

Manutenção e Ferramentas CASE

Marcelo Morandini
EACH - USP



O que é manutenção de software?

- “... mudanças que devem ser feitas nos programas de computadores depois de terem sido distribuídos ao cliente.” (McClure 1983)
- “... realização de atividades exigidas para manter o sistema operacional depois de ele ter sido aceito e colocado em operação.” (FIPS 1984)

Definição

- O que é manutenção de software?
 - “Manutenção cobre a vida do software do momento em que é instalado até ser desativado.” (von Mayrhauser 1990)
 - “... modificação de um produto de software depois de distribuído para ou corrigir defeitos, ou melhorar o desempenho e outros atributos, ou para adaptar o produto a um ambiente modificado.” (IEEE 1219 1993)

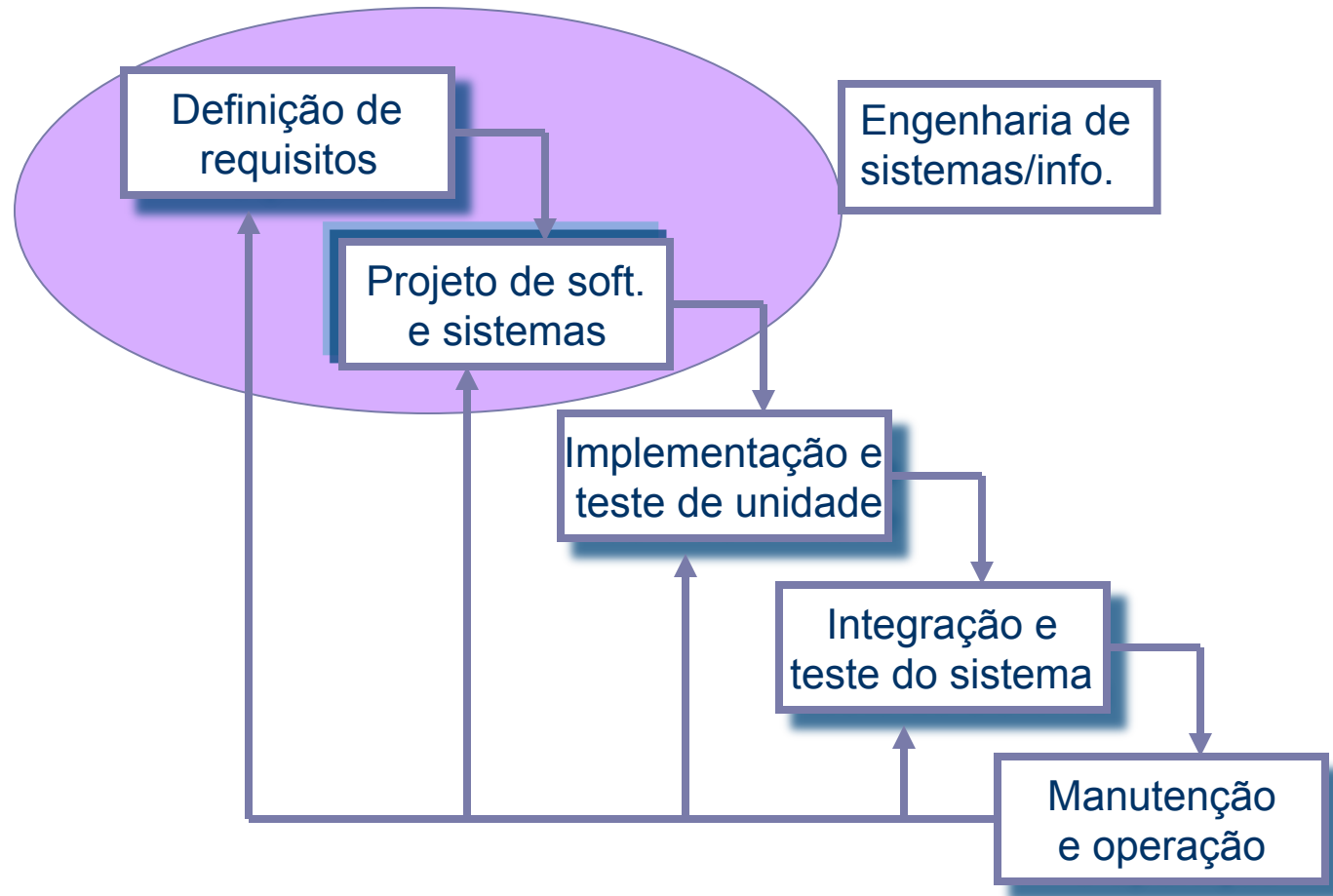
Definição

- O que é manutenção de software?
 - “... o produto de software sofre modificações no código ou na documentação associada devido a um problema ou uma necessidade de melhoria. O objetivo é modificar o produto de software existente mantendo, porém, sua integridade.” (ISO/IEC 12207 1995)

Manutenção no Ciclo de Vida

- Onde está inserida atividade de manutenção no ciclo de vida de software?
- Ciclo de vida? Lembrando...
 - Cascata;
 - Prototipação;
 - Espiral;
 - Incremental;
 - Evolucionário.

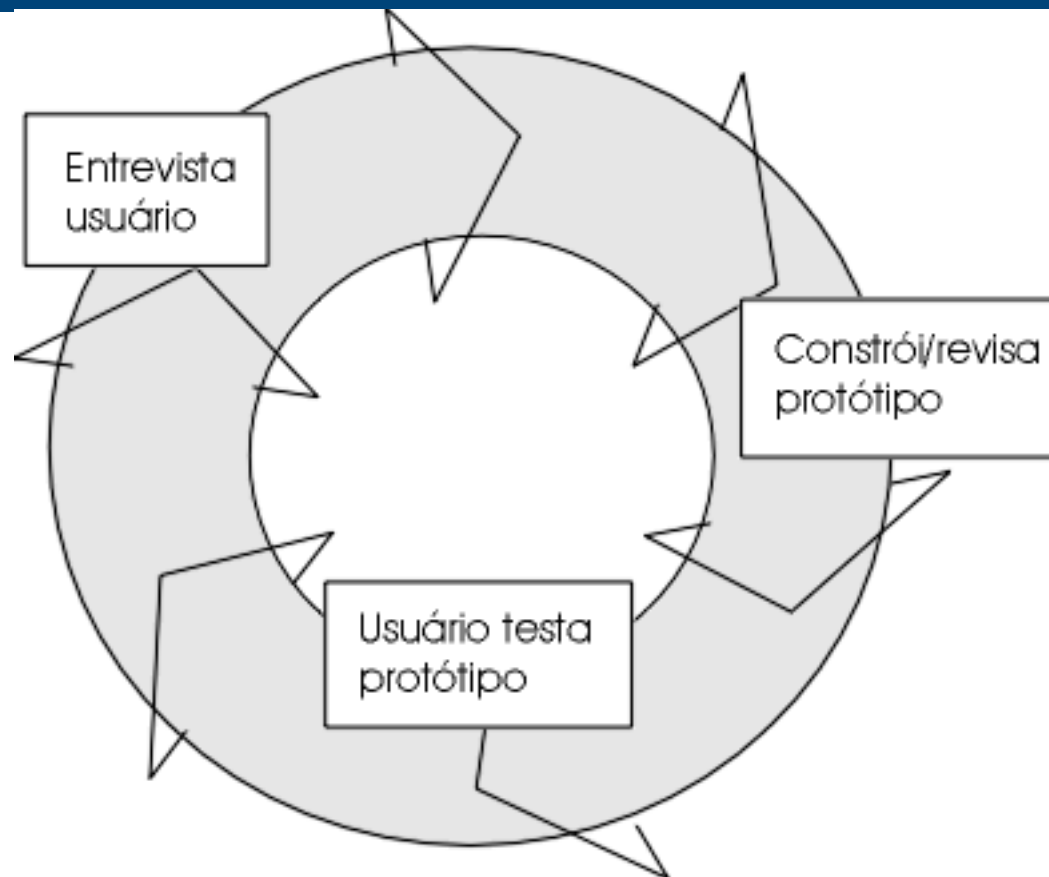
Ciclo de Vida em Cascata



Ciclo de Vida em Cascata

- Problemas:
 - projetos raramente seguem um fluxo seqüencial;
 - difícil de definir todas as restrições *a priori*;
 - primeira versão em um estágio tardio;
- Apesar dos problemas,
 - largamente utilizado ainda hoje;
 - melhor que não ter uma sistemática de desenvolvimento;
- Manutenção é uma atividade pós-entrega do software.

Prototipação



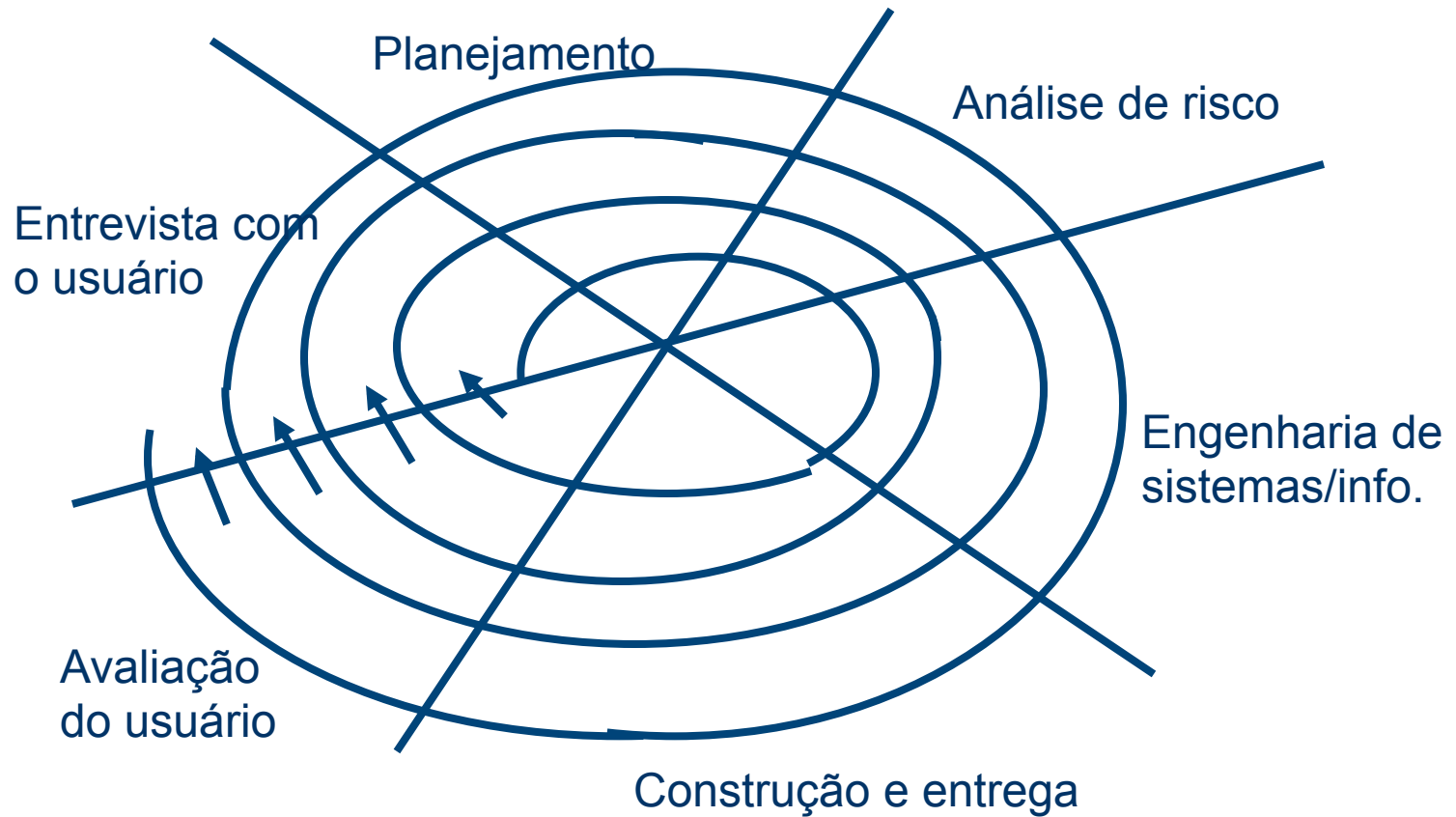
Prototipação

- Benefícios:
 - idéias confusas podem ser identificadas
 - entendimento errado pode ser esclarecido
 - complementar idéias vagas
 - utilidade do sistema antes de pronto

Prototipação

- Problemas:
 - usuários vêem o protótipo como produto final e pensam que o produto está praticamente pronto;
 - ferramentas/linguagem provisórias podem se tornar definitivas - inércia;
 - manutenção de um protótipo tende a ser problemática: falta de estrutura porque não foi preparado para ser a versão final.

Espiral (Boehm)



Necessidade de Manutenção

- Possíveis razões para realizar manutenção:
 - Defeitos são detectados depois de entregue o sistema.
 - Usuários *descobrem* novas funcionalidades necessárias.
 - Sistema precisa executar em um novo ambiente, com nova interface.
 - Sistema precisa ser alterado para evitar problemas futuros ou melhorar o desempenho.

Tarefas de Manutenção

- Correção de defeitos;
- Correção de falhas de projeto;
- Interfaceamento com outros sistemas;
- Melhorias no sistema;
- Realização de mudanças necessárias ao sistema.
- Realização de mudanças em arquivos ou bases de dados.

Tarefas de Manutenção

- Melhorias no projeto;
- Converter programas para que diferentes sistemas de software, hardware e de telecomunicações possam ser utilizados.
- Melhorias de desempenho e prevenção de problemas.

Necessidade de Manutenção

- Manutenção é o resultado de um desenvolvimento ruim?
 - Elicitação de requisitos mal feita?
 - Análise, projeto e codificação ruins introduziram defeitos nos programas?
- Se usássemos ferramentas CASE ou COTS (components off-the-shelf) esses problemas não seriam resolvidos?

Leis de Lehman

- Primeira Lei:
 - *Lei da Mudança Contínua*
 - Sistemas precisam mudar para continuar serem úteis.
- Segunda Lei:
 - *Lei do Aumento da Complexidade*
 - Estrutura do código tende a degradar com as sucessivas manutenções.

Categorias de Manutenção

- ***Corretiva:***
 - Mudanças causadas por defeitos no sistema.
- **Exemplo:**
 - defeitos residuais do sistema encontrados pelo usuário.
- **Metáfora carro:**
 - *recall* para trocar mangueirinha do radiador do VW Gol 1995.

Categorias de Manutenção

- ***Adaptativa:***

- Mudanças devido a alterações no ambiente em que o sistema vai executar.

- **Exemplo:**

- Sistema deve ser adaptado para executar na nova versão do sistema operacional.

- **Metáfora carro:**

- Alteração do motor do carro para funcionar com gás natural.

Categorias de Manutenção

- ***Aperfeiçoadora*** (perfectiva):
 - mudanças são realizadas para satisfazer as novas necessidades do usuário.
- **Exemplo:**
 - Sistema de venda *on-line* deve também incluir vendas de brinquedos e material de escritório.
- **Metáfora carro:**
 - Usuário deseja perua com aspecto esportivo (e.g., *pallio adventure*)

Categorias de Manutenção do IEEE

- Três categorias de manutenção: corretiva, aperfeçoadora e preventiva.
 - Aperfeçoadora = adaptativa + aperfeçoadora
- **Preventiva:**
 - Manutenção realizada visando prevenir futuros problemas.
- Exemplo:
 - Sistemas de missão crítica (ônibus espacial, software de aviãoica) requerem a constante prevenção de problemas.
- Metáfora carro:
 - Adicionar um *break light* no carro.
- Manutenção preventiva é vista por alguns autores como manutenção corretiva.

Importância da Categorização

- Diferenciar entre *correção* (manutenção corretiva) e *melhoramento* (aperfeiçoadora, adaptativa, preventiva).
- Pergunta a ser feita:
 - A mudança é para corrigir defeitos ou melhorar o software?
 - Se é para melhorar, que tipo de melhora? Aperfeiçoamento, adaptação ou prevenção?

Dificuldade da Manutenção

- Falta de rastreabilidade das representações pré-código até o código gerado.
- Alterações tendem a degradar o código.
- Usuários confundem manutenção de hardware com a manutenção de software.
- Manutenção de hardware visa aumentar a *confiabilidade*.
- Manutenção de software visa aumentar a *confiabilidade* e adicionar *funcionalidade*.

Quem deve realizar a manutenção? Desenvolvedor?

- Vantagens do desenvolvedor:
 - Desenvolvedor é quem mais conhece o sistema;
 - Não há necessidade de elaborar documentação;
 - Não há necessidade de estabelecer comunicação entre desenvolvedor e mantenedor (são a mesma pessoa!);
 - Usuários precisam se comunicar com uma única organização;
 - Diversidade de trabalho na organização desenvolvedora.

Quem deve realizar a manutenção? Desenvolvedor?

- Desvantagens do desenvolvedor:
 - Será que o desenvolvedor deseja dar manutenção? E se ele sair da organização?
 - Pouca documentação é escrita.
 - Desenvolvedor pode ser alocado em um projeto mais prioritário.

Quem deve realizar a manutenção? Uma outra organização?

- Vantagens da organização separada:
 - Documentação gerada é melhor (pelo menos alguma é gerada...);
 - Procedimentos formais de entrega para a organização mantenedora são definidos;
 - Pontos fortes e fracos do software são *descobertos* pelos mantenedores;
 - Equipe *sabe* que vai dar manutenção.

Quem deve realizar a manutenção? Uma outra organização?

- Desvantagens da organização separada:
 - Transição entre organizações (desenvolvedora e mantenedora) pode ser lento;
 - Muito treinamento é necessário;
 - Pode faltar recursos (dinheiro) para as tarefas de manutenção;
 - Leva-se tempo para aprender/entender o sistema;

Quem deve realizar a manutenção? Uma outra organização?

- Desvantagens da organização separada:
 - Mantenedores podem não estar preparados para as tarefas de manutenção;
 - Leva-se tempo para ajustar a organização mantenedora;
 - Suporte ao usuário pode ficar prejudicado;
 - Credibilidade do software pode ficar comprometida nesse período.

Qual a melhor abordagem de manutenção?

- Depende...
 - Para software grande, de grandes organizações, a tendência é transferir a atividade de manutenção para uma outra organização;
 - Em pequenas organizações, os desenvolvedores tendem a ser responsáveis pela manutenção.
- Independentemente da opção, é importante planejar a atividade de manutenção.

Resumo

- Definição do conceito de manutenção.
- Manutenção no ciclo de vida.
- Tarefas de manutenção.
- Leis de Lehman.
- Categorias de manutenção.
- Quem deve fazer a manutenção?
Desenvolvedor ou um terceiro?

Exercícios

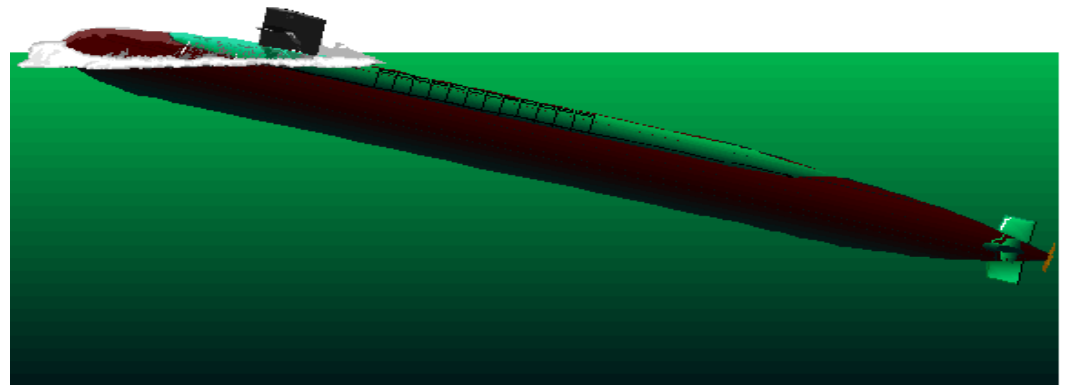
1. Qual o ciclo de vida de software na sua organização? Cascata? Prototipação? Espiral? Incremental? Se diferente de todos, explique.
2. Como são realizadas as tarefas de manutenção na sua organização? Elas são planejadas? Como estão inseridas no ciclo de vida?
3. Liste três requisições de manutenção típicas que ocorre no seu ambiente de trabalho e classifique-as.
4. Quem é o responsável pela manutenção na sua organização? Você acha esta a melhor solução?
5. Pela segunda lei de Lehman, o software tende a degradar sua estrutura depois de sucessivas manutenções. Como explicar então que produtos com M Office, Linux, Oracle etc. estão cada vez melhores depois de várias versões?

Custo da Manutenção

- É senso comum de que os custos de manutenção são bem maiores do que os de desenvolvimento.

Superfície = desenvolvimento

Submerso =
Manutenção



Custo da Manutenção no Ciclo de Vida

- Estimativas de custo de manutenção:
 - Começo dos anos 70: **40%**;
 - Começo dos anos 80: **55%**;
 - Fim dos anos 80: **75%**;
 - Começo dos anos 90: **90%**.
- Estranho... O custo de manutenção não deveria diminuir com o uso de tecnologias como *orientação a objetos*.

Análise de Custo de Manutenção

- Manutenção é essencialmente uma tarefa realizada por seres humanos (programadores). E gente custa dinheiro!
- Principal tarefa de manutenção: *compreensão de código*.
- Código antigo, em geral, não possui uma estrutura (projeto) adequado.

Análise de Custo de Manutenção

- Código novo, por sua vez, possui estrutura *implícita*, de difícil compreensão (comentário pessoal do autor):
 - Mecanismos de OO como *polimorfismo*, herança, construtores e destrutores definem um estrutura que não está tão evidente no código fonte.
 - Novos sistemas possuem arquiteturas mais complexas, envolvendo múltiplas camadas, estruturas distribuídas (e.g., CORBA) etc.

Custo da Manutenção

- Dados dos anos 70 indicam que:
 - 20% do custo é devido a manutenções corretivas;
 - 25% do custo é par a manutenções adaptativas;
 - 55% do custo é para manutenções aperfeiçoadoras.
- Dados mais recentes (anos 80 e 90) indicam que a manutenção corretiva é responsável por 16 a 22% do custo de manutenção.
- Ou seja, não mudou muito!

Custo da Manutenção

- Observação:
 - Em geral, os desenvolvedores têm feito um bom trabalho, pois pouco defeitos são descobertos depois de liberado o software.
 - O problema então são os clientes?
- Problema:
 - O tempo passa; as pessoas mudam; requisitos mudam; software tem que mudar; mudanças implicam *manutenção*.

Custo da Manutenção

- Problema:
 - As manutenções *adaptativa* e *aperfeiçoadora* são mais difíceis de implementar do que as corretivas.
 - Envolvem, muitas vezes, alterações no projeto e mais codificação, ou seja, um *novo* ciclo de desenvolvimento.
 - Portanto, fica claro para onde vai o dinheiro gasto em manutenção: *melhorias* no software.
- Solução:
 - Projetar o software tendo em vista que ele irá mudar!
 - Planejar as atividades de manutenção!

Resumo

- Custo da manutenção tem aumentado com o decorrer dos anos.
- Maior parcela do custo é gasto em compreensão do software a ser modificado -- atividade realizada por seres humanos. Gente custa dinheiro.
 - Novos sistemas, mais complexos, são mais difíceis de entender.
- Maior parte dos custos de manutenção é devido a manutenções aperfeiçoadoras e adaptativas.
- Solução: planejar o software tendo em vista que ele vai mudar; planejar atividade de manutenção.

Exercícios

1. Qual é a sua estimativa do custo da manutenção de software na sua organização?
2. Qual é a maior causa de solicitações de manutenção na sua organização? Correções ou melhorias?
3. Que tipo de apoio (ferramenta, documentação) é fornecido na sua organização para auxiliar a compreensão do software a ser modificado? Ou que tipo de apoio está faltando?