

Engenharia de Requisitos

Curso de Ciência da Computação
Disciplina de Engenharia de Software I
Prof. Humberto Torres Marques Neto

Setembro de 2018



Objetivos

- Discutir e propor soluções para os principais problemas do processo de identificação e especificação de requisitos de Sistemas de Informação
- Desenvolver habilidades de uso de técnicas de análise e modelagem de requisitos

Referências Bibliográficas

PRESSMAN, Roger S. Engenharia de Software: uma abordagem profissional. 8 ed. McGraw-Hill, 2016.

SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

BOOCH, Grady, RUMBAUGH, James, JACOBSON, Ivair. UML: Guia do Usuário. 2 ed. Rio de Janeiro: Campus, 2005.

LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e projeto orientado a objetos. 3 ed. Porto Alegre: Bookman, 2007.

Problemas dos Processos de Software (1/3)

- Alterações de metas e estratégias organizacionais
 - Os negócios requerem ciclos de desenvolvimento mais curtos e dinâmicos
 - Requisitos iniciais fracamente definidos

Problemas dos Processos de Software (2/3)

- Falha no gerenciamento de riscos
 - Não há garantias que o sistema irá funcionar
 - O resultado é um grande risco
 - Riscos clássicos são relacionados à pessoas, a processos, a produtos e à tecnologia

Problemas dos Processos de Software (3/3)

- Complexidade do software
 - Demanda crescente por novos softwares
 - Complexidade do domínio
 - Mudanças de equipe

O que fazer?



Modelagem de Sistemas de Informação

- Criar uma linguagem para comunicar decisões que não são óbvias ou que não podem ser inferidas

“Construímos modelo de sistemas porque não é possível compreendê-los em sua totalidade”

Por que modelar?

- Auxiliar:
 - Avaliação dos riscos
 - Entendimento do Problema
 - Definição do problema
 - Gerenciamento do projeto

Por onde começar a modelagem?



Requisitos (SOMMERVILLE, 2009)

- Pode variar de uma declaração abstrata de alto nível de um serviço ou de uma restrição do sistema para uma especificação matemática funcional
- São escritos para serem lidos por qualquer usuário e também pelos arquitetos de sistemas

Requisitos (SOMMERVILLE, 2009)

- Podem ser a base para a proposta de um contrato - portanto, deve ser aberto à interpretação
- Pode ser a base para o contrato em si, portanto, deve ser definido em detalhe

Abstração de requisitos (Davis)

"Se uma empresa quer fechar um contrato para um projeto de desenvolvimento de software de grande porte, deve definir as suas necessidades de forma abstrata o suficiente para que a solução não seja pré-definida. Os requisitos devem ser escritos de forma que vários contratantes possam concorrer pelo contrato e oferecer diferentes maneiras de atender às necessidades da organização do cliente. Uma vez que um contrato tenha sido adjudicado, o contratante deve escrever para o cliente uma definição mais detalhada do sistema, para que esse entenda e possa validar o que o software fará. Ambos os documentos podem ser chamados de documentos de requisitos para o sistema." (SOMMERVILLE, 2009)

Requisitos (SOMMERVILLE, 2009)

- Requisitos de usuário
 - Declarações, em linguagem natural com diagramas, de quais serviços são esperados do sistema e as restrições sob as quais ele deve operar
 - É escrito para ser lido por qualquer usuário e também pelos arquitetos de sistemas

Requisitos (SOMMERVILLE, 2009)

- Requisitos de sistema (1/2)
 - Definem, detalhadamente, as funções, os serviços e as restrições operacionais do sistema
 - Devem ser precisos e definir exatamente o que será implementado

Requisitos (SOMMERVILLE, 2009)

- Requisitos de sistema (2/2)
 - São escritos para serem lidos por usuários finais, por arquitetos de sistemas e também pelos desenvolvedores de software
 - São classificados em requisitos **funcionais**, requisitos **não-funcionais** e requisitos de **domínio**

Tipos de requisitos (SOMMERVILLE, 2009)

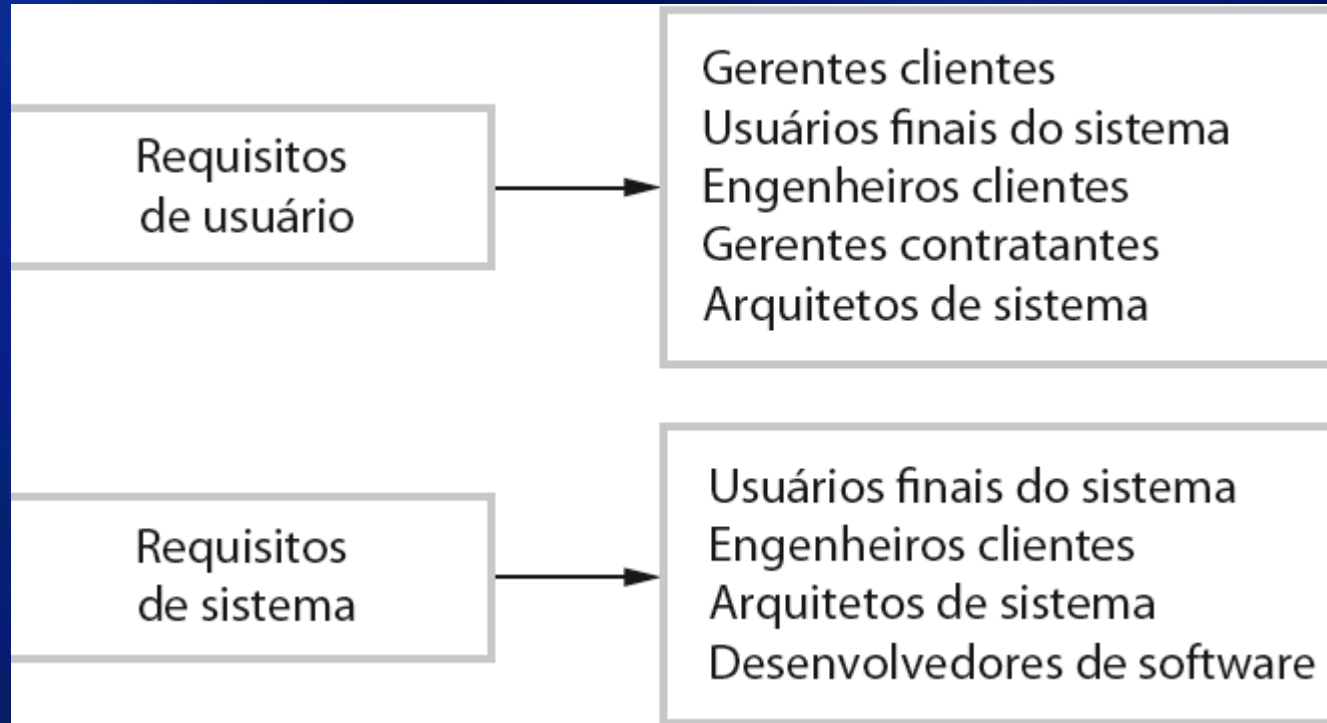
Definição de requisitos de usuário

1. O MHC-PMS deve gerar relatórios gerenciais mensais que mostrem o custo dos medicamentos prescritos por cada clínica durante aquele mês.

Especificação de requisitos de sistema

- 1.1 No último dia útil de cada mês deve ser gerado um resumo dos medicamentos prescritos, seus custos e as prescrições de cada clínica.
- 1.2 Após 17:30h do último dia útil do mês, o sistema deve gerar automaticamente o relatório para impressão.
- 1.3 Um relatório será criado para cada clínica, listando os nomes dos medicamentos, o número total de prescrições, o número de doses prescritas e o custo total dos medicamentos prescritos.
- 1.4 Se os medicamentos estão disponíveis em diferentes unidades de dosagem (por exemplo, 10 mg, 20 mg), devem ser criados relatórios separados para cada unidade.
- 1.5 O acesso aos relatórios de custos deve ser restrito a usuários autorizados por uma lista de controle de gerenciamento de acesso.

Leitores dos requisitos (SOMMERVILLE, 2009)



Requisitos (SOMMERVILLE, 2009)

- Requisitos funcionais
 - São declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em determinadas situações
 - Também podem estabelecer explicitamente o que o sistema não deve fazer

Requisitos funcionais (SOMMERVILLE, 2009)

- Devem descrever a funcionalidade ou os serviços do sistema
- Dependem do tipo de software, possíveis usuários e do tipo de sistema em que o software é usado

Requisitos funcionais (SOMMERVILLE, 2009)

- Requisitos funcionais dos usuários podem ser declarações de alto nível a respeito do que o sistema deve fazer
- Requisitos funcionais do sistema devem descrever detalhadamente os serviços do sistema

Requisitos funcionais do MHC-PMS

1. Um usuário deve ser capaz de pesquisar as listas de agendamentos para todas as clínicas
2. O sistema deve gerar, a cada dia, para cada clínica, uma lista de pacientes esperados para as consultas daquele dia
3. Cada membro da equipe que usa o sistema deve ser exclusivamente identificado pelo seu número de funcionário de 8 dígitos

Requisitos funcionais (SOMMERVILLE, 2009)

- Problemas surgem quando os requisitos não são precisamente definidos
- Requisitos ambíguos podem ser interpretados de maneiras diferentes por desenvolvedores e usuários

Requisitos funcionais (SOMMERVILLE, 2009)

- Considere o termo 'pesquisa' no requisito 1
- A intenção do usuário

Busca pelo nome de um paciente em todas as consultas em todas as clínicas

- Interpretação do desenvolvedor

Busca pelo nome de um paciente em uma clínica. O usuário escolhe a clínica e em seguida pesquisa

Requisitos funcionais (SOMMERVILLE, 2009)

- Integridade e consistência:

Em princípio, os requisitos devem ser completos e consistentes

- Completos: devem incluir descrições de todos os serviços necessários
- Consistentes: não deve haver conflitos ou contradições nessas descrições

Requisitos (SOMMERVILLE, 2009)

- Requisitos não-funcionais
 - São restrições sobre os serviços ou as funções oferecidos pelo sistema
 - Normalmente, aplicam-se ao sistema como um todo e incluem restrições de *timing*, restrições sobre o processo de desenvolvimento e padrões

Requisitos não-funcionais (SOMMERVILLE, 2009)

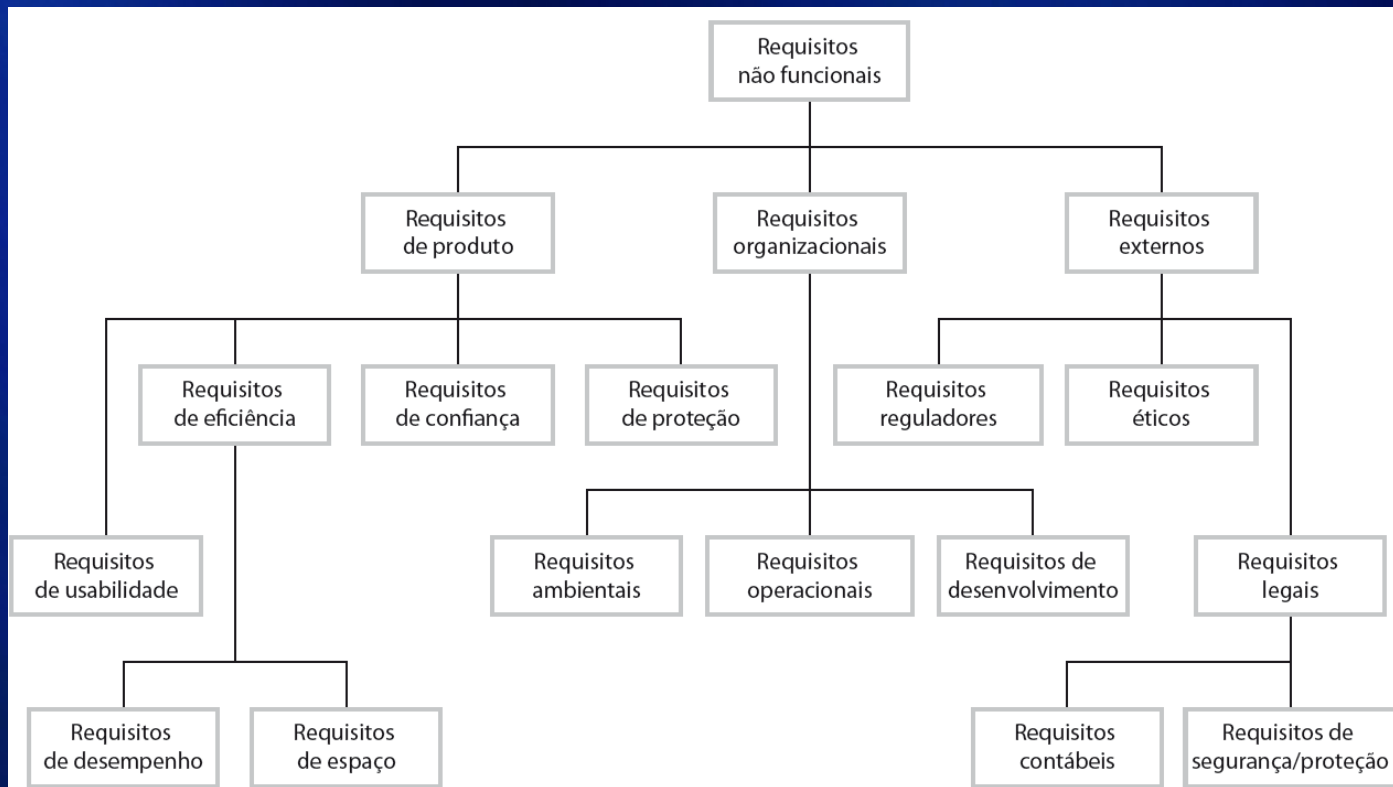
- Podem afetar a arquitetura geral de um sistema, em vez de componentes individuais

Por exemplo, para assegurar que os requisitos de desempenho sejam cumpridos, você pode ter que organizar o sistema para minimizar a comunicação entre os componentes

Requisitos não-funcionais (SOMMERVILLE, 2009)

- Um único requisito não-funcional, como um requisito de proteção, pode gerar uma série de requisitos funcionais relacionados que definem os serviços do sistema que são necessários
- Ele também pode gerar requisitos que restringem os requisitos existentes

Tipos de requisitos não-funcionais



Requisitos não-funcionais do MHC-PMS

Requisito de produto

O MHC-PMS deve estar disponível para todas as clínicas durante as horas normais de trabalho (segunda a sexta-feira, 8h30 às 17h30). Períodos de não operação dentro do horário normal de trabalho não podem exceder cinco segundos em um dia.

Requisito organizacional

Usuários do sistema MHC-PMS devem se autenticar com seus cartões de identificação da autoridade da saúde.

Requisito externo

O sistema deve implementar as disposições de privacidade dos pacientes, tal como estabelecido no HStan-03-2006-priv.

Requisitos não-funcionais (SOMMERVILLE, 2009)

- Pode ser difícil definir precisamente os requisitos não-funcionais, o que torna difícil a sua verificação
- Requisito não-funcional mensurável possui alguma métrica que pode ser objetivamente testada

Requisitos de Usabilidade do MHC-PMS

1. O sistema deve ser de fácil uso pelo pessoal médico e deve ser organizado de tal forma que os erros dos usuários sejam minimizados. (Meta)
2. A equipe médica deve ser capaz de usar todas as funções do sistema depois de quatro horas de treinamento.
3. Após esse treinamento, o número médio de erros cometidos pelos usuários experientes não deve exceder dois por hora de uso do sistema. (Requisito não-funcional testável)

Medidas de requisitos não-funcionais

Propriedade	Medida
Velocidade	Transações processadas/segundo Tempo de resposta de usuário/evento Tempo de atualização de tela
Tamanho	Megabytes Número de chips de memória ROM
Facilidade de uso	Tempo de treinamento Número de <i>frames</i> de ajuda
Confiabilidade	Tempo médio para falha Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo de reinício após falha Percentual de eventos que causam falhas Probabilidade de corrupção de dados em caso de falha
Portabilidade	Percentual de declarações dependentes do sistema-alvo Número de sistemas-alvo

Requisitos (SOMMERVILLE, 2009)

- Requisitos de domínio
 - São provenientes do domínio da aplicação e refletem as características e as restrições desse domínio
 - Podem ser funcionais ou não-funcionais

Requisitos de domínio (SOMMERVILLE, 2009)

- O domínio operacional do sistema impõe requisitos ao sistema

Por exemplo, um sistema de controle de trem deve levar em conta as características de frenagem em diferentes condições climáticas

Requisitos de domínio (SOMMERVILLE, 2009)

- Requisitos de domínio criam novos requisitos funcionais, restrições sobre requisitos existentes ou definem cálculos específicos
- Se os requisitos de domínio não forem satisfeitos, o sistema pode ser impraticável

Requisitos de domínio (SOMMERVILLE, 2009)

- Exemplo do Sistema de Segurança de Trem

- A desaceleração do trem deve ser computada

como: $D_{train} = D_{control} + D_{gradient}$

onde $D_{gradient}$ é $9.81ms^2 * \text{gradiente} / \alpha$

compensado e onde os valores de $9.81ms^2 / \alpha$

são conhecidos para diferentes tipos de trem

Requisitos de domínio (SOMMERVILLE, 2009)

- Compreensibilidade
 - Requisitos são expressos na linguagem do domínio da aplicação
 - Isto pode não ser compreendido pelos engenheiros de software que desenvolvem o sistema

Requisitos de domínio (SOMMERVILLE, 2009)

- “Implicitude”
 - Especialistas de domínio compreendem tão bem essa área que eles não pensam em tornar explícitos os requisitos de domínio

Pontos importantes (1/2)

- Os requisitos para um sistema de software estabelecem o que o sistema deve fazer e definir restrições sobre o seu funcionamento e implementação
- Os requisitos funcionais são declarações dos serviços que o sistema deve fornecer ou são descrições de como alguns processamentos devem ser realizados

Pontos importantes (2/2)

- Muitas vezes os requisitos não-funcionais, limitam o sistema a ser desenvolvido e o processo de desenvolvimento a ser usado
- Muitas vezes eles se relacionam com as propriedades emergentes do sistema e, portanto, se aplicam ao sistema como um todo

Documento de Requisitos (1/2)

- O documento de requisitos de software é a declaração oficial do que é demandado dos desenvolvedores do sistema
- Deve incluir uma definição de requisitos do usuário e uma especificação de requisitos do sistema

Documento de Requisitos (2/2)

- Não necessariamente é um documento de projeto
- Na medida do possível, deve definir O QUE o sistema deve fazer ao invés de COMO deve fazê-lo

Requisitos e Métodos Ágeis

- Muitos métodos ágeis argumentam que a produção de um documento de requisitos é um desperdício de tempo pois esses mudam rapidamente
- Ou seja, o documento estará sempre desatualizado

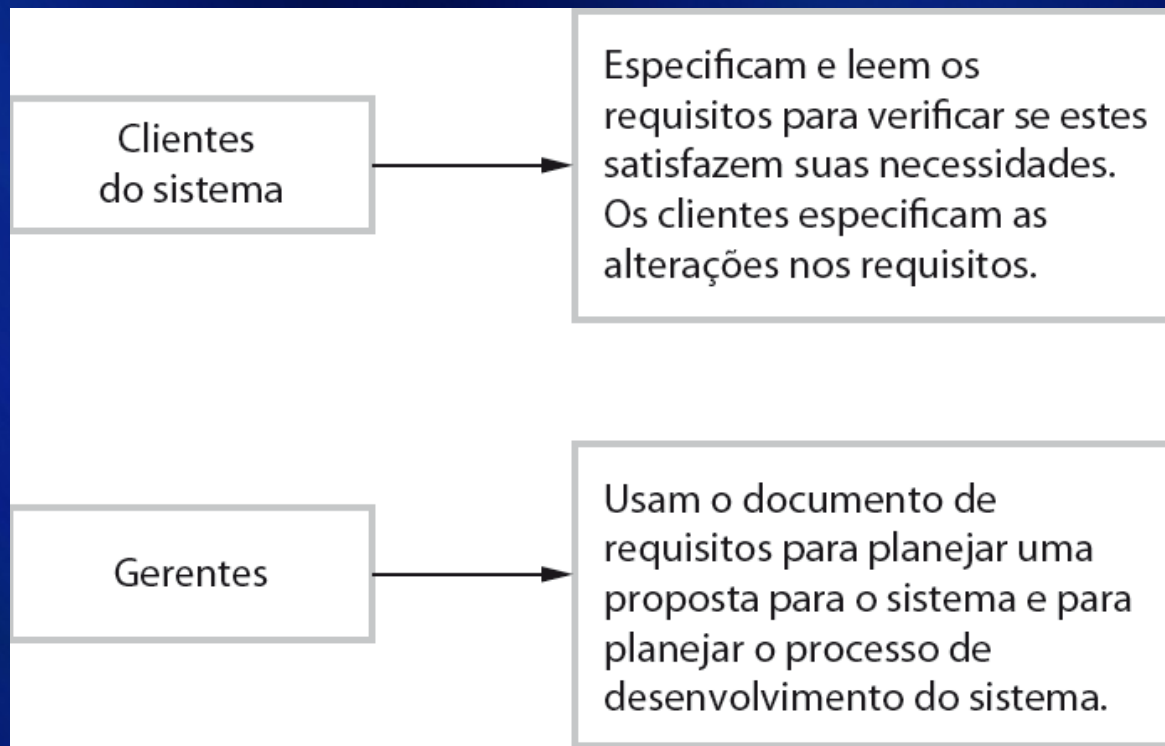
Requisitos e Métodos Ágeis

- Métodos ágeis, tais como XP usam a engenharia de requisitos incrementais e expressam os requisitos como “histórias de usuário”

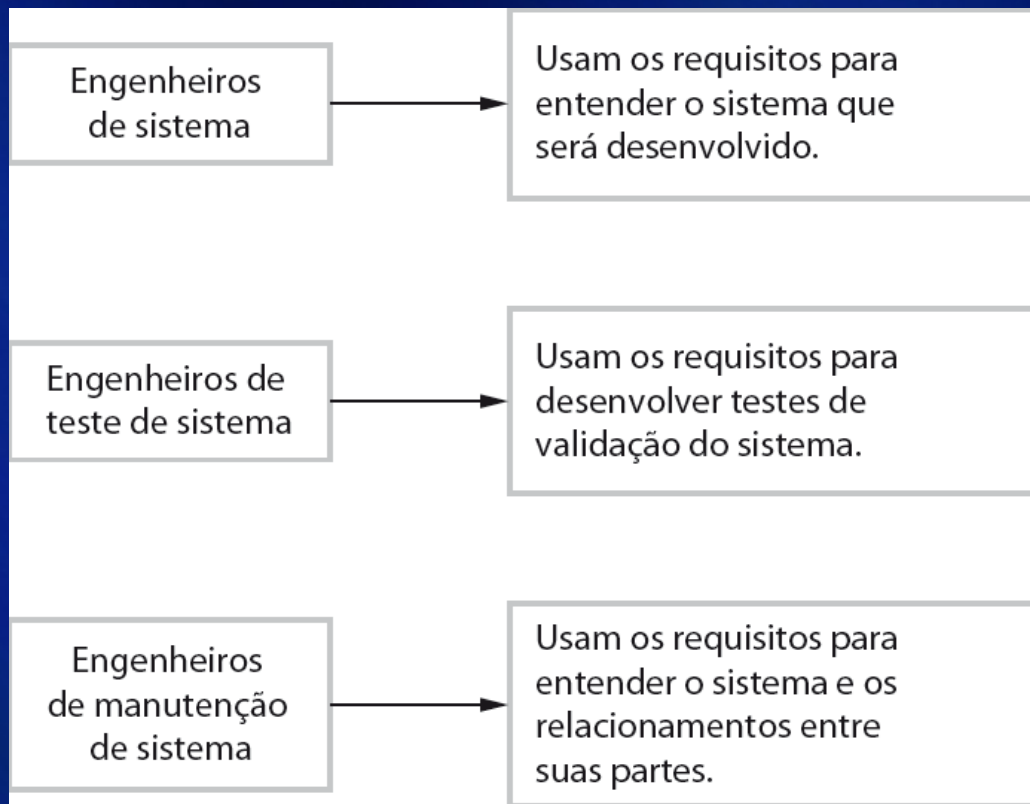
Requisitos e Métodos Ágeis

- Isto é problemático para sistemas que exigem várias análises pré-entrega (por exemplo, sistemas críticos) ou sistemas desenvolvidos por várias equipes

Usuários de um documento de requisitos



Usuários de um documento de requisitos



Variabilidade do documento de requisitos

- As informações no documento de requisitos dependem do tipo de sistema e da abordagem de desenvolvimento usada
- Normalmente, os sistemas desenvolvidos de forma incremental terão menos detalhes no documento de requisitos

Variabilidade do documento de requisitos

- Os padrões dos documentos de requisitos foram concebidos, tendo como exemplo, a norma IEEE
- Esses são aplicáveis, principalmente, aos requisitos para projetos de engenharia de sistemas de grande porte

Estrutura de um documento de requisitos

Capítulo	Descrição
Prefácio	Deve definir os possíveis leitores do documento e descrever seu histórico de versões, incluindo uma justificativa para a criação de uma nova versão e um resumo das mudanças feitas em cada versão.
Introdução	Deve descrever a necessidade para o sistema. Deve descrever brevemente as funções do sistema e explicar como ele vai funcionar com outros sistemas. Também deve descrever como o sistema atende aos objetivos globais de negócio ou estratégicos da organização que encomendou o software.
Glossário	Deve definir os termos técnicos usados no documento. Você não deve fazer suposições sobre a experiência ou o conhecimento do leitor.
Definição de requisitos de usuário	Deve descrever os serviços fornecidos ao usuário. Os requisitos não funcionais de sistema também devem ser descritos nessa seção. Essa descrição pode usar a linguagem natural, diagramas ou outras notações compreensíveis para os clientes. Normas de produto e processos que devem ser seguidos devem ser especificados.
Arquitetura do sistema	Deve apresentar uma visão geral em alto nível da arquitetura do sistema previsto, mostrando a distribuição de funções entre os módulos do sistema. Componentes de arquitetura que são reusados devem ser destacados.

Estrutura de um documento de requisitos

Especificação de requisitos do sistema	Deve descrever em detalhes os requisitos funcionais e não funcionais. Se necessário, também podem ser adicionados mais detalhes aos requisitos não funcionais. Interfaces com outros sistemas podem ser definidas.
Modelos do sistema	Pode incluir modelos gráficos do sistema que mostram os relacionamentos entre os componentes do sistema, o sistema e seu ambiente. Exemplos de possíveis modelos são modelos de objetos, modelos de fluxo de dados ou modelos semânticos de dados.
Evolução do sistema	Deve descrever os pressupostos fundamentais em que o sistema se baseia, bem como quaisquer mudanças previstas, em decorrência da evolução de hardware, de mudanças nas necessidades do usuário etc. Essa seção é útil para projetistas de sistema, pois pode ajudá-los a evitar decisões capazes de restringir possíveis mudanças futuras no sistema.
Apêndices	Deve fornecer informações detalhadas e específicas relacionadas à aplicação em desenvolvimento, além de descrições de hardware e banco de dados, por exemplo. Os requisitos de hardware definem as configurações mínimas ideais para o sistema. Requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e os relacionamentos entre esses dados.
Índice	Vários índices podem ser incluídos no documento. Pode haver, além de um índice alfabético normal, um índice de diagramas, de funções, entre outros pertinentes.

Especificação de requisitos

- O processo de escrever os requisitos de usuário e de sistema em um documento de requisitos
- Os requisitos precisam ser compreensíveis para usuários finais e clientes que não têm formação técnica

Especificação de requisitos

- Requisitos de sistema são mais detalhados e podem incluir informações mais técnicas
- Os requisitos podem ser parte de um contrato para o desenvolvimento do sistema
- Portanto, é importante que esses sejam tão completos quanto possível

Forma de escrever

Notação	Descrição
Sentenças em linguagem natural	Os requisitos são escritos em frases numeradas em linguagem natural. Cada frase deve expressar um requisito.
Linguagem natural estruturada	Os requisitos são escritos em linguagem natural em um formulário padrão ou <i>template</i> . Cada campo fornece informações sobre um aspecto do requisito.
Linguagem de descrição de projeto	Essa abordagem usa uma linguagem como de programação, mas com características mais abstratas, para especificar os requisitos, definindo um modelo operacional do sistema. Essa abordagem é pouco usada atualmente, embora possa ser útil para as especificações de interface.
Notações gráficas	Para definição dos requisitos funcionais para o sistema são usados modelos gráficos, suplementados por anotações de texto; diagramas de caso de uso e de sequência da UML são comumente usados.
Especificações matemáticas	Essas notações são baseadas em conceitos matemáticos, como máquinas de estado finito ou conjuntos. Embora essas especificações inequívocas possam reduzir a ambiguidade de um documento de requisitos, a maioria dos clientes não entende uma especificação formal. Eles não podem verificar que elas representam o que eles querem e são relutantes em aceitá-las como um contrato de sistema.

Projeto e requisitos (1/3)

- Em princípio, os requisitos devem indicar o que o sistema deve fazer e o projeto deve descrever como fazer isso
- Na prática, os requisitos e o projeto são inseparáveis

Projeto e requisitos (2/3)

- A arquitetura do sistema pode ser projetada para estruturar os requisitos
- O sistema pode interoperar com outros sistemas que restringem o projeto e impõem requisitos sobre o novo sistema

Projeto e requisitos (3/3)

- O uso de uma arquitetura específica para satisfazer os requisitos não funcionais pode ser um requisito de domínio
- Essa pode ser a consequência de um requisito de um regulador tão completo quanto possível

Especificação em Linguagem Natural

- Os requisitos são escritos como sentenças em linguagem natural complementadas por diagramas e tabelas
- Usado para escrever os requisitos, pois é expressivo, intuitivo e universal
- Isso significa que os requisitos podem ser entendidos pelos usuários e pelos clientes

Diretrizes (1/2)

- Criar um formato padrão e usá-lo para todos os requisitos
- Usar a linguagem de uma forma consistente
- Usar 'deve' para requisitos obrigatórios e 'pode' para os requisitos desejáveis

Diretrizes (2/2)

- Usar o realce de texto para identificar as partes fundamentais do requisito
- Evitar o uso de jargões da área de computação
- Incluir uma justificativa (lógica) de por que um requisito é necessário

Problemas da Linguagem Natural

- Falta de clareza: é difícil conseguir precisão sem tornar o documento de difícil leitura
- Confusão de requisitos: requisitos funcionais e não funcionais tendem a ser misturados
- Amálgama de requisitos: vários requisitos diferentes podem ser expressos juntos

Exemplo (SOMMERVILLE, 2009)

3.2 O sistema deve medir o açúcar no sangue e fornecer insulina, se necessário, a cada dez minutos. *(Mudanças de açúcar no sangue são relativamente lentas, portanto, medições mais frequentes são desnecessárias; medições menos frequentes podem levar a níveis de açúcar desnecessariamente elevados.)*

3.6 O sistema deve, a cada minuto, executar uma rotina de autoteste com as condições a serem testadas e as ações associadas definidas na Quadro 4.3 *(A rotina de autoteste pode descobrir problemas de hardware e software e pode alertar o usuário para a impossibilidade de operar normalmente.)*

Especificações estruturadas (1/2)

- Uma abordagem para escrever requisitos em que a liberdade do escritor de requisitos é limitada e os requisitos são escritos de uma maneira padrão

Especificações estruturadas (2/2)

- Isso funciona bem para alguns tipos de requisitos, por exemplo, requisitos para o sistema embutido de controle, mas às vezes é demasiado rígido para escrever os requisitos de sistema de negócios

Especificações em formulários (1/2)

- Definição da função ou entidade
- Descrição de entradas e de onde eles vêm
- Descrição das saídas e para onde irão
- Informações sobre as informações necessárias para o processamento e outras entidades usadas

Especificações em formulários (2/2)

- Descrição da ação a ser tomada
- Pré e pós condições (se for o caso)
- Os efeitos colaterais (se houver) da operação

Outro exemplo (SOMMERVILLE, 2009)

Bomba de insulina/Software de controle/SRS/3.3.2

Função	Calcula doses de insulina: nível seguro de açúcar.
Descrição	Calcula a dose de insulina a ser fornecida quando o nível de açúcar está na zona de segurança entre três e sete unidades.
Entradas	Leitura atual de açúcar (r2), duas leituras anteriores (r0 e r1).
Fonte	Leitura atual da taxa de açúcar pelo sensor. Outras leituras da memória.
Saídas	CompDose — a dose de insulina a ser fornecida.
Destino	Loop principal de controle.
Ação	CompDose é zero se o nível de açúcar está estável ou em queda ou se o nível está aumentando, mas a taxa de aumento está diminuindo. Se o nível está aumentando e a taxa de aumento está aumentando, então CompDose é calculado dividindo-se a diferença entre o nível atual de açúcar e o nível anterior por quatro e arredondando-se o resultado. Se o resultado é arredondado para zero, então CompDose é definida como a dose mínima que pode ser fornecida.
Requisitos	Duas leituras anteriores, de modo que a taxa de variação do nível de açúcar pode ser calculada.
Pré-condição	O reservatório de insulina contém, no mínimo, o máximo de dose única permitida de insulina.
Pós-condições	r0 é substituída por r1 e r1 é substituída por r2.
Efeitos colaterais	Nenhum.

Especificação tabular (1/2)

- Normalmente é usada para complementar a linguagem natural
- Particularmente útil quando é necessário definir um número de situações alternativas possíveis

Especificação tabular (2/2)

- Por exemplo, o sistema de bomba de insulina baseia seus cálculos sobre a taxa de mudança de nível de açúcar no sangue e a especificação tabular explica como calcular a necessidade de insulina para diferentes cenários

Exemplo (SOMMERVILLE, 2009)

Condição	Ação
Nível de açúcar diminuindo ($r_2 < r_1$)	CompDose = 0
Nível de açúcar estável ($r_2 = r_1$)	CompDose = 0
Nível de açúcar aumentando e a taxa de aumento decrescente [$(r_2 - r_1) < (r_1 - r_0)$]	CompDose = 0
Nível de açúcar aumentando e a taxa de aumento estável ou crescente [$(r_2 - r_1) \geq (r_1 - r_0)$]	CompDose = arredondar $[(r_2 - r_1) / 4]$. Se o resultado arredondado = 0, então CompDose = MinimumDose

Requisitos de Software

- Características
 - Clareza
 - Ser completo
 - Sem ambiguidade
 - Implementável
 - Consistente
 - Verificável
 - Rastreável



Engenharia de Requisitos (1/2)

- Processo de descobrir, analisar, documentar e verificar os serviços e as restrições de um sistema, ou seja, os seus requisitos (SOMMERVILLE, 2007)

Engenharia de Requisitos (2/2)

- A engenharia de requisitos se concentra em problemas e objetivos do mundo real para definir funções e restrições dos sistemas de software

Requisitos são importantes

- Erros introduzidos durante a etapa de requisitos podem causar:
 - perda de vidas
 - prejuízos financeiros
 - atrasos nas entregas
 - aumento de riscos
 - baixa qualidade

Perdas de vidas atribuídas a software (1/2)

- Therac-25

- 3 mortes, 8 pessoas gravemente contaminadas
- especificação incompleta, interface ruim

Perdas de vidas atribuídas a software (2/2)

- Sistema de Ambulâncias de Londres
 - 11 mortes (estimado)
 - inúmeros problemas, inclusive especificação incompleta

Prejuízos financeiros atribuídos a SW (1/2)

- Ariane 5
 - \$500 milhões de dólares
 - overflow durante conversão de ponto flutuante p/ inteiro

Prejuízos financeiros atribuídos a SW (2/2)

- Sonda Orbital de Marte
 - \$125 milhões de dólares
 - um componente usou escala métrica e outro a escala imperial
 - software correto, mas especificação incorreta

Terminologia da engenharia de requisitos



Onde pode dar errado?

No computador (muito raro)

Causas:

falha na energia

falha nos dispositivos de hardware

falha no sistema operacional

falha na rede



requisitos do cliente
propriedades do domínio

especificação



programa
~~computador~~

domínio de aplicação do cliente

domínio da máquina

Onde pode dar errado?

No programa (não tão raro)

Causas:

- erros de programação

- especificação mal entendida

- ausência de controle de mudanças

Deteccção:

- testes do programa em relação á especificação

- inspeções, walkthroughs



Onde pode dar errado?

Na especificação (comum)

Causas:

- requisitos mal entendidos

- escolha inadequada da linguagem de especificação

- especificação ambígua, inconsistente ou incompleta

Deteção:

- inspeções, verificação formal



Onde pode dar errado?

Nos requisitos (comum)

Causas:

- comunicação insuficiente com o cliente/usuários
- ausência de análise
- falha ao lidar com as mudanças

Deteccção:

- inspeções, revisões feitas pelo cliente, modelagem, validação formal, prototipagem



Onde pode dar errado?

Nas propriedades do domínio (muito comum)

Causas:

- Ausência de especialistas no domínio

- Premissas que não foram questionadas

- Análise do domínio insuficiente

Deteção:

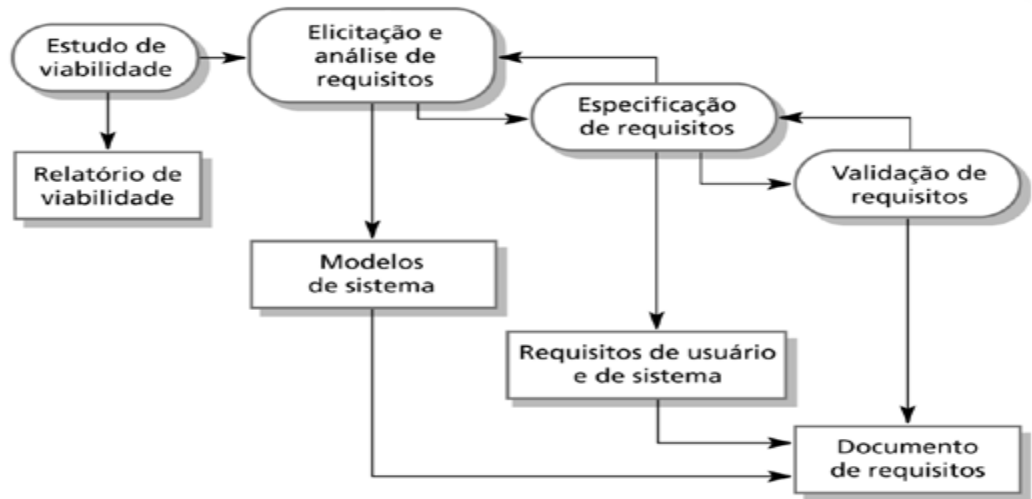
- comunicação com os especialistas que detêm a informação



Processo de Engenharia de Requisitos (SOMMERVILLE, 2007)

Figura 7.1

Processo de engenharia de requisitos.



Engenharia de Requisitos (PRESSMAN, 2011)

■ Principais etapas (1/2):

- Concepção
- Levantamento
- Elaboração



Engenharia de Requisitos (PRESSMAN, 2011)

■ Principais etapas (2/2):

- Negociação
- Especificação
- Validação
- Gestão



Concepção

- Entendimento básico do problema
- Definição dos *stakeholders*
- Natureza da solução desejada
- Eficácia da comunicação e colaboração preliminares

Levantamento de Requisitos

- Por que é tão difícil?
 - Problemas de escopo
 - Problemas de entendimento do que realmente é necessário fazer
 - Problemas de *volatilidade*



Como identificar os requisitos? (1/4)

- Avaliar o negócio e a viabilidade técnica do sistema proposto
- Identificar as pessoas certas que participarão desse processo



Como identificar os requisitos? (2/4)

- Definir o ambiente tecnológico necessário (visão geral)
- Identificar restrições do domínio do problema (características do negócio)



Como identificar os requisitos? (3/4)

- Escolher os métodos que serão utilizados (entrevistas, JAD, reuniões, etc.)
- Envolver pessoas que podem ter pontos de vistas diferentes acerca do que está sendo identificado



Como identificar os requisitos? (4/4)

- Identificar a existência de requisitos ambíguos
- Criar cenários para facilitar a análise dos envolvidos no processo



Elaboração e Negociação de Requisitos

- Questões que devem ser respondidas (1/2):
 - Os requisitos identificados estão de acordo com o objetivo do sistema?
 - O nível de abstração de cada requisito está adequado para se iniciar a sua análise?

Elaboração e Negociação de Requisitos

- Questões que devem ser respondidas (2/2):
 - Existe a possibilidade de integrar e simplificar requisitos identificados?
 - Qual a inter-relação dos requisitos?
 - Qual o impacto da retirada de cada um? (minimizar custo e prazo de entrega)

Especificação dos Requisitos

- Documentação escrita
- Modelagem gráfica
- Modelagem matemática
- Protótipos
- Combinação dos anteriores



Modelagem

- Diagrama de casos de uso (diagrama de contexto)
- Modelo de domínio
- Diagrama de atividades
- Diagrama de pacotes
- Diagrama de integração de sistemas de informação



Validação dos Requisitos Identificados

- Tudo que foi especificado pode ser interpretado corretamente?
- O desenvolvimento do requisito é factível tanto no âmbito do negócio e quanto no da tecnologia?

Gestão dos Requisitos

- O que fazer com a volatilidade das premissas que geraram um requisito?
- Até que ponto é necessário manter um histórico das modificações?

Engenharia de Requisitos (SOMMERVILLE, 2009)

- Os processos usados para a engenharia de requisitos variam muito, dependendo do domínio da aplicação, das pessoas envolvidas e da organização que desenvolve os requisitos

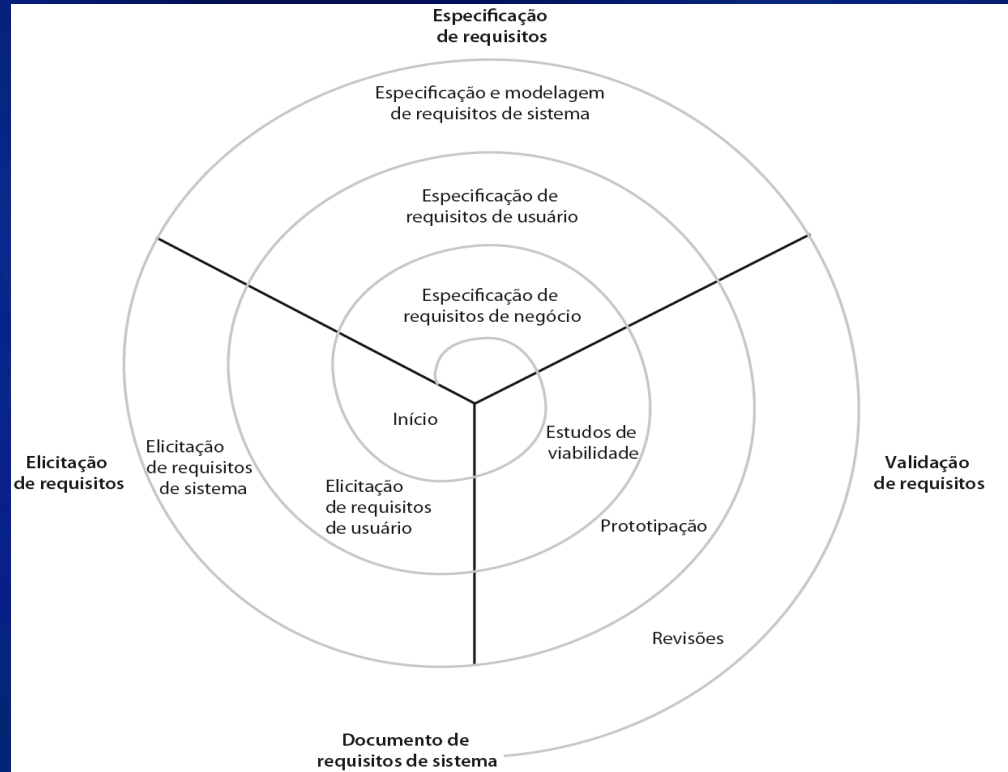
Engenharia de Requisitos (SOMMERVILLE, 2009)

- No entanto, existe uma série de atividades genéricas comuns a todos os processos
 - Elicitação de requisitos
 - Análise de requisitos
 - Validação de requisitos
 - Gerenciamento de requisitos

Engenharia de Requisitos (SOMMERVILLE, 2009)

- Na prática, engenharia de requisitos é uma atividade iterativa em que estes processos são intercalados

Engenharia de requisitos (SOMMERVILLE, 2007)



Estudos de viabilidade (1/2)

- Um estudo de viabilidade decide se vale a pena ou não gastar tempo e esforço com sistema proposto

Estudos de viabilidade (2/2)

- É um estudo breve e focalizado que verifica
 - Se o sistema contribui para os objetivos da organização
 - Se o sistema pode ser implementado usando tecnologia atual e dentro do orçamento
 - Se o sistema pode ser integrado a outros

Implementação do estudo de viabilidade

- Baseado na avaliação de informação (o que é requerido), coleta de informação e escrita de relatório

Implementação do estudo de viabilidade

- Questões para as pessoas da organização
 - O que faria se o sistema não fosse implementado?
 - Quais são os problemas com processo atuais?
 - Como o sistema proposto ajudará?

Implementação do estudo de viabilidade

- Questões para as pessoas da organização
 - Quais serão os problemas de integração?
 - Tecnologia nova é necessária? Quais habilidades?
 - Quais recursos devem ser apoiados pelo sistema proposto?

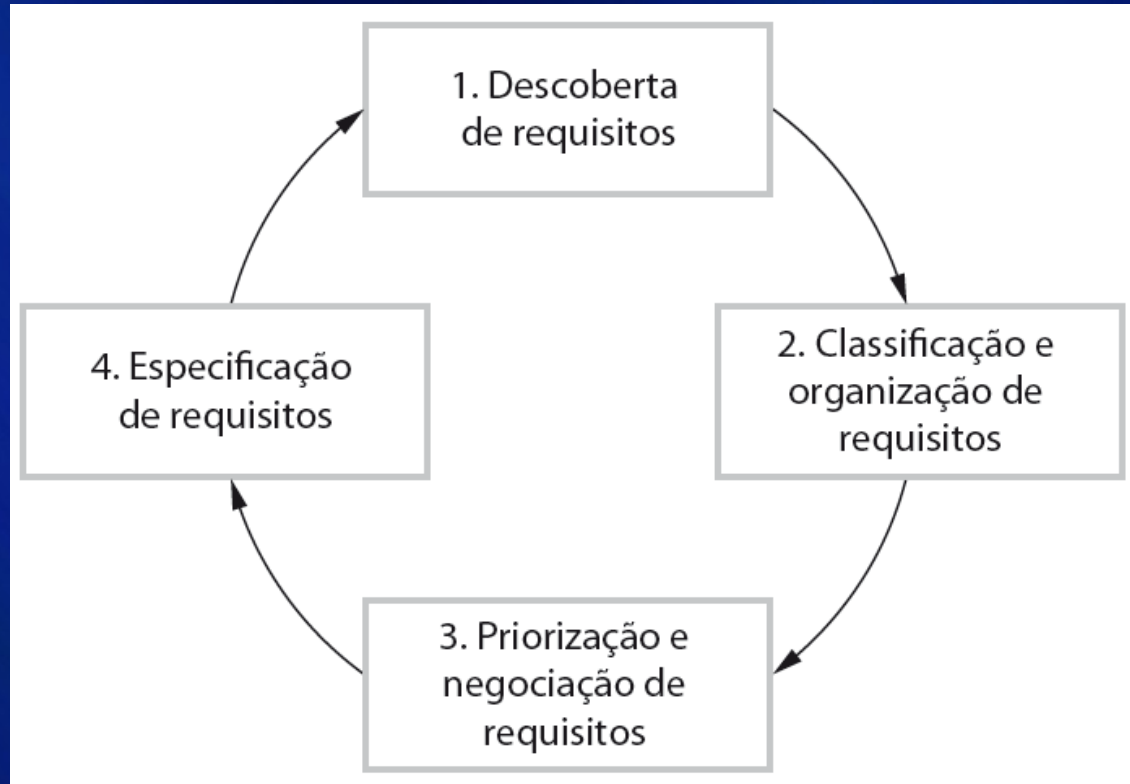
Elicitação e análise (1/2)

- Algumas vezes chamada de elicitación de requisitos ou de descoberta de requisitos
- Envolve pessoal técnico trabalhando com os clientes para descobrir sobre o domínio de aplicação, os serviços que o sistema deve fornecer e sobre as restrições operacionais

Elicitação e análise (2/2)

- Pode envolver usuários finais, gerentes, engenheiros envolvidos na manutenção, especialistas de domínio, representantes de sindicato, etc.
- Estes são chamados *stakeholders*

Espiral de elicitação e análise de requisitos



Problemas de análise de requisitos (1/2)

- *Stakeholders* não sabem o que eles realmente querem
- *Stakeholders* expressam requisitos em seus próprios termos
- Diferentes *stakeholders* podem ter requisitos conflitantes

Problemas de análise de requisitos (2/2)

- Fatores organizacionais e políticos podem influenciar os requisitos de sistema
- A mudança de requisitos durante o processo de análise
- Novos *stakeholders* podem surgir e o ambiente de negócio muda

Pontos importantes (1/3)

- O documento de requisitos de software é uma declaração dos requisitos do sistema acordada
- Deve ser organizada de forma que os clientes do sistema e desenvolvedores de software possam usá-la

Pontos importantes (2/3)

- O processo de engenharia de requisitos é um processo iterativo incluindo um estudo de viabilidade, elicitação e análise, especificação e validação de requisitos

Pontos importantes (3/3)

- A elicitação e análise é um processo iterativo que pode ser representado como uma espiral de atividades – descoberta de requisitos, classificação e organização de requisitos, negociação de requisitos e documentação de requisitos.

Descoberta de requisitos

- É o processo de reunir informações sobre os sistemas propostos e existentes, e obter requisitos de usuário e de sistema a partir dessas informações
- As fontes de informação incluem documentação, *stakeholders* e as especificações de sistemas similares

Pontos de vista (1/2)

- Pontos de vista são uma maneira de estruturar os requisitos para representar as perspectivas de *stakeholders* diferentes
- *Stakeholders* podem ser classificados em diferentes pontos de vista

Pontos de vista (2/2)

- Essa análise de múltiplas perspectivas é importante, pois, não há uma maneira única correta para analisar os requisitos de sistema

Tipos de pontos de vista (1/3)

- Pontos de vista de interação
 - São as pessoas ou os outros sistemas que interagem diretamente com o sistema
 - Em um sistema de caixa eletrônica bancário, os clientes e o banco de dados de contas são pontos de vista de interação

Tipos de pontos de vista (2/3)

- Pontos de vista indiretos
 - São os *stakeholders* que não usam o sistema diretamente, mas que influenciam os requisitos
 - Em um sistema de caixa eletrônico bancário, gerência e pessoal de proteção são pontos de vista indiretos

Tipos de pontos de vista (3/3)

- Pontos de vista de domínio
 - São as características e restrições de domínio que influenciam os requisitos
 - Em um sistema de caixa eletrônico bancário, um exemplo seria os padrões para comunicações entre bancos

Identificação de pontos de vista (1/2)

- Identificar pontos de vista usando:
 - Fornecedores e receptores de serviços do sistema
 - Sistemas que devem interfacear diretamente com o sistema que está sendo especificado
 - Regulamentos e padrões

Identificação de pontos de vista (2/2)

- Identificar pontos de vista usando:
 - Fontes de requisitos de negócio e de requisitos não funcionais
 - Engenheiros que têm de desenvolver e manter o sistema
 - Marketing e outros pontos de vista de negócio

Entrevista (1/2)

- Em entrevista formal ou informal, a equipe de ER formula questões para os *stakeholders* sobre o sistema que eles usam e o sistema a ser desenvolvido

Entrevista (2/2)

- Existem dois tipos de entrevistas
 - Entrevistas fechadas, onde um conjunto de questões predefinidas são respondidas
 - Entrevistas abertas, onde não há um roteiro predefinido e onde uma variedade de assuntos são explorados com os *stakeholders*

Entrevistas na prática (1/2)

- Normalmente, uma mistura de entrevistas fechadas e abertas
- Entrevistas são boas para obtenção de um entendimento geral do que os *stakeholders* fazem e como eles podem interagir com o sistema

Entrevistas na prática (2/2)

- Entrevistas não são boas para a compreensão de requisitos de domínio
 - Os engenheiros de requisitos podem não entender a terminologia específica de domínio
 - Alguns conhecimentos de domínio são tão específicos que as pessoas acham difícil explicar ou pensam que não valem a pena mencioná-los

Entrevistas efetivas

- Os entrevistadores devem ter mente aberta, desejarem ouvir os *stakeholders* e não ter ideias preconcebidas sobre os requisitos
- Eles devem induzir os entrevistados com uma questão ou uma proposta, e não simplesmente esperar que eles respondam a uma questão tal como '*o que você quer?*'

Cenários (1/2)

- Cenários são exemplos reais de como um sistema pode ser usado

Cenários (2/2)

- Os cenários devem incluir:
 - Uma descrição da situação inicial
 - Uma descrição do fluxo normal de eventos
 - Uma descrição do que pode dar errado
 - Informação sobre outras atividades concorrentes
 - Uma descrição do estado quando o cenário termina

Um Cenário do MHC-PMS (1/2)

Suposição inicial:

O paciente é atendido em uma clínica médica por uma recepcionista; ela gera um registro no sistema e coleta suas informações pessoais (nome, endereço, idade etc.). Uma enfermeira é conectada ao sistema e coleta o histórico médico do paciente.

Normal:

A enfermeira busca o paciente pelo sobrenome. Se houver mais de um paciente com o mesmo sobrenome, o nome e a data de nascimento são usados para identificar o paciente.

A enfermeira escolhe a opção do menu para adicionar o histórico médico.

A enfermeira segue, então, uma série de *prompts* do sistema para inserir informações sobre consultas em outros locais, os problemas de saúde mental (entrada de texto livre), condições médicas (enfermeira seleciona condições do menu), medicação atual (selecionado no menu), alergias (texto livre) e informações da vida doméstica (formulário).

Um Cenário do MHC-PMS (2/2)

O que pode dar errado:

O prontuário do paciente não existe ou não pôde ser encontrado. A enfermeira deve criar um novo registro e registrar as informações pessoais. As condições do paciente ou a medicação em uso não estão inscritas no menu. A enfermeira deve escolher a opção 'outros' e inserir texto livre com descrição da condição/medicação.

O paciente não pode/não fornecerá informações sobre seu histórico médico. A enfermeira deve inserir um texto livre registrando a incapacidade/relutância do paciente em fornecer as informações. O sistema deve imprimir o formulário-padrão de exclusão afirmando que a falta de informação pode significar que o tratamento será limitado ou postergado. Este deverá ser assinado e entregue ao paciente.

Outras atividades:

Enquanto a informação está sendo inserida, o registro pode ser consultado, mas não editado por outros agentes.

Estado do sistema na conclusão:

O usuário está conectado. O prontuário do paciente, incluindo seu histórico médico, é inserido no banco de dados e um registro é adicionado ao *log* do sistema, mostrando o tempo de início e fim da sessão e a enfermeira envolvida.

Casos de uso (1/3)

- Os casos de uso constituem uma técnica baseada em cenários UML que identificam os agentes em uma interação, e que descrevem a interação em si

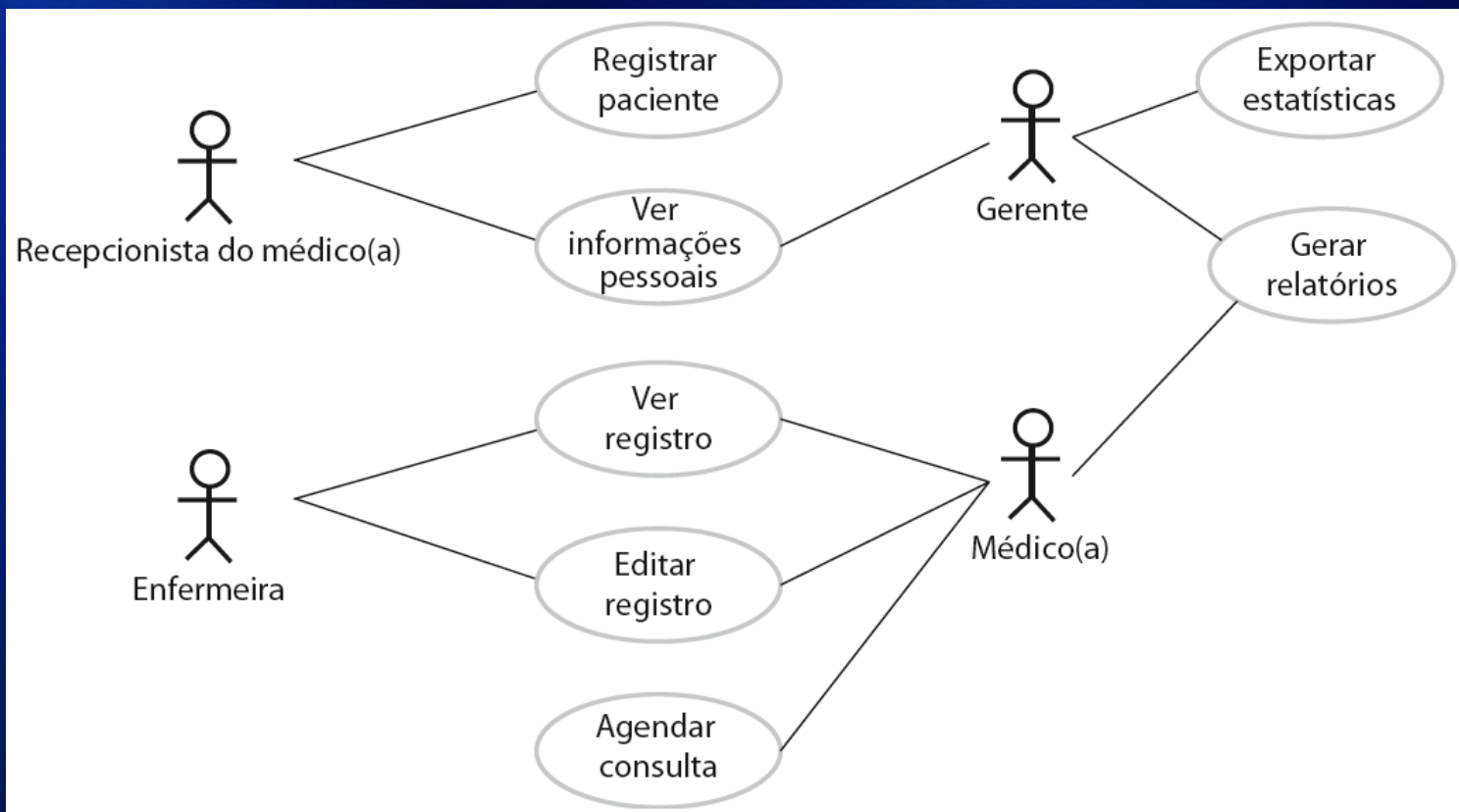
Casos de uso (2/3)

- Um conjunto de casos de uso deve descrever todas as possíveis interações com o sistema
- Modelo gráfico de alto nível complementado por uma descrição tabular mais detalhada

Casos de uso (3/3)

- Diagramas de sequência podem ser usadas para adicionar detalhes aos casos de uso, mostrando a sequência de processamento de eventos no sistema

Um Caso de Uso do MHC-PMS



Etnografia (1/2)

- Um analista gasta um tempo considerável observando e analisando como as pessoas realmente trabalham
- As pessoas não precisam explicar ou articular seu trabalho

Etnografia (2/2)

- Podem ser observados fatores sociais e organizacionais de importância
- Estudos etnográficos têm mostrado que o trabalho geralmente é mais rico e complexo do que o sugerido pelos modelos simples de sistemas

Âmbito da Etnografia (1/3)

- Requisitos que são derivados da maneira como as pessoas realmente trabalham e não da maneira como as definições de processo sugerem que elas deveriam trabalhar

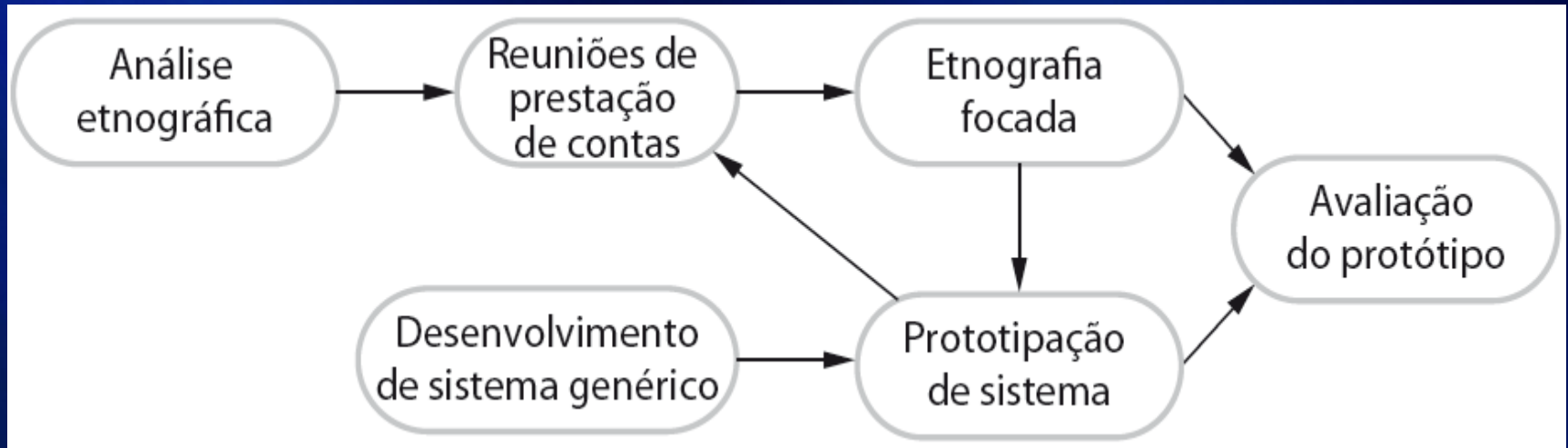
Âmbito da Etnografia (2/3)

- Requisitos que são derivados da cooperação e conscientização das atividades das outras pessoas
 - Consciência do que outras pessoas estão fazendo leva a mudanças no modo como fazemos as coisas

Âmbito da Etnografia (3/3)

- A etnografia é eficaz para a compreensão dos processos existentes, mas não pode identificar novos recursos que devem ser adicionados a um sistema

Etnografia e prototipação (SOMMERVILLE, 2009)



Validação de requisitos (1/2)

- Dedicar-se a mostrar que os requisitos definem o sistema que o cliente realmente deseja

Validação de requisitos (2/2)

- Custos de erros de requisitos são altos e, desse modo, a validação é muito importante
 - A custo da reparação de um erro de requisitos depois da entrega pode equivaler a 100 vezes o custo de reparação de um erro de implementação

Verificação de requisitos (1/2)

- Verificação de validade: o sistema fornece as funções que melhor apoiam as necessidades do cliente?
- Verificação de consistência: existe algum tipo de conflito de requisitos?
- Verificação de completeza: todas as funções requisitadas pelo cliente foram incluídas?

Verificação de requisitos (2/2)

- Verificação de realismo: os requisitos podem ser implementados com o orçamento e a tecnologia disponíveis?
- Facilidade de verificação: os requisitos podem ser verificados?

Técnicas de validação de requisitos (1/2)

- Revisões de requisitos
 - Análise manual sistemática dos requisitos
- Prototipação
 - Uso de um modelo executável do sistema para verificar requisitos

Técnicas de validação de requisitos (2/2)

- Geração de casos de teste.
 - Desenvolvimento de testes para requisitos a fim de verificar a *testabilidade*

Revisões de requisitos (1/2)

- Revisões regulares devem ser feitas enquanto a definição de requisitos está sendo formulada
- Ambos, cliente e fornecedor, devem ser envolvidos nas revisões

Revisões de requisitos (2/2)

- Revisões podem ser formais (com documentos completos) ou informais
- Uma boa comunicação entre desenvolvedores, clientes e usuários podem resolver problemas nos estágios iniciais

Verificação de requisitos (1/2)

- Facilidade de verificação: o requisito é realisticamente testável?
- Facilidade de compreensão: o requisito é adequadamente compreendido?

Verificação de requisitos (2/2)

- Rastreabilidade: a origem do requisito é claramente estabelecida?
- Adaptabilidade: o requisito pode ser mudado sem um grande impacto em outros requisitos?

Gerenciamento de requisitos (1/4)

- Gerenciamento de requisitos é o processo de gerenciamento de mudanças de requisitos durante o processo de engenharia de requisitos e o desenvolvimento de sistema

Gerenciamento de requisitos (2/4)

- Requisitos são, inevitavelmente, incompletos e inconsistentes
 - Novos requisitos surgem durante o processo a medida que as necessidades de negócio mudam e uma melhor compreensão do sistema é desenvolvida
 - Os diferentes pontos de vista têm requisitos diferentes e estes são frequentemente contraditórios.

Gerenciamento de requisitos (3/4)

- É preciso manter o controle das necessidades individuais e manter ligações entre os requisitos dependentes para que você possa avaliar o impacto das mudanças nos requisitos

Gerenciamento de requisitos (4/4)

- É necessário estabelecer um processo formal para fazer propostas de mudança e ligar essas aos requisitos de sistema

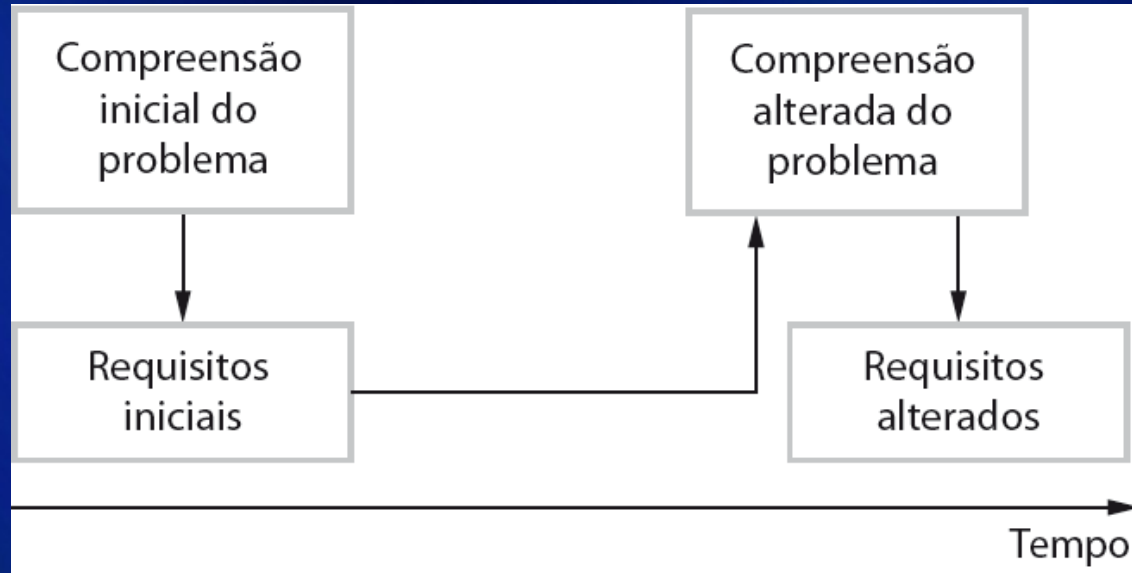
Mudança de requisitos (1/2)

- A priorização dos requisitos em consequência das mudanças de pontos de vista durante o processo de desenvolvimento

Mudança de requisitos (2/2)

- Os clientes do sistema podem especificar os requisitos a partir de uma perspectiva de negócio que conflitam com os requisitos do usuário final
- Os ambientes técnico e de negócio do sistema mudam durante seu desenvolvimento

Evolução de requisitos (SOMMERVILLE, 2009)



Planejamento de gerenciamento de requisitos

1. Identificação de requisitos: cada requisito deve ser identificado exclusivamente para que ele possa ser comparado com outros requisitos

Planejamento de gerenciamento de requisitos

2. Processo de gerenciamento de mudanças:

Esse é o conjunto de atividades que avaliam o impacto e o custo das mudanças

3. Políticas de rastreabilidade: essas políticas definem as relações entre cada requisito e entre os requisitos e o projeto do sistema que deve ser registrado

Planejamento de gerenciamento de requisitos

4. Ferramentas de suporte: as ferramentas de suporte que podem ser usadas variam desde sistemas especialistas, sistemas de gerenciamento de requisitos até planilhas e sistemas de banco de dados simples

Gerenciamento de mudança de requisitos (SOMMERVILLE, 2009)

- Decisão sobre a pertinência de uma mudança:



Requisitos permanentes e voláteis (1/2)

- Requisitos permanentes: são requisitos estáveis, derivados da atividade central da organização do cliente. Por exemplo, um hospital terá sempre médicos, enfermeiros, etc. Podem ser derivados dos modelos de domínio

Requisitos permanentes e voláteis (2/2)

- Requisitos voláteis: são requisitos que mudam durante o desenvolvimento, ou quando o sistema estiver em operação. Um exemplo seria, em um hospital, os requisitos derivados da política de saúde

Classificação de requisitos voláteis (SOMMERVILLE, 2007)

Tabela 7.1 Classificação de requisitos voláteis

Tipo de requisito	Descrição
Requisitos mutáveis	Requisitos que mudam devido a mudanças no ambiente no qual a organização está operando. Por exemplo, em sistemas hospitalares, o financiamento do tratamento de pacientes pode mudar e, assim, exigir que informações de diferentes tratamentos sejam coletadas.
Requisitos emergentes	Requisitos que surgem à medida que a compreensão do sistema pelo cliente progride durante o desenvolvimento do sistema. O processo de projeto pode revelar novos requisitos emergentes.
Requisitos conseqüentes	Requisitos que resultam da introdução do sistema de computador. A introdução do sistema de computador pode mudar os processos da organização e criar novas formas de trabalho que geram novos requisitos de sistema.
Requisitos de compatibilidade	Requisitos que dependem de sistemas ou processos de negócios específicos dentro de uma organização. À medida que eles mudam, os requisitos de compatibilidade do sistema encomendado ou entregue podem também evoluir.

Planejamento (1/2)

- A Identificação de requisitos
 - Como os requisitos são identificados individualmente
- O processo de gerenciamento de mudanças
 - É o processo seguido quando da análise de uma mudança de requisitos

Planejamento (2/2)

- Políticas de rastreabilidade
 - É a quantidade de informações que é mantida sobre os relacionamentos de requisitos
- Apoio de ferramenta CASE
 - O apoio de ferramenta requisitada para auxiliar no gerenciamento das mudanças requisitos

Rastreabilidade (1/2)

- A rastreabilidade está relacionada aos relacionamentos entre os requisitos, suas fontes e o projeto de sistema
- Rastreabilidade da fonte
 - Ligam os requisitos aos *stakeholders* que propuseram os requisitos

Rastreabilidade (2/2)

- Rastreabilidade de requisitos
 - É a ligação dos requisitos dependentes
- Rastreabilidade de projeto
 - Ligam os requisitos aos módulos de projeto

Uma matriz de rastreabilidade (SOMMERVILLE, 2007)

Tabela 7.2 Matriz de rastreabilidade

ID de requisito	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2
1.1		D	R					
1.2			D			R		D
1.3	R			R				
2.1			R		D			D
2.2								D
2.3		R		D				
3.1								R
3.2							R	

Apoio de ferramenta CASE (1/3)

- Armazenamento de requisitos
 - Os requisitos devem ser mantidos em um repositório de dados seguro e gerenciado

Apoio de ferramenta CASE (2/3)

- Gerenciamento de mudanças
 - O processo de gerenciamento de mudanças é um processo de workflow cujos estágios podem ser definidos, e o fluxo de informações entre esses estágios, parcialmente automatizado

Apoio de ferramenta CASE (3/3)

- Gerenciamento de rastreabilidade
 - Recuperação automatizada das ligações entre os requisitos

Gerenciamento de mudanças (1/2)

- Deve ser aplicado à todas as mudanças propostas aos requisitos

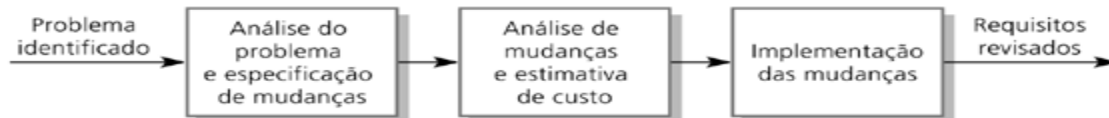
Gerenciamento de mudanças (2/2)

- Estágios principais
 - Análise de problema: discutir problemas e mudanças de requisitos
 - Análise de mudança e estimativa de custo: avaliar os efeitos das mudanças sobre outros requisitos
 - Implementação de mudança: Modificar documentos de requisitos e outros documentos para refletir as mudanças

Gerenciamento de mudanças (SOMMERVILLE, 2007)

Figura 7.10

Gerenciamento de mudanças
de requisitos.



Pontos chave (1/4)

- O processo de engenharia de requisitos inclui um estudo de viabilidade, elicitação e análise de requisitos, validação de requisitos e gerenciamento de requisitos

Pontos chave (2/4)

- A elicitação e a análise de requisitos constituem um processo iterativo, envolvendo entendimento de domínio, coleta, classificação, estruturação, priorização e validação de requisitos
- Os sistemas têm múltiplos *stakeholders* com diferentes requisitos

Pontos chave (3/4)

- Fatores sociais e organizacionais influenciam os requisitos de sistema
- A validação de requisitos está relacionado às verificações de validade, consistência, completeza, realismo e facilidade de verificação

Pontos chave (4/4)

- Mudanças de negócio levam, inevitavelmente, às mudanças de requisitos
- O gerenciamento de requisitos inclui planejamento e gerenciamento de mudanças

Análise do Requisitos



Principais Métodos de Análise

- Decomposição Funcional
- Enfoque de Fluxo de Dados
- Modelagem de Informações
- Baseado em Objetos ou Orientado à Objetos

Qual seria o melhor método? É o melhor em qualquer caso?

Motivações e Benefícios da AOO (1/3)

- Compreensão de domínios de problemas complexos
- Melhoria na interação entre o analista e o especialista de domínio de problema
- Aumento da consistência interna dos resultados da análise (estrutura e comportamento)

Motivações e Benefícios da AOO (2/3)

- Representação explícita dos pontos comuns
- Elaboração de especificações que suportam alterações
- Reutilização dos resultados de análise

Motivações e Benefícios da AOO (3/3)

- Apresentação de uma representação básica consistente (análise e desenho)
- Evolução das ferramentas CASE (*round trip engineering*)
- Desenvolvimento de sistemas multicamadas