



Certified Professional for Requirements Engineering

Syllabus

Gerenciamento de Requisitos

Practitioner | Specialist



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 poderá utilizar este Syllabus como base para artigos, livros ou outras publicações derivadas, desde que tais publicações reconheçam e citem os autores do presente documento e o IREB e.V. como fonte e detentor dos direitos autorais do mesmo.

© IREB e.V.

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

Agradecimentos

Este syllabus foi criado por (em ordem alfabética) Stan Bühne, Frank Engel, Sven Eselgrimm, Günter Halmans, Andrea Herrmann, Frank Houdek, Patrick Mäder, Alexander Rachmann, Thomas Schölzl, Amin Soesanto, Frank Stöckel e Malik Tayeh. Traduzido do Inglês por George Fialkovitz, Guilherme Simões, Ana Moreira, Luciana Ribeiro, Carlos André e Silva, Martin Tornquist.

Agradecemos a todos por sua contribuição voluntária.

Copyright © 2015 – 2024 deste syllabus está com os autores listados acima. Os direitos foram transferidos para o IREB International Requirements Engineering Board e.V.

Objetivo do documento

Este syllabus define os objetivos educacionais e um resumo do conteúdo educacional para as certificações Requirements Management Practitioner e Specialist, estabelecidas 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 módulo Gerenciamento de Requisitos é voltado para profissionais das áreas de Engenharia de Requisitos, análise de negócios, engenharia de negócios, design organizacional etc. que desejam aprofundar seus conhecimentos e habilidades na área de Gerenciamento de Requisitos.

Escopo do conteúdo

No nível Practitioner/Specialist – como no Foundation Level – são fornecidos princípios de Engenharia de Requisitos que são igualmente válidos para qualquer sistema – como sistemas incorporados, sistemas críticos de segurança e sistemas de informação tradicionais. Isso não significa que a adequação das abordagens para as áreas individuais, levando em conta suas particularidades, não possa ser tratada em um curso de treinamento. No entanto, o objetivo não é apresentar métodos específicos de Engenharia de Requisitos de um determinado domínio.

O conteúdo abordado neste syllabus pode ser usado da mesma forma em projetos (de desenvolvimento) e no gerenciamento de produtos, na evolução de sistemas existentes ou no gerenciamento contínuo de requisitos entre projetos. Por motivos de simplicidade, muitas vezes apenas os projetos são mencionados em vez dessa extensa lista.

Nenhum conjunto específico de procedimentos ou modelo de processo associado é assumido, o que prescreveria o planejamento, a direção e a sequência de aplicação dos conceitos aprendidos na prática.

O objetivo não é promover um processo específico para a Engenharia de Requisitos ou para a engenharia de software ou de sistemas em geral. Entretanto, no capítulo 10 são discutidos aspectos específicos do Gerenciamento de Requisitos em projetos ágeis.

O Gerenciamento de Requisitos sem o uso de ferramentas é difícil na prática. Portanto, as possibilidades e limitações do suporte a ferramentas são discutidas nos capítulos individuais sem, no entanto, colocar uma ferramenta especial em primeiro plano.

Nível de Detalhe

O nível de detalhamento deste Syllabus possibilita uma consistência de cursos e avaliações em âmbito internacional. Para atingir este objetivo, o syllabus contém o seguinte:

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

Objetivos Educacionais/ Níveis de Conhecimento Cognitivo

A todos os módulos e objetivos educacionais deste syllabus é atribuído um nível cognitivo. Os níveis são classificados da seguinte forma:

- **L1: Conhecer** (descrever, enumerar, caracterizar, reconhecer, nomear, lembrar, ...) – O candidato consegue se lembrar ou recuperar o material aprendido anteriormente.
- **L2: Compreender** (explicar, interpretar, completar, resumir, justificar, classificar, comparar, ...) – O candidato consegue entender/construir o significado de determinado material ou situação.
- **L3: Aplicar** (especificar, escrever, projetar, desenvolver, implementar, ...) – O candidato pode aplicar conhecimentos e habilidades em determinadas situações.

- **L4: Analisar** (investigar, concluir, fornecer argumentos para, ...) – O candidato pode analisar um determinado problema, argumentar o que deve/pode ser feito, dividir o problema em partes, aplicar o pensamento crítico, argumentar sobre causas e efeitos.
- **L5: Avaliar** (criticar, julgar) – O candidato pode fazer uma crítica bem argumentada de um determinado artefato; fazer um julgamento profundo em um determinado caso.

Observe que um objetivo educacional no nível de conhecimento cognitivo Ln também contém elementos de todos os níveis de conhecimento cognitivo inferiores (L1 a Ln-1).

Exemplo: Um objetivo educacional do tipo "Aplicar a técnica de ER xyz" está no nível de conhecimento cognitivo (L3). Entretanto, a capacidade de aplicar exige que os alunos conheçam primeiro a técnica de ER xyz (L1) e que entendam para que a técnica é usada (L2).



Todos os termos definidos no glossário devem ser conhecidos (L1), mesmo que não sejam mencionados explicitamente nos objetivos educacionais.

O glossário está disponível para download na página inicial do IREB em:

<https://www.ireb.org/en/downloads/#cpre-glossary-2-0>

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

Estrutura do Syllabus

O Syllabus consiste de 11 capítulos principais. Cada capítulo cobre uma unidade educacional (UE). Os títulos dos capítulos principais contêm o nível cognitivo de seus capítulos, que é o nível mais alto de seus subcapítulos. Além disso, o tempo de ensino que é sugerido é o mínimo que um curso deve investir nesse capítulo. As empresas de treinamento são livres para dedicar mais tempo às UEs e aos exercícios, mas certifique-se de que as proporções entre as UEs sejam mantidas. Os termos importantes utilizados em cada capítulo estão listados no seu início.

Exemplo:

Capítulo 3 Designação de Atributos e Visualizações para os Requisitos (L5)

Duração: 2 horas

Termos: Atributo, esquema de atributos, visualização

Esse exemplo mostra que o capítulo 3 contém objetivos educacionais no nível L5 e duas horas são destinadas ao ensino do material desse 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 a que subcapítulo eles pertencem.

Exemplo: OE 3.3.1

Este exemplo mostra que o objetivo educacional OE 3.3.1 está descrito no subcapítulo 3.3.

0 Exame

Este syllabus cobre as unidades e objetivos educacionais para os exames de certificação do

- Requirements Management – Practitioner
- Requirements Management – Specialist

O exame para obter o certificado Requirements Management – Practitioner – consiste em **um exame de múltipla escolha**.

O exame para obter o certificado Requirements Management – Specialist – consiste em um **trabalho escrito**.

Ambos os exames incluem questões que abrangem todas as unidades educacionais e todos os objetivos educacionais do syllabus.

Cada pergunta do exame pode incluir material de vários capítulos do syllabus, bem como de vários objetivos educacionais ou partes de um objetivo educacional.

O **exame de múltipla escolha** para o certificado **Practitioner**

- testa todos os objetivos educacionais do syllabus. No entanto, para os objetivos educacionais nos níveis de conhecimento cognitivo L4 e L5, as questões do exame estão limitadas a itens nos níveis cognitivos L1 a L3.
- pode ser feito imediatamente após um curso, mas também independentemente dele (p. ex., remotamente ou em um centro de testes).

O **trabalho escrito** para o certificado de **Specialist**

- testa todos os objetivos educacionais do syllabus nos níveis de conhecimento cognitivo indicados para cada objetivo educacional.
- segue a descrição da tarefa Requirements Management – Specialist –, encontrada em <https://www.ireb.org/en/downloads/tag:advanced-level-written-assignment#top>.
- é individualizado e enviado a um órgão de certificação licenciado.

Os objetivos educacionais genéricos a seguir também se aplicam ao **trabalho escrito** para o certificado de **Specialist**:

EO G1: Analisar e ilustrar problemas de Gerenciamento de Requisitos em um contexto com o qual o candidato esteja familiarizado ou que seja semelhante a esse contexto (L4).

EO G2: Avaliar e refletir sobre o uso de práticas, métodos, processos e ferramentas de Gerenciamento de Requisitos em projetos nos quais o candidato esteve envolvido (L5).

A lista de organizações de certificação licenciadas pelo IREB encontra-se no site <https://www.ireb.org>.

Histórico de versões

Data	Versão	Descrição das mudanças
31 de outubro de 2024	2.1.0	Versão Inicial baseada no Syllabus original do IREB em inglês

1	O que é o Gerenciamento de Requisitos? (L2)	11
1.1	Definição de Gerenciamento de Requisitos (L1)	11
1.2	Tarefas no Gerenciamento de Requisitos (L1)	12
1.3	Objetivos e benefícios do Gerenciamento de Requisitos (L2)	13
1.4	Plano de Gerenciamento de Requisitos (L2)	14
1.5	Padrões relevantes (L1)	14
2	Modelo de informações de requisitos (L3)	16
2.1	Fundamentos (L2)	16
2.2	Formas de apresentação (L2)	18
2.3	Criação de um modelo de informações de requisitos (L3)	20
3	Associação de atributos e visões sobre os requisitos (L5)	23
3.1	Objetivos da associação de atributos (L1)	23
3.2	Uso de um esquema de atributos (L2)	24
3.3	Projetando um esquema de atributos (L3)	25
3.4	Gerenciamento de mudanças de esquemas de atributos (L5)	27
3.5	Objetivos e tipos de visualizações (L2)	28
3.6	Definição de pontos de vista e riscos de pontos de vista (L1)	30
3.7	Otimização da população de atributos e criação de visualizações (L4)	31
4	Avaliação e priorização de requisitos (L3)	33
4.1	Princípios de avaliação (L1)	33
4.2	Priorização dos requisitos (L2)	34
4.3	Técnicas de priorização Ad-hoc (L3)	35
4.3.1	Classificação de dois critérios (L3)	35

4.3.2	Técnica 100-Dollar (L3)	36
4.4	Técnicas de priorização analítica (L3)	37
4.5	Combinação de técnicas de priorização (L2)	37
5	Gerenciamento de versão e mudança (L3)	38
5.1	Versionando os requisitos (L2)	38
5.1.1	Controle de versão para requisitos e documentos de requisitos (L1).....	38
5.1.2	Configuração de requisitos (L2)	39
5.1.3	Baseline dos requisitos (L1)	39
5.1.4	Ramificação de requisitos (L2)	40
5.2	Gerenciamento de mudanças para requisitos (L2)	40
5.2.1	Causas, fontes e cronograma de alterações de requisitos (L2).....	40
5.2.2	Tipos de alterações nos requisitos (L2)	41
5.2.3	Análise e documentação da estabilidade dos requisitos (L2).....	41
5.3	Processo de gerenciamento de mudanças (L3)	42
6	Rastreabilidade de requisitos (L3)	44
6.1	Razões para a rastreabilidade dos requisitos (L1)	44
6.1.1	O que significa rastreabilidade de requisitos? (L1).....	44
6.1.2	Por que a rastreabilidade de requisitos e artefatos é importante (L1).....	45
6.2	Diferentes visualizações de rastreabilidade (L2)	45
6.3	Tipos de relacionamento para relacionamentos de rastreabilidade (L1)	47
6.4	Formas de apresentação das relações de rastreabilidade (L3)	49
6.4.1	Documentação implícita e explícita de rastreabilidade (L3).....	49
6.4.2	Relações de rastreabilidade bidirecionais e unidirecionais (L2).....	49
6.4.3	Formas de apresentação das relações de rastreabilidade (L3).....	50
6.5	Desenvolvimento de uma estratégia para rastreabilidade específica do projeto (L3).....	54
6.6	Criação e uso de modelos específicos de rastreabilidade (L3)	55
6.6.1	Um processo para definir um modelo de rastreabilidade específico.....	55
6.6.2	Uso de um modelo de rastreabilidade específico	56

6.7	Medidas para a avaliação da rastreabilidade implementada (L2)	56
6.8	Desafios na rastreabilidade de artefatos não textuais (L1)	57
7	Gerenciamento de variantes para requisitos (L5)	59
7.1	Uso de variantes de requisitos (L1)	59
7.2	Formas de documentação explícita de variantes e sua avaliação (L5)	61
7.3	Modelagem de recursos (L3)	65
8	Relatórios sobre Gerenciamento de Requisitos (L3)	70
8.1	Objetivos e benefícios dos relatórios no Gerenciamento de Requisitos (L1).....	70
8.2	Estabelecimento de um sistema de relatórios no Gerenciamento de Requisitos (L2).....	71
8.2.1	Interfaces (L2)	71
8.2.2	Conteúdo de um relatório (L1)	71
8.2.3	Dicas para o desenvolvimento e a aplicação de relatórios (L2).....	73
8.2.4	Processo de definição de relatório (L2)	73
8.3	Figuras-chave na Engenharia de Requisitos (L3)	74
8.3.1	Figuras-chave no Gerenciamento de Requisitos (L1).....	74
8.3.2	Derivação de índices usando o método Meta-Questionário-Métrico (L3).....	75
8.4	Riscos e problemas nos relatórios (L2)	75
9	Gerenciamento dos processos de Engenharia de Requisitos (L3)	77
9.1	Engenharia de Requisitos como um processo (L2)	77
9.2	Parâmetros do processo de Engenharia de Requisitos (L3)	79
9.3	Documentar o processo de Engenharia de Requisitos (L3)	82
9.4	Monitoramento e controle do processo de Engenharia de Requisitos (L1)	83
9.5	Melhoria do processo de Engenharia de Requisitos (L3)	83

10	Gerenciamento de requisitos em projetos ágeis (L2)	86
10.1	Histórico (L1).....	86
10.2	Gerenciamento de requisitos em projetos ágeis (L1)	86
10.3	Mapeamento das atividades de Gerenciamento de Requisitos para as atividades do Scrum (L2).....	88
11	Uso de ferramentas no Gerenciamento de Requisitos (L1) ..	91
11.1	Papel das ferramentas no Gerenciamento de Requisitos (L1)	91
11.2	Procedimento básico para a seleção de ferramentas (L1)	92
11.3	Intercâmbio de dados entre ferramentas de Gerenciamento de Requisitos (L1).....	92
12	Referências	95

1 O que é o Gerenciamento de Requisitos? (L2)

Duração: 75 min

Termos: Engenharia de Requisitos, Gerenciamento de Requisitos, Gerente de Requisitos, Plano de Gerenciamento de Requisitos

Objetivos Educacionais

- OE 1.1 Conhecer a definição de Engenharia de Requisitos e Gerenciamento de Requisitos (L1)
- OE 1.2 Conhecer as tarefas de Gerenciamento de Requisitos (L1)
- OE 1.3 Conhecer os objetivos e benefícios do Gerenciamento de Requisitos (L2)
- OE 1.4 Conhecer a necessidade de um plano de Gerenciamento de Requisitos (L2)
- OE 1.5 Conhecer os padrões relevantes para o Gerenciamento de Requisitos (L1)

1.1 Definição de Gerenciamento de Requisitos (L1)

O Gerenciamento de Requisitos (RM) pode ser considerado sob duas perspectivas:

1. Gerenciar requisitos e artefatos de requisitos no processo de desenvolvimento
2. Gerenciamento das atividades de Engenharia de Requisitos (ou seja, RM como gerenciamento de processos)[POHL2010], capítulo 30.1

O RM pode ser usado no contexto de um projeto de desenvolvimento, na evolução de um sistema existente, no gerenciamento de produtos de software ou no gerenciamento contínuo de requisitos entre projetos.

O Glossário do IREB [IREB Glossary] define RM como um processo para gerenciar os requisitos existentes e os artefatos relacionados. Isso inclui o armazenamento, a modificação e o rastreamento de requisitos e outros artefatos. Entre outras coisas, isso inclui a estruturação, a preparação, a alteração consistente e a implementação de requisitos [RuSo2009].

Na literatura, é comum encontrar diferentes definições de Engenharia de Requisitos (RE) e Gerenciamento de Requisitos (RM). Dependendo da definição

- A Engenharia de Requisitos é uma parte do Gerenciamento de Requisitos (p. ex., em [SCHI2001]), ou
- O Gerenciamento de Requisitos é uma parte da Engenharia de Requisitos (p. ex., em [IREB2023]), ou
- A Engenharia de Requisitos e o Gerenciamento de Requisitos são definidos como aspectos coexistentes (p. ex., [CMMI2010]).

O IREB define o Gerenciamento de Requisitos **como parte** da Engenharia de Requisitos.

Na estrutura deste syllabus, os termos são definidos da seguinte forma:

- O termo **Engenharia de Requisitos** abrange as áreas de elaboração, documentação, verificação/reconciliação e Gerenciamento de Requisitos.

- O termo **Gerenciamento de Requisitos**, por outro lado, refere-se à tarefa de gerenciar requisitos dentro da Engenharia de Requisitos.

1.2 Tarefas no Gerenciamento de Requisitos (L1)

O Gerenciamento de Requisitos é responsável por fornecer informações sobre os requisitos de modo que outras funções no projeto possam trabalhar com elas de forma eficiente. Além disso, devem ser fornecidas regras e métodos que permitam o fornecimento de informações [RuSo2009].

Se os requisitos forem elicitados, três premissas básicas se aplicam, com base nas quais todas as tarefas de Gerenciamento de Requisitos podem ser derivadas e com as quais as tarefas e os métodos a seguir podem ser justificados (consulte também [RuSo2009]):

- Os requisitos devem ser usados por muitas pessoas;
- Mudança de requisitos;
- Os requisitos devem ser reutilizados.

As tarefas e os conceitos a seguir fazem parte do Gerenciamento de Requisitos [PoRu2011]:

- **Associação de atributos:** Os atributos permitem descrever os requisitos e suas metainformações de forma mais estruturada, agrupá-los ou torná-los comparáveis a outros requisitos. Os requisitos atribuídos são a base para a criação de visualizações de requisitos.
- **Avaliação e priorização:** As prioridades possibilitam a diferenciação entre requisitos importantes e menos importantes, complexos e menos complexos, com o auxílio de critérios de avaliação e priorização. Essa distinção, por sua vez, serve de base para as decisões de gerenciamento de projetos e planejamento de lançamentos.
- **Rastreabilidade:** A rastreabilidade possibilita o rastreamento de um requisito durante todo o ciclo de vida do sistema. Com base nessas informações, os requisitos dependentes e outros artefatos de desenvolvimento podem ser identificados, por exemplo, quando um requisito é alterado.
- **Controle de versão:** O controle de versão permite rastrear as alterações nos requisitos em seu ciclo de vida.
- **Relatórios:** O relatório é a coleta, a avaliação e a apresentação de informações sobre os requisitos ou sobre o processo de Engenharia de Requisitos (resumidamente: processo de ER). As informações contidas nos relatórios servem não apenas como informações puras, mas também como base para as decisões do projeto e para o controle do processo de ER.
- **Gerenciamento de processos:** O gerenciamento dos processos de trabalho da Engenharia de Requisitos permite projetar os processos da equipe de forma eficiente.

As tarefas de Gerenciamento de Requisitos são planejadas e executadas pelo engenheiro de requisitos ou pelo gerente de requisitos.

1.3 Objetivos e benefícios do Gerenciamento de Requisitos (L2)

O objetivo do Gerenciamento de Requisitos é gerenciar os requisitos e outros artefatos relacionados a eles, de modo que os requisitos possam ser sistematicamente verificados, agrupados, avaliados, alterados e rastreados com esforço razoável. Entre outras coisas, ele fornece respostas às seguintes perguntas:

- Quais requisitos fazem parte do sistema? (Associação de atributos)
- Quais requisitos vêm de qual fonte? (Associação de atributos e rastreabilidade)
- Quais requisitos são urgentes e importantes? (Avaliação e priorização)
- Qual requisito gera custos muito altos com poucos benefícios? (Avaliação e priorização)
- Quais requisitos do (sub)sistema pertencem a quais requisitos dos Stakeholders? (Rastreabilidade)
- Quais requisitos fazem parte de meus sistemas/produtos? (Rastreabilidade)
- Qual versão do requisito foi implementada em meu sistema? (Controle de versão)
- Quem alterou o requisito por último? (Controle de versão)
- Quais métricas-chave (indicadores-chave de desempenho) podem ser usadas para controlar minha Engenharia de Requisitos? (Relatório)
- Qual é a eficiência do meu trabalho de Engenharia de Requisitos? (Gerenciamento de Processo)

A importância do Gerenciamento de Requisitos no processo de desenvolvimento está intimamente relacionada às condições de limite do projeto [RuSo2009]. O Gerenciamento de Requisitos se torna ainda mais importante...

- ... quanto maior o número de requisitos,
- ... maior será a vida útil estimada do produto,
- ... quanto mais mudanças forem esperadas,
- ... quanto maior o número de participantes no processo de ER,
- ... mais difícil é alcançar ou envolver os Stakeholders,
- ... maiores serão as exigências de qualidade do sistema,
- ... quanto maior for a reutilização a ser realizada,
- ... quanto mais complexo for o processo de desenvolvimento,
- ... quanto mais heterogêneas forem as opiniões dos Stakeholders,
- ... mais versões serão desenvolvidas,
- ... mais importante é o uso de padrões.

Bom Gerenciamento de Requisitos ...[RuSo2009]

- ... aumenta a qualidade dos requisitos, produtos e processos,
- ... reduz os custos e a duração do projeto,
- ... facilita o monitoramento de projetos complexos em todas as fases,
- ... melhora a comunicação dentro e entre as equipes,
- ... aumenta a satisfação do cliente,
- ... reduz o risco do projeto.

1.4 Plano de Gerenciamento de Requisitos (L2)

Semelhante a um guia de gerenciamento de projetos, o plano de Gerenciamento de Requisitos (RMP) descreve as especificações para a implementação da Engenharia de Requisitos. Isso inclui o planejamento de quais tipos de requisitos devem ser documentados, como os requisitos são gerenciados e quem tem quais responsabilidades no processo de ER.

O Plano de Gerenciamento de Requisitos abrange no total:

- o Modelo de Informações de Requisitos (RIM), que descreve o cenário de requisitos, ou seja, os tipos de requisitos a serem gerenciados e seus níveis de detalhe (consulte 2),
- atributos e visões dos requisitos (consulte 3),
- critérios e métodos de priorização (consulte 4),
- diretrizes para o gerenciamento de versões de requisitos e artefatos de requisitos, bem como o processo de mudança (consulte 5),
- diretrizes para gerenciar a rastreabilidade dos requisitos (consulte 6),
- diretrizes para a descrição de variantes de requisitos (consulte 7),
- Diretrizes para relatórios sobre requisitos (consulte 8),
- Processo de ER com atividades, funções e responsabilidades (consulte 9),
- Diretrizes para as ferramentas a serem usadas (consulte 11).

Na prática, o Plano de Gerenciamento de Requisitos geralmente não é um documento independente, mas sim parte do guia de gerenciamento do projeto, do plano de gerenciamento de configuração ou de outros documentos de especificação do processo de desenvolvimento.

1.5 Padrões relevantes (L1)

O tópico de Gerenciamento de Requisitos é abordado em uma variedade de padrões e diretrizes de diferentes maneiras. Alguns representantes de destaque estão listados abaixo.

- O CMMI (Versão 1.3) [CMMI2010] considera, entre outros, os processos "Desenvolvimento de requisitos" e "Gerenciamento de requisitos", sendo que alguns dos objetivos atribuídos diferem significativamente das definições do IREB.
- ISO 9000 / ISO 9001 [ISO 9000] é um padrão para o gerenciamento da qualidade nas organizações. A ISO 9001 define os requisitos mínimos para um sistema de gerenciamento de qualidade e descreve, por exemplo, os requisitos para a realização do produto, bem como a medição e o aprimoramento e, portanto, aborda tópicos como identificabilidade ou rastreabilidade.
- As normas ISO/IEC 12207:2008 e 15288 ("Systems and software engineering – Software life cycle processes" e "Systems engineering – Systems life cycle processes") definem os processos mais importantes para o desenvolvimento de sistemas e software e, portanto, também descrevem tarefas na área de Engenharia de Requisitos e Gerenciamento de Requisitos.
- A IEC 61508 ("Functional safety of safety-related electrical/electronic/programmable electronic systems") trata da definição de

requisitos para a segurança funcional de sistemas. É dada atenção especial ao tópico de rastreabilidade.

- A ISO/IEC/IEEE 29148:2011 ("Engenharia de sistemas e software – Processos do ciclo de vida – Engenharia de Requisitos") descreve processos específicos para a Engenharia de Requisitos.
- SOX (Sarbanes–Oxley Act) [USCO2002], [NEMN2006] é uma lei federal dos EUA em resposta a escândalos contábeis com o objetivo de melhorar a confiabilidade dos relatórios das empresas listadas no mercado público de capitais dos EUA. Essencialmente, a Lei Sarbanes–Oxley exige que saibamos quem fez quais alterações e quando, e, portanto, também está relacionada às principais tarefas do Gerenciamento de Requisitos.

Devido a restrições externas (p. ex., requisitos explícitos do cliente ou especificações regulamentares), um ou mais desses padrões ou diretrizes podem ser aplicados e, portanto, a implementação do Gerenciamento de Requisitos também é obrigatória.

Alguns dos padrões usam definições diferentes de termos e, portanto, são apenas parcialmente compatíveis entre si.

2 Modelo de informações de requisitos (L3)

Duração: 2 horas

Termos: cenário de requisitos, tipo de requisito, formas de apresentação, modelo de informações de requisitos (RIM)

Objetivos Educacionais

- OE 2.1.1 Conhecer as classificações básicas de requisitos (L1)
- OE 2.1.2 Conhecer o modelo Twin Peaks (L2)
- OE 2.2 Conhecer formas típicas de apresentação de requisitos (L2)
- OE 2.3 Dominar e usar o Modelo de Informações de Requisitos (L3)

2.1 Fundamentos (L2)

Os requisitos podem ser classificados de acordo com vários critérios:

- Tipos de requisito
- Independência da solução (independência da solução)
- Nível de detalhe ou nível de abstração do requisito

A classificação por tipo de requisito é conhecida do IREB Foundation Level ([IREB2023], capítulo 1), [POHL2010]:

- Requisitos funcionais.
- Requisitos de qualidade
- Restrições

As características dos diferentes tipos de requisitos e sua categorização posterior são discutidas em detalhes em [POHL2010], [PoRu2011].

Em projetos específicos, esses tipos de requisitos são muitas vezes refinados ou são usados nomes diferentes para eles. Os esquemas de classificação detalhados para os requisitos podem ser encontrados, por exemplo, em [YOUN2004], [RuSo2009], [WiBe2013], [RoRo2014].

Além disso, os requisitos podem ser classificados ortogonalmente de acordo com seu grau de independência de solução [POHL2010]:

Alta independência de solução, por exemplo, metas: As metas descrevem a intenção do sistema sem entrar na implementação e, portanto, descrevem a forma mais independente de solução de um requisito.

Independência de solução média, por exemplo, cenários: Os cenários descrevem de forma exemplar as possíveis sequências de interação para atingir um ou mais objetivos. Eles são descritos do ponto de vista do usuário.

Baixa independência da solução, por exemplo, requisitos orientados à solução: Os requisitos orientados à solução descrevem os dados, as funções, o comportamento do sistema, os estados e a qualidade do sistema necessários para atingir as metas e implementar os cenários.

Na forma documentada, o site [POHL2010] também chama essas três classes de tipos de artefatos de requisitos.

Uma terceira classificação usada para diferenciar os requisitos é o nível de detalhe, também conhecido como nível de abstração. Os requisitos geralmente são descritos em vários níveis de detalhes: em um nível baixo de detalhe com poucos detalhes, para fornecer uma visão geral, e em um nível mais alto de detalhe com mais detalhes, para especificar os requisitos de forma completa e precisa.

Modelo Twin-Peaks

Os requisitos formam a base para o projeto do sistema. No entanto, a prática tem mostrado que a abordagem em cascata, na qual o sistema é projetado sequencialmente após a especificação dos requisitos, muitas vezes não corresponde à realidade [POHL2010], [CHSM2013]. Em vez disso, trata-se de um processo iterativo no qual os requisitos moldam a arquitetura de um sistema e vice-versa. Figura 1 ilustra essa relação no chamado modelo Twin-Peaks [NUSE2001]. O eixo vertical representa o nível de detalhe da descrição do sistema, enquanto o eixo horizontal representa a orientação crescente da descrição do problema para a implementação. A figura mostra que as descrições de requisitos mais detalhadas são desenvolvidas iterativamente em paralelo com arquiteturas de sistema mais detalhadas.

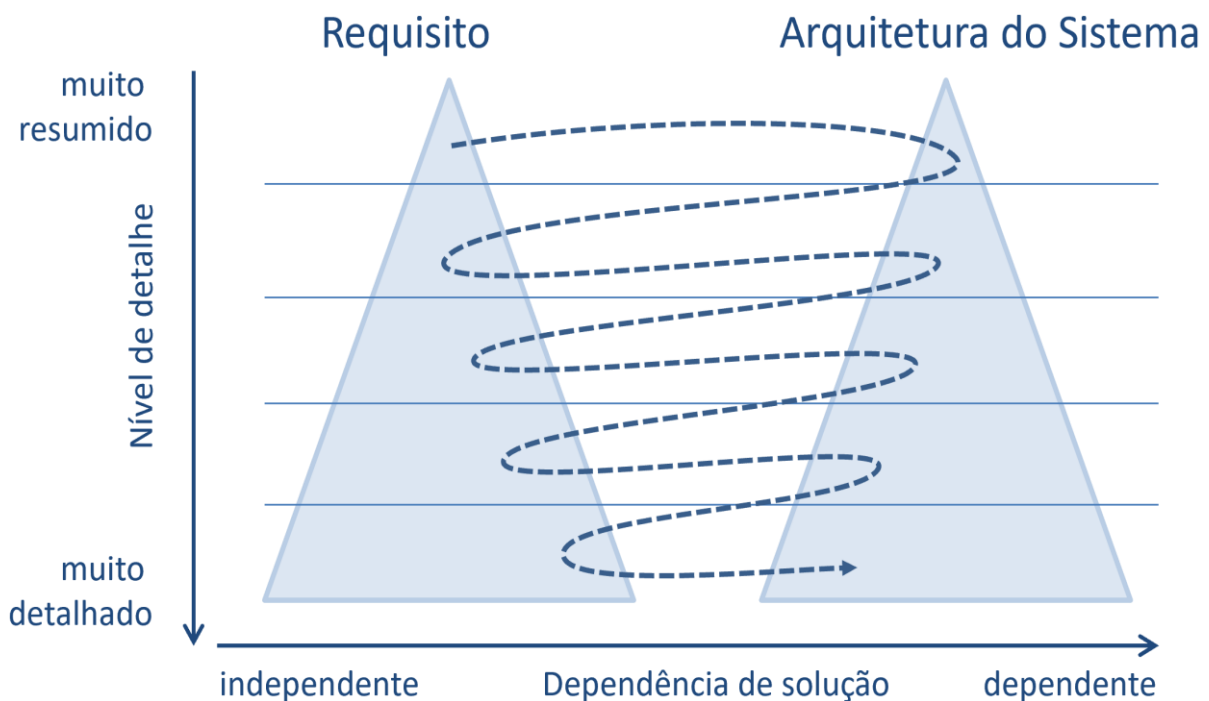


Figura 1: Modelo Twin-Peaks

A definição das camadas de abstração necessárias, ou seja, as camadas nas quais os diferentes níveis de detalhes dos requisitos são descritos, é específica do projeto e não pode ser descrita universalmente. No entanto, geralmente recomenda-se detalhar os requisitos até o momento:

- Todos os Stakeholders alcançaram um entendimento comum do requisito e está claro para todos exatamente o que é necessário.
- O grau de latitude restante para a solução é tão pequeno que um esclarecimento adicional geraria mais custos do que benefícios (ou seja, o risco residual resultante do grau de latitude restante é aceitável).
- Os requisitos são especificados na medida em que são claramente verificáveis (testáveis) em relação à solução subsequente.

Os níveis de abstração usados com frequência são, por exemplo: "sistema geral", "subsistema", "componente técnico" ou "requisitos do cliente e dos Stakeholders", "requisitos de projeto de sistema, arquitetura e componente", "requisitos de implementação" ou também a divisão em "especificações comerciais" e "especificações técnicas".

Dois aspectos desse procedimento são cruciais para o Gerenciamento de Requisitos:

- Para dar suporte a diferentes perspectivas sobre os requisitos, eles são especificados e armazenados em diferentes níveis de abstração em níveis explícitos de detalhes.
- Para que posteriormente esse processo incremental e iterativo faça sentido e para garantir que a integridade e a consistência sejam mantidas em diferentes níveis de detalhe, é essencial vincular os requisitos em diferentes níveis de detalhe, mas também os requisitos e seus elementos de implementação na arquitetura. Isso é obtido por meio de relações de rastreabilidade (consulte 6).

2.2 Formas de apresentação (L2)

Basicamente, os três tipos de requisitos, os três tipos de independência de solução e os níveis de detalhe selecionados resultam em vários "pontos de cruzamento" potenciais (consulte a Figura 2), doravante denominados classes de requisitos.

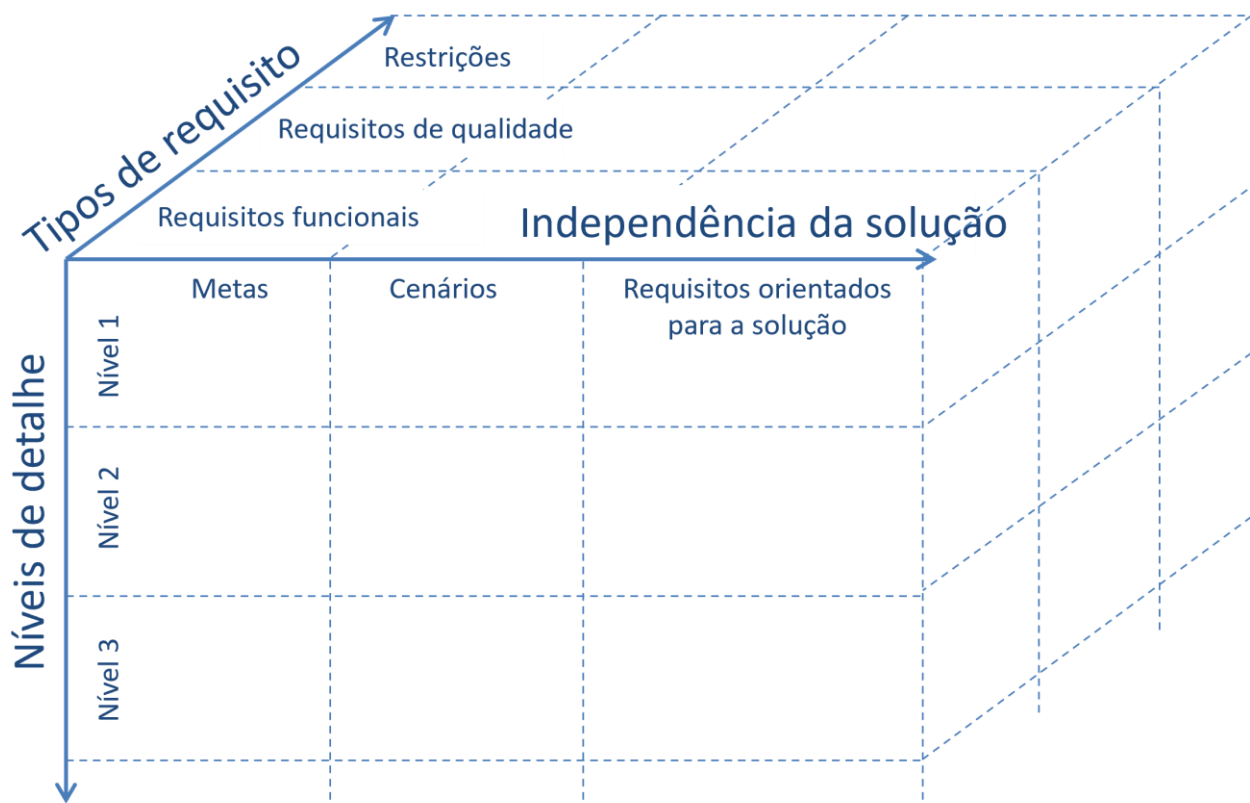


Figura 2: Diversas "interseções" potenciais (classes de requisitos)

Agora é preciso determinar (1) quais das "interseções" devem ser consideradas e (2) como elas devem ser documentadas. Um exame completo de todas as interseções normalmente não é economicamente possível e, muitas vezes, também não é significativo em termos de conteúdo.

Para a documentação dos requisitos, há diferentes formas de apresentação, graus de formalidade e notações/idiomas, por exemplo:

- Linguagem natural (p. ex., descrição em prosa pura)
- Texto estruturado (p. ex., modelos, padrões)
- Formas de notação gráfica baseadas em modelos (p. ex., UML, árvore de recursos)
- Descrições formais (p. ex., funções matemáticas)

Para cada "ponto de interseção" necessário, deve ser especificado qual forma de representação é necessária. Também é possível que sejam necessárias características diferentes para um "ponto de passagem", por exemplo, dependendo dos grupos de Stakeholders ou dos componentes do sistema. Por exemplo, no nível de detalhe do "subsistema", para o tipo "requisito funcional" e com uma independência de solução de "requisito orientado à solução", pode fazer sentido usar requisitos textuais (para documentar os requisitos gerais do cliente) ou diagramas de sequência de mensagens (para documentar os requisitos de interface).

Além disso, o idioma no qual os requisitos textuais e baseados em modelos devem ser documentados deve ser definido desde o início para evitar esforços adicionais desnecessários por meio de processos de tradução mais tarde.

2.3 Criação de um modelo de informações de requisitos (L3)

Em projetos reais, apenas algumas das possíveis "interseções" resultantes de todas as combinações matemáticas de tipos de requisitos, independência de soluções e níveis de detalhes serão consideradas.

Todas as "interseções" usadas em um projeto são chamadas de cenário de requisitos. Um cenário de requisitos é uma definição (a) da classificação a ser usada em relação aos tipos de requisitos, (b) da classificação a ser usada em relação à independência dos requisitos em relação às soluções, (c) dos níveis de detalhes necessários para cada tipo de artefato de requisito e (d) das formas de apresentação a serem usadas para cada tipo de artefato de requisito.

Em alguns casos, as normas e diretrizes relevantes já fornecem uma primeira indicação dos "pontos de passagem" a serem considerados em cada caso. Por exemplo, o site [ISO 26262] exige que os requisitos de segurança funcional e os requisitos de segurança técnica resultantes sejam derivados de objetivos de segurança (ou seja, uma forma especial de requisitos expressos como objetivos).

[RuSo2009] fornece um exemplo de cinco níveis de detalhes, que são chamados de níveis de especificação:

- **Nível de especificação 0:** descreve aproximadamente o projeto geral e suas metas
- **Nível de especificação 1:** descreve os casos de uso e os processos de negócios das áreas de negócios que cumprem as metas (especificação funcional)
- **Nível de especificação 2:** detalha os processos de negócios e os requisitos de negócios das áreas de negócios no nível de especificação 1 (especificação funcional)
- **Nível de especificação 3:** descreve os requisitos detalhados do usuário com divisão em subsistemas e uma descrição das interfaces (especificação funcional)
- **Nível de especificação 4:** descreve a especificação técnica com uma separação em hardware, software e outros componentes (especificação técnica)

O nível 0 de especificação inclui, portanto, todos os tipos de requisitos com "metas" de independência de solução e o nível de detalhe mais baixo (ou seja, mais grosseiro).

Esse exemplo também mostra que nem todos os níveis de detalhes são usados para documentar todos os requisitos com relação ao tipo de requisito e à independência da solução.

As definições do cenário de requisitos (ou seja, as "interseções" necessárias e a(s) forma(s) de representação escolhida(s)) devem ser documentadas como um Modelo de Informações de Requisitos (RIM). Por exemplo, isso pode ser feito por meio da criação de um diagrama entidade-relacionamento ou de um diagrama de classes. Um exemplo pode ser encontrado em Figura 3. O exemplo baseia-se nos níveis de especificação de [RuSo2009], mas não os implementa totalmente.

Além das especificações de tipo de requisitos, independência de soluções, nível de abstração e forma de representação já definidas, o modelo de informações de requisitos deve ser complementado por outros aspectos:

- Quais atributos serão usados onde? (ver 3)
- Quais pontos de vista serão apoiados? (ver 3)
- Quais critérios de avaliação estão planejados para os requisitos? (ver 4)
- Quais funções são responsáveis por manter e alterar quais informações? (ver 5)
- Quais relações de rastreabilidade entre os requisitos e os artefatos upstream e downstream serão documentadas? (ver 6)
- Como as variantes dos requisitos serão documentadas? (ver 7)

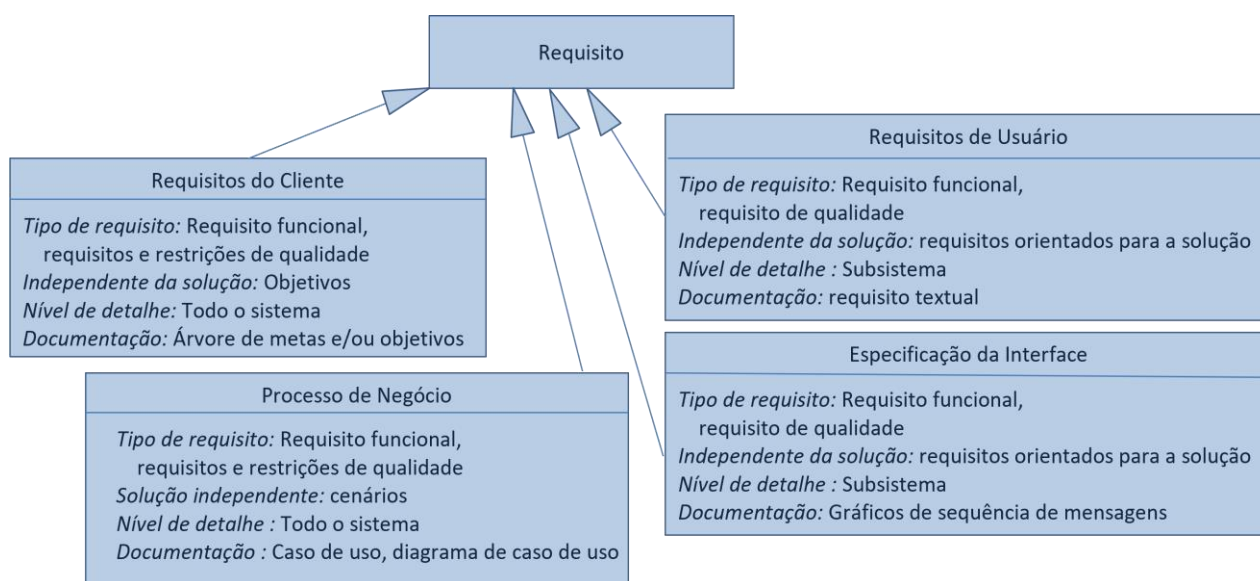


Figura 3: Exemplo de um modelo de informações de requisitos (RIM) específico

O RIM fornece a todos os Stakeholders de um projeto uma visão central dos artefatos de requisitos usados e seu uso no Gerenciamento de Requisitos e, portanto, é uma parte essencial do plano de Gerenciamento de Requisitos (RMP).

Ao definir o RIM, é sempre importante equilibrar os benefícios trazidos por uma documentação de requisitos mais abrangente e os custos decorrentes dela [GLIN2008], [DAVI2005].

Ao examinar um RIM, as seguintes perguntas de controle podem ser usadas:

- **Completo formal:** Está claro, para cada classe de requisitos, que tipo de requisito e grau de independência da solução ele tem, a que nível de detalhe ele é atribuído e com que forma(s) de representação ele é documentado?
- **Relações de conteúdo:** Está claro quais níveis de detalhes existem e como eles estão conectados? Está claro como os requisitos das diferentes classes de requisitos estão interconectados?

- **Adequação:** todas as classes de requisitos são apropriadas para fornecer requisitos suficientemente detalhados e completos para que as atividades de desenvolvimento dependentes possam ser executadas?

Em particular, a questão da adequação também depende muito do modelo de processo escolhido. As abordagens ágeis (cf. 10), por exemplo, concentram-se em "metas" e "cenários" independentes da solução e normalmente omitem os requisitos orientados para a solução (mais complexos). Por meio do contato próximo com o cliente, da criação de protótipos e de iterações curtas, as informações necessárias podem ser obtidas na hora certa.

3 Associação de atributos e visões sobre os requisitos (L5)

Duração: 2 horas

Termos: Atributo, esquema de atributos, visualização

Objetivos Educacionais

- OE 3.1 Conhecer o objetivo da associação de atributos e os seus benefícios nas atividades de gerenciamento (L1)
- OE 3.2 Conhecer as vantagens dos esquemas de atributos (L2)
- OE 3.3.1 Conhecer as etapas para projetar um esquema de atributos (L1)
- OE 3.3.2 Dominar e usar etapas predefinidas para projetar um esquema de atributos (L3)
- OE 3.4.1 Conhecer os tipos de mudanças nos esquemas de atributos (L2)
- OE 3.4.2 Avaliação de mudanças nos esquemas de atributos com base nos diferentes tipos de alterações (L5)
- OE 3.5 Conhecer os objetivos da criação de visualizações sobre requisitos e tipos de visualizações (L2)
- OE 3.6 Conhecer as etapas do processo de definição de visualizações (L1)
- OE 3.7 Analisar o grau de população de atributos (L4)

3.1 Objetivos da associação de atributos (L1)

O gerenciamento dos requisitos de produtos complexos durante todo o seu ciclo de vida requer a administração de uma grande quantidade de informações sobre esses requisitos. Por exemplo, para determinar quem apresentou um requisito conflitante no caso de um conflito, é necessário documentar a origem desse requisito.

O objetivo de atribuir atributos aos requisitos é permitir que os Stakeholders documentem informações sobre os requisitos de forma estruturada como parte do processo de ER [POHL2010], capítulo 18.2, [IREB2023], capítulo 8.1. Portanto, são definidos diferentes atributos nos quais essas informações (p. ex., fonte, data de criação, autor etc.) são registradas.

A associação de atributos aos requisitos fornece uma base para várias tarefas no Gerenciamento de Requisitos e para outras atividades de gerenciamento, por exemplo:

- **Rastreabilidade:** Os atributos são usados no Gerenciamento de Requisitos para realizar a rastreabilidade. Um pré-requisito, por exemplo, para a rastreabilidade da origem de um requisito – para que seja possível solicitar mais detalhes, por exemplo – é que a origem tenha sido documentada em um atributo apropriado. Além disso, os atributos também facilitam os vínculos com outros requisitos em cada caso, de modo que, por exemplo, a rastreabilidade possa ser estabelecida entre as metas e os requisitos orientados para a solução.
- **Visualizações:** A implementação de visualizações de requisitos geralmente se baseia em atributos (consulte 3.6).

- **Priorização:** A respectiva prioridade é documentada em um ou mais atributos correspondentes. Vários atributos podem ser definidos para diferentes conteúdos de priorização.
- **Gerenciamento de variantes:** Os atributos podem ser usados no gerenciamento de variantes (consulte 6) para atribuir requisitos a variantes específicas.
- **Relatórios:** Os atributos fornecem uma base para a geração de relatórios, como uma avaliação do número de requisitos em um determinado status (como "em andamento" ou "verificado").
- **Gerenciamento de projetos:** Se o esforço de implementação por requisito for coletado como parte do gerenciamento de projetos, esse esforço de implementação deverá ser registrado em um atributo apropriado. Além disso, também é necessário documentar o status de um requisito para apoiar o gerenciamento de projetos.
- **Gerenciamento de liberações:** A interface para o gerenciamento de versões é suportada por um atributo "Release" correspondente. Isso permite documentar quais requisitos foram implementados em qual versão. Em muitos casos, será feita uma distinção entre a versão desejada e a planejada para esclarecer as diferenças que frequentemente ocorrem entre essas duas versões.

3.2 Uso de um esquema de atributos (L2)

O conjunto de todos os atributos definidos para um tipo de requisito (p. ex., requisitos funcionais, requisitos de qualidade) é chamado de esquema de atributos ([PoRu2011], seção 8.1.2). O fornecimento de um esquema de atributos para os requisitos leva às seguintes vantagens no Gerenciamento de Requisitos [POHL2010], capítulo 18.2:

- **Definição precisa e consistente das informações necessárias:** Um esquema predefinido define quais informações ou atributos dos requisitos devem ser inseridos e quais valores são permitidos para essas informações.
- **Detecção de lacunas:** É possível detectar lacunas na elaboração de requisitos se determinados atributos não estiverem preenchidos.
- **Suporte para treinamento de funcionários:** Os funcionários que já trabalharam com esquemas de atributos iguais ou semelhantes em um projeto anterior, por exemplo, podem encontrar rapidamente as informações necessárias e onde as informações específicas sobre o requisito devem ser documentadas.
- **Encontrar as mesmas informações no mesmo local:** Como todos os requisitos em um projeto são documentados com base no mesmo esquema de atributos, é claramente especificado onde as informações – como o autor – podem ser encontradas para um requisito.

Os fatores típicos que influenciam a definição de um esquema de atributos são [PoRu2011], capítulo 8.1, [IREB2023], capítulo 8.1:

- Características específicas de um projeto, por exemplo, o tamanho do projeto
- Diretrizes fornecidas pela empresa, por exemplo, padrões da empresa
- Propriedades e regulamentos na área de aplicação, por exemplo, modelos de referência, padrões

- Condições de limite e restrições no processo de desenvolvimento, por exemplo, padrões de processo, convenções etc.

Além das vantagens motivadas pelo processo de ER, as metas de outros processos complementares também devem ser levadas em conta ao definir um esquema de atributos, por exemplo, para o Gerenciamento de Projetos (consulte também 3.1). As informações necessárias devem então ser mapeadas usando um tipo de atributo relevante para o requisito.

3.3 Projetando um esquema de atributos (L3)

A definição de um esquema de atributos deve ser feita antes de iniciar a documentação dos requisitos e deve ser acordada com todos os participantes do processo de ER. Os aprimoramentos e as alterações subsequentes geralmente só são possíveis com muito esforço [RuSo2009], capítulo 14.1.

As etapas a seguir são necessárias para criar um esquema de associação de atributos para uso em um projeto específico:

1. Identificação de fontes de atributos

Para selecionar atributos, primeiro é necessário identificar as fontes relevantes para os atributos. As fontes que podem ser usadas para selecionar atributos incluem:

- Um esquema de atributos de um projeto semelhante (p. ex., semelhante em escopo, número de funcionários envolvidos etc.)
- Um esquema de referência da organização ou outro padrão
- Regras de organização que determinam, por exemplo, quais atributos devem ser usados em todos os esquemas de atributos de todos os projetos
- Stakeholders do processo de ER

2. Seleção de atributos

Para que o uso do esquema de atributos no Gerenciamento de Requisitos seja bem-sucedido, é essencial que os atributos sejam selecionados de forma intencional e adequada ao projeto. Isso se aplica tanto ao uso de esquemas de referência quanto à definição de um esquema de atributos sem o uso de um esquema de referência.

Algumas etapas para a seleção de atributos estão listadas abaixo:

- Verificar os atributos no esquema de referência; se necessário, alterar ou complementar os atributos.
- Para a seleção de atributos para um esquema de atributos, bem como para a avaliação da integridade de um esquema de referência no escopo de um projeto, as sete categorias apresentadas em [POHL2010], capítulo 18.3: "Identificabilidade", "Relações de contexto", "Aspectos de documentação", "Aspectos de conteúdo", "Aspectos de conformidade", "Aspectos de validação" e "Aspectos de gerenciamento" podem ser úteis.
- Limitar os atributos a um conjunto pragmaticamente utilizável, concentrando-se nos objetivos do Gerenciamento de Requisitos no projeto [RuSo2009].

- Definição do contexto para a exclusividade do número de identificação (único no projeto, único na empresa etc.).

3. Definição de valores de atributos permitidos e propriedades de atributos

Após definir quais atributos são usados em um esquema de atributos, as propriedades dos atributos devem ser definidas, como, por exemplo

- De que tipo é o atributo em questão (texto, número, valor lógico, enumeração de datas etc.)?
- Quais valores dos tipos de enumeração devem estar disponíveis para seleção em um atributo correspondente?
- O atributo é um campo obrigatório?
- O atributo pode conter vários valores ou apenas um valor?
- Os auxílios ao usuário, como "todos os valores" ou "nenhum valor", estão disponíveis para seleção e quais são as consequências disso?

4. Definição de dependências entre atributos e seus valores

Os atributos podem ser interdependentes com relação aos seus valores. Por exemplo, é possível evitar que um requisito com o valor "volátil" no atributo "estabilidade" receba simultaneamente o valor "released" no atributo "status". Isso garante que, por exemplo, somente os requisitos considerados estáveis sejam aprovados para desenvolvimento.

Ao definir um esquema de atributos, também deve ser determinado se determinadas combinações não são permitidas para dois atributos com, por exemplo, valores de atributos predefinidos. Nesse caso, pode fazer sentido combinar esses dois atributos em um único atributo e oferecer apenas as combinações permitidas. Isso é particularmente útil se a ferramenta usada não suportar a análise de dependências entre valores de atributos.

No Gerenciamento de variantes, a atribuição de requisitos a variantes específicas pode ser proibida. Também é possível que determinadas combinações de variantes não sejam permitidas.

As dependências entre os atributos e seus valores também podem surgir por meio da hierarquização dos requisitos. Por exemplo, se o requisito A for detalhado pelos requisitos A.1 e A.2, será necessário determinar, para os atributos do esquema de atribuição, se o valor de A depende ou não dos valores de A.1 e A.2.

5. Fornecimento de suporte para registro de dados

A coleta de informações adicionais sobre o requisito real representa um esforço adicional para o engenheiro de requisitos, o que, em alguns casos, beneficia outros Stakeholders (p. ex., gerentes de projeto) em vez dele próprio. Esse é um dos motivos pelos quais é muito útil que o engenheiro de requisitos tenha o máximo de apoio possível para coletar essas informações.

Esse tipo de suporte é fornecido principalmente (ou até mesmo exclusivamente) pelo uso de uma ferramenta apropriada para registrar e gerenciar requisitos. Um exemplo simples disso é a definição de valores padrão para atributos.

6. Documentar o esquema de atributos

Os esquemas de atributos são definidos em um formato tabular ou em um modelo de informações, dependendo do grau de complexidade (p. ex., com relação ao número de atributos, dependências entre atributos ou seus valores) (consulte 2). A definição em um modelo de informações é usada por uma ferramenta que mapeia o esquema de atributos correspondente e ajuda o engenheiro de requisitos a documentar as informações associadas em um projeto.

3.4 Gerenciamento de mudanças de esquemas de atributos (L5)

Alterações subsequentes em um esquema de atributos durante o curso do projeto devem ser evitadas, se possível [RuSo2009], capítulo 14.1.

As consequências de uma alteração posterior em um esquema de atributos dependem do tipo de alteração:

- **Adição, alteração ou exclusão de um atributo:**

Ao adicionar um novo atributo, é necessário avaliar o tempo necessário para manter os requisitos documentados anteriormente em termos do novo atributo. Se um atributo tiver que ser excluído, isso poderá ter consequências negativas se, por exemplo, em uma ferramenta, determinados módulos consultarem o atributo. Em vez de excluí-lo, é possível adicionar "(não mais usado)" ao seu nome.

- **Adicionar, alterar ou excluir os possíveis valores de atributos (intervalo de valores):**

A adição de valores de atributos geralmente não é um problema para a ferramenta subjacente. Do ponto de vista técnico, é preciso examinar se a adição resulta em um novo significado para os valores existentes de um atributo, de modo que os requisitos avaliados anteriormente em relação a esse atributo tenham que ser analisados novamente e, se necessário, novos valores sejam definidos. Ao excluir valores de atributos de um intervalo de valores, é importante garantir que os requisitos não se tornem inconsistentes devido a entradas vazias. Os problemas são causados principalmente por campos obrigatórios, pois o requisito deve ter um valor no atributo em questão. Nesse caso, uma solução pode ser inserir um valor padrão no campo.

Quando os valores dos atributos são alterados, é importante garantir que as alterações sejam feitas em todos os requisitos que contêm o valor original. Em geral, ao alterar ou excluir valores de requisitos, é importante decidir se essa alteração afetará os requisitos que já foram inseridos ou se aplicará apenas a requisitos futuros. Para atributos dependentes com valores de atributos dependentes, é preciso garantir que a remoção de um valor de atributo não resulte em dados inconsistentes.

- **Adição ou exclusão de relações entre atributos:**

A adição de um relacionamento pode levar a inconsistências no conjunto de requisitos já capturados. Por exemplo, se uma definição for adicionada de forma que a seleção de um valor para o atributo "Estabilidade" sempre leve ao preenchimento do atributo "Risco" (e, portanto, sempre seja preenchido quando "Estabilidade" for preenchido), os requisitos com um valor para "Estabilidade", mas nenhum para "Risco", deverão ser verificados novamente.

Se a dependência entre os atributos for removida, você deverá garantir que os valores padrão (no sentido de: Se o valor do atributo A for selecionado, então, por exemplo, o valor do atributo B será automaticamente predefinido por uma ferramenta) também serão removidos.

- **Alteração dos valores padrão do tipo de atributo:**

A alteração dos valores padrão deve afetar inicialmente apenas a entrada de novos requisitos. Nesse contexto, no entanto, os requisitos aos quais foi atribuído o valor padrão anterior devem ser analisados para verificar se ainda têm o valor correto ou se devem ser ajustados.

- **Alterar a propriedade de "Campos obrigatórios" e "Campos opcionais":**

A alteração de um campo obrigatório para um campo opcional geralmente não resulta em nenhum esforço posterior. Se for planejada uma alteração de um atributo opcional para um atributo obrigatório, deve-se garantir que o atributo seja preenchido com um valor apropriado para todos os requisitos já documentados. Pode ser necessário atribuir valores personalizados em vez de usar um valor padrão.

Em geral, ao fazer alterações nos esquemas de atributos, é necessário analisar até que ponto as ferramentas subjacentes são afetadas. Por exemplo, se tiverem sido criados scripts na ferramenta DOORS que verificam ou processam um determinado atributo, uma alteração correspondente no atributo pode fazer com que os scripts não sejam mais executáveis.

3.5 Objetivos e tipos de visualizações (L2)

Uma vantagem de definir esquemas de atributos é a possibilidade de definir visualizações para o modelo de informações e implementá-las como parte do suporte da ferramenta (consulte também [IREB2023], capítulo 8.2).

Em um projeto, a complexidade das informações sobre os requisitos pode ser muito alta, especialmente quando muitos requisitos a serem documentados. Além disso, muitos Stakeholders estão envolvidos no projeto com relação aos requisitos. Além do engenheiro de requisitos, há, por exemplo, desenvolvedores, testadores ou gerentes de projeto.

O objetivo de oferecer visualizações específicas é fornecer aos respectivos participantes informações específicas e direcionadas sobre os requisitos e, assim, reduzir a complexidade da perspectiva do participante específico. Além disso, o acesso aos requisitos pode ser controlado em uma base específica de função por meio da criação de visualizações (consulte também [POHL2010], capítulo 18.5).

As visualizações podem ser classificadas da seguinte forma:

- **Visualizações seletivas:** As visualizações seletivas limitam o número de requisitos a serem considerados usando um filtro predefinido.
- **Visualizações projetivas:** As visualizações projetivas ocultam atributos que não são relevantes para a visualização considerada.

- **Visualizações compactadas:** As visualizações resumidas permitem avaliações (p. ex., criação de totais) de um conjunto de requisitos com referência a determinados atributos.
- **Combinação de visualizações seletivas, projetivas e agregadas:** Nas visualizações combinadas, determinados atributos são ocultos, o conjunto de requisitos é restrito com relação a um ou mais critérios e avaliado separadamente usando um atributo, por exemplo.

As ferramentas que oferecem visualizações de requisitos geralmente também podem ser classificadas por diferentes atributos [RuSo2009], capítulo 14.1. Isso permite que os participantes, em suas respectivas funções, mudem o foco dos requisitos (p. ex., todos os requisitos com alto esforço de implementação no início de uma lista) sem ocultar os requisitos. Além disso, as ferramentas geralmente permitem uma estruturação adicional das visualizações, agrupando os requisitos de acordo com um atributo.

3.6 Definição de pontos de vista e riscos de pontos de vista (L1)

O processo de definição de visualizações inclui as seguintes etapas:

- **Determinar os Stakeholders que precisam de uma ou mais visualizações:** As visualizações podem fornecer diferentes perspectivas sobre os requisitos. Para definir as visualizações necessárias, primeiro é preciso definir todos os participantes que usarão as visualizações dos requisitos. Os Stakeholders servem, então, como fonte para definir uma visão.

Reutilização: As visualizações de outros projetos ou de um projeto de referência também podem ser usadas como fonte de visualizações a serem definidas.

Determinação dos objetivos de uma visão em relação aos diferentes Stakeholders: Dependendo dos respectivos Stakeholders, é preciso determinar qual objetivo será perseguido com a visão específica. Isso pode ser usado, por exemplo, para determinar se a consolidação é necessária ou qual ordenação deve ser definida inicialmente. Nesse contexto, os direitos e as visões dos participantes também devem ser determinados, ou seja, qual participante deve ser capaz de ativar qual visão. Ao fazer isso, faz sentido que uma visão possa ser usada por muitos interessados.

Determinação dos atributos necessários e comparação com o esquema de atributos: Para poder cumprir as metas de uma visualização, é preciso garantir que seja possível coletar as informações necessárias e que os atributos correspondentes também estejam disponíveis. No entanto, a comparação com o esquema de atributos geralmente leva à geração de novas visões, porque somente quando o esquema de atributos é examinado de perto é que os Stakeholders percebem, por exemplo, quais avaliações ainda seriam possíveis. A definição de visualizações e esquemas de atributos é, portanto, um processo altamente integrativo; ambos os processos influenciam um ao outro.

Implementação da visualização: Por fim, as visualizações predefinidas devem ser implementadas e testadas na ferramenta subjacente.

A criação de visualizações traz alguns riscos. Em muitos casos, os usuários de um sistema de Gerenciamento de Requisitos não sabem que a grande quantidade de informações sobre um requisito pode ser restringida por visualizações, conforme necessário.

Eles costumam trabalhar com uma visão global e abrangente e percebem a ferramenta como muito extensa e possivelmente obstrutiva em seu trabalho. Há também o perigo de definir visões nas quais se perde muito do contexto. Por exemplo, se for criada uma visualização na qual os requisitos atômicos são fornecidos em uma lista sem nenhum contexto (p. ex., casos de uso), essa visão geral só será significativa até certo ponto. Para evitar ao máximo essas visualizações ineficazes, é muito importante considerar o objetivo subjacente de uma visualização.

3.7 Otimização da população de atributos e criação de visualizações (L4)

A prática mostra que, em alguns projetos, os atributos não são preenchidos na medida esperada. Portanto, recomenda-se verificar regularmente e em pontos específicos de um projeto se, e como, um atributo deve ser usado [RuSo2009], capítulo 14.1.

Uma maneira de garantir o grau de população do atributo no contexto do uso da ferramenta é definir o atributo como um "campo obrigatório". Isso força a entrada quando um requisito é criado. Nesse contexto, deve-se observar que a definição de muitos campos obrigatórios pode dificultar muito a inserção de requisitos, pois, por exemplo, as informações podem ainda não estar disponíveis quando um requisito é inserido pela primeira vez. Por esse motivo, os campos obrigatórios só devem ser declarados como tal com parcimônia e cuidado.

Um pré-requisito para verificar a captura de atributos é uma avaliação correspondente, que geralmente pode ser realizada dentro do escopo de uma ferramenta usada.

Para atributos opcionais que não precisam necessariamente ser inseridos, as seguintes consequências podem resultar de sua avaliação:

- **O atributo não foi usado em uma visualização ou em um relatório:** Esse caso indica que o atributo em questão não tem um destino específico, pois não está visível em nenhuma visualização ou análise.
- **O atributo é sempre preenchido com o mesmo valor, por exemplo, o valor padrão:** Nesse caso, não parece haver nenhuma distinção real no contexto do atributo para os diferentes requisitos, o que sugere que a lista de valores proposta não é adequada. O engenheiro de requisitos deve verificar se o atributo pode ser omitido (porque não há um objetivo real para ele) ou se a lista de seleção deve ser ajustada. Nesse último caso, as notas do site 3.4 devem ser levadas em consideração.
- **O atributo nunca é preenchido:** Se o atributo não tiver sido preenchido deliberadamente, as informações podem não ser importantes. Se essa suposição for confirmada, o atributo deverá ser removido. No entanto, o motivo muitas vezes reside no fato de que os usuários não estão cientes da intenção do atributo ou não veem diretamente nenhum benefício, já que as informações são usadas por outra parte interessada. Em seguida, os usuários devem ser (re)treinados para explicar a eles o valor agregado do atributo específico. Nesse caso, o atributo ainda pode ser usado.
- **O atributo é preenchido apenas para alguns requisitos:** A questão aqui é se o objetivo associado ao atributo pode ser alcançado ou se ele ainda é relevante. Se esse não for o caso, o atributo poderá ser removido. No entanto, se for constatado que o atributo é importante, ele pode ser declarado como um campo obrigatório, o que força a entrada no futuro. Nesse caso, os requisitos que não têm valor no atributo em consideração devem ser atualizados retrospectivamente (p. ex., população automática com um valor padrão).
- **O atributo não é preenchido em casos individuais:** Primeiro, é necessário determinar se esse atributo ainda é relevante para o projeto. Em caso afirmativo, o engenheiro

de requisitos deve preencher os requisitos relevantes. Se o atributo não for mais considerado essencial, ele pode ser removido ou as lacunas podem ser toleradas.

- **O atributo é sempre preenchido:** Nesse caso, nenhuma outra atividade é necessária.

Se o feedback dos Stakeholders do processo de ER resultar em novos requisitos com relação às visões definidas e implementadas, eles deverão ser examinados, priorizados e, quando necessário, implementados. Esses requisitos para visualizações podem levar ao Gerenciamento de mudanças do esquema de atributos se, por exemplo, for necessário um atributo adicional ou um valor de atributo adicional (consulte 3.4).

4 Avaliação e priorização de requisitos (L3)

Duração: 1,75 horas

Termos: priorização, avaliação, técnicas de priorização, técnicas de priorização ad hoc, técnicas de priorização analítica

Objetivos Educacionais

- OE 4.1 Conhecer as características básicas de uma avaliação (L1)
- OE 4.2.1 Conhecer a relação entre avaliação e priorização (L2)
- OE 4.2.2 Conhecer o procedimento básico de priorização (L1)
- OE 4.3 Dominar e usar técnicas de priorização ad hoc (L3)
- OE 4.4 Dominar e usar técnicas analíticas de priorização (L3)
- OE 4.5 Conhecer o procedimento de combinação de técnicas de priorização (L2)

4.1 Princípios de avaliação (L1)

Em todas as atividades da Engenharia de Requisitos, os requisitos são avaliados com base em vários critérios. Assim, durante a elicitação, por exemplo, os requisitos são categorizados de acordo com os critérios de Kano (consulte [IREB2023]), enquanto durante a documentação eles são diferenciados de acordo com sua natureza juridicamente vinculativa, ou os requisitos são avaliados de acordo com sua criticidade devido a diretrizes e padrões a serem seguidos (como [ISO 26262]).

A tarefa do Gerenciamento de Requisitos é documentar esses resultados da avaliação em um formato apropriado (consulte 3) e prever as consequências da avaliação para o processo de ER (consulte 9).

Por exemplo, a avaliação de um requisito pode ter um efeito sobre o tratamento posterior do requisito avaliado. Um requisito que tenha sido avaliado como particularmente crítico para a segurança poderia, por exemplo, ser submetido a um controle de qualidade mais detalhado durante os testes do que um requisito considerado não crítico.

Os possíveis critérios de avaliação são, por exemplo:

- A natureza juridicamente vinculativa de um requisito,
- Esforço/custos de implementação,
- Criticidade,
- Estabilidade,
- Grau de inovação.

As fontes para os critérios de avaliação são, por exemplo:

- Padrões de gerenciamento de projetos,
- Diretrizes e padrões,
- O esquema de atributos de requisitos,
- As disciplinas de desenvolvimento subsequentes.

É importante definir as seguintes características básicas de cada avaliação ao definir os critérios de avaliação necessários:

- Faixas de valores da avaliação (p. ex., "baixo", "médio" e "alto" para o critério de avaliação "Criticidade"),
- O grupo de pessoas que pode realizar a avaliação (p. ex., a "relevância arquitetônica" só deve ser avaliada por arquitetos),
- O primeiro momento para a avaliação (p. ex., o esforço de implementação só pode ser estimado quando um projeto detalhado estiver disponível),
- O último momento para a avaliação (p. ex., o esforço de implementação deve ser estimado antes que um requisito possa ser atribuído a uma versão ou a uma iteração de desenvolvimento).

Os resultados das avaliações que já foram realizadas podem ser usados no decorrer do projeto para tomar decisões relacionadas a todas as atividades de Engenharia de Requisitos [POHL2010]. Assim, as avaliações podem servir como base para a consolidação usando técnicas de consolidação analítica (consulte [IREB2023]) ou como base para priorizar os requisitos.

4.2 Priorização dos requisitos (L2)

Não há nenhum projeto em que os recursos disponíveis sejam ilimitados. Para garantir que os recursos disponíveis sejam usados de forma orientada para as metas, é essencial priorizar os requisitos existentes. Em princípio, a priorização deve sempre seguir o seguinte processo:

- Determinação das metas de priorização
- Definir os critérios de priorização
- Determinação dos Stakeholders envolvidas na priorização
- Determinação dos requisitos a serem priorizados
- Seleção da técnica de priorização
- Ajustar o esquema de atributos, se necessário
- Realização da priorização.
- Verificação regular e, se necessário, redefinição de prioridades dos requisitos

Os critérios de priorização necessários dependem do objetivo da priorização. Portanto, a primeira etapa é determinar quais decisões devem ser tomadas com base nos resultados da priorização. Portanto, ao decidir quais funções do sistema devem ser especificadas em detalhes primeiro, outros critérios de priorização devem ser considerados em comparação com a determinação da ordem de implementação. Por esse motivo, faz sentido determinar no início do projeto quais priorizações serão necessárias durante o curso do projeto e quais avaliações devem ser realizadas antes da priorização.

Dependendo da meta perseguida, é necessário determinar qual critério de avaliação ou priorização, ou qual combinação de critérios, é usado para determinar a prioridade do requisito.

Ao selecionar os requisitos a serem priorizados, deve-se observar que os requisitos devem estar no mesmo nível de detalhe para evitar distorcer o resultado da priorização [WiBe2013].

4.3 Técnicas de priorização Ad-hoc (L3)

Há várias técnicas para priorizar requisitos, que às vezes diferem muito em termos do esforço necessário para a priorização e da adequação a determinadas situações de priorização. "Em projetos, técnicas simples de priorização ad-hoc [...] são uma abordagem pragmática [...] para a priorização eficaz de requisitos" [POHL2010]. As técnicas de priorização ad-hoc a seguir funcionam bem na prática:

- Triagem de Requisitos [DAVI2003]
- Ranking [PoRu2011]
- Técnica Top-Ten [PoRu2011]
- Classificação de critério único [PoRu2011]
- Planning poker
- Classificação de dois critérios
- Técnica 100-Dólar
- Classificação Kano [PoRu2011]

4.3.1 Classificação de dois critérios (L3)

Se o resultado de uma classificação de critério único não for suficientemente diferenciado, por exemplo, porque vários critérios de priorização precisam ser considerados ou porque muitos requisitos estão dentro da mesma especificação do critério selecionado, os valores de vários critérios podem ser multiplicados e os resultados priorizados por meio de uma classificação. Cada requisito é agora atribuído a uma possível combinação de valores e, portanto, recebe a priorização atribuída a essa combinação. Esse procedimento geralmente é apresentado na forma de matrizes. Um exemplo de matriz com os critérios de priorização *Benefício para o cliente* e *Custos* é mostrado em Figura 4.

O princípio Eisenhower, nomeado em homenagem ao ex-presidente dos EUA, também é muito popular e propõe uma classificação baseada em critérios de *Importância* e *Urgência* (consulte Figura 5).

		Custo		
		Alta	Média	Baixa
Benefício para o cliente	Alta	Prioridade 5	Prioridade 2	Prioridade 1
	Média	Prioridade 6	Prioridade 4	Prioridade 3
	Baixa	Prioridade 9	Prioridade 8	Prioridade 7

Figura 4: Exemplo de classificação de dois critérios

		Urgência	
		Urgente	Não urgente
Importância	Importante	Prioridade 1	Prioridade 2
	Não é importante	Prioridade 3	Prioridade 4

Figura 5: Classificação de dois critérios por importância e urgência

4.3.2 Técnica 100-Dollar (L3)

A técnica dos 100 dólares é adequada para priorizar alguns requisitos. Recomenda-se o uso dessa técnica para requisitos mais grosseiros em um nível mais alto de detalhes ou para grupos de requisitos.

Os Stakeholders recebem 100 unidades imaginárias (dinheiro, tempo etc.), que podem ser distribuídas entre os requisitos. Os Stakeholders envolvidos têm um período definido para refletir sobre a alocação dos recursos disponíveis antes de alocá-los aos requisitos. Quanto mais unidades um requisito tiver recebido no final, maior será a prioridade desse requisito.

Teoricamente, a técnica também funciona com quantidades maiores de unidades (1000, 10000, ...). No entanto, isso custa muito mais tempo e esforço. Recomenda-se o uso de uma ferramenta (p. ex., software) para verificar a soma das unidades alocadas por parte interessada.

Essa técnica só deve ser aplicada uma vez a um conjunto específico de requisitos, pois os Stakeholders podem ser influenciadas pela distribuição de outros Stakeholders e, assim, distribuir suas unidades de forma diferente na próxima vez.

4.4 Técnicas de priorização analítica (L3)

Em alguns casos, o uso de técnicas de priorização ad-hoc é fortemente influenciado pelos Stakeholders envolvidas no processo de priorização e, portanto, só é recomendado de forma limitada para decisões muito críticas. Resultados de priorização mais neutros podem ser obtidos usando as seguintes técnicas de priorização analítica.

- **Matriz de priorização de Wieggers:** A matriz proposta por Karl Wieggers para priorizar os requisitos compara a vantagem relativa e a desvantagem relativa de cada requisito com os custos relativos e o risco relativo de cada requisito. Dessa forma, os requisitos com alto benefício para o cliente e, ao mesmo tempo, baixos custos e riscos recebem prioridade mais alta do que os requisitos com benefício comparativamente baixo para o cliente e os mesmos custos e riscos [WiBe2013].
- **Processo de Hierarquia Analítica (AHP):** A ideia básica por trás do AHP é a comparação em pares de todos os requisitos a serem priorizados. Uma escala de 1 a 9 determina o quanto o requisito A é mais importante em comparação com o requisito B [KaRy1997].

4.5 Combinação de técnicas de priorização (L2)

As técnicas de priorização analítica têm uma escala muito ruim em comparação com as técnicas de priorização ad-hoc. Assim, tanto para o Processo de Hierarquia Analítica quanto para a matriz de priorização de Wieggers, recomenda-se um número máximo de 25 a 30 requisitos para priorização, a fim de manter a complexidade e o tempo necessários para os métodos dentro de limites gerenciáveis [WiBe2013], [MOIS2002].

Como muitos projetos trabalham com um número muito maior de requisitos e a priorização detalhada na faixa inferior da escala de priorização oferece pouco valor agregado, uma combinação de técnicas de priorização ad-hoc e analíticas tem se mostrado bem-sucedida.

Por exemplo, ao planejar uma versão, se for necessário decidir quais requisitos devem ser implementados nessa versão, todos os requisitos serão priorizados ad hoc em uma primeira etapa. A Triagem de Requisitos [DAVI2003], uma Classificação de Critério Único inspirada no domínio médico, ou o Princípio Eisenhower, são amplamente utilizados aqui.

Se a técnica de priorização ad-hoc tiver identificado mais ou menos requisitos a serem implementados do que podem ser implementados na versão planejada, os requisitos próximos ao limite de aceite são priorizados usando uma técnica de priorização analítica e os requisitos com a prioridade mais alta são transferidos para a versão ou, ao contrário, aqueles com a prioridade mais baixa são removidos.

5 Gerenciamento de versão e mudança (L3)

Duração: 2 horas

Termos: controle de versão, configuração de requisitos, baseline de requisitos, liberação, alteração de requisitos, processo de gerenciamento de mudança

Objetivos Educacionais

- OE 5.1.1 Conhecer as atividades de controle de versão (L1)
- OE 5.1.2 Conhecer as características da Configuração de Requisitos (L2)
- OE 5.1.3 Conhecer as tarefas de desenvolvimento apoiadas por linhas de base de requisitos (L1)
- OE 5.1.4 Conhecer a necessidade e as desvantagens da ramificação de requisitos (L2)
- OE 5.2.1 Conhecer os principais motivos das mudanças de requisitos (L2)
- OE 5.2.2 Conhecer os tipos de mudanças de requisitos (L2)
- OE 5.2.3 Conhecer a heurística para avaliar a estabilidade/volatilidade dos requisitos (L2)
- OE 5.3.1 Conhecer o objetivo e as tarefas de um Conselho de Controle de Mudanças (CCB) (L1)
- OE 5.3.2 Dominar e usar o processo de Gerenciamento de Mudanças (L3)

5.1 Versionando os requisitos (L2)

5.1.1 Controle de versão para requisitos e documentos de requisitos (L1)

O controle de versão dos requisitos é uma parte essencial do Gerenciamento de Requisitos [WiBe2013]. O controle de versão de requisitos refere-se ao processo que permite que estágios específicos de desenvolvimento de requisitos e documentos de requisitos sejam mantidos disponíveis durante todo o ciclo de vida de um sistema ou produto [IREB2023]. Assim, os Stakeholders podem rastrear o histórico dos requisitos e dos documentos de requisitos e sempre podem se referir claramente a um status específico de um requisito ou de um documento de requisitos. Essas possibilidades são úteis em todos os projetos, mas apoiam especialmente os projetos colaborativos.

De acordo com [WiBe2013], o controle de versão consiste em três atividades principais:

1. Definição de um esquema para identificar versões
2. Identificação de versões de requisitos individuais
3. Identificação de artefatos de requisitos

No mesmo projeto, os requisitos podem ser versionados tanto no nível dos requisitos individuais (consulte a atividade 2) quanto no nível dos artefatos de requisitos (consulte a atividade 3). O controle de versão no nível do documento permite apenas uma prova aproximada das alterações, enquanto no nível dos requisitos cada alteração em um requisito é rastreável com precisão. Portanto, o controle de versão no nível do requisito é muito mais complexo do que no nível do documento.

As versões de um requisito ou documento de requisitos devem ser identificadas de forma exclusiva. Para esse fim, um esquema de rotulagem correspondente deve ser definido e usado em cada projeto.

Todos os participantes do projeto devem ter acesso à versão atual dos requisitos. As mudanças devem ser claramente documentadas e comunicadas às pessoas afetadas [WiBe2013].

Ao criar uma versão de um requisito ou documento de requisitos, é importante registrar o participante responsável do projeto, a hora da alteração e o motivo da alteração [RuSo2009].

5.1.2 Configuração de requisitos (L2)

Uma configuração de requisitos resume um conjunto consistente de requisitos ou artefatos de requisitos logicamente relacionados [IREB Glossary], em que cada requisito e cada artefato está disponível em, no máximo, uma versão na configuração. De acordo com [POHL2010], uma configuração tem as seguintes propriedades:

- **Consistência** – os requisitos combinados e os documentos de requisitos são consistentes e pertencem logicamente uns aos outros.¹
- **Exclusividade** – uma configuração tem um identificador que a identifica de forma exclusiva.
- **Inalterabilidade** – as alterações nos requisitos individuais ou nos documentos de requisitos de uma configuração levam a uma nova versão da configuração.

5.1.3 Baseline dos requisitos (L1)

Uma baseline de requisitos é uma configuração consolidada de requisitos que é estável em termos de conteúdo [IREB Glossary], [POHL2010].

Portanto, a baseline deve conter apenas os requisitos planejados para uma versão específica do produto (p. ex., lançamento) e não aqueles que são propostos ou ainda estão em andamento [WiBe2013].

Isso cria uma base inequívoca de comunicação para todos os participantes do projeto. O controle de versão permite vincular o status atual dos requisitos selecionados a uma baseline e, posteriormente, reconstruí-la sem ambiguidade.

De acordo com [RuSo2009] e [POHL2010], as linhas de base de requisitos dão suporte a três atividades essenciais no processo de desenvolvimento:

¹ Na prática, muitas vezes são criadas configurações que não são consistentes em termos de conteúdo. Essas configurações são criadas a partir da necessidade de congelar o status atual do trabalho para poder acessá-lo posteriormente, se necessário. Por exemplo, pode ser criada uma configuração que documente o ponto de partida das atividades de revisão.

- Eles fornecem a base **para o planejamento de liberações**. Como configurações de requisitos estáveis do produto que são visíveis para o cliente, elas servem como base para discussão ao definir uma versão.
- Eles são usados para **estimar os custos de implementação** de uma determinada versão.
- Eles permitem uma **comparação com os produtos concorrentes** no mercado.

Um momento adequado para criar uma baseline de requisitos pode ser a conquista de um marco, a conclusão de uma especificação de subsistema ou uma versão do sistema [RuSo2009].

5.1.4 Ramificação de requisitos (L2)

Uma ramificação de requisitos descreve um conjunto de requisitos que foram copiados da configuração de requisitos atual em um determinado momento e alterados independentemente do original desde aquele momento. Ao contrário das versões, os requisitos são válidos em diferentes filiais ao mesmo tempo [RuSo2009]. As ramificações de requisitos são criadas, por exemplo, se for necessário criar variantes específicas do cliente de uma especificação de requisitos para um produto.

O Gerenciamento de Requisitos é complicado e causa mais danos do que benefícios quando feito de maneira indisciplinada [RuSo2009]. Em particular, ocorrem os seguintes problemas:

- A identificação clara dos requisitos se tornará mais difícil.
- Além das versões e melhorias, a ramificação forma uma terceira dimensão do desenvolvimento de requisitos e, portanto, aumenta a complexidade do Gerenciamento de Requisitos.
- As ramificações geram informações de requisitos redundantes, que devem ser mantidas em paralelo e, em seguida, mescladas novamente a longo prazo.

5.2 Gerenciamento de mudanças para requisitos (L2)

5.2.1 Causas, fontes e cronograma de alterações de requisitos (L2)

Os requisitos de um sistema (de software) estão sujeitos a alterações durante o ciclo de vida desse sistema (de software). Essas mudanças podem ser necessárias por vários motivos:

- Necessidades novas ou alteradas dos Stakeholders [POHL2010], capítulo 37.1, [VANL2009], página 222
- Mudanças no contexto do sistema (p. ex., mudanças legislativas)[POHL2010], capítulo 37.1, [VANL2009], página 222
- Defeitos nos requisitos existentes
- Impacto da alteração de um requisito em outros requisitos dependentes
- Decisões de arquitetura e implementação que têm repercussões nos requisitos

As alterações nos requisitos podem ocorrer durante todo o desenvolvimento e a vida operacional de um sistema.

A complexidade resultante dos diferentes momentos em que as mudanças podem ocorrer exige um processo bem definido e métodos e ferramentas adequados.

5.2.2 Tipos de alterações nos requisitos (L2)

As alterações nos requisitos são classificadas da seguinte forma:

- Integração de um novo requisito
- Exclusão de um requisito existente
- Modificação de um requisito

As alterações incluem não apenas a descrição direta do requisito, mas também outros atributos (p. ex., avaliação da estabilidade) ou relacionamentos com outros artefatos (p. ex., casos de uso ou outros requisitos).

5.2.3 Análise e documentação da estabilidade dos requisitos (L2)

Os requisitos podem ser classificados com relação à sua estabilidade e, portanto, com relação à probabilidade de a versão atual do requisito ser alterada. Os requisitos instáveis serão mais afetados pelas mudanças do que os requisitos estáveis. Essa classificação pode ser realizada pela avaliação de um atributo fornecido para essa finalidade (consulte 3). Essa avaliação é usada, por exemplo, para avaliar o risco de implementar um requisito específico. Uma avaliação específica também pode expressar a estimativa do esforço ainda necessário para criar uma documentação estável do requisito. Além disso, no caso de grandes mudanças em um sistema, a avaliação também pode ser usada para considerar separadamente os requisitos que são fundamentalmente voláteis, pois se baseiam em decisões entre várias alternativas.

As regras heurísticas a seguir servem para detectar prováveis mudanças ou requisitos instáveis e manter o foco neles (consulte também [VANL2009], página 224).

- Os grupos de requisitos que atendem ao mesmo objetivo e que, em geral, são altamente estáveis (medidos pela frequência das alterações) têm menor probabilidade de alteração do que os requisitos individuais.
- As metas e os aspectos conceituais são mais estáveis do que os requisitos orientados para a solução.
- Os requisitos funcionais que atendem aos objetivos principais são mais estáveis do que os requisitos de qualidade.
- Os requisitos funcionais que aparecem repetidamente no conjunto de requisitos (como amálgamas, extensões ou variantes) são geralmente considerados como requisitos estáveis.

- Os requisitos que descrevem escolhas alternativas devem ser tratados com especial cautela e, em geral, são menos estáveis do que os anteriores, pois as decisões geralmente se baseiam em conhecimentos e suposições incompletos.
- Os requisitos atribuídos a uma variante ou aprimoramento do sistema são mais estáveis do que os requisitos que ainda não foram atribuídos.
- É improvável que os requisitos que foram alterados com frequência sejam estáveis.
- Os requisitos que pertencem a um grupo de requisitos mais estáveis são mais estáveis do que aqueles que pertencem a um grupo de requisitos mais voláteis.

5.3 Processo de gerenciamento de mudanças (L3)

O objetivo do gerenciamento de mudanças é verificar previamente cada mudança em um ou mais requisitos para controlar o risco associado e poder rastrear cada mudança posteriormente. O processo de gerenciamento de mudanças consegue isso definindo atividades, responsabilidades e artefatos necessários que descrevem um procedimento claro para lidar com solicitações de mudanças de requisitos.

O Conselho de Controle de Mudanças (CCB) desempenha um papel central no processo de Gerenciamento de Mudanças. Para poder realizar sua tarefa, ou seja, a avaliação das mudanças, da melhor maneira possível, faz sentido preencher o conselho com funções de várias áreas, por exemplo, [PoRu2011], capítulo 8.5:

- Cliente
- Arquiteto
- Engenheiro de Requisitos
- Desenvolvedor
- Testador

É essencial que pelo menos um representante de cada lado – o cliente e o fornecedor – esteja envolvido.

As principais atividades do processo de Gerenciamento de Mudanças podem ser descritas da seguinte forma. A entrada para um processo de gerenciamento de mudanças é uma solicitação de mudança, que é descrita em um formulário predefinido (modelo) para o processo.

Etapa 1: Preparação da solicitação de mudança

Etapa 2: Avaliação formal da solicitação de mudança

Etapa 3: Ao classificar uma solicitação de mudança, é determinado se ela é uma mudança corretiva, adaptativa ou excepcional. O engenheiro de requisitos está envolvido na avaliação para determinar, por exemplo, a causa de uma mudança. Essa avaliação é importante, por exemplo, para determinar se uma alteração foi acionada pelo fornecedor do sistema (software) ou pelo cliente.

Etapa 4: O objetivo da análise de impacto é estimar e documentar as consequências das mudanças. Essas consequências devem ser avaliadas não apenas para outros requisitos,

mas também para outros artefatos (arquitetura, código-fonte, casos de teste, materiais de treinamento).

Etapa 5: Os resultados da análise de impacto são usados pelo Conselho de Controle de Mudanças para determinar se a solicitação de mudança deve ser aprovada ou rejeitada. Nem sempre é razoável aceitar e implementar uma solicitação de mudança. Os motivos para uma possível rejeição de uma solicitação de mudança são, por exemplo:

- A mudança é muito cara e não se justifica em relação ao esforço necessário para sua implementação ou ao benefício esperado.
- A alteração desejada contradiz outros requisitos.
- A implementação da alteração levaria a um risco muito alto com relação à estabilidade do sistema (de software) em questão.
- A mudança não está coberta por um contrato.

Por motivos de rastreabilidade e de obtenção de acordo entre os Stakeholders envolvidas, é essencial documentar as decisões de um Conselho de Controle de Mudanças.

Etapa 6: As solicitações de mudança aceitas são priorizadas pelo Conselho de Controle de Mudanças.

Etapa 7: As solicitações de mudança aceitas são planejadas para implementação e implementadas.

6 Rastreabilidade de requisitos (L3)

Duração: 2 horas e 30 minutos

Termos: artefato, artefato de requisitos, rastreabilidade, modelo de rastreabilidade

Objetivos Educacionais

- OE 6.1 Conhecer as razões para a rastreabilidade dos requisitos (L1)
- OE 6.2 Conhecer várias visões de rastreabilidade (L2)
- OE 6.3 Conhecer os tipos de relacionamento para relacionamentos de rastreabilidade (L1)
- OE 6.4 Conhecer as formas de apresentação das relações de rastreabilidade (L3)
- OE 6.5 Dominar e usar uma estratégia de rastreabilidade específica (L3)
- OE 6.6 Dominar e usar um modelo de rastreabilidade específico (L3)
- OE 6.7 Conhecer medidas para avaliar a rastreabilidade implementada (L2)
- OE 6.8 Conhecer os desafios do rastreamento de artefatos não textuais (L1)

6.1 Razões para a rastreabilidade dos requisitos (L1)

A rastreabilidade dos requisitos é essencial para o Gerenciamento de Requisitos. No contexto do Gerenciamento de Requisitos, a implementação da rastreabilidade refere-se basicamente à manutenção de relacionamentos entre diferentes requisitos e outros artefatos de desenvolvimento ou de garantia de qualidade.

A manutenção orientada por objetivos das relações de rastreabilidade permite que as dependências existentes entre os artefatos sejam conhecidas, por exemplo, para comprovar a implementação de requisitos ou para identificar quais alterações resultam da adaptação de um determinado requisito.

Nas seções subsequentes desta unidade de aprendizagem, recapitularemos o conceito de rastreabilidade antes de explicarmos seus benefícios.

Na primeira etapa, são explicados os diferentes termos de rastreabilidade de requisitos. Na literatura, há diferentes termos que significam essencialmente a mesma coisa: rastreamento, verificabilidade, rastreabilidade, rastreabilidade de requisitos etc. Nesta unidade de aprendizagem, usaremos o termo **rastreabilidade**, a menos que façamos referência a um termo específico na literatura.

6.1.1 O que significa rastreabilidade de requisitos? (L1)

Por rastreabilidade de requisitos, entendemos a capacidade de rastrear dependências de requisitos para outros artefatos ao longo de seu ciclo de desenvolvimento ou de vida. As informações a serem documentadas para rastreabilidade são definidas pelo objetivo a ser alcançado por meio da rastreabilidade.

Por exemplo, se a rastreabilidade for usada para garantir que todos os requisitos de negócios em um projeto sejam cobertos pelos requisitos do sistema ou, inversamente, que um requisito do sistema atenda a pelo menos um requisito de negócios, uma simples referência bidirecional entre esses artefatos pode ser suficiente.

6.1.2 Por que a rastreabilidade de requisitos e artefatos é importante (L1)

A rastreabilidade dos requisitos e de outros artefatos geralmente não é uma meta do projeto, mas um meio para atingir um objetivo, por exemplo, para provar se e como um requisito foi implementado e testado. Há várias razões que motivam a rastreabilidade entre artefatos na literatura, como [HuJD2011], [PoRu2011], [WiBe2013]:

- Demonstração de como as metas e os requisitos devem ser alcançados
- Verificabilidade quanto ao motivo, se e como um requisito foi implementado
- Identificação de requisitos e propriedades desnecessários do sistema (soluções banhadas a ouro)
- Identificação de artefatos ausentes (p. ex., casos de teste ausentes)
- Simplificação da alocação do custo de desenvolvimento aos requisitos
- Suporte para reutilização de artefatos
- Suporte para manutenção, administração e desenvolvimento adicional de sistemas

A rastreabilidade dos requisitos também ajuda a responder a perguntas importantes, como o impacto de uma mudança ou o motivo da existência de um requisito. Em particular, as análises a seguir são significativamente simplificadas pela presença de relações de rastreabilidade cf. [HuJD2011], página 11 e seguintes, [EBER2012], página 305 e seguintes, [PMI2021]:

- **Análise de impacto:** Análise de quais artefatos são afetados por uma mudança (redução ou extensão do escopo) (consulte 5.3 Change Management Process)
- **Análise de origem:** Análise do motivo pelo qual um determinado artefato (p. ex., requisito) existe, a fim de identificar e evitar requisitos desnecessários, por exemplo
- **Análise de cobertura:** Análise para verificar se todos os requisitos e artefatos de desenvolvimento subsequentes foram considerados, de modo que o produto desejado possa ser completamente delimitado, desenvolvido e testado
- **Análise de valor agregado:** Análise para determinar o progresso do trabalho (valor de desempenho), a fim de compará-lo com o plano original do projeto e, se necessário, tomar as medidas adequadas

Além disso, a rastreabilidade entre os requisitos e outros artefatos é necessária para atender a determinados níveis de maturidade dos modelos de referência (p. ex., CMMI) ou restrições legais (p. ex., ISO 12207).

6.2 Diferentes visualizações de rastreabilidade (L2)

A rastreabilidade de requisitos pode ser distinguida essencialmente pelas seguintes dimensões:

- **Rastreabilidade entre os requisitos no mesmo nível de detalhe.** Esse tipo de rastreabilidade descreve, por exemplo, as dependências relacionadas ao conteúdo entre os requisitos funcionais.

- **Rastreabilidade entre os requisitos em diferentes níveis de detalhe.** Esse tipo de rastreabilidade descreve, por exemplo, o detalhamento dos requisitos legais em requisitos do sistema.
- **Rastreabilidade entre versões de requisitos:** Esse tipo descreve a rastreabilidade da evolução de um requisito ao longo do tempo. Uma particularidade dessa visão é que há apenas uma versão válida em um determinado momento.
- **Rastreabilidade entre os requisitos e os artefatos de desenvolvimento downstream.** Esse tipo de rastreabilidade descreve, por exemplo, dependências que documentam a implementação/realização de um requisito como um componente do sistema ou caso de teste.
- **Rastreabilidade entre requisitos e artefatos upstream.** Esse tipo de rastreabilidade descreve a justificativa ou a fonte de um requisito.

As duas últimas dimensões podem ser encontradas com frequência na literatura especializada sob o termo **Pre- and Post-Requirements Specification Traceability** [GoFi1994] ou **Pre- and Post-RS Traceability** [POHL2010], [PoRu2011].

Pré e pós-requisitos Rastreabilidade da especificação

- **Rastreabilidade da especificação de pré-requisitos** é a rastreabilidade dos requisitos até sua origem, por exemplo, até as metas e visões anteriores ou outras fontes de requisitos do contexto do sistema, como documentos existentes e Stakeholders.
- **A rastreabilidade da especificação pós-requisitos** é a rastreabilidade dos requisitos para artefatos de desenvolvimento subsequentes, como o projeto arquitetônico, a implementação e os casos de teste.

Rastreabilidade avançada de especificações pré e pós-requisitos

Além da diferenciação entre a rastreabilidade das especificações pré e pós-requisitos, a rastreabilidade entre os artefatos de requisitos também é considerada. Isso inclui refinamentos e rastreabilidade de funcionalidades dependentes, requisitos de qualidade etc.

Figura 6 ilustra graficamente o conceito de rastreabilidade estendida da especificação pré e pós-requisito descrita acima a partir da perspectiva de requisitos para artefatos upstream e downstream, bem como a rastreabilidade entre requisitos, tanto no mesmo nível de abstração quanto entre níveis de abstração diferentes.

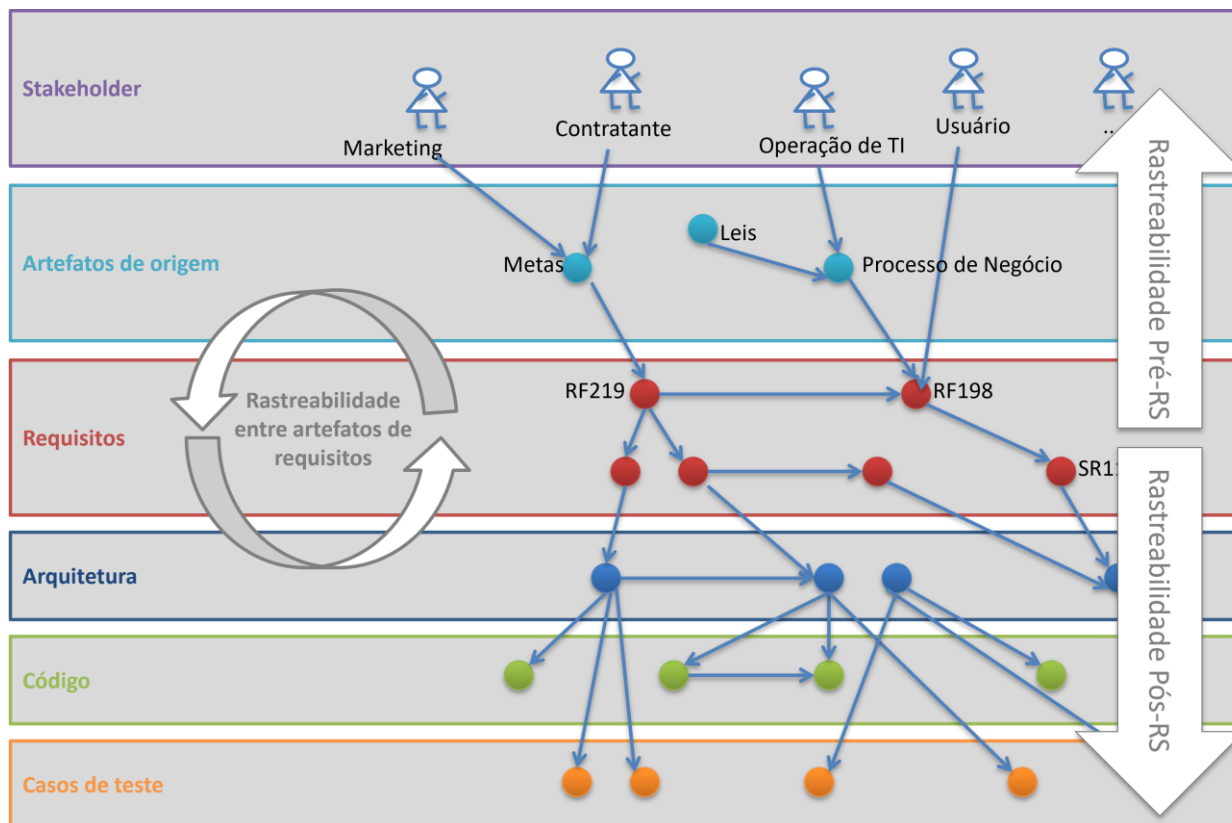


Figura 6: Diferenciação das visualizações de rastreabilidade

É importante perceber que a rastreabilidade não começa nem termina com os requisitos, mas que a rastreabilidade dos requisitos deve ser "holística", ou seja, em todos os níveis de abstração e fases – portanto, desde a fonte (rastreabilidade pré-RS) até a implementação ou aceite (rastreabilidade pós-RS).

A granularidade da rastreabilidade pode estar no nível de requisitos individuais, mas também no nível de grupos de requisitos ou artefatos de requisitos.

6.3 Tipos de relacionamento para relacionamentos de rastreabilidade (L1)

Não há uma definição geralmente aceita dos tipos de relações de rastreabilidade. É essencial que as relações de rastreabilidade sejam usadas de acordo com o objetivo da documentação e que haja um acordo na equipe sobre quais tipos de relações são usados, o que significam e entre quais artefatos são usados.

Teoricamente, os relacionamentos de rastreabilidade poderiam ser representados por um único tipo de relacionamento (p. ex., relacionado a). Entretanto, essa relação não diz nada sobre como os artefatos se relacionam entre si. Um artefato pode ser um detalhamento, uma realização, um caso de teste, uma variante ou até mesmo uma contradição de outro artefato.

Portanto, é possível encontrar na literatura diferentes tipos de relações de rastreabilidade que podem ser usadas para documentação específica de rastreabilidade.

Pohl divide as relações de rastreabilidade em diferentes classes [POHL2010], às quais podem ser atribuídos diferentes tipos de relações:

- **Condição:** A classe *condição* contém relações de rastreabilidade que descrevem dependências relacionadas ao conteúdo entre dois artefatos (restrição, pré-condição etc.).

Exemplo: **Requisito 1** (is_condition_for) **Requisito 12**

- **Conteúdo:** O *conteúdo* da classe contém relações de rastreabilidade que comparam o conteúdo de dois artefatos (igualdade, contradição, conflito etc.).

Exemplo: **Requisito 6** (stands_in_contradiction_to) **Requisito 10**

- **Documentação:** A *documentação* da classe contém relações de rastreabilidade que fornecem mais informações sobre um artefato (motivo, exemplo, comentário, caso de teste etc.).

Exemplo: **Artefato de teste 99** (is_test_case_for) **Requisito 3**

- **Abstração:** A *abstração* de classe inclui relações de rastreabilidade que descrevem as relações de abstração entre dois artefatos (classificação, agregação, generalização etc.).

Exemplo: **Requisito 43** (generaliza) **Requisito 84**

- **Evolução:** A *evolução da* classe inclui relações de rastreabilidade que descrevem a maneira pela qual um requisito é desenvolvido (atendido, refinado, substituído, ampliado etc.).

Exemplo: **Requisito 73, versão 1.2** (substitui) **o requisito 73, versão 1.1**

Não é possível dar uma resposta geral sobre quais tipos/classes de relacionamento são relevantes para um projeto específico. É importante pensar sobre o objetivo e as relações de rastreabilidade a serem usadas antes de iniciar a documentação (como política da empresa ou específica do projeto) e defini-las para todos os participantes (consulte também [MaGP2009], [MJZC2013]).

Exemplo: Para garantir que todos os requisitos dentro do escopo de um projeto sejam justificados e que cada requisito seja testado, os seguintes tipos de relações de rastreabilidade podem ser úteis:

- Relacionamento do tipo "cumprido" para garantir que não haja nenhum requisito que não possa ser atribuído a nenhuma meta de negócios
- Relacionamento do tipo "testado" para garantir que exista um caso de teste para cada requisito
- Relacionamento do tipo "justificado" para garantir que as decisões sobre alterações de requisitos tenham sido documentadas

Para garantir que as relações de rastreabilidade sejam definidas e usadas conscientemente, os artefatos relevantes e os tipos de relacionamento entre esses artefatos devem ser documentados em um modelo de rastreabilidade (consulte 6.6). Ver também 2.3.

6.4 Formas de apresentação das relações de rastreabilidade (L3)

6.4.1 Documentação implícita e explícita de rastreabilidade (L3)

Uma relação de rastreabilidade pode ser documentada implícita ou explicitamente. Embora esta unidade de aprendizagem se concentre principalmente na documentação explícita da rastreabilidade, esta seção abordará essa distinção.

Documentação explícita da rastreabilidade: A rastreabilidade explícita é obtida por meio de relações definidas e deliberadamente estabelecidas entre os artefatos (consulte o capítulo 6.4.3).

Documentação implícita de rastreabilidade: A rastreabilidade implícita pode ser obtida, por exemplo, por meio de convenções de nomenclatura ou estrutura de documentação.

6.4.2 Relações de rastreabilidade bidirecionais e unidirecionais (L2)

As relações de rastreabilidade podem ser documentadas como unidirecionais (direcionadas) ou bidirecionais (não direcionadas), dependendo do objetivo a ser alcançado com a rastreabilidade.

- **Rastreabilidade unidirecional:** permite a rastreabilidade de um artefato para outro, mas não vice-versa. Por exemplo, a referência de um requisito de teste a um requisito do sistema permite verificar por que o requisito de teste existe ou do que ele depende. No entanto, o requisito do sistema não conseguirá encontrar uma referência exclusiva para um requisito de teste. Esse tipo de relacionamento é frequentemente encontrado em técnicas baseadas em documentos, em que os relacionamentos são mantidos manualmente, por exemplo, por referências textuais, e se referem ao artefato predecessor ou sucessor. Na direção da documentação, é

importante observar que é feita referência ao artefato do qual existe uma dependência.

- **Rastreabilidade bidirecional:** permite a rastreabilidade de um artefato para outro e vice-versa. Diferentemente da relação unidirecional, é possível navegar entre os artefatos, por exemplo, de um requisito para um caso de teste (p. ex., por meio de uma referência textual a um caso de teste) e de um caso de teste para o requisito correspondente que deve ser verificado com esse caso de teste. Esse tipo de relacionamento permite que você observe os artefatos predecessores e sucessores (rastreabilidade pré e pós-especificação de requisitos). Nas ferramentas de Gerenciamento de Requisitos, essas relações geralmente são geradas automaticamente, de modo que a ferramenta ofereça suporte à navegação ou à análise de impacto em ambas as direções. Para referências puramente textuais, no entanto, é necessária a manutenção explícita de cada artefato envolvido.

6.4.3 Formas de apresentação das relações de rastreabilidade (L3)

Para a documentação explícita da rastreabilidade, podem ser selecionadas diferentes formas de representação. Ver [POHL2010], [PoRu2011]:

- **Referências baseadas em texto:** A documentação e a representação por referências textuais é a maneira mais fácil de implementar relações de rastreabilidade entre artefatos (consulte Figura 7). A relação descreve o tipo de relação e uma ID exclusiva do artefato ao qual a relação se refere (p. ex., [TEST_CASE_FOR → ReqID 1189]). Esse tipo de apresentação tem a vantagem de poder ser usado independentemente de uma ferramenta de Gerenciamento de Requisitos e de ser fácil de entender. Normalmente, ele é documentado diretamente em um artefato, por exemplo, em um caso de teste há uma referência a um requisito.

Caso de teste	Referência ao requisito	Descrição do Caso de Teste	Prioridade
CT_0021	RF_0012 RF_0013 RF_0016	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea ...	Alta
...			
CT_0150	RF_0020	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed ...	Alta

Figura 7: Referências baseadas em texto

- **Hiperlinks:** Ao contrário das referências baseadas em texto, os hiperlinks permitem a navegação direta para o artefato de destino. Os hiperlinks sempre serão criados do artefato de origem para o artefato de destino. As relações bidirecionais podem ser

criadas por meio de referências cruzadas. Em comparação com as referências textuais simples, o uso de hiperlinks tem a vantagem de poder "pular" diretamente para os artefatos referenciados. No entanto, isso geralmente só é possível em uma ferramenta.

- **Matrizes de rastreabilidade:** Em uma matriz de rastreabilidade, as relações de rastreabilidade são representadas por referências nas células de uma matriz (consulte Figura 8). A matriz resultante documenta a relação entre o artefato de origem e o artefato de destino. Esse tipo de apresentação permite uma apresentação abstrata das dependências entre artefatos, por exemplo, BR_0010 é detalhado por UC_10; BR_0020 é detalhado por UC_30 e UC_40; UC_40 detalha (direção de leitura inversa) BR_0020 e BR_0030.
- As matrizes de rastreabilidade podem ser usadas para representar exatamente um tipo de relacionamento entre dois artefatos. Entretanto, a apresentação de diferentes tipos de relacionamento também é possível. Ferramentas de Gerenciamento de Requisitos, como o DOORS, criam automaticamente essas apresentações com base em artefatos e relacionamentos, fornecendo uma visão geral dos relacionamentos entre artefatos. Na prática, porém, essas matrizes rapidamente se tornam muito grandes e difíceis de entender.

		Artefato de destino			
		UC_10	UC_20	UC_30	UC_40
Artefato de origem	RN_0010	é detalhado por			
	RN_0011		é detalhado por		
	RN_0020			é detalhado por	é detalhado por
	RN_0030				é detalhado por

Figura 8: Matriz de rastreabilidade [BR = requisito comercial, UC = caso de uso]

- **Tabelas de rastreabilidade:** Diferentemente das matrizes de rastreabilidade, as tabelas de rastreabilidade permitem descrever as relações de rastreabilidade entre todos os artefatos em diferentes níveis de detalhe (consulte Figura 9). Assim, eles oferecem uma ferramenta poderosa para documentar a rastreabilidade dos objetivos, por meio de casos de uso e requisitos funcionais, até os casos de teste. Essa ferramenta pode ser usada independentemente de uma ferramenta especializada de Gerenciamento de Requisitos para documentar a rastreabilidade entre os próprios artefatos documentados em diferentes ferramentas (Word, Excel, Rational Rose, Visual Paradigm, Quality Center etc.).

Requisito de negócio	Caso de Uso	Requisito funcional	Requisito de sistema	Elemento da GUI	Caso de teste
RN_0010	UC_10	RF_0012 RF_0013 RF_0016	CRM_0011 CRM_0020 DWH_0010 Faturamento_0020	GUI_0081	CT_0021 CT_0022 CT_0025
RN_0011	UC_20	RF_0020	CRM_0011 CRM_0020		CT_0060 CT_0150

Figura 9: Tabela de rastreabilidade (BR = requisito de negócios, UC = caso de uso, FR = requisito funcional, CRM = gerenciamento de relacionamento com o cliente, DWH = data warehouse, GUI = interface gráfica do usuário, TC = caso de teste)

- **Gráficos de rastreabilidade:** Em um gráfico de rastreabilidade, os nós representam artefatos e as bordas representam as relações entre os artefatos (consulte Figura 10). Para poder distinguir entre os diferentes artefatos de desenvolvimento (p. ex., cenário, requisito, caso de teste) e relacionamentos (p. ex., refina, implementa, testcase_for) em um piscar de olhos, recomenda-se definir uma forma de notação apropriada.

No entanto, seu uso só é recomendado se esses gráficos puderem ser criados automaticamente com base nos artefatos e nos relacionamentos. A reprodução manual de tais gráficos e sua manutenção serão muito complexas na prática. Em princípio, esses gráficos oferecem uma maneira fácil de entender a verificação de dependências e a navegação entre os diferentes artefatos.

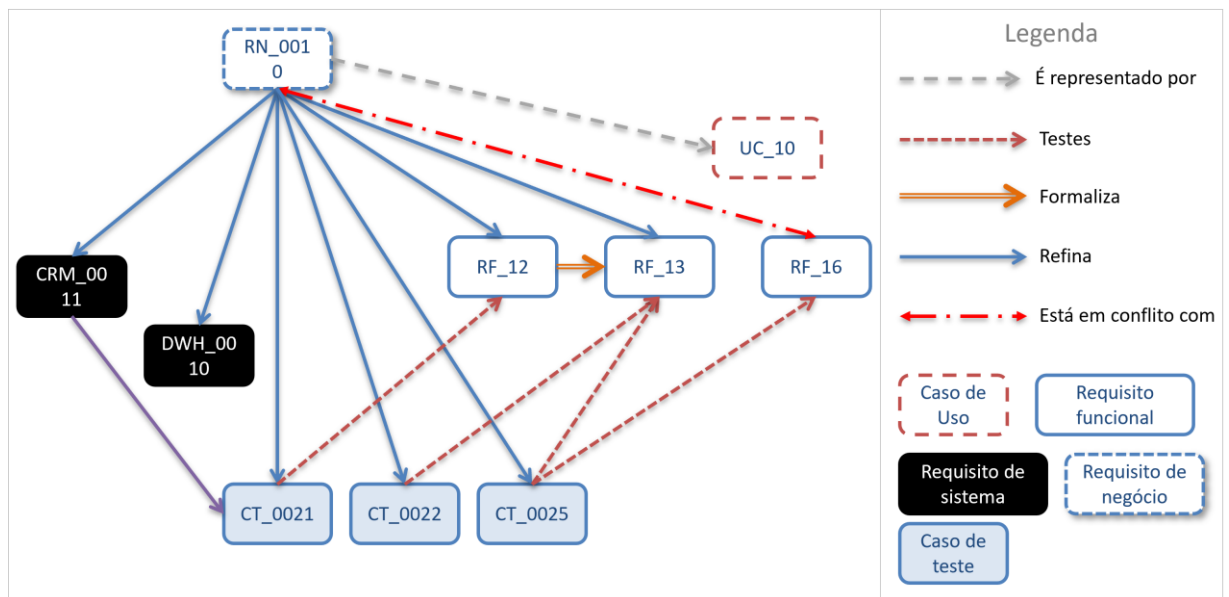


Figura 10: Gráfico de rastreabilidade

A Tabela 1 resume e avalia as formas descritas de relações de rastreabilidade.

Forma de apresentação	Positivo	Negativo	Adequado para
Documentação de rastreabilidade em linha			
Referências baseadas em texto	<p>Pode ser implementado independentemente das ferramentas e de forma abrangente</p> <p>O relacionamento é visível no artefato como texto simples.</p>	<p>As análises de rastreabilidade são muito complexas.</p>	<p>Para representar a rastreabilidade em especificações textuais baseadas em papel.</p>
Hiperlinks	<p>O relacionamento é visível no artefato como texto simples.</p> <p>Fácil navegação entre artefatos para detectar dependências diretas.</p>	<p>A rastreabilidade entre diferentes ferramentas não é prontamente possível.</p>	<p>Para representar a rastreabilidade em especificações eletrônicas.</p>
Documentação ortogonal de rastreabilidade			
Matrizes de rastreabilidade	<p>A dependência entre dois artefatos é rápida e facilmente visível.</p>	<p>A criação manual de matrizes de rastreabilidade consome muito tempo e resulta em matrizes grandes, mas mal preenchidas.</p>	<p>Representação de um único tipo de relacionamento entre dois tipos de artefatos específicos (p. ex., casos de uso e requisitos)</p>
Tabelas de rastreabilidade	<p>Permite a exibição clara da rastreabilidade estendida pré e pós-RS.</p> <p>Permite uma variedade de análises de rastreabilidade.</p>	<p>Alta complexidade de criação.</p>	<p>Representação da rastreabilidade entre artefatos baseados em texto e em modelos em diferentes documentos/ferramentas</p>
Gráficos de rastreabilidade	<p>Apresentação gráfica da rastreabilidade; permite a apresentação "abstrata" das relações de rastreabilidade entre artefatos.</p>	<p>O uso só é possível com o suporte adequado da ferramenta.</p>	<p>Representação de rastreabilidade complexa entre artefatos em uma ferramenta de Gerenciamento de Requisitos.</p>

Tabela 1: Formas de apresentação das relações de rastreabilidade

6.5 Desenvolvimento de uma estratégia para rastreabilidade específica do projeto (L3)

Como já mencionado, o estabelecimento e o uso da rastreabilidade em um projeto devem ser planejados. Normalmente, não é apropriado documentar todos os relacionamentos entre artefatos. Em vez disso, no início do projeto, deve-se pensar no motivo pelo qual a rastreabilidade é necessária nesse projeto e em quais pontos o tipo de rastreabilidade será necessário.

Além de definir os artefatos relevantes e os tipos de relacionamento, também é necessário que uma estratégia de rastreabilidade específica defina um sistema para registrar e usar essas informações.

Uma estratégia específica de rastreabilidade considera os seguintes aspectos:

- **Objetivo da rastreabilidade:** determinar por qual motivo (ou seja, por que) a rastreabilidade é necessária no contexto do respectivo projeto ou o que se pretende alcançar (ou seja, para quê) (cf. 6.1.2).
- **Estratégia de uso:** Definição de estratégias para o uso de informações de rastreabilidade pela equipe de desenvolvimento. Por exemplo, uma estratégia de uso poderia ser a análise do impacto da mudança, em que as relações de rastreabilidade são usadas para determinar quais artefatos são afetados por uma mudança.
- **Estratégia de registro:** Definição de estratégias para o registro de informações de rastreabilidade pela equipe. A responsabilidade pela documentação das relações de rastreabilidade deve ser explicitamente atribuída para que possa ser executada. A responsabilidade deve ser definida para cada tipo de relacionamento (p. ex., caso de uso para requisito funcional pelo analista de negócios; requisito funcional para caso de teste pelo gerente de qualidade). Por exemplo, uma das estratégias de registro poderia ser a documentação cronológica das relações de rastreabilidade propostas por [HuJD2011] ou [WiBe2013]. A relação entre dois artefatos é criada assim que o novo artefato (p. ex., refinamento de requisitos ou caso de teste) é criado. A vantagem disso é que há uma responsabilidade clara pela definição das relações de rastreabilidade.
- **Modelo de rastreabilidade específico do projeto:** Definição das informações de rastreabilidade a serem registradas e a forma de apresentação. Um modelo de rastreabilidade descreve quais tipos de relacionamentos (p. ex., `is_refined_by`; `is_tested_by`) entre quais artefatos (p. ex., como uma referência baseada em texto do outro artefato em ambos os artefatos) devem ser documentados (consulte 6.6).

Ao definir uma estratégia de rastreabilidade, deve-se prestar atenção ao seguinte:

- Toda a equipe está ciente da necessidade de rastreamento de requisitos,
- o modelo de rastreabilidade seja compreensível e aceito por todas as partes envolvidas,
- os membros da equipe conheçam e aceitem as responsabilidades atribuídas a eles para a documentação das relações de rastreabilidade,

- os pré-requisitos necessários são estabelecidos para documentar corretamente a rastreabilidade.

6.6 Criação e uso de modelos específicos de rastreabilidade (L3)

Para criar um modelo de rastreabilidade específico do projeto, é preciso primeiro considerar entre quais artefatos a rastreabilidade deve ser estabelecida e quais relações de rastreabilidade entre esses artefatos são necessárias ou permitidas. Essas especificações devem ser descritas por um modelo de rastreabilidade específico do projeto (cf. [MJZC2013] e [POHL1996]) e comunicadas dentro do projeto.

Em um modelo de rastreabilidade específico, todos os participantes do projeto podem ver claramente quais artefatos existem, quais tipos de relacionamento devem ser mantidos e quem deve mantê-los e como (cf. [POHL1996], [POHL2010], [MaGP2009]).

6.6.1 Um processo para definir um modelo de rastreabilidade específico

A seguir, descrevemos um exemplo de processo para definir um modelo de rastreabilidade específico.

1. Seleção de um esquema de referência
A primeira etapa deve ser verificar se um modelo de rastreabilidade existente pode ser reutilizado e adaptado. Uma maneira eficaz de definir um modelo de rastreabilidade específico é reutilizar um modelo de rastreabilidade existente de um projeto semelhante ou um modelo de rastreabilidade de toda a empresa. Esse modelo de rastreabilidade pode servir de base para a definição do modelo de rastreabilidade específico e geralmente já contém muitos artefatos e dependências a serem definidos.
2. Seleção de artefatos
Nessa etapa, determina-se entre quais artefatos a rastreabilidade deve ser garantida para apoiar a meta definida na estratégia de rastreabilidade e os cenários de uso, por exemplo, a rastreabilidade entre o caso de uso e o requisito funcional e entre o requisito e o caso de teste.
3. Definição de tipos de relacionamento permitidos entre tipos de artefatos
Aqui deve ser especificado quais tipos de relacionamento são permitidos para representar a rastreabilidade (consulte 6.3) entre dois tipos de artefato, por exemplo, um relacionamento válido entre requisito e caso de teste é: "validado por".
4. Determinação do número de relações de rastreabilidade (em nível de instância)
Aqui é especificado o número mínimo de relações entre os artefatos reais (no nível de instância do modelo de rastreabilidade), por exemplo, cada requisito exige uma relação de rastreabilidade com um caso de teste.
5. Definição da dependência entre artefatos

Aqui é especificado qual artefato é dependente de outro artefato, por exemplo, um caso de teste depende do conteúdo de um requisito. Ao usar relações unidirecionais, deve-se prestar atenção à referência (consulte 6.4.2)

6.6.2 Uso de um modelo de rastreabilidade específico

Além da definição de artefatos e relações de rastreabilidade, por exemplo, documentados em um modelo de informações, outros aspectos devem ser considerados para a implementação e o uso de um modelo de rastreabilidade específico:

1. Definição da forma de apresentação
Depois de definir quais relações entre quais artefatos devem ser documentadas, é preciso esclarecer em que tipo e forma de apresentação as relações de rastreabilidade devem ser documentadas. A seleção da forma de apresentação das relações de rastreabilidade geralmente é influenciada pela forma de representação dos artefatos (consulte 6.4.3).
2. Fornecimento de suporte para registro de dados
O registro das relações de rastreabilidade entre os artefatos representa um esforço adicional (consulte **0Tabela 1**), que geralmente atende a outros Stakeholders (p. ex., gerentes de projeto). Portanto, é muito útil que a documentação das relações de rastreabilidade seja apoiada na medida do possível. Isso pode ser apoiado, por um lado, por ferramentas de Gerenciamento de Requisitos ou por soluções autoprogramadas, por exemplo, com macros do Word.
3. Alinhamento de uma ferramenta com os artefatos do projeto
Ao usar uma ferramenta de RM, geralmente é necessária uma tradução para a terminologia existente da ferramenta. Nessa etapa, os identificadores de artefatos e tipos de relacionamento definidos no modelo são vinculados aos identificadores oferecidos pela ferramenta e referenciados de forma exclusiva. Por exemplo, se a ferramenta oferecer apenas um tipo de artefato, "Requisito", mas o modelo de rastreabilidade distinguir entre "Requisito do usuário" e "Requisito do sistema", será necessário um mapeamento adequado e, se necessário, a atribuição de um atributo adicional, permitindo a diferenciação posterior.

6.7 Medidas para a avaliação da rastreabilidade implementada (L2)

A rastreabilidade inevitavelmente levanta a questão de quão bem e completamente as informações de rastreabilidade entre artefatos (requisitos, decisões, fragmentos de código, casos de teste etc.) são de fato documentadas e se a rastreabilidade implementada cumpre seu objetivo real (consulte 6.1).

A verificação das informações de rastreabilidade fornece uma visão da qualidade da documentação atual. Além disso, esses resultados também são úteis para identificar processos sem saída ou modelos de rastreabilidade "inadequados".

Os exemplos de medidas a seguir podem ajudar a verificar a integridade e a qualidade das relações de rastreabilidade:

- Razão entre o número de relações de rastreabilidade corretas e o número total de relações de rastreabilidade (correção)
- Relação entre o número de relações de rastreabilidade existentes e o número total de relações de rastreabilidade necessárias (completude)
- Relação entre o número de requisitos com relações de rastreabilidade e o número total de requisitos (densidade)

Uma proporção insuficiente de relações entre artefatos sugere que as relações não foram mantidas de forma consistente. Por outro lado, uma baixa proporção de relações corretas sugere que as relações foram mantidas de forma negligente ou que as alterações não foram aplicadas de forma consistente a todos os artefatos em questão. Qualquer desvio da meta declarada pode ter motivos diferentes que precisam ser discutidos. Os valores de limite correspondentes, para os quais as ações devem ser tomadas, devem ser definidos especificamente.

Além de reconhecer que a estratégia de rastreabilidade definida não está implementada, surge a questão de saber por que a rastreabilidade não está implementada ou está implementada incorretamente.

Os possíveis motivos para a falta ou incorreção da documentação de rastreabilidade são

- A necessidade de rastreabilidade não é reconhecida
- Falta de estratégia de rastreabilidade (quem documenta o quê e por quê)
- As restrições de tempo não permitem a documentação da rastreabilidade
- Não há um modelo de rastreabilidade acordado
- Suporte insuficiente da ferramenta ao registrar as relações de rastreabilidade

6.8 Desafios na rastreabilidade de artefatos não textuais (L1)

A rastreabilidade entre artefatos textuais (p. ex., requisitos funcionais) e artefatos baseados em modelos (p. ex., atividades em diagramas de atividades UML), ou entre os próprios artefatos baseados em modelos, só pode ser obtida com muito esforço.

Independentemente de como as relações de rastreabilidade foram criadas, o uso de informações de rastreabilidade para avaliações e análises (p. ex., que efeito a alteração de um requisito comercial tem sobre o software existente) ainda é complexo, pois geralmente não é possível abrir o artefato correspondente diretamente (como em um hiperlink). No entanto, no final, é essa manutenção que torna possível uma análise de impacto.

Uma das opções independentes de ferramentas para documentar as referências entre artefatos textuais e elementos do modelo é, por exemplo, as tabelas de rastreabilidade (consulte 6.4.3), que podem fazer referência clara aos artefatos textuais e aos elementos do modelo. Para fazer isso, no entanto, é necessário atribuir IDs exclusivos (manual ou automaticamente) aos elementos do modelo. Usando uma referência textual ou uma tabela de rastreabilidade, é possível, por exemplo, referenciar claramente os elementos do modelo, como casos de uso, aos requisitos textuais correspondentes.

Figura 11 mostra um exemplo simples de referência a requisitos textuais e casos de uso em um diagrama de casos de uso usando uma tabela de rastreabilidade.

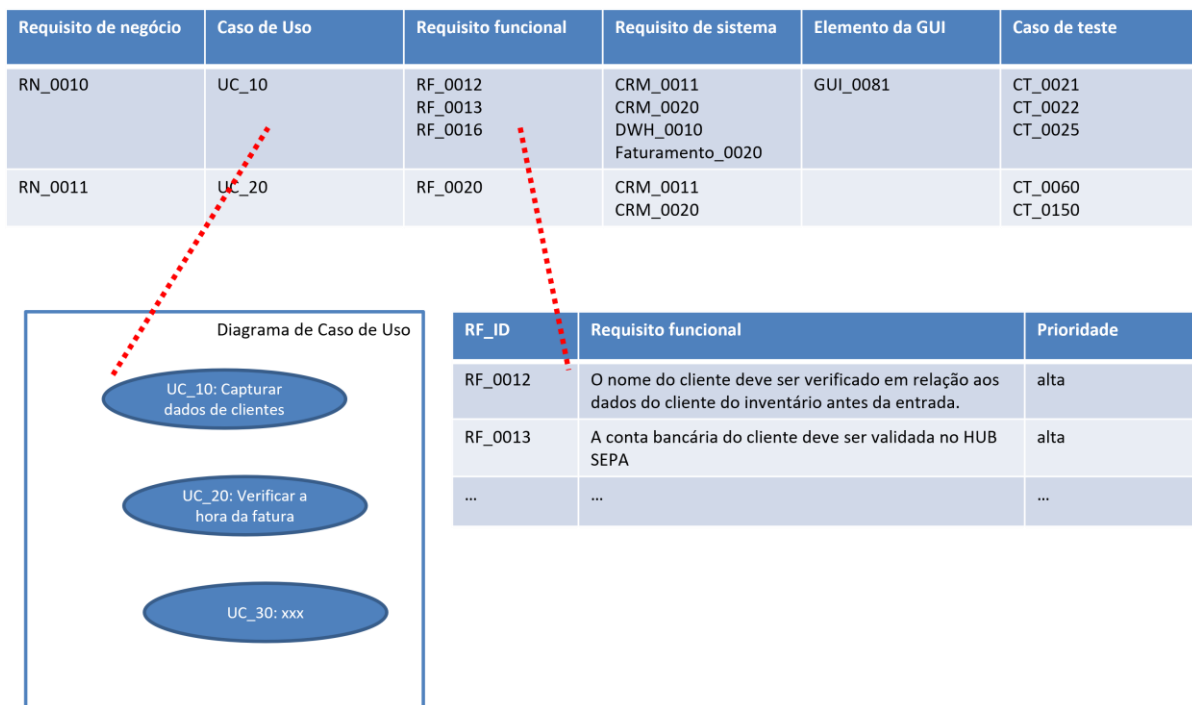


Figura 11: Rastreabilidade entre artefatos textuais e baseados em modelos

7 Gerenciamento de variantes para requisitos (L5)

Duração: 3 horas

Termos: Variabilidade, reutilização, família de produtos, linha de produtos, recurso, modelo de recurso, ponto de variação, variante, tempo de vinculação

Objetivos Educacionais

- OE 7.1.1 Conhecer as razões para usar variantes de requisitos (L1)
- OE 7.1.2 Conhecer os principais termos no campo de Gerenciamento de Variantes para Requisitos (L1)
- OE 7.1.3 Conhecer os benefícios da documentação explícita da variabilidade (L2)
- OE 7.2.1 Conhecer formas comuns de apresentação para criar variantes de requisitos (L3)
- OE 7.2.2 Avaliar uma determinada forma de apresentação de variabilidade de acordo com determinados critérios (L5)
- OE 7.2.3 Analisar uma determinada forma de representação da variabilidade com relação a uma situação operacional em um determinado contexto (L4)
- OE 7.3.1 Conhecer o conceito de modelagem de recursos (L1)
- OE 7.3.2 Dominar e usar um modelo de recursos (L3)

7.1 Uso de variantes de requisitos (L1)

Em muitos casos, o desenvolvimento de produtos envolve não apenas a criação de um único produto, mas também – em paralelo ou sequencialmente – uma série de produtos semelhantes. Nesse caso, geralmente faz sentido não executar os desenvolvimentos de produtos independentemente uns dos outros, mas reutilizar artefatos de desenvolvimento, como requisitos, arquiteturas, código de programa ou casos de teste de forma apropriada [ISO 26550].

Se produtos semelhantes forem desenvolvidos por meio da reutilização sistemática e planejada de artefatos de desenvolvimento, isso é chamado de linha de produtos ou família de produtos [CINo2007].

A definição do termo linha de produtos é a seguinte: "Uma linha de produtos de software é um conjunto de sistemas com uso intensivo de software que compartilham um conjunto comum e gerenciado de recursos que satisfazem as necessidades específicas de um determinado segmento de mercado ou missão e que são desenvolvidos a partir de um conjunto comum de ativos principais de uma maneira prescrita." (de [C1No2007]).

Os ativos principais normalmente incluem requisitos. Os requisitos de um conjunto de produtos não são gerenciados independentemente uns dos outros; em vez disso, os requisitos estão adequadamente localizados em um "pool de requisitos" comum e são atribuídos aos produtos individuais.

O conceito de linha de produtos também pode ser limitado apenas ao nível dos requisitos. Também nesse caso, os requisitos para diferentes produtos são gerenciados em um "pool de requisitos" e os requisitos para produtos especiais são derivados desse pool, mesmo que o desenvolvimento dos diferentes produtos seja feito de forma independente. Nesse caso, ele é chamado de linha de produtos de requisitos.

Em ambos os casos, um pool de requisitos é um conjunto de requisitos que contém mais do que o conjunto de requisitos para um produto específico. Um pool de requisitos também pode conter requisitos que não estão incluídos atualmente em nenhum produto.

No desenvolvimento da linha de produtos, é comum distinguir entre dois processos [WeLa1999], [ISO 26550]:

- Engenharia de domínio: na engenharia de domínio, um "produto básico" será desenvolvido
- Engenharia de aplicativos: aqui o "produto básico" será ajustado ao produto específico

Um conceito central para a reutilização de requisitos é a **variabilidade** [POHL2010], [PoBL2005]. A variabilidade permite a definição e a realização de diferentes produtos, selecionando variantes concretas de um determinado conjunto de variantes possíveis.

Cada variante sempre se refere a um ponto de variação. Os pontos de variação descrevem onde algo varia em uma linha de produtos; as variantes descrevem as especificações possíveis (permitidas) (de requisitos) em um ponto de variação. Normalmente, um documento de requisitos contém muitos pontos de variação com várias variantes. No entanto, nem toda combinação de variantes individuais é permitida. Normalmente, há dependências de variantes. As dependências de variantes podem ser opcionais ou obrigatórias. Opcional significa que uma variante pode ser selecionada, mas não precisa ser. Obrigatório significa que uma variante deve ser selecionada (p. ex., entre várias alternativas).

Ao selecionar variantes específicas para cada ponto de variação, um produto específico é "configurado". Se todas as dependências de variantes forem consideradas, o resultado será um produto válido e específico.

Na prática, nem todos os pontos de variação são atribuídos a variantes concretas durante a coleta de requisitos, ou seja, uma das diversas variantes possíveis é selecionada. Alguns pontos de variação permanecem abertos, ou seja, ainda não estão "vinculados" a uma variante específica. O tempo de ligação (cf. [CoHW1998]) pode permanecer aberto até que o sistema seja entregue ou até a operação do sistema.

Os tempos de vinculação possíveis são "antes do desenvolvimento", "durante o desenvolvimento" (tempo de implementação), "durante a montagem" (tempo de construção), "durante o comissionamento" (tempo de instalação), "durante o lançamento do sistema" (tempo de inicialização) e "durante o tempo de execução" (tempo de execução) [ATKI2002].

A variabilidade pode ser documentada implícita ou explicitamente. No caso da documentação implícita, o leitor deve reconhecer, a partir da formulação de um requisito, que diferentes variantes do produto (especificações do produto) são possíveis aqui.

Na documentação **implícita**, a palavra "ou" geralmente indica que são possíveis diferentes versões do produto. No entanto, a palavra "ou" não é um indicador confiável de um ponto de variação, pois também é usada com muita frequência em condições lógicas. Além disso, outros termos-chave também são possíveis, o que também indica diferentes especificações do produto (p. ex., "both... as well as"). Geralmente, elas também não são suficientemente claras.

A documentação **explícita** da variabilidade pode ser integrada à documentação dos requisitos ou de forma ortogonal, ou seja, em um modelo separado. No caso de requisitos textuais, tanto os pontos de variação quanto as possíveis variantes são explicitamente mostrados no texto do requisito em uma documentação integrada.

No caso da documentação ortogonal, o requisito textual permanece inalterado. A documentação dos pontos de variação e das variantes é feita em um modelo separado.

No desenvolvimento da linha de produtos em geral e no Gerenciamento de Requisitos em particular, a documentação explícita da variabilidade tem as seguintes vantagens [POHL2010]:

- **Comunicação:** A documentação explícita dos pontos de variação e das possíveis variantes apoia a comunicação com os Stakeholders afetadas, pois é fácil ver quais variantes podem ser selecionadas em quais pontos.
- **Apoio à decisão:** A documentação explícita leva a decisões mais conscientes, (1) em quais pontos a variabilidade é fornecida e (2) qual variante concreta foi selecionada para um determinado produto.
- **Rastreabilidade:** Por meio da documentação explícita das dependências de variantes, as dependências podem ser analisadas e, no caso de alterações de requisitos, usadas para analisar as alterações subsequentes.

7.2 Formas de documentação explícita de variantes e sua avaliação [L5]

Na prática, muitas formas diferentes de documentação de variantes podem ser encontradas nos documentos de requisitos. Esses formulários usam os conceitos introduzidos na seção 7.1, como ponto de variação, variante, alocação de produto e documentação de tempos de ligação de maneiras muito diferentes.

Primeiro, as formas comuns de representação [BOUT2011] serão apresentadas em forma de esboço e ilustradas com um breve exemplo do setor automotivo. Em seguida, essas formas de apresentação serão analisadas com relação aos conceitos introduzidos em 7.1.

Posteriormente, serão identificados os critérios para avaliar os pontos fortes e fracos das formas de apresentação e as formas apresentadas serão avaliadas.

Os modelos de recursos representam outra forma de representação. Eles são discutidos em mais detalhes na seção 7.3.

Formulário 1: Atribuição textual de requisitos a produtos de concreto

Nesse caso, os produtos afetados são explicitamente nomeados nos requisitos individuais (Figura 12).

Identificação	Requisito
R32	O para-sol do Classe A deve ser revestido de plástico.
R33	O para-sol do Classe E deve ser revestido de couro.
R34	A viseira solar de todos os produtos deve conter um espelho de maquiagem iluminado.

Figura 12: Atribuição textual de requisitos a produtos concretos

Forma 2: Atribuição explícita de requisitos a produtos específicos

Nesse caso, os requisitos individuais são atribuídos diretamente aos produtos (variantes de produtos) em questão (Figura 13).

Identificação	Requisito	A-Class	E-Class
R32	A viseira solar deve ser revestida de plástico.	X	
R33	A viseira solar deve ser coberta com couro.		X
R34	A viseira solar deve conter um espelho de maquiagem iluminado.	X	X

Figura 13: Atribuição explícita de requisitos a produtos específicos

Há várias subvariantes relacionadas à forma concreta de atribuição direta:

- Atribuição explícita para separar colunas de produtos
- Seleção múltipla em uma coluna de produto (consulte também a primeira coluna de produto em Figura 14)

Forma 3: Atribuição explícita de requisitos a recursos específicos do produto

Nesse caso, os requisitos individuais são atribuídos diretamente a vários recursos do produto (Figura 14). Um produto específico é definido por vários recursos do produto, que podem ter características diferentes. Um requisito pertence a um produto se o requisito for atribuído aos recursos do produto pertencentes ao produto. Por exemplo, os requisitos R33 e R34 pertencem ao produto "E-Class in the USA".

Identificação	Requisito	Série	Mercado
R32	A viseira solar deve ser revestida de plástico.	A-Class	EUA Europa
R33	A viseira solar deve ser coberta com couro.	E-Class	EUA
R34	A viseira solar deve conter um espelho de maquiagem iluminado.	A-Class E-Class	EUA Europa

Figura 14: Atribuição explícita de requisitos a recursos específicos do produto

Observação: essa forma de apresentação apresenta o risco de configurações inválidas se tornarem válidas repentinamente devido à combinatória. Exemplo: No caso do R34, o requisito deve se aplicar apenas ao Classe A EUA e ao Classe E EUA e Classe E Europa. A exclusão da Classe A da Europa não pode ser representada pela metodologia de classificação escolhida.

Forma 4: atribuição indireta de requisitos a produtos por meio de recursos

Nesse caso, os requisitos individuais são atribuídos aos recursos do produto. Uma configuração de produto separada determina quais recursos do produto estão contidos em um produto específico (ou seja, uma variante de produto específica), o que determina indiretamente os requisitos relevantes (Figura 15).

Identificação	Requisito	Característica, recurso, funcionalidade
R32	A viseira solar deve ser revestida de plástico.	Superfície plástica
R33	A viseira solar deve ser coberta com couro.	Superfície de couro
R34	A viseira solar deve conter um espelho de maquiagem iluminado.	

Produto	Recursos
A-Class	Superfície plástica e ...
E-Class	Superfície de couro e ...

Figura 15: Atribuição indireta de requisitos a recursos e configuração do produto

Os requisitos não atribuídos a nenhum recurso se aplicam a todos os produtos.

Análise das formas de apresentação

Os seguintes aspectos devem ser considerados na análise das formas de apresentação, de acordo com 7.1:

- **Descrição dos tempos de vinculação**
Todas as formas de apresentação descritas são limitadas a apenas um tempo de vinculação. Nesse caso, apenas o tempo de vinculação "durante o desenvolvimento" (tempo de implementação) é considerado.
- **Pontos de variação e variantes**

Os pontos de variação podem ser identificados apenas indiretamente nessas formas de representação, pois há vários requisitos que são obviamente contraditórios e a contradição só pode ser resolvida com a atribuição de variantes a produtos específicos (consulte os requisitos R32 e R33). As variantes específicas são descritas diretamente no texto do requisito.

- **Dependências de variantes e verificação**

Nos casos apresentados, as dependências de variantes não são documentadas (formulários 1, 2, 3) ou são documentadas apenas indiretamente (formulário 4) e, portanto, não podem ser verificadas. Erros na configuração, no melhor dos casos, podem ser identificados por um inspetor humano se, por exemplo, requisitos conflitantes forem selecionados para o mesmo produto (se, por exemplo, R32 e R33 tiverem sido selecionados para o Classe A). No Formulário 4, uma dependência de variantes pode ser deduzida indiretamente se houver vários requisitos que se refiram aos mesmos recursos.

Análise dos pontos fortes e fracos

Ao avaliar uma forma específica de apresentação usada para variabilidade, os seguintes critérios são relevantes para a aplicação prática [BOUT2011]:

- Facilidade de ensino: com que facilidade a forma de apresentação escolhida pode ser treinada para pessoal não técnico?
- Escalabilidade: com que facilidade a forma de apresentação escolhida pode ser usada para muitos produtos?
- Capacidade de expansão: quanto esforço é necessário para configurar um novo produto?
- Migrabilidade: Até que ponto a documentação de requisitos existente pode ser desenvolvida na direção da forma de apresentação escolhida sem informações explícitas de variabilidade?
- Verificabilidade: Até que ponto as configurações incorretas na forma de apresentação selecionada podem ser identificadas automaticamente?
- Comparabilidade: Até que ponto os requisitos de diferentes produtos podem ser facilmente comparados?
- Capacidade de alteração: com que facilidade os requisitos existentes para um único produto podem ser alterados sem afetar outros produtos da família de produtos?

7.3 Modelagem de recursos (L3)

A modelagem de recursos é uma técnica comum para documentar a variabilidade. O representante mais conhecido da modelagem de recursos é o – Feature-Oriented Domain Analysis [KCeA1990]. Nesse meio tempo, a modelagem de recursos passou a ser amplamente utilizada, especialmente no desenvolvimento de linhas de produtos, e há muitas extensões e adições à esta abordagem original[ScHT2006].

Um recurso é, portanto, definido como um "*aspecto, qualidade ou característica proeminente ou distinta, visível ao usuário, de um sistema ou sistema de software*" [KCeA1990]. Um modelo

de recursos descreve os recursos e suas interdependências. Uma variante de produto consiste em um conjunto de recursos que descrevem o produto.

Uma configuração de produto (válida) e os recursos que ela contém são definidos pelas condições de limite especificadas pelo modelo de recurso. Os modelos de recursos geralmente são representados graficamente na forma de um diagrama de recursos.

Os elementos descritivos de um modelo de recurso podem ser divididos nas três categorias a seguir:

- Elementos básicos
- Elementos avançados
- Elementos baseados em cardinalidade

Os elementos básicos de um modelo de recurso descrevem os recursos principais e seus filhos e expressam os relacionamentos entre eles. Os recursos filhos podem ter os seguintes relacionamentos com os recursos pais:

- Obrigatório – O recurso filho é obrigatório
- Opcional – O recurso infantil pode ser usado
- Ou – Pelo menos um dos recursos secundários deve ser selecionado
- Alternativa – exatamente um dos recursos secundários deve ser selecionado

Os elementos avançados podem ser usados para definir dependências adicionais entre os recursos. As dependências mais conhecidas são

- A requer B – A seleção do recurso A implica a seleção do recurso B.
- A exclui B – Os recursos A e B não podem estar contidos no mesmo produto.

Os elementos baseados em cardinalidade podem ser usados para especificar melhor os relacionamentos permitidos entre os elementos básicos, por exemplo, adicionando notações como [min, max] ao relacionamento pai-filho.

A notação usada abaixo foi extraída de [CzEi2000].

Figura 16 mostra um modelo de recurso de amostra. O recurso F contém dois recursos obrigatórios f1 e f4, sendo que f1 contém dois recursos obrigatórios f2 e f3, enquanto f4 contém apenas um recurso opcional f5.

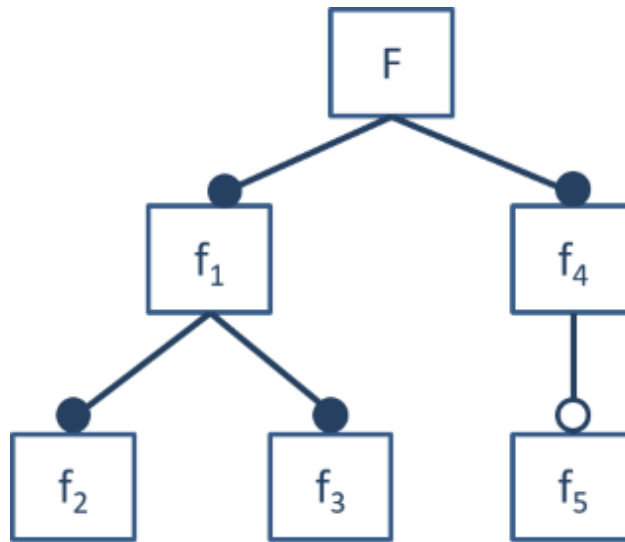
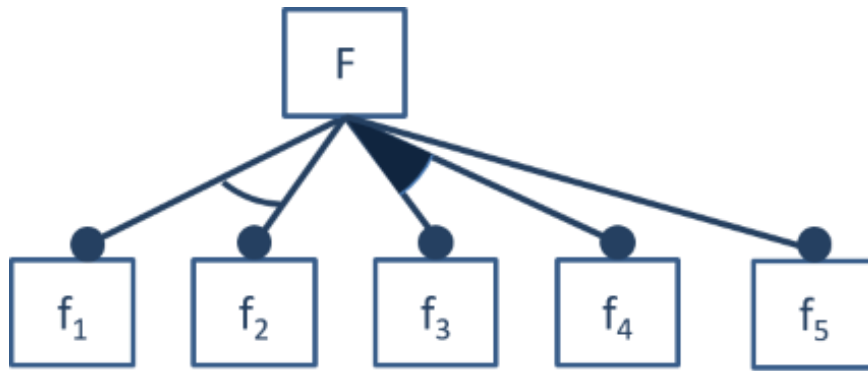


Figura 16: Exemplo de um modelo de recurso simples

Figura 17 mostra a apresentação de relacionamentos alternativos (arco vazio) e ou (arco preenchido) em modelos de recursos. No lado direito, são mostradas todas as configurações possíveis de produtos resultantes desse modelo.



Possíveis configurações do produto:

{F; f1; f3; f5}, {F; f1; f3; f4; f5} {F; f1; f4; f5}

{F; f2; f3; f5}, {F; f2; f3; f4; f5} {F; f2; f4; f5}

Figura 17: Representação das relações de alternativa e ou (à esquerda) e as possíveis configurações de produto resultantes (à direita)

Um exemplo adicional de dependências entre recursos agora pode ser encontrado em Figura 18.

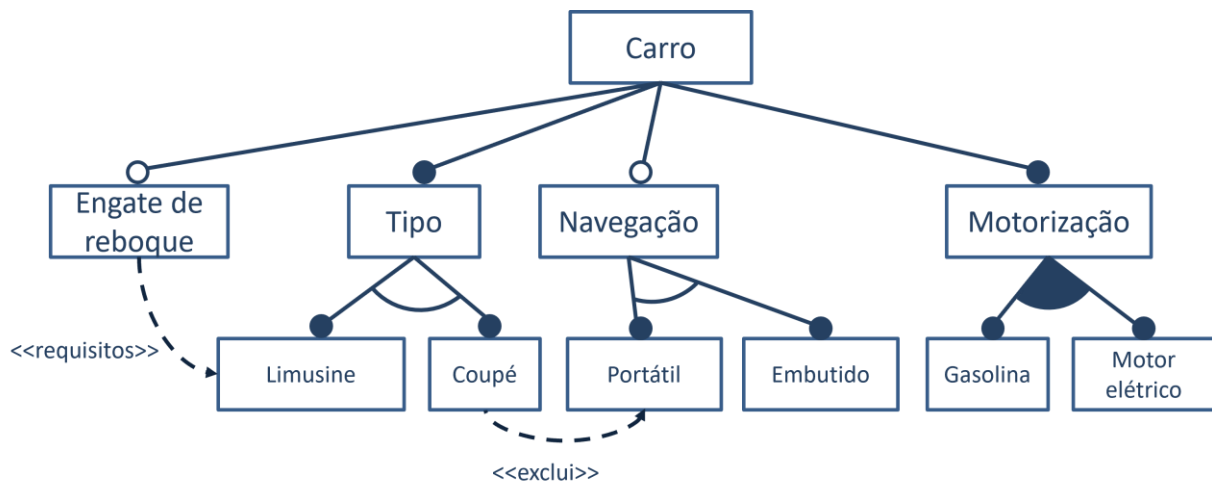


Figura 18: Modelo de recursos para um carro com dependências

Nesse exemplo, um engate de reboque só pode ser instalado se for uma limusine.

Um cupê exclui a instalação de um dispositivo de navegação portátil. O motor pode ser um motor de combustão interna puro (gasolina), puramente elétrico (motor elétrico) ou uma combinação (ou seja, acionamento híbrido).

Na modelagem de recursos, as variantes são representadas por elementos de folha no modelo de recursos. Os pontos de variação são elementos não-folha.

Os relacionamentos de definição de grupo "Ou" e "Alternativa" têm um valor mais alto do que os relacionamentos de definição única "Obrigatório" e "Opcional".

Ao usar "Ou" e "Alternativa", o uso dos rótulos "Opcional" ou "Obrigatório" para os recursos secundários afetados pode, portanto, ser omitido.

Um exemplo de elementos baseados em cardinalidade pode ser encontrado na Figura 19. O produto F precisa ter dois ou três dos recursos f1 a f4.

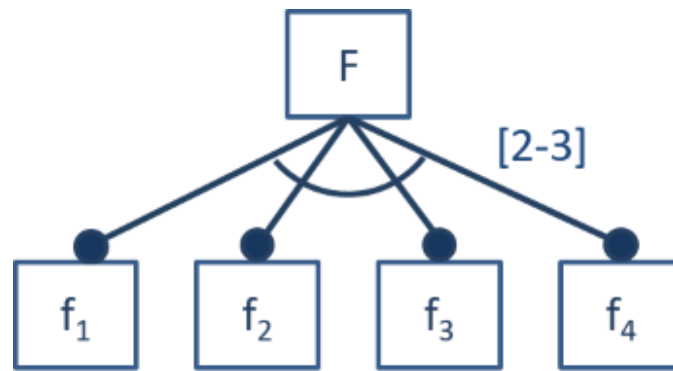


Figura 19: Modelo de recurso com elementos baseados em cardinalidade

Identificação de recursos

Se o Gerenciamento de Variantes baseado em recursos for introduzido, surge a questão de quais recursos serão usados no futuro. Como regra geral, faz sentido não inventar recursos completamente novos, mas usar os documentos de requisitos existentes ao definir os recursos [BoHo2011].

Um bom ponto de partida para identificar recursos é considerar os substantivos em um texto de requisito como recursos em potencial. É importante ignorar os substantivos gerais que não têm nada a ver com o produto em si (p. ex., em seções que tratam de aspectos contratuais ou do processo de desenvolvimento, esses substantivos também são chamados de stop words). Com base nessas listas de substantivos, um especialista pode, em geral, identificar rapidamente os possíveis recursos. No entanto, uma desvantagem desse procedimento é que não é possível identificar pontos de variação específicos que não estejam explicitamente mencionados no texto.

Os pontos de variação geralmente podem ser identificados quando o analista de requisitos pergunta persistentemente o "porquê" de diferentes variantes.

Ferramentas de Suporte

Se a variabilidade deve ser explicitamente documentada, isso geralmente não é possível sem o uso de ferramentas especiais. Há ferramentas disponíveis no mercado que permitem ao usuário

- Criar modelos de recursos,
- Criar configurações de produtos,
- Verifique a admissibilidade das configurações do produto.

Normalmente, essas ferramentas têm interfaces distintas com outras ferramentas nas quais os artefatos de desenvolvimento reais (p. ex., requisitos ou casos de teste) estão localizados e que, portanto, possibilitam relacionar os artefatos de desenvolvimento às características.

8 Relatórios sobre Gerenciamento de Requisitos (L3)

Duração: 1,75 horas

Termos: Relatório, Relatório, Índice [Medida], Meta-Questão-Métrica-Método

Objetivos Educacionais

- OE 8.1 Conhecer os objetivos e benefícios dos relatórios no Gerenciamento de Requisitos (L1)
- OE 8.2 Conhecer as interfaces, o conteúdo e o processo de definição de um relatório (L2)
- OE 8.3.1 Conhecer as figuras-chave típicas do Gerenciamento de Requisitos (L1)
- OE 8.3.2 Dominar os índices de Gerenciamento de Requisitos usando o método Meta-Questionário-Métrico (L3)
- OE 8.4 Conhecer os riscos e problemas no uso de relatórios (L2)

8.1 Objetivos e benefícios dos relatórios no Gerenciamento de Requisitos (L1)

Os relatórios fazem parte do controle organizacional e de projetos. Eles servem para coletar informações sobre projetos ou unidades organizacionais e prepará-las adequadamente para determinados grupos-alvo, a fim de atender às suas necessidades de informação.

[ZIEG1998] define relatório como "a criação e a disseminação de relatórios multifuncionais no sentido de uma compilação organizada de mensagens exclusivamente para a gerência". Outra definição enfatiza a preparação e os objetivos do sistema de relatórios: "Ele pode ser entendido como todas as pessoas, instalações, normas, dados e processos usados para criar e distribuir relatórios. Assim, os relatórios representam informações resumidas sob um objetivo abrangente, um propósito de informação." [KÜPP2005].

O relatório no Gerenciamento de Requisitos (RM) é a coleta, a avaliação e a apresentação de informações sobre os requisitos ou sobre o processo de Gerenciamento de Requisitos e o fornecimento dessas informações.

Um relatório é um documento que combina uma ou mais visualizações para uma finalidade e Stakeholders específicos.

Os comentários a seguir sobre relatórios no Gerenciamento de Requisitos concentram-se principalmente no desenvolvimento de novos sistemas. No entanto, esses procedimentos e princípios também podem ser aplicados em outros contextos, como o aprimoramento contínuo de sistemas, bem como no contexto do Gerenciamento de Requisitos entre projetos.

8.2 Estabelecimento de um sistema de relatórios no Gerenciamento de Requisitos (L2)

8.2.1 Interfaces (L2)

O Gerenciamento de Requisitos está intimamente integrado ao gerenciamento de projetos, ao gerenciamento de produtos e ao gerenciamento da qualidade. Conseqüentemente, essas interfaces também existem no sistema de relatórios. Portanto, faz sentido coordenar o relatório dessas três áreas e seus dados. O gerenciamento de projetos e o gerenciamento da qualidade serão os destinatários mais importantes dos relatórios sobre o Gerenciamento de Requisitos.

Os relatórios do Gerenciamento de Requisitos e do gerenciamento de qualidade geralmente são criados em conjunto.

8.2.2 Conteúdo de um relatório (L1)

Em princípio, os relatórios podem ser criados em qualquer formato (p. ex., informalmente em um texto de e-mail). No entanto, há muitos modelos, portanto, todos os relatórios têm a mesma estrutura. Isso torna os relatórios fáceis e eficientes de ler e criar. As mesmas informações estão sempre no mesmo lugar no relatório. Para o autor, é particularmente prático se o relatório puder ser gerado automaticamente a partir da ferramenta na qual as informações necessárias são gerenciadas de qualquer forma.

O conteúdo padrão dos relatórios é o seguinte:

- **Nome do projeto:** O relatório deve especificar a qual projeto ele se refere. (Se o relatório for para uma unidade organizacional, por exemplo, um departamento, o nome do departamento será exibido aqui em vez de um nome de projeto)
- **Data de criação do relatório:** O conteúdo apresentado no relatório muda diariamente ou até mesmo de hora em hora. Portanto, é importante especificar quando os dados foram extraídos, ou seja, em qual estado de informação o relatório se baseia.
- **Número da versão:** se houver várias versões de um relatório, por exemplo, porque alguém fez uma adição, a nova versão deverá ter um novo número de versão para garantir a exclusividade do relatório e uma melhor rastreabilidade das alterações.
- **Período do relatório:** Os relatórios podem se referir a dias, semanas, meses, anos ou qualquer outro intervalo de tempo. Os relatórios semanais e mensais são os mais comuns, mas, em fases críticas do projeto, também podem ser gerados relatórios diários ou de meio dia. É claro que, ao interpretar o conteúdo do relatório, faz diferença se ele se refere ao que foi alcançado em uma semana ou em um mês.
- **Criador e destinatário:** Um relatório tem um criador (autor) e um destinatário (lista de distribuição). Os destinatários também podem ser distinguidos entre aqueles que recebem apenas para informação e aqueles que precisam aprová-los. Os nomes dessas pessoas geralmente são mencionados no relatório e, portanto, são documentados.

- **Status da liberação:** Se o relatório exigir uma liberação, esse status deve ser anotado aqui. O relatório pode conter conteúdos diferentes em diferentes status de liberação.
- **Status geral:** Logo no início do relatório, o leitor apressado gostaria de ter uma visão geral do grau de criticidade do projeto. Gerentes ocupados só leem o relatório se o projeto for crítico. Os relatórios sobre projetos que seguem o planejado não contêm nenhum valor informativo para o supervisor, pois seu apoio não é necessário. As escalas de semáforos com as conhecidas cores verde, amarela e vermelha são populares.
- **Conteúdo técnico:** Este é o núcleo do relatório.

O conteúdo técnico de um relatório de Gerenciamento de Requisitos consiste em uma ou mais das seguintes classes de conteúdo:

- **Visões dos requisitos** seletivas ou agregadas (consulte 3.5). Geralmente, elas podem ser geradas automaticamente por uma ferramenta de Gerenciamento de Requisitos.
- **Índices** sobre os requisitos ou o processo de ER. Esses índices geralmente são obtidos por meio de avaliações automatizadas. Essa avaliação pode ser feita com uma ferramenta de Gerenciamento de Requisitos ou fora dela.
- **Descrições** descritivas ou avaliativas.

8.2.3 Dicas para o desenvolvimento e a aplicação de relatórios (L2)

No desenvolvimento e na aplicação de relatórios (baseados em requisitos), há algumas dicas ou sugestões práticas que precisam ser consideradas:

- **Concentrar-se no essencial:** Mesmo quando os Stakeholders e os benefícios dos relatórios são conhecidos, a arte é se concentrar no essencial. O método GQM, que descrevemos no capítulo 08.3.2, ajuda nesse caso.
- **Reconciliação:** As informações necessárias para o relatório devem ser fornecidas em um modelo de informações de requisitos e em um esquema de atributos. Como é difícil alterar posteriormente o modelo de informações e o esquema de atributos, e a introdução de um novo atributo exige uma extensa manutenção de conteúdo, os modelos de dados do RM devem ser esclarecidos com antecedência, mesmo antes de um possível projeto de desenvolvimento. É útil usar modelos de referência que já tenham sido alinhados entre si.
- **Coleta de dados:** As pessoas que precisam coletar os dados não são as mesmas que precisam das informações e criam ou leem o relatório. Portanto, os coletores de dados não têm motivação inerente para inserir os dados. Portanto, é ainda mais importante que a coleta de dados esteja bem integrada aos processos de trabalho diários e que esteja claro quem deve inserir quais dados e quando.
- **Qualidade dos dados:** A mera presença de atributos não significa necessariamente que todo o conteúdo seja mantido, atualizado e correto. Embora não faça sentido que um processo de trabalho eficiente introduza muitos campos obrigatórios, especialmente porque algumas informações ainda não estão disponíveis ao criar um requisito, seria importante que os atributos fossem mantidos para a geração de relatórios. A falta de conteúdo leva a informações incompletas nos relatórios.

8.2.4 Processo de definição de relatório (L2)

O gerente de requisitos é responsável por definir os relatórios baseados em requisitos e coordená-los com o modelo de informações de requisitos, ou delega essa tarefa a uma pessoa apropriada.

De acordo com a ISO 15288 ([ISO 15288], 6.3.7.3 a) 1) a 4)), um sistema de medição e relatório é definido nestas etapas:

1. Descrição das características da organização relevantes para a medição.
2. Identificação e priorização das necessidades de informações.
3. Seleção e documentação de índices que atendam a essas necessidades de informação.
4. Definição de procedimentos para coleta, análise e relatório de dados.

8.3 Figuras-chave na Engenharia de Requisitos (L3)

Os índices ou medidas (muitas vezes chamados erroneamente de métricas) são uma parte importante dos relatórios. [EBER2012] na página 436, define-se uma medida como:

"(1) Uma atribuição formal, precisa, reproduzível e objetiva de um número ou símbolo a um objeto para caracterizar uma característica específica.

(2) Matemático: Figura M de um sistema empírico C e suas relações R em um sistema numérico M .

(3) O uso (coleta, análise, avaliação) de uma medida. Exemplos: Medida para um produto (p. ex., defeitos, duração, desvio do plano) ou um processo (p. ex., custos de defeitos, eficiência, eficácia)".

8.3.1 Figuras-chave no Gerenciamento de Requisitos (L1)

As figuras-chave do Gerenciamento de Requisitos podem ser divididas em duas classes principais:

- **Índices do produto** (ou seja, índices sobre os requisitos):
Isso inclui índices sobre o número de requisitos e as características dos requisitos.

Exemplos: Número de requisitos críticos de segurança, comprimento médio de um requisito, número de requisitos defeituosos

- **Índices do processo** (ou seja, índices do processo de ER):

Exemplos: Alterações de requisitos por mês, esforço médio para verificar um requisito, frequência média de alterações nos requisitos nos últimos n meses

8.3.2 Derivação de índices usando o método Meta-Questionário-Métrico (L3)

"Goal-Question-Metric" (GQM) [BaCR1984] é um método possível para garantir que apenas os índices orientados por objetivos sejam definidos para relatórios ou conteúdo de relatórios. O GQM é um procedimento sistemático para identificar essas figuras-chave. Um índice adequado é identificado ao responder às seguintes perguntas:

- Qual objetivo deve ser alcançado com a medição? (*Goal*)
- O que deve ser medido e quais perguntas a medição deve responder? (*Question*)
- Qual(is) índice(s) pode(m) descrever as características necessárias? (*Metric*)

8.4 Riscos e problemas nos relatórios (L2)

Na prática, há dificuldades práticas na coleta e avaliação de dados que resultam em relatórios que não refletem adequadamente a realidade. Como os relatórios devem levar a decisões gerenciais importantes, um relatório incompleto ou até mesmo deliberadamente embelezado pode ter consequências de longo alcance.

Avaliação de dados: representação condensada da realidade

Um relatório é sempre um modelo altamente condensado da realidade, no qual coisas semelhantes são agrupadas em categorias e detalhes insignificantes são omitidos. É muito difícil fazer isso de forma que qualquer pergunta futura possa ser bem respondida a qualquer momento.

É por isso que a superficialidade de um relatório deve ser sempre levada em conta. Em particular, é importante evitar tirar conclusões falsas dos dados disponíveis. Por exemplo, um relatório que mostra 99% de rastreabilidade para todos os requisitos ainda não permite uma declaração sobre o andamento do projeto ou a qualidade dos relacionamentos. Os requisitos que ainda não foram vinculados podem ser os mais importantes ou os que consomem mais tempo e que contribuem significativamente para o sucesso de um projeto. Ao reduzir a complexidade dos índices, é preciso estar sempre ciente desse problema. Muitas vezes, só é possível fazer afirmações e tirar conclusões muito aproximadas.

Baixa qualidade dos dados

Os dados ausentes geralmente são fáceis de detectar. Não é tão fácil avaliar a qualidade dos dados: Os dados correspondem à realidade? Eles estão atualizados? Eles medem exatamente o que deveriam, por exemplo, o atributo "esforço" mede apenas o esforço de implementação, embora o esforço de teste também deva ser levado em conta? A criticidade é realmente o resultado de uma pesquisa especializada ou foi definida provisoriamente?

Deficiências não descobertas, mas também conhecidas, na qualidade dos dados levam ao fato de que o relatório não reflete corretamente a realidade em desenvolvimentos novos e futuros. Devido a dados incorretos, é difícil tomar as decisões gerenciais corretas. E mesmo que a falta de qualidade dos dados seja conhecida, é difícil tomar decisões.

A baixa qualidade dos dados geralmente resulta do fato de que as partes envolvidas negligenciam a manutenção dos dados porque elas próprias têm pouco benefício com isso. Por outro lado, às vezes eles podem estar interessados em embelezar os dados ou, pelo menos, em economizar tempo na manutenção dos dados, não realizando análises cuidadosas, mas inserindo dados que pareçam plausíveis apressadamente.

No entanto, a baixa qualidade dos dados também pode resultar do fato de que nem todos os envolvidos têm a mesma visão. No desenvolvimento ágil (cf. 10), a "*definição de feito*" é um importante tópico de discussão. Deve ser claramente definido quando um requisito é considerado concluído. Os critérios possíveis para a implementação de um requisito são, por exemplo: o código foi criado, os testes de unidade foram criados e executados com sucesso, a documentação foi adaptada e a convenção de código foi seguida.

Regulamentos de proteção de dados

Os regulamentos de proteção de dados gerais e específicos da empresa devem ser seguidos ao definir e implementar o sistema de relatórios. Se os dados pessoais forem fornecidos pelos participantes e posteriormente comunicados dentro da empresa na forma de relatórios sem o conhecimento deles, isso pode gerar problemas. Nesse contexto, é importante acordar claramente com os criadores de dados quem recebe quais dados dentro do escopo das decisões a serem tomadas. Em geral, os dados pessoais e relacionados a pessoas devem ser usados com parcimônia ou não devem ser inseridos. Ao definir as opiniões, deve-se também tomar cuidado para garantir que não sejam feitas declarações sobre pessoas individuais, para não violar involuntariamente as normas de proteção de dados.

Relatórios inflacionários

Se o volume de informações do relatório aumentar constantemente, isso também poderá levar a uma situação em que os destinatários do relatório não consigam processar esses dados devido a restrições de tempo, e decisões importantes não poderão mais ser tomadas com base sólida.

Portanto, menos é mais! O objetivo é concentrar-se nas informações realmente necessárias. Isso também pode significar que diferentes grupos-alvo recebem relatórios diferentes, nos quais apenas determinados aspectos são representados ou em vários níveis de detalhes.

9 Gerenciamento dos processos de Engenharia de Requisitos (L3)

Duração: 2 horas e 30 minutos

Termos: ER iterativa, antecipada, requisitos leves, processo de Engenharia de Requisitos (processo de ER), ciclo PDCA (Planejar, Fazer, Verificar, Agir), melhoria contínua de processos (CPI)

Objetivos Educacionais

- OE 9.1 Conhecer a Engenharia de Requisitos como um processo (L2)
- OE 9.2.1 Conhecer os parâmetros selecionáveis do processo de ER (L1)
- OE 9.2.2 Avaliar a adequação de um processo de ER com relação aos parâmetros do processo (L3)
- OE 9.3 Conhecer e aplicar diferentes métodos para documentar o processo de ER (L3)
- OE 9.4 Conhecer a necessidade de monitorar e controlar o processo de ER (L1)
- OE 9.5.1 Dominar e usar métodos para melhorar o processo de ER (L3)
- OE 9.5.2 Conhecer a necessidade de um plano de Gerenciamento de Requisitos RMP (L2)

9.1 Engenharia de Requisitos como um processo (L2)

Um processo consiste em atividades interdependentes realizadas para atingir uma meta específica. Para cada atividade, os inputs (informações, materiais, energia, recursos) são transformados em outputs (resultados) [ISO 9000]. Cada atividade é atribuída exclusivamente a uma entidade organizacional responsável, por exemplo, uma função. Assim, a Engenharia de Requisitos e o Gerenciamento de Requisitos também são um processo.

O processo de ER é um processo de coleta e Gerenciamento de Requisitos: "Um processo sistemático de desenvolvimento de requisitos por meio de um processo cooperativo iterativo de análise do problema, documentando as observações resultantes em uma variedade de formatos de representação e verificando a precisão do entendimento obtido." [LoKa1995], seção 13.

Esse processo de ER inclui as seguintes atividades principais [IREB2023], capítulo 1:

- Obtenção de requisitos
- Documentação de requisitos.
- Validação e Negociação de Requisitos;
- Gerenciamento de Requisitos

Em cada projeto específico, há várias atividades de investigação, como workshops e reuniões com os Stakeholders, análise de documentos e assim por diante. O mesmo se aplica às outras atividades principais.

O processo de ER usa as necessidades e ideias dos Stakeholders como informações de entrada. Além disso, o status quo antes do início do projeto (p. ex., o sistema legado) e os produtos concorrentes também desempenham um papel importante. O resultado do processo de ER é uma especificação de requisitos validada, livre de conflitos, consistente,

priorizada e com qualidade garantida, que pode servir como base confiável para o trabalho posterior do projeto.

Em geral, as quatro principais atividades têm as seguintes informações de entrada e resultados, que podem, obviamente, ser diferentes, especialmente se for necessário atender a requisitos ou padrões específicos da empresa (consulte a Tabela 2):

Atividade principal	Entrada	Resultados
Elicitando Requisitos	Stakeholders e suas necessidades e ideias; Se aplicável: um sistema legado existente e sua documentação; produtos da concorrência	Requisitos orais e escritos, incluindo a visão do sistema
Documentação de requisitos.	Requisitos orais e escritos	Especificação de requisitos por escrito (textual, baseada em modelo ou ambos)
Validação e Negociação de Requisitos	Especificação de requisitos por escrito	Especificação de requisitos validada, sem conflitos, consistente, priorizada e com garantia de qualidade
Requirements Management de Requisitos	Especificação de requisitos por escrito e solicitações de alteração	Especificação de requisitos sempre atualizada, validada, livre de conflitos, consistente, priorizada e com garantia de qualidade; Preparação de requisitos para grupos individuais de Stakeholders

Tabela 2: Quatro atividades principais de Engenharia de Requisitos e suas entradas e resultados

Essas atividades principais devem sempre ser realizadas, sejam elas documentadas de forma explícita ou implícita. Padrões diferentes exigem uma implementação diferente dessas atividades e definem diretrizes diferentes com relação aos artefatos (consulte também 1.4).

Os resultados do processo de ER devem atender aos critérios de qualidade em três dimensões independentes: especificação, representação e concordância [POHL1994]. Os requisitos devem se tornar mais maduros ao longo do tempo dentro dessas dimensões.

- **Especificação:** Essa dimensão descreve a integridade da especificação. No início do processo de ER, os requisitos são vagos e pouco claros (opacos). À medida que o processo avança, os requisitos se tornam mais completos no sentido de uma cobertura completa do problema a ser resolvido e de uma descrição detalhada o suficiente para ser compreendida adequadamente.

Vários padrões fornecem diretrizes sobre as condições que devem ser atendidas pelos requisitos para que eles sejam considerados completos. No entanto, não é possível comprovar a integridade dos requisitos.

- **Apresentação:** A escala varia de informal a formal. A apresentação informal inclui esboços, textos livres e protótipos. A apresentação semiformal inclui modelos gráficos, como diagramas de classe, máquinas de estado, diagramas de casos de uso ou diagramas de fluxo de dados. Os casos de uso apresentados em forma de tabela, que seguem estritamente uma determinada estrutura sintática, também são semiformais. As especificações formais descrevem claramente os requisitos usando linguagens lógicas e semântica formal. A preparação de uma especificação formal geralmente começa com formas informais de apresentação.
- **Consentimento:** Estabelecer um acordo é outro objetivo durante o processo de ER. A dimensão do acordo passa da visão pessoal para uma visão comum dos requisitos.

A especificação de requisitos deve ser otimizada em todas as três dimensões. Aqui, as atividades de elicitación contribuem principalmente para o aprimoramento da dimensão de especificação, as atividades de documentação para a apresentação e validação e as atividades de negociação para o aprimoramento da dimensão de consentimento. O Gerenciamento de Requisitos visa a manter o nível de qualidade em todas as três dimensões.

9.2 Parâmetros do processo de Engenharia de Requisitos (L3)

O processo de ER pode ser muito variado e deve se adaptar especialmente às restrições impostas. Em todos os vários processos de ER existentes em diferentes abordagens, há apenas um determinado número de parâmetros de processo que podem ser alterados ao selecionar ou ajustar o processo de ER:

- Momento da elicitación,
- Nível de detalhe da documentação, ou seja, especificação pesada versus especificação leve,
- Incorporação de alterações, em particular: Solicitação de alteração versus Backlog do Produto,
- Alocação de responsabilidade.

Esses parâmetros devem ser selecionados de acordo com as restrições. Essas restrições são:

- O tamanho do projeto.
- Trata-se de uma nova implementação ou de um pequeno aprimoramento, melhoria ou variação de um sistema ou produto existente e maduro?
- Foi acordado um preço fixo ou não?
- Há uma equipe estável que trabalha junta há anos?
- Disponibilidade de pessoas e suas qualificações.

Cronograma da elicitação (antecipada ou iterativa)

Os requisitos podem ser determinados completamente no início do projeto (upfront) ou iterativamente (Engenharia de Requisitos Iterativa): No primeiro caso (upfront), uma especificação de requisitos (p. ex., uma folha de especificações) é criada no início do projeto, que descreve completamente o escopo planejado do projeto. Com a Engenharia de Requisitos Iterativa, não se busca definir os requisitos, ou mesmo apenas o escopo do projeto, completamente no início, mas sim considerar a documentação de requisitos (p. ex., o Backlog do Produto) como uma lista preliminar. Os requisitos podem ser adicionados ou alterados a qualquer momento, mesmo durante a implementação.

Observação: há uma diferença entre a Engenharia de Requisitos Iterativa e o desenvolvimento iterativo. Portanto, é possível criar primeiro uma especificação de requisitos completa e, posteriormente, implementar os requisitos por meio de desenvolvimento iterativo.

Se o projeto for um pequeno aprimoramento, melhoria ou variação de um sistema ou produto existente e maduro, é de se esperar que requisitos estáveis possam ser definidos para todo o projeto, com poucas surpresas. É nesse ponto que a determinação antecipada de requisitos é possível e útil.

No entanto, se o projeto for muito inovador, com muitas incertezas, em um ambiente volátil, com Stakeholders indecisos ou conflitantes, ou se houver outros fatores de risco que impossibilitem uma especificação inicial confiável, a Engenharia de Requisitos Iterativa servirá para reduzir o risco.

Nível de detalhamento da documentação de requisitos

O nível de detalhe da documentação ou especificação pode variar entre requisitos pesados e leves: a especificação pesada descreve todos os requisitos em detalhes, incluindo todos os seus atributos e relações de rastreabilidade, tornando a especificação muito abrangente.

A especificação leve descreve os requisitos apenas da forma mais abrangente possível e não antes do necessário. O momento em que determinadas informações são necessárias depende do modelo do processo. O que é necessário depende dos Stakeholders, de suas necessidades e de seu histórico. Uma análise dos Stakeholders específica do projeto ajuda a definir o nível de detalhamento da especificação de requisitos.

Entre outras coisas, o objetivo de uma especificação é permitir que o desenvolvedor entenda o que os Stakeholders querem. Os detalhes da implementação são deixados para o desenvolvedor (especialmente se ele estiver muito familiarizado com o domínio), discutidos verbalmente sem documentação ou refinados usando um protótipo. A especificação de requisitos leves descreve os requisitos como Histórias de Usuários, por exemplo. Os requisitos só são especificados em detalhes quando sua implementação está prestes a começar.

Embora a especificação inicial seja geralmente pesada (p. ex., no modelo em cascata e no V-Modell XT) e iterativa leve (como no Scrum e em outros métodos ágeis, consulte 10), os dois parâmetros de tempo e nível de detalhe são independentes um do outro. É possível

criar uma especificação leve antecipadamente e uma especificação pesada iterativamente (como no Rational Unified Process).

Gerenciamento de mudanças: incorporação de mudanças (solicitação de mudança versus backlog do produto)

Os requisitos mudam durante o projeto. Alguns processos de ER integram requisitos novos ou alterados como solicitações de alteração no processo de especificação e desenvolvimento de requisitos. Em geral, são projetos com preço fixo e especificação antecipada de requisitos, ou seja, a determinação de requisitos é concluída em um determinado momento, do ponto de vista organizacional e jurídico. Do ponto de vista jurídico, as alterações posteriores são alterações contratuais. Uma solicitação de alteração significa legalmente um novo contrato. Normalmente, o contrato já especifica como as alterações devem ser tratadas. Geralmente, elas passam por um procedimento de aprovação simplificado com as seguintes etapas: análise (dos requisitos e seus benefícios), análise de impacto (ou seja, análise das alterações no sistema, seus custos e riscos), decisão do Conselho de Controle de Alterações e, em seguida, implementação. Uma solicitação de mudança geralmente é descrita usando um modelo de solicitação de mudança que atribui a ela um número e um título exclusivos, descreve o problema a ser resolvido e a solução proposta, quantifica custos, benefícios e riscos e gerencia o status (solicitado, aceito, rejeitado, adiado, implementado).

Na Engenharia de Requisitos Iterativa, entretanto, os requisitos são coletados no Backlog do Produto e todos os requisitos – antigos e novos – são tratados igualmente. Isso é possível pelo fato de que nunca nos comprometemos com um escopo de sistema definido. No entanto, não é obrigatório que uma especificação de requisitos inicial trate os requisitos posteriores como solicitações de alteração.

Seria possível ajustar posteriormente o artefato de requisitos criado antecipadamente sem registrar e aprovar as alterações como solicitações de alteração. É claro que as alterações nos artefatos de requisitos devem ser documentadas e rastreáveis.

Alocação de responsabilidade

Uma única função (p. ex., o gerente de requisitos) pode ser responsável pelo processo de ER no sentido de que ele planeja, controla e melhora o processo de ER.

Ele ou ela também pode realizar todas as atividades do processo de ER por conta própria. No entanto, também pode haver uma equipe inteira ou várias funções responsáveis pela Engenharia de Requisitos, seja para diferentes atividades ou diferentes conteúdos (p. ex., requisitos funcionais versus requisitos de usabilidade ou utilidade). A Engenharia de Requisitos também pode ser estreitamente integrada ao processo de desenvolvimento sem um processo de ER separado ou a função de analista de requisitos. Nesse caso, a equipe de desenvolvimento executa as atividades de Engenharia de Requisitos, ou seja, os membros da equipe extraem, documentam, verificam e gerenciam os requisitos.

9.3 Documentar o processo de Engenharia de Requisitos (L3)

O processo de ER consiste em várias atividades dos quatro tipos mencionados acima, como workshops de elicitação, revisões de especificações etc., conforme explicado no syllabus do Foundation Level [IREB2023]. Muitas dessas atividades são planejadas na forma de reuniões, pois frequentemente há muitas pessoas envolvidas. A ordem dessas atividades resulta da escolha dos parâmetros do processo (consulte 9.2), que podem ser definidos especificamente para o projeto ou também para toda a empresa. As atividades e sua sequência podem ser apresentadas como um diagrama de atividades UML. O diagrama de atividades também pode mostrar a atribuição de atividades a funções.

A atribuição de responsabilidades pelas atividades às funções também pode ser apresentada em mais detalhes usando uma matriz RACI como a seguinte. RACI significa:

- **R** = *responsible* = o responsável pela execução
- **A** = *accountable* = aprovador, ou seja, a pessoa que autoriza, por exemplo, a atividade e seu orçamento
- **C** = *consultado* = (será) consultado, especialmente em termos de responsabilidade técnica e relacionada ao conteúdo
- **I** = *informed* = informado, a pessoa deve ser informada

A Tabela 3 mostra um exemplo de um trecho de uma matriz RACI.

Atividade	Engenheiro de Requisitos	Chefe de projeto	Palavras-chave
Análise de documentos: Manual dos sistemas legados	R, A	I	
Workshop de criatividade com os principais usuários	R	A	C
...			

Tabela 3: Exemplo de uma matriz RACI para Engenharia de Requisitos.

Para gerenciar datas e orçamentos quantitativamente, o processo de ER também pode ser apresentado como um plano de projeto.

Outros documentos que podem representar e apoiar o processo de ER são: Plano de projeto, listas de verificação, modelos, exemplos de documentos e diretrizes.

Se muitas pessoas estiverem envolvidas no processo de ER, também faz sentido apoiar esse processo com uma ferramenta. Todos os sistemas de gerenciamento de fluxo de trabalho, em seu sentido mais amplo, são adequados para isso.

9.4 Monitoramento e controle do processo de Engenharia de Requisitos (L1)

Monitorar o processo de ER significa garantir que todas as atividades sejam realizadas e que os resultados definidos sejam entregues no prazo e que as atividades permaneçam dentro do orçamento. Relatórios que registram regularmente as datas, o orçamento consumido, o status e a porcentagem de conclusão do processo de ER e suas atividades individuais e comparam os valores reais com os valores-alvo do planejamento são úteis para isso (consulte 8).

Controlar o processo de ER significa executá-lo de acordo com o plano ou, se o processo se desviar do plano, tomar medidas corretivas. Por exemplo, se ficar claro que o prazo ou o orçamento não poderá ser cumprido, as consequências para o projeto como um todo devem ser determinadas e, se apropriado, devem ser tomadas contramedidas. Para ajustar o processo em andamento ao plano, as atividades planejadas podem ter de ser omitidas, antecipadas ou realizadas com menos esforço. Devem ser feitas compensações cuidadosas quando causarem o menor dano possível, por exemplo, grupos individuais de Stakeholders não são entrevistados, perguntas abertas individuais não são esclarecidas, detalhes não são especificados, solicitações de mudanças sem importância são rejeitadas e assim por diante. É importante considerar o risco: O benefício da economia supera o possível dano?

9.5 Melhoria do processo de Engenharia de Requisitos (L3)

Um processo ainda pode ser melhorado. A base de toda melhoria de processo é a análise e a documentação do status atual (consulte 9.3). O status atual do Gerenciamento de Requisitos pode ser documentado em um plano de Gerenciamento de Requisitos (RMP). Ele descreve o Gerenciamento de Requisitos de uma empresa ou projeto: o processo de ER, o Modelo de Informações de Requisitos, os atributos e visualizações e todas as outras especificações descritas neste syllabus. O RMP serve como documentação do estado atual como base para refletir a abordagem atual e também pode estruturar a melhoria planejada do processo.

A melhoria de um processo pode ser realizada de forma abrupta – um rearranjo do processo – ou contínua. Um rearranjo de processo altera muitas atividades e parâmetros do processo ao mesmo tempo. Isso tem a vantagem de possibilitar um aumento significativo na eficiência, o que, no entanto, geralmente só ocorre depois que todos os participantes se acostumam com o novo processo. No entanto, há também o risco de que o novo processo não prove seu valor e reduza a eficiência. A redefinição envolverá novamente um grande esforço.

A melhoria contínua do processo evita esse risco e leva a melhorias de curto prazo com pouco esforço. De acordo com o princípio do aprimoramento contínuo de processos (CPI), os processos são otimizados gradualmente por meio da repetição iterativa das quatro atividades a seguir (PDCA):

- **Planejar:** O processo real e, em particular, a necessidade de aprimoramento são analisados. Com base nisso, o processo desejado é planejado e documentado.

- **Fazer:** As ações de melhoria são desenvolvidas e testadas em um projeto piloto e acompanhadas de medições.
- **Verificação:** Verifica-se se as ações produziram a melhoria desejada. Os valores reais são comparados com os valores planejados.
- **Agir:** Com base nos resultados da análise do plano real, as ações de melhoria são introduzidas continuamente ou, se necessário, novas ações são planejadas. A implementação das ações é monitorada e acompanhada por medições.

O processo real e o processo alvo são caracterizados usando quantidades medidas (consulte 8). Essas quantidades medidas podem ser:

- A proporção do orçamento do projeto investido em Engenharia de Requisitos. Tanto o excesso quanto a falta podem ser questionáveis. Normalmente, é de 10% a 30% do orçamento do projeto.
- O número de requisitos ainda a serem implementados (ponderado de acordo com o esforço esperado). Ele mede o trabalho ainda a ser feito antes do final do projeto.
- Taxa de burndown ou velocidade, ou seja, o número de requisitos que são implementados por unidade de tempo, ponderados de acordo com o esforço. Juntamente com o conhecimento dos requisitos ainda a serem implementados, é possível fazer previsões sobre a duração restante do projeto.
- Alterar a taxa de requisitos. Uma taxa de 1 a 5% dos requisitos por mês (medida em esforço) e de 30 a 50% durante a duração do projeto é considerada normal [EBER2012]. Menos mudanças podem significar que ninguém está realmente interessado nos requisitos e que os Stakeholders não estão suficientemente envolvidos. O excesso de mudanças também é um sinal de alarme: Os requisitos ainda não são estáveis, os grupos de Stakeholders podem ser muito heterogêneos ou estar em conflito, e ainda é muito cedo para implementar os requisitos.
- Tempo de processamento das solicitações de mudança, do pedido à implementação.

Com a ajuda do benchmarking, é possível descobrir quais números são significativos e alcançáveis como valores-alvo.

As ações de melhoria podem se referir aos parâmetros do processo descritos em 9.2 ou à forma como as atividades individuais são realizadas em detalhes, por exemplo, com a ajuda de quais métodos.

Outra possibilidade de aprimoramento do processo é analisar os erros cometidos na Engenharia de Requisitos, por exemplo, defeitos encontrados durante a inspeção de especificação ou defeitos entregues com o sistema que podem ser rastreados até a Engenharia de Requisitos. Em seguida, perguntamos sobre suas causas e as causas básicas. Isso dá ideias para ações de aprimoramento.

Modelos de maturidade como o CMMI [CMMI2010] ou o ITIL [BEIM2012], [EBEL2014] oferecem ajuda mais concreta para o aprimoramento de processos na Engenharia de Requisitos (mas não só). Eles descrevem atividades ou práticas que devem ser realizadas para atingir um determinado nível de maturidade. A introdução de novas atividades e práticas que ainda não foram implementadas representa, então, um aprimoramento do

processo. Todos os outros métodos de aprimoramento de processos também podem ser usados, como o TQM (Total Quality Management) e o Six Sigma.

Em particular, o aprimoramento da Engenharia de Requisitos apoia a coleção de práticas recomendadas por Sommerville e Sawyer [SoSa1997].

Plano de Ação [WIEG2005], página 66] apoia o planejamento concreto da melhoria de processos. Esse plano de ação contém o seguinte:

- Nome do projeto de melhoria,
- Data,
- Metas (de aprimoramento, expressas como metas de negócios),
- Indicadores de sucesso (ou seja, cumprimento de metas),
- Influência organizacional da mudança,
- Participantes (funcionários, suas funções e orçamentos de tempo),
- Processo de medição e relatório (quando o progresso das ações dentro deste plano será monitorado, por quem e como),
- Dependências, riscos e condições de limite,
- Data estimada de conclusão de todas as ações deste plano,
- Ações (3 a 10 por plano) com pessoa responsável, data-alvo, objetivo, descrição, itens de entrega e requisitos de recursos.

Ao aprimorar o processo de ER, deve-se observar que ele não pode ser otimizado por si só, mas somente em cooperação com outras atividades do projeto, como gerenciamento de projetos, desenvolvimento e testes. As mudanças no processo de ER também afetarão o trabalho dessas pessoas.

10 Gerenciamento de requisitos em projetos ágeis (L2)

Duração: 90 min

Termos: História de usuário, Sprint, Backlog do Produto, Gráfico Burndown

Objetivos Educacionais

OE 10.1 Conhecer os princípios básicos do desenvolvimento ágil de software (L1)

OE 10.2 Conhecer as principais atividades e artefatos no Gerenciamento de Requisitos em projetos Agile (L1)

OE 10.3 Saber como a atividade de Gerenciamento de Requisitos é mapeada para as atividades do Scrum (L2)

10.1 Histórico (L1)

As abordagens clássicas para o desenvolvimento de software e, portanto, também para a Engenharia de Requisitos, enfatizam um plano que se estende até o futuro. No final da década de 1990 e no início dos anos 2000, surgiu um movimento contrário a esse. Isso afastou os planos de longo prazo e deu ênfase aos planos de curto prazo com muitos ciclos de feedback. A classificação de [BoTu2003] baseia-se nessa diferença: os dois grupos são chamados de "ágeis" (conforme proposto no Manifesto Ágil) e "orientados por planos" (no original: plan-driven). Em geral, o Manifesto Ágil é a base comum de todas as abordagens Ágeis. Do ponto de vista dos desenvolvedores de software, o Manifesto Ágil afirma [AgiMa2001]:

"Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o e ajudando outros a fazê-lo. *Através deste trabalho, passamos a valorizar:*

Indivíduos e interações *mais que processos e ferramentas*

Software funcionando *mais que documentação abrangente.*

Colaboração com o cliente *mais que negociação de contratos.*

Responder a mudanças *mais que seguir um plano.*

Ou seja, embora haja valor nos itens da direita, valorizamos mais os itens da esquerda."

Esses valores mostram que a abordagem Agile tem mais a ver com a colaboração da equipe, a produtividade e os pontos fortes individuais do que com contratos e documentação. Isso distingue os métodos ágeis das abordagens orientadas por planos que exigem elementos contratuais claros (p. ex., escopo do projeto, planos de liberação ou um processo de mudança definido).

10.2 Gerenciamento de requisitos em projetos ágeis (L1)

Tanto as abordagens ágeis quanto as orientadas por planos são inerentemente heterogêneas (consulte, por exemplo, [KoBE2013]). Portanto, não se pode afirmar, em geral, como os requisitos são tratados em projetos ágeis. Portanto, as declarações a seguir não

devem ser entendidas como completas ou abrangentes. A seção a seguir usa principalmente o Scrum [SuSc2013] como um representante típico das abordagens ágeis.

O Scrum é uma "estrutura na qual as pessoas podem lidar com problemas adaptativos complexos e, ao mesmo tempo, entregar produtos do mais alto valor possível de forma produtiva e criativa" [SuSc2013].

A forma de requisito mais discutida em projetos Agile é a história do usuário. Uma História de Usuário mostra a função solicitante, o requisito em si e o benefício. O Product Owner (PO) é responsável pela história do usuário. A partir disso, a equipe de desenvolvimento (abreviada como DT) obtém instruções de trabalho concretas (tarefas) para um ciclo de desenvolvimento (sprint).

Uma História de Usuário geralmente tem o seguinte formato:

Como <ROLE>, quero <TARGET/WISH>, para que <BENEFIT>.

A especificação do benefício é opcional. Os critérios de aceite e os casos de teste são especificados com mais precisão para uma História de Usuário. Eles têm o seguinte formato:

Sob a condição de que <PRECONDIÇÃO>, se <TRIGGER>, então <RESULTADO>.

As Histórias de Usuários são frequentemente anotadas com a estimativa de implementação ("Qual é a complexidade da implementação da História de Usuário?") e sua contribuição de valor para o projeto/visão. As Histórias de Usuários geralmente são classificadas pela ordem de implementação planejada. As Histórias de Usuários podem ser agrupadas, por exemplo, em épicos (uma História de Usuário que é grande demais para ser implementada sozinha e, portanto, compartilhada) ou em temas (Histórias de Usuários com conteúdo semelhante).

A maioria das abordagens ágeis não especifica estritamente com quais artefatos se deve trabalhar. O Caso de Uso, que também está bem estabelecido como uma forma de expressão no desenvolvimento clássico de software, pode ser usado como um artefato em um ambiente ágil (p. ex., [COCK2001] ou [JaSB2011]).

Os artefatos (p. ex., Histórias de Usuários ou casos de uso) são frequentemente avaliados e priorizados em um Backlog do Produto. No Scrum, essa é a responsabilidade do Product Owner. O Product Owner é a pessoa que gerencia os artefatos a serem implementados. Para isso, o Product Owner prioriza os artefatos no Backlog do Produto, com uma prioridade baseada na contribuição de valor do artefato e nas necessidades dos Stakeholders. A ordem de implementação dos artefatos resulta, por um lado, de sua prioridade e, por outro, das dependências tecnológicas.

Os relatórios escritos são minimizados nas abordagens ágeis por meio de um trabalho altamente interativo: a comunicação de todas as partes envolvidas deve reduzir o esforço necessário para os relatórios escritos. Reuniões regulares ("eventos") são usadas para esse fim.

Novamente, isso varia entre as diferentes abordagens. No Scrum, serão usadas reuniões diárias da equipe de desenvolvimento ("Daily Standup"), revisões e retrospectivas. Durante as revisões, o resultado de um ciclo de desenvolvimento (no Scrum: "Sprint") será apresentado e aprovado pelo Product Owner. Em uma Retrospectiva, a interação de uma equipe será avaliada retrospectivamente. Os relatórios escritos incluem gráficos Burndown ("Quanto trabalho ainda precisa ser feito neste sprint?") ou um Backlog de impedimentos ("O que atrapalha a equipe em seu trabalho?"). O Scrum Master (abreviado como SM) garante que essas reuniões sejam realizadas de acordo com as regras.

Embora o termo processo raramente seja usado nas abordagens ágeis, as diretrizes das abordagens implicam em um fluxo de trabalho concreto. O aprimoramento desses processos será obtido por meio de retrospectivas regulares.

O dimensionamento de abordagens ágeis para equipes grandes e distribuídas ainda está engatinhando, mas atualmente algumas estruturas estão sendo desenvolvidas. Algumas abordagens podem ser encontradas em [ECKS2004], [ECKS2010], [LEFF2011] e [KoBE2013].

10.3 Mapeamento das atividades de Gerenciamento de Requisitos para as atividades do Scrum (L2)

O Scrum especifica apenas os processos gerais de trabalho. Na Tabela 4, as atividades de RM são atribuídas às atividades ou artefatos do Scrum. Além disso, a função de execução é especificada no Scrum. Nem todas as atividades de RM são abordadas no Guia do Scrum. Além do Guia do Scrum, há uma quantidade não desprezível de literatura que descreve adições mais ou menos bem-sucedidas ao Scrum. Se e como a atividade de RM correspondente será executada em um projeto Scrum é decisão da equipe Scrum.

Atividades de RM	Atividade ou artefato do Scrum	Função Scrum
Atributo	Histórias de usuário no backlog: Descrição, ordem, estimativa, status e valor. Opcional: Agrupamento	PO, DT
Avaliação e definição de prioridades	<ul style="list-style-type: none"> ▪ Estimativa dos benefícios e custos por meio do Planning Poker (DT) ▪ Disposição das Histórias de Usuário em um Backlog do Produto (PO) ▪ Seleção de Histórias de Usuário para uma Sprint (PO e DT) ▪ Priorização em uma Sprint (DT) 	<ul style="list-style-type: none"> ▪ DT ▪ PO ▪ PO e DT ▪ DT
Rastreabilidade	<p>Há uma rastreabilidade implícita das Histórias de Usuários para os casos de teste de aceite correspondentes e, com a associação adequada de atributos, para as fontes das Histórias de Usuários.</p> <p>Além disso, a rastreabilidade é possível dentro do Backlog do Produto (dependências) e das Histórias de Usuário para o código-fonte.</p> <p>O Scrum não diz nada sobre conectar as Histórias de Usuário dentro do Backlog do Produto. A rastreabilidade por meio de épicos (Histórias de Usuário agrupadas) seria concebível.</p> <p>A rastreabilidade será documentada somente se necessário.</p>	Nenhum
Controle de versão dos requisitos	O controle de versão das Histórias de Usuários é desnecessário. A versão atual de uma História de Usuário é sempre relevante.	Nenhum ou PO
Mudanças	As alterações podem ser propostas a qualquer momento. Novos requisitos levam a novas Histórias de Usuários, mudanças de requisitos levam ao fato de que uma História de Usuário é alterada ou substituída por uma nova.	PO
Gerenciamento de variantes	Os métodos ágeis não oferecem suporte explícito ao gerenciamento de variantes. No entanto, é possível usar métodos padrão de gerenciamento de variantes.	PO

Atividades de RM	Atividade ou artefato do Scrum	Função Scrum
Relatórios	Os relatórios são principalmente verbais. Os artefatos usados para rastrear o status de conclusão também podem servir como relatórios: <ul style="list-style-type: none"> ▪ Standup diário ▪ Revisão do Sprint ▪ Retrospectiva da Sprint ▪ Backlog do Produto ▪ Backlog da Sprint ▪ Gráfico de Burndown 	DT
Gerenciamento de processos	Retrospectiva de Sprint e Backlog de Impedimentos	SM, DT

Tabela 4: Mapeamento das atividades de Gerenciamento de Requisitos para as atividades do Scrum.

11 Uso de ferramentas no Gerenciamento de Requisitos (L1)

Duração: 0,75 horas

Termos: Ferramenta, seleção de ferramenta, troca de dados

Objetivos Educacionais

OE 11.1 Conhecer o papel das ferramentas no Gerenciamento de Requisitos e suas principais funções (L1)

OE 11.2 Conhecer o procedimento básico para a seleção de ferramentas de Gerenciamento de Requisitos (L1)

OE 11.3 Conhecer a necessidade de troca de dados entre ferramentas (L1)

11.1 Papel das ferramentas no Gerenciamento de Requisitos (L1)

O uso de ferramentas tem o objetivo de facilitar a documentação e o gerenciamento dos requisitos pelo gerente de requisitos. Devido às suas funcionalidades especiais, as ferramentas de Gerenciamento de Requisitos permitem uma visão holística dos requisitos, pois, entre outras coisas, podem ser representadas as relações entre diferentes requisitos, bem como o ciclo de vida de requisitos individuais.

Uma ferramenta de Gerenciamento de Requisitos é um aplicativo de software cujo principal objetivo é dar suporte às atividades de Gerenciamento de Requisitos.

Muitos aplicativos são tradicionalmente usados no desenvolvimento de software e sistemas. Muitos deles abrangem alguns aspectos da Engenharia de Requisitos e/ou do Gerenciamento de Requisitos. Portanto, a distinção entre essas ferramentas e as de Gerenciamento de Requisitos nem sempre é clara.

As ferramentas de Gerenciamento de Requisitos geralmente se baseiam em suposições sobre o método de desenvolvimento. Algumas ferramentas são projetadas para fins específicos:

- Modelos de processo (p. ex., desenvolvimento ágil, prototipagem ou gerenciamento de produtos)
- Domínios de aplicação (p. ex., indústria automotiva, engenharia médica ou sistemas de defesa)
- Ambientes de trabalho (p. ex., escritório virtual, colaboração local ou colaboração global)

Isso resulta em uma variedade de ferramentas de Gerenciamento de Requisitos que, muitas vezes, suportam apenas algumas das tarefas de um gerente de requisitos.

As funções centrais de uma ferramenta de Gerenciamento de Requisitos são [SoSa1997]:

- Editor de requisitos, incluindo seus atributos
- Importação de requisitos de documentos existentes para a ferramenta e exportação de requisitos gerenciados para outros formatos

- Rastreamento de requisitos
- Controle de versão dos requisitos e criação de configurações
- Criação de visualizações de requisitos

Conforme descrito em [RuSo2009], é importante garantir que a ferramenta de Gerenciamento de Requisitos se adapte aos procedimentos e processos estabelecidos na empresa. Uma vez criados o processo e o modelo de dados básicos da empresa, é possível realizar uma avaliação da ferramenta.

11.2 Procedimento básico para a seleção de ferramentas (L1)

Com base em [RuSo2009], as seguintes etapas são necessárias ao selecionar uma ferramenta para o Gerenciamento de Requisitos:

- Lançamento de um projeto de seleção de ferramentas
- Definição de critérios de seleção aproximados por meio da formulação de requisitos básicos.
- Realização da seleção aproximada (lista longa) para identificar os primeiros sistemas potenciais.
- Refinamento do catálogo de critérios com base em requisitos novos e refinados para a ferramenta.
- Realização de uma boa seleção (lista restrita), até um candidato de software favorito.
- Se nenhuma ferramenta atender exatamente aos requisitos, será necessário adaptar (personalizar) o aplicativo de software.
- Para fortalecer a aceitação na empresa e eliminar possíveis últimas preocupações, um projeto piloto será finalmente lançado.

Caso o projeto piloto revele que a ferramenta selecionada não oferece o suporte desejado, a seleção da ferramenta deve ser repetida. Se nenhuma ferramenta atender exatamente aos requisitos, poderá ser feito um ajuste do processo em vez de ajustar uma ferramenta.

11.3 Intercâmbio de dados entre ferramentas de Gerenciamento de Requisitos (L1)

Quando várias empresas trabalham juntas em um projeto, muitas vezes é necessário trocar requisitos entre diferentes ferramentas de Gerenciamento de Requisitos.

Esse é o caso, por exemplo, se um fornecedor quiser refinar ainda mais os requisitos do cliente dentro do escopo do desenvolvimento do sistema ou estabelecer a rastreabilidade para artefatos de desenvolvimento subsequentes, como design ou casos de teste.

A troca de informações por meio de formatos de documentos padrão, que normalmente podem ser exportados e importados, é insuficiente, pois geralmente são perdidas informações relevantes, como um identificador exclusivo, o editor, informações estruturais ou informações de controle de versão sobre os requisitos. Acima de tudo, essa abordagem não oferece suporte suficiente para a troca de atualizações (p. ex., após a incorporação de alterações).

Para permitir uma troca de requisitos independente de ferramentas e, ao mesmo tempo, manter as características centrais dos requisitos, o padrão do setor Requirements Interchange Format (ReqIF) foi criado e padronizado pelo Object Management Group.

As vantagens essenciais resultantes do uso do ReqIF são:

- Com o ReqIF, a colaboração entre empresas pode ser aprimorada com a aplicação de métodos de Gerenciamento de Requisitos em todas as empresas.
- Os parceiros não precisam trabalhar com a mesma ferramenta. Os fornecedores não precisam ter sua própria ferramenta de Gerenciamento de Requisitos para cada cliente.
- Os requisitos podem ser transferidos dentro de uma organização, mesmo entre os limites da ferramenta.
- Com o ReqIF, os requisitos com todos os atributos e meta-informações podem ser trocados sem perda, ao contrário das exportações de documentos para Word, PDF etc.

Atualmente, várias ferramentas suportam o formato ReqIF.

12 Referências

- [AgiMa2001] Manifesto for Agile Software Development. Disponível em <http://agilemanifesto.org/>. Última alteração em fevereiro de 2024.
- [ATKI2002] C. Atkinson: Component-based Product Line Engineering with UML Addison-Wesley, 2002.
- [BaCR1984] V. Basili, G. Caldiera, e H. D. Rombach: The Goal Question Metric Approach. Em: Encyclopedia of Software Engineering. John Wiley & Sons, 1994, S. 528–532.
- [BEIM2012] Martin Beims: IT-Service Management mit ITIL®: ITIL® Edition 2011, ISO 20000:2011 und PRINCE2® in der Praxis, Carl Hanser Verlag GmbH & Co. KG, 3., aktualisierte Auflage, 2012.
- [BoTu2003] B. Boehm e R. Turner: Using Risk to Balance Agile and Plan-Driven Methods. Em: IEEE Computer, Volume 36, Edição 6, junho de 2003, pp. 57–66.
- [BOUT2011] E. Boutkova: Experience with Variability Management in Requirement Specifications. Em: D.E. Almeida, T. Kishi, C. Schwanninger, I. John e K. Schmid (eds): Software Product Lines – 15th International Conference (SPLC), München, 2013, pp. 303–312.
- [BoHo2011] E. Boutkova e F. Houdek: Semi-automatic identification of features in requirement specifications. Em: Anais da 19ª Conferência Internacional de Engenharia de Requisitos, Trento, Itália, setembro de 2011, pp. 313–318.
- [COCK2001] A. Cockburn: Writing Effective Use Cases. Addison-Wesley, 2001.
- [CzEi2000] K. Czarnecki e U.W. Eisenecker: Generative Programming: Methods, Tools, and Applications. Addison-Wesley, 2000.
- [CHSM2013] J. Cleland-Huang, R. S. Hanmer, S. Supakkul, e M. Mirakhorli: The Twin Peaks of Requirements and Architecture. IEEE Software, vol. 30, no. 2, pp. 24–29, março-abril, 2013
- [CINo2007] P. Clements and L. Northrop: Software Product Lines: Practices and Patterns (Práticas e padrões). Addison-Wesley, Boston, 6. Auflage, 2007.
- [CMMI2010] CMMI® for Development, versão 1.3. CMU/SEI-2010-TR-033. Disponível em https://insights.sei.cmu.edu/documents/853/2010_005_001_15287.pdf. Última alteração em fevereiro de 2024.
- [CoHW1998] J. Coplien, D. Hoffmann, e D. Weiss: Commonality and Variability in Software Engineering (Commonality e Variabilidade na Engenharia de Software). In: IEEE Software, volume 15, assunto 6, 1998, pp. 37–45.
- [DAVI2003] A. Davis: The Art of Requirements Triage (A arte da triagem de requisitos). Em: IEEE Computer, Volume 36, Edição 3, 2003, pp. 42–49.
- [DAVI2005] A. Davis: Just Enough Requirements Management, Dorset House, 2005.

- [EBEL2014] N. Ebel: ITIL®(R) 2011 Edition: Grundlagen und Know-how für das IT Service Management und die ITIL®(R)-Foundation-Prüfung, dpunkt.verlag GmbH, 1. Auflage, 2014
- [EBER2012] C. Ebert: Systematisches Requirements Engineering. Dpunkt, 4. Auflage, 2012.
- [ECKS2010] J. Eckstein: Agile Software Development with Distributed Teams. Dorset House Publishing, 2010.
- [ECKS2004] J. Eckstein: Agile Software Development in the Large. Dorset House Publishing, 2004.
- [GABLI2014] Springer Gabler Verlag (publisher), Gabler Wirtschaftslexikon, keyword: product line, online on the Internet: <https://wirtschaftslexikon.gabler.de/definition/produktlinie>, available in german only.-43488
- [GLIN2008] M. Glinz: A Risk-Based, Value-Oriented Approach to Quality Requirements. IEEE Software, Nr. 2, pp. 34-41, 2008.
- [GoFi1994] O.C.Z. Gotel and A.C.W Finkelstein: An Analysis of the Requirements Traceability Problem. In Proceedings of IEEE International Conference on Requirements Engineering, 1994, pp. 94-101.
- [HuJD2011] E. Hull, K. Jackson, and J. Dick: Requirements Engineering. Springer, 3. Auflage, 2011.
- [DIN 61508] IEC DIN EN 61508-2 Functional safety of safety-related electrical/electronic/programmable electronic systems. VDE Verlag, 2002.
- [IREB2023] Syllabus IREB Certified Professional for Requirements Engineering – Foundation Level <https://www.ireb.org/en/downloads/#cpre-foundation-level-syllabus-3-0>. Última alteração em fevereiro de 2024.
- [IREB Glossary] M. Glinz: A Glossary of Requirements Engineering Terminology. <https://www.ireb.org/en/downloads/tag:glossary#top>. Última alteração em fevereiro de 2024.
- [ISO 12207] International Organization for Standardization (ISO): ISO/IEC 12207: 2008, Systems and software engineering — Software life cycle processes. 2008.
- [ISO 15288] International Organization for Standardization (ISO): ISO/IEC 15288: 2015, Systems and software engineering — Software life cycle processes. 15288.
- [ISO 26262] International Organization for Standardization (ISO): ISO 26262, Road vehicles -- Functional safety. 2011.
- [ISO 26550] International Organization for Standardization (ISO): ISO/IEC 26550:2013: Software and systems engineering -- Reference model for product line engineering and management. 2013.
- [ISO 9000] International Organization for Standardization (ISO): ISO 9000:2005: Quality management systems — Fundamentals and vocabulary. 2005.

- [JaSB2011] I. Jacobson, I. Spence, e K. Bittner: Use Cases 2.0. Ivar Jacobson International, 2011.
- [KCeA1990] C. Kang, S. Cohen, J. Hess, W. Novak, e A. Peterson: Feature-Oriented Domain Analysis – Estudo de viabilidade. Software Engineering Institute, 1990.
- [KaRy1997] J. Karlsson e K. Ryan: A Cost-Value Approach for Prioritizing Requirements. IEEE Software 14, Nr. 5, pp. 67–74, 1997.
- [KoBE2013] H.-P. Korn e J.P. Berchez (eds.): Agiles IT-Management in großen Unternehmen. Symposium, 2013. Disponível apenas em alemão.
- [KÜPP2005] H.-U. Küpper: Controle: Konzeption, Aufgaben, Instrumente. Schäffer-Poeschel, 4. Auflage, 2005. Disponível apenas em alemão.
- [LEFF2011] D. Leffingwell: Agile Software Requirements, Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison-Wesley Professional, 2011.
- [LoKa1995] P. Loucopoulos e V. Karakostas: System Requirements Engineering. McGraw-Hill, 1995.
- [MaGP2009] P. Mäder, O. Gotel, e I. Philippow: Getting Back to Basics: Promoting the Use of a Traceability Information Model in Practice. Em: Anais do 5º Workshop Internacional sobre Rastreabilidade em Formas Emergentes de Engenharia de Software (TEFSE2009), Vancouver, Canadá, maio de 2009, pp. 21–25.
- [MJZC2013] P. Mäder, P.L. Jones, Y. Zhang, e J. Cleland-Huang: Strategic Traceability for Safety-Critical Projects. In: IEEE Software, volume 30, assunto 3, maio / junho 2013, pp. 58–66.
- [MOIS2002] F. Moisiadis: The fundamentals of prioritizing requirements. In: Systems Engineering, Test & Evaluation Conference, Sydney, October 2002.
- [NEMN2006] B. Nemnich: Der Sarbanes-Oxley Act und die Wirkung auf die IT. Diplomarbeit, Grin Verlag, 2006. Disponível apenas em alemão.
- [NUSE2001] B. Nuseibeh: Weaving the Software Development Process between Requirements and Architecture. Em: Proc. do Workshop STRAW-01 da ICSE2001, Toronto, maio de 2001.
- [OMG2013] Requirements Interchange Format (ReqIF). Object Management Group, Versão 1.1, 2013, disponível em <http://www.omg.org/spec/ReqIF/1.1/PDF/>. Última alteração em fevereiro de 2024.
- [PMI2021] PMI: Project Management Book of Knowledge (PMBOK). Project Management Institute, 7th Ed., 2021.
- [POHL1994] K. Pohl: The Three Dimension of Requirements Engineering: A Framework and its Application. Information Systems 3, 19 (junho de 1994), pp. 243–258.
- [POHL1996] K. Pohl: Process-Centered Requirements Engineering. John Wiley Research Science Press, 1996.
- [POHL2010] K. Pohl: Requirements Engineering – Fundamentals, Principles, Techniques. Springer, 2010.

- [PoBL2005] K. Pohl, G. Böckle e F. van der Linden: Software Product Line Engineering – Foundations, Principles, and Techniques. Springer, 2005.
- [PoRu2011] K. Pohl e Chris Rupp: Requirements Engineering Fundamentals. Rocky Nook, 2011.
- [RoRo2014] S. Robertson, J. Robertson: Mastering the Requirements Process – Getting Requirements Right. Addison–Wesley, 3a Edição, 2014.
- [RuSo2009] C. Rupp & die SOPHISTen: Requirements–Engineering und –Management, Hanser, 5. aktualisierte und erweiterte Auflage, 2009.
- [SEI1999] Carnegie Mellon SEI (1999): The Capability Maturity Model, Guidelines for Improving the Software Process. Addison–Wesley, 1999.
- [SEI2010] Carnegie Mellon SEI (2010): CMMI for Services, versão 1.3, Improving processes for providing better services, 2010.
- [SCHI2001] B. Schienmann: Kontinuierliches Anforderungsmanagement Prozesse – Techniken – Werkzeuge. Addison–Wesley, 2001.
- [ScHT2006] P.–Y. Schobbens, P. Heymans, e J.C. Trigaux: Feature Diagrams: A Survey and a Formal Semantics. Em: Anais da 14ª Conferência Internacional de Engenharia de Requisitos (RE'06), setembro de 2006, pp. 139–148.
- [SoSa1997] I. Sommerville e P. Sawyer: Requirements Engineering: A Good Practice Guide. John Wiley & Sons, 1997.
- [SuSc2013] J. Sutherland e K. Schwaber: Scrum Guide, julho de 2013, disponível em www.scrum.org. Última alteração em fevereiro de 2024.
- [USCO2002] Congresso dos EUA: Lei Sarbanes–Oxley. Washington, EUA, 107º Congresso dos Estados Unidos da América, 23.01.2002.
- [VANL2009] A. van Lamsweerde: Requirements Engineering – from System Goals to UML Models to Software Specifications. John Wiley and Sons, 2009.
- [WeLa1999] D. Weiss e C. Lai: Engenharia de linha de produtos de software: um processo de desenvolvimento de software baseado em família. Addison–Wesley, 1999.
- [WIEG2005] Karl Wiegers: Software Requirements, Microsoft Press Deutschland, 1. Auflage, 2005.
- [WiBe2013] K. Wiegers and J. Beatty: Software Requirements, 3a. Edição. Microsoft Press, 2013.
- [YOUN2004] R. Young: The Requirements Engineering Handbook, Artech House, Boston, 2004.
- [ZIEG1998] K. Ziegbein: Controlling. Kiehl Friedrich Verlag, 6. Auflage, 1998.