



# Certified Professional for Requirements Engineering

Nível Fundamental  
Syllabus

Stan Bühne  
Martin Glinz  
Hans van Loenhoud  
Stefan Staal



## Termos de uso

1. Indivíduos e os provedores de treinamento podem utilizar este syllabus como base para cursos, desde que os direitos autorais sejam reconhecidos e incluídos no material do curso. Além disso, o Syllabus somente poderá ser utilizado para fins de publicidade mediante autorização por escrito do IREB e.V.
2. Qualquer indivíduo ou grupo de indivíduos podem utilizar este syllabus como fonte para artigos, livros ou outras publicações, desde que os direitos autorais dos autores e IREB e.V. sejam reconhecidos como fonte e proprietário deste documento em tais publicações.

© IREB e.V.

Todos os direitos reservados. Nenhuma parte desta publicação pode ser reproduzida, armazenada num sistema de recuperação ou transmitida de qualquer forma ou outro meio, eletrônico, mecânico, fotocópia, gravação ou outro, sem a autorização prévia escrita dos autores ou do IREB e.V.

## Agradecimentos

Este syllabus foi inicialmente criado em 2007 por Karol Frühauf, Emmerich Fuchs, Martin Glinz, Rainer Grau, Colin Hood, Frank Houdek, Peter Hruschka, Barbara Paech, Klaus Pohl, e Chris Rupp. Eles foram apoiados por Ian Alexander, Joseph Bruder, Samuel Fricker, Günter Halmans, Peter Jaeschke, Sven Krause, Steffen Lentz, Urte Pautz, Suzanne Robertson, Dirk Schüpferling, Johannes Staub, Thorsten Weyer, e Joy Beatty.

A versão 3 é uma grande revisão, criada por Stan Bühne, Martin Glinz, Hans van Loenhoud, e Stefan Staal. Eles foram apoiados por Karol Frühauf, Rainer Grau, Kim Lauenroth, Chris Rupp, e Camille Salinesi.

Durante esta revisão, o feedback foi fornecido por Xavier Franch, Karol Frühauf, Rainer Grau, Frank Houdek, e Thorsten Weyer. Feedbacks adicionais foram fornecidos por Wim Decoutere e Hans-Jörg Steffe.

As revisões foram feitas por Christoph Ebert, Barbara Paech, e Chris Rupp.

Aprovado para liberação em 22 de julho de 2020 pelo Conselho do IREB sob recomendação de Xavier Franch e Frank Houdek. Traduzido do Inglês por Ana Moreira, Carlos André e Silva, George Fialkovitz, Guilherme Simões e Stênio Viveiros.

Agradecemos a todos pelo seu envolvimento.

Copyright © 2007–2022 para este syllabus está com os autores listados acima. Os direitos foram transferidos para o IREB e.V. (International Requirements Engineering Board). (IREB), Karlsruhe, Alemanha.

## Prefácio

No verão de 2017, realizamos uma pesquisa para investigar a relevância da certificação Certified Professional for Requirements Engineering (CPRE) Foundation Level (versão 2.2). O objetivo da pesquisa foi obter feedback sobre a relevância prática da certificação no mercado a partir da perspectiva de provedores de treinamento, bem como de profissionais

certificados CPRE atuando como Engenheiros de Requisitos [MFeA2019]. A pesquisa apontou que o syllabus CPRE Nível Fundamental v2.2 em geral ainda atende às necessidades mais importantes do mercado e transmite o conhecimento da ER relevante aos candidatos. No entanto, recebemos feedback de que várias técnicas não são mais utilizadas na prática, enquanto outras voltadas ao desenvolvimento mais iterativo e adaptativo estão ausentes. Este feedback estava alinhado com a percepção do IREB sobre as mudanças na Engenharia de Requisitos (ER). Decidimos, portanto, criar uma grande revisão do syllabus CPRE Nível Fundamental, removendo o conteúdo desatualizado e adicionando novos elementos. A versão 3 do syllabus reflete o estado da ER contemporânea, cobrindo as abordagens orientadas ao planejamento e as ágeis, para especificar e gerenciar os requisitos.

Candidatos buscando a certificação CPRE de acordo com este syllabus precisam ter conhecimento básico em desenvolvimento de sistemas com abordagens orientadas ao planejamento e ágeis.

### Objetivo do documento

Este syllabus define o nível fundamental da certificação Certified Professional for Requirements Engineering estabelecida pelo International Requirements Engineering Board (IREB). O syllabus fornece aos provedores de treinamento a base para a criação de seus materiais de curso. Os participantes dos cursos podem utilizá-lo (além de subsídios adicionais na literatura especializada) como preparo para o exame de certificação.

### Conteúdo do Syllabus

O nível fundamental atende às necessidades de todos os envolvidos no tópico de Engenharia de Requisitos. Isto inclui engenheiro de requisitos, analista de negócios, analista de sistemas, product owner, gerente de produto, desenvolvedor, gerente de projeto ou TI e especialista em domínio.

Este syllabus e o respectivo manual utilizam a abreviatura "ER" para Engenharia de Requisitos.

### Escopo do conteúdo

O CPRE Nível Fundamental comunica os princípios básicos que são válidos para qualquer tipo de sistema (p. ex., aplicativos móveis, sistemas de informação ou ciberfísicos). Além disso, o CPRE Nível Fundamental não assume um processo de desenvolvimento específico, nem está apoiado em um domínio de aplicação específico. Os provedores de treinamento podem oferecer treinamentos concentrados em tipos específicos de sistemas, processos ou domínios, desde que os objetivos deste syllabus sejam totalmente cobertos.

### Nível de Detalhe

O nível de detalhe deste syllabus permite que o ensino e os exames sejam consistentes internacionalmente. Para atingir este objetivo, o syllabus contém:

- Objetivos educacionais gerais;
- Conteúdos com a descrição dos objetivos educacionais
- Referências a outras literaturas (quando necessário).

## Formulação sensível a gênero

Abstivemo-nos deliberadamente de usar formulações específicas de gênero neste documento. No IREB apoiamos formulações sensíveis ao gênero. Todavia, temos a necessidade de formular questões complexas de forma a que possam ser facilmente compreendidas.

Textos que exigem uma forma feminina e masculina seriam menos legíveis e, portanto, mais difíceis de entender. No entanto, o objetivo deste documento é apresentar e comunicar o conteúdo de forma precisa e clara. Visto que queremos ajudar os leitores a manter o foco no conteúdo, usamos deliberadamente apenas a forma masculina neste documento. Isso não deve ser interpretado como uma expressão de falta de respeito.

## Objetivos Educacionais / Níveis de Conhecimento Cognitivo

A todos os módulos e objetivos educacionais deste syllabus é atribuído um nível cognitivo. Os seguintes níveis são usados:

- **L1: Conhecer** (descrever, enumerar, caracterizar, reconhecer, nomear, lembrar, ...) – lembrar ou recuperar material previamente aprendido.
- **L2: Compreender** (explicar, interpretar, completar, resumir, justificar, classificar, comparar, ...) – entender/construir o significado de determinado material ou situações.
- **L3: Aplicar** (especificar, escrever, desenvolver, implementar, ...) – aplicar conhecimento e habilidade em dadas situações.

Os níveis mais altos incluem os mais baixos. Observe que todos os termos do glossário designados como fundamentais devem ser conhecidos (L1), mesmo que não sejam explicitamente mencionados nos objetivos educacionais. O glossário está disponível para download na página do IREB em <https://www.ireb.org/downloads/#cpre-glossary>

## Estrutura do Syllabus

O syllabus é composto por sete capítulos principais. Cada capítulo cobre uma unidade educacional (UE). Os títulos dos capítulos principais contêm seus níveis cognitivos, que são os níveis mais altos dos seus subcapítulos. A duração indica o tempo que um treinamento deve investir nesse capítulo. As empresas de treinamento podem atribuir mais tempo, mas devem manter as proporções entre os UE. Os termos importantes utilizados em um capítulo estão listados no seu início.

### Exemplo

UE 4 Práticas para a elaboração de requisitos (L3)

Duração: 4 horas 30 minutos

Termos: Fonte de requisitos, limite do sistema, contexto do sistema, elicitação de requisitos, validação de requisitos, stakeholder, modelo de Kano, conflito e resolução de conflito.

Este exemplo mostra que o Capítulo 4 contém objetivos educacionais no nível L3 e que quatro horas e meia são destinadas a ensinar o material deste capítulo.

Cada capítulo contém subcapítulos. Seus títulos também contêm o nível cognitivo de seu conteúdo.

Os objetivos educacionais (OE) são enumerados antes do texto real. A numeração mostra o subcapítulo a que pertencem. Por exemplo, o objetivo educacional OE 4.2.1 é descrito no subcapítulo 4.2.

### Ordem dos Tópicos no Syllabus

A ordem dos capítulos desse syllabus constitui uma ordem lógica de tópicos. No entanto, os tópicos não precisam ser ensinados nesta ordem. Os provedores de treinamento são livres para usar o material em qualquer ordem (incluindo a intercalação de tópicos de diferentes UE) que considerem adequada no contexto de seu curso e que se enquadre em seu conceito didático.

### O Exame

Este syllabus é a base para o exame de certificação CPRE Nível Fundamental.



O conteúdo de uma questão pode abranger material de diversos capítulos do syllabus. Todos os capítulos (UE 1 até UE 7) deste syllabus são examináveis.

O formato das questões é de múltipla escolha.

Os exames podem ocorrer imediatamente após um treinamento, ou independentemente dos cursos (p. ex., em um centro de exames). A lista de organizações de certificação licenciadas pelo IREB encontra-se no site <https://www.ireb.org>.

## Histórico de versões

Versão	Data	Comentário
2.1.4	9 de novembro de 2011	Versão Inicial baseada no Syllabus original do IREB em alemão
3.0.0	1 de abril de 2021	Grande atualização refletindo a situação atual da ER, cobrindo as abordagens planejadas e as ágeis para especificar e gerenciar os requisitos.
3.1.0	1 de setembro de 2022	<p>Correções de digitação e referência no documento para melhorar a leitura.</p> <p>UE 1:</p> <ul style="list-style-type: none"><li>▪ Movido o objetivo de estudo OE 1.3.2 para OE 1.1.2.</li><li>▪ Atualização dos OE 1.2.2 e 1.3.1</li></ul> <p>UE 3</p> <ul style="list-style-type: none"><li>▪ Atualização do objetivo de estudo OE 3.1.3</li><li>▪ UE 3.1.2: Reformulado o parágrafo sobre os níveis de abstração</li><li>▪ UE 3.4: Todos os tipos de modelos que não precisam ser aplicados no nível fundamental foram transferidos para uma nova subseção 3.4.6</li><li>▪ UE 3.6: Título atualizado para "Documentos de requisitos e estruturas de documentação".</li></ul> <p>UE 4</p> <ul style="list-style-type: none"><li>▪ UE 4.1: A descrição de stakeholder e dos papéis dos stakeholders tornou-se mais precisa.</li><li>▪ UE 4.2: A introdução e a justificativa para as técnicas de geração de idéias e design, foi atualizada para ser mais precisa.</li><li>▪ UE 4.3: Título atualizado para "Resolução de Conflitos relativos aos Requisitos".</li></ul> <p>UE 6</p> <ul style="list-style-type: none"><li>▪ OE 6.3.1 e 6.5.2 foram ligeiramente modificados</li></ul>
3.1.1	1º de janeiro de 2024	Novo design corporativo e pequenas correções
3.2.0	26 de fevereiro de 2024	Correções de layout para o novo design corporativo
3.2.1	18 de março de 2024	UE 1.2: Correção de um problema menor de tradução

# Conteúdo

<b>1</b>	<b>Introdução e Visão Geral da Engenharia de Requisitos (L2)</b>	<b>10</b>
1.1	Engenharia de Requisitos: O quê (L1) .....	10
1.2	Engenharia de Requisitos: Porquê (L2) .....	11
1.3	Engenharia de Requisitos: Onde (L2) .....	11
1.4	Engenharia de Requisitos: Como (L1) .....	11
1.5	O Papel e Tarefas de um Engenheiro de Requisitos (L1) .....	12
1.6	O que aprender sobre Engenharia de Requisitos (L1) .....	12
<b>2</b>	<b>Princípios Fundamentais da Engenharia de Requisitos (L2)</b>	<b>13</b>
2.1	Visão Geral dos Princípios (L1) .....	13
2.2	Os Princípios Explicados (L2) .....	13
<b>3</b>	<b>Produtos de Trabalho e Práticas de Documentação (L3) ....</b>	<b>18</b>
3.1	Produtos de Trabalho na Engenharia de Requisitos (L2) .....	19
3.1.1	Características dos Produtos de Trabalho (L1) .....	19
3.1.2	Níveis de Abstração (L2) .....	20
3.1.3	Nível de detalhe (L2) .....	20
3.1.4	Aspectos a considerar nos Produtos de Trabalho (L1) .....	20
3.1.5	Diretrizes Gerais de Documentação (L1) .....	21
3.1.6	Planejar os produtos de trabalho a serem utilizados (L1) .....	21
3.2	Produtos de Trabalho com Base em Linguagem Natural (L2) .....	22
3.3	Produtos de trabalho baseados em template (L3) .....	23
3.4	Produtos de Trabalho Baseados em Modelos (L3) .....	23
3.4.1	O Papel dos Modelos na Engenharia de Requisitos (L2) .....	24
3.4.2	Modelagem de contexto (L2) .....	25
3.4.3	Modelagem de estruturas e dados (L3) .....	25
3.4.4	Modelagem de funções e fluxo (L3) .....	26
3.4.5	Modelagem de Estado e Comportamento (L2) .....	26

3.4.6	Outros tipos de modelos em Engenharia de Requisitos (L1).....	26
3.5	Glossários (L2).....	27
3.6	Documentos de Requisitos e Estruturas de Documentação (L2) .....	27
3.7	Protótipos na Engenharia de Requisitos (L1) .....	28
3.8	CrITÉrios de Qualidade para Produtos de trabalho e requisitos (L1)	29
4	Práticas de Elaboração de Requisitos (L3) .....	31
4.1	Fontes de requisitos (L3).....	31
4.2	Elicitação de requisitos (L2) .....	33
4.3	Resolução de Conflitos relativos aos Requisitos (L2) .....	34
4.4	Validação dos Requisitos (L2) .....	35
5	Processo e Estrutura de Trabalho (L3) .....	37
5.1	Fatores de Influência (L2).....	37
5.2	Vertentes de Processos de Engenharia de Requisitos (L2) .....	38
5.3	Configuração de um Processo de Engenharia de Requisitos (L3) .....	40
6	Práticas de Gestão de Requisitos (L2) .....	42
6.1	O que é o Gerenciamento de Requisitos? (L1) .....	42
6.2	Gestão do Ciclo de Vida (L2) .....	42
6.3	Controle de Versão (L2).....	43
6.4	Configurações e baselines (L1) .....	43
6.5	Atributos e Visualizações(L2) .....	43
6.6	Rastreabilidade (L1).....	44
6.7	Lidando com mudanças (L1) .....	44
6.8	Priorização (L1).....	45
7	Ferramentas de Suporte (L2) .....	46

7.1	Ferramentas na Engenharia de Requisitos (L1) .....	46
7.2	Introduzindo Ferramentas (L2) .....	47
	Referências .....	48

# 1 Introdução e Visão Geral da Engenharia de Requisitos (L2)

Objetivo: Conhecer o que é a ER e compreender o seu valor

Duração: 1 hora

Termos: Requisito, especificação de requisitos, Engenharia de Requisitos (ER), stakeholder, sistema, Engenheiro de Requisitos

## Objetivos Educacionais

- OE 1.1.1 Relembrar a terminologia fundamental (L1)
- OE 1.1.2 Entender os diferentes tipos de requisitos (L2)
- OE 1.2.1 Explicar o valor da ER (L2)
- OE 1.2.2 Enumerar sintomas de uma ER (L1) inadequada
- OE 1.3.1 Saber onde a ER pode ser aplicada e onde os requisitos ocorrem (L1)
- OE 1.4.1 Conhecer as principais tarefas da ER e saber que é necessário adaptar um processo da ER para executá-las (L1)
- OE 1.5.1 Caracterizar o papel e tarefas de um Engenheiro de Requisitos (L1)
- OE 1.6.1 Relembrar o que um Engenheiro de Requisitos precisa aprender (L1)

## 1.1 Engenharia de Requisitos: O quê (L1)

As pessoas e organizações têm desejos e necessidades de coisas a serem construídas ou de serem evoluídas. Chamamos essas necessidades de *requisitos*.

As coisas a serem construídas ou evoluídas podem ser:

- *Produtos* fornecidos aos clientes
- *Serviços* disponibilizados aos clientes
- Qualquer *entregável* como dispositivos, procedimentos ou ferramentas que ajudem pessoas e organizações a alcançar um objetivo específico
- *Composições* ou *componentes* de produtos, serviços ou outros entregáveis

Tudo isso pode ser considerado como *sistema*. Neste syllabus, usamos o termo *sistema* para denotar todo o tipo de coisas para a qual os stakeholders têm requisitos. Os *stakeholders* são pessoas ou organizações que influenciam os requisitos de um sistema ou são impactados por esse sistema.

O objetivo da ER é especificar e gerenciar os requisitos do sistema de forma que, quando implementado e instalado, satisfaça os desejos e necessidades dos stakeholders.

Na ER, distinguimos três tipos de requisitos [Glin2020]:

- *Requisitos funcionais* dizem respeito a um resultado ou comportamento que deve ser fornecido por uma função de um sistema. Isso inclui requisitos de dados ou de interação de um sistema com seu ambiente.
- *Requisitos de qualidade* dizem respeito a questões de qualidade que não são cobertas por requisitos funcionais, tais como desempenho, disponibilidade, segurança ou confiabilidade.

- *Restrições* são requisitos que limitam o espaço da solução além do que é necessário para atender aos requisitos funcionais e de qualidade.

## 1.2 Engenharia de Requisitos: Porquê (L2)

A ER adequada agrega *valor* no processo de desenvolvimento e evolução de um sistema:

- Reduzindo o risco de desenvolver o sistema errado
- Melhorando a compreensão do problema
- Sendo base na estimativa de esforço e custo de desenvolvimento
- Sendo pré-requisito no teste de sistema

Sintomas típicos de uma ER inadequada são requisitos não especificados, sem clareza ou incorretos. Isto é particularmente devido a:

- Pressa para construção do sistema
- Problemas de comunicação entre stakeholders
- O pressuposto de que os requisitos são evidentes por si mesmos
- Capacitação e competências inadequadas na ER

## 1.3 Engenharia de Requisitos: Onde (L2)

A ER pode ser aplicada aos requisitos de qualquer sistema. No entanto, a aplicação dominante da ER hoje é basicamente em sistemas nas quais o software desempenha um papel importante. Tais sistemas normalmente consistem em componentes de software, elementos físicos e organizacionais.

Os requisitos podem ocorrer como:

- *Requisitos do sistema* – o que um sistema deve fazer
- *Requisitos de stakeholders* – o que os stakeholders querem de sua perspectiva
- *Requisitos de usuário* – o que os usuários querem de sua perspectiva
- *Requisitos de domínio* – propriedades de domínio necessárias
- *Requisitos de negócio* – metas de negócio, objetivos, e necessidades de uma organização

## 1.4 Engenharia de Requisitos: Como (L1)

As principais tarefas na ER são a elicitacão (4.2), documentação (3), validação (4.4) e gerenciamento (6) de requisitos. O uso de ferramentas (7) pode ajudar a realizar estas tarefas. A análise de requisitos e a resolução de conflitos de requisitos (4.3) são considerados parte da elicitacão. Para executar corretamente as atividades da ER, um processo da ER deve ser adaptado a partir de um leque de possibilidades (5).

## 1.5 O Papel e Tarefas de um Engenheiro de Requisitos (L1)

Engenheiro de Requisitos não é um cargo, mas um *papel* desempenhado por pessoas que:

- Elicitam, documentam, validam e/ou gerenciam os requisitos como parte de suas funções.
- Têm um profundo conhecimento em ER.
- Pode preencher a lacuna entre o problema e possíveis soluções.

Na prática, analistas de negócios, especialistas em aplicações, product owners, engenheiros de sistemas e até desenvolvedores atuam como Engenheiro de Requisitos.

## 1.6 O que aprender sobre Engenharia de Requisitos (L1)

Este syllabus abrange as habilidades fundamentais que um Engenheiro de Requisitos deve aprender. Abrange os princípios fundamentais da ER (2), como documentar de várias formas os requisitos (3), elaborar requisitos com várias práticas (4), como definir e trabalhar com processos da ER adequados (5), como gerenciar os requisitos existentes (6), e como usar ferramenta de apoio (7).

# 2 Princípios Fundamentais da Engenharia de Requisitos (L2)

Objetivo: Conhecer e entender os princípios da ER

Duração: 1 hora e 30 minutos

Termos: Contexto, requisito, Engenharia de Requisitos (RE), stakeholder, entendimento compartilhado, validação

## Objetivos Educacionais

OE 2.1.1 Enumerar os princípios da ER (L1)

OE 2.2.1 Relembrar os termos associados com os princípios (L1)

OE 2.2.2 Explicar os princípios e as razões pelas quais são importantes (L2)

## 2.1 Visão Geral dos Princípios (L1)

A ER é regida por um conjunto de princípios fundamentais aplicados às tarefas, atividades e práticas. Os nove princípios seguintes formam a base para as práticas apresentadas nos capítulos subsequentes deste syllabus.

1. Orientação para o valor: Os requisitos são um meio para atingir o fim, não o fim em si
2. Stakeholders: ER é a satisfação dos desejos e necessidades dos stakeholders
3. Entendimento compartilhado: O sucesso no desenvolvimento de sistemas é impossível sem uma base comum
4. Contexto: Sistemas não podem ser compreendidos isoladamente
5. Problema – Requisito – Solução: Uma trinca inevitavelmente entrelaçada
6. Validação: Os requisitos não validados são inúteis
7. Evolução: A mudança de requisitos não é um acidente, mas o caso normal
8. Inovação: Mais do mesmo não é suficiente
9. Trabalho sistemático e disciplinado: Não podemos fazer sem a ER

## 2.2 Os Princípios Explicados (L2)

**Princípio 1 - Orientação para o valor: Requisitos são um meio para o fim, não o fim em si**

O valor de um requisito é igual ao seu benefício menos o custo para elicitar, documentar, validar e geri-lo. O benefício de um requisito é o grau com o qual ele contribui para:

- Construir sistemas que satisfaçam os desejos e necessidades de stakeholders.
- Reduzir o risco de falha e do retrabalho no desenvolvimento do sistema.

### Princípio 2 - Stakeholders: ER é a satisfação dos desejos e necessidades dos stakeholders

Como a ER trata de compreender os desejos e necessidades dos stakeholders, lidar devidamente com eles é uma tarefa essencial da ER. Cada stakeholder tem um papel no contexto do sistema a ser construído, por exemplo, usuário, cliente, operador ou regulador. Um stakeholder também pode ter mais de um papel simultaneamente. Para papéis de stakeholders com muitos indivíduos ou quando estes são desconhecidos, a descrição fictícia conhecida como *personas* pode ser definida como substituta. Não é suficiente considerar apenas os requisitos dos usuários finais ou clientes. Fazer isso significaria que poderíamos negligenciar requisitos críticos de outros stakeholders. Os usuários que proveem feedback sobre o uso do sistema também são considerados stakeholders.

Os stakeholders podem ter necessidades e pontos de vista diferentes, o que resulta em requisitos conflitantes. É uma tarefa da ER identificar e resolver tais conflitos.

Envolver as pessoas certas nos papéis relevantes de stakeholder é crucial para o sucesso da ER. As práticas para identificar, priorizar e trabalhar com stakeholders são descritas em 4.

### Princípio 3 - Entendimento comum: O desenvolvimento bem-sucedido de sistemas é impossível sem uma base comum

A ER cria, fomenta e assegura o entendimento comum entre e dentre as partes envolvidas: stakeholders, Engenheiros de Requisitos e desenvolvedores. Há duas formas de entendimento comum:

- *Entendimento comum explícito* (alcançada por requisitos documentados e acordados)
- *Entendimento comum implícito* (baseada no conhecimento comum de necessidades, visões, contexto etc.)

O conhecimento de domínio, colaboração prévia bem-sucedida, cultura, valores compartilhados e confiança mútua são facilitadores, enquanto a distância geográfica, terceirização ou grandes equipes com alta rotatividade são obstáculos.

As práticas comprovadas para alcançar o entendimento comum incluem a criação de glossários (3.5), protótipos (3.7), ou o uso de um sistema existente como ponto de referência. As práticas para avaliar o entendimento comum incluem exemplos de resultados esperados,

uso de protótipos ou estimativa de custo da implementação de um requisito. A prática mais importante para reduzir o impacto dos mal-entendidos é o uso de um processo com pequenos ciclos de feedback (5).

#### Princípio 4 - Contexto: Sistemas não podem ser entendidos isoladamente

Os sistemas estão embutidos em um *contexto*. Sem entender esse contexto, é impossível especificar um sistema adequadamente. Na ER, o contexto de um sistema é a parte do ambiente de um sistema, que é relevante para a compreensão do sistema e seus requisitos. O *limite do sistema* é a fronteira entre o sistema e o seu contexto adjacente. Inicialmente, o limite do sistema pode não ser claro, podendo mudar com o tempo.

Esclarecer os limites do sistema e definir as interfaces externas entre o sistema e os elementos de contexto com os quais interage são tarefas genuínas da ER. Ao mesmo tempo, o *escopo* do sistema, ou seja, a gama de coisas que podem ser moldadas e projetadas ao desenvolver o sistema, precisa ser determinado. Também precisamos considerar o chamado *limite do contexto* que separa a parte do ambiente que é relevante para a ER do resto do mundo.

Ao especificar um sistema, não é suficiente considerar apenas os requisitos dentro dos limites do sistema. A ER também deve considerar:

- Mudanças no contexto que podem impactar os requisitos do sistema.
- Requisitos do mundo real relevantes para o sistema (e como mapeá-los aos requisitos do sistema).
- Premissas de contexto que devem ser mantidas para que o sistema funcione e atenda aos requisitos do mundo real.

#### Princípio 5 - Problema - Requisito - Solução: Uma trinca inevitavelmente entrelaçada

Ocorre um *problema* quando os stakeholders não estão satisfeitos com a situação atual. Os *requisitos* capturam o que os stakeholders precisam para se livrarem ou simplificarem um problema. Um sistema sócio-técnico que satisfaça estes requisitos constitui uma *solução*.

Os problemas, requisitos e soluções não ocorrem necessariamente nesta ordem. Ideias de soluções podem criar necessidades de usuários que devem ser trabalhadas em requisitos e implementadas em uma solução real. Este é um típico caso de inovação.

- Problemas, requisitos e soluções estão intimamente entrelaçados: não podem ser enfrentados isoladamente.

- No entanto, os Engenheiros de Requisitos procuram separar problemas, requisitos e soluções uns dos outros, tanto quanto possível, ao pensar, comunicar e documentar. Separar as preocupações tornam as tarefas da ER mais fáceis de lidar.

### Princípio 6 - Validação: Os requisitos não validados são inúteis

Eventualmente, temos de validar se o sistema implantado satisfaz os desejos e necessidades dos stakeholders. A fim de controlar desde o início o risco dos stakeholders insatisfeitos, a validação dos requisitos deve começar já durante a ER. Temos de checar se:

- Um acordo sobre os requisitos foi alcançado entre os stakeholders,
- Os desejos e necessidades dos stakeholders são devidamente cobertos pelos requisitos,
- As premissas de contexto (ver Princípio 4 acima) são razoáveis.

As práticas para validação dos requisitos são discutidas em 4.4.

### Princípio 7 - Evolução: A mudança de requisitos não é um acidente, mas o caso normal

Os sistemas e seus requisitos estão sujeitos a *evolução*. Isto significa que eles mudam constantemente. As solicitações para modificar um requisito, ou um conjunto de requisitos, de um sistema podem ser causadas por:

- Processos de negócio alterados
- Concorrentes que lançam novos produtos ou serviços
- Clientes que mudam suas prioridades ou opiniões
- Mudanças na tecnologia
- Alterações de leis ou regulamentos
- Feedback de usuários do sistema pedindo recursos novos ou alterados

Além disso, os requisitos podem mudar devido ao feedback dos stakeholders ao validá-los, devido à detecção de falhas em requisitos elicitados anteriormente, ou devido a mudanças nos requisitos por stakeholders.

Como consequência, os Engenheiros de Requisitos devem buscar dois objetivos contraditórios:

- Permitir mudança nos requisitos
- Manter os requisitos estáveis

Detalhes sobre como conseguir isso são discutidos em 6.7.

## Princípio 8 - Inovação: Mais do mesmo não é o suficiente

Ao dar aos stakeholders exatamente o que querem, perde-se a oportunidade de construir sistemas que satisfaçam melhor as suas necessidades do que eles mesmo esperam. Uma boa ER esforça-se não só para satisfazer os stakeholders, mas deixá-los felizes, motivados ou seguros. É disto que se trata a inovação.

A ER dá forma a sistemas inovadores:

- Em pequena escala, esforçando-se por novas e excitantes características e facilidade de uso.
- Em larga escala, lutando por ideias disruptivas.

Na 4.2 são discutidas várias técnicas para fomentar a inovação na ER.

## Princípio 9 - Trabalho sistemático e disciplinado: Não podemos desenvolver sem a ER

Existe a necessidade de empregar processos e práticas adequadas para sistematicamente elicitar, documentar, validar e gerenciar os requisitos, independentemente do processo de desenvolvimento em uso. Mesmo quando um sistema é desenvolvido de forma ad-hoc, uma abordagem sistemática e disciplinada da ER melhora a qualidade do sistema resultante.

Não há um único processo ou prática na ER que funcione bem em cada situação ou, pelo menos, na maioria das situações. Trabalho sistemático e disciplinado significa que os Engenheiros de Requisitos:

- Adaptam seus processos e práticas ao problema, contexto e ambiente em questão.
- Não utilizam sempre o mesmo processo e conjunto de práticas.
- Não reutilizam processos e práticas de trabalhos anteriores de sucesso da ER sem reflexão.

Para cada ação da ER, devem ser escolhidos processos, práticas e produtos de trabalho que melhor se ajustem à situação. Os detalhes são elaborados em 3, 4, 5 e 6.

# 3 Produtos de Trabalho e Práticas de Documentação (L3)

Objetivo: Entender o papel fundamental dos produtos de trabalho na ER e ser capaz de criá-los

Duração: 7 horas

Termos: Produto de trabalho, produtos de trabalho em linguagem natural, produtos de trabalho baseados em templates, produtos de trabalho baseados em modelos, glossário, critérios de qualidade, especificação de requisitos

## Objetivos Educacionais

- OE 3.1.1 Saber as características dos produtos de trabalho ER e enumerar os tipos de produtos de trabalho frequentemente utilizados (L1)
- OE 3.1.2 Saber o uso de cada produto de trabalho e conhecer o seu tempo de vida útil (L1)
- OE 3.1.3 Explicar os diferentes níveis de abstração para os requisitos, incluindo como escolher os níveis apropriados e níveis de detalhe (L2)
- OE 3.1.4 Conhecer os aspectos a considerar nos produtos de trabalho e suas relações (L1)
- OE 3.1.5 Nomear as diretrizes gerais de documentação (L1)
- OE 3.1.6 Descrever por que vale a pena planejar o uso dos produtos de trabalho (L1)
- OE 3.2.1 Conhecer os produtos de trabalho em linguagem natural, suas vantagens e desvantagens (L1)
- OE 3.2.2 Explicar as regras para escrever bons requisitos em linguagem natural (L2)
- OE 3.3.1 Conhecer as categorias de produtos de trabalho baseados em template, suas vantagens e desvantagens (L1)
- OE 3.3.2 Especificar um requisito individual e uma história de usuário usando um template de frase (L3)
- OE 3.3.3 Especificar um caso de uso usando um template de formulário (L3)
- OE 3.4.1 Entender o papel, a finalidade e o uso de modelos na ER (L2)
- OE 3.4.2 Entender as vantagens e limitações da modelagem na ER (L2)
- OE 3.4.3 Conhecer os termos: modelo, linguagem de modelo, modelo de atividade, diagrama de atividade, modelo de classe, diagrama de classe, modelo de contexto, diagrama de contexto, modelo de domínio, modelo de objetivo, modelo de interação, modelo de processo, diagrama de sequência, gráfico de estados, máquina de estados, diagrama de máquina de estados, caso de uso, diagrama de caso de uso (L1)
- OE 3.4.4 Entender como selecionar um modelo apropriado para especificar requisitos em uma determinada situação (L2)
- OE 3.4.5 Compreender e interpretar modelos simples, escritos em UML onde aplicável, dos seguintes tipos: modelos de contexto, de domínio, de classe, de atividade, de processos; casos de uso; e diagramas de caso de uso, e de estado (L2)
- OE 3.4.6 Especificar um modelo simples de dados de um sistema ou os objetos de um domínio usando um diagrama de classes UML (L3)
- OE 3.4.7 Especificar uma função simples do sistema ou processo de negócio usando um diagrama de atividade UML (L3)
- OE 3.5.1 Explicar a função dos glossários e como criar um (L2)
- OE 3.6.1 Conhecer os documentos de especificação de requisitos comumente usados (L1)
- OE 3.6.2 Explicar a função e os critérios de construção das estruturas de documentação (L2)
- OE 3.7.1 Conhecer os diferentes tipos de protótipos e suas funções (L1)
- OE 3.8.1 Conhecer os critérios de qualidade para requisitos individuais (L1)

## 3.1 Produtos de Trabalho na Engenharia de Requisitos (L2)

Um produto de trabalho é um resultado intermediário ou final gerado em um processo de trabalho. Há uma variedade de produtos de trabalho na ER, desde esboços gráficos de curta duração, passando por coleções de histórias de usuários em evolução, até documentos de especificação de requisitos contratuais formais com centenas de páginas.

### 3.1.1 Características dos Produtos de Trabalho (L1)

Os produtos de trabalho são caracterizados por sua finalidade, representação, tamanho e tempo de vida útil. Os seguintes produtos de trabalho ocorrem frequentemente na prática para seus propósitos. Note que um produto de trabalho pode conter outros.

- Produtos de trabalho para um único requisito incluem requisitos individuais e histórias de usuários.
- Produtos de trabalho para um conjunto de requisitos incluem casos de uso, modelos gráficos (3.4), descrições de tarefas, descrições de interfaces externas, e épicos.
- Os produtos de trabalho que constituem documentos abrangentes ou estruturas de documentação incluem especificações de requisitos de sistema, backlogs de produtos e backlog de sprint, e mapas de histórias.
- Outros produtos de trabalho incluem glossários, notas textuais, esboços gráficos e protótipos.

Produtos de trabalho podem ser *representados* de várias formas:

- Baseado em linguagem natural (3.2)
- Baseado em templates (3.3)
- Baseado em modelos (3.4)
- Outras representações, como desenhos ou protótipos (3.7)

A maioria dos produtos de trabalho são *armazenados* digitalmente em arquivos, bancos de dados ou ferramentas da ER. Produtos de trabalho informais e temporários também podem ser armazenados em outros meios, por exemplo, anotações ou cartões em um quadro Kanban.

Observando a duração de vida dos produtos de trabalho, distinguimos três categorias:

- *Produtos de trabalho temporário*: criados para ajudar a comunicação e criar um entendimento comum.
- *Produtos de trabalho evolutivos*: crescem a cada iteração; necessitam de alguns metadados (6.5), e podem usar controle de alterações.
- *Produtos de trabalho duráveis*: foram arquivados ou implantados; precisam de um conjunto completo de metadados (6.5); a gestão de mudança deve ser seguida (6.3, 6.4).

Um produto de trabalho temporário pode ser convertido em um produto de trabalho em evolução (mantendo e adicionando metadados). Analogamente, um produto de trabalho temporário ou evolutivo pode se tornar um durável ao ser arquivado ou implantado.

### 3.1.2 Níveis de Abstração [L2]

Os requisitos geralmente existem em *níveis diferentes de abstração* desde, por exemplo, requisitos de alto nível para um novo processo de negócio até requisitos em um nível detalhado, como a reação de um componente de software específico a um evento excepcional.

A escolha do nível adequado de abstração depende do tema a ser especificado e da finalidade da especificação. É importante, contudo, não misturar requisitos que estão em diferentes níveis de abstração. Em produtos de trabalho de pequeno e médio porte, os requisitos devem estar aproximadamente no mesmo nível de abstração.

Em grandes produtos de trabalho, como uma especificação de requisitos de sistema, os requisitos em diferentes níveis de abstração devem ser mantidos separados, por uma estrutura de especificação adequada (3.6). Um requisito com um alto nível de abstração pode ser refinado em vários requisitos detalhados em níveis mais baixos e concretos.

Quando os requisitos de alto nível empresarial ou de stakeholders são expressos em produtos de trabalho duráveis – tais como especificações de requisitos empresariais, de stakeholders, ou documentos de visão – eles precedem a especificação dos requisitos do sistema. Em outros ambientes, os requisitos de negócio, de stakeholders e de sistema podem evoluir conjuntamente.

### 3.1.3 Nível de detalhe [L2]

O nível de *detalhe* que os requisitos serão especificados depende de vários fatores, em particular:

- O problema e o contexto de desenvolvimento
- O grau de entendimento comum do problema
- O grau de liberdade deixado aos designers e programadores
- Disponibilidade de feedback rápido dos stakeholders durante a concepção e implementação
- Custo vs. valor de uma especificação detalhada
- Normas impostas e restrições regulatórias

Quanto maior o nível de detalhe nos requisitos especificados, menor será o risco de eventualmente obter algo inesperado ou não especificado. No entanto, o custo da especificação aumenta à medida que o nível de detalhe aumenta.

### 3.1.4 Aspectos a considerar nos Produtos de Trabalho [L1]

Ao especificar requisitos em produtos de trabalho, diferentes aspectos precisam ser considerados.

1. Os requisitos são classificados pelo seu tipo (1.1) em:
  - a) Requisitos funcionais
  - b) Requisitos de qualidade
  - c) Restrições
2. Os requisitos funcionais concentram-se em diferentes aspectos da funcionalidade de um sistema:
  - a) Estrutura e dados
  - b) Função e fluxo
  - c) Estado e comportamento
3. Finalmente, os requisitos só serão entendidos no contexto (Princípio 4 em 2):
  - a) *Contexto do sistema*, incluindo atores externos
  - b) *Limites do sistema e interfaces externas*

Há muitas relações e dependências entre os aspectos mencionados. Por exemplo, uma solicitação emitida por um usuário (contexto) pode levar a uma transição de estado (estado e comportamento) que inicia ações em cadeia (função e fluxo) que requerem dados (estrutura e dados) para fornecer um resultado ao usuário (contexto) dentro de um intervalo de tempo (qualidade).

Alguns produtos de trabalho focam em um aspecto específico e abstraem os outros aspectos. Este é particularmente o caso dos modelos de requisitos (3.4). Outros produtos de trabalho, como uma especificação dos requisitos do sistema, cobrem todos os aspectos citados. Quando aspectos diferentes são documentados em produtos de trabalho separados ou em capítulos separados do mesmo produto de trabalho, esses produtos de trabalho ou capítulos de trabalho devem ser mantidos consistentes entre si.

### 3.1.5 Diretrizes Gerais de Documentação (L1)

Independentemente das técnicas utilizadas, as seguintes diretrizes aplicam-se na criação de produtos de trabalho:

- Selecionar um tipo de produto de trabalho que se adapte à *finalidade pretendida*.
- *Evitar a redundância* referenciando o conteúdo em vez de repetir o mesmo conteúdo.
- *Assegurar que não existem inconsistências* entre os produtos de trabalho, particularmente quando cobrem aspectos diferentes.
- *Usar termos de forma consistente*, conforme definido no glossário.
- *Estruturar* produtos de trabalho de forma apropriada.

### 3.1.6 Planejar os produtos de trabalho a serem utilizados (L1)

Cada cenário de projeto e cada domínio são diferentes, portanto, o conjunto de produtos de trabalho resultante deve ser definido para cada empreendimento. Portanto, as seguintes questões devem ser acordadas:

- Em que produtos de trabalho devem ser registrados os requisitos e para que fins?

- Quais os níveis de abstração devem ser considerados?
- Até que nível de detalhe deve ser documentado os requisitos em cada nível de abstração?
- Como devem ser representados os requisitos nestes produtos de trabalho?

Os produtos de trabalho a serem utilizados devem ser definidos numa fase inicial de um projeto. Isto tem várias vantagens, como:

- Ajudar a planejar esforços e recursos.
- Assegurar que são utilizadas as notações apropriadas.
- Assegurar que todos os resultados são registrados nos produtos de trabalho corretos.
- Assegurar que não é necessária uma grande remodelação de informação e "edição final".
- Ajuda a evitar a redundância, resultando em menos trabalho e facilidade na manutenção.

### 3.2 Produtos de Trabalho com Base em Linguagem Natural (L2)

Desde o início da ER sistemática, os requisitos em linguagem natural têm sido um meio essencial para especificar os requisitos na prática.

Os produtos de trabalho em linguagem natural têm grandes vantagens:

- A linguagem natural sem restrições é extremamente expressiva e flexível.
- Quase qualquer requisito concebível em qualquer aspecto pode ser expresso em linguagem natural.
- A linguagem natural é usada na vida cotidiana e ensinada na escola, não é necessário nenhum treinamento específico para ler e compreender textos em linguagem natural.

Em contrapartida, os textos escritos na linguagem natural, podem frequentemente ser interpretados de diferentes maneiras, o que constitui um problema ao especificar requisitos. Além disso, a detecção de ambiguidades, omissões e inconsistências em tais textos é difícil e cara.

A escrita de bons requisitos em linguagem natural pode ser suportada por:

- Escrever frases curtas e bem estruturadas.
- Definição e uso consistente de uma terminologia uniforme (3.5).
- Evitar termos e frases vagos ou ambíguos.
- Conhecendo as armadilhas da escrita técnica listada abaixo.

Ao escrever documentos técnicos em linguagem natural, há algumas armadilhas bem conhecidas que devem ser evitadas ou usadas com cuidado [GoRu2003].

Coisas a evitar:

- Descrições incompletas
- Substantivos inespecíficos
- Condições incompletas

- Comparações incompletas

Coisas a serem usadas com cuidado:

- Voz passiva
- Quantificadores universais (tais como "todos" ou "nunca")
- Nomeações (isto é, substantivos derivados de um verbo, p. ex., "autenticação")

### 3.3 Produtos de trabalho baseados em template (L3)

Os produtos de trabalho baseados em templates são usados para superar algumas das deficiências dos produtos de trabalho baseados na linguagem natural, fornecendo estruturas pré-definidas para os requisitos.

- Os *templates de frases* fornecem uma estrutura sintática predefinida para uma frase que expressa um requisito, especialmente um requisito individual ou uma história de usuário.
- Os *templates de formulário* fornecem um conjunto de campos em um formulário, por exemplo, para escrever um caso de uso ou um requisito de qualidade mensurável.
- Os *templates de documento* fornecem uma estrutura pré-definida para um documento de requisitos.

Vários templates são descritos na literatura. [ISO29148], [MWHN2009], e [Rupp2014] fornecem templates de frases para requisitos individuais. [Cohn2004] define um template de frase muito utilizado para histórias de usuários e [Cock2001] descreve template de formulários para casos de uso. [Laue2002] propôs um template para descrição de tarefas. [ISO29148] e [RoRo2012] fornecem template de documentos para especificações completas. Além disso, um cliente pode prescrever o uso de templates específicos do cliente em um projeto.

*Vantagens* de produtos de trabalho baseados em template:

- Fornecer uma estrutura clara e reutilizável
- Ajudar na captura de informações mais relevantes
- Fazer com que os requisitos e as especificações dos requisitos pareçam uniformes
- Melhorar a qualidade geral dos requisitos e especificações dos requisitos

*Desvantagens* e armadilhas dos produtos de trabalho baseados em template:

- As pessoas comumente se concentram na conclusão formal do modelo e não no seu conteúdo.
- Os aspectos que não estão incluídos no template são mais prováveis de serem omitidos.

### 3.4 Produtos de Trabalho Baseados em Modelos (L3)

Os requisitos representados em linguagem natural têm limitações [Davi1993], em particular no que diz respeito a obter uma visão geral de um conjunto de requisitos e compreender as relações entre eles. A modelagem de requisitos atende a essas limitações.

### 3.4.1 O Papel dos Modelos na Engenharia de Requisitos (L2)

Um *modelo* é uma representação abstrata de uma parte existente da realidade ou de uma parte da realidade a ser criada. A noção da realidade inclui qualquer conjunto concebível de elementos, fenômenos ou conceitos, incluindo outros modelos. Com respeito a um modelo, a parte modelada da realidade é chamada de *original*. Exemplos para modelos fora do domínio da engenharia de software são os modelos de informação de construção (BIM) [ISO19650], que modelam os elementos necessários para planejar, construir e gerenciar edifícios e outros elementos de construção.

Na ER, os modelos ajudam a compreender as relações e conexões entre os requisitos, dando uma visão geral de um conjunto de requisitos. Isto é conseguido principalmente ao focar de alguns aspectos (p. ex., comportamento) enquanto se abstrai de todos os outros aspectos. Usar uma notação gráfica para um modelo também ajuda a obter uma visão geral. No entanto, os modelos também podem ser representados de forma não gráfica, p. ex., com tabelas.

Os modelos de requisitos têm vantagens em comparação com os requisitos representados em linguagem natural:

- As relações e interconexões entre os requisitos são mais fáceis de entender com modelos gráficos do que os especificados em linguagem natural.
- O foco em um único aspecto reduz a carga cognitiva para compreender os requisitos modelados.
- As linguagens de modelagem de requisitos têm uma sintaxe restrita que reduz possíveis ambiguidades e omissões.

Os modelos também têm limitações:

- Manter modelos que se concentram em diferentes aspectos consistentes entre si é um desafio.
- Informações de diferentes modelos precisam ser compiladas para se obter um entendimento casual.
- Os modelos concentram-se principalmente em requisitos funcionais; a maioria dos requisitos de qualidade e restrições de qualidade não pode ser expressa em modelos com esforço razoável.
- A sintaxe restrita de uma linguagem de modelagem gráfica implica que nem todas as informações relevantes podem ser expressas em um modelo.

Portanto, os modelos de requisitos e requisitos em linguagem natural são frequentemente combinados.

Na ER, os modelos podem ser usados para:

- *Especificar* requisitos (principalmente funcionais) parcial ou completamente, como um meio de substituir os requisitos representados textualmente.
- *Decompor* uma realidade complexa em aspectos bem definidos e complementares, sendo cada aspecto representado por um modelo específico.

- *Parafrasear* requisitos textuais representados de forma a melhorar a sua compreensibilidade, em particular no que diz respeito às relações entre eles.
- *Validar* requisitos representados textualmente com o objetivo de descobrir omissões, ambiguidades e inconsistências.

As *linguagens de modelagem* são usadas para expressar modelos. Várias linguagens de modelagem, por exemplo, UML [OMG2017] ou BPMN [OMG2013], foram padronizadas. Quando requisitos são especificados em uma linguagem de modelagem não-padrão, é necessária uma menção que explique a sintaxe e a semântica da linguagem de modelagem utilizada.

Há muitos tipos de modelos que podem ser usados na ER. Um engenheiro de requisitos deve entender qual o tipo de modelo mais adequado para especificar os requisitos em uma determinada situação.

Em uma fase inicial, o Engenheiro de Requisitos frequentemente começa com a modelagem do contexto (3.4.2) ou dos objetivos do sistema pretendido.

### 3.4.2 Modelagem de contexto [L2]

Os modelos que focam o aspecto do contexto especificam a incorporação estrutural de um sistema em seu ambiente e a interação entre um sistema e os atores no contexto do sistema.

*Modelos de contexto* especificam um sistema e os atores no contexto do sistema que interagem com o sistema. Um modelo de contexto também esboça as interfaces entre um sistema e seu contexto (p. ex., em termos de quais informações são trocadas).

*Diagramas de contexto* são usados como linguagem de modelagem gráfica para expressar modelos de contexto. Não existe uma notação padronizada para os diagramas de contexto. Diagramas de contexto de análise estruturada [DeMa1978] ou diagramas de caixa e linha sob medida [Glin2019] podem ser usados para expressar modelos de contexto.

Na linguagem de modelagem UML [OMG2017], *os diagramas de casos de uso* fornecem um meio para modelar um sistema e seu contexto em termos de casos de uso do sistema e os atores no contexto do sistema que interagem com o sistema através desses casos de uso.

*Casos de Uso* modelam a interação dinâmica entre um ator no contexto do sistema e um sistema a partir da perspectiva do ator. Os casos de uso são na sua maioria escritos usando templates de formulários em linguagem natural (3.3) ou usando diagramas de atividade UML (3.4.4).

### 3.4.3 Modelagem de estruturas e dados [L3]

Os modelos que se concentram em aspectos de estrutura e dados especificam os requisitos de propriedades estruturais estáticas de um sistema ou de um domínio.

Os modelos de domínio estático especificam os objetos (de negócios) e suas relações em um domínio de interesse. Eles podem ser expressos com diagramas [OMG2017] de classes UML.

Os *modelos de classe* especificam principalmente as classes de um sistema, seus atributos e relações. As classes representam entidades tangíveis e intangíveis no mundo real que o sistema deve conhecer para cumprir suas tarefas. Os diagramas de classe UML são tipicamente usados como linguagem de modelagem para modelos de classe.

#### 3.4.4 Modelagem de funções e fluxo (L3)

Os modelos que se concentram nos aspectos de função e fluxo especificam os requisitos para a sequência de ações necessárias para produzir os resultados esperados de determinadas entradas ou para executar um processo (negócio), incluindo o fluxo de controle e dados entre as ações e o responsável por qual ação.

*Modelos de atividade* são usados para especificar as funções do sistema. Na linguagem de modelagem UML [OMG2017], os *diagramas de atividade* são usados para expressar modelos de atividade. Fornecem elementos para modelar as ações e fluxo de controle entre as ações. Os diagramas de atividades também podem expressar o responsável pela ação. Os elementos de modelagem avançados (não cobertos pelo CPRE Nível Fundamental) fornecem os meios para a modelagem do fluxo de dados.

*Modelos de processos* são usados para descrever processos de negócio ou técnicos. Podem ser expressos com diagramas de atividade UML ou com linguagens específicas de modelagem de processos, como BPMN [OMG2013]. No CPRE Nível Fundamental, usamos apenas diagramas de atividade UML para modelagem de processos.

#### 3.4.5 Modelagem de Estado e Comportamento (L2)

Os modelos focados no estado e no comportamento especificam os requisitos para o comportamento de um sistema ou componente de domínio em termos de reações dependentes do estado a eventos ou da dinâmica da interação dos componentes.

*Máquinas de estado* modelam eventos que desencadeiam a transição de um estado para outro e as ações que precisam ser realizadas quando ocorre uma transição de estado. Os *gráficos de estado* [Hare1988] são máquinas de estado com condições que se decompõem hierarquicamente e/ou ortogonalmente. As máquinas de estado, incluindo os gráficos de estado, podem ser expressas na linguagem de modelagem UML [OMG2017] com *diagramas de máquina de estado* (também chamados de *diagramas de estado*).

#### 3.4.6 Outros tipos de modelos em Engenharia de Requisitos (L1)

No nível fundamental CPRE, a compreensão e aplicação de modelos é restrita a tipos de modelos selecionados e importantes. Há outros tipos de modelos que são usados na Engenharia de Requisitos. No nível fundamental CPRE, é suficiente conhecê-los e para que são usados.

Os *modelos de metas* representam um conjunto de metas, submetas e as relações entre elas. Os modelos de metas também podem incluir tarefas e recursos necessários para

atingir um objetivo, atores que querem atingir um objetivo e obstáculos que impedem a realização de um objetivo.

Em SysML [OMG2019], os *diagramas de definição de blocos* podem ser adaptados para expressar diagramas de contexto usando blocos estereotipados para o sistema e atores. Os diagramas de definição de blocos também podem ser usados para modelar a estrutura de um sistema em termos das entidades conceituais do sistema e das relações entre elas.

Os *modelos de histórias de domínio* podem ser usados para modelar a função e o fluxo, especificando histórias visuais sobre como os atores interagem com dispositivos, artefatos e outros itens em um domínio, tipicamente usando símbolos específicos do domínio [HoSch2020]. Eles são um meio para entender o domínio de aplicação no qual um sistema irá operar.

*Modelos de interação* modelam as interações dinâmicas entre objetos ou atores. Os *diagramas de sequência* UML são um meio popular para especificar a interação entre objetos.

### 3.5 Glossários (L2)

Em cada iniciativa da ER que envolve mais de uma pessoa, há o risco de falta de um entendimento comum da terminologia, ou seja, que algumas pessoas interpretem diferentemente os mesmos termos. Para mitigar este problema, o entendimento comum dos termos relevantes é registrado em um glossário.

Um *glossário* é uma coleção central de definições para: termos específicos do contexto, termos cotidianos com um significado especial no contexto dado, abreviaturas e acrônimos. Os sinônimos (termos diferentes que indicam a mesma coisa) devem ser marcados como tal. Os homônimos (usando o mesmo termo para coisas diferentes) devem ser evitados ou marcados como tal.

As seguintes regras aplicam-se aos glossários:

- Gerenciar o glossário de forma centralizada.
- Manter o glossário ao longo de todo o desenvolvimento do sistema.
- Definir uma pessoa ou um pequeno grupo responsável pelo glossário.
- Usar um estilo e uma estrutura uniformes para o glossário.
- Envolver os stakeholders buscando um acordo sobre a terminologia.
- Tornar o glossário acessível a todos os envolvidos.
- Tornar obrigatório o uso do glossário.
- Verificar se os produtos de trabalho respeitam o glossário.

### 3.6 Documentos de Requisitos e Estruturas de Documentação (L2)

Os documentos de especificação de requisitos (3.1.1) compreendem vários produtos de trabalho na ER. Portanto, é importante organizar tais documentos com uma estrutura bem definida, a fim de criar um conjunto de requisitos consistente e de manutenção razoável. Além dos requisitos, um documento de requisitos também pode conter informações e

explicações adicionais – por exemplo, um glossário, condições de aceite, informações sobre o projeto ou características da implementação técnica.

Os requisitos também podem ser organizados em estruturas de documentação além dos documentos clássicos.

Os documentos mais utilizados são:

- Especificação de Requisitos dos Stakeholders
- Especificação dos Requisitos de Usuários (um subconjunto de uma especificação dos requisitos dos stakeholders, abrangendo apenas os requisitos dos usuários)
- Especificação dos Requisitos de Sistema
- Especificação de Requisitos de Negócio
- Documento de Visão

Estruturas alternativas de documentação frequentemente utilizadas:

- Backlog do produto
- Backlog da sprint
- Mapa de histórias

Tanto a seleção de uma estrutura de documentação como a organização interna da estrutura escolhida dependem de:

- O processo de desenvolvimento escolhido (5)
- O tipo de desenvolvimento e domínio
- O contrato (um cliente pode prescrever o uso de uma determinada estrutura de documentação)
- O tamanho do documento

Os templates de documentos podem ajudar a estruturar uma especificação de requisitos. Os templates estão disponíveis na literatura [Vole2020], [RoRo2012] e em normas [ISO29148]. Os templates também podem ser reutilizados de projetos anteriores, similares ou podem ser impostos por um cliente. Uma organização também pode decidir criar um template como um padrão interno.

### 3.7 Protótipos na Engenharia de Requisitos (L1)

Na ER, os *protótipos* são um meio de especificar os requisitos através do exemplo e de validar os requisitos. Em particular, os protótipos podem ser usados se os stakeholders não quiserem escrever e revisar produtos de trabalho baseados em linguagem natural ou em modelos.

Os *protótipos exploratórios* [LiSZ1994] são usados para criar um entendimento comum, esclarecer requisitos ou validar requisitos em diferentes níveis de fidelidade. Eles são descartados após o uso.

- *Wireframes* são protótipos de baixa fidelidade construídos com materiais simples ou ferramentas de esboço que servem principalmente para discutir e validar ideias de design e conceitos de interface com o usuário.

- *Maquetes* são protótipos de média fidelidade. Ao especificar sistemas digitais, eles usam telas reais e fluxos de cliques, mas sem funcionalidade real. Eles servem principalmente para especificar e validar interfaces de usuário.
- *Protótipos Nativos* são protótipos de alta fidelidade que implementam partes críticas de um sistema de tal forma que os stakeholders possam usar o protótipo para ver se a parte prototipada do sistema funcionará e se comportará como esperado.

Dependendo do grau de fidelidade, os protótipos exploratórios podem ser um produto de trabalho caro, por isso há sempre um balanço entre custo e valor ganho.

*Protótipos evolutivos* [LiSZ1994] são sistemas-piloto que formam o núcleo de um sistema a ser desenvolvido. O sistema final evolui através da ampliação e melhoria progressiva do sistema-piloto em várias iterações.

### 3.8 Critérios de Qualidade para Produtos de trabalho e requisitos (L1)

Um requisito tem de cumprir critérios de qualidade específicos para ser considerado como um bom requisito. Na ER moderna com abordagens orientadas para o valor (Princípio 1 em 2), o grau de cumprimento de um critério de qualidade deve corresponder ao valor criado por este requisito. Isto significa que os requisitos não têm de aderir completamente a todos os critérios de qualidade (mas quanto maior for o valor de um requisito, mais relevantes são os critérios de qualidade, para reduzir o risco de insucesso).

Adequação e compreensibilidade são os critérios de qualidade mais importantes para requisitos individuais. Sem eles, um requisito é inútil ou mesmo prejudicial, independentemente do cumprimento de todos os outros critérios.

Critérios de qualidade para *requisitos individuais*:

- Adequado (descreve as necessidades reais e acordadas dos stakeholders)
- Necessário
- Inequívoco
- Completo (autocontido)
- Compreensível
- Verificável

Como descrito em 3.1.1, os requisitos são geralmente documentados em vários produtos de trabalho que cobrem requisitos únicos ou múltiplos. Os critérios de qualidade acima devem ser usados para criar requisitos únicos de alta qualidade num produto de trabalho. Para produtos de trabalho que cobrem mais de um requisito, devem ser considerados adicionalmente os seguintes critérios de qualidade.

Cr terios de qualidade para produtos de trabalho que cobrem *m ltiplos requisitos*:

- Consistente
- N o redundante
- Completo (nenhum requisito conhecido e relevante   negligenciado)
- Modific vel
- Rastre vel
- Em conformidade

## 4 Práticas de Elaboração de Requisitos (L3)

**Objetivo:** Compreender o uso de práticas para identificar fontes de requisitos, para elicitare requisitos, para identificar e resolver conflitos, e para validar requisitos

**Duração:** 4 horas 30 minutos

**Termos:** Fonte de requisitos, limite do sistema, contexto do sistema, elicitação de requisitos, validação de requisitos, stakeholder, modelo Kano, resolução de conflitos

### Objetivos Educacionais

- OE 4.1.1 Determinar os limites do sistema para focar nos requisitos relevantes (L3)
- OE 4.1.2 Relembrar as fontes relevantes para o sistema a ser criado (L1)
- OE 4.1.3 Identificar stakeholders e relacioná-los em uma lista (L3)
- OE 4.1.4 Entender os benefícios do gerenciamento de stakeholders (L2)
- OE 4.2.1 Entender como o modelo de Kano pode ajudar a elicitar os requisitos certos (L2)
- OE 4.2.2 Entender a diferença entre técnicas de coleta, de design e de geração de ideias (L2)
- OE 4.2.3 Entender como escolher a técnica de elicitação correta para uma dada situação (L2)
- OE 4.3.1 Lembrar os diferentes tipos de conflito (L1)
- OE 4.3.2 Entender quais atividades são necessárias para resolver conflitos (L2)
- OE 4.3.3 Entender como aplicar técnicas apropriadas da resolução de conflitos (L2)
- OE 4.4.1 Entender por que os documentos de requisitos precisam ser validados (L2)
- OE 4.4.2 Lembrar os quatro aspectos importantes para a validação dos requisitos (L1)
- OE 4.4.3 Entender como aplicar técnicas apropriadas para validação de requisitos (L2)

### 4.1 Fontes de requisitos (L3)

A qualidade e a completude dos requisitos dependem muito das fontes de requisitos envolvidas. A falta de uma fonte relevante levará a uma compreensão incompleta dos requisitos ou a requisitos incompletos. A identificação das fontes de requisitos é um processo iterativo e recursivo que requer uma constante reconsideração.

Um entendimento comum (Princípio 3 em 2) do contexto do sistema a ser desenvolvido é um pré-requisito para identificar as fontes de requisitos relevantes. A área entre o limite do sistema e o limite do contexto é chamada de contexto (do sistema) (Princípio 4 em 2). O contexto (do sistema) é necessário para compreender a natureza dos requisitos a serem desenvolvidos e assim identificar suas fontes.

As fontes de requisitos são classificadas em três tipos:

- Stakeholders
- Documentos
- Sistemas

Os stakeholders de um sistema (ver [Glin2020] para uma definição; ver também Princípio 2 em 2) são a principal fonte de requisitos. As funções típicas dos stakeholders incluem [BiSp2003]:

- Usuários (também chamados de usuários-finais)
- Patrocinadores
- Gestores
- Desenvolvedores
- Autoridades
- Clientes

Além disso, as pessoas ou organizações que são afetadas por um sistema devem ser consideradas como stakeholders (indiretas).

A identificação sistemática dos stakeholders deve ocorrer no início do desenvolvimento e os resultados devem ser gerenciados ao longo do desenvolvimento como uma atividade contínua. Isso inclui a identificação tanto dos papéis dos stakeholders quanto das pessoas que exercem esses papéis.

Para todos os sistemas com interface de usuário, os *usuários finais* do sistema constituem um grupo de stakeholders que é de particular interesse para o engenheiro de requisitos. Os usuários finais devem ser agrupados (por exemplo, por funções, tarefas ou responsabilidades similares).

Quando os usuários finais podem ser identificados individualmente, devem ser escolhidos representantes de cada grupo. Caso contrário, podem ser definidas pessoas para representar os grupos de usuários finais relevantes [Coop2004].

As fontes potenciais para identificar os stakeholders relevantes e seus papéis são:

- Listas de verificação de grupos e papéis típicos de stakeholders
- Estruturas organizacionais
- Documentação do processo de negócio
- Relatórios de mercado
- Stakeholders iniciais para a identificação de *outros* stakeholders

Os stakeholders devem ser documentados numa lista de stakeholders atualizada com (pelo menos) as seguintes informações:

- Nome
- Função (cargo)
- Dados pessoais adicionais e de contato
- Disponibilidade temporal e espacial durante o andamento do projeto
- Relevância
- Área e tempo de experiência
- Objetivos e interesses em termos do projeto

Problemas com os stakeholders podem surgir se os direitos e obrigações de um stakeholder não forem claros ou se suas necessidades não forem suficientemente atendidas. O gerenciamento de relacionamento de stakeholders [Bour2009] é uma forma eficaz de combater problemas com eles.

Na maioria dos contextos de sistema, mais fontes estão disponíveis. Eles também devem ser considerados para um novo sistema bem-sucedido, pois a maioria dos stakeholders não fala sobre o óbvio: seus requisitos "subconscientes" (4.2).

As fontes adicionais para os requisitos incluem:

- Sistemas existentes e legados
- Documentos de processo
- Documentos legais ou regulamentares
- Regulamentos específicos da empresa
- Informação (de marketing) sobre potenciais futuros usuários

Outras fontes de requisitos podem ser encontradas analisando situações semelhantes em domínios completamente diferentes.

## 4.2 Elicitação de requisitos (L2)

Dentro da elicitação, é tarefa do Engenheiro de Requisitos compreender os desejos e necessidades dos stakeholders, assegurando que os requisitos de todas as fontes relevantes foram coletados, aplicando técnicas apropriadas para obtê-los. Um ponto importante na elicitação é transformar exigências implícitas, desejos e expectativas em requisitos explícitos.

Para elicitar requisitos, é crucial conhecer a natureza e a importância de cada requisito. Estes podem mudar de projeto para projeto e ao longo do tempo. O modelo de Kano [KaeA1984] classifica os requisitos em três categorias relevantes:

- Atrativos (sinônimos: fatores de excitação, requisitos inconscientes)
- Satisfatórios (sinônimos: fatores de desempenho, requisitos conscientes)
- Insatisfatórios (sinônimos: fatores básicos, requisitos subconscientes)

Existem muitas técnicas diferentes para elicitar estas categorias de requisitos. Nós diferenciamos entre:

- Técnicas de coleta
- Técnicas de Desenho e Geração de Ideias

*Técnicas de coleta* são técnicas estabelecidas para elicitação de requisitos [BaCC2015] que ajudam elicitações satisfatórias e insatisfatórias pela investigação de diferentes fontes.

Quatro categorias principais podem ser discernidas:

- Técnicas de Questionamentos
- Técnicas de Colaboração
- Técnicas de Observação
- Técnicas baseadas em Artefatos

As *técnicas de geração de ideias e design* destinam-se a estimular a criatividade durante a elicitação de requisitos. O seu objetivo é criar ideias para resolver um problema e explorar ideias de design [Kuma2013]. Isto pode levar a requisitos inovadores que muitas vezes são

encantadores. Exemplos populares de tais técnicas são o brainstorming [Osbo1979], analogias, prototipagem (p. ex., maquetes), cenários, e storyboards.

Um conceito mais amplo relacionado ao design e à geração de ideias é o *design thinking*. Existem diferentes abordagens, tais como *d.school* [Sdsc2012] e *Designing for Growth* [LiOg2011], oferecendo um amplo conjunto de técnicas que podem ser utilizadas para eliciar requisitos.

As técnicas de elicitação devem ser capazes de detectar todos os tipos de requisitos (requisitos funcionais e de qualidade), bem como restrições. Na prática, os requisitos de qualidade e restrições recebem frequentemente menos atenção.

Para eliciar os *requisitos de qualidade*, um modelo de qualidade como a norma ISO 25010 [ISO25010] deve ser usada como uma lista de verificação. Este modelo também pode ser útil para quantificar os requisitos.

As *restrições* podem ser encontradas considerando possíveis limitações do espaço potencial de solução, por exemplo, questões técnicas, legais, organizacionais, culturais ou ambientais.

Escolher as técnicas certas de elicitação é uma competência fundamental que depende de muitos fatores diferentes, como por exemplo:

- Tipo de sistema
- Modelo de ciclo de vida de desenvolvimento de software
- Pessoas envolvidas
- Configuração organizacional

Os melhores resultados são geralmente obtidos com uma combinação de diferentes técnicas de elicitação. [CaDJ2014] representa uma abordagem sistemática para a seleção das técnicas.

### 4.3 Resolução de Conflitos relativos aos Requisitos (L2)

As técnicas de elicitação por si só não garantem que o conjunto de requisitos resultante seja consistente, completo, conforme etc. (3.8). Para o conjunto final, entretanto, todos os stakeholders devem compreender e aceitar todos os requisitos relevantes para eles. Se alguns stakeholders não concordarem, a situação deve ser reconhecida como um conflito que deve ser resolvido de acordo. As técnicas adequadas de resolução de conflitos devem ser selecionadas com base no tipo de conflito e informações contextuais. Isto requer uma compreensão profunda da natureza do conflito de requisitos e das atitudes dos stakeholders envolvidos.

As tarefas para identificar e resolver conflitos são:

- Identificação de conflitos
- Análise de conflitos
- Resolução de conflitos
- Documentação da resolução de conflitos (decisões tomadas)

É útil distinguir entre diferentes tipos [Moor2014] de conflitos. Os seguintes tipos de conflito frequentemente requerem a atenção do Engenheiro de Requisitos:

- Conflito de temas
- Conflito de dados
- Conflito de interesses
- Conflito de valor
- Conflito de relacionamento
- Conflito estrutural

Para resolver conflitos com sucesso, técnicas comuns podem ser aplicadas:

- Acordo (Agreement)
- Compromisso (Compromise)
- Votação (Voting)
- Imposição (Overruling)
- Análise de alternativas

Além disso, existem várias técnicas auxiliares, por exemplo:

- "Obter mais informações"
- Pontos fortes e pontos fracos
- Matriz de Decisão

## 4.4 Validação dos Requisitos (L2)

A validação dos requisitos é um passo importante para um sistema bem-sucedido (Princípio 6 em 2). Garantir a qualidade dos requisitos no início reduzirá o desperdício de esforços posteriores. Validar os requisitos significa verificar os produtos de trabalho mais disponíveis, bem como os requisitos individuais que os mesmos exigem em termos de qualidade (ver 3.8 para detalhes).

Aspectos importantes a serem considerados na validação dos requisitos são:

- Envolvimento dos stakeholders certos
- Separar a identificação e a correção de defeitos
- Validar de diferentes perspectivas
- Validações Adicionais

Existem várias técnicas para validação (p. ex., [GiGr1993], [OleA2018]). Estas técnicas de validação são frequentemente classificadas em:

- *Técnicas de revisão*, incluindo:
  - Walkthrough
  - Inspeção
- *Técnicas exploratórias*, por exemplo:
  - Protótipos
  - Testes alfa e beta
  - Teste A/B [KoTh2017]

- Construindo um Produto Mínimo Viável (MVP)
- *Desenvolvimento de amostras*

Estas técnicas diferem em formalidade e esforço. A seleção da técnica depende de fatores como modelo de ciclo de vida do desenvolvimento de software, maturidade do processo de desenvolvimento, complexidade e nível de risco do sistema, requisitos legais ou normativos, e/ou a necessidade de uma trilha de auditoria.

## 5 Processo e Estrutura de Trabalho (L3)

Objetivo: Explicar os conceitos de processos da ER e aplicar configurações de processos apropriadas

Duração: 1 hora 15 minutos

Termos: Processo, processo da ER

### Objetivos Educacionais

OE 5.1.1 Conhecer os fatores importantes que influenciam um processo da ER (L1)

OE 5.1.2 Entender como e por que estes fatores influenciam (L2)

OE 5.2.1 Entender as vertentes a serem consideradas para a configuração de um processo da ER (L2)

OE 5.3.1 Conhecer as configurações típicas do processo da ER (L1)

OE 5.3.2 Entender os passos para a configuração de um processo da ER (L2)

OE 5.3.3 Selecionar e aplicar uma configuração de processo própria da ER para sistema e desenvolvimento simples (L3)

Um processo é necessário para moldar e estruturar o trabalho da ER a ser feito em um determinado contexto. Como não há um processo de ER que sirva para todos (1.4), um processo sob medida deve ser configurado de acordo com o contexto do sistema e seu desenvolvimento.

O processo da ER molda o fluxo de informação e o modelo de comunicação entre os vários participantes (p. ex., clientes, usuários, engenheiros de requisitos, desenvolvedores, testadores) e define os produtos de trabalho a serem utilizados ou produzidos. Assim, o processo da ER fornece a estrutura para elicitar, documentar, validar e gerenciar os requisitos.

### 5.1 Fatores de Influência (L2)

Muitos fatores influenciam a configuração de um processo da ER. Os principais fatores são:

- Adequação global do processo: o processo da ER deve ser adequado ao processo global de desenvolvimento do sistema.
- Contexto de desenvolvimento
- Capacidade e disponibilidade dos stakeholders
- Entendimento comum
- Complexidade e criticidade do sistema a ser desenvolvido
- Restrições
- Tempo e orçamento disponíveis
- Volatilidade dos requisitos
- Experiência dos Engenheiros de Requisitos

Uma análise dos fatores influenciadores fornece informações sobre como configurar o processo da ER. Os fatores de influência também limitam o espaço de possíveis configurações de processos. Por exemplo, quando os stakeholders só estão disponíveis no

início do projeto, nenhum processo pode ser escolhido com base no feedback contínuo dos stakeholders.

## 5.2 Vertentes de Processos de Engenharia de Requisitos (L2)

Há três vertentes decisivas que precisam ser consideradas na configuração de um processo da ER [Glin2019].

### Dimensão de Tempo: Sequencial vs. Iterativo

Em um processo linear, os requisitos são especificados antecipadamente em uma única fase do processo. Em um processo iterativo, os requisitos são especificados gradualmente, começando com objetivos gerais e alguns requisitos iniciais, adicionando ou modificando os requisitos a cada iteração.

Critérios para a escolha de um processo de ER *linear*:

- O processo de desenvolvimento do sistema é orientado a planejamento antecipado e, na sua maioria, sequencial.
- Os stakeholders conhecem seus requisitos e podem especificá-los antecipadamente.
- Uma especificação abrangente dos requisitos é necessária como base contratual para terceirizar o projeto e a implementação do sistema.
- Autoridades regulatórias exigem uma especificação abrangente e formalmente liberada dos requisitos em uma fase inicial do desenvolvimento.

Critérios para a escolha de um processo da ER *iterativo*:

- O processo de desenvolvimento do sistema é iterativo e ágil.
- Muitos requisitos não são conhecidos previamente, mas vão surgir e evoluir durante o desenvolvimento do sistema.
- Os stakeholders estão disponíveis de tal forma que podem ser estabelecidos pequenos ciclos de feedback como forma de mitigar o risco de desenvolver o sistema errado.
- A duração do desenvolvimento permite mais do que apenas uma ou duas iterações.
- A capacidade de mudar facilmente os requisitos é importante.

### Dimensões de Finalidade: Prescritivo vs. Exploratório

Num processo da ER prescritivo, a especificação dos requisitos constitui um contrato: todos os requisitos são obrigatórios e devem ser implementados. Em um processo da ER exploratório, apenas os objetivos são conhecidos a priori, enquanto os requisitos concretos têm que ser explorados.

Critérios para a escolha de um processo da ER *prescritivo*:

- O cliente necessita de um contrato fixo para o desenvolvimento do sistema.
- A funcionalidade e o escopo têm precedência sobre o custo e os prazos.
- O desenvolvimento do sistema especificado pode ser licitado ou terceirizado.

Critérios para a escolha de um processo da ER *exploratório*:

- Os stakeholders inicialmente só têm uma vaga ideia sobre seus requisitos.
- Os stakeholders estão fortemente envolvidos e fornecem um feedback contínuo.
- Prazos e custos precedem a funcionalidade e o escopo.
- Não está claro a priori quais requisitos serão realmente implementados e em que ordem serão implementados.

### Dimensão Alvo: Específico para o Cliente vs. Orientado para o Mercado

Em um processo da ER específico do cliente, o sistema é encomendado por um cliente e desenvolvido por um fornecedor. Em um processo de ER orientado ao mercado, o sistema é desenvolvido como um produto ou serviço para um mercado, visando segmentos específicos de usuários.

Critérios para a escolha de um processo de ER *específico para o cliente*:

- O sistema será utilizado principalmente pela organização que encomendou o sistema e paga pelo seu desenvolvimento.
- Os stakeholders mais importantes estão principalmente associados à organização do cliente.
- Indivíduos podem ser identificados para os papéis de stakeholders.
- O cliente quer uma especificação de requisitos que possa servir como um contrato.

Critérios para a escolha de um processo da ER *orientado para o mercado*:

- A organização que desenvolve o sistema pretende vender em algum segmento de mercado como um produto ou serviço.
- Os usuários potenciais não são identificáveis individualmente.
- Os Engenheiros de Requisitos têm de conceber os requisitos de modo a corresponderem às necessidades previstas dos usuários visados.
- Product owner, pessoal de marketing, designers digitais e arquitetos de sistemas são os principais stakeholders.

### Dicas e Ressalvas

- Os critérios apresentados acima são *heurísticos* e não regras fixas. Por exemplo, a terceirização do desenvolvimento do sistema é feita preferencialmente com um processo da ER prescritivo e não com um processo exploratório, pois o contrato entre o cliente e o fornecedor é normalmente baseado em uma especificação de requisitos abrangente. Entretanto, também é possível negociar um contrato de terceirização com base em um processo de ER exploratório.
- Os processos sequenciais de ER só funcionam se houver um processo sofisticado de mudança de requisitos.
- Os processos lineares da ER implicam longos ciclos de feedback: Para mitigar o risco de desenvolver o sistema errado, os requisitos devem ser validados intensivamente.
- Ao definir um processo da ER, o *linear* e o *prescritivo* são frequentemente escolhidos em conjunto.

- Os processos de ER exploratórios são tipicamente também processos iterativos (e vice-versa).
- Num processo orientado para o mercado, o feedback dos usuários é o único meio de validar se o produto irá realmente satisfazer as necessidades do segmento de usuários visados.
- A vertente orientada para o mercado não se combina bem com as vertentes lineares e prescritivas.

## 5.3 Configuração de um Processo de Engenharia de Requisitos (L3)

Num contexto concreto de desenvolvimento do sistema, os responsáveis pela ER têm de configurar o processo da ER a ser aplicado. Com base em uma análise dos fatores de influência em 5.1, uma combinação adequada das vertentes do processo descritas em 5.2 pode ser usada [Glin2019]. A seguir, três combinações típicas são descritas.

### Processo da ER participativo: Iterativo & Exploratório & Específico do Cliente

Principal aplicação::	Fornecedor e cliente colaboram estreitamente; os stakeholders estão fortemente envolvidos tanto na ER como nos processos de desenvolvimento.
Produtos de trabalho típicos:	Backlog do produto com histórias de usuários e/ou descrições de tarefas, protótipos
Fluxo de informação típico:	Interação contínua entre stakeholders, product owners, engenheiros de requisitos e desenvolvedores; pode incluir feedback dos usuários

### Processo da ER contratual: Tipicamente Linear (às vezes Iterativo) & Prescritivo & Específico para o Cliente

Principal aplicação::	A especificação de requisitos constitui a base contratual para o desenvolvimento de um sistema por pessoas não envolvidas na especificação e com pouca interação dos stakeholders após a fase de requisitos.
Produtos de trabalho típicos:	Especificação dos requisitos do sistema clássico, que consiste em requisitos e modelos baseados em linguagem natural
Fluxo de informação típico:	Principalmente dos stakeholders para os Engenheiros de Requisitos

## Processo de ER Orientado para o Produto: Iterativo, Exploratório e Orientado para o Mercado

Principal aplicação::	Uma organização específica e desenvolve software para vendê-lo ou distribuí-lo como um produto ou serviço
Produtos de trabalho típicos:	Backlog do Produto, protótipos
Fluxo de informação típico:	Interação entre o Product Owner, marketing, engenheiros de requisitos, designers digitais, desenvolvedores e (talvez) feedback rápido de clientes/usuários

Note que pode haver contextos de sistema e desenvolvimento onde nenhuma das configurações acima mencionadas se encaixe. Por exemplo, restrições regulatórias podem impor a utilização de um processo que esteja em conformidade com determinadas normas, como a ISO/IEC/IEEE 29148 [ISO29148].

Ao configurar um processo da ER, recomendamos a utilização de um procedimento em cinco etapas:

1. Analisar os fatores de influência (5.1)
2. Avaliar os critérios de vertentes (5.2)
3. Configurar o processo (5.3)
4. Determinar produtos de trabalho (3)
5. Selecione as práticas apropriadas

## 6 Práticas de Gestão de Requisitos (L2)

Objetivo: Compreender a necessidade e o benefício do Gerenciamento de Requisitos

Duração: 2 horas

Termos: Gestão de Requisitos, gerenciamento de mudanças, rastreabilidade, atributos de requisitos, ciclo de vida de requisitos, priorização

### Objetivos Educacionais

- OE 6.1.1 Conhecer o que é a Gerenciamento de Requisitos e porque é necessária (L1)
- OE 6.2.1 Explicar por que é que os produtos de trabalho de requisitos precisam de um modelo de status/ciclo de vida (L2)
- OE 6.3.1 Explicar o conceito de versão de requisitos em uma determinada situação de projeto (L2)
- OE 6.4.1 Conhecer o uso de configurações de requisitos e baselines (L1)
- OE 6.5.1 Conhecer o propósito dos atributos para os requisitos (L1)
- OE 6.5.2 Explicar um conjunto apropriado de atributos para requisitos em uma determinada situação de projeto (L2)
- OE 6.5.3 Explicar o propósito das visualizações e nomear as diferentes visões dos requisitos (L2)
- OE 6.6.1 Nomear as razões para a rastreabilidade dos requisitos (L1)
- OE 6.6.2 Resumir as diferenças entre rastreabilidade implícita e explícita (L1)
- OE 6.6.3 Saber como a rastreabilidade explícita pode ser documentada (L1)
- OE 6.7.1 Saber lidar com mudanças em abordagens lineares (orientadas por planos) e ágeis (L1)
- OE 6.8.1 Conhecer o motivo da priorização e os critérios de avaliação significativos (L1)
- OE 6.8.2 Nomear as etapas para priorizar os requisitos (L1)
- OE 6.8.3 Nomear as diferentes categorias de técnicas de priorização (L1)

### 6.1 O que é o Gerenciamento de Requisitos? (L1)

A Gerenciamento de Requisitos é o processo de gerenciamento dos requisitos existentes documentados em vários produtos de trabalho. Em particular, isto inclui o armazenamento, mudança e rastreamento de requisitos [Glin2020]. O gerenciamento de requisitos pode acontecer de diferentes maneiras e níveis, dependendo do processo de desenvolvimento e do contexto escolhido, veja por exemplo, [Leff2011], [Rupp2014], [WiBe2013].

Independentemente das circunstâncias, a tarefa do gerenciamento de requisitos é manter os requisitos de tal forma, que todas os papéis em um projeto possam funcionar de forma eficaz e eficiente.

### 6.2 Gestão do Ciclo de Vida (L2)

A gestão do ciclo de vida refere-se ao processo de acompanhamento de todos os produtos de trabalho no que diz respeito ao status no seu ciclo de vida. Cada requisito documentado e cada produto de trabalho tem o seu próprio ciclo de vida: é criado, avaliado e refinado antes de ser revisado, retrabalhado, consolidado, acordado, e assim por diante. Para permitir a identificação de qual produto de trabalho está em qual estado, é necessário um modelo de ciclo de vida definindo cada status de ciclo de vida permitido e a transição de estado. O

status atual de um produto de trabalho deve ser sempre claro, incluindo (geralmente) o histórico de suas transições.

### 6.3 Controle de Versão [L2]

O controle da versão dos requisitos refere-se ao processo de acompanhamento de todos os produtos de trabalho durante a sua evolução. Qualquer mudança em um produto de trabalho deve ser refletida em uma nova versão. O versionamento permite que o histórico de um produto de trabalho seja rastreado até a sua origem e a recuperação de qualquer versão anterior de um produto de trabalho. Para este fim, o controle de versão requer a implementação de três medidas:

- Um número de versão para identificar de forma única a versão de um produto de trabalho.
- Um histórico do que foi mudado.
- Um conceito de armazenamento de produtos do trabalho.

O processo de criação de versões deve ser considerado para todos os produtos de trabalho [WiBe2013]. Um número de versão normalmente consiste em pelo menos duas partes: a versão e o incremento.

### 6.4 Configurações e baselines [L1]

Uma *configuração* de requisitos é um conjunto consistente de produtos de trabalho que contém requisitos. Cada configuração é definida para um propósito específico e contém, no máximo, uma versão de cada produto de trabalho [Glin2020]. A finalidade das configurações é, por exemplo, rever um conjunto de produtos de trabalho ou facilitar uma estimativa do esforço de desenvolvimento.

Um *baseline* é uma *configuração* estável e controlada de produtos de trabalho, usada para planejamento de liberação ou outros marcos de entrega em um projeto [Glin2020].

As configurações têm as seguintes propriedades:

1. Ligação lógica
2. Consistência
3. Singularidade
4. Inalterabilidade
5. Base para redefinição

### 6.5 Atributos e Visualizações[L2]

Os *atributos* são necessários para documentar metadados importantes para um produto de trabalho e são normalmente usados para responder a uma série de perguntas importantes durante o projeto ou ciclo de vida do produto.

O objetivo de utilizar atributos para caracterizar os requisitos é permitir que os membros da equipe e stakeholders obtenham as informações sobre os requisitos de que necessitam em qualquer momento durante o projeto.

A definição do conjunto relevante de atributos depende das necessidades de informação dos diferentes stakeholders no projeto. Os padrões existentes, por exemplo [ISO29148], fornecem uma visão geral dos atributos mais relevantes.

As *visões* são uma parte do conjunto total de requisitos que contém apenas o conteúdo que é atualmente de interesse. De uma perspectiva técnica, uma visualização é uma combinação de configurações de filtro e ordenação que pode ser disponibilizada ou reutilizada por outros participantes através do armazenamento da combinação selecionada.

Diferenciamos entre três tipos de visualizações:

- *Visualizações seletivas*
- *Visualizações projetadas*
- *Visualizações agregadas*

Na maioria dos casos, as visualizações dos requisitos são combinações de visualizações seletivas, projetadas e agregadoras para a criação de relatórios.

## 6.6 Rastreabilidade (L1)

Rastreabilidade [GoFi1994] é a capacidade de rastrear um requisito *até a sua origem* (ou seja, stakeholders, documentos, justificativas etc.) e *encaminhar para produtos de trabalho subsequentes* (p. ex., casos de teste), bem como *para outros requisitos* dos quais este requisito depende.

A rastreabilidade é um pré-requisito para a Gerenciamento de Requisitos e é muitas vezes claramente exigida por normas, leis e regulamentos. Implementar a rastreabilidade significa basicamente manter as dependências entre diferentes produtos de trabalho (3.1) em diferentes níveis de abstração (3.1.2), níveis de detalhe (3.1.3) e para todos os antecessores e sucessores relevantes por razões de análise, conformidade e informação.

A rastreabilidade pode ser documentada *implicitamente*, estruturando e padronizando produtos de trabalho, ou *explicitamente*, relacionando produtos de trabalho com base nos seus identificadores únicos de várias formas [HuJD2011]. As formas comuns de representação são hiperlinks, referências, matrizes, tabelas ou gráficos.

## 6.7 Lidando com mudanças (L1)

Os requisitos não são estáticos. Mudanças nos requisitos acontecem devido a diferentes razões e precisam ser tratadas adequadamente (Princípio 7 em 2), por exemplo, criando um *pedido de mudança* formal, ou adicionando um novo item ao *backlog do produto*.

A tomada de decisões, o planejamento e o controle da implementação de uma mudança dependem da abordagem de desenvolvimento e do momento em que a mudança ocorre.

Numa abordagem *linear*, a decisão sobre uma mudança é frequentemente tomada por um Conselho de Controle de Mudança (em projetos) ou por um Conselho Consultivo de Mudança (em funcionamento). Em uma abordagem mais *iterativa*, o Product Owner inclui a mudança no backlog do produto e prioriza o novo item correspondente.

## 6.8 Priorização (L1)

Nem todos os requisitos são igualmente importantes [Davi2005]. A avaliação e priorização são usadas para determinar os requisitos mais relevantes para a próxima liberação ou incremento do produto.

A *avaliação* dos requisitos é a base para a sua priorização, muitas vezes determinada pelo uso de múltiplos critérios de avaliação como valor do negócio, urgência, esforço, dependências e outros.

A *prioridade* de um requisito descreve a importância de um único requisito em comparação com outros requisitos, de acordo com determinados critérios [Glin2020]. A *priorização* em si é feita com base num único critério ou em múltiplos critérios; isto depende principalmente da técnica de priorização escolhida.

Passos para a priorização:

- Definir as principais metas e restrições para a priorização
- Definir os critérios de avaliação desejados
- Definir os stakeholders que têm de estar envolvidos
- Definir os requisitos a serem priorizados
- Selecionar a técnica de priorização
- Realizar priorização

As *técnicas de priorização* podem ser classificadas em:

- Técnicas de priorização *Ad-hoc*
- Técnicas de priorização *Analíticas*

## 7 Ferramentas de Suporte (L2)

Objetivo: Fornecer uma visão geral do papel das ferramentas da ER e aspectos para a implementação

Duração: 30 minutos

Termos: ferramenta, ferramenta da ER

### Objetivos Educacionais

OE 7.1.1 Conhecer os diferentes tipos de ferramentas da ER (L1)

OE 7.2.1 Explicar o que considerar ao introduzir ferramentas da ER (L2)

### 7.1 Ferramentas na Engenharia de Requisitos (L1)

O processo da ER pode ser apoiado por ferramentas que suportam tarefas e atividades dedicadas. Como os processos da ER são bastante singulares (5), as ferramentas da ER existentes muitas vezes se concentram apenas em certos aspectos da ER e raramente suportam todas as atividades. Antes de selecionar uma ferramenta, os Engenheiros de Requisitos devem decidir quais tarefas e atividades durante o processo da ER devem ser suportadas e como. Nós diferenciamos entre as ferramentas que suportam:

- Gerenciamento de requisitos:
  - Definição e armazenamento de atributos de requisitos
  - Priorização de requisitos
  - Gestão de versões e configurações
  - Monitorar e rastrear requisitos
  - Gestão de mudanças de requisitos
- Gestão do processo da ER:
  - Medição e relatórios sobre o processo da ER
  - Medição e relatórios sobre a qualidade do produto
  - Gerenciando o fluxo de trabalho da ER
- Documentação de conhecimento sobre os requisitos:
  - Compartilhamento de requisitos
  - Criar um entendimento comum dos requisitos
- Modelagem de requisitos
- Colaboração na ER
- Teste/simulação de requisitos

As ferramentas muitas vezes possuem uma mescla das características acima mencionadas. Para garantir que todas as tarefas da ER sejam cobertas adequadamente, diferentes ferramentas podem ser combinadas.

Algumas vezes, outros tipos de ferramentas, por exemplo ferramentas de escritório ou de rastreamento de problemas, são usados para documentar ou gerenciar requisitos. Essas

ferramentas têm limitações e só devem ser usadas quando o processo da ER estiver sob controle e os requisitos estiverem alinhados e bastante estáveis.

## 7.2 Introduzindo Ferramentas (L2)

Selecionar uma ferramenta da ER não é diferente de selecionar uma ferramenta para qualquer outro propósito. O objetivo, o contexto e os requisitos devem ser descritos antes que uma escolha possa ser bem-sucedida [Fugg1993].

Uma ferramenta apropriada só pode ser procurada uma vez que os procedimentos e técnicas da ER adequadas tenham sido introduzidos. A introdução de ferramentas requer responsabilidades e procedimentos claros da ER. Os seguintes aspectos são relevantes no processo de introdução de uma ferramenta da ER:

- Levar em conta todos os custos do ciclo de vida além dos custos de licença
- Considerar os recursos necessários
- Utilizar projetos-piloto para evitar riscos
- Avaliar a ferramenta de acordo com critérios definidos
- Instruir os funcionários sobre como usar a ferramenta

# Referências

- [AnPC1994] Annie I. Antón, W. Michael McCracken, Colin Potts: Goal Decomposition and Scenario Analysis in Business Process Reengineering. CAiSE (Conference on Advanced Information Systems Engineering), 1994, 94–104.
- [BaCC2015] K. Baxter, C. Courage, K. Caine: Understanding Your Users: A Practical Guide to User Research Methods, 2nd edition. Morgan Kaufmann, Burlington, 2015.
- [BiSp2003] K. Bittner, I. Spence: Use Case Modelling. Pearson Education, Boston, 2003.
- [Bour2009] L. Bourne: Stakeholder Relationship Management: A Maturity Model for Organisational Implementation. Gower Publishing Ltd, Burlington, 2009.
- [CaDJ2014] D. Carrizo, O. Dieste, N. Juristo: Systematizing requirements elicitation technique selection. Information and Software Technology 2014, 56(6): 644–669.
- [Cock2001] A. Cockburn: Writing Effective Use Cases. Addison–Wesley, Boston 2001.
- [Cohn2004] M. Cohn: User Stories Applied – For Agile Software Development. Addison–Wesley, Boston, 2004.
- [Coop2004] A. Cooper: The Inmates Are Running the Asylum: Why High–Tech Products Drive Us Crazy and How to Restore the Sanity. Que, Indianapolis, 2004.
- [Davi2005] A. M. Davis: Just Enough Requirements Management – Where Software Development Meets Marketing. Dorset House Publishing, New York, 2005.
- [Davi1993] A. M. Davis: Software Requirements – Objects, Functions, & States, 2nd edition, Prentice Hall, New Jersey, 1993.
- [DeMa1978] T. DeMarco: Structured Analysis and System Specification. Yourdon Press, New York, 1978.
- [Fugg1993] A. Fuggetta: A classification of CASE technology. IEEE Computer 1993, 26 (12): 25–38.
- [GiGr1993] T. Gilb, D. Graham: Software Inspection. Addison Wesley, Boston, 1993.
- [Glin2019] M. Glinz: Requirements Engineering I. Course Notes, University of Zurich, 2019. <https://www.ifi.uzh.ch/en/rerg/courses/archives/hs19/re-i.html#resources>. Última visita Julho de 2020.
- [Glin2020] M. Glinz: A Glossary of Requirements Engineering Terminology. Versão 2.0. <https://www.ireb.org/en/downloads/#cpre-glossary>. Última visita Julho de 2020.
- [GoFi1994] O. Gotel, A. Finkelstein: An Analysis of the Requirements Traceability Problem. 1st International Conference on Requirements Engineering, Colorado Springs, 1994. 94–101.

- [GoRu2003] R. Goetz, C. Rupp: Psychotherapy for System Requirements. 2nd IEEE International Conference on Cognitive Informatics (ICCI'03), London, 2003. 75–80.
- [GRL2020] Goal oriented Requirement Language. University of Toronto, Canada <https://www.cs.toronto.edu/km/GRL>. Última visita Mayy de 2020..
- [Hare1988] D. Harel: On Visual Formalisms. Communications of the ACM 1988, 31 (5): 514–530.
- [HoSch2020] S. Hofer, H. Schwentner: Domain Storytelling — A Collaborative Modeling Method. Available from Leanpub, <http://leanpub.com/domainstorytelling>. Última visita Julho de 2020.
- [HuJD2011] E. Hull, K. Jackson, and J. Dick: Requirements Engineering. Springer, 3rd Ed, 2011.
- [ISO29148] ISO/IEC/IEEE 29148: Systems and Software Engineering – Life Cycle Processes – Requirements Engineering, International Organization for Standardization, 2018.
- [ISO19650] ISO 19650: Organization and Digitization of Information about Buildings and Civil Engineering Works, including Building Information Modelling (BIM)– Information Management Using Building Information Modelling – Part 1 and 2, International Organization for Standardization, 2018.
- [ISO25010] ISO/IEC/IEEE25010:2011: Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models. International Organization for Standardization, Geneva, 2011.
- [Jack1995] M. A. Jackson: Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices. Addison–Wesley, New York, 1995.
- [Jack1995b] M. Jackson: The World and the Machine. 17th International Conference on Software Engineering 1995 (ICSE 1995). 287–292.
- [KaeA1984] N. Kano et al.: Attractive quality and must–be quality. Journal of the Japanese Society for Quality Control 1984, 14(2): 39–48. (in Japanese)
- [KoTh2017] R. Kohavi, S. Thomke: The Surprising Power of Online Experiments – Getting the most out of A/B and other controlled tests. Harvard Business Review, Sept–Oct 2017: 74–82.
- [Kuma2013] V. Kumar: 101 Design Methods – A Structured Approach for Driving Innovation in Your Organization. John Wiley & Sons, Hoboken, 2013.
- [Laue2002] S. Lauesen: Software Requirements. Styles and Techniques. Addison–Wesley, Harlow, 2002.
- [Leff2011] D. Leffingwell: Agile Software Requirements, Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison–Wesley, Boston, 2011.
- [LiOg2011] J. Liedtka, T. Ogilvie: Designing for Growth: A Design Thinking Tool Kit For Managers. Columbia University Press, 2011.

- [LISZ1994] H. Lichter, M. Schneider–Hufschmidt, H. Zullighoven: Prototyping in Industrial Software Projects – Bridging the Gap Between Theory and Practice. IEEE Transactions on Software Engineering 1994, 20 (11): 825–832.
- [MFeA2019] D. Méndez Fernández, X. Franch, N. Seyff, M. Felderer, M. Glinz, M. Kalinowski, A. Volgelsang, S. Wagner, S. Bühne, K. Lauenroth: Do We Preach What We Practice? Investigating the Practical Relevance of Requirements Engineering Syllabi – The IREB Case. ClbSE 2019: 476–487.
- [Moor2014] C. W. Moore: The Mediation Process – Practical Strategies for Resolving Conflicts, 4th edition. John Wiley & Sons, Hoboken, 2014.
- [MWHN2009] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak: Easy Approach to Requirements Syntax (EARS). 17th IEEE International Requirements Engineering Conference (RE'09), Atlanta, Georgia, 2009. 317–322.
- [OleA2018] K. Olsen et al.: Certified Tester, Foundation Level Syllabus – Version 2018. International Software Testing Qualifications Board, 2018.
- [OMG2013] Object Management Group: Business Process Model and Notation (BPMN), version 2.0.2. OMG document formal/2013–12–09 <http://www.omg.org/spec/BPMN>. Última visita Julho de 2020.
- [OMG2017] Object Management Group: OMG Unified Modeling Language (OMG UML), version 2.5.1. OMG document formal/2017–12–05. <https://www.omg.org/spec/UML/About-UML/>. Última visita Julho de 2020.
- [OMG2019] Object Management Group: OMG Systems Modeling Language (OMG SysML™), Version 1.6. OMG Document formal/19–11–01. <https://www.omg.org/spec/SysML/>. Última visita em Janeiro de 2022.
- [Osbo1979] A. F. Osborn: Applied Imagination, 3rd revised edition. Charles Scribner’s Sons, New York, 1979.
- [RoRo2012] S. Robertson and J. Robertson: Mastering the Requirements Process, 3rd edition. Addison–Wesley, Boston, 2012.
- [Rupp2014] C. Rupp: Requirements–Engineering und Management, 6. Auflage. Hanser, München, 2014. (in German).
- [Sdsc2012] Stanford d.school: An Introduction to Design Thinking. Hasso Plattner Institute of Design, Stanford, 2012. <https://dschool-old.stanford.edu/groups/designresources/wiki/36873>. Última visita Julho de 2020.
- [vLam2009] Axel van Lamsweerde: Requirements Engineering: From System Goals to UML Models to Software Specifications. Chichester: John Wiley & Sons, 2009.
- [Vole2020] Volere: Requirements Resources. <https://www.volere.org>. Última visita Julho de 2020.
- [WiBe2013] K. Wiegers and J. Beatty: Software Requirements, 3rd edition. Microsoft Press, Redmond, 2013.