

**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Campus Feliz

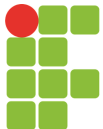
Requisitos de Software

Profa. Dra. Ana Paula Lemke

Última atualização do material em
12/09/2019.

Objetivos da Aula

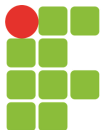
- Compreender o conceito de requisito.
- Entender a importância da Engenharia de Requisitos no processo de desenvolvimento de software.
- Apresentar as diferentes classificações para requisitos.
- Identificar as diferenças entre os requisitos de software funcionais e não funcionais.
- Discutir diferentes técnicas de levantamento de requisitos.
- Introduzir técnicas para verificação e validação de requisitos.



Contextualização

Como ter sucesso nos projetos?

- Análise do Problema (Requisitos de Negócio)
 - Entenda o problema
 - Obtenha concordância dos envolvidos
- Levantamento dos Requisitos (Requisitos do Sistema)
 - Identifique quem usará o sistema (atores)
 - Descubra como o sistema será usado (casos de uso)
- Gerência de Requisitos
 - Especifique os requisitos
 - Gerencie expectativas, mudanças e erros
 - Gerencie o aumento do escopo
 - Defina a equipe e a mantenha informada



Contextualização - Requisitos

- Visão histórica:
 - Requisitos são funções.
- Visão atual:
 - Requisitos são objetivos, funções, propriedades, restrições que o sistema deve possuir/obedecer para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s);
 - Requisitos são as funcionalidades e condições que o sistema – e mais genérico, o projeto – deve estar em conformidade.

Os requisitos do sistema são influenciados pelas preferências, recusas e preconceitos dos usuários, além das questões políticas e organizacionais (Sommerville, 2007).



Requisito - Definição

- Definição “*Requirement*” (IEEE Standard Glossary of Software Engineering Terminology - 1997)
 - 1) *A condition or capability needed by a user to solve a problem or achieve an objective.*
 - 2) *A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.*
 - 3) *A documented representation of a condition or capability as in (1) or (2).*



Requisito - Definição

- Definição “Requisito” (IEEE *Standard Glossary of Software Engineering Terminology* - 1997)
 - 1) Uma condição ou uma funcionalidade necessária a um usuário para resolver um problema
 - 2) Uma condição ou funcionalidade que deve ser atingida ou influenciada por um componente de sistema para satisfazer um contrato, padrão, especificação, ou outro documento formalmente definido
 - 3) Uma representação documentada de uma condição ou funcionalidade conforme itens 1 ou 2.



Requisitos – Exemplos*

- O sistema deve fornecer informações sobre todas as ações executadas por seus usuários em qualquer período de tempo.
- O software deve permitir ampla consulta sobre os dados dos pedidos do cliente, inclusive via WWW.
- O sistema deve estar integrado ao sistema bancário XYZ utilizado pela organização.
- O sistema deve oferecer ao cliente a compra menos custosa que satisfaça seus parâmetros de definição do produto.
- O sistema deve seguir um processo de desenvolvimento baseado no RUP.
- O sistema deve gerar relatórios sobre vendas por período, evidenciando os parâmetros de avaliação dos envolvidos.

* Existem diferentes formas de se escrever requisitos. Os exemplos apresentados nesse slide não estão necessariamente escritos no formato que será solicitado em aula.

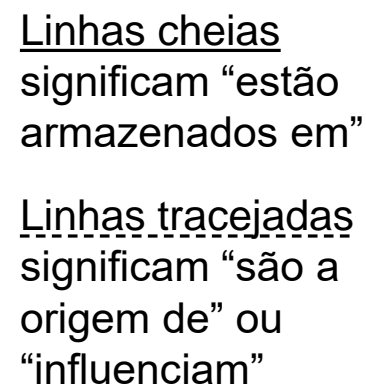


Tipos de Informações de Requisitos [1..2]

Tabela 1-1 Alguns tipos de informações de requisitos. Livre tradução de (WIEGERS e BEATTY, 2014)

Termo	Definição
Requisito de negócio	Objetivo de negócio de alto nível da organização que constrói o produto ou do cliente que adquire o novo produto.
Regra de negócio	Uma política, guia, padrão ou lei que define ou restringe algum aspecto do negócio. Não é um requisito de software propriamente dito, mas a origem de vários tipos de requisitos de software.
Restrição	Uma restrição que é imposta sobre as escolhas disponíveis para o desenvolvedor para o projeto e construção de um produto.
Requisito de interface externa	Uma descrição de uma conexão entre um sistema de software e um usuário, outro sistema de software ou um dispositivo de hardware.
Feature	Uma ou mais “capacidades” do sistema relacionadas logicamente que proveem valor para um usuário e são descritas por um conjunto de requisitos funcionais.
Requisito funcional	Descrição de um comportamento que um sistema irá exibir sobre condições específicas.
Requisito não funcional	Descrição de uma propriedade ou característica que um sistema deve apresentar ou uma restrição que ele deve respeitar.
Atributo de qualidade	Um tipo de requisito não funcional que descreve uma característica de serviço ou desempenho de um produto.
Requisito de sistema	Um requisito de alto nível para um produto que contém múltiplos subsistemas, que podem ser totalmente de software ou de software e hardware.
Requisito de usuário	Uma meta ou tarefa que classes específicas de usuários devem ser capazes de executar com um sistema ou um atributo de produto desejado.

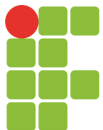




Requisitos – Níveis de descrição [1..3]

- **Requisitos de usuários:** requisitos abstratos de alto nível.
 - São declarações, em uma linguagem natural com diagramas, de quais serviços são esperados do sistema e as restrições as quais ele deve operar.
- **Requisitos de sistema:** descrição detalhada de o que o sistema deve fazer.
 - Definem, detalhadamente, as funções, serviços e as restrições operacionais do sistema. O documento de requisitos, chamado de **especificação funcional**, pode servir como um contrato entre cliente e desenvolvedor.

Baseado em Sommerville, I. “Engenharia de Software”, 9º ed, 2011 (pág. 58).



Requisitos – Níveis de descrição [2..3]

Exemplo (sistema para biblioteca LIBSYS):

Definição do requisito de usuário

1. LIBSYS deve manter o acompanhamento de todos os dados exigidos pelas agências de licenciamento de direitos autorais no Reino Unido e em outros lugares.

Especificação dos requisitos de sistema:

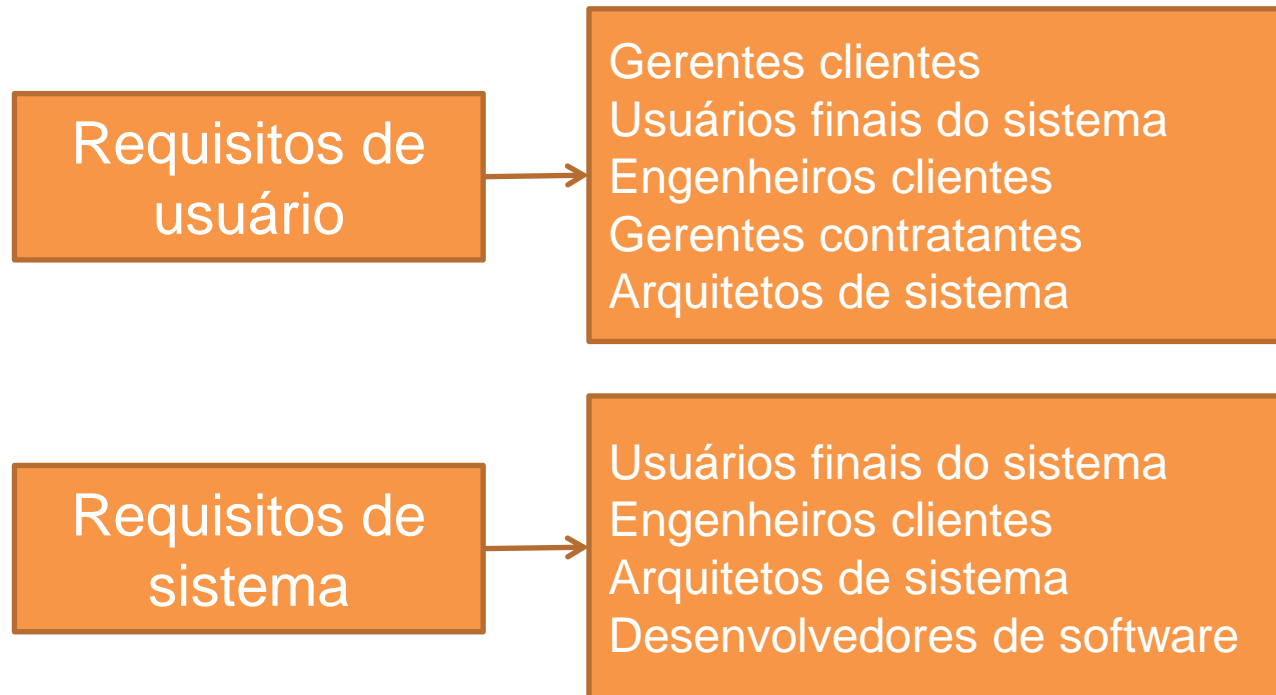
- 1.1. Ao solicitar um documento ao LIBSYS, deve ser apresentado ao solicitante um formulário que registra os detalhes do usuário e da solicitação feita.
- 1.2. Os formulários de solicitação do LIBSYS devem ser armazenados no sistema durante 5 anos, a partir da data de solicitação.
- 1.3. Todos os formulários do LIBSYS devem ser indexados por usuário, nome do material solicitado e fornecedor da solicitação.
- 1.4. O LIBSYS deve manter um registro de todas as operações feitas no sistema

Fonte: Sommerville, 2007.

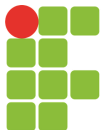


Requisitos – Níveis de descrição [3..3]

- Leitores dos diferentes tipos de requisitos



Baseado em Sommerville, I. "Engenharia de Software", 9º ed, 2011 (pág. 59).



Requisitos – Classificação [1..2]

- **Requisitos Funcionais**: são declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também estabelecer explicitamente o que o sistema não deve fazer.
- **Requisitos não Funcionais**: são restrições aos serviços ou funções oferecidos pelo sistema. Eles incluem restrições de *timing*, restrições no processo de desenvolvimento e restrições impostas pelas normas. Frequentemente aplicam-se sobre o sistema com um todo.

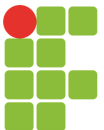
Fonte: Sommerville, I. “Engenharia de Software”, 9º ed, 2011 (pág. 59).



Requisitos – Classificação [2..2]

- A distinção entre os dois tipos de requisitos não é tão clara como sugerem as definições:
 - “Um requisito de usuário relacionado com a proteção, tal como uma declaração de limitação de acesso a usuários autorizados, pode parecer um requisito não funcional. No entanto, quando desenvolvido em mais detalhes, esse **requisito pode gerar outros requisitos**, claramente funcionais, como a necessidade de incluir recursos de autenticação de usuário no sistema.”

Fonte: Sommerville, I. “Engenharia de Software”, 9º ed, 2011 (pág. 59).



Atividade 1

Classifique os requisitos* abaixo em funcionais e não funcionais.

1. A base de dados deve ser protegida para acesso apenas de usuários autorizados.
2. O tempo de resposta do sistema não deve ultrapassar 30 segundos.
3. O software deve rodar no sistema Linux.
4. Cada vendedor pode ter pedidos atendidos e em aberto, e deve ser possível visualizar as vendas por dia e os pedidos atendidos por dia para cada vendedor.
5. O sistema deve fornecer telas apropriadas para o usuário ler os documentos no repositório de documentos.
6. Cada pedido deve ser associado a um identificador único (PID), o qual o usuário pode copiar para a área de armazenamento permanente da conta.

* Existem diferentes formas de se escrever requisitos. Os exemplos apresentados nesse slide não estão necessariamente escritos no formato que será solicitado em aula.



Requisitos Funcionais [1..2]

- Representam o comportamento do sistema.
- Descrevem os comportamentos do sistema em detalhes através de cenários de negócio e/ou sistema
- Devem prever **o que** se espera que o software faça (resultado), sem a preocupação de **como** ele será construído internamente para que isto ocorra.
- Todos os casos de uso são requisitos funcionais.
 - Regras de negócio, glossário, requisitos de processos e diversos outros elementos que caem na categoria de comportamento.



Requisitos Funcionais [2..2]

- Exemplos*:
 - “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, a lista dos pacientes para as consultas daquele dia”.
 - “Cada membro da equipe que usa o sistema deve ser identificado apenas por seu número de oito dígitos”.

Fonte: Sommerville, I. “Engenharia de Software”, 9º ed, 2011 (pág. 59).

* Existem diferentes formas de se escrever requisitos. Os exemplos apresentados nesse slide não estão necessariamente escritos no formato que será solicitado em aula.



Requisitos Não Funcionais [1..5]

- Representam características e restrições do sistema.
- Referem-se às qualidades globais de um software.
- Tipos de requisitos não funcionais:
 - **Requisitos de produto:** especificam ou restringem o comportamento do software.
 - **Requisitos organizacionais:** são derivados de políticas e procedimentos da organização do cliente e do desenvolvedor.
 - **Requisitos externos:** são todos os requisitos derivados de fatores externos ao sistema e ao seu processo de desenvolvimento.

Fonte: Sommerville, I. “Engenharia de Software”, 9ª ed, 2011 (pág. 61).



Requisitos Não Funcionais [2..5]

○ Requisitos de produto

- Requisitos de desempenho quanto à rapidez com que o sistema deve executar e quanta memória ele requer, os requisitos de confiabilidade que estabelecem a taxa aceitável de falhas, os requisitos de proteção e os requisitos de usabilidade.

○ Requisitos organizacionais

- Requisitos do processo operacional, que definem como o sistema será usado, os requisitos do processo de desenvolvimento (linguagem de programação, ambiente de desenvolvimento ou normas de processo a serem utilizadas) e requisitos ambientais, que especificam o ambiente operacional do sistema.

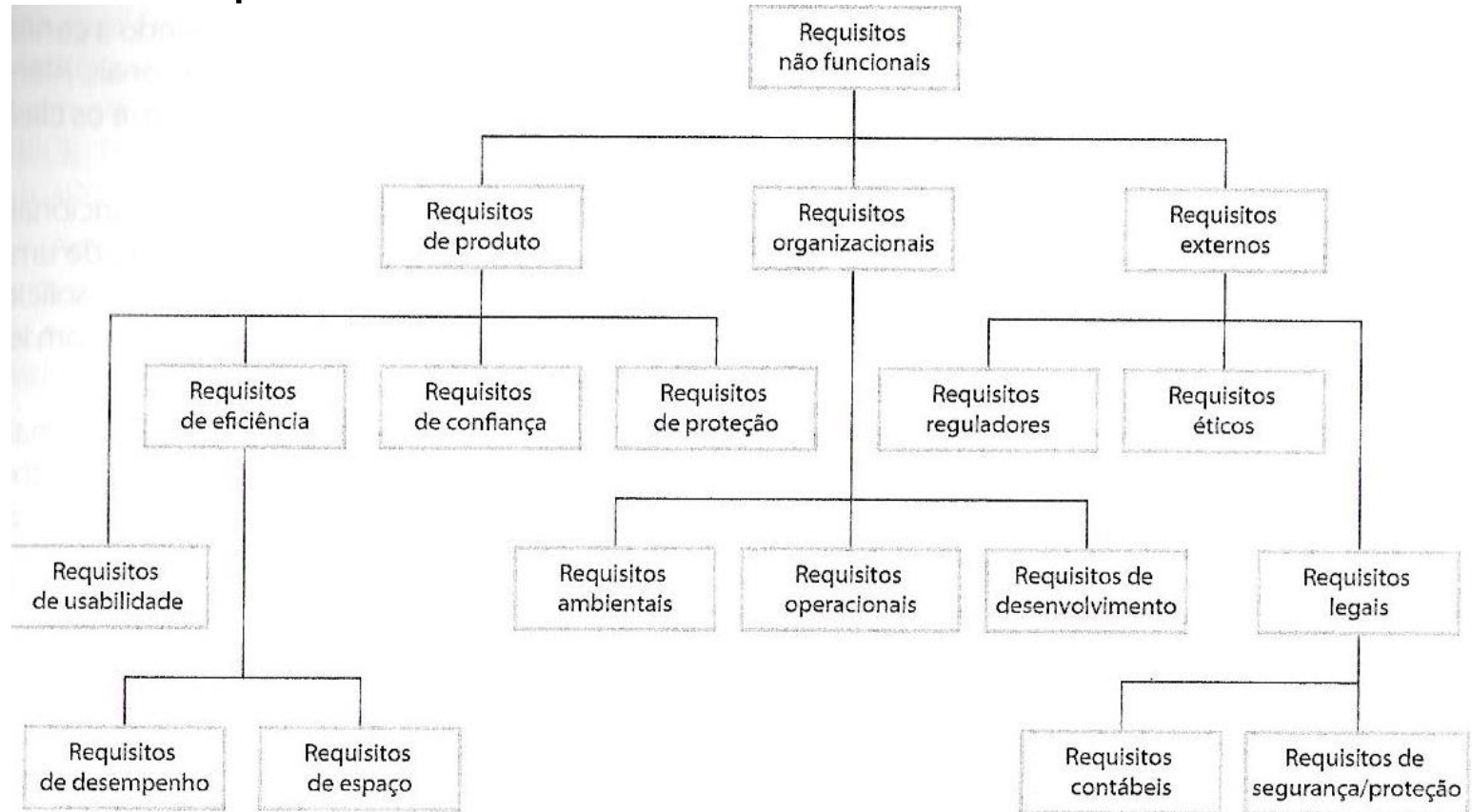
○ Requisitos externos

- Podem incluir requisitos reguladores, que definem o que deve ser feito para que o sistema seja aprovado para uso, requisitos legais (para o sistema operar dentro da lei) e requisitos éticos.



Requisitos Não Funcionais [3..5]

- Tipos de requisitos não funcionais



Requisitos Não Funcionais [4..5]

- Requisito não funcional de usabilidade no estilo “meta geral”:
 - *“O sistema deve ser de fácil uso pelo pessoal médico e deve ser organizado de tal maneira que os erros dos usuários sejam minimizados”.*
 - **Problema:** deixa margem para interpretação, o que pode gerar disputas na entrega do sistema.
- Requisito não funcional de usabilidade reescrito de forma “testável” (escrito quantitativamente):
 - *“A equipe médica deve ser capaz de usar todas as funções do sistema após quatro horas de treinamento. Após esse treinamento, o número médio de erros cometidos por usuários experientes não deve exceder dois por hora de uso do sistema”.*



Requisitos Não Funcionais [5..5]

- Métricas para especificar 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
Facilidade de uso	Tempo de treinamento Número de frames 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



O que **NÃO** faz parte de um Requisito?

- **Design:** como o software atenderá aos requisitos
- **Verificação:** como você sabe que o requisito foi atendido
- **Dados gerenciais do projeto:** cronogramas, controles, etc.

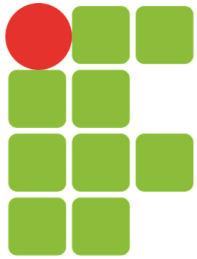


Atividade 2

Com base na descrição de um Sistema de Agendamento de Estudos Orientados, faça o que se pede.

- Em nossa escola, os professores controlam individualmente e manualmente o agendamento e frequência de alunos nos estudos orientados. Como não há um gerenciamento desse agendamento, muitas salas são reservadas em vão ou não são reservadas quando precisa. Também não temos uma estimativa (enquanto diretores) do quantitativo de alunos que estão frequentando os horários de atendimento. Assim, gostaríamos de um sistema Web que substituísse o processo manual atual, permitindo que os alunos agendassem estudos orientados com determinados professores. Os professores seriam responsáveis por indicar quais os dias e horários que disponibilizam para atendimento, além da sala e da quantidade de vagas. Os alunos não precisarão se cadastrar, mas precisarão indicar o número de matrícula na hora de agendar um atendimento com determinado professor (também será necessário indicar qual o conteúdo que tem dúvida para realizar o agendamento). Os professores e diretores devem poder ver os alunos que agendaram atendimento em determinada data e também outros relatórios.
 - Enumere 5 **funcionalidades** desejadas (requisitos funcionais).
 - Enumere 5 **características** desejadas (requisitos não funcionais).



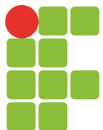


**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Campus Feliz

Engenharia de Requisitos

Engenharia de Requisitos

- Processo de descobrir, analisar, documentar e verificar requisitos (Sommerville, 2011, pág. 57).
- Enfoque sistemático para identificação e manutenção dos requisitos.
- O processo geral envolve, basicamente, quatro subprocessos relacionados à obtenção, documentação e verificação de requisitos:
 - Estudo de Viabilidade
 - Elicitação e análise de requisitos
 - Especificação de requisitos
 - Validação de requisitos
- Como os requisitos mudam continuamente, há o processo de **gerenciamento de requisitos**, que permeia os outros quatro subprocessos.



Dimensões da Engenharia de Requisitos



- Compreensão Vaga



- Compreensão mais completa e clara do sistema



- Pontos de vistas diferentes



- Consenso negociado



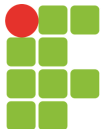
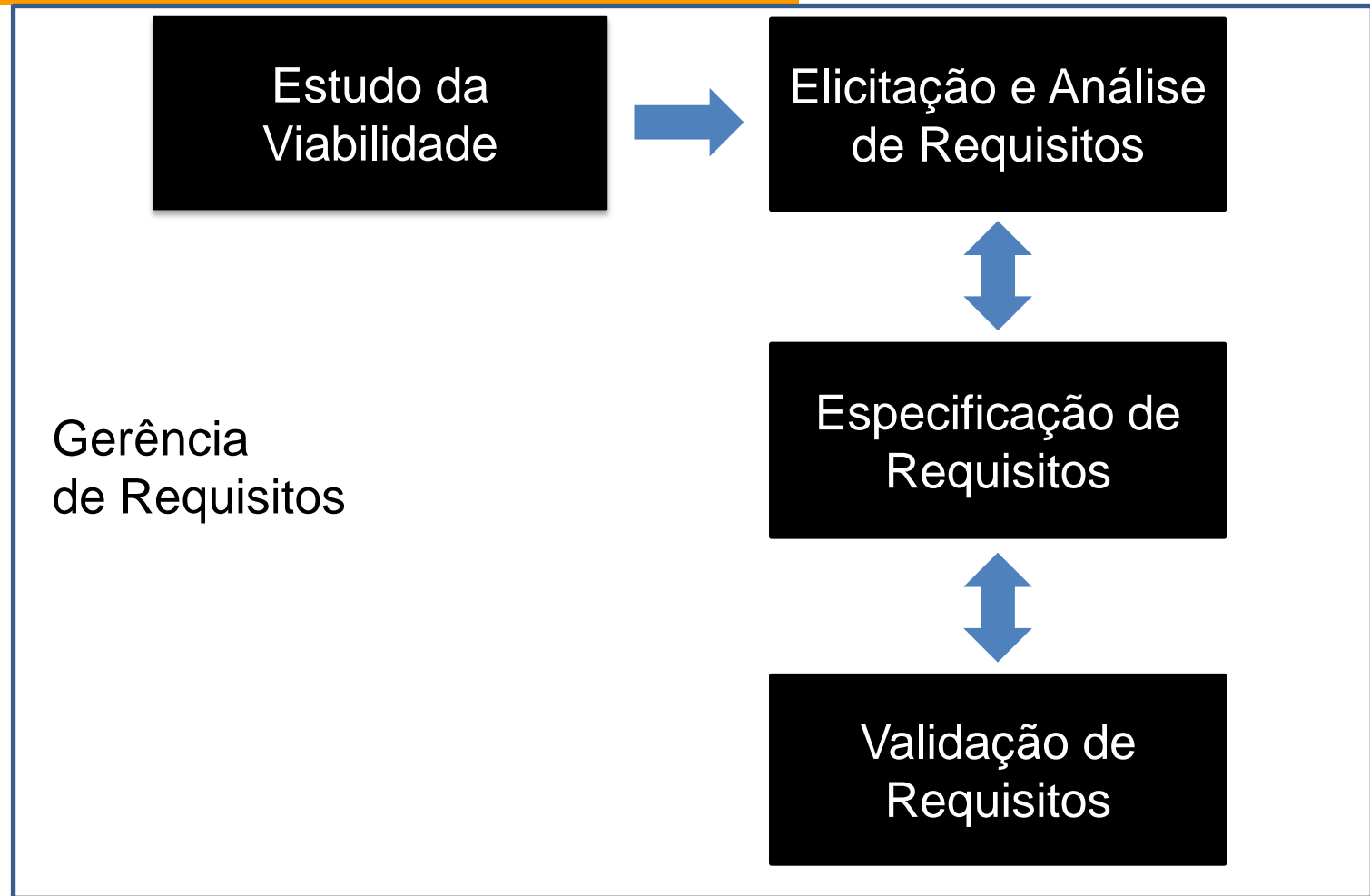
- Representação informal



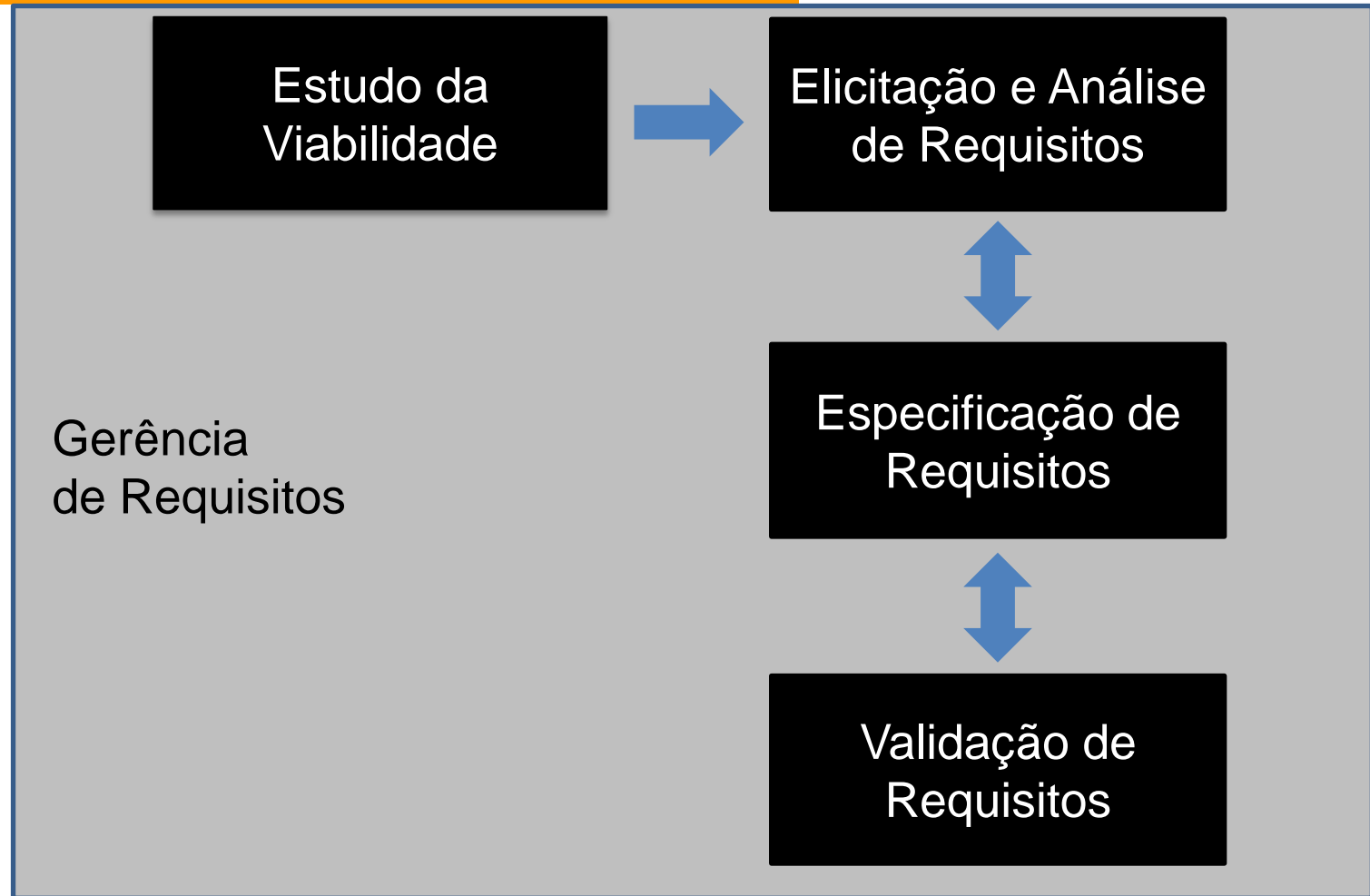
- Representação Formal



Engenharia de Requisitos



Engenharia de Requisitos



Gerência de Requisitos [1..2]

- Abordagem sistemática para **levantar, documentar, organizar, e rastrear mudanças** em requisitos.
- “*O gerenciamento de requisitos é um processo para compreender e controlar as mudanças dos requisitos dos sistemas*”. (Sommerville, 2007)
 - É preciso acompanhar os requisitos individuais.
 - É preciso manter as ligações entre os requisitos dependentes, para poder avaliar o impacto de alterações.
 - É preciso estabelecer um processo formal para se fazer mudanças nos requisitos e ligá-las aos requisitos do sistema.

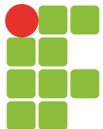
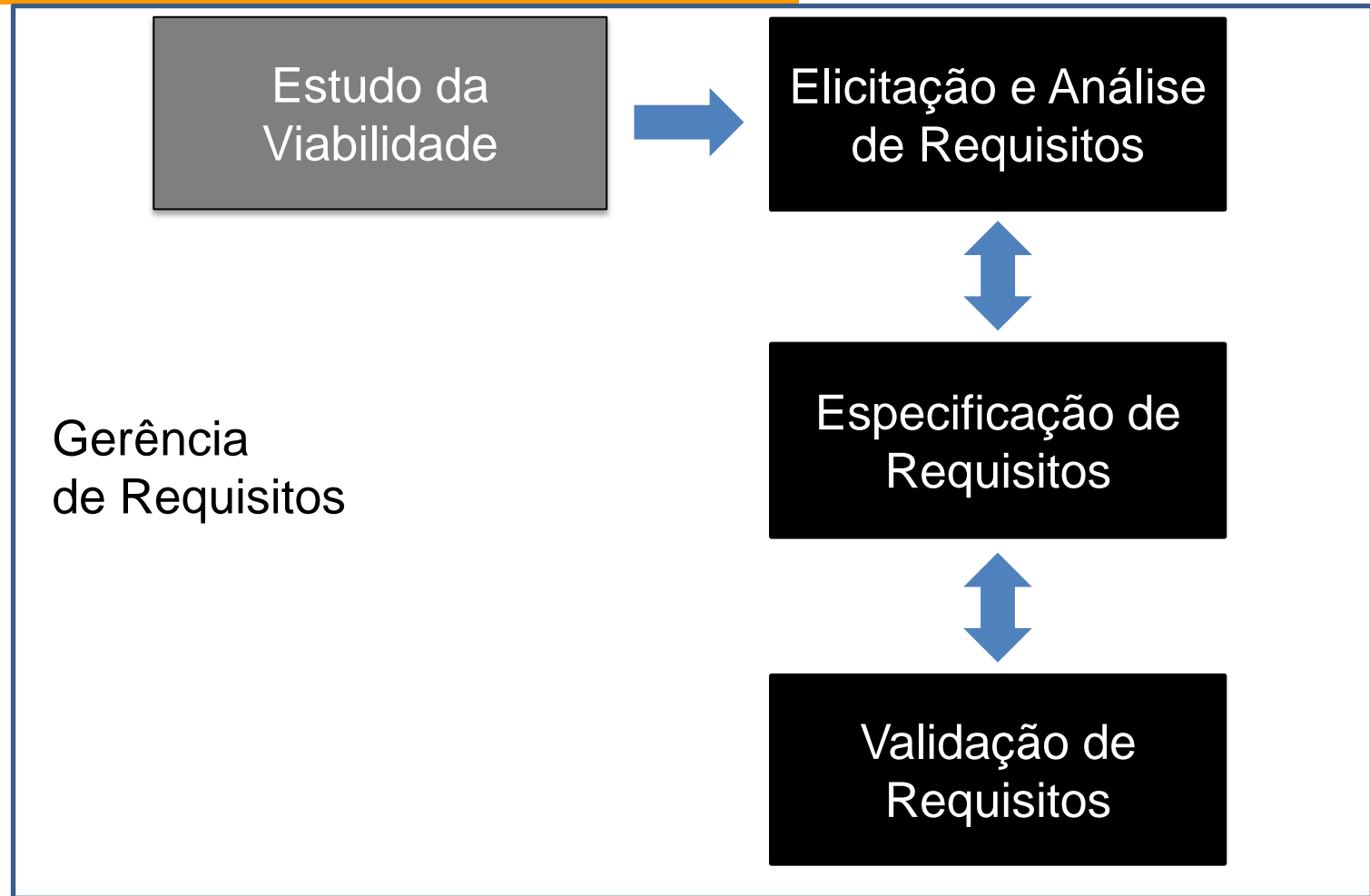


Gerência de Requisitos [2..2]

- A Gerência de Requisitos nem sempre é uma tarefa fácil, pois os requisitos:
 - Nem sempre são óbvios
 - Chegam por várias fontes (diferentes *stakeholders*)
 - Nem sempre são facilmente expressos em palavras
 - Estão relacionados entre si e entre outros produtos do processo de engenharia de software
 - Possuem propriedades e valores únicos
 - MUDAM!!
- Se os requisitos não são controlados, eles podem se tornar ingerenciáveis.



Engenharia de Requisitos

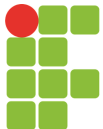
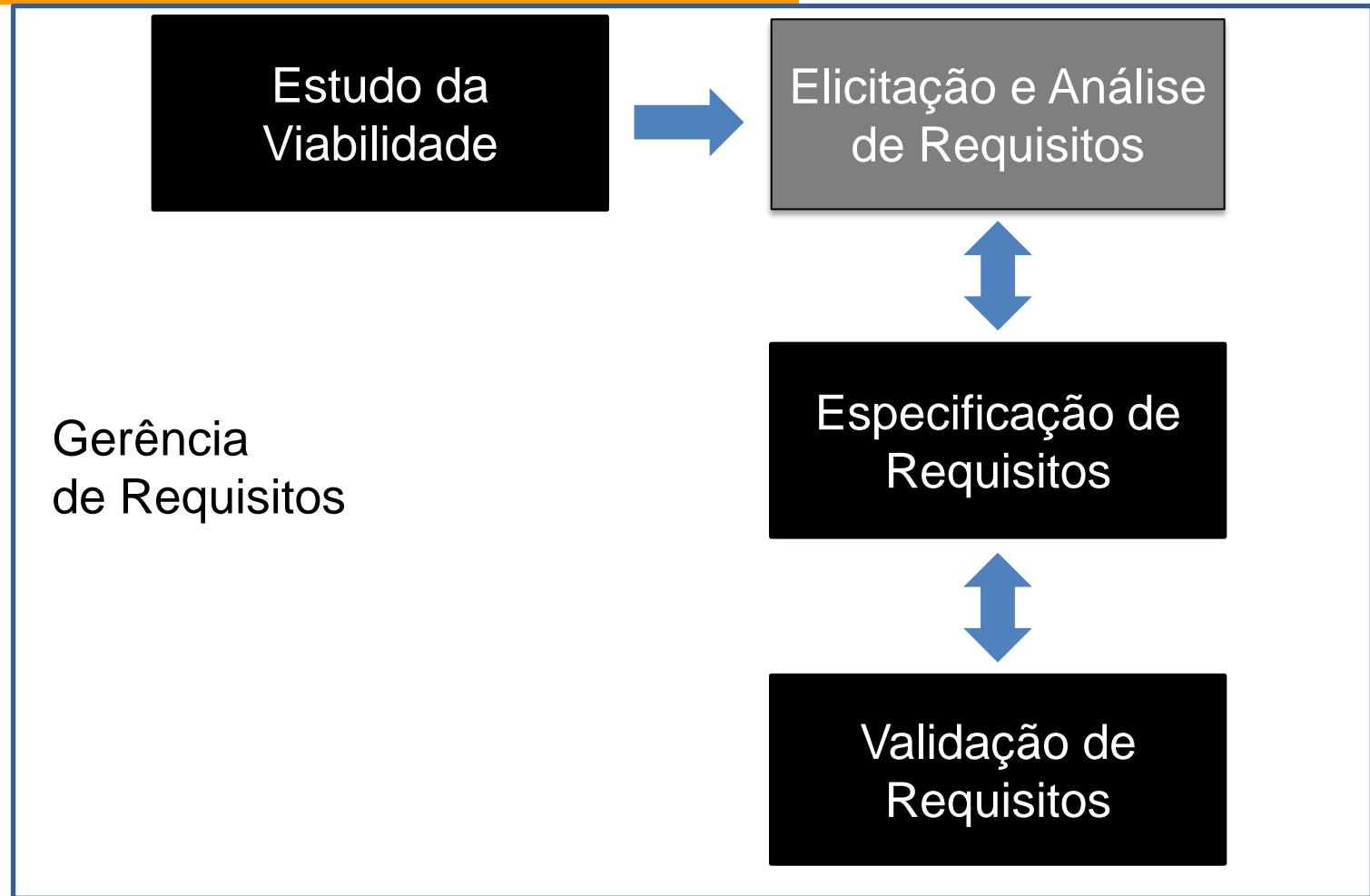


Estudo de Viabilidade

- Processo essencial principalmente em sistemas novos.
 - Entradas: conjunto preliminar de requisitos de negócios, um esboço da descrição do sistema e como o sistema pretende apoiar os processos de negócios.
 - Resultados: um relatório dizendo se vale a pena ou não seguir com o processo de engenharia de requisitos.
- Estudo breve e focalizado que procura responder questões como:
 - O sistema contribui para os objetivos gerais da organização?
 - O sistema pode ser implementado com tecnologia atual e dentro das restrições definidas de custo e prazo?
 - O sistema pode ser integrado a outros sistemas já implantados?



Engenharia de Requisitos



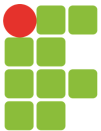
Elicitação e Análise de Requisitos

- Nesta atividade, os engenheiros de software trabalham com os **clientes** e os **usuários finais** do sistema para **obter informações sobre o domínio da aplicação, os serviços que o sistema deve oferecer, o desempenho esperado do sistema, restrições de hardware** e outros.
 - A elicitação de requisitos é o processo de descobrimento dos requisitos do sistema.



Elicitação e Análise de Requisitos

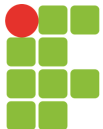
- Atividades do processo de elicitación e análise de requisitos:
 - **Descoberta de requisitos:** processo de interação com os *stakeholders* do sistema para coletar seus requisitos. Os requisitos de domínio são também descobertos durante essa atividade, provenientes dos *stakeholders* e da documentação.
 - **Classificação e organização de requisitos:** envolve o agrupamento de requisitos relacionados e sua organização em conjuntos coerentes (costuma-se utilizar um modelo de arquitetura do sistema para identificar subsistemas e associar os requisitos a estes subsistemas)..
 - **Priorização e negociação de requisitos:** quando vários *stakeholders* participam do processo, geralmente os requisitos são conflitantes. Esta atividade relaciona-se à priorização de requisitos, à procura e à resolução de conflitos de requisitos por meio de negociação.
 - **Especificação de requisitos:** os requisitos devem ser documentados de forma informal e/ou formal (versão inicial, incompleta, do documento de requisitos).



Elicitação e Análise de Requisitos

Descoberta de Requisitos

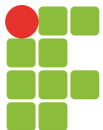
- A atividade de capturar ou obter requisitos passa por algumas etapas, como:
 - Entendimento do **domínio da aplicação**: conhecimento geral onde o sistema será aplicado.
 - Entendimento do **problema**: os detalhes dos problemas específicos do cliente.
 - Entendimento do **negócio**: como os sistemas interagem e contribuem de forma geral com os objetivos de negócio.
 - Entendimento das necessidades e limitações dos *stakeholders* do sistema.



Elicitação e Análise de Requisitos

Descoberta de Requisitos

- Técnicas/práticas para obtenção de requisitos:
 - Algumas práticas são comuns e bem conhecidas quando se fala em captura de requisitos
 - É necessário selecionar e integrar diferentes práticas conforme a disponibilidade e situação do ambiente
 - É importante utilizar uma técnica de modelagem de apoio para que os fatos capturados fiquem corretamente representados para futuro tratamento.
 - A escolha das técnicas e seu esquema de integração dependerá do problema e da equipe participante. O ponto importante é ter conhecimento sobre estas técnicas e identificar onde uma técnica é mais interessante que outra



Elicitação e Análise de Requisitos

Descoberta de Requisitos

- Técnicas/práticas para obtenção de requisitos:
 - *Brainstorming* ou redução de ideias
 - Entrevistas
 - Questionários
 - *Workshop* de requisitos
 - Etnografia
 - Protótipos
 - Revisão de especificações de requisitos do cliente
 - Leitura de documentos
 - JAD – *Joint Application Development*
 - Mapas mentais
 - ...



Elicitação e Análise de Requisitos

Descoberta de Requisitos – Entrevistas

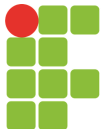
- Técnica que não exige um conhecimento muito aprofundado e é acessível, tanto financeiramente quanto operacionalmente.
- A ideia principal da entrevista é oportunizar um momento onde o Cliente, os Analistas de Negócios e Analistas de Sistemas possam entrar em contato e discutir o escopo do sistema a ser trabalhado.
- Neste encontro, o Cliente deve ser questionado sobre todas as situações possíveis às quais o software, de alguma forma, deverá atender.
- Deve-se propor tanto situações de sucesso e insucesso para capturar o comportamento desejado.



Elicitação e Análise de Requisitos

Descoberta de Requisitos – Entrevistas

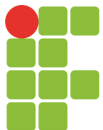
- As principais vantagens da entrevista são o contato direto com o usuário e a validação imediata.
- As principais desvantagens são o conhecimento tácito e diferenças de cultura e visão.
 - Conhecimento tácito é aquele que o indivíduo adquiriu ao longo da vida, que está na cabeça das pessoas.
 - Geralmente é difícil de ser formalizado ou explicado a outra pessoa, pois é subjetivo e inerente às habilidades de uma pessoa e, por isso, difícil de ser repassado.



Elicitação e Análise de Requisitos

Descoberta de Requisitos – Entrevistas

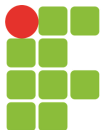
- Entrevista fechada:
 - Analista de Sistemas busca respostas para um conjunto de questões pré-definidas, geralmente preparadas em um formulário e enviadas por e-mail ou utilizadas na reunião presencial.
 - Ao final, o Analista tem uma série de afirmações que poderão ser trabalhadas e, com isso, decidir o que é ou não é requisito.
- Entrevista aberta:
 - Não há uma agenda pré-definida e o Analista de Sistemas discute, de forma aberta, o que os clientes querem do sistema.
 - Este tipo de reunião se assemelha muito ao *Brainstorm* e somente deve ser utilizada quando há urgência no contato com os clientes.



Elicitação e Análise de Requisitos

Descoberta de Requisitos – Entrevistas

- Independente do tipo da entrevista, o importante é analisar previamente o perfil do cliente e da equipe e sugerir um tipo ou outro.
- Se estiver em dúvida em relação ao tipo de entrevista, **utilize a fechada**, pois geralmente os clientes esperam que o Analista de Sistemas faça as perguntas corretas e de forma inteligente e, assim, acelerar o processo de passagem da informação.
- É essencial nas entrevistas que tanto os entrevistadores quanto os entrevistados estejam de “cabeça aberta” e não façam a entrevista com noções pré-concebidas sobre o que é necessário.



Elicitação e Análise de Requisitos

Descoberta de Requisitos – Entrevistas

- Desenvolva sempre questões abertas e livres de contexto em suas entrevistas/questionários.
- Por exemplo, observe as perguntas abaixo:
 - Você gostaria da sua nova aplicação em um *browser*?
 - Você gostaria de sua nova aplicação em um browser mais do que como uma aplicação nativa do Windows mesmo que isso signifique uma menor performance, experiência, flexibilidade e interatividade?
 - O que você gostaria de abrir mão se sua nova aplicação fosse desenvolvida em um browser?
 - Você trocaria performance de sua aplicação por uma melhor experiência do usuário?
 - A performance é mais importante que outras partes da aplicação?



Elicitação e Análise de Requisitos

Descoberta de Requisitos – Entrevistas

- Para ter uma entrevista bem-sucedida com o usuário é importante:
 - Desenvolver um plano geral de entrevistas
 - O analista deve estar contextualizado sobre o tema.
 - Certificar-se da autorização para falar com os usuários
 - Planejar a entrevista para fazer uso eficiente do tempo
 - Utilizar ferramentas automatizadas que sejam adequadas
 - Tentar descobrir a informação no qual o usuário está mais interessado
 - Usar um estilo adequado ao entrevistar
 - Uma entrevista não é uma competição nem uma prova de conhecimento técnico.

Ao entrevistar o usuário/cliente, esqueça sua perspectiva e princípios do que é melhor (tecnologia, interface). Apenas entenda o que o usuário espera e garanta sua confiança.



Elicitação e Análise de Requisitos

Descoberta de Requisitos – Entrevistas

Plano de entrevista

- O planejamento de uma entrevista envolve vários passos:
 - Estudar material existente sobre o entrevistado e sua organização
 - Estabelecer objetivos
 - Decidir quem entrevistar
 - Preparar a entrevista
 - **Decidir sobre o tipo de questões e a estrutura da entrevista**
 - Decidir como registrar a entrevista



Elicitação e Análise de Requisitos

Descoberta de Requisitos – Entrevistas

Plano de entrevista – Tipos de questões

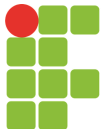
Tipo	Descrição	Exemplo	Vantagens	Desvantagens
Subjetiva	Permitem respostas “abertas”	O que você acha de...? Explique como você...	<ul style="list-style-type: none">- Riqueza de detalhes- Permitem novos questionamentos- Deixam o entrevistado a vontade e mais espontâneo	<ul style="list-style-type: none">- Muitos detalhes irrelevantes- Perda do controle da entrevista- Respostas muito longas com pouca informação útil- Podem dar a impressão que o entrevistado está perdido.
Objetiva	Limitam as respostas possíveis	Quantos...? Quem...? Quanto tempo...?	<ul style="list-style-type: none">- Ganho de tempo- Manter o controle da entrevista- Conduzem a dados relevantes	<ul style="list-style-type: none">- Podem ser maçantes para o entrevistado- Podem falhar na obtenção de detalhes importantes- Não criam afinidade entre entrevistador e entrevistado
De aprofundamento	Permitem explorar os detalhes de uma questão	Por quê? Você poderia dar um exemplo? Como isso acontece?		



Elicitação e Análise de Requisitos

Descoberta de Requisitos – Entrevistas

- Relatório da entrevista:
 - Deve capturar a essência da entrevista
 - Deve registrar entrevistado, entrevistador, data, assunto e objetivos
 - Deve indicar se os objetivos da entrevista foram alcançados e listar objetivos para entrevistas futuras
 - Registrar também os pontos principais da entrevista e a opinião do entrevistador



Elicitação e Análise de Requisitos

Descoberta de Requisitos – Etnografia

- Técnica baseada em observação, analisando o trabalho de determinada pessoa através do acompanhamento intenso e presente do desenvolvimento das atividades desta no dia-a-dia.
- Durante o seu trabalho, o Analista de Sistemas pode realizar perguntas diretamente para a pessoa observada.
 - Observação direta, verbalizada ou seguida de diálogo.
- Vantagem: permite esclarecer fatos não óbvios ou que não tenham sido comentados
- Desvantagem: constatar situações em que “o que se faz é diferentes do que se diz”.



Elicitação e Análise de Requisitos

Descoberta de Requisitos – **Leitura de Documentos**

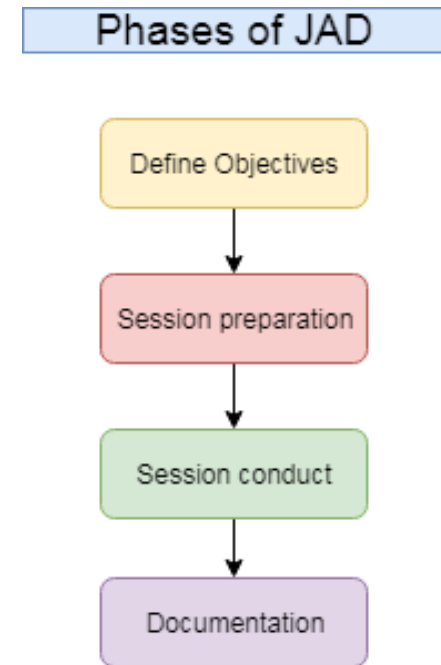
- Muitos requisitos já encontram-se expressos, implícita ou explicitamente, em documentos da empresa.
 - Estes documentos podem estar impressos, disponíveis na intranet da empresa, escritos na forma de manual ou até mesmo em arquivos de texto ou planilhas eletrônicas.
 - Aconselha-se ler a fundo os documentos encontrados e, em se tratando de planilhas eletrônicas, ler as fórmulas, gráficos e estruturas já criadas.
- Vantagem: facilidade de acesso e o volume de informação
- Desvantagem: dispersão das informações e volume de trabalho envolvido.



Elicitação e Análise de Requisitos

Descoberta de Requisitos – **JAD: Joint Application Development**

- JAD é uma metodologia de definição de requisitos baseada em **reuniões estruturadas (as JAD sessions)**.
 - Usuários e analistas projetam juntos o sistema, em sessões de grupos estruturadas e orientadas por um líder de reunião.
 - As reuniões buscam **decisões por consenso**.
 - Ela substitui as entrevistas e questionários individualizados.



Fonte da imagem:
<https://study.com/academy/lesson/joint-application-development-definition-phases-methodology.html>



Elicitação e Análise de Requisitos

Descoberta de Requisitos

- Comparação das técnicas de elicitación de requisitos:

	Entrevistas	JAD	Questionários	Análise de documentos	Observação
Tipo de Informação que se quer coletar	Existente, melhorias, novo	Existente, melhorias, novo	Existente, melhorias	Existente	Existente
Profundidade da informação (grau de detalhe)	Alto	Alto	Médio	Baixo	Baixo
Amplitude da informação (alcance da informação e das fontes)	Baixo	Médio	Alto	Alto	Baixo
Integração da informação	Baixo	Alto	Baixo	Baixo	Baixo
Envolvimento do usuário	Médio	Alto	Baixo	Baixo	Baixo
Custo	Médio	Baixo-Médio	Baixo	Baixo	Baixo-Médio

Fonte: Adaptado de WIXOM, B. H.; DENNIS, A. Análise e projeto de sistemas. 2. ed. São Paulo: LTC, 2005. (págs. 124 e 125).



Elicitação e Análise de Requisitos

Descoberta de Requisitos

- Técnicas de elicitación sugeridas de acordo com características do projeto:

	Entrevistas	Workshops	Observação	Questionários	Análise de Interface de usuário	Análise de documentos
Software para o público em massa	x			x		
Software corporativo interno	x	x	x			x
Substituição de um sistema existente	x	x	x		x	x
Melhoria de um sistema existente	x	x			x	x
Nova aplicação	x	x				
Implementação de software empacotado	x	x	x			x
Sistemas embutidos	x	x				x
<i>Stakeholders</i> distribuídos geograficamente	x	x		x		

Fonte: WIEGERS, K. E.; BEATTY, J. Software requirements. 3rd ed. Redmond: Microsoft Press, 2013 (pág. 130).



Elicitação e Análise de Requisitos

- Para descontrair...



Elicitação e Análise de Requisitos

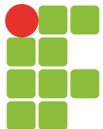
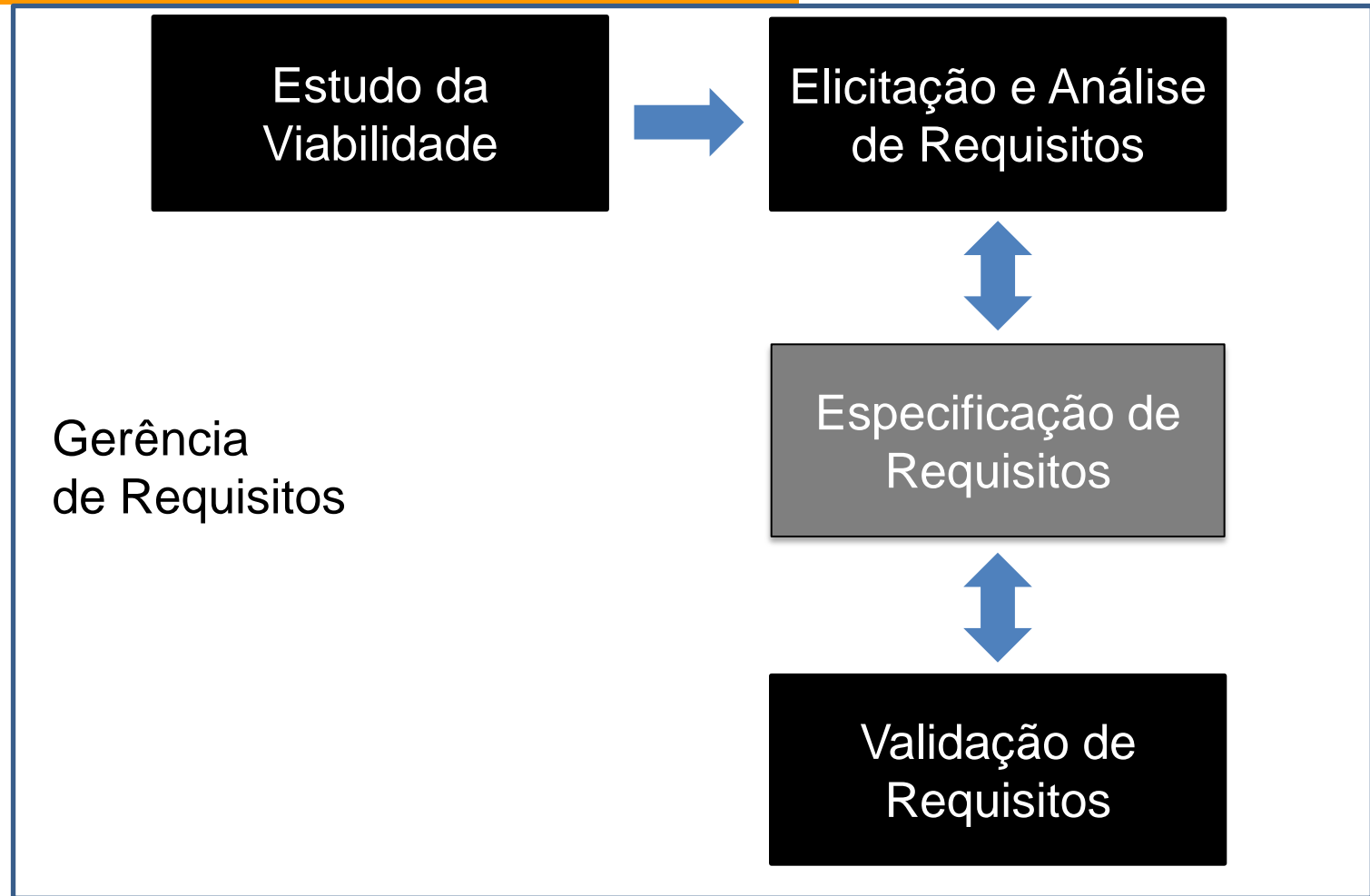
Descoberta de requisitos – **Habilidades Necessárias**



Fonte: Tonsig, 2013 (pág. 123).



Engenharia de Requisitos

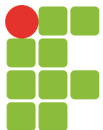


Especificação de Requisitos

Documento de requisitos de software ou
Especificação de requisitos de software ou
SRS – *Software Requirements Specification*

- É a declaração oficial do que os desenvolvedores do sistema devem implementar. Ele deve incluir os requisitos de usuário de um sistema e uma especificação detalhada dos requisitos do sistema.
 - Em alguns casos, os requisitos de usuário e de sistema podem ser integrados em uma única descrição.
 - Em outros, os requisitos de usuários são apresentados antes dos requisitos de sistema, introduzindo-os.

Fonte: Sommerville, 2007.

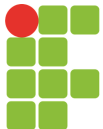


Especificação de Requisitos

SRS – *Software Requirements Specification*

- Geralmente é utilizado um modelo próprio para a criação do documento de requisitos.
- O documento de requisitos deve respeitar alguns atributos de qualidade para não comprometer o processo de desenvolvimento do software.
 - Um bom documento de requisitos deve ser: correto, completo, sem ambiguidade, verificável, modificável, deve permitir a priorização dos requisitos e também sua rastreabilidade.

Fonte: Sommerville, 2007.



Especificação de Requisitos

Tabela 4.3

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.

Fonte: Sommerville, I. "Engenharia de Software", 9º ed, 2011 (pág. 66).



Especificação de Requisitos

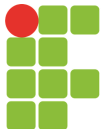
Quadro 4.2

Exemplo de requisitos para o sistema de software de bomba de insulina.

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.)*

Fonte: Sommerville, I. “Engenharia de Software”, 9º ed, 2011 (pág. 67).



Especificação de Requisitos

Quadro 4.3

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

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 de Requisitos

Tabela 4.4

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

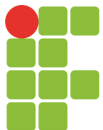
Fonte: Sommerville, I. "Engenharia de Software", 9ª ed, 2011 (pág. 69).



Especificação de Requisitos

Diretrizes para a escrita

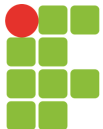
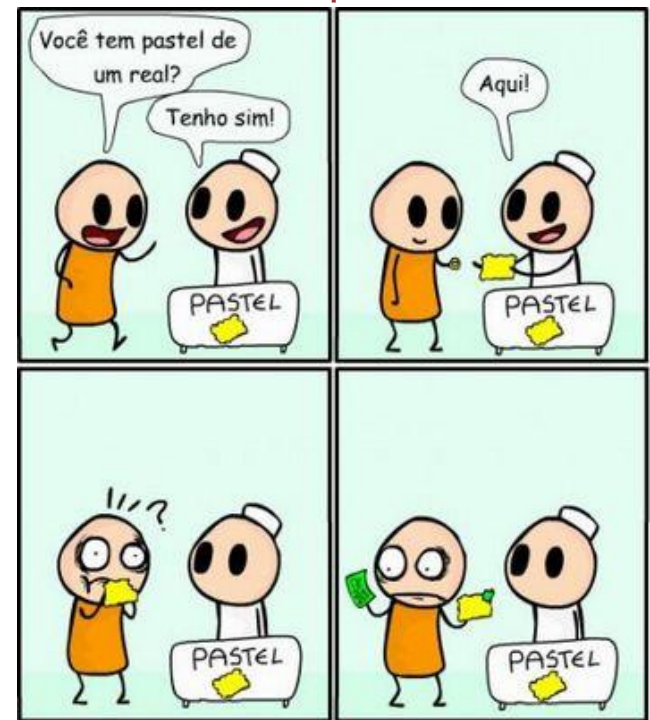
- Definir um formato padrão e usá-lo para todos os requisitos.
- Utilizar o idioma de forma consistente.
 - Usar “deve” para requisitos obrigatórios
 - Usar “deveria” (ou é recomendável) para requisitos desejáveis.
- Evitar o uso de jargões de computação
- Empregar termos característicos do problema
- Usar sentenças diretas e objetivas
- Usar vocabulário limitado



Especificação de Requisitos

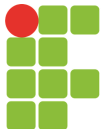
Diretrizes para a escrita

- Definir requisitos verificáveis
- Evitar ambiguidades
- Evitar sentenças muito longas
- Evitar uso de conjunções como **ou, e, com, também**
- Evitar termos vagos ou indefinidos



Qualidades de um SRS

- Correto
- Completo
- Consistente
- Unambíguo
- Verificável
- Priorizável
- Modificável
- Rastreável



Especificação de Requisitos

Estrutura do documento

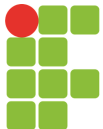
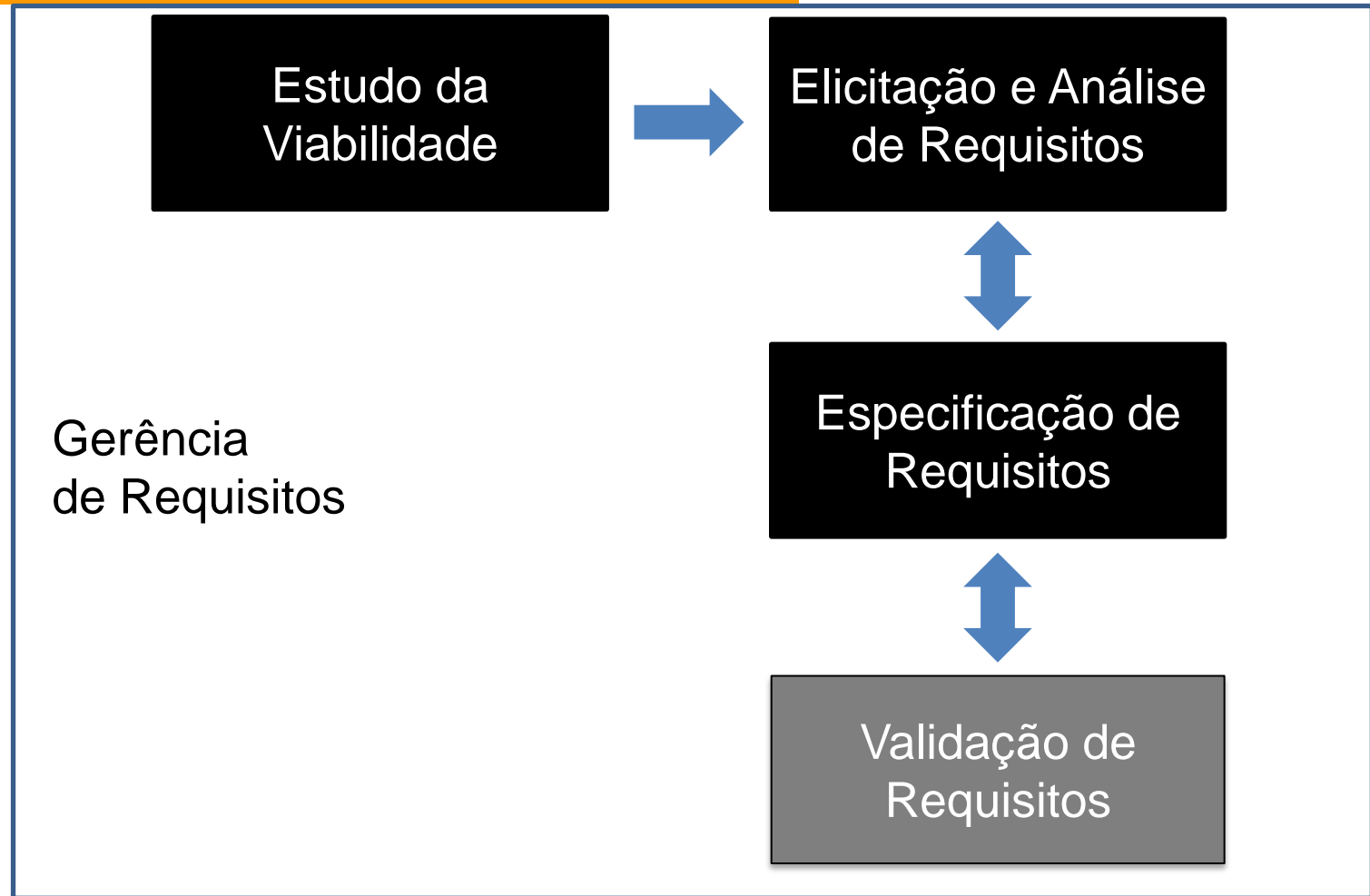
Capítulo	Descrição
Prefácio	Deve definir o público-alvo do documento e descrever seu histórico de versões, incluindo uma justificativa lógica para a criação da nova versão e um resumo das mudanças feitas em cada versão.
Introdução	Deve descrever a necessidade do sistema. Deve descrever brevemente suas funções e explicar como o sistema irá funcionar com outros sistemas. Deve descrever como o sistema atende aos objetivos gerais de negócios e 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 as habilidades do leitor.
Definição de requisitos de usuário	Os serviços fornecidos ao usuário e os requisitos não funcionais do sistema devem ser descritos nesta seção. Essa descrição pode usar linguagem natural, diagramas e outras notações compreensíveis pelos clientes. Padrões de produto e de processo a serem seguidos devem ser especificados.
Arquitetura de sistema	Este capítulo deve apresentar uma visão geral de alto nível da arquitetura prevista do sistema, mostrando a distribuição das funções nos módulos do sistema. Os componentes de arquitetura reusados devem ser destacados.
Especificação de requisitos de sistema	Deve descrever os requisitos funcionais e não funcionais mais detalhadamente. Caso necessário, mais detalhes podem também ser adicionados aos requisitos não funcionais; por exemplo, interfaces com outros sistemas devem ser definidas.
Modelos de sistema	Deve estabelecer um ou mais modelos de sistema, mostrando os relacionamentos entre os componentes e o sistema e seu ambiente. Podem ser modelos de objetos, modelos de fluxos de dados e modelos semânticos de dados.
Evolução de sistema	Deve descrever as hipóteses fundamentais sobre as quais o sistema está baseado, além de mudanças previstas devido à evolução do hardware, mudança das necessidades do usuário etc.
Apêndices	Deve fornecer informações detalhadas e específicas relacionadas à aplicação que está sendo desenvolvida. Exemplos de apêndices que podem ser incluídos são descrições de hardware e de banco de dados. Os requisitos de hardware definem as configurações mínima e ideal para o sistema. Os requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e os relacionamentos entre os dados.
Índice	Podem ser incluídos diversos índices para o documento. Assim como um índice alfabético normal, pode haver um índice dos diagramas, índice das funções etc.

Fonte:
Sommerville,
2007.



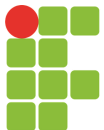
INSTITUTO FEDERAL
EDUCAÇÃO, CIÊNCIA
RIO GRANDE DO SUL
Campus Feliz

Engenharia de Requisitos



Validação de Requisitos

- Objetivo: mostrar que os requisitos realmente definem o sistema que o usuário deseja.
- “Está relacionada à descoberta de problemas com os requisitos” (Sommerville, 2007).
 - Erros em documentos de requisitos podem levar a custo excessivos de retrabalho quando são descobertos durante o desenvolvimento ou só quando o sistema está em operação.



Validação de Requisitos [1..3]

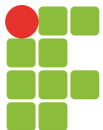
- Exemplos de verificações:
 - **Verificações de validade:** estudos e análises podem identificar que funções adicionais e diferentes são necessárias.
 - **Verificações de consistência:** os requisitos **não podem ser conflitantes** (não devem existir restrições ou descrições contraditórias para a mesma função do sistema).
 - **Verificações de completeza:** o documento deve incluir requisitos que **definam todas as funções e as restrições desejadas** pelo usuário do sistema.
 - **Verificações de realismo:** deve ser verificado se os requisitos **podem realmente ser implementados**. Deve-se levar em consideração a tecnologia existente, o orçamento e o prazo para desenvolver o sistema.
 - **Facilidade de verificação:** os requisitos devem ser **verificáveis**. Deve ser possível escrever testes que permitam verificar se o sistema atende aos requisitos definidos.

Fonte: Sommerville, 2007




Validação de Requisitos [2..3]

- Técnicas de validação:
 - **Revisões de requisitos:** os requisitos são analisados sistematicamente por uma equipe de revisores.
 - As revisões podem ser formais ou informais. Em ambas, os requisitos são discutidos com os *stakeholders* a fim de buscar anomalias e omissões.
 - **Prototipação:** um modelo executável do sistema é apresentado para usuários finais e clientes, para que eles possam verificar e verificar se atende às suas expectativas.
 - No caso de sistemas Web, pode-se fazer **protótipos de tela**.
 - **Geração de casos de teste:** ao elaborar casos de teste é possível revelar problemas nos requisitos. Se um teste for difícil ou impossível de definir, possivelmente também difícil de implementar.
 - O desenvolvimento de testes antes do código é prática da metodologia ágil XP.



Validação de Requisitos [3..3]

- Lista de controle para validação de requisitos:



Lista de controle para validação de requisitos

Muitas vezes é útil examinar cada requisito em relação a um conjunto de perguntas contidas em uma lista de controle. A seguir, um pequeno subconjunto daquelas que poderiam ser perguntadas:

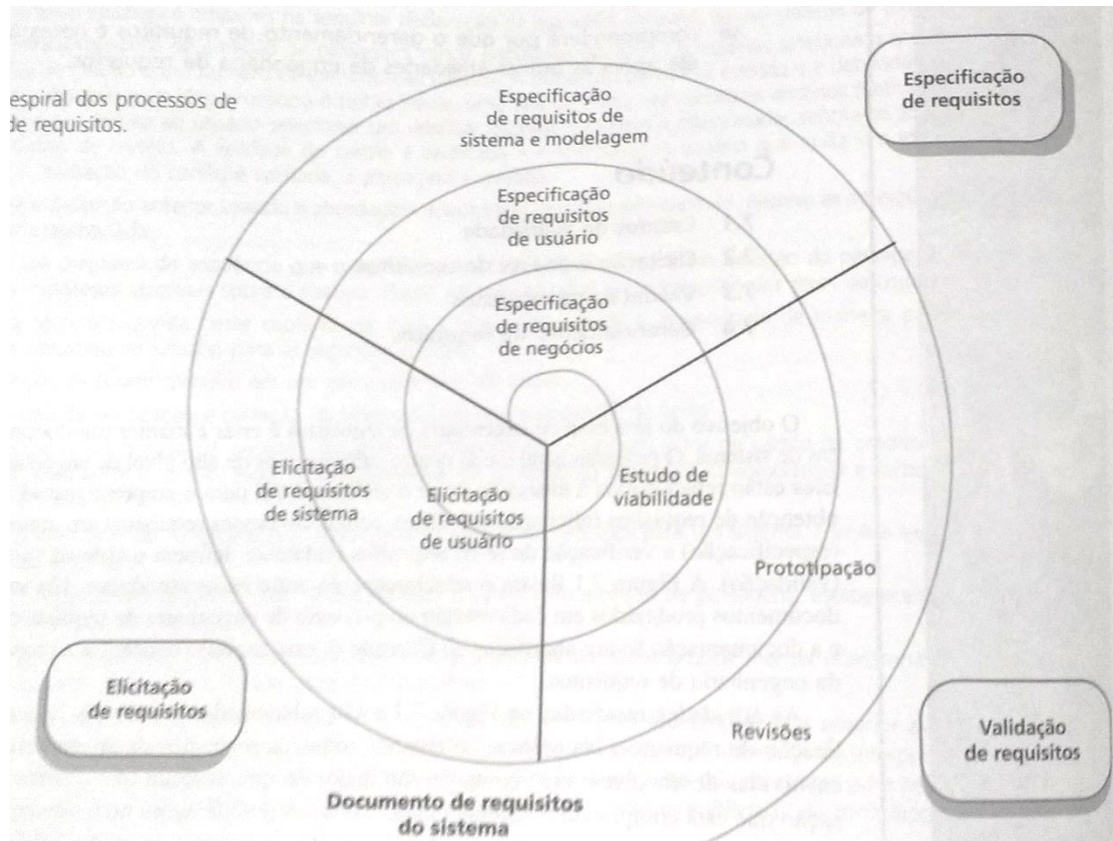
- Os requisitos estão declaradas de forma clara? Eles podem ser mal-interpretados?
- A fonte (por exemplo, uma pessoa, uma regulamentação, um documento) do requisito foi identificada? A declaração final do requisito foi examinada pela fonte original ou com ela?
- O requisito está limitado em termos quantitativos?
- Que outros requisitos se relacionam a este requisito? Eles estão claramente indicados por meio de uma matriz de referência cruzada ou algum outro mecanismo?
- O requisito viola quaisquer restrições do domínio do sistema?
- O requisito pode ser testado? Em caso positivo, podemos especificar testes (algumas vezes denominados critérios de validação) para testar o requisito?
- O requisito pode ser associado a qualquer modelo de sistema que tenha sido criado?
- O requisito pode ser associado aos objetivos globais do sistema/produto?
- A especificação é estruturada de forma que leve ao fácil entendimento, fácil referência e fácil tradução em artefatos mais técnicos?
- Criou-se um índice para a especificação?
- Os requisitos associados ao desempenho, ao comportamento e às características operacionais foram declarados de maneira clara? Quais requisitos parecem estar implícitos?

INFORMAÇÕES

Fonte: PRESSMAN, R. S. **Engenharia de software: uma abordagem profissional**. 7. ed. Rio de Janeiro: McGraw-Hill, 2011 (pág. 130).



Visão Geral da Engenharia de Requisitos [1..2]



Fonte: Sommerville, I. "Engenharia de Software", 8ª ed, 2007 (pág. 96).



Visão Geral da Engenharia de Requisitos [2..2]

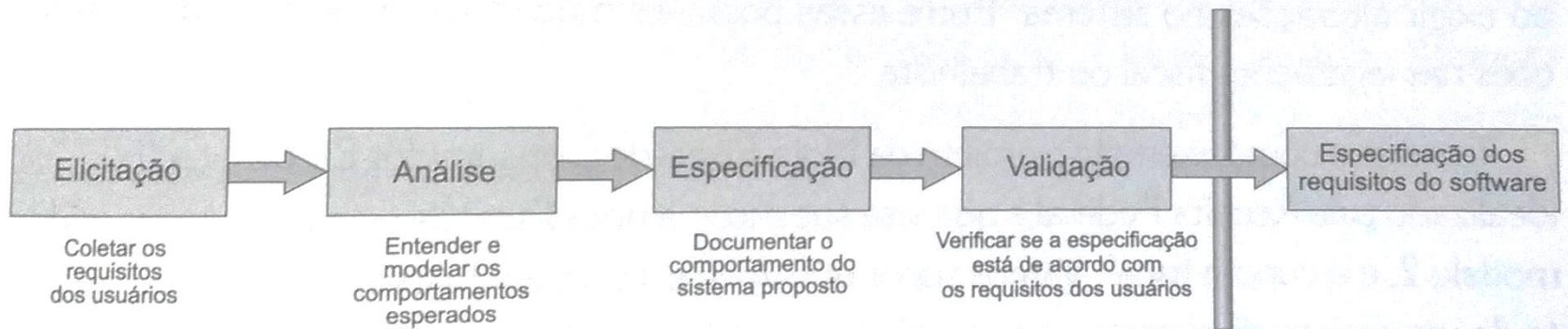


Figura 5.1 ■ Processo para a definição de requisitos.

Fonte: Pfleeger e Atlee (2010, p. 144).

Fonte: Pfleeger e Atlee (2010, p. 144) apud Alves (2017, p. 81)

#ClientesAmados



Atividade 3

Case

- Joaquim e Tereza estão inaugurando uma pousada em uma pequena cidade. Eles terão três quartos para hóspedes. Eles querem um sistema para gerenciar reservas. Quando um cliente potencial telefona para reservar, eles irão verificar a agenda e, se existir vaga, o sistema deve solicitar o nome do cliente, endereço, telefone, data de entrada e saída, concordância de preço, número do cartão de crédito e número do(s) apartamento(s). Cada quarto possui um preço e uma descrição. Não há frigobar nem serviços no quarto. As reservas são garantidas por meio do pagamento de uma diária realizada a partir de depósito bancário. Caso não seja indicado que o cliente efetuou o pagamento até três dias antes da data prevista de entrada, a reserva é automaticamente cancelada pelo sistema. Um relatório de reservas canceladas é gerado pelo sistema diariamente. Outros relatórios diários são o de reservas não pagas e o de reservas a serem efetivadas no dia.



Atividade 3

Plano de Entrevista

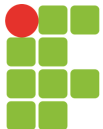
- No primeiro momento, é necessário avaliar o sistema sob a perspectiva do usuário, incluindo:
 - Propósito;
 - Funcionalidades básicas;
 - Restrições.
- Com base na avaliação prévia, elabore no mínimo 10 questões que deverão ser efetuadas ao especialista de domínio em uma primeira entrevista.



Atividade 3

Elicitação e Análise dos Requisitos

- Faça as suas perguntas para um colega de turma e anote as respostas (relatório da entrevista).
- Após, faça uma **lista de funcionalidades** esperadas do sistema.
- Classifique e priorize os requisitos de sua lista.



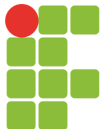
Atividade 3

Especificação dos Requisitos

- Especifique formalmente os requisitos identificados no formato discutido em aula:

O sistema deve permitir que <ATOR> faça <AÇÃO>.

- Utilize **regras relacionadas** para especificar os campos dos formulário e o que deve ser realizado em caso de sucesso ou falha.
- Faça os **protótipos de tela** mais relevantes.



Atividade 3

Validação dos Requisitos

- Apresente os requisitos e os protótipos de tela desenvolvidos para o colega entrevistado.
- Faça as alterações necessárias.



Bibliografia

- SOMMERVILLE, Ian. **Engenharia de Software**. 8. ed. São Paulo: Pearson Addison-Wesley. 2007.
- SOMMERVILLE, I. **Engenharia de software**. 9. ed. São Paulo: Pearson Education, 2011.
- PRESSMAN, R. S. **Engenharia de software: uma abordagem profissional**. 7. ed. Rio de Janeiro: McGraw-Hill, 2011.
- ALVES, William Pereira. **Análise e Projeto de Sistemas: estudo prático**. São Paulo: Érica, 2017.
- Slides gentilmente cedido pelo professor Rodrigo Noll, do IFRS/Campus Canoas.

Bibliografia adicional:

- Pressman, Roger. **Engenharia de Software**. São Paulo: McGraw-Hill, 2006, 6ª edição.

