



**Garantia de Processo**

**Leis de Lehman**

**Manutenção de Softwares**



## Garantia de Processo

- Acidentes são eventos raros em sistemas críticos e pode ser impossível simulá-los durante testes de um sistema.
- Requisitos de segurança são requisitos do tipo “Não deve” que exclui o comportamento inseguro do sistema. É praticamente impossível demonstrar a maneira conclusiva que esses requisitos foram cumpridos.
- A análise de perigo (Processo contínuo durante o desenvolvimento do sistema) se ocupa em demonstrar que os perigos foram considerados em todos os estágios do processo.

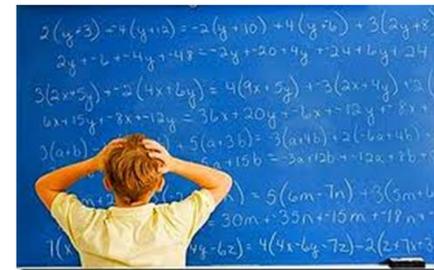


## **Avaliação de proteção:**

- Requisitos de proteção são semelhantes a requisitos de segurança (“Não deve”). Eles especificam o comportamento dos sistemas que não é permitido ao invés do que é esperado.
- A diferença fundamental entre segurança e proteção é que problemas de segurança são geralmente acidentais, enquanto ataques são propositais.

## Abordagens para verificação de segurança:

- Validação baseada na experiência.
- Validação baseada em ferramentas.
- Equipe de testes (Equipes Hacker).
- Verificação Formal.



Nota: É muito difícil para os usuários finais de um sistema verificar sua proteção.

# MANUTENÇÃO DE SISTEMAS





## MANUTENÇÃO DE SISTEMAS

- Depois dos sistemas serem implantados, eles devem inevitavelmente mudar para permanecerem úteis
- A evolução de software é importante porque as organizações são dependentes dos sistemas que custaram milhões (\$\$\$)



## Leis de LEHMAN

- Leis propostas por Lehman referentes a mudanças nos sistemas consideradas pela engenharia de software como invariáveis e amplamente aplicáveis.
- O crescimento e a evolução de vários sistemas de grande porte foram examinados
- Objetivo: Definir uma teoria unificada para evolução de software
- Resultados: Um conjunto de oito leis que “governam” a evolução de sistemas.

Fonte: M. M. Lehman. Rules and Tools for Software Evolution Planning and Management. Annals of Software Engineering, 2001.



# Leis de LEHMAN

## ***Primeira Lei - Mudança Contínua***

- A manutenção de um sistema é um processo inevitável.
- O ambiente muda, novos requisitos surgem e o sistema deve ser modificado
- Após o sistema modificado ser reimplantado, ele muda o ambiente

*“Um sistema em um ambiente do mundo real necessariamente tem de ser modificado ou se tornará uma maneira progressiva menos útil para esse ambiente”.*



# Leis de LEHMAN

## *Segunda Lei - Complexidade Crescente*

- *A medida que o sistema é modificado sua estrutura é degradada.*
- *A medida que um sistema evolui, sua complexidade aumenta, a menos que seja realizado esforço para mantê-la ou diminuí-la.*
- *Manutenções preventivas são necessárias e tem custos adicionais, além daqueles para implementar as mudanças necessárias.*

*“A 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”.*



# Leis de LEHMAN

## *Terceira Lei – Auto Regulação*

- Sistemas de grande porte têm uma dinâmica própria que é estabelecida no estágio inicial do processo de desenvolvimento.
- A evolução de software é um processo auto-regulável. Atributos do sistema como tamanho, tempo entre versões e número de erros reportados é quase invariável em cada versão do sistema.
- Consequência de fatores estruturais e organizacionais

*“A evolução do programa grande é um processo auto-regulado. Os atributos do sistema, como tamanho, tempo entre releases e número de erros relatados são aproximadamente invariáveis para cada release do sistema”.*

# Leis de LEHMAN

## **Quarta Lei – Estabilidade Organizacional**

- A maioria dos projetos de programação trabalha no “Estado saturado”, isto é, uma mudança nos recursos ou no pessoal tem efeitos imperceptíveis na evolução de longo prazo do sistema.
- Durante o ciclo de vida de um programa, sua taxa de desenvolvimento é quase constante
  - Indepe de recursos dedicados ao desenvolvimento do sistema
  - Alocação de grandes equipes é improdutivo, pois o *overhead* de comunicação predomina

“Durante o tempo de duração de um programa, sua taxa de desenvolvimento é aproximadamente constante e independente dos recursos dedicados ao desenvolvimento do sistema”.

# Leis de LEHMAN

## ◦ **Quinta Lei - Conservação de Familiaridade**

- Acrescentar nova funcionalidade em um sistema inevitavelmente introduz novos defeitos. Quanto mais funcionalidades forem acrescentadas em cada release maior o número de defeitos.
- Durante o ciclo de vida de um sistema, mudanças incrementais em cada versão são quase constantes.
  - Um grande incremento em uma release leva a muitos defeitos novos. A release posterior será dedicada quase exclusivamente para corrigir os defeitos
  - Ao orçar grandes incrementos, deve-se considerar as correções de defeitos

“Durante o tempo de duração de um sistema, as mudanças incrementais em cada release são aproximadamente constantes”.

# Leis de LEHMAN

## **Sexta Lei – Crescimento Contínuo**

- O conteúdo funcional de sistemas devem ser continuamente aumentado para manter a satisfação do usuário.



# Leis de LEHMAN

## ***Sétima Lei* – Qualidade Declinante**

- A qualidade de sistemas parecerá estar declinando a menos que eles sejam mantidos e adaptados às modificações do ambiente

# Leis de LEHMAN

## **Oitava Lei – Sistema de Feedback**

- Os processos de evolução incorporam sistemas de *feedback* com vários agentes e *loops*
- Estes sistemas devem ser considerados para conseguir aprimoramentos significativos de produto



# MANUTENÇÃO DE SISTEMAS

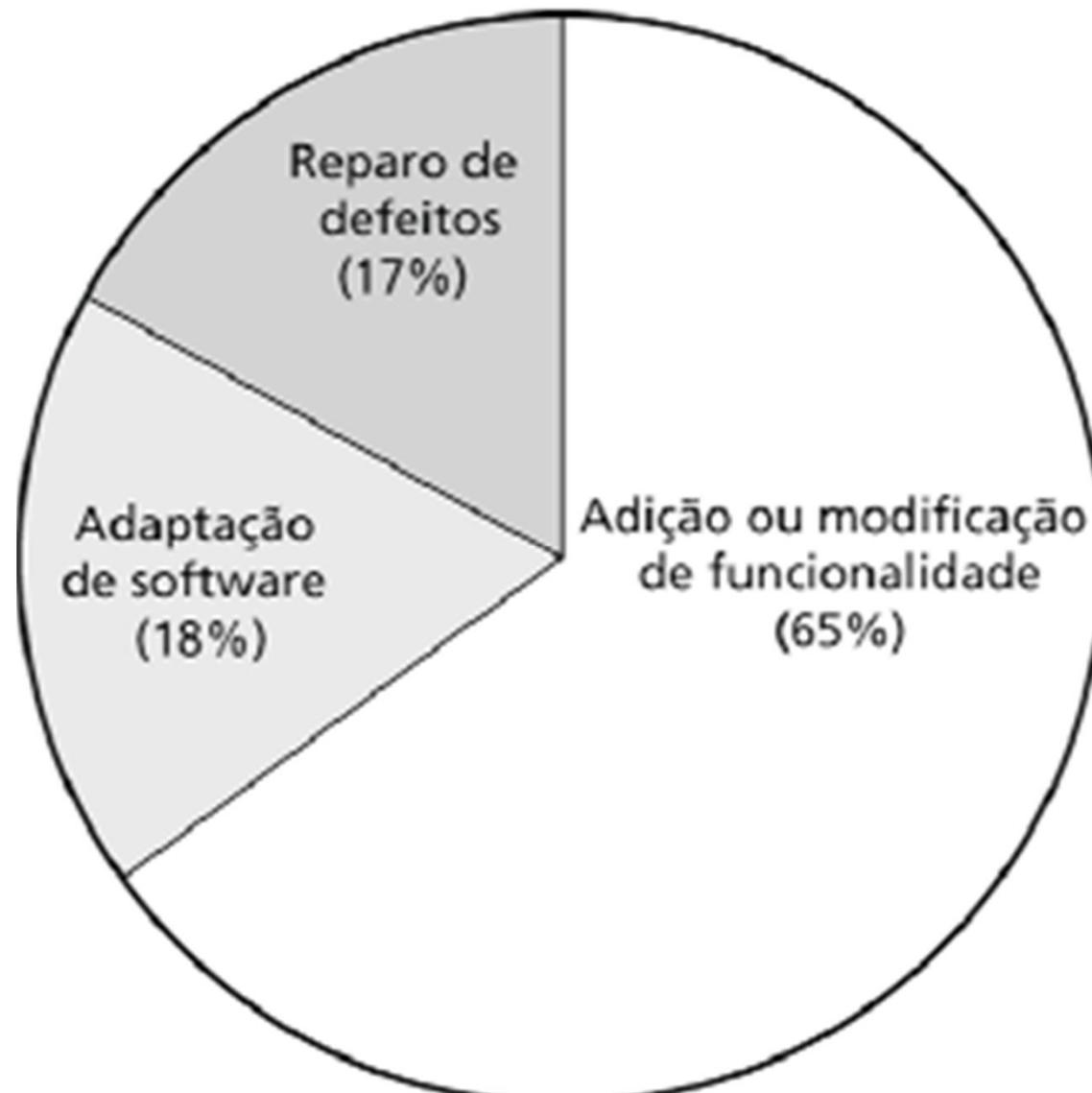
Software precisa de mudanças para corrigir erros, melhorar seu desempenho ou outras características não funcionais ou incluir, alterar e excluir requisitos funcionais.



## MANUTENÇÃO DE SISTEMAS

- **Manutenção corretiva:**
  - Mudanças para reparo de defeitos de software
- **Manutenção adaptativa:**
  - Mudanças para adaptar o software a outro ambiente
- **Manutenção evolutiva:**
  - Mudanças para adicionar funcionalidade ao sistema

# MANUTENÇÃO DE SISTEMAS



# MANUTENÇÃO DE SISTEMAS

## Estratégias para mudanças de software:

- Manutenção de software: Mudanças realizadas em respostas a requisitos (Mantendo estável a estrutura fundamental).
- Transformação de Arquitetura: Envolve transformações de arquitetura de software.  
Ex. Sistema centralizado transformado em um sistema com arquitetura cliente-servidor.
- Reengenharia de software: Nenhuma nova funcionalidade é adicionada ao sistema. O sistema é alterado afim de tornar mais fácil sua compreensão e alteração, mas sem envolver grandes mudanças em sua arquitetura.

**Nota:** As estratégias não são mutuamente exclusivas.

# MANUTENÇÃO DE SISTEMAS

## **Manutenção de Software:**

É o processo geral de modificação de um sistema após ser colocado em uso.

### **Tipos de manutenção:**

- Manutenção para reparar os defeitos de software.
- Manutenção para adaptar o software a um ambiente operacional diferente.
- Manutenção para fazer acréscimo à funcionalidade do sistema ou modificá-los.



## Custeio do Ciclo de Vida do Produto

- Analisar o custo de todo ciclo de vida do produto e não apenas o custo do projeto.

## MANUTENÇÃO DE SISTEMAS

É geralmente benéfico investir esforço no desenvolvimento para reduzir custos de manutenção

### Exemplos de investimento

- Especificação precisa do software
- Uso de orientação a objetos
- Gerência de configuração, etc.

## MANUTENÇÃO DE SISTEMAS

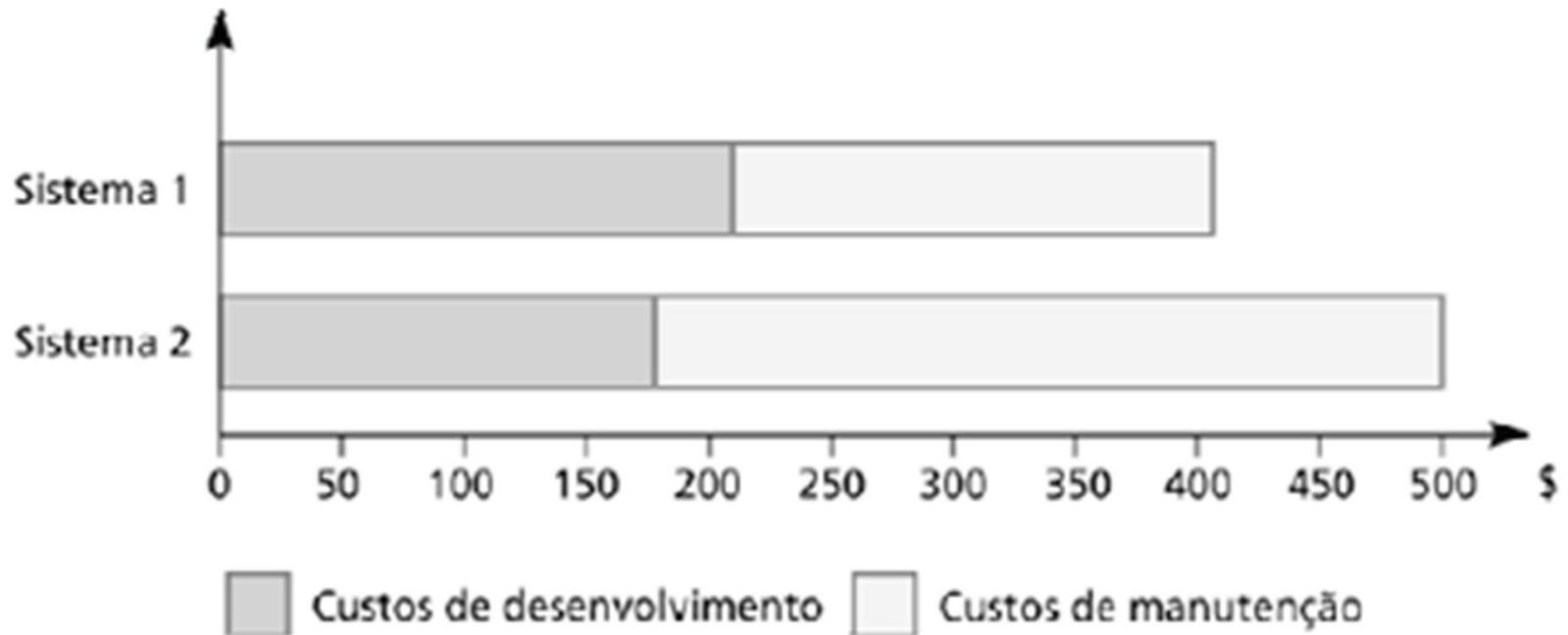
É geralmente benéfico investir esforço no desenvolvimento para reduzir custos de manutenção

### Exemplos de investimento

- Especificação precisa do software
- Uso de orientação a objetos
- Gerência de configuração, etc.

## MANUTENÇÃO DE SISTEMAS

- Sistema 1 têm investimento maior na fase de desenvolvimento



## MANUTENÇÃO DE SISTEMAS

Os custos de manutenção variam de acordo com a natureza da aplicação, mas como geralmente são bem superiores ao custo de desenvolvimento. Os principais fatores que elevam os custos de manutenção são:

- Ao término do desenvolvimento do projeto geralmente a equipe de desenvolvimento é desfeita e no caso de manutenção dificilmente será a mesma equipe que desenvolveu.
- Geralmente o processo de manutenção é tratado como mais simples pela alta direção designando profissionais menos experientes. Isto é um erro!!!
- Linguagem ultrapassada, técnicas obsoletas, documentação incompleta ou inexistente e código com alta complexidade (devido a manutenções anteriores).
- Contratos de desenvolvimento, geralmente não preveem manutenção, o que é mais um problema no processo.



# Sistemas Legados

# Razão

- As empresas gastam muito dinheiro em sistemas, para se obter um maior retorno o software deve ser utilizado por vários anos;
- Muitas empresas dependem de sistemas de mais de 20 anos de existência;
- A interrupção de um sistema dessas características poderia causar prejuízos se deixasse de funcionar por algum período.



# Sistemas Legados

- São sistemas antigos que possuem uma função crítica em relação ao processo funcional de uma organização que sofreram diversas modificações de requisitos durante seu tempo de vida;
- Também operam em um hardware antigo e específico, com diversas limitações.

# Sistemas Legados

- Os sistemas legados respondem pela maior parte do processamento de dados mundial, ou, dependendo de entendimentos, por todo. Isto porque, para SEACORD (2003) qualquer sistema em produção pode ser considerado como legado. Entre 60% e 70% dos sistemas estão em COBOL, estima-se em 200 milhões de linhas (SEACORD et al, 2003 e ULRICH, 2002).
- A participação do custo de manutenção no custo total de um sistema tem crescido de 40%, nos anos 70, até 90%, atualmente (PIGOSKI, 1996). Destes custos, cerca de 20% são consumidos com correções e 80% com melhorias diversas (Martin apud ULRICH, 2002).



# Riscos de Substituição de Sistemas Legados

- Raramente existe uma especificação completa do sistema legado. E se existir, é muito improvável que ela abranja todas as modificações feitas no sistema.
- Os processos corporativos e o modo como os sistemas legados operam estão sempre intrinsecamente ligados. Esses processos foram projetados para tirar vantagens dos serviços de software e para evitar seus pontos negativos.



# Riscos de Substituição de Sistemas Legados

- Importantes regras corporativas podem estar inseridas no software e pode não estar documentadas em nenhum outro lugar. Simplesmente substituir o software pode ter consequências imprevisíveis;
- O simples desenvolvimento de um sistema para tomar essa função pode trazer risco em si para a organização (atrasos, custos altos...).



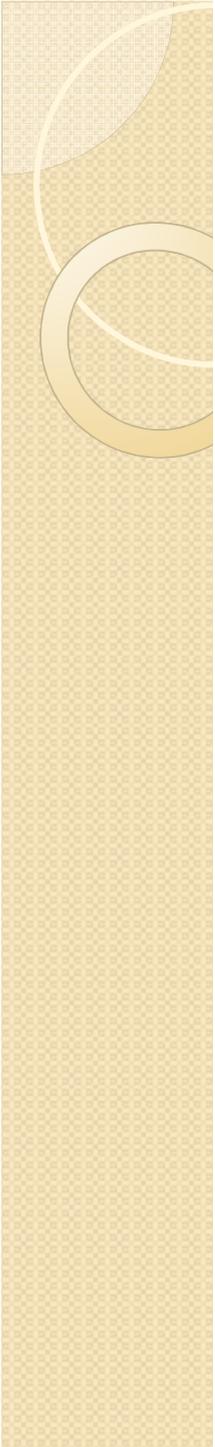
# Manutenção de Sistemas Legados

- Manter sistemas legados envolve um grande dispêndio de dinheiro e tempo porque:
  - Diferentes partes do sistema foram implementadas por diferentes equipes. Não há um estilo de programação consistente em todo sistema.
  - Uma parte ou todo sistema pode ser implementado em uma linguagem obsoleta. Pode ser difícil encontrar pessoal que tenha conhecimento dessa linguagem.



# Manutenção de Sistemas Legados

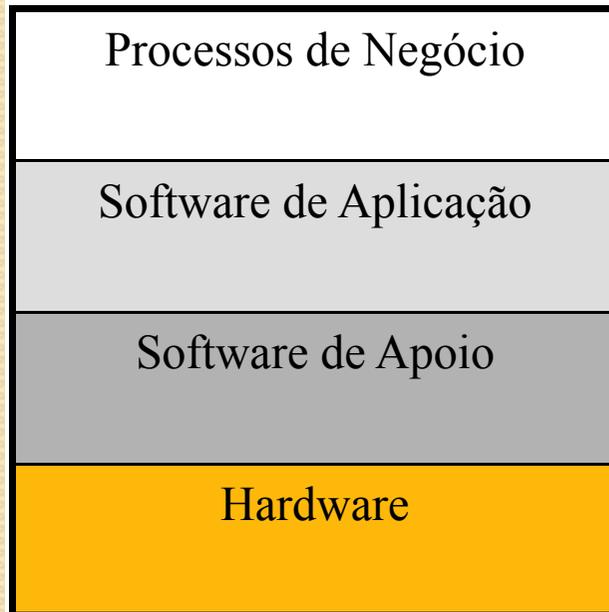
- Frequentemente documentação é inadequada e desatualizada. Em alguns casos, a única documentação é o código-fonte do sistema. Outras, só existe a versão executável do sistema;
- Os muitos anos de manutenção podem ter corrompido a estrutura do sistema, tornando-o cada vez mais difícil de ser compreendido.



# Manutenção de Sistemas Legados

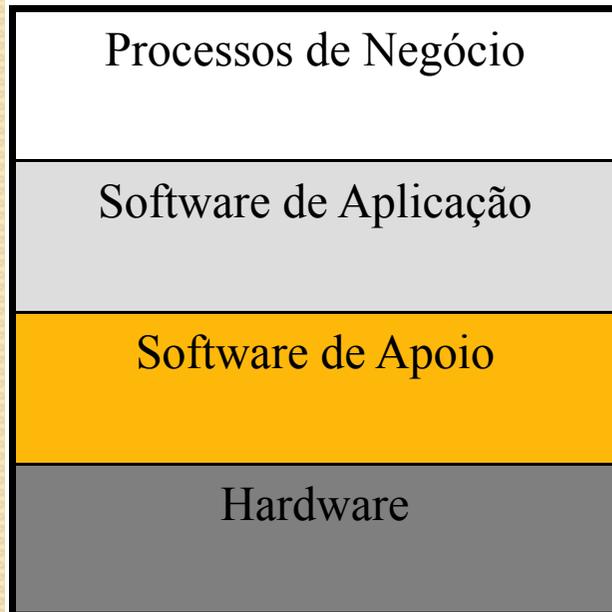
- O sistema pode ter sido otimizado para melhorar a utilização de espaço ou velocidade de execução, em vez de ter sido escrito para facilitar a compreensão;
- Os dados processados pelo sistema podem estar armazenados em diferentes arquivos, que podem ter estruturas incompatíveis. É possível que alguns dados tenha sido duplicados, ou estejam desatualizados, e inexatos ou incompletos.

# Estruturas dos Sistemas Legados



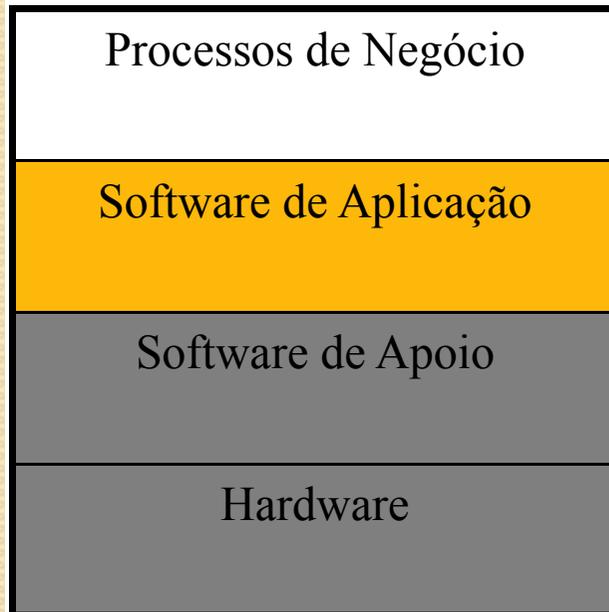
**Hardware de Sistema:** em muitos casos, os sistemas legados foram escritos para hardware de mainframe, que não está mais disponível, tem uma manutenção dispendiosa e pode não ser compatível com as atuais políticas de compras na área de TI.

# Estruturas dos Sistemas Legados



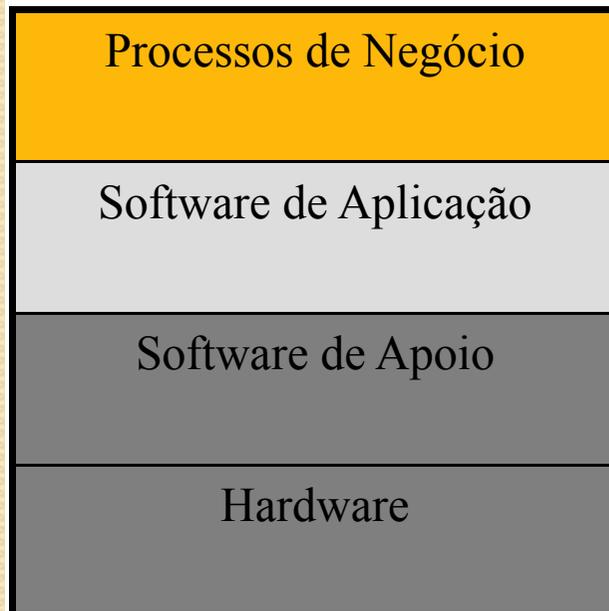
Software de apoio: O Sistema Legado pode depender de diferentes produtos de software de apoio, desde o sistema operacional e utilitários oferecidos pelo fabricante de hardware até os compiladores utilizados para o desenvolvimento do sistema. Novamente, esses produtos de software podem estar obsoletos e não ter mais a assistência técnica de seus fornecedores originais.

# Estruturas dos Sistemas Legados



**Software de Aplicação:** fornece os serviços ao negócio. É composto por vários programas separados, desenvolvidos em épocas diferentes

# Estruturas dos Sistemas Legados



Processos de negócios: são os processos utilizados nas empresas a fim de atingir algum objetivo de negócio. Envolve regras e políticas de negócio, que especificam como as transações devem ser feitas.

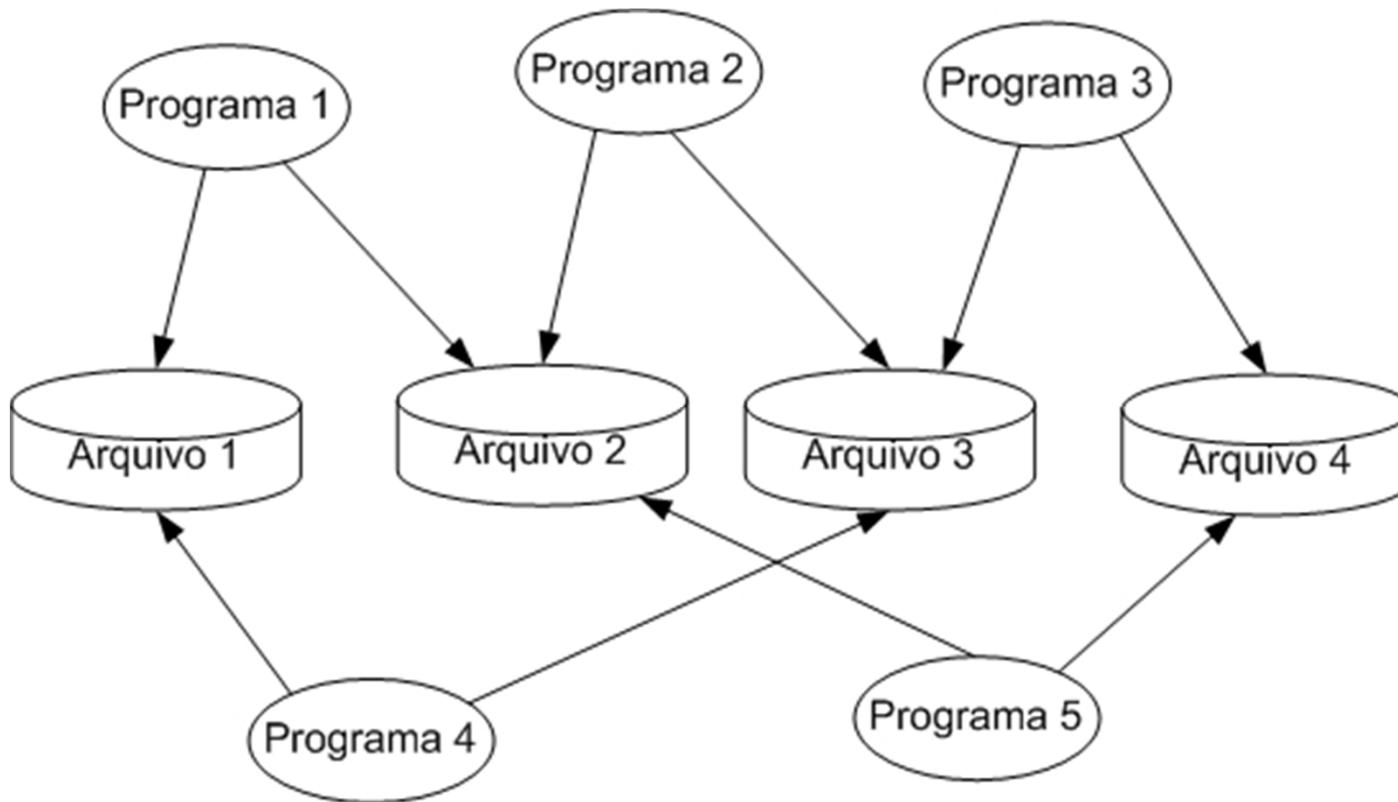


# Estruturas dos Sistemas Legados

- A modificação de uma camada do sistema pode introduzir novos recursos, e as camadas superiores do sistema podem ser modificadas para se beneficiarem desse recurso;
- A modificação do software no sistema pode torná-lo mais lento, de modo que um novo hardware é necessário;
- Muitas vezes é impossível manter interfaces de hardware, especialmente se for proposta uma mudança radical para um novo tipo de hardware.

# Estrutura dos Sistemas Legados

- Exemplo:



# Avaliar um Sistema Legado

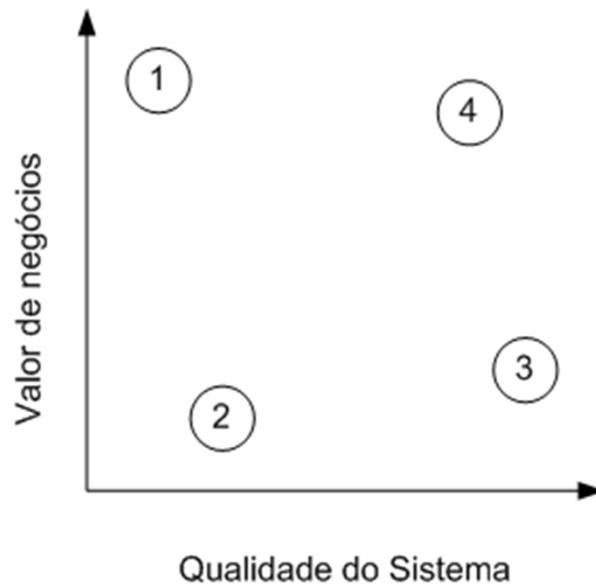
- Descartar completamente o sistema: essa opção deve ser escolhida quando o sistema não tiver prestando uma contribuição efetiva aos processos de negócio. Isso ocorre quando os processos corporativos foram modificados, desde que o sistema foi instalado, e não mais dependem do sistema;
- Substituir um sistema por um novo: pode ser consequência de uma mudança de hardware que inviabiliza o sistema antigo, mas ainda mantendo utilidade.



# Avaliar um Sistema Legado

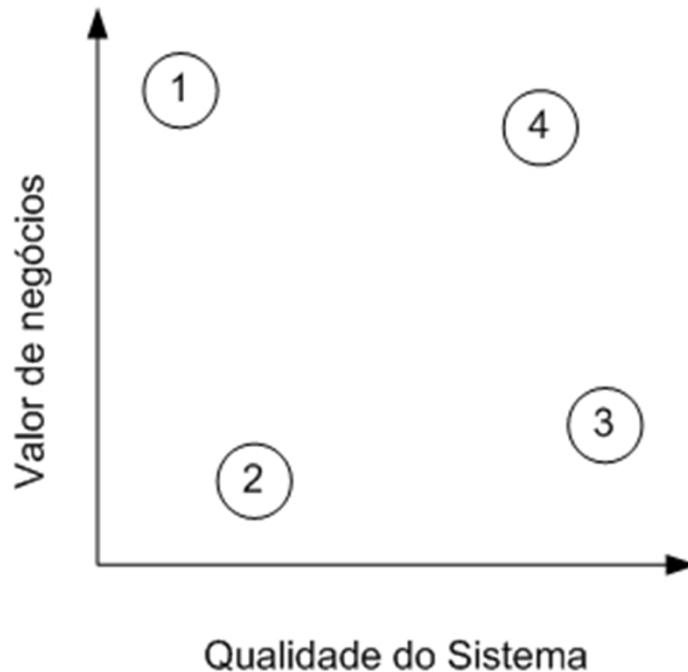
- Continuar a manter o sistema: esta opção deve ser escolhida quando o sistema ainda é necessário, ele for bastante estável e os usuários não pedirem grandes modificações no sistema;
- Transformar o sistema para melhorar sua facilidade de manutenção: acontece quando um sistema sofre várias modificações que são realmente necessárias.

# Avaliar um Sistema Legado



**I - Baixa qualidade, alto valor de negócio:** esses sistemas estão prestando uma importante contribuição à empresa e não podem ser descartados. No entanto, sua baixa qualidade o transforma num candidato a transformação ou substituição, se aplicável.

# Avaliar um Sistema Legado



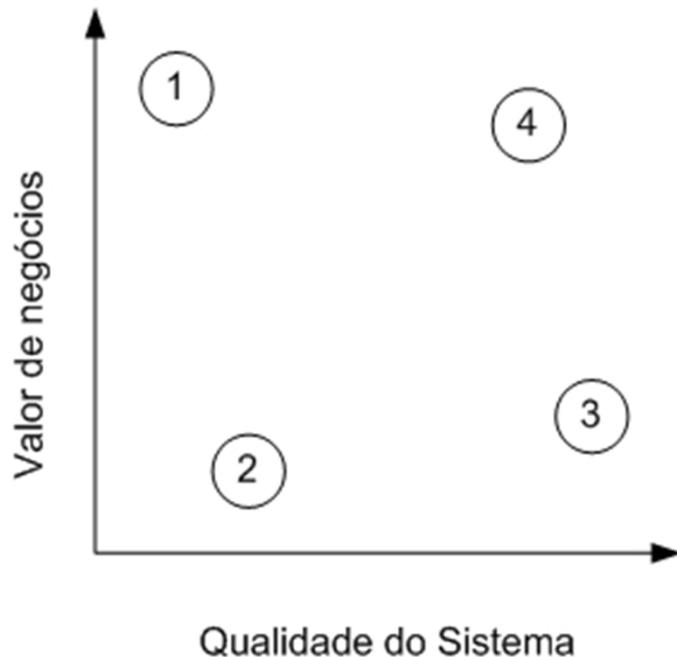
**2 - Baixa qualidade, baixo valor de negócio:** manter esse sistema em operação será dispendioso e a taxa de retorno de investimento será bastante pequena. São candidatos a serem descartados.

# Avaliar um Sistema Legado



**3 - Alta qualidade, baixo valor de negócio:** não contribuem muito para os negócios, mas cuja manutenção pode não ser muito dispendiosa. Não vale o risco substituir esses sistemas, que deve ser mantido ou descartado.

# Avaliar um Sistema Legado



**4 - Alta qualidade, Alto valor de negócio:** esses sistemas devem ser mantidos em operação e sua alta qualidade significa que não é necessário investir na sua transformação ou substituição. Deve-se continuar a manutenção normal do sistema.



# Avaliar um Sistema Legado

- Avaliação do Valor de negócio:
  - Usuários finais do sistema: Qual é a eficácia dos sistemas que apoia seus processos de negócios? Quanto da funcionalidade do sistema é utilizada?
  - Clientes: O uso do sistema é transparente para os clientes ou suas interações são limitadas pelo sistema? Eles têm que esperar por causa do sistema? Os erros do sistema têm impacto direto sobre os clientes?



# Avaliar um Sistema Legado

- Avaliação do Valor de negócio:
  - Gerentes de linha: os gerentes pensam que o sistema é eficaz para o sucesso de uma unidade? Os custos de manutenção do sistema são justificáveis? Os dados manipulados pelo sistema são importantes para o funcionamento da unidade do gerente?
  - Gerentes de TI: existem dificuldades em encontrar pessoas para trabalhar no sistema? O sistema consome recursos que poderiam ser usados mais efetivamente em outros sistemas?



# Avaliar um Sistema Legado

- Avaliação do Valor de negócio:
  - Gerentes sênior: o sistema e o processo de negócio associado prestam uma contribuição eficaz aos objetivos da empresa?

# Notas de Aula retiradas do Livro

Ian Sommerville. Engenharia de Software, 8ª Edição. Pearson Education, 2007.

Capítulo 21 Evolução de Software

Sistemas Legados – disponível em  
:fafica.codethe.net/**Sistemas\_Legados**.ppt, acessado em 08 de Novembro  
de 2011