

# Capítulo 4

## Engenharia de requisitos



## Tópicos abordados

- Requisitos funcionais e não funcionais
- O documento de requisitos de software
- Especificação de requisitos
- Processos de engenharia de requisitos
- Elicitação e análise de requisitos
- Validação de requisitos
- Gerenciamento de requisitos

- O processo de estabelecer os serviços que o cliente necessita do sistema e as restrições sob as quais ele opera e é desenvolvido.
- Os próprios requisitos são as descrições dos serviços do sistema e restrições geradas durante o processo de engenharia de requisitos.



## O que é um requisito?

- Pode variar de uma declaração abstrata de alto nível de um serviço ou de uma restrição do sistema para uma especificação matemática funcional.
- Isso é inevitável quando os requisitos podem servir a uma função dupla.
  - ✓ Pode 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;
  - ✓ Ambas as declarações podem ser chamadas de requisitos.

## 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. "

## Tipos de requisitos

- Requisitos de usuário
  - ✓ Declarações em linguagem natural com diagramas dos serviços que o sistema deverá fornecer e suas restrições operacionais. Escrito para os clientes.
- Requisitos de sistema
  - ✓ Um documento estruturado estabelecendo descrições detalhadas das funções do sistema, serviços e restrições operacionais. Define o que deve ser implementado assim, pode ser parte de um contrato entre o cliente e o empreiteiro.



# Requisitos de usuário e de sistema

# engenharia de SOFTWARE

## Definição de requisitos de usuário

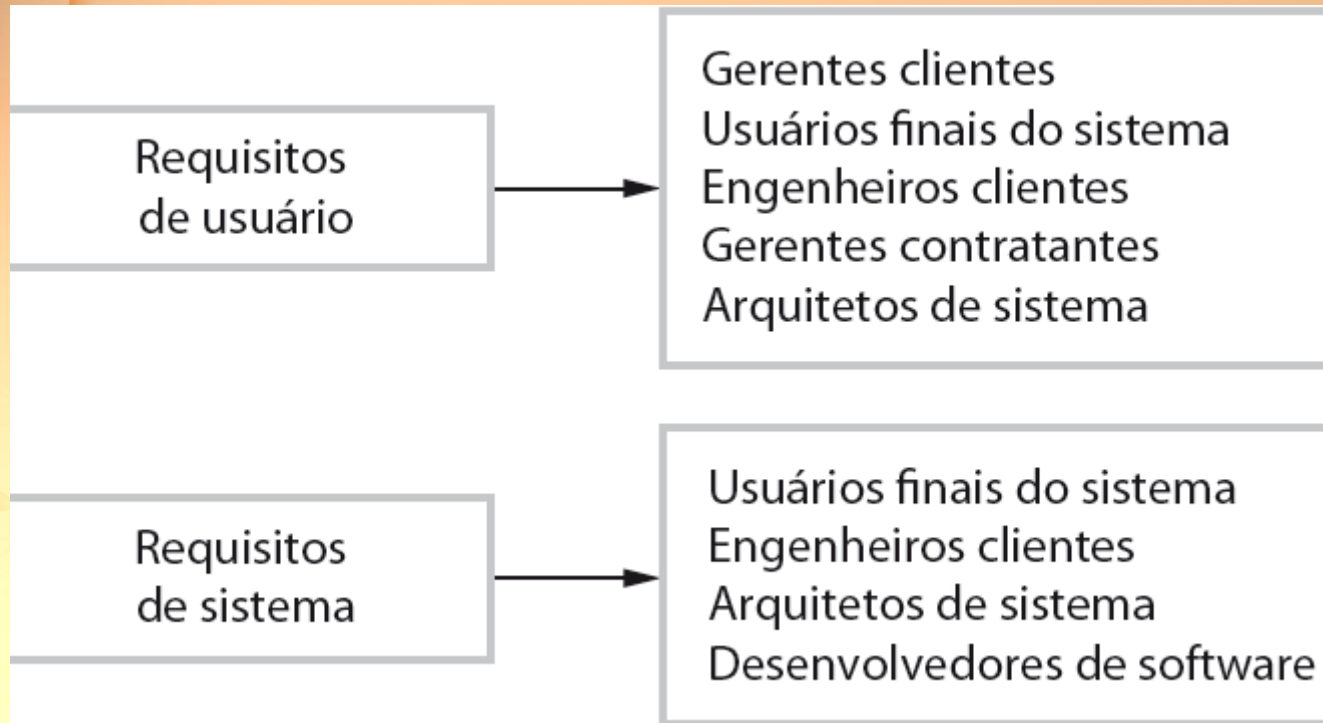
1. O MHC-PMS deve gerar relatórios gerenciais mensais que mostrem o custo dos medicamentos prescritos por cada clínica durante aquele mês.

## 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 de diferentes tipos de especificação de requisitos

# engenharia de SOFTWARE





- Requisitos funcionais
  - ✓ O sistema deve fornecer declarações de serviços, como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em determinadas situações.
  - ✓ Pode explicitar o que o sistema não deve fazer.
- Requisitos não-funcionais
  - ✓ Restrições aos serviços ou funções oferecidas pelo sistema, tais como restrições de tempo, restrições no processo de desenvolvimento, padrões.
  - ✓ Muitas vezes se aplica ao sistema como um todo ao invés de características individuais ou serviços.
- Requisitos de domínio
- Restrições no sistema a partir do domínio de operação

- Descrever a funcionalidade ou os serviços do sistema.
- Depende do tipo de software, possíveis usuários e o tipo de sistema em que o software é usado.
- 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 para o MHC-PMS

# engenharia de SOFTWARE

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



## Imprecisão de requisitos

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

## Integridade e consistência dos requisitos

# engenharia de SOFTWARE

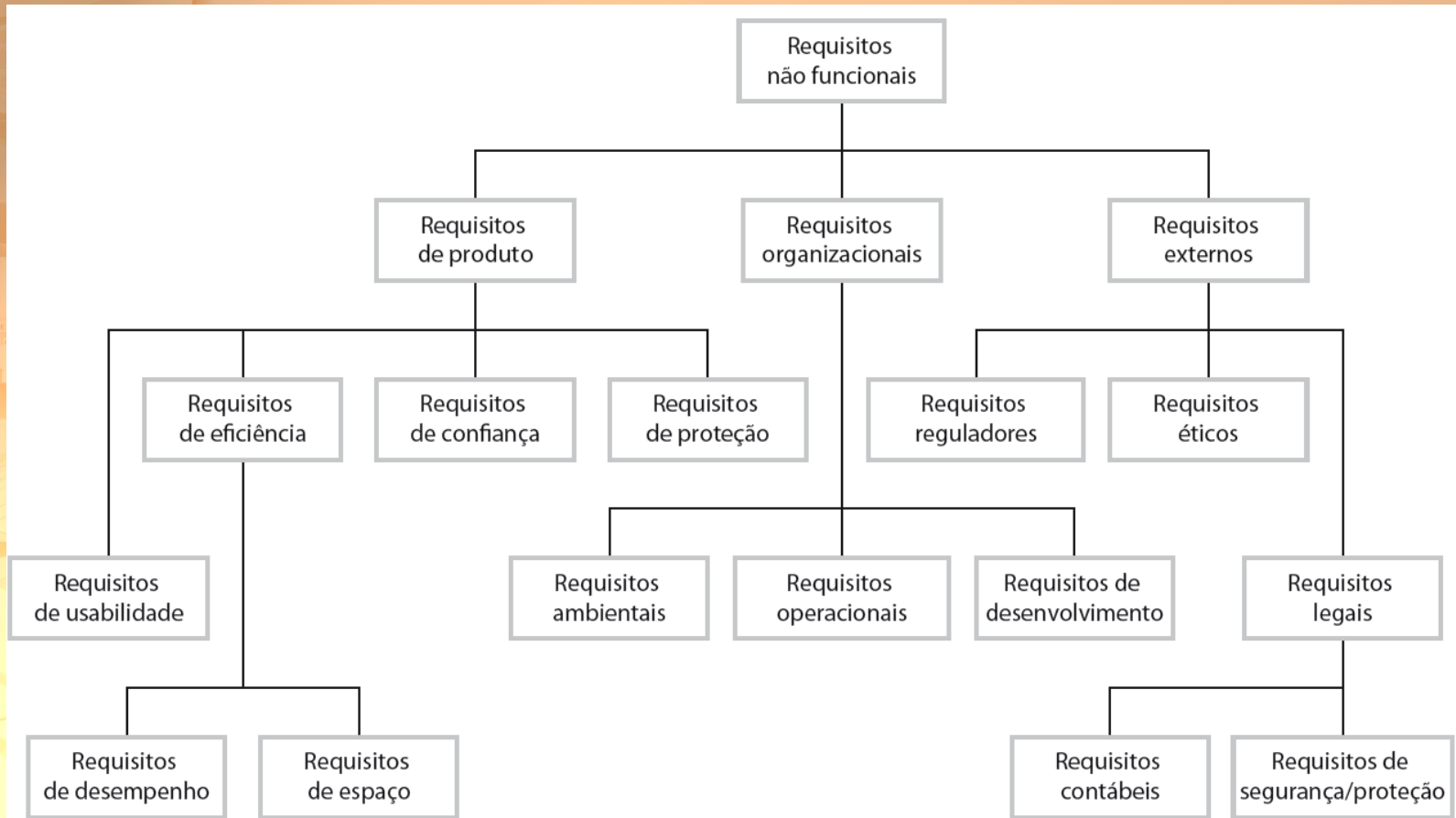
- Em princípio, os requisitos devem ser completos e consistentes.
- Completos
  - ✓ Eles devem incluir descrições de todos os serviços necessários.
- Consistentes
  - ✓ Não devem haver conflitos ou contradições nas descrições dos recursos do sistema.
- Na prática, é impossível produzir documentos de requisitos completos e consistentes .

## Requisitos Não-funcionais

- Esses requisitos definem as propriedades e as restrições do sistema por exemplo, confiabilidade, tempo de resposta e ocupação de área.
- As restrições são capacidades de dispositivos de E/S, as representações do sistema, etc.
- Os requisitos de processo também podem ser especificados impondo um IDE particular, linguagem de programação ou método de desenvolvimento.
- Os requisitos não-funcionais podem ser mais críticos do que os requisitos funcionais. Se esses não forem atendidos, o sistema pode ser inútil.



# Tipos de requisitos não funcionais



# Implementação de requisitos não funcionais

- Requisitos não-funcionais 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.
- 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.

# Classificações de requisitos não funcionais

## engenharia de SOFTWARE

- Requisitos de produto
  - ✓ Requisitos que especificam que o produto entregue deve se comportar de uma maneira particular, por exemplo velocidade de execução, confiabilidade, etc.
- Requisitos organizacionais
  - ✓ Requisitos que são consequência de políticas e procedimentos organizacionais, por exemplo padrões de processo usados, requisitos de implementação, etc.
- Requisitos externos
  - ✓ Requisitos que surgem de fatores externos ao sistema e seu processo de desenvolvimento, por exemplo, requisitos de reguladores, requisitos legais, etc.



# Exemplos de requisitos não funcionais no MHC-PMS

# engenharia de SOFTWARE

## 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.

## Metas e requisitos

- Requisitos não-funcionais podem ser muito difíceis de se definir precisamente e requisitos imprecisos podem ser difíceis de se verificar.
- Metas
  - ✓ A intenção geral do usuário, facilmente usável.
- Requisito não-funcional mensurável.
  - ✓ Uma declaração usando alguma métrica que pode ser objetivamente testada.
- Metas são úteis para desenvolvedores quando exprimem as intenções dos usuários do sistema.

## Requisitos de Usabilidade

# engenharia de SOFTWARE

- 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)
- A equipe médica deve ser capaz de usar todas as funções do sistema depois de quatro horas de treinamento.
- 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)



# Métricas para especificar requisitos não funcionais

## engenharia de SOFTWARE

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 de domínio

- 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 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.

Esse é um requisito de domínio de um sistema de segurança de um trem:

- A desaceleração do trem deve ser computada como:
- $D_{train} = D_{control} + D_{gradient}$ 
  - ✓ onde  $D_{gradient}$  é  $9.81ms^2 * \text{gradiente} / \text{alfa compensado}$  e onde os valores de  $9.81ms^2 / \text{alpha}$  são conhecidos para diferentes tipos de trem.
- É difícil para um não-especialista entender as implicações desse requisito e de como ele interage com outros requisitos.



## Problemas de requisitos de domínio

- Compreensibilidade
  - ✓ Requisitos são expressos na linguagem do domínio da aplicação;
  - ✓ O que geralmente não é compreendido pelos engenheiros de software que desenvolvem o sistema.
- Implicitude
- Especialistas de domínio compreendem tão bem essa área que eles não pensam em tornar explícitos os requisitos de domínio.

## Pontos importantes

# engenharia de SOFTWARE

- 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.
- 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.

## O documento de requisitos de software

# engenharia de SOFTWARE

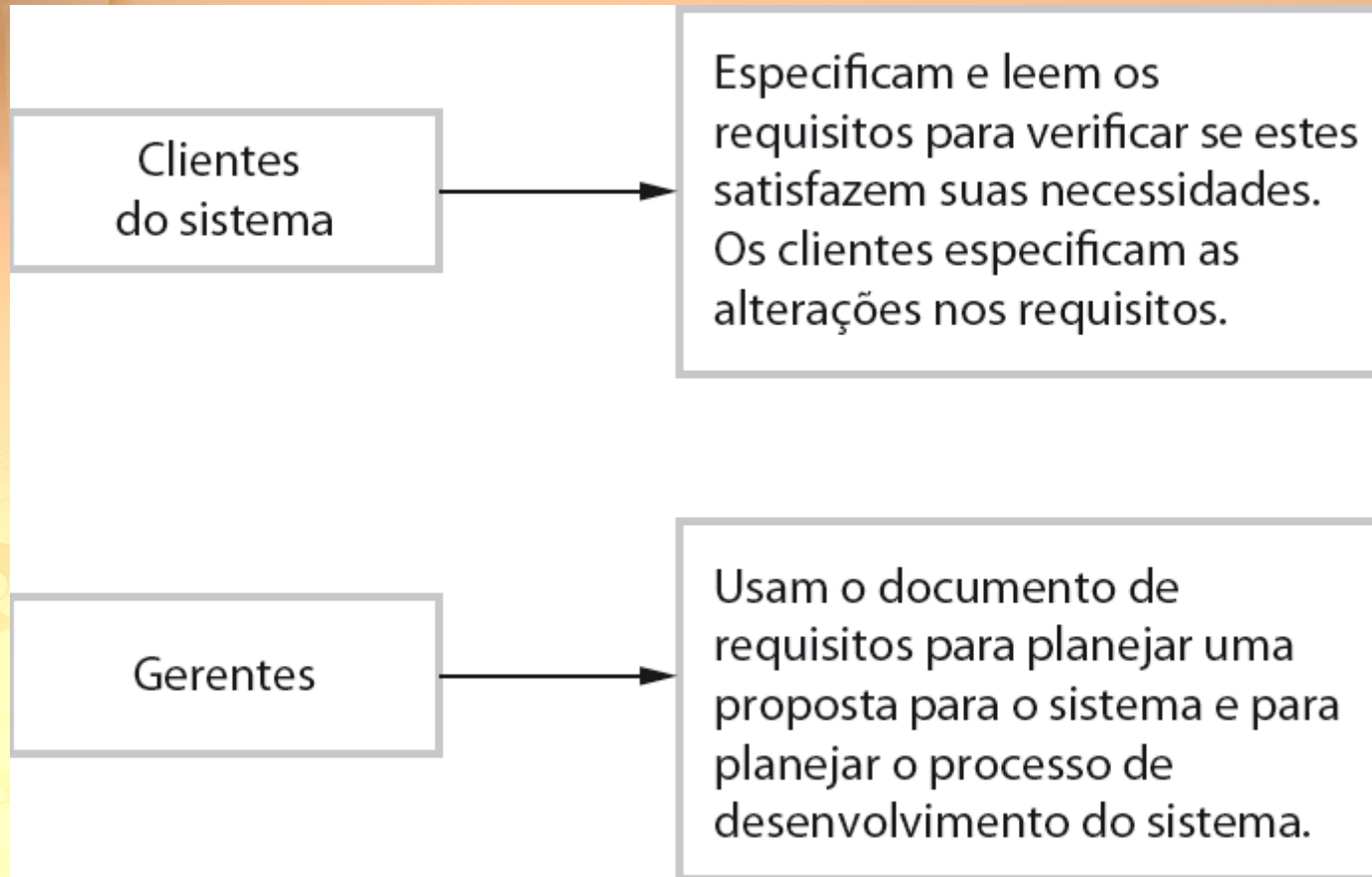
- O documento de requisitos de software é a declaração oficial do que é demandado dos desenvolvedores do sistema.
- Deve incluir ambas, uma definição de requisitos do usuário e uma especificação de requisitos do sistema.
- NÃO é um documento de projeto. Na medida do possível, deve definir O QUE o sistema deve fazer ao invés de COMO deve fazê-lo.



- Muitos métodos ágeis argumentam que a produção de um documento de requisitos é um desperdício de tempo pois esses mudam rapidamente.
- Portanto, o documento estará sempre desatualizado.
- Métodos ágeis, tais como XP usam a engenharia de requisitos incrementais e expressam os requisitos como “estórias de usuário” (discutido no Capítulo 3).
- O que é prático para os sistemas de negócios, mas problemático para sistemas que exigem várias análises pré-entrega (por exemplo, sistemas críticos) ou sistemas desenvolvidos por várias equipes.

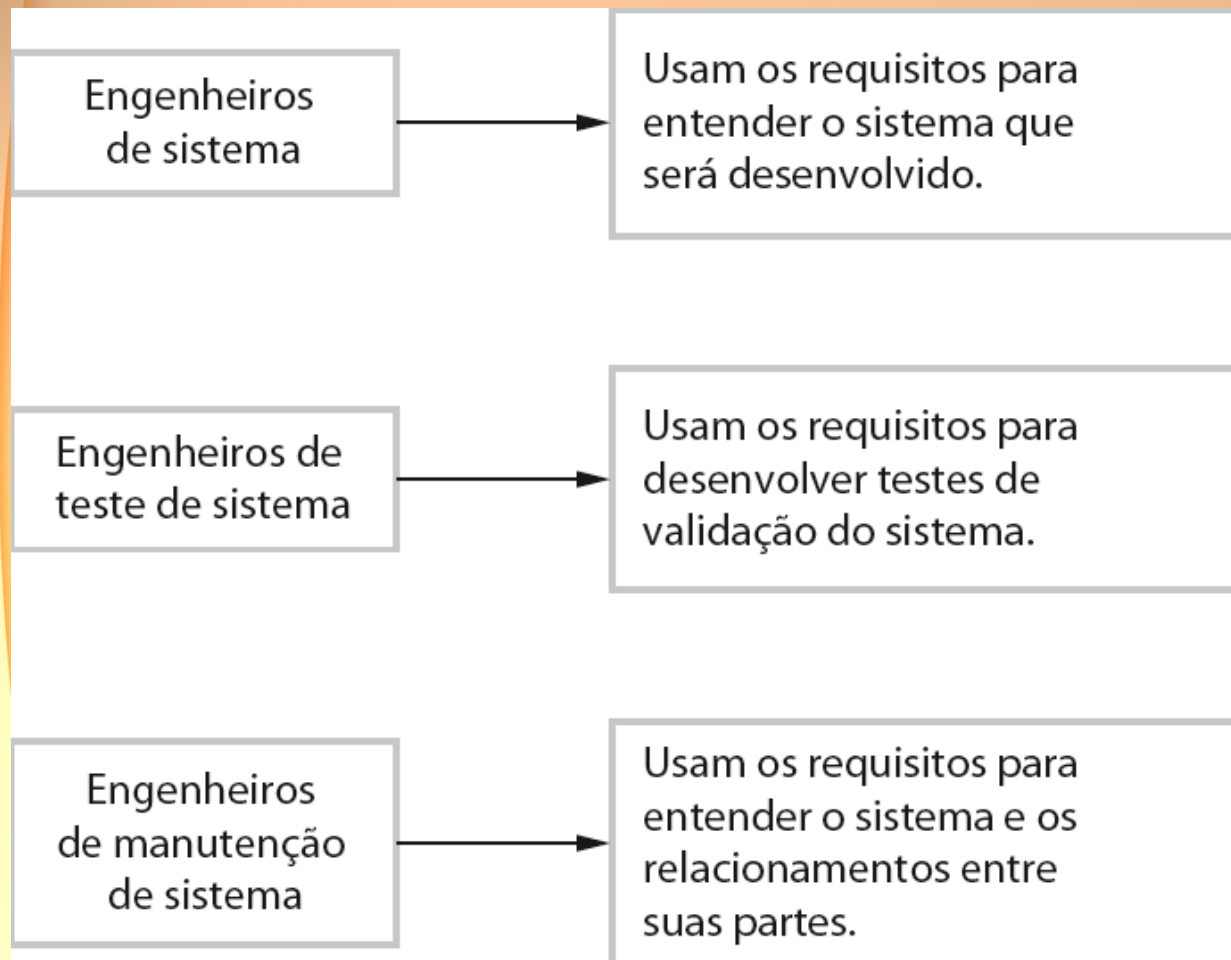
# engenaria de SOFTWARE

## Usuários de um documento de requisitos



# engenaria de SOFTWARE

## Usuários de um documento de requisitos





# Variabilidade do documento de requisitos

# engenharia de SOFTWARE

- 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.
- 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.

# A estrutura de um documento de requisitos

# engenharia de SOFTWARE

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.

# A estrutura de um documento de requisitos

# engenharia de SOFTWARE

Capítulo	Descrição
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.
- 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.

# Formas de escrever uma especificação de requisitos de sistema

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

- 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
  - ✓ 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;
  - ✓ 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 completos quanto possível.



- 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 para escrever requisitos

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

## Problemas com a 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 de requisitos para o sistema de software de bomba de insulina

# engenharia de SOFTWARE

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.)*

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

# engenharia de SOFTWARE

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



# Uma especificação estruturada de um requisito para uma bomba de insulina

## Bomba de insulina/Software de controle/SRS/3.3.2

<b>Função</b>	Calcula doses de insulina: nível seguro de açúcar.
<b>Descrição</b>	Calcula a dose de insulina a ser fornecida quando o nível de açúcar está na zona de segurança entre três e sete unidades.
<b>Entradas</b>	Leitura atual de açúcar (r2), duas leituras anteriores (r0 e r1).
<b>Fonte</b>	Leitura atual da taxa de açúcar pelo sensor. Outras leituras da memória.
<b>Saídas</b>	CompDose — a dose de insulina a ser fornecida.
<b>Destino</b>	Loop principal de controle.
<b>Ação</b>	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.
<b>Requisitos</b>	Duas leituras anteriores, de modo que a taxa de variação do nível de açúcar pode ser calculada.
<b>Pré-condição</b>	O reservatório de insulina contém, no mínimo, o máximo de dose única permitida de insulina.
<b>Pós- -condições</b>	r0 é substituída por r1 e r1 é substituída por r2.
<b>Efeitos colaterais</b>	Nenhum.

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

# Especificação tabular de processamento para uma bomba de insulina

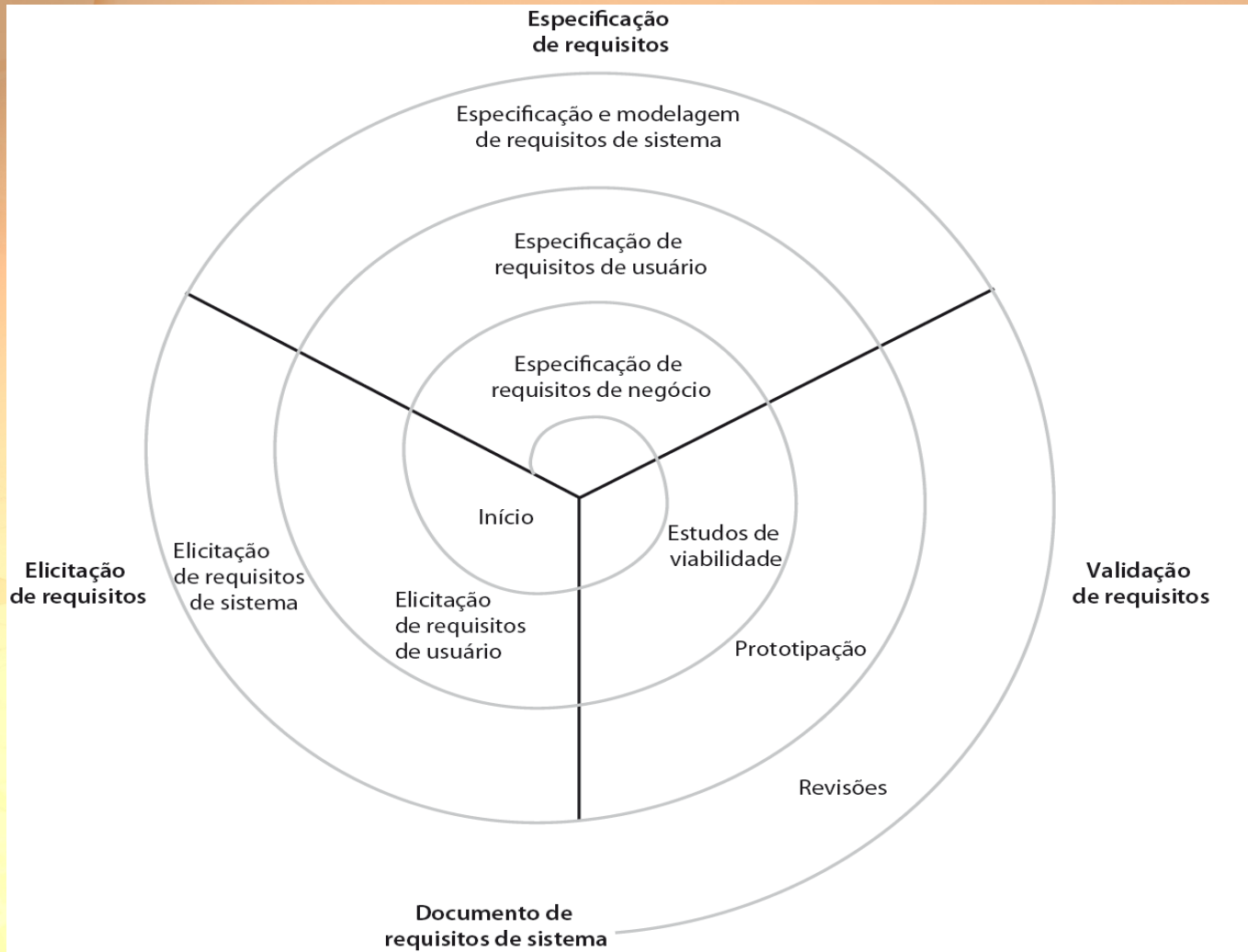
Condição	Ação
Nível de açúcar diminuindo ( $r_2 < r_1$ )	CompDose = 0
Nível de açúcar estável ( $r_2 = r_1$ )	CompDose = 0
Nível de açúcar aumentando e a taxa de aumento decrescente [ $(r_2 - r_1) < (r_1 - r_0)$ ]	CompDose = 0
Nível de açúcar aumentando e a taxa de aumento estável ou crescente [ $(r_2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



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

# Uma visão em espiral do processo de engenharia de requisitos

# engenharia de SOFTWARE



## Elicitação e análise de requisitos

# engenharia de SOFTWARE

- Às vezes chamada de elicitación ou descoberta de requisitos.
- Envolve técnicos trabalhando com os clientes para levantar dados sobre o domínio da aplicação, os serviços que o sistema deve fornecer e as restrições operacionais do sistema.
- Pode envolver usuários finais, gerentes, engenheiros envolvidos na manutenção, especialistas de domínio, sindicatos, etc.
- Esses são chamados *stakeholders*.



## Elicitação e análise de requisitos

- Engenheiros de software trabalham com uma gama de *stakeholders* do sistema para descobrir sobre o domínio da aplicação, os serviços que o sistema deve fornecer, o desempenho do sistema necessários, restrições de hardware, outros sistemas, etc.
- Estágios incluem:
  - ✓ Descoberta de requisitos,
  - ✓ Classificação e organização de requisitos,
  - ✓ Priorização e negociação de requisitos,
  - ✓ Especificação de requisitos.

# O processo de elicitação e análise de requisitos

## engenharia de SOFTWARE



## Problemas de análise de requisitos

- Os *stakeholders* não sabem o que realmente querem.
- Os *stakeholders* expressam requisitos em seus próprios termos.
- Diferentes *stakeholders* podem ter requisitos conflitantes.
- Fatores políticos e organizacionais podem influenciar os requisitos de sistema.
- Os requisitos mudam durante o processo de análise. Novos *stakeholders* podem surgir e o ambiente de negócios pode mudar.



## Pontos importantes

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

- O processo de coleta de informações sobre os sistemas necessários e os existentes, e separar os requisitos do usuário e sistema dessas informações.
- A interação é com os *stakeholders* do sistema desde os gerentes até os reguladores externos.
- Normalmente, os sistemas têm vários *stakeholders*.

- Pacientes cujas informações são registradas no sistema.
- Médicos que são responsáveis por avaliar e tratar os pacientes.
- Enfermeiros que coordenam as consultas com médicos e administram alguns tratamentos.
- Recepcionistas dos médicos que gerenciam as consultas dos pacientes.
- A equipe de TI responsável pela instalação e manutenção do sistema.



- Um gerente de ética médica, que deve garantir que o atual sistema atenda às diretrizes éticas para o cuidado do paciente.
- Gerentes de cuidados de saúde que obtiverem informações de gerenciamento do sistema.
- Registros médicos, equipes responsáveis por garantir que as informações do sistema possam ser mantidas e preservadas, e que a manutenção de registros foi executada corretamente.

- Entrevistas formais ou informais com os *stakeholders* fazem parte da maioria dos processos de engenharia de requisitos.
- Tipos de entrevista
  - ✓ Entrevistas fechadas com base em uma lista de perguntas pré-determinada.
  - ✓ Entrevistas abertas, em que várias questões são exploradas com os *stakeholders*.
- Entrevistar eficazmente
  - ✓ Ter a mente aberta, evitar ideias pré-concebidas sobre os requisitos e estar disposto a ouvir os *stakeholders*.
  - ✓ Induzir os entrevistados a discutir usando uma questão trampoline, uma proposta de requisitos, ou trabalhando em conjunto em um sistema protótipo.

## Entrevistas, na prática

# engenharia de SOFTWARE

- Normalmente, uma mistura de entrevistas fechadas e abertas.
- Entrevistas são boas para a obtenção de um entendimento geral do que os *stakeholders* fazem e como eles podem interagir com o sistema.
- Entrevistas não são boas para a compreensão dos requisitos de domínio:
  - ✓ Engenheiros de requisitos não podem entender a terminologia específica de domínio;
  - ✓ Algum conhecimento de domínio é tão familiar que as pessoas acham difícil articular ou pensam que não vale a pena articular.



## Cenários

- Cenários são exemplos da vida real de como um sistema pode ser usado.
- Eles devem incluir:
  - ✓ A descrição da situação inicial;
  - ✓ A descrição do fluxo normal de eventos;
  - ✓ A descrição do que pode dar errado;
  - ✓ Informações sobre outras atividades concorrentes;
  - ✓ A descrição do estado do sistema quando o cenário acaba.

# Cenário para a coleta do histórico médico em MHC-PMS

## engenharia de SOFTWARE

### **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).

# Cenário para a coleta do histórico médico em MHC-PMS

## engenharia de SOFTWARE

### **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:**

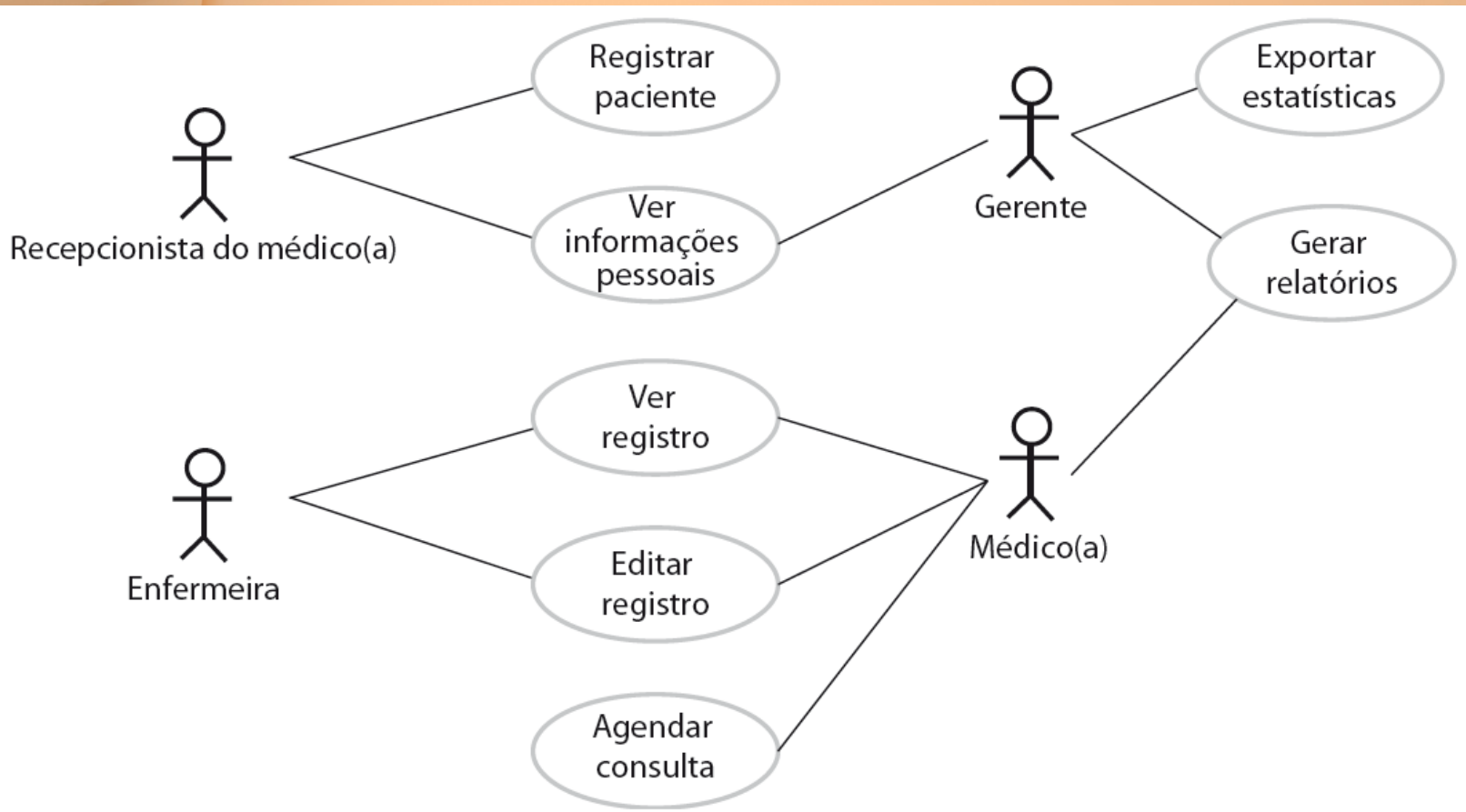
O usuário está conectado. O prontuário do paciente, incluindo seu histórico médico, é inserido no banco de dados e um registro é adicionado ao *log* do sistema, mostrando o tempo de início e fim da sessão e a enfermeira envolvida.



- Casos de uso é uma técnica da UML baseada em cenários que identificam os atores em uma interação e que descreve a interação em si.
- 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.
- Diagramas de sequência podem ser usados para adicionar detalhes aos casos de uso, mostrando a sequência de processamento de eventos no sistema.

## Casos de uso para o MHC-PMS

# engenharia de SOFTWARE



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

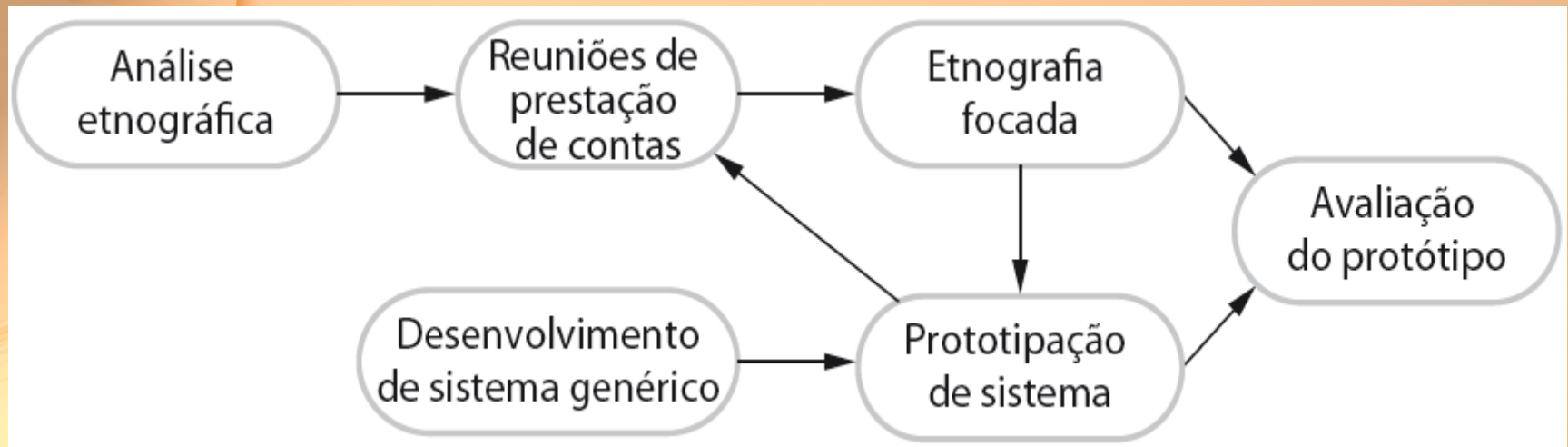


- Requisitos que são derivados da maneira como as pessoas realmente trabalham e não da maneira como as definições de processo sugerem que elas deveriam trabalhar.
- Requisitos que são derivados da cooperação e conscientização das atividades das outras pessoas.
  - ✓ Consciência do que outras pessoas estão fazendo leva a mudanças no modo como fazemos as coisas.
- A etnografia é eficaz para a compreensão dos processos existentes, mas não pode identificar novos recursos que devem ser adicionados a um sistema.

- Desenvolvida em um projeto de estudo do processo de controle do tráfego aéreo.
- Combina etnografia com prototipação.
- O desenvolvimento de protótipos resultou em questões sem respostas, as quais se centram na análise etnográfica.
- O problema com a etnografia é que ela estuda as práticas existentes, as quais podem ter alguma base histórica que não continua sendo relevante.

# Etnografia e prototipação para análise de requisitos

## engenharia de SOFTWARE





- Preocupados em demonstrar se os requisitos definem o sistema que o cliente realmente quer.
- Os custos de erros de requisitos são altos, logo, a validação é muito importante.
  - ✓ Corrigir um erro de requisitos após a entrega pode custar até 100 vezes o custo de corrigir um erro de execução.

## Verificação de requisitos

- Validade. O sistema fornece as funções que melhor atendem às necessidades do cliente?
- Consistência. Existe algum conflito de requisitos?
- Completude. Estão incluídas todas as funções e restrições requeridas pelo cliente ?
- Realismo. Os requisitos podem ser implementados com o orçamento e a tecnologia disponíveis?
- Verificabilidade. Os requisitos podem ser verificados?

- Revisões de requisitos
  - ✓ Análise manual sistemática dos requisitos.
- Prototipação
  - ✓ Usando um modelo executável do sistema para verificar os requisitos.
- Geração de casos de teste
  - ✓ Desenvolvimento de testes para verificar os requisitos implementados.



- Revisões periódicas devem ser feitas enquanto a definição dos requisitos está sendo formulada.
- Ambos, cliente e fornecedor, devem ser envolvidos nas revisões.
- Os comentários podem ser formais (com documentos completos) ou informais.
- Uma boa comunicação entre os desenvolvedores, clientes e usuários pode resolver os problemas numa fase inicial.

## Avaliação da revisão

- Verificabilidade
  - ✓ A exigência é realmente testável?
- Compreensibilidade
  - ✓ O requisito é adequadamente compreendido?
- Rastreabilidade
  - ✓ A origem do requisito é clara?
- Adaptabilidade
  - ✓ O requisito pode ser alterado sem causar um grande impacto sobre outros requisitos?

- Gerenciamento de requisitos é o processo de gerenciar os requisitos em constante mudança durante o processo de engenharia de requisitos e desenvolvimento de sistemas.
- Após o sistemas começar a ser usado, surgem novos requisitos.
- 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.
- É necessário estabelecer um processo formal para fazer propostas de mudança e ligar essas aos requisitos de sistema.



## Mudanças nos requisitos

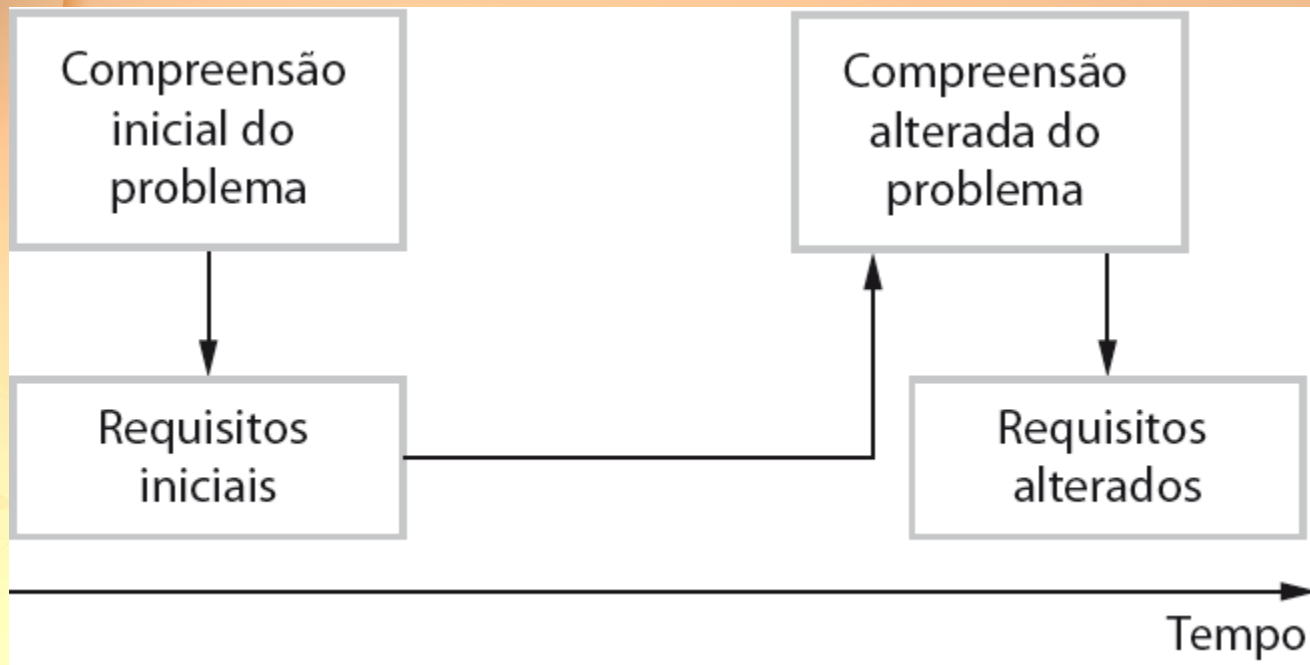
- O ambiente técnico e de negócios do sistema sempre muda após a instalação.
  - ✓ Um novo hardware pode ser introduzido, pode ser necessário para a interface do sistema com outros sistemas, as prioridades do negócio podem mudar (com as consequentes alterações no sistema de apoio necessário) e, podem ser que o sistema deve, necessariamente, respeitar.
- As pessoas que pagam por um sistema e os usuários desse sistema raramente são as mesmas pessoas.
  - ✓ Clientes do sistema impõem requisitos devido a restrições orçamentais e organizacionais. Esses podem entrar em conflito com os requisitos do usuário final e, após a entrega, pode ser necessário adicionar novos recursos para suporte ao usuário, caso o sistema seja para atender a seus objetivos.

## Mudanças nos requisitos

- Sistemas de grande porte costumam ter uma comunidade de usuários diversos, com muitos usuários tendo necessidades diferentes e prioridades que podem ser conflitantes ou contraditórias.
- ✓ Os requisitos do sistema final são, inevitavelmente, um compromisso entre eles e, a experiência mostra que, muitas vezes se descobre que o balanço de apoio dado aos diferentes usuários precisa ser mudado.

## Evolução dos requisitos

# engenharia de SOFTWARE





# Planejamento de gerenciamento de requisitos

# engenharia de SOFTWARE

- Estabelece o nível de detalhamento necessário para o gerenciamento de requisitos. Decisões do gerenciamento de requisitos:
  - ✓ *Identificação de requisitos.* Cada requisito deve ser identificado exclusivamente para que ele possa ser comparado com outros requisitos.
  - ✓ *Processo de gerenciamento de mudanças.* Esse é o conjunto de atividades que avaliam o impacto e o custo das mudanças. Esse processo é discutido em mais detalhes na seção seguinte.
  - ✓ *Políticas de rastreabilidade.* Essas políticas definem as relações entre cada requisito e entre os requisitos e o projeto do sistema que deve ser registrado.
  - ✓ *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

## engenharia de SOFTWARE

- Decidir se uma mudança de requisitos deve ser aceita.
- Análise de problema e especificação de mudanças
  - ✓ Durante essa fase, o problema ou a proposta de mudança é analisada para verificar se é válida. O feedback dessa análise é devolvido para o solicitante, que pode responder com uma proposta mais específica de mudança dos requisitos, ou decidir retirar o pedido.
- Análise de mudanças e custos
  - ✓ O efeito da mudança proposta é avaliado por meio de informações de rastreabilidade e conhecimento geral dos requisitos do sistema. Uma vez que essa análise é concluída, toma-se a decisão de prosseguir ou não com a mudança de requisitos.

# Gerenciamento de mudança de requisitos

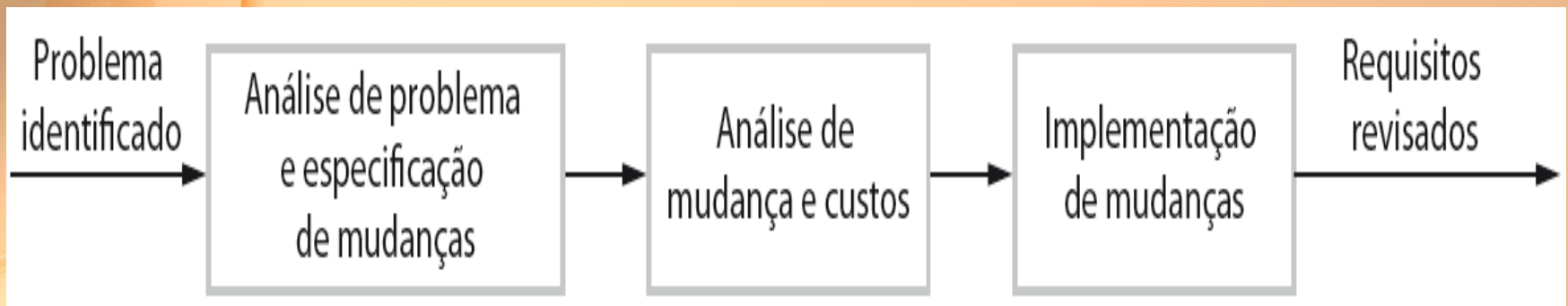
# engenharia de SOFTWARE

- Implementação de mudanças
  - ✓ O documento de requisitos e, se necessário, o projeto e implementação do sistema, são modificados. Idealmente, o documento deve ser organizado de modo que as mudanças possam ser facilmente implementadas.



# Gerenciamento de mudança de requisitos

# engenharia de SOFTWARE



## Pontos importantes

# engenharia de SOFTWARE

- Você pode usar uma variedade de técnicas para a elicitación de requisitos, incluindo entrevistas, cenários, casos de uso e etnografia.
- A validação dos requisitos é o processo de verificação da validade, consistência, completude, realismo e verificabilidade dos requisitos.
- Mudanças organizacionais e técnicas, e de negócios, inevitavelmente levam a mudanças nos requisitos de um sistema de software.
- O gerenciamento dos requisitos é o processo de gerenciamento e controle dessas mudanças.