

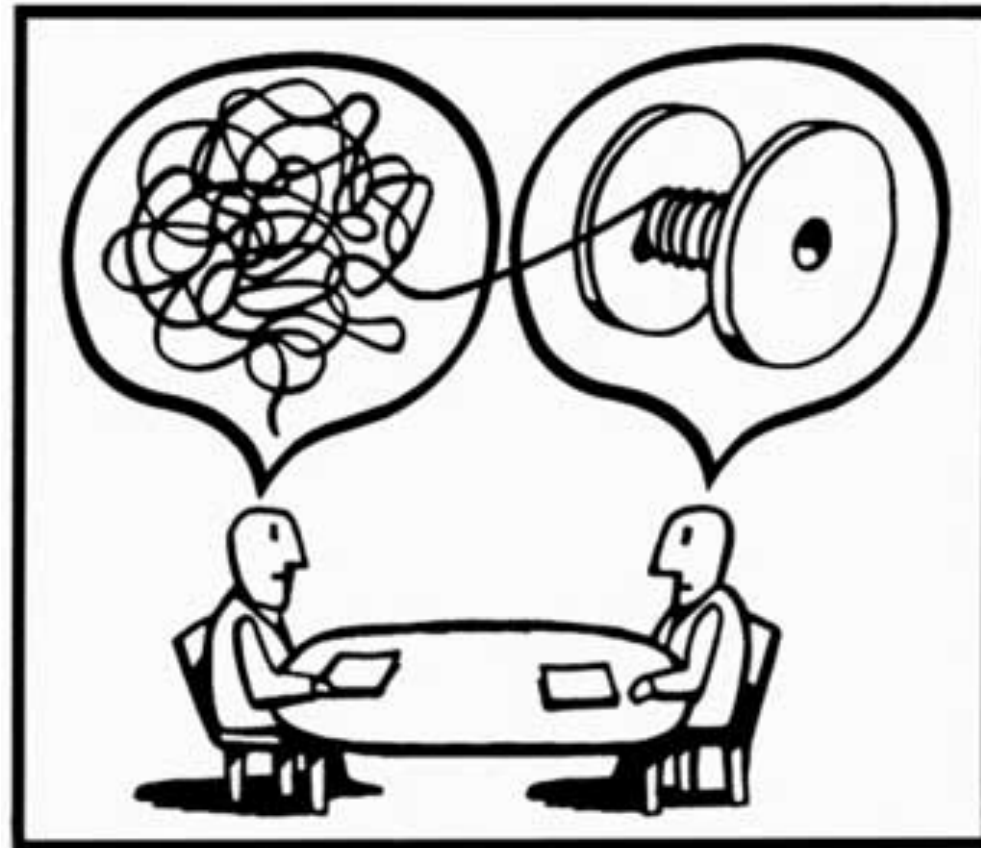
# Princípios de Engenharia de Requisitos

Edgar Gurgel

[edgargurgel@ic.uff.br](mailto:edgargurgel@ic.uff.br)

**Apresentação baseada nos  
slides previamente  
elaborados pelo Prof.  
Leonardo Murta**

# Problema chave: Comunicação



Cliente

Engenheiro de  
Software

# Engenharia de Requisitos

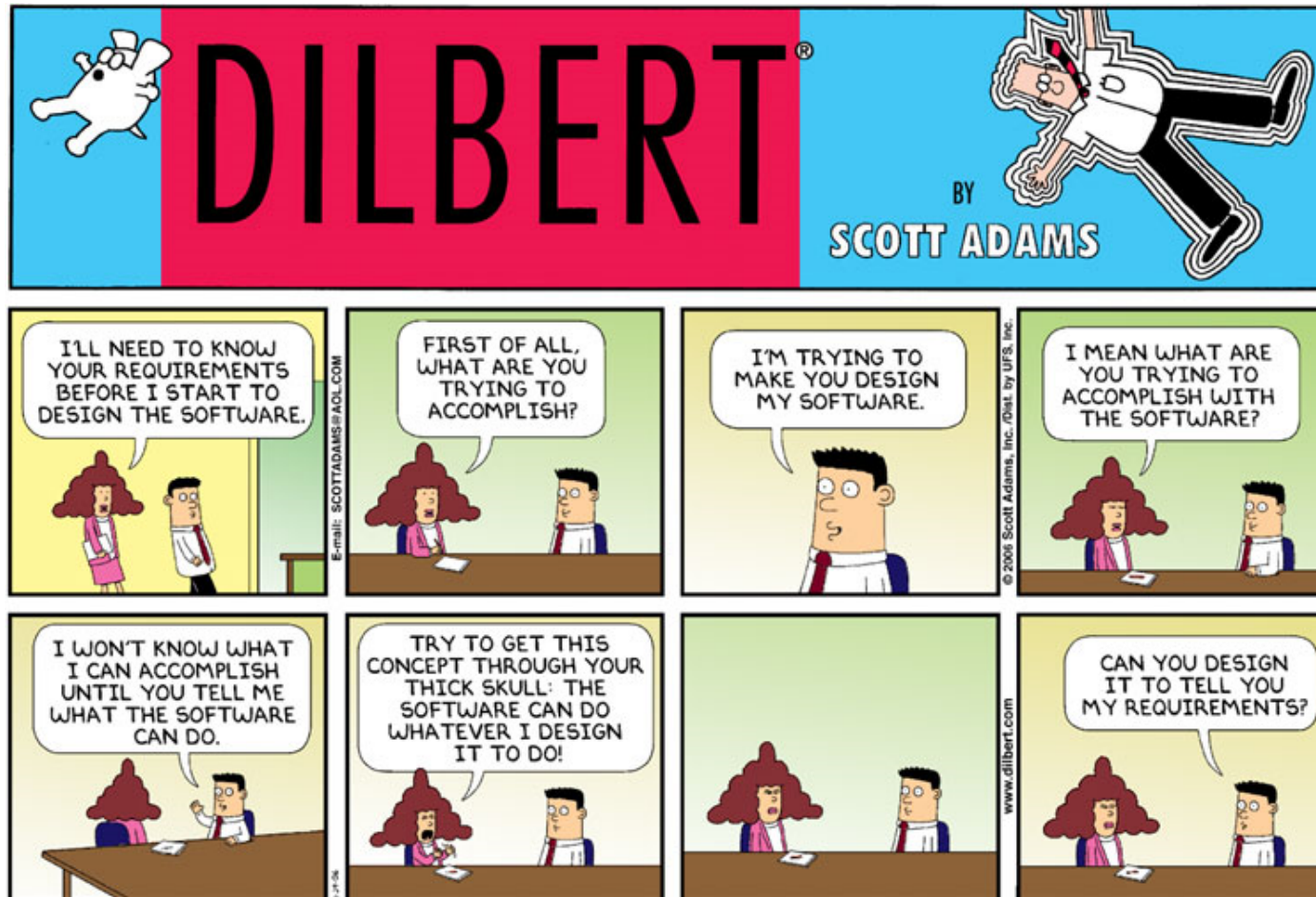
- Concepção
- Elicitação
- Elaboração
- Negociação
- Especificação
- Validação
- Gerenciamento

# Concepção

- Objetivo
  - Ter uma visão geral do negócio
  - Conhecer o cliente e suas expectativas
- Resultados esperados
  - Identificação dos interessados (*stakeholders*)
  - Identificação dos diferentes pontos de vista
  - Visão geral do escopo do sistema



# Elicitação



© Scott Adams, Inc./Dist. by UFS, Inc.

# Elicitação

- Objetivo
  - Entender o que o cliente espera do software
- Problemas mais comuns
  - Escopo variável (mas contrato fixo)
  - Incertezas do cliente
  - Volatilidade dos requisitos

# Elicitação

- Elementos a serem identificados
  - Objetos manipulados pelo sistema
  - Serviços prestados pelo sistema
  - Restrições que devem ser obedecidas (regras de negócio)
  - Critérios de desempenho
- Resultados esperados
  - Narrativa em linguagem natural dos requisitos do sistema
  - Lista de requisitos do sistema



# Elicitação (técnicas)

- Entrevistas
- *Workshops*
- Reuniões de *Brainstorming*
- Prototipação
- Maquetes
- Análise de documentação existente
- Análise de sistemas existentes
- Observação de pessoas trabalhando
- Etc.

# Exercício

- Faça uma pesquisa sobre as Técnicas de Elicitação
  - Reuna-se em grupos;
  - cada grupo deve fazer uma pesquisa das técnicas de elicitação listadas;
  - Faça resumo de três linhas de cada uma das técnicas;
  - Ao final, cada grupo deverá explicar o uso de duas técnicas para a turma.

# Elicitação

## (tipos de requisitos)

- Requisito normal
  - O cliente lembra de falar
  - O cliente espera encontrar esse requisito no sistema
- Requisito esperado
  - Requisito implícito
  - O cliente não lembra de falar
  - O cliente ficará insatisfeito se esse requisito não estiver no sistema
- Requisito excitante
  - O cliente não lembra de falar
  - O cliente não espera encontrar esse requisito no sistema
  - O cliente ficará satisfeito se esse requisito estiver no sistema

# Elicitação

## (cliente x usuário final)

- Nem sempre o cliente é o usuário final
- Cliente
  - Quem contrata e paga pelo serviço
  - Ex.: Administrador de um hospital
- Usuário final
  - Quem usa o software no dia a dia
  - Ex.: Médicos e enfermeiros
- Importante
  - Nunca deixe de elicitar requisitos com os usuários finais, pois sem a colaboração deles, o software não será usado

# Elicitação

## (escolha dos usuários fonte)

- Alguns sistemas serão utilizados por milhares ou milhões de usuários
- Escolha usuários fonte dos requisitos que sejam representativos
- Lembre-se que a regra de Pareto (80-20) aparenta ser válida
  - 20% dos requisitos satisfazem a 80% dos usuários
  - Escolher um usuário muito especialista pode levar a implementação de requisitos que nunca serão utilizados

# Elicitação (requisitos funcionais)

- Narrativa livre
  - “O sistema deve mostrar uma mensagem de status”
- Lista de requisitos
  - RF-1: Uma mensagem de status deve ser mostrada na área inferior da janela (desenho da Fig.1)
  - RF-2: A mensagem deve ser atualizada a cada 60 segundos, com tolerância de 10 segundos para mais ou para menos
  - RF-3: A mensagem deve estar sempre visível
  - RF-4: Se a mensagem for referente a uma tarefa em andamento, o percentual de andamento deve ser mostrado
  - RF-5: Se a mensagem for referente a uma tarefa já terminada, isso deve ser informado com o texto “Finalizada”

# Elicitação

## (requisitos não funcionais)

- Sinônimo: atributos de qualidade
- Disponibilidade
  - DS-1: O sistema deve ficar disponível por 99,5% do tempo nos dias úteis, das 6h às 22h, e 99,95% do tempo nos dias úteis, das 16h às 18h
- Eficiência
  - EF-1: Em condições de pico de uso, deve ter uma reserva de 25% de capacidade de processamento e memória
  - EF-2: O cálculo de interferência deve ser finalizado com sucesso em menos de 5 minutos
  - EF-3: O módulo de parser de XML deve processar 1.000.000 de documentos por segundo

# Elicitação

## (requisitos não funcionais)

- Flexibilidade
  - FL-1: Um novo tipo de sensor deve poder ser configurado no sistema em menos de 3 horas
- Integridade
  - IN-1: Transações históricas dos consumidores só poderão ser vistas por usuários com privilégios de “auditor”
- Interoperabilidade
  - IT-1: O sistema deve ser capaz de importar dados tanto do MS Office (versão 2003 ou maior) quanto do OpenOffice (versão 2.4 ou maior)
- Confiabilidade
  - CF-1: Em cada 1.000 execuções, não mais do que 2 podem apresentar falhas de software



# Elicitação

## (requisitos não funcionais)

- Robustez
  - RB-1: Se acontecer uma falha antes do usuário salvar, o sistema deve recuperar uma versão não salva com perda de conteúdo menor que 1 minuto de trabalho
- Usabilidade
  - US-1: Um usuário treinado deve ser capaz de submeter um pedido de compra em menos que 5 minutos
  - US-2: Um usuário não treinado deve ser capaz de submeter um pedido de compra em menos que 30 minutos
  - US-3: Todos os comandos de menu devem ter teclas de atalho associadas
- Manutenibilidade
  - MN-1: Todos os métodos devem ser documentados utilizando a notação Javadoc
  - MN-2: Modificações corretivas devem ser feitas em menos de 5 horas
  - MN-3: Modificações adaptativas devido a instrumentos legais devem ser feitas em menos de 20 horas

# Elicitação

## (requisitos não funcionais)

- Portabilidade
  - PR-1: O sistema deve poder ser executado em sistema operacional Windows e Linux, nas arquiteturas i386, AIX e SPARC
- Reusabilidade
  - RS-1: O controle de usuários deve reutilizar componentes de autenticação já utilizados no sistema PORTMAP
- Testabilidade
  - TS-1: A complexidade ciclomática máxima de um módulo não pode ser maior que 20

# Elaboração

- Objetivo
  - Explicitar o conhecimento obtido na concepção e elicitação
- Transformar narrativas de linguagem natural para UML
- Sinônimo: Análise de requisitos
- Resultados esperados
  - Casos de uso
  - Classes conceituais

# Negociação

- Objetivo
  - Priorizar e identificar os riscos dos requisitos
  - Eliminar, combinar ou modificar os requisitos
  - Chegar a um consenso sobre a lista final de requisitos
- Conflitos comuns
  - Entre representantes do cliente
    - Requisitos contraditórios
    - Prioridades
  - Entre o cliente e a equipe de desenvolvimento
    - Prazo
    - Custo

# Negociação

- Dimensões principais em negociações
  - Escopo
  - Custo
  - Prazo
  - Qualidade
- As dimensões são interligadas
  - Mudança de posição em uma das dimensões pode gerar consequências nas outras dimensões

# Negociação (dicas)

- Identifique o objetivo do interlocutor
- Defina uma estratégia
  - Saiba de antemão o que pode ser cedido e o que é fundamental de ser mantido
- Ceda nos aspectos relevantes para o interlocutor que não são relevantes para você
  - Não é uma competição. Ambos têm que ganhar!
- Escute com cuidado os argumentos do interlocutor
  - Reavalie a sua posição caso necessário
- Caso chegue a uma situação confortável, faça um acordo de imediato
  - Não busque melhorar a sua posição se a posição atual já é adequada para ambos!

# Especificação

- Objetivo
  - Produzir a especificação de requisitos
- Especificação de requisito engloba
  - Regras de negócio
  - Requisitos funcionais
  - Requisitos não funcionais
  - Casos de uso
  - Classes conceituais

# Validação

- Objetivo
  - Assegurar que a especificação de requisitos está precisa
- Problemas comuns
  - Ambiguidade
  - Inconsistência
  - Omissão
  - Erro



# Validação (questões)

- Os requisitos estão claros?
- A fonte dos requisitos está identificada?
- Os requisitos foram mostrados para essa fonte?
- Os requisitos estão descritos de forma quantitativa?
- Os requisitos estão relacionados via referência cruzada?
- Os requisitos violam alguma restrição do domínio?
- O requisito é testável? Os testes foram especificados?
- Os requisitos são rastreáveis para os modelos e o código subsequente?
- Existem requisitos implícitos?

# Validação

## (exemplos de ambiguidade)

- A janela deve abrir **rapidamente**
- O sistema deve ser **flexível**
- O cálculo deve ser **eficiente**
- A interface com o usuário deve ser **melhor** que a atual
- Não devem ser mostradas **muitas** mensagens de erro
- A exibição do mapa de navegação deve ser **amigável**

# Gerenciamento

- Objetivo
  - Controlar as mudanças nos requisitos
  - Permitir a análise de impacto das mudanças
- Tipos de rastreabilidade
  - Características do sistema
  - Fonte do requisito
  - Dependências entre requisitos
  - Subsistemas
  - Interfaces

# Gerenciamento (matriz de rastreabilidade)

Relationship Matrix: Requirements Trace

Source: Formal Requirements Type: Requirement Link Type: Realisation Profile: Requirements Trace  
Target: Use Case Model Type: UseCase Direction: Target -> Source Refresh Options

	Add New Titles	Add To Shopping Basket	Close Account	Create Account	Create Orders	Delete User	Edit Titles	Go To Checkout	List Current Orders	List Stock Levels	Login	Manage Publishers	Manage Titles	Package Order	Process Order	Receive Orders	Remove From Shopping Basket	Search for Books	Ship Order	View Account details	View History
REQ015 -Process Credit Car...															X						
REQ016 -Add Users				X																	
REQ017 -Remove User		X				X															
REQ018 - Report on User Ac...																			X	X	
REQ019 - Manage Inventory													X								
REQ020 -Receive Books																X					
REQ021 - List Stock Levels										X											
REQ022 -Order Books				X																	
REQ023 -Store and Manage ...																					
REQ024 - Secure Access											X										
REQ025 - Store User Details				X																	
REQ026 - Validate User											X										
REQ027 - Add Books	X																				

Fonte: <http://www.modernanalyst.com>

# 10 princípios de engenharia de requisitos

- Primeiro passo para se resolver um problema é entender o problema
  - Não basta comunicar, é necessário entender!
- Princípio 1: Escute
  - Tente prestar a atenção no que o interlocutor fala
  - Evite interromper a linha de raciocínio do interlocutor
  - Peça detalhes de algo que não ficou claro
  - Não desestimule seu interlocutor com gestos ou palavras

# 10 princípios de engenharia de requisitos

- Princípio 2: Se prepare antes da reunião
  - Tente entender o problema antes da reunião
  - Tente compreender qual é o jargão utilizado no domínio
  - Elabore uma agenda para a reunião
- Princípio 3: É importante ter um mediador
  - O mediador é responsável por manter a reunião com foco apropriado
  - O mediador é responsável por resolver conflitos
- Princípio 4: Comunicação face a face é o ideal
  - Na comunicação face a face é possível perceber gestos
  - A dedicação na comunicação face a face é maior

# 10 princípios de engenharia de requisitos

- Princípio 5: Tome nota das decisões
  - Em pouco tempo, não será possível saber por que uma decisão foi tomada
  - É fundamental documentar as razões de cada decisão
- Princípio 6: Estimule colaborações
  - Duas ou mais mentes pensam melhor que uma
  - Colaborações geram cumplicidade na equipe
- Princípio 7: Mantenha o foco
  - Evite que o reunião se desvie muito do seu objetivo
  - Lembre às pessoas o que ainda precisa ser visto

# 10 princípios de engenharia de requisitos

- Princípio 8: Se algo estiver obscuro, desenhe!
  - Representações visuais ajudam a uniformizar idéias
  - Faça uso de papel e quadro branco em abundância
- Princípio 9: Siga em frente!
  - Se concordarem, siga em frente
  - Se discordarem, siga em frente
  - Se estiverem em dúvida e não for possível tirar a dúvida no momento, siga em frente
- Princípio 10: Negociação não é um jogo
  - Busque por soluções boas para ambas as partes
  - Ceda em aspectos que não são fundamentais
  - Brigue somente pelas batalhas que valem a pena



# Um possível processo...

1. Identifique os interessados no software
2. Se reunia com os interessados e faça perguntas genéricas sobre como funciona o sistema
3. Faça um diagnóstico de uma página sobre o escopo do projeto
4. Revise o diagnóstico com os interessados, visando validar a comunicação anterior
5. Faça reuniões técnicas com os interessados para descobrir os cenários de uso do sistema (entradas, saídas, características, funcionalidades e comportamentos)
6. Faça um breve relatório desses cenários
7. Refina com os interessados esse relatório
8. Priorize esses cenários com os interessados
9. Revise com os interessados o relatório de cenários
10. Inicie o planejamento das etapas de projeto, codificação e testes

# De engenharia de requisitos para implantação

- A priorização dos requisitos determina o conteúdo de cada iteração de implantação do software
  - Dependências entre requisitos pode influenciar nessa ordem
- Entregar mais que o prometido pode ser uma faca de dois gumes
  - Alegria o cliente naquela iteração
  - Chateia o cliente em iterações futuras se isso não se repetir
- Requisitos não funcionais podem implicar em custos pós-implantação
  - Ex: SLA determinando 4 horas para correção de defeitos

# Exercício

- Se coloquem como clientes que desejam contratar uma *software house* para desenvolver uma IDE
  - Inicialmente, cada grupo deve fazer uma “reunião interna” de elicitação de requisitos (5-10 minutos)
    - O resultado dessa reunião deve ser uma lista de até 10 requisitos
  - 3 grupos devem apresentar seus requisitos no quadro, e entrar numa fase de conciliação: combinação, divisão ou remoção de requisitos duplicados
  - Ao final, cada grupo deverá distribuir 10 pontos para os requisitos mais prioritários

# Bibliografia

- Roger Pressman. 2004. *Software Engineering: A Practitioner's Approach*. 6th ed. McGraw-Hill.
- Wiegers, Karl E. 2003. *Software Requirements, Second Edition*. 2nd ed. Microsoft Press.

# Princípios de Engenharia de Requisitos

Edgar Gurgel

[edgargurgel@ic.uff.br](mailto:edgargurgel@ic.uff.br)