são menores se o mesmo staff está envolvido com o sistema
 - Responsabilidade contratual – Se os engenheiros de SW podem não ter nenhuma responsabilidade contratual p/ a manutenção, portanto não há incentivo em desenhar p/ mudança futura
 - Habilidades do staff - O staff de manutenção é inexperiente e tem conhecimento limitado do domínio
 - Idade e estrutura do programa - Enquanto o programa envelhece, sua estrutura se degrada tornando-o mais difícil de compreender e mudar
- Software evolutivo
 - Em vez de ter fases separadas p/ manutenção e desenvolvimento, é preferível que o SW seja desenhado permitindo sua evolução contínua no seu ciclo de vida

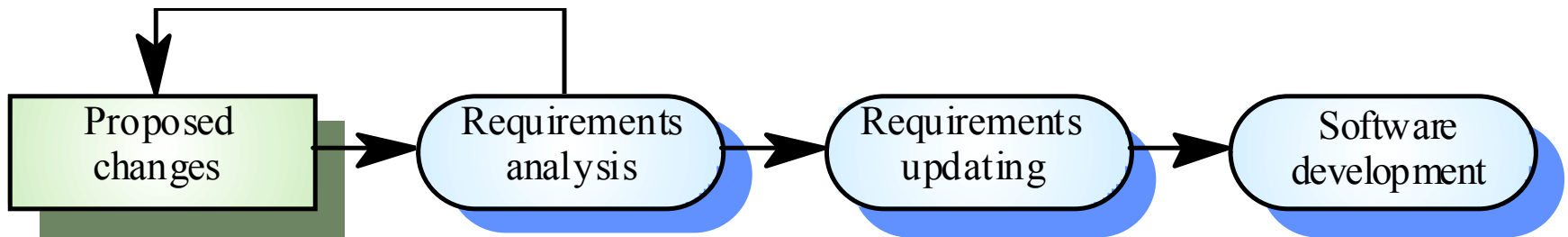
O processo de manutenção



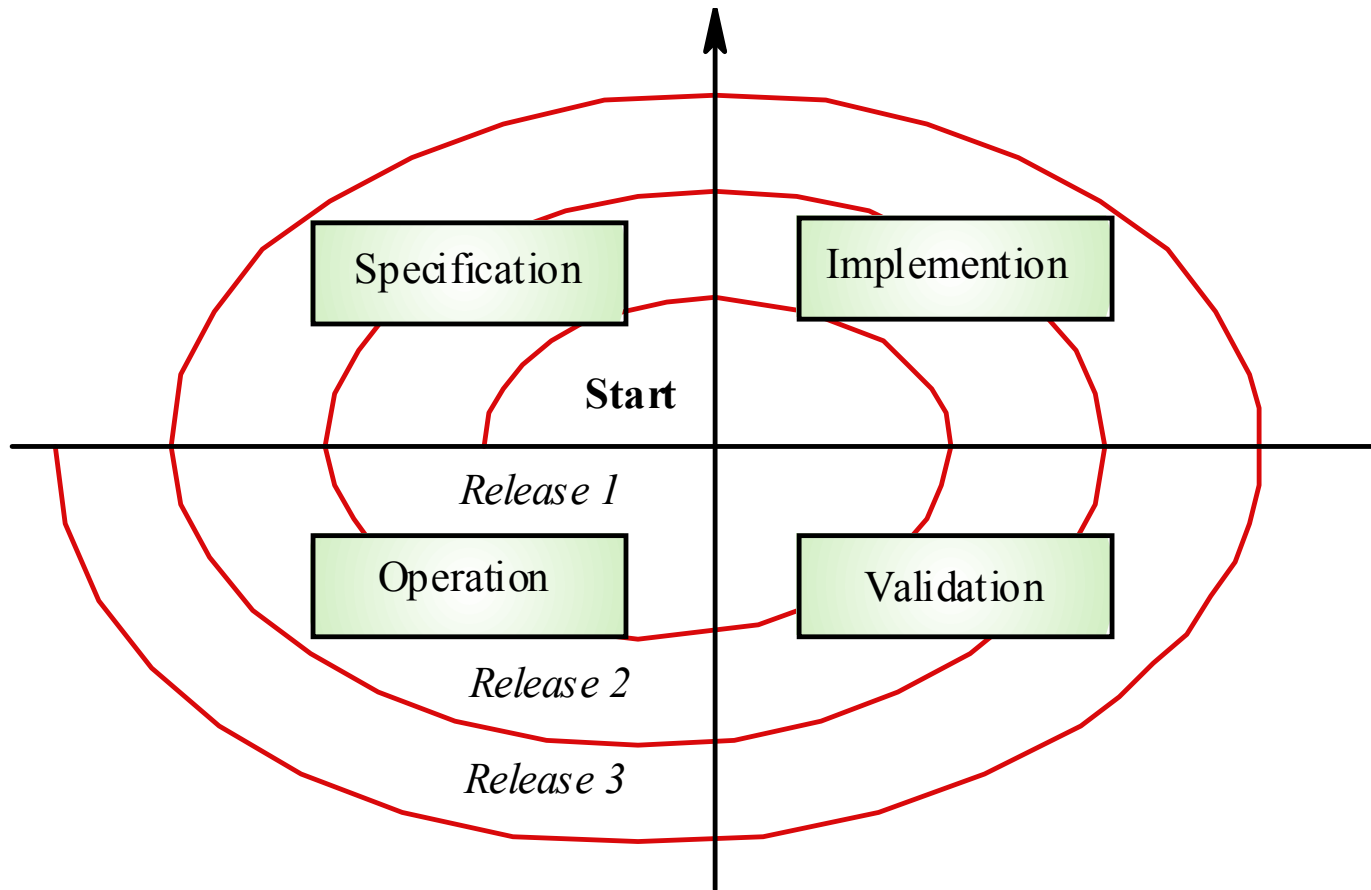
Pedidos de mudança

- São feitos pelos utilizadores, clientes ou gestão
- Em princípio devem ser analisados cuidadosamente como parte do processo de manutenção, e então, implementados
- Na prática, alguns pedidos devem ser implementados urgentemente
 - Reparar falhas
 - Mudanças no ambiente
 - Mudanças urgentes do negócio

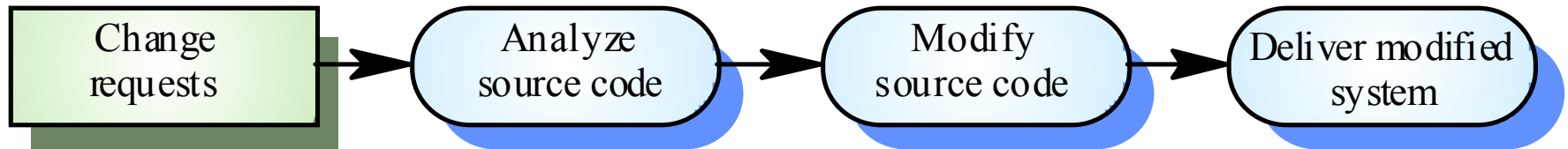
Implementação da mudança



Modelo espiral p/ manutenção



Reparo de emergência



Previsão de mudança

- A previsão de mudança preocupa-se em avaliar as partes do sistema que podem causar problemas e ter custos de manutenção altos
 - A aceitação da mudança depende da manutenibilidade dos componentes afetados pela mudança
 - Implementar mudanças degrada o sistema e reduz sua manutenibilidade
 - Custos de manutenção dependem do número de mudanças e os custos de mudança dependem da manutenibilidade

Previsão da mudança

- Prever o número de mudanças requer a compreensão das relações entre um sistema e seu ambiente
- Sistemas fortemente acoplados c/ o ambiente requerem mudanças quando quer que o ambiente mude
- Fatores que influenciam:
 - Número e complexidade das interfaces do sistema
 - Número de requisitos inerentemente voláteis
 - Os processos do negócio onde o sistema é utilizado

Métricas de complexidade

- Previsões de manutenção podem ser feitas através da avaliação da complexidade dos componentes do sistema
- Estudos mostram que a maior parte dos esforços de manutenção são gastos em um número relativamente pequeno de componentes de um sistema
- A complexidade depende das estruturas de controle, das estruturas de dados e do tamanho dos módulos e procedimentos
- Métricas do processo
 - Medidas de processo podem ser usadas p/ medir a manutenibilidade
 - Número de pedidos p/ a manutenção corretiva
 - Tempo médio requerido p/ a análise de impacto
 - Tempo médio p/ implementar um pedido de mudança
 - Número de pedidos de mudanças importantes
 - Se qualquer um destes aumentar, pode indicar um declínio na manutenibilidade

Evolução da arquitetura

- Há uma necessidade de converter muitos sistemas legados de uma arquitetura centralizada p/ uma cliente-servidor
- Razões
 - Custos de HW. Servidores são mais baratos que mainframes
 - Usuários querem acessar ao sistema a partir de computadores diferentes e localizados em pontos geograficamente distantes
- Fatores de Impacto
 - A importância do negócio
 - A idade do sistema (quanto mais velho mais difícil)
 - Estrutura do sistema (quanto mais modularizado, mais fácil)

Contratos de Manutenção

- Tipo 1
 - Um único contrato para desenvolvimento e manutenção
 - Valor fixo, disponível para todos os tipos de manutenção
- Tipo 2
 - Contrato separado para manutenção
 - Período de manutenções corretivas predefinido
 - Cada manutenção preventiva, adaptativa ou perfectiva contratada separadamente

Quanto Cobrar?

- Tempo
- Custos
- Pessoas
- Esforço

Atividade N2

- Resumo de Artigo
 - Václav Rajlich. 2014. **Software evolution and maintenance**. In Proceedings of the on Future of Software Engineering (FOSE 2014). ACM, New York, NY, USA, 133-144. DOI: <http://dx.doi.org/10.1145/2593882.2593893>
- Resumo manuscrito
- Entrega: 16/07/2018
- 1 página
- Estrutura obrigatória
 - Visão geral, Metodologia, Principais contribuições, Análise crítica: pontos + e –