

# Requisitos

Prof. Marcelo Werneck

Disciplina: Engenharia de Requisitos

Curso: Engenharia de Software – Praça da  
Liberdade – PUC Minas

# Tópicos da Aula

- Introdução a Requisitos
  - Definição
  - Importância
  - Tipos de Requisitos
    - Funcionais x Não-funcionais
    - Cliente x Sistema

# Introdução



# O que são requisitos?

- Uma condição ou capacidade necessitada por um usuário para resolver um problema ou atingir um objetivo
- Uma condição ou capacidade que deve ser cumprida ou possuída por um sistema ou componente do sistema para satisfazer um contrato, padrão, especificação ou outro documento formal imposto
- A representação documental de condição ou capacidade, como em 1 e 2

# O que são requisitos?

- Características que definem os critérios de aceitação de um produto  
Wilson de Pádua
- Propriedade que um software deve exibir para resolver um problema do mundo real  
SWEBOK
- Condição ou capacidade que o sistema deve contemplar, que pode ser derivada diretamente das necessidades do usuário e/ou definida em contrato, padrão, especificação ou outro documento formal imposto  
Rational Unified Process
- Característica do sistema ou a descrição de algo que o sistema é capaz de realizar para atingir os seus objetivos  
Shari Pfleeger

# O que são requisitos?

- Uma declaração que identifica um produto ou processo operacional, funcional, ou restrição ou característica de desenho, que é não ambígua, testável, mensurável e necessária para a aceitação do produto ou processo (por consumidores ou diretrizes internas de garantia da qualidade).

# Aspectos Relevantes

- Requisitos envolvem questões na fronteira do domínio
- Problema existe no mundo real, não no software
  - Requisitos descrevem os efeitos do sistema no ambiente, não o sistema em si!
- Distinguir entre domínio do problema e domínio da solução.
- Requisitos iniciais não devem descrever mais do que o necessário para se definir o problema.
- A princípio, evitar referência a soluções particulares.
- Engenheiros de Requisitos devem definir a melhor solução sem ideias pré-concebidas.

# Aspectos Relevantes

- Sem uma distinção clara entre problema e solução, o seguinte pode ocorrer:
  - Falta de entendimento do problema real
  - Falta de capacidade de se definir o escopo do sistema e compreender as funções que devem ser incluídas
  - Discussões centradas no sistema (descritas em termos das soluções)
  - Dificuldade de se encontrar soluções ótimas devido à falta de liberdade no projeto.



# Aspectos Relevantes

- Domínio
  - Tudo aquilo ligado ao ambiente
  - Onde uma solução de software (ou qualquer outro tipo) executa

# Aspectos Relevantes

- Domínio



# Aspectos Relevantes

- Conhecimento do software
  - Todo conhecimento necessário para o desenvolvimento da solução



# Por que requisitos são importantes?

“A parte individual mais difícil de se fazer na construção de um sistema de software é decidir precisamente o que construir. Nenhuma outra parte do trabalho conceitual é tão difícil quanto estabelecer detalhadamente os requisitos técnicos, incluindo todas as interfaces com pessoas, máquinas e outros sistemas. Nenhuma outra parte do trabalho prejudica tanto o sistema final se feita incorretamente. Nenhuma outra parte do trabalho é mais difícil de se reparar a posteriori”



Fred Brooks: No Silver Bullet:  
Essence and Accidents of Software Engineering

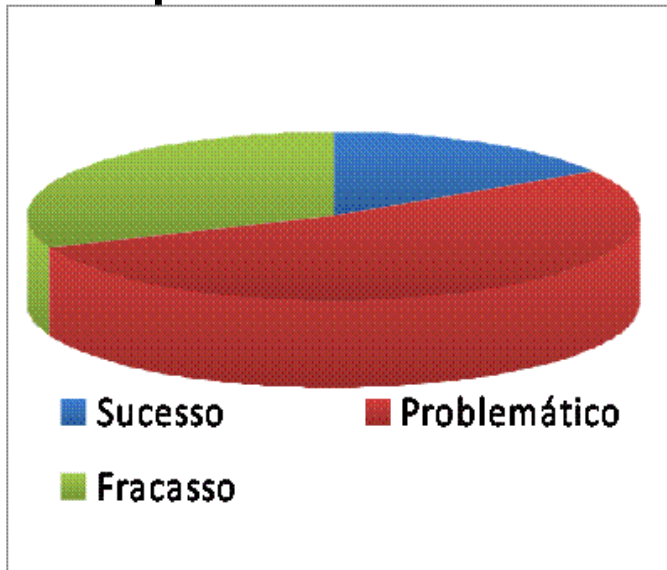
# Por que requisitos são importantes?

O que acontece quando os requisitos não estão ok?

- O sistema pode ser entregue com atraso e com custos além do orçamento
- Os clientes e usuários finais podem não ficar satisfeitos com o sistema
  - Abandono do sistema
  - Reconstruir o sistema
- O sistema pode ficar inviável de ser utilizado devido aos defeitos apresentados
- Se o sistema continuar em uso, o custo de manter e evoluir o sistema pode ser muito significativo

# Por que requisitos são importantes?

- Estudo realizado pelo *Standish Group*, considerando 350 companhias e 8.000 projetos de software, em 1995.
- Estudos mais recentes mostram melhora mas problemas continuam.



Fator Crítico de Sucesso	
1. Requisitos Incompletos	13,10%
2. Falta de Envolvimento do Usuário	12,40%
3. Falta de Recursos	10,60%
4. Expectativas Irreais	9,90%
5. Falta de Apoio Executivo	9,30%
6. Mudanças de Requisitos e Especificações	8,70%
7. Falta de Planejamento	8,10%
8. Sistema não mais necessário	7,50%

# Atualização do Relatório do Caos

## Project Success is Rare



Source: *Extreme Chaos, The Standish Group International, Inc., 2004, 2006, 2009*



Average cost overrun:

45%

Time overrun:

63%

Functionality delivered on average:

67%

Standish Group

# Por que requisitos são importantes?

- Dois aspectos básicos
  - Custos



- Defeitos





# Por que requisitos são importantes?

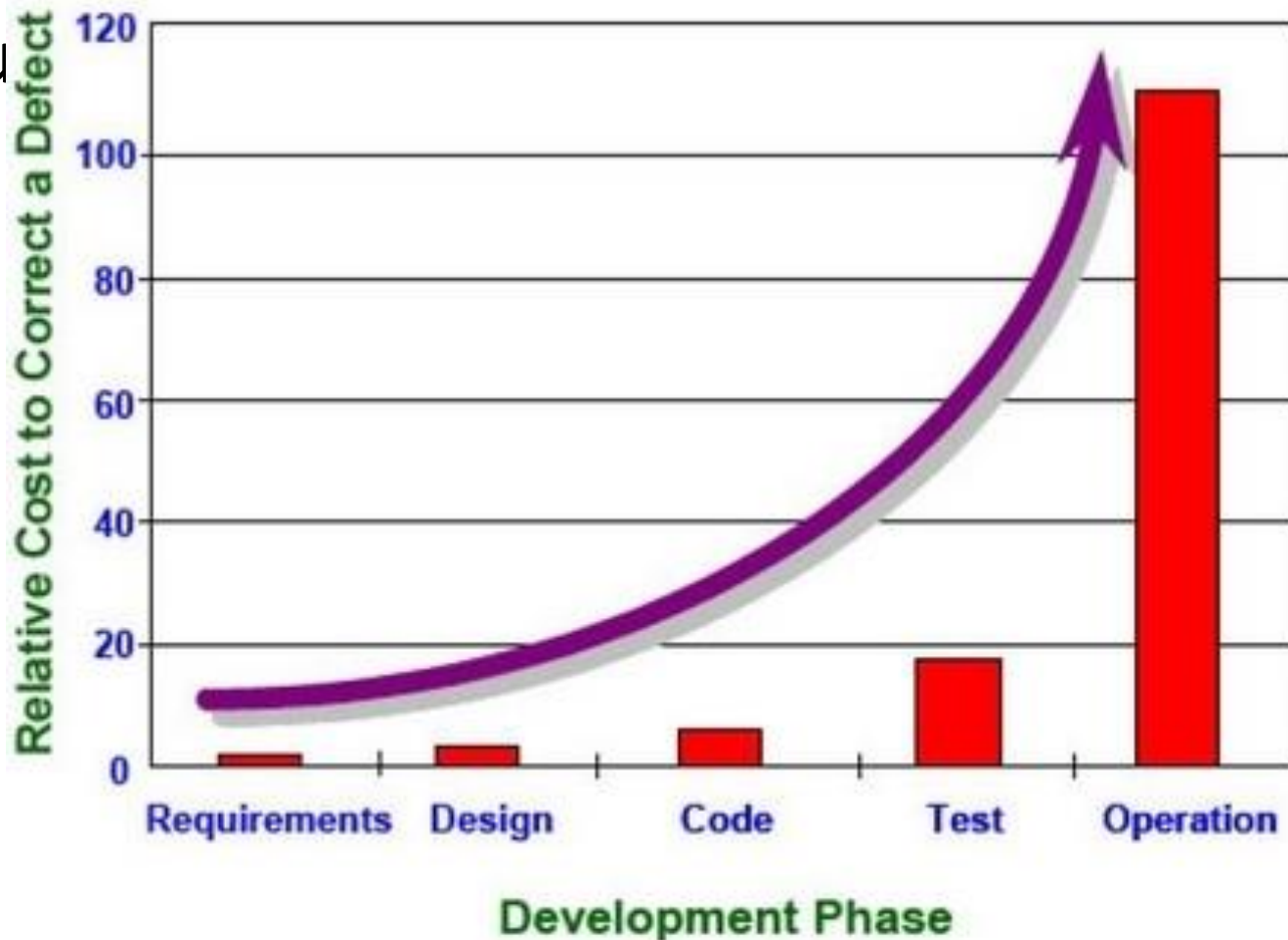
- Custos
  - Para cada \$1 gasto em resolução de defeitos em requisitos, quanto gastaríamos se só encontrássemos esse erro mais tarde?
    - \$3 a \$6 em desenho
    - \$10 em codificação
    - \$15 a \$40 em testes em desenvolvimento
    - \$30 a \$70 em testes de aceitação
    - \$40 a \$1000 em produção (IBM \$82)



(Fonte: IMB, GTE et al)

# Por que requisitos são importantes?

- Cu



# Por que requisitos são importantes?

- Defeitos
  - Erros em requisitos acarretam os seguintes problemas
    - Atraso nas entregas
    - Baixa qualidade de produtos
    - Baixa qualidade de vida dos desenvolvedores
    - Prejuízos de produção
    - Riscos de morte



# Por que requisitos são importantes?

- Provêm a base para o planejamento do projeto.
- Essenciais para estudo de solicitações de mudança.
- Permitem a gerência de riscos desde os estágios iniciais de desenvolvimento.
- São a base para os testes de aceitação.
- Gerenciamento de contratos.



# Conceitos de requisitos

- Classificações de requisitos
  - Funcionais x Não Funcionais
  - Cliente x Sistema
  - Outras classificações



# Funcionais x Não-Funcionais

- Requisitos funcionais
  - Especificam ações que um sistema deve executar, sem levar em consideração restrições físicas
  - Exemplos:
    - Será necessário realizar o fechamento contábil mensalmente
    - O Gerente deverá visualizar todos os empréstimos efetuados no mês, indicando do funcionário que disponibilizou o empréstimo, o cliente que obteve o empréstimo e o valor emprestado

# Funcionais x Não-Funcionais

- Requisitos não-funcionais
  - Descrevem restrições desejadas ou necessárias, atributos do sistema ou de seu ambiente
    - Condições que o sistema deve atender
    - Qualidades específicas que o sistema deve ter
    - Especificidades que o ambiente deve cumprir
  - São também chamados de restrições ou requisitos de qualidade
  - Determinam a arquitetura do sistema

# Funcionais x Não-Funcionais

- Requisitos não-funcionais
  - Exemplos
    - O fechamento contábil do mês deverá ser realizado em no máximo 4h para um volume de até 40 milhões de registros.
    - A base de dados do sistema deverá ser armazenada em Oracle 10g
    - O sistema deverá suportar dois idiomas: português e espanhol



# Tipos de Req. Não-Funcionais

- **Usabilidade:** estética, consistência da interface do usuário, ajuda on-line sensível ao contexto, wizards, documentação de usuário e material de treinamento
- **Confiabilidade:** frequência e severidade das falhas, possibilidade de recuperação, tempo médio entre falhas, previsibilidade, acurácia
- **Desempenho:** impõe condições aos requisitos funcionais: velocidade, eficiência, throughput, tempo de resposta, tempo de recuperação, faixa de recurso utilizado.
- **Suporte:** testabilidade, extensibilidade, adaptabilidade, manutenibilidade, compatibilidade, configurabilidade, instalabilidade, internacionalização

# Tipos de Req. Não-Funcionais

- **Desenho:** define restrições ao desenho do sistema, como uso de padrões de desenho
- **Implementação:** que define restrições à implementação do sistema, como linguagens de desenvolvimento, padrões de codificação, políticas de uso do Banco de Dados, limites de recursos e ambiente de desenvolvimento
- **Interface:** que especifica itens externos com os quais o sistema precisa interagir e restrições em formatos, sincronismo, protocolos ou outros fatores utilizados nessa interação
- **Físico:** que delimita material, forma, tamanho e peso que o sistema deve cumprir. É o tipo de requisito relacionado a hardware, ambientes de rede, sistemas embutidos, etc.

# Cliente x Sistema

- O detalhamento dos requisitos evolui durante um projeto
  - Origem no cliente e nos usuários
  - Termina detalhado o suficiente para prosseguir o desenvolvimento do sistema

# Cliente x Sistema

- Exemplo

- Necessidades do cliente

- Construir um sistema para predição do movimento do mercado de ações da BOVESPA com taxa de acerto de 70%

- Funcionalidades a serem cumpridas pelo sistema

- Analisar séries históricas
    - Treinar o sistema com dados históricos
    - Gerar modelos matemáticos para conjunto de ações
    - Compilar portfólios de ações
    - (...)

# Cliente x Sistema

- Requisitos do cliente
  - Expressam os resultados desejados para superar os problemas reais.
  - Descrevem o problema enfrentado pelo cliente ou usuário, materializado a partir de suas necessidades, expectativas, restrições e interface apontadas
  - Descrevem as características desejáveis de uma solução
  - Delimitam o universo de soluções possíveis (domínio), ao invés de uma solução específica
  - Também chamados de **requisitos de usuário**

# Cliente x Sistema

- Requisitos do cliente (cont.)
  - Normalmente materializados em documentos iniciais, tais como:
    - Proposta técnica
    - Documento de visão
    - Lista de funcionalidades
  - Linguagem muito próxima ao usuário

# Cliente x Sistema

- Requisitos do sistema
  - Descrevem o comportamento de um sistema de computação apresentado como solução para o problema do cliente
    - Uma escolha dentro do domínio delimitado
  - Delimitam as interfaces de um sistema que soluciona o problema
  - Também chamados de **requisitos do produto**
  - Podem ser detalhados em
    - Requisitos do produto como um todo
    - Requisitos dos componentes do produto

# Cliente x Sistema

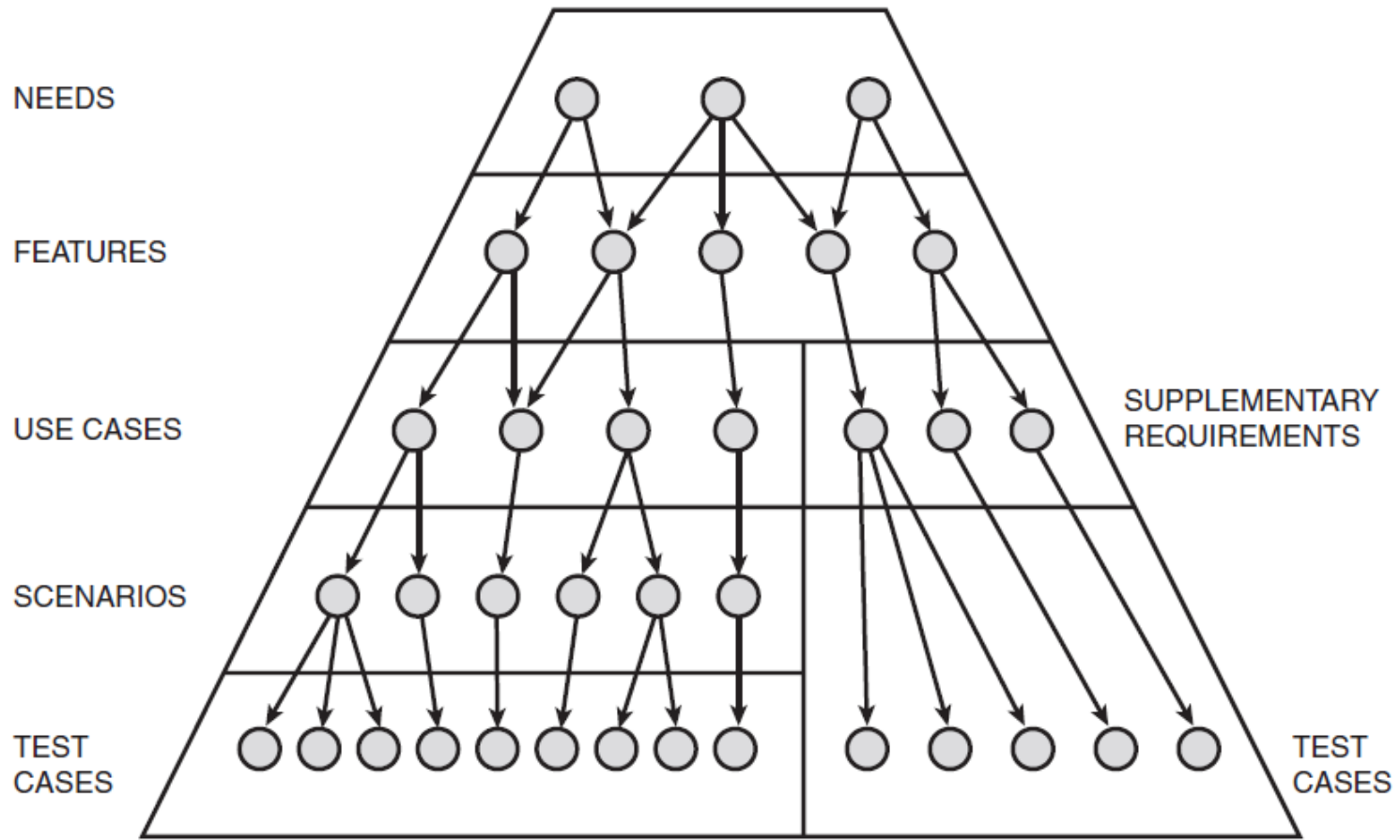
- Requisitos de sistema
  - Normalmente materializados em
    - Documentos detalhados de requisitos
    - Casos de uso (modelagem e detalhamento)
    - Máquinas de estado
    - Diagramas de seqüência
    - Protótipos
    - Mapas de navegação
    - (...)
  - Linguagem mais próxima dos analistas
    - Entretanto, continua a descrever **o que deve ser feito**, não **como deverá ser feito**



# Cliente x Sistema

- Componente de sistema
  - Parte do produto final ou algo usado no desenvolvimento que faz parte da entrega
    - Subproduto
    - Processo
    - Ferramenta
  - Exemplos
    - A especificação detalhada de requisitos deverá utilizar a notação de casos de uso
    - O módulo de visualização deverá utilizar os recursos do IE 7

# Evolução dos Requisitos



# Outras Classificações de Requisitos

- Técnico x Sistema
- Técnico x Não-Técnico
  - Diferentes formas de dizer a mesma coisa.
- Requisitos de domínio
  - Requisitos que vem do domínio da aplicação do sistema e refletem características ou restrições para aquele domínio