Engenharia de Requisitos

- Tipos de Requisitos
- Documentos de Requisitos
- Processo de Engenharia de Requisitos

Objetivos

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

Referências Bibliográficas

- Básica:

  PRESSMAN, Roger S; MAXIM, Bruce R. Engenharia de software: uma abordagem profissional. 9 ed. Porto Alegre AMGH 2021 1 recurso online.

  SOMMERVILLE, Ian. Engenharia de software. 10 ed. São Paulo. Cap. 4: Engenharia de Requisitos. São Paulo: Pearson, 2019 recurso online.

- Complementar:

  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

Problemas dos Processos de Software (1/2)

- Alterações de metas e estratégias organizacionais
  - Os negócios requerem ciclos de desenvolvimento mais curtos e dinâmicos
  - Requisitos iniciais fracamente definidos
- 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 (2/2)

- 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"

Modelagem de Sistemas de Informação

- Por que modelar?
  - Avaliação dos riscos
  - Entendimento do Problema
  - Definição do problema
  - Gerenciamento do projeto

Por onde começar a modelagem?

Requisitos (SOMMERVILLE, 2019)

- 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
- 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

Requisitos de Usuários (SOMMERVILLE, 2019)

- 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 de Sistemas (SOMMERVILLE, 2019)

- Definem, detalhadamente, as funções, os serviços e as restrições operacionais do sistema
- Devem ser precisos e definir exatamente o que será implementado
- 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, 2019)

FIGURA 4.1 Requisitos de usuário e requisitos de sistema.

Definição de requisitos de usuário

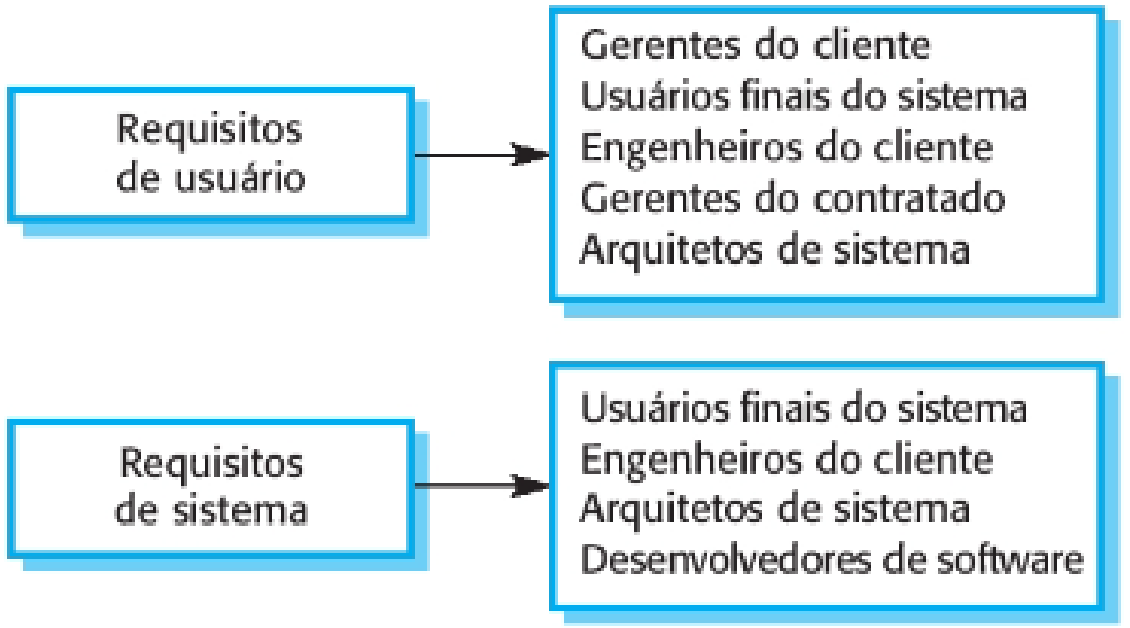
1. O sistema Mentcare deve gerar relatórios de gestão mensais, mostrando o custo dos medicamentos prescritos por cada clínica naquele mês.

Especificação dos requisitos de sistema

1.1 No último dia útil de cada mês, deve ser gerado um resumo dos medicamentos prescritos, seu custo e a clínica que os prescreveu.
1.2 O sistema deve gerar o relatório para impressão após as 17h30 do último dia útil do mês.
1.3 Deve ser criado um relatório para cada clínica, listando o nome de cada medicamento, a quantidade total de prescrições, a quantidade de doses prescritas e o custo total dos medicamentos prescritos.
1.4 Se os medicamentos estiverem disponíveis em dosagens diferentes (por exemplo, 10 mg, 20 mg etc.) devem ser criados relatórios diferentes para cada dosagem.
1.5 O acesso aos relatórios de medicamentos deve ser restrito aos usuários autorizados, conforme uma lista de controle de acesso produzida pela gestão.

Leitores de requisitos (SOMMERVILLE, 2019)

FIGURA 4.2 Leitores dos diferentes tipos de especificação de requisitos.



Requisitos Funcionais (SOMMERVILLE, 2019)

- 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, 2019)

- Devem descrever as funcionalidades ou os serviços do sistema
- Dependem do tipo de software, possíveis usuários e do tipo de domínio onde 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 do Mentcare (SOMMERVILLE, 2019)

1. Um usuário deve poder fazer uma busca na lista de consultas de todas as clínicas
2. O sistema deve gerar, a cada dia, para cada clínica, uma lista de pacientes que devam comparecer às consultas naquele dia
3. Cada membro da equipe que utiliza o sistema deve ser identificado exclusivamente por seu número de funcionário de oito dígitos

Requisitos Funcionais (SOMMERVILLE, 2019)

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

Requisitos Funcionais (SOMMERVILLE, 2019)

- Considere o termo “busca” 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, 2019)

- 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 Não-Funcionais (SOMMERVILLE, 2019)

- São restrições e/ou condições 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, 2019)

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

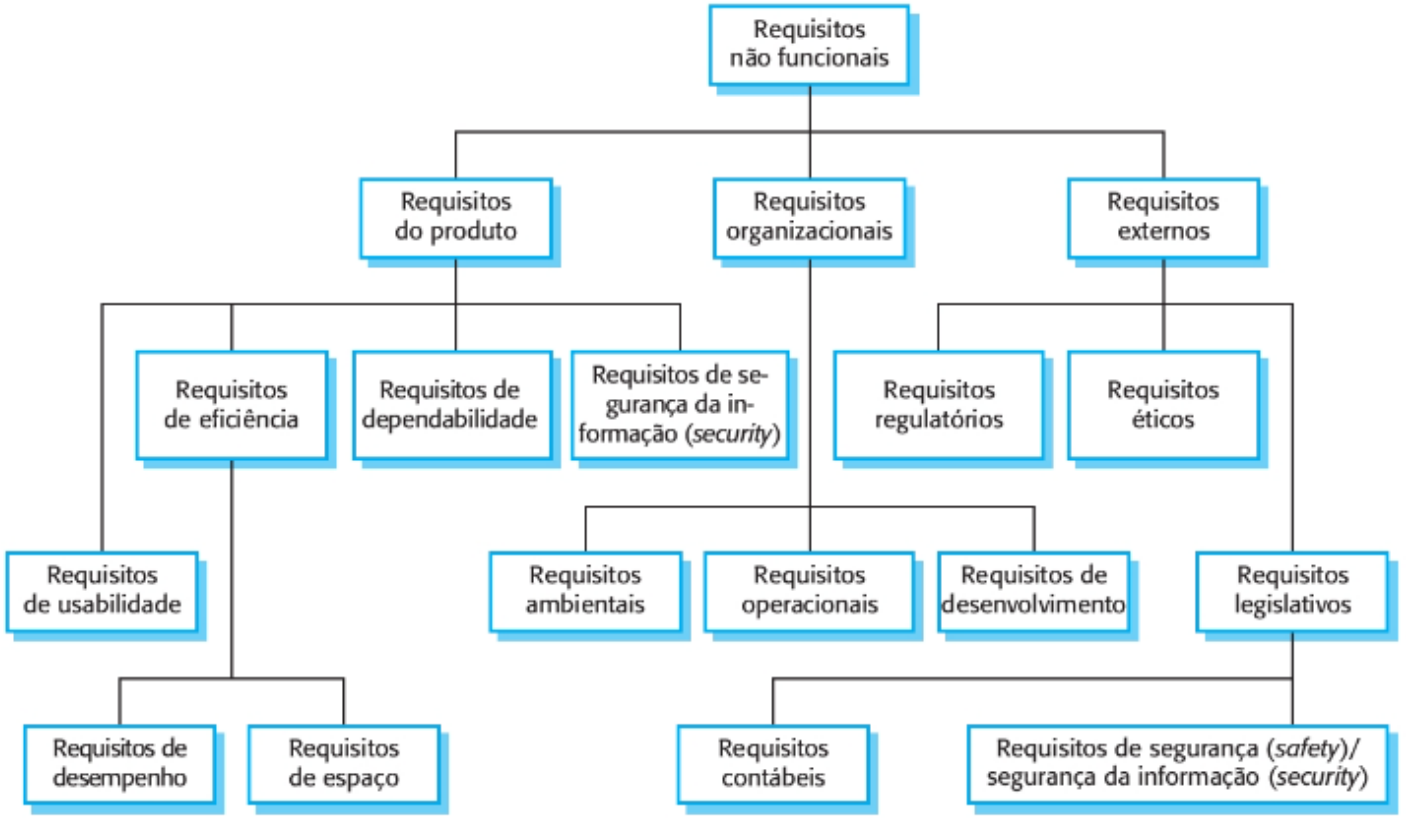
  Por exemplo, para assegurar que os requisitos de desempenho sejam cumpridos, você pode ter que organizar o sistema para minimizar a comunicação entre os componentes

Requisitos Não-Funcionais (SOMMERVILLE, 2019)

- 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 (SOMMERVILLE, 2019)

FIGURA 4.3 Tipos de requisitos não funcionais.



Requisitos Não-Funcionais do Mentcare (SOMMERVILLE, 2019)

FIGURA 4.4 Exemplos de possíveis requisitos não funcionais do sistema Mentcare.

Requisito do produto

O sistema Mentcare deve ficar disponível para todas as clínicas durante o expediente normal (segunda-sexta, 8h30-17h30).

O tempo que o sistema pode permanecer fora do ar no expediente normal não deve ultrapassar 5 segundos em qualquer dia.

Requisito organizacional

Os usuários do sistema Mentcare devem se identificar usando o cartão de identificação de autoridade de saúde.

Requisito externo

O sistema deve implementar providências para a privacidade do paciente, conforme estabelecido em HStan-03-2006-priv.

Requisitos Não-Funcionais (SOMMERVILLE, 2019)

- 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 Metacare (SOMMERVILLE, 2019)

1. O sistema deve ser de fácil uso pelo pessoal médico e ser organizado de tal modo que os erros dos usuários sejam minimizados. (Meta)
2. A equipe médica deve ser capaz de usar todas as funções do sistema depois de duas horas de treinamento. Após essa etapa, a quantidade média de erros cometidos pelos usuários experientes não deve ultrapassar dois por hora de uso do sistema. (Requisito não-funcional testável)

Métricas para Requisitos Não-Funcionais (SOMMERVILLE, 2019)

FIGURA 4.5 Métricas para especificar requisitos não funcionais.

Propriedade	Métrica
Velocidade	Transações processadas/segundo Tempo de resposta do usuário/evento Tempo de atualização da tela
Tamanho	Megabytes/número de chips de ROM
Facilidade de uso	Tempo de treinamento Número de quadros de ajuda
Confiabilidade	Tempo médio até a falha Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo para reiniciar após a falha Porcentagem de eventos causando falhas Probabilidade de corromper dados em uma falha
Portabilidade	Porcentagem de declarações dependentes do sistema-alvo Número de sistemas-alvo

Requisitos de Domínio (SOMMERVILLE, 2019)

- São provenientes do domínio da aplicação e refletem as características e as restrições desse domínio
- Podem ser funcionais ou não-funcionais
- 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, 2019)

- 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, 2019)

- Exemplo do Sistema de Segurança de Trem
  - A desaceleração do trem deve ser computada como:

    Dtrain = Dcontrol + Dgradient

    onde Dgradient é 9.81ms2 * gradiente / alfa compensado e onde os valores de 9.81ms2 / alpha são conhecidos para diferentes tipos de trem

Requisitos de Domínio (SOMMERVILLE, 2019)

- Compreensibilidade
  - Requisitos são expressos na linguagem do domínio da aplicação
  - Isto pode não ser 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 (1/2)

- Os requisitos para um sistema de software estabelecem o que o sistema deve fazer e definem 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

Especificação de Requisitos

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

Projeto e Requisitos (1/2)

- 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

Projeto e Requisitos (2/2)

- 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 completo quanto possível

Formas de escrever (SOMMERVILLE, 2019)

FIGURA 4.11 Notações para escrever requisitos de sistema.

Notação	Descrição
Sentenças em linguagem natural	Os requisitos são escritos usando frases numeradas em linguagem natural. Cada frase deve expressar um requisito.
Linguagem natural estruturada	Os requisitos são escritos em linguagem natural em um formulário ou *template*. Cada campo fornece informações sobre um aspecto do requisito.
Notações gráficas	Modelos gráficos, suplementados por anotações em texto, são utilizados para definir os requisitos funcionais do sistema. São utilizados com frequência os diagramas de casos de uso e de sequência da UML.
Especificações matemáticas	Essas notações se baseiam em conceitos matemáticos como as máquinas de estados finitos ou conjuntos. Embora essas especificações inequívocas possam reduzir a ambiguidade em um documento de requisitos, a maioria dos clientes não compreende uma especificação formal. Eles não conseguem averiguar se ela representa o que desejam e relutam em aceitar essa especificação como um contrato do sistema (discutirei essa abordagem no Capítulo 10, que aborda a dependabilidade do sistema).

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

- 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
- 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, 2019)

FIGURA 4.12 Exemplos de requisitos do sistema de software da bomba de insulina.

3.2 O sistema deve medir o nível de açúcar no sangue e fornecer insulina, se for necessário, a cada 10 minutos. (*As variações do açúcar no sangue são relativamente lentas, então é desnecessário medir com uma frequência maior; a medição menos frequente poderia levar a níveis de açúcar sanguíneo desnecessariamente elevados.*)

3.6 O sistema deve executar uma rotina de autoteste a cada minuto com as condições a serem testadas e as ações associadas, definidas na Tabela 1 do documento de requisitos. (*Uma rotina de autoteste pode descobrir problemas de hardware e software e alertar o usuário de que a operação normal pode ser impossível.*)

Especificações estruturadas

- 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 em formulários

- Definição da função ou entidade
- Descrição de entradas e de onde eles vêm
- Descrição das saídas e para onde irão
- Informações necessárias para o processamento e outras entidades usadas
- Descrição da ação a ser tomada
- Pré e pós condições (se for o caso)
- Os efeitos colaterais (se houver) da operação

Outro exemplo (SOMMERVILLE, 2019)

FIGURA 4.13 Especificação estruturada de um requisito para uma bomba de insulina.

Bomba de insulina/Software de controle/SRS/3.3.2	
Função	Computar a dose de insulina: nível de açúcar seguro.
Descrição	Computa a dose de insulina a ser fornecida quando o nível de açúcar atual estiver na zona segura entre 3 e 7 unidades.
Entradas	Leitura atual do açúcar (r2), as duas leituras prévias (r0 e r1).
Fonte	Leitura atual de açúcar do sensor. Outras leituras da memória.
Saídas	DoseComp – a dose de insulina a ser fornecida.
Destino	Laço de controle principal.
Ação	DoseComp é igual a zero se o nível de açúcar estiver estável ou caindo; ou se o nível estiver aumentando, mas a taxa de crescimento estiver diminuindo. Se o nível estiver aumentando e a taxa de crescimento também, então a DoseComp é obtida pela divisão por 4 da diferença entre o nível de açúcar atual e o nível anterior, arredondando o resultado. Se o resultado for arredondado para zero, então a DoseComp é definida como dose mínima que pode ser fornecida (ver Figura 4.14).
Requer	Duas leituras prévias para que a taxa de variação do nível de açúcar possa ser calculada.
Pré-condição	O reservatório de insulina contém pelo menos a dose máxima permitida.
Pós-condição	r0 é substituída por r1, então r1 é substituída por r2.
Efeitos colaterais	Nenhum.

Especificação tabular

- Normalmente é usada 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

Exemplo (SOMMERVILLE, 2019)

FIGURA 4.14 Especificação tabular do cálculo em uma bomba de insulina.

Condição	Ação
Nível de açúcar em queda (r2 < r1)	DoseComp = 0
Nível de açúcar estável (r2 = r1)	DoseComp = 0
Nível de açúcar em alta e taxa de crescimento em queda ((r2 – r1) < (r1 – r0))	DoseComp = 0
Nível de açúcar em alta e taxa de crescimento estável ou em alta r2 > r1 & ((r2 – r1) ≥ (r1 – r0))	DoseComp = arredondar ((r2 – r1) / 4) Se resultado arredondado = 0, então DoseComp = DoseMínima

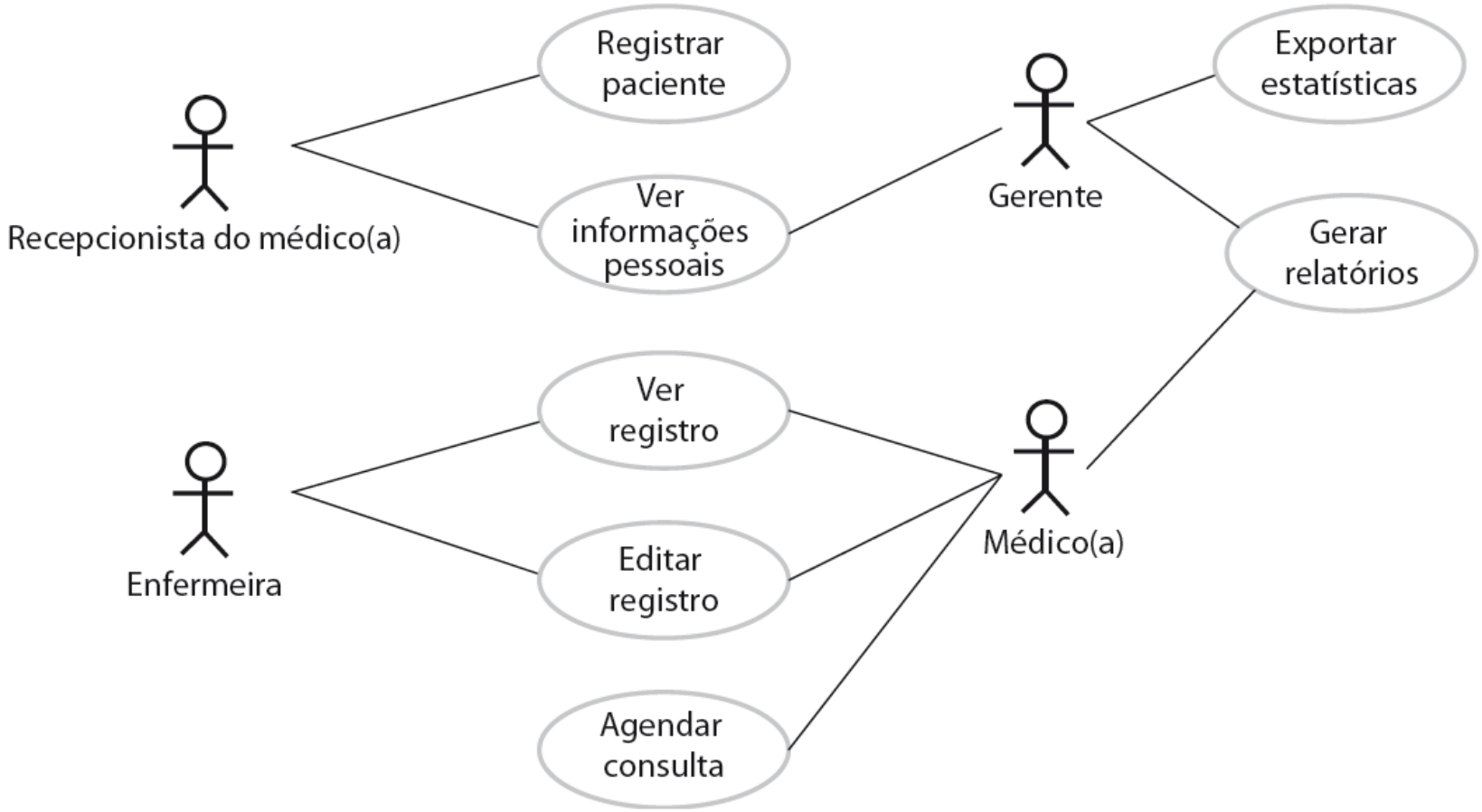
Cenários

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

Casos de Uso

- 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
- Um conjunto de casos de uso deve descrever todas as possíveis interações dos atores com o sistema
- Modelo gráfico de alto nível complementado por um algoritmo que representa cada interação e também por outros diagramas (diagramas de atividades e diagrama de sequencia)

Diagrama de Casos de Uso



Documento de Requisitos

Documento de Requisitos

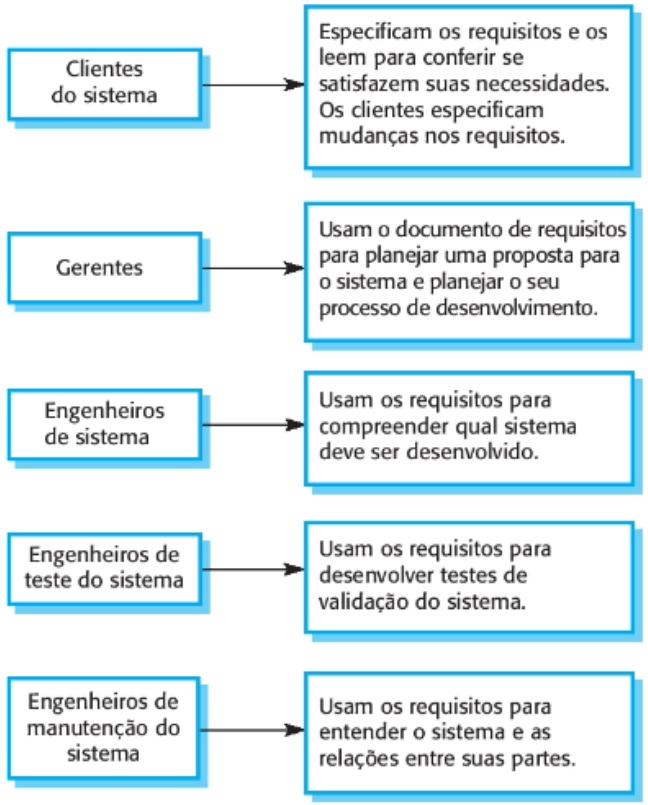
- O documento de requisitos de software é a declaração oficial do que é demandado dos desenvolvedores do sistema
- Deve incluir uma definição de requisitos do usuário e uma especificação de requisitos do sistema
- Não necessariamente é um documento de projeto
- Na medida do possível, deve definir O QUE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

Requisitos e Métodos Ágeis

- A produção de um documento de requisitos pode ser considerada um desperdício de tempo, pois, esses mudam rapidamente e o documento estará sempre desatualizado
- Métodos ágeis, tais como XP usam a engenharia de requisitos incrementais e expressam os requisitos como “histórias de usuário”
- Isto é problemático para sistemas que exigem várias análises pré-entrega (por exemplo, sistemas críticos) ou sistemas desenvolvidos por várias equipes

Usuários de Documento de Requisitos (SOMMERVILLE, 2019)

FIGURA 4.16 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
- 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 do Documento de Requisitos (SOMMERVILLE, 2019)

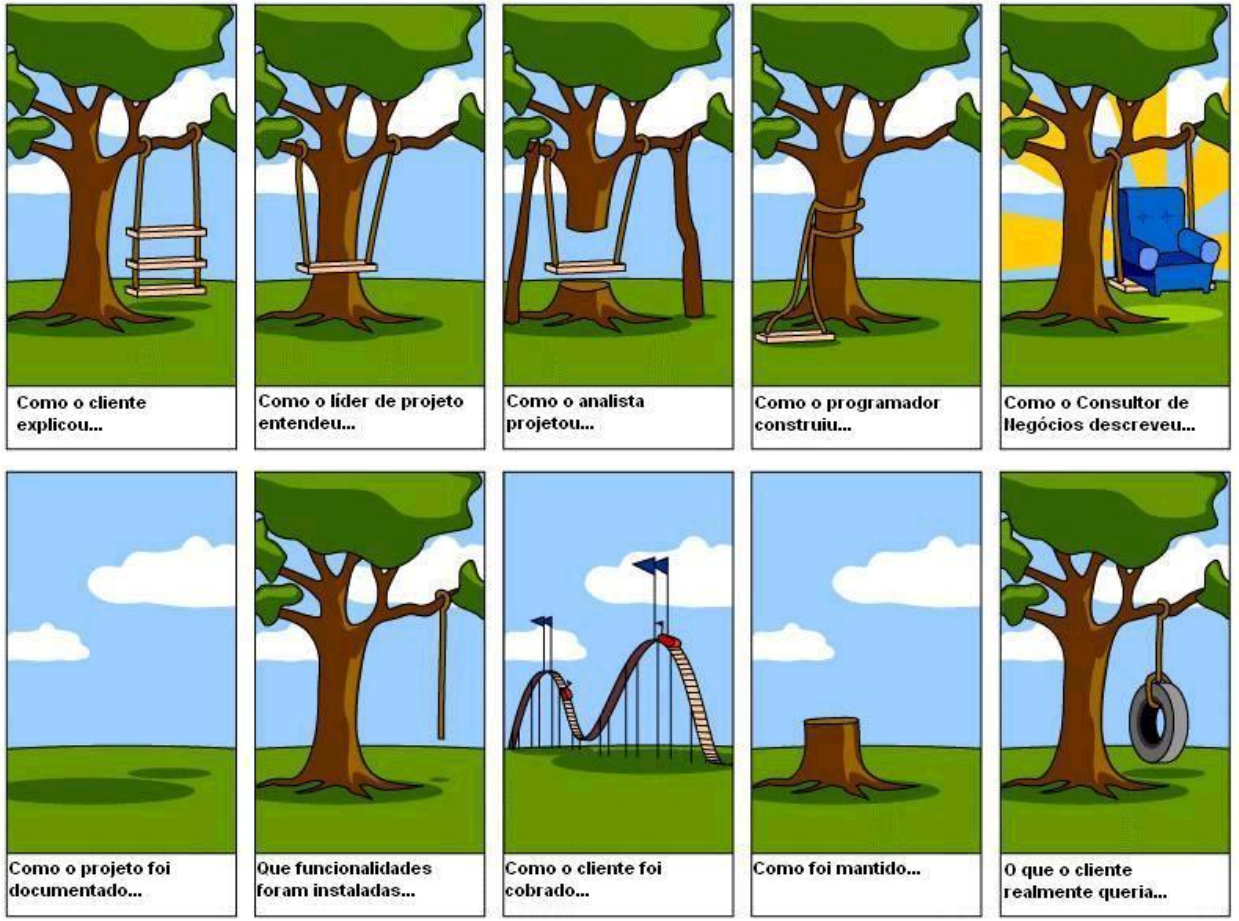
FIGURA 4.17 Estrutura de um documento de requisitos.

Capítulo	Descrição
Prefácio	Define o público-alvo do documento e descreve seu histórico de versões, incluindo a fundamentação para a criação de uma nova versão e um resumo das mudanças feitas em cada uma.
Introdução	Descreve a necessidade do sistema. Deve descrever resumidamente as funções do sistema e explicar como ele vai trabalhar com outros sistemas. Também precisa descrever como o sistema se encaixa nos objetivos de negócio gerais ou estratégicos da organização que contratou o software.
Glossário	Define os termos técnicos utilizados no documento. Deve-se evitar fazer pressupostos sobre a experiência ou a especialização do leitor.
Definição dos requisitos de usuário	Descreve os serviços fornecidos para o usuário. Os requisitos não funcionais do sistema também devem ser descritos nesta seção. Essa descrição pode usar linguagem natural, diagramas ou outras notações compreensíveis para os clientes. Os padrões de produto e processo que devem ser seguidos têm de ser especificados.
Arquitetura do sistema	Esse capítulo apresenta uma visão geral e de alto nível da arquitetura prevista para o sistema, mostrando a distribuição das funções pelos módulos do sistema. Os componentes de arquitetura reusados devem ser destacados.
Especificação dos requisitos de sistema	Descreve os requisitos funcionais e não funcionais em mais detalhes. Se for necessário, mais detalhes também são acrescentados aos requisitos não funcionais. Podem ser definidas interfaces com outros sistemas.
Modelos do sistema	Esse capítulo inclui modelos gráficos do sistema, mostrando as relações entre os componentes do sistema e entre o sistema e seu ambiente. Exemplos possíveis são os modelos de objeto, modelos de fluxo de dados ou modelos semânticos de dados.
Evolução do sistema	Descreve os pressupostos fundamentais nos quais o sistema se baseia e quaisquer mudanças previstas em virtude da evolução do hardware, da mudança nas necessidades dos usuários etc. Essa seção é útil para os projetistas do sistema, já que pode ajudá-los a evitar decisões de projeto que restringiriam futuras mudanças prováveis no sistema.
Apêndices	Fornecem informações específicas, detalhadas, relacionadas à aplicação que está sendo desenvolvida — por exemplo, descrições de hardware e banco de dados. Os requisitos de hardware definem as configurações mínima e ideal do sistema; os requisitos de banco de dados definem a organização lógica dos dados utilizados pelo sistema e seus relacionamentos.
Índice	Vários índices para o documento podem ser incluídos, bem como índice alfabético normal, índice de diagramas, índice de funções etc.

Engenharia de Requisitos

Características dos bons requisitos

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



Engenharia de Requisitos (SOMMERVILLE, 2019)

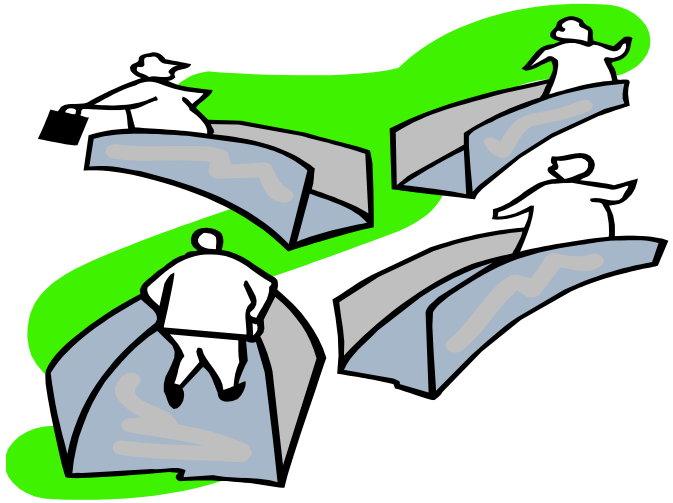
- Processo de descobrir, analisar, documentar e verificar os serviços e as restrições de um sistema, ou seja, os seus requisitos
- A engenharia de requisitos se concentra em problemas e objetivos do mundo real para definir funções e restrições dos sistemas de software

Porque a definição de requisitos é importante?

- Podem causar:
  - Perda de vidas
  - Prejuízos financeiros
  - Atrasos nas entregas
  - Aumento de riscos
  - Baixa qualidade

Engenharia de Requisitos (PRESSMAN, 2021)

- Principais etapas:
  - Concepção
  - Levantamento
  - Elaboração
  - 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? (1/2)

- Avaliar o negócio e a viabilidade técnica do sistema proposto
- Identificar as pessoas certas que participarão desse processo
- Definir o ambiente tecnológico necessário (visão geral)
- Identificar restrições do domínio do problema (características do negócio)

Como identificar os requisitos? (2/2)

- Escolher bem os métodos (entrevistas, JAD, reuniões, etc.)
- Envolver pessoas que podem ter pontos de vistas diferentes acerca do que está sendo identificado
- Criar cenários para facilitar a análise dos envolvidos no processo
- Identificar a existência de requisitos ambíguos

Elaboração e negociação de requisitos

- Questões que devem ser respondidas:
  - 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?
  - 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 de 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, 2019)

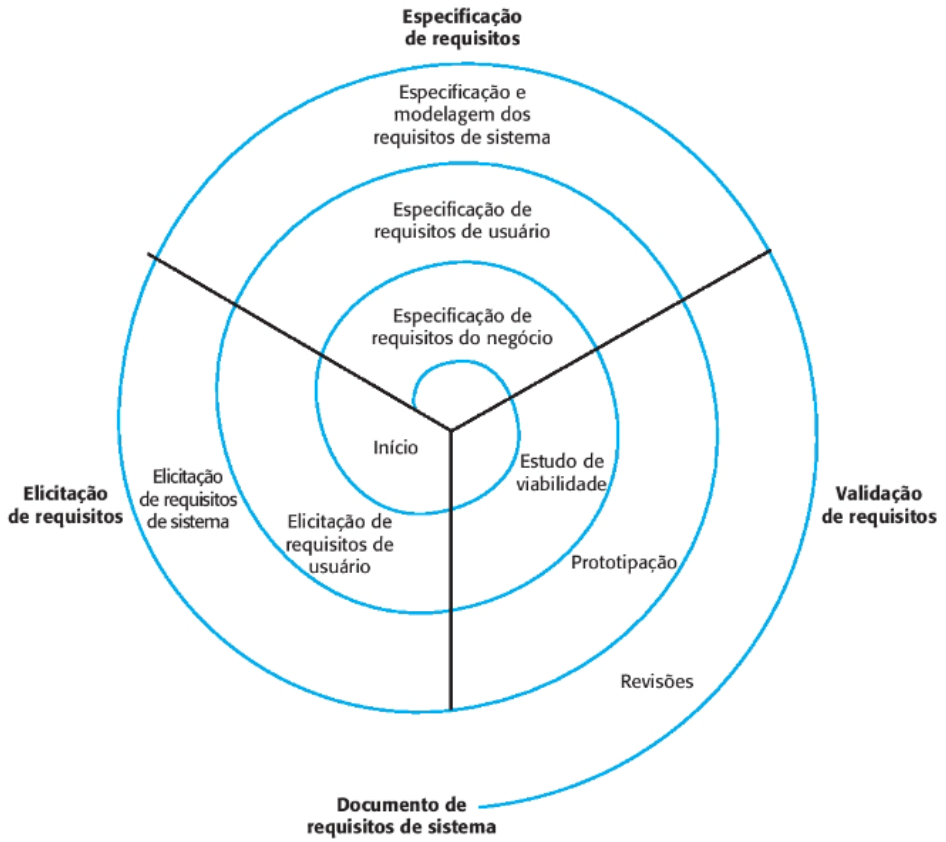
- Os processos usados para a engenharia de requisitos variam muito, dependendo do domínio da aplicação, das pessoas envolvidas e da organização que desenvolve os requisitos

Engenharia de Requisitos (SOMMERVILLE, 2019)

- 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

Processo de Engenharia de Requisitos (SOMMERVILLE, 2019)

FIGURA 4.6 Uma visão em espiral do processo de engenharia de requisitos.



Estudo de viabilidade

- Um estudo de viabilidade decide se vale a pena ou não gastar tempo e esforço com sistema proposto
- 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?
  - Quais serão os problemas de integração?
  - Tecnologia nova é necessária? Quais habilidades?
  - Quais recursos devem ser apoiados pelo sistema proposto?

Elicitação e Análise

- Algumas vezes chamada de elicitação 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
- Pode envolver usuários finais, gerentes, engenheiros envolvidos na manutenção, especialistas de domínio, representantes de sindicato, etc, ou seja, os *stakeholders*.

Espiral de Elicitação e Análise de Requisitos

FIGURA 4.7 O processo de elicitação e análise de requisitos.

Problemas de análise de requisitos

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

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

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

Pontos de vista

Ponto de vista é uma maneira de coletar e organizar um conjunto de requisitos de um grupo de *stakeholders* que têm algo em comum. Portanto, cada ponto de vista inclui um conjunto de requisitos de sistema. Os pontos de vista poderiam vir de usuários finais, gerentes ou outros, e ajudam a identificar as pessoas que podem fornecer informações sobre seus requisitos e estruturá-los para análise.

Entrevistas

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

- 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 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?*”

Etnografia

- 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

Âmbito da Etnografia

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

Validação de requisitos

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

- Conferência de validade: o sistema fornece as funções que melhor apoiam as necessidades do cliente?
- Conferência de consistência: existe algum tipo de conflito de requisitos?
- Conferência de completude: todas as funções requisitadas pelo cliente foram incluídas?
- Conferência de realismo: os requisitos podem ser implementados com o orçamento e a tecnologia disponíveis?
- Verificabilidade: os requisitos podem ser verificados?

Técnicas de validação de requisitos

- Revisões de requisitos
  - Análise manual sistemática dos requisitos
- Prototipação
  - Uso de um modelo executável do sistema para verificar requisitos
- Geração de casos de teste.
  - Desenvolvimento de testes para requisitos a fim de verificar a *testabilidade*

Revisões de requisitos

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

- Facilidade de verificação: o requisito é realisticamente testável?
- Facilidade de compreensão: o requisito é adequadamente compreendido?
- Rastreabilidade: a origem do requisito é claramente estabelecida?
- Adaptabilidade: o requisito pode ser mudado sem um grande impacto em outros requisitos?

Gerenciamento de requisitos (1/2)

- Gerenciamento de requisitos é o processo de gerenciamento de mudanças de requisitos durante o processo de engenharia de requisitos e o desenvolvimento de sistema
- 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 (2/2)

- 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

Mudança de requisitos

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

Planejamento de gerenciamento de requisitos

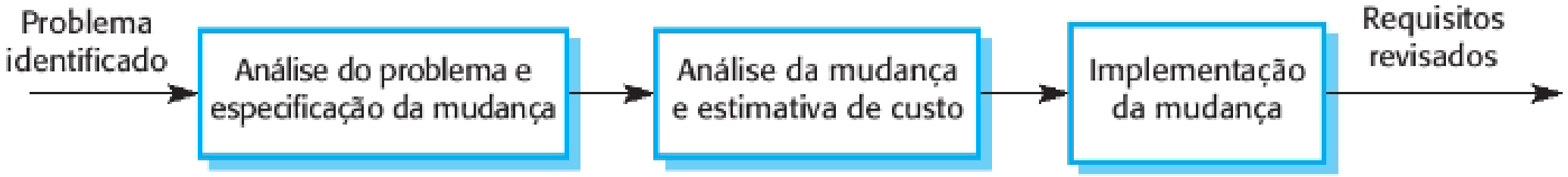
- Identificação de requisitos: cada requisito deve ser identificado exclusivamente para que ele possa ser comparado com outros requisitos
- Processo de gerenciamento de mudanças: Esse é o conjunto de atividades que avaliam o impacto e o custo das mudanças

Planejamento de gerenciamento de requisitos

- 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 mudanças de requisitos

FIGURA 4.19 Gerenciamento de mudança de requisitos.



Requisitos permanentes e voláteis

- 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 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

Rastreabilidade

- 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 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

- Armazenamento de requisitos
  - Os requisitos devem ser mantidos em um repositório de dados seguro e gerenciado
- 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
- Gerenciamento de rastreabilidade
  - Recuperação automatizada das ligações entre os requisitos

Pontos-chave (1/4) (SOMMERVILLE, 2019)

- Os requisitos de um sistema de software estabelecem o que o sistema deve fazer e definem as restrições à sua operação e implementação
- Os requisitos funcionais são declarações de serviços que o sistema deve prestar ou descrições de como devem ser feitas algumas computações

Pontos-chave (2/4) (SOMMERVILLE, 2019)

- Os requisitos não-funcionais costumam restringir o sistema que está sendo desenvolvido e seu processo de desenvolvimento, sejam requisitos de produto, de negócios ou externos. Muitas vezes, estão relacionados com as propriedades emergentes do sistema e, portanto, se aplicam ao sistema como um todo
- O processo de engenharia de requisitos inclui a elicitação, a especificação , a validação e o gerenciamento de requisitos

Pontos-chave (3/4) (SOMMERVILLE, 2019)

- A elicitação de requisitos é um processo iterativo que pode ser representado como uma espiral de atividades – descoberta, classificação e organização, negociação e documentação de requisitos
- A especificação de requisitos é um processo de documentar formalmente os requisitos de usuário e de sistema e de criar um documento de requisitos de software
- O documento de requisitos é uma declaração de acordo dos requisitos do sistema. Deve ser organizado para que os clientes do sistema e os desenvolvedores de software possam usá-lo

Pontos-chave (4/4) (SOMMERVILLE, 2019)

- A validação de requisitos é o processo de conferir sua validade, consistência, completude, realismo e verificabilidade
- As mudanças nos ambientes de negócios, nas organizações e nas tecnologias levam, inevitavelmente, a mudanças nos requisitos de um sistema de software. O gerenciamento de requisitos é o processo de gerenciar e controlar essas mudanças.