

## Unidade 1

### Definições de Requisitos de Software

#### Aula 1

##### Fundamentos da Engenharia de Requisitos

#### Fundamentos da engenharia de requisitos

##### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

###### Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer os fundamentos da engenharia de requisitos. Esse conteúdo é importante para a sua prática profissional, pois você compreenderá os requisitos de Software, a delimitação do escopo do sistema e a engenharia de requisitos na engenharia de software.

Prepare-se para essa jornada de conhecimento! Vamos lá!!!

#### Ponto de Partida

CaféSaúde é uma organização que comercializa cafés em pó ou em grãos selecionados e embalados à vácuo, classificados em Standard, Premium e High Special.

Sua equipe de pesquisadores optou pelo desenvolvimento de um software para o gerenciamento da cadeia produtiva do CaféSaúde. Aqui começa o seu papel de analista de requisitos para explicar aos gestores os principais fundamentos de requisitos de software.

Estudante, com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos. Porém, continue pesquisando e atualizando seu conhecimento

sobre os fundamentos desta disciplina para aplicar no seu cotidiano profissional.

Bons estudos!

## Vamos Começar!

Estudante, seja bem-vindo a este conteúdo, que é a base para qualquer software que se deseja desenvolver.

A Engenharia de Requisitos é a disciplina que envolve atividades destinadas ao conhecimento das necessidades de melhoria dos processos de negócios, quando a proposta é desenvolver uma solução em Sistemas de Informações. Tais conhecimentos consistem em identificação, documentação, gerenciamento e melhoria contínua dos requisitos de softwares.

## Compreensão de requisitos de software

Vejamos uma definição clássica, que esclarece definitivamente o que é requisito de software, conforme ISO/IEC/IEEE (2010):

- (1) Uma condição ou capacidade necessária por um usuário para resolver um problema ou alcançar um objetivo;
- (2) Uma condição ou capacidade que deve ser atingida ou possuída por um sistema ou componente de um sistema para satisfazer um contrato, padrão, norma, especificação ou outro documento formalmente imposto;
- (3) Uma representação documentada de uma condição ou capacidade como em (1) ou (2).

Os requisitos são identificados de acordo com o contexto da empresa, das pessoas, da sociedade, do país, enfim, conforme a situação na qual o software será utilizado. Basicamente, as partes que envolvem os requisitos, conforme ilustra a Figura 1, são: pessoas (stakeholders), processos e regras (de negócios e regulamentações), gerando um resultado e comportamento (que o software desempenhará).

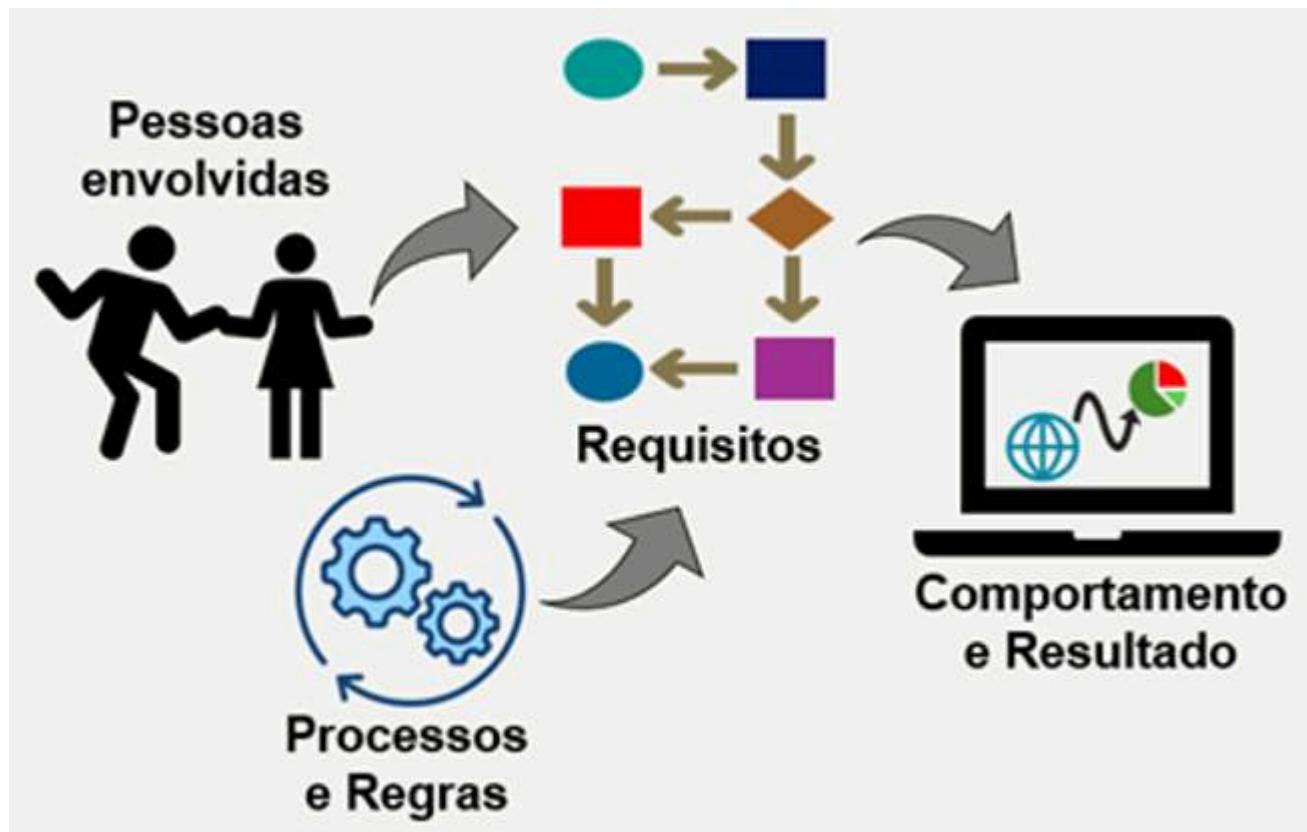


Figura 1 | Requisitos de software e as partes envolvidas

Em princípio, os requisitos são classificados com de negócios, uma vez que refletem as necessidades estabelecidas pelos processos realizados pela organização para gerar produtos ou serviços ofertados pelos negócios. Os requisitos de negócios são especificados com base nos procedimentos operacionais, táticos e estratégicos, além das necessidades em atender a uma regulamentação, sempre visando atender parcial ou totalmente os objetivos dos negócios de uma organização.

Uma vez que a empresa opta pelo desenvolvimento de software, com o auxílio dos analistas de sistemas ou de requisitos, começa a “corrida” pela especificação dos requisitos de software. Porém, segundo Pressman (p. 58, 2021):

É importante reconhecer duas realidades: (1) é impossível que os envolvidos descrevam um sistema completo antes de observarem o software operacional; e (2) é difícil que os envolvidos descrevam os requisitos de qualidade necessários para o software antes de vê-lo em ação.

Percebemos que o papel dos analistas e desenvolvedores passa a ser fundamental no reconhecimento dos requisitos que, por sua vez, serão adicionados, complementados e refinados à medida que os incrementos de software são criados. Enfim, a colaboração dos envolvidos (stakeholders) é imprescindível, embora saibamos das limitações, por meio de descrições sobre o que o sistema precisa fazer e em que precisa ser útil, em suas próprias palavras, talvez em uma história de usuário.

Por outro prisma, podemos reconhecer requisitos de software como uma parte intermediária, ilustrada na Figura 2, entre dois pontos: (1) os conceitos e desejos ou problemas e necessidades de mudanças nos processos de negócios; (2) as soluções em tecnologia da informação, por meio de softwares ou hardwares apropriados ao contexto da organização.

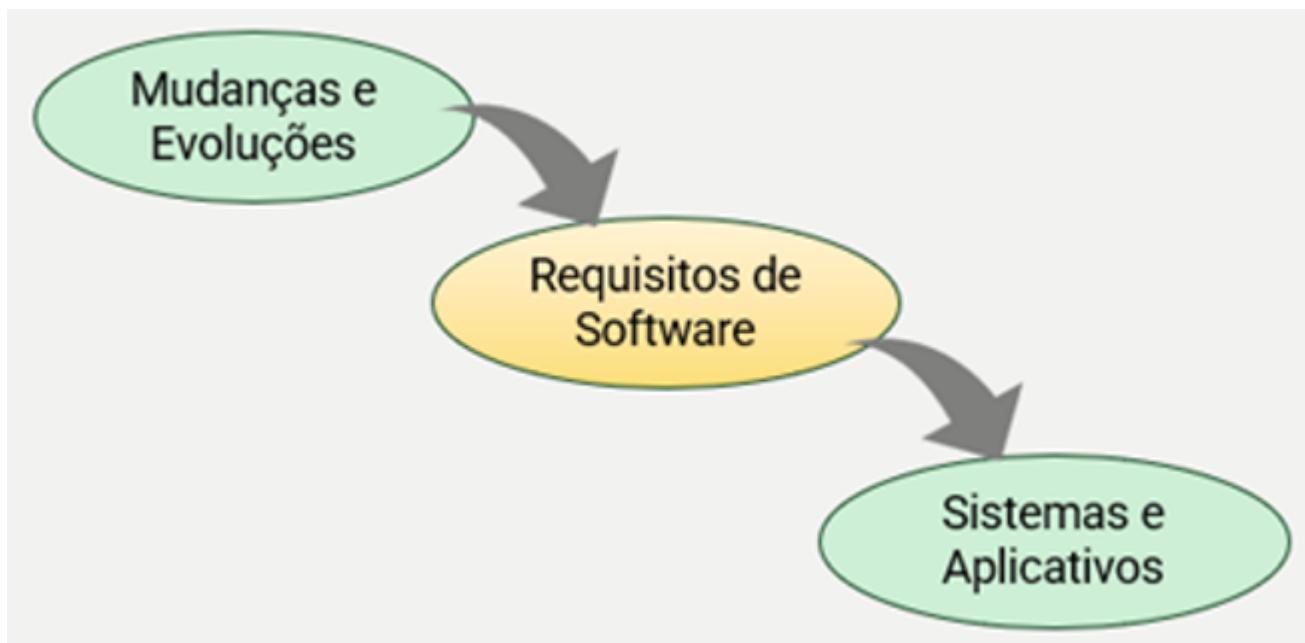


Figura 2 | Requisitos entre necessidades e soluções

Da documentação os requisitos envolvem várias pessoas, por um determinado período, que serão manipuladas e gerenciadas, e são fonte de diretrizes e premissas para o desenvolvimento de sistemas. Consequentemente, é imprescindível que sejam devidamente documentados por texto, diagramas, fluxogramas, vídeos, áudios ou algum artefato que represente o requisito. Pode-se combinar dois ou mais artefatos, um complementando o outro, para a compreensão e uso de todas as pessoas envolvidas, cada qual em seus papéis.

O documento de um requisito de software será referência para aprovação pelos usuários finais, pessoas que necessitam das funcionalidades. Na sequência, será referência para os arquitetos de software e demais membros do time de desenvolvimento de software, inclusive para os testadores das implementações. Ainda, será base para os usuários finais e clientes na validação entre o requisito e o software implementado.

Os requisitos podem ser classificados para facilitar o gerenciamento. Caso tenha requisitos de regulamentações (conformidade com instituições externas), podem ter prioridade maior que algumas funcionalidades complementares do processo. Dessa classificação, a mais comum é o requisito funcional e o requisito não funcional, que são, respectivamente: regras de negócios e funcionalidades que atendem diretamente ao operacional, tático ou estratégico da organização; ou aquele que estabelece alguma restrição ou premissa relacionada à qualidade ou segurança ou de comportamento do software.

## Delimitação de escopo

No contexto dos negócios, ainda vale o conhecimento de que todos os requisitos estão restritos a um escopo, cujo conceito determina quais serão as fronteiras com outros processos ou sistemas. Por exemplo: ao escolher o departamento de Finanças para o desenvolvimento de software, podemos escolher dois escopos: (1) Contas a Pagar contemplará Requisitos de fornecedores, títulos devidos, pagamentos de despesas, valores de saída do caixa; enquanto (2) Contas a Receber tratará de requisitos de clientes, títulos recebíveis, receitas, valores de entrada do caixa, entre outros.

Assim, a determinação do escopo (limites das funcionalidades e das responsabilidades dos stakeholders), a respeito de um sistema ou de um subsistema, implicará demasiadamente na gestão das atividades de definição dos Requisitos de Software. Consequentemente, nos esforços das demais tarefas do processo de desenvolvimento (análise, projeto, codificação, teste, validação, entre outros). Claramente, se consegue limitar, com maior precisão, os esforços do time de desenvolvimento, dos recursos financeiros no projeto de software e no prazo de entrega do sistema.

A determinação do limite do escopo de um sistema também limitará a definição dos requisitos e facilitará a escolha de prioridades do cliente. Para Pressman (2021), é imprescindível o escopo de um projeto, caso contrário, pode se tornar impossível prever como as prioridades do cliente vão se modificar ao longo do projeto.

Outro fator associado às causas mais frequentes dos problemas relacionados a requisitos, de acordo com Wiegers e Beatty (2013), é quando os objetivos de negócio, a visão e o escopo do projeto nunca foram definidos claramente.

É comum a prática do refinamento do escopo. Inicia-se com atividades de definição de requisitos com escopo em alto nível de granularidade, porém a intensificação dos esforços para a elicitação dos requisitos é aumentada gradualmente, quando ocorrem as delimitações do escopo do sistema (Reinehr, 2020).

Vários aspectos são relevantes para obter eficiência na definição e elicitação dos requisitos de softwares, conforme Leffingwel (2011), em ambientes ágeis de desenvolvimento:

- Especialmente, a definição de escopo e das prioridades deve ocorrer em momentos adequados, pelos stakeholders, fato que ocorrerá somente com a participação ativa desse público.
- Os membros da equipe precisam ter uma visão organizacional e enxergar a necessidade de integração com outros sistemas. Essa percepção é possível somente com os escopos claramente definidos.
- A definição do escopo facilitará a escolha dos envolvidos no projeto, desde o início, tanto de profissionais de produção e suporte, quanto dos profissionais que atuarão na manutenção do produto (caso seja um time diferente da equipe de desenvolvimento).

Siga em Frente...

## Engenharia de requisitos na engenharia de software

A engenharia de requisitos compõe o amplo espectro de técnicas e tarefas para o entendimento adequado dos requisitos de software.

Entendemos que o desenvolvimento de software depende de processos especificamente adequados para tal produção, conhecido como engenharia de software que, por sua vez, conta com a engenharia de requisitos para a definição das necessidades a serem implementadas pelo software, como ilustrado na Figura 3.

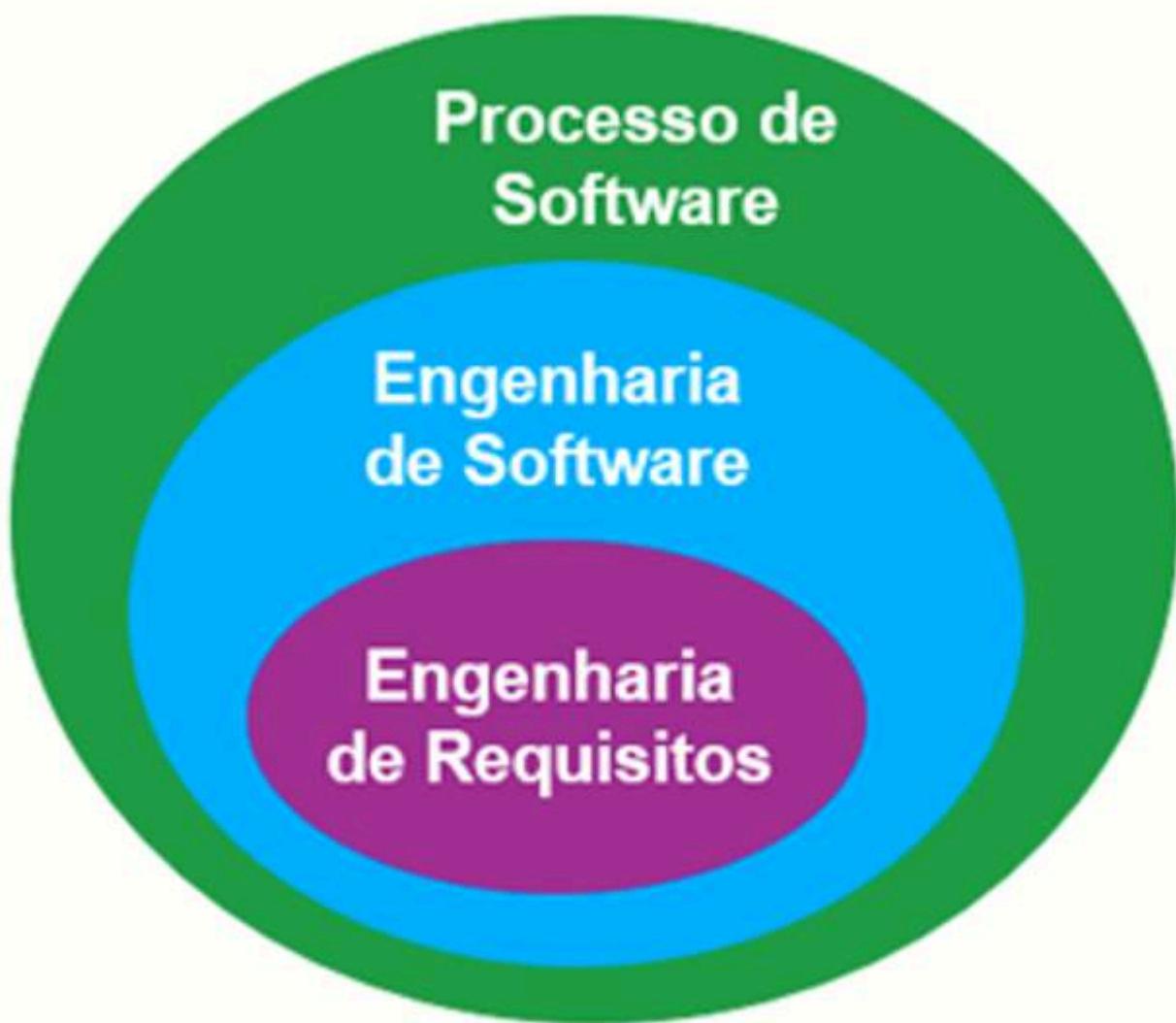


Figura 3 | Engenharia de requisitos em engenharia de software

Do ponto de vista do processo de software, a engenharia de requisitos é uma ação importante que se inicia durante a atividade de comunicação e continua, intensamente, na de modelagem. Porém, ela é acionada frequentemente em atividades de codificação, testes e, principalmente, em validação, quando poderá receber feedback de melhorias.

A engenharia de requisitos estabelece uma base sólida em projeto de software, diretamente relacionada à construção do software. Sem ela, o software produzido poderá não atender às necessidades do cliente. As tarefas pertencentes à engenharia de requisitos são realizadas de maneira iterativa à medida que a equipe do projeto e os envolvidos continuam compartilhando informações sobre as suas respectivas problemáticas ou evoluções.

A engenharia de software se estende a atividades de gerenciamento dos recursos, uso e criação de ferramentas, melhoria de técnicas, de forma otimizada, econômica, de qualidade e eficiente. Ou seja, ela é responsável pela construção e pelas atividades e recursos de suporte à produção do software.

Continue seus estudos, pesquisando e estudando sobre a importância de escopo de sistemas, engenharia de requisitos e engenharia de software.

Bons estudos!

## Vamos Exercitar?

Para desenvolver um software de gerenciamento da cadeia produtiva para a CaféSaúde é necessária a participação ativa dos principais usuários finais, gestores e especialistas de outras áreas. Dentre os usuários finais, temos: supervisor da produção do café, os captadores que visitam as fazendas produtoras de café, baristas internos e terceirizados, motoristas que efetuam as coletas do grão de café no campo, entre outros.

Os requisitos serão complementados à medida em que o software for construído e entregue. Porém, desde o início do projeto, os requisitos mínimos são necessários para especificar as funcionalidades da primeira entrega (versão) do software.

Os requisitos devem ser apresentados em formato documental, possuindo textos explicativos, diagramas esclarecendo o fluxo de atividades na produção do café em pó e, ainda, podem mostrar vídeos e áudios de situações reais. A linguagem ou os termos técnicos devem ser adequados para a compreensão de todos os envolvidos, incluindo os membros do time de TI (tecnologia da informação).

Quanto ao escopo, deve concentrar esforço e foco em rotinas operacionais de coleta e produção do café em pó, mesmo sabendo que esse subsistema será integrado ao de pagamento de fornecedores do grão de café, de gestão da produção por meio de equipamentos modernos. As primeiras implementações serão os requisitos de rotinas operacionais. Ao longo do processo de

desenvolvimento de software, os analistas eliciarão os requisitos de monitoramento da produção do café em pó, rotinas de qualificação do café em pó, e assim sucessivamente.

Bons estudos!

## Saiba mais

### I. Saiba mais sobre: definição de requisito

Leia o item 2.2, Definição de requisito, páginas 29 e 30.

[Engenharia de requisitos: software orientado ao negócio.](#)

### II. Saiba mais sobre: delimitação do escopo

Leia o item 2.4.2: Delimitar o escopo, páginas 34 e 35.

[Engenharia de requisitos: software orientado ao negócio.](#)

### III. Saiba mais sobre: requisitos e engenharia de requisitos

Leia a página 10.

[Gerenciamento de requisitos.](#)

## Referências

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/>. Acesso em: 5 nov. 2024.

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos**: software orientado ao negócio. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

## Aula 2

Elicitação de Requisitos

### Levantamento e elicitação de requisitos

#### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

##### Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer a elicitação de requisitos.

Esse conteúdo é importante para a sua prática profissional, pois compreenderá sobre a análise e refinamento do escopo, o papel do analista de sistemas e engenheiro de software na engenharia de requisitos e técnicas de levantamento de requisitos.

Prepare-se para essa jornada de conhecimento! Vamos lá!!!

### Ponto de Partida

Na primeira semana de atividades, com a comunicação do projeto de desenvolvimento de software na CaféSaúde, é chegada a hora para explanar como funciona a análise e refinamento do escopo, o papel do analista de sistemas e engenheiro de software na engenharia de requisitos e técnicas de levantamento de requisitos.

O seu desafio é demonstrar ao gestor da CaféSaúde as principais técnicas de levantamento de requisitos, como também explanar sobre as atividades do analista de sistemas e do engenheiro de software no projeto da CaféSaúde.

Estudante, com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos. Porém, continue pesquisando e atualizando seu conhecimento sobre elicitação de requisitos para aplicar no cotidiano profissional.

Bons estudos!

## Vamos Começar!

A elicitação de requisitos combina elementos da compreensão do problema, em um contexto de um negócio, cujo resultado será instrumento para a proposição da solução para esses problemas. Durante a elicitação, o analista de sistemas deverá concentrar o foco em capturar as reais necessidades (problema ou evolução nos negócios) dos usuários finais, clientes ou da sociedade; garantir que todos os stakeholders contribuam nessa etapa, assim obtendo uma visão completa e precisa do contexto ou escopo.

O levantamento de requisitos é uma etapa intensa e, muitas vezes, complexa, portanto, é valioso quando acontece com a abordagem colaborativa e orientada à equipe, com a presença dos usuários finais, clientes, gestores da organização, representantes do sistema, analista de sistemas, engenheiro de software e demais membros do time de TI.

## Análise e refinamento do escopo

Sabemos que o processo de levantamento de requisitos de software, acontece em um escopo definido, por algumas pessoas designadas para contribuir em um setor ou segmento de negócio, mas, ao longo do processo da análise do sistema, poderá haver ajustes nesses componentes.

Ao iniciar a análise de um sistema, sabemos que nem sempre teremos os dados completos a respeito do escopo definido, apesar de ser esse o objetivo do analista de sistemas. Ao longo das atividades da elicitação, os processos de negócios do escopo inicial são descobertos, podendo ser mais bem organizados e estruturados.

Assim, o refinamento do escopo acontece naturalmente, ao obter o conhecimento do contexto, após algumas iterações da análise do sistema, com as interações das pessoas envolvidas nesse contexto. Esse refinamento é necessário na prática para aumentar a taxa de sucesso de um projeto de desenvolvimento de software, com os seguintes objetivos:

- Planejar e controlar o cronograma de atividades: se existe maior complexidade para compreender os requisitos e propor as soluções, poderá alterar os recursos imaginados antes dessa descoberta; também, pode alterar o orçamento do projeto.
- Alinhamento com os objetivos de negócio: o software deverá operacionalizar e resultar soluções, conforme definido estrategicamente, para alcançar os objetivos de negócio de uma organização.
- Estruturação da implementação do software: com o refinamento do escopo, aumenta a compreensão dos processos e reduz as divergências ou ambiguidades, até mesmo possíveis conflitos; contribuindo fundamentalmente na elaboração do projeto de arquitetura, até a implementação dos componentes do software.

Vejamos um exemplo de contexto de escopo em uma organização. Inicialmente, seria apenas para eliciar requisitos do Controle Financeiros e Apresentação de Resultados com a participação do responsável pelo financeiro, ilustrado na Figura 1.

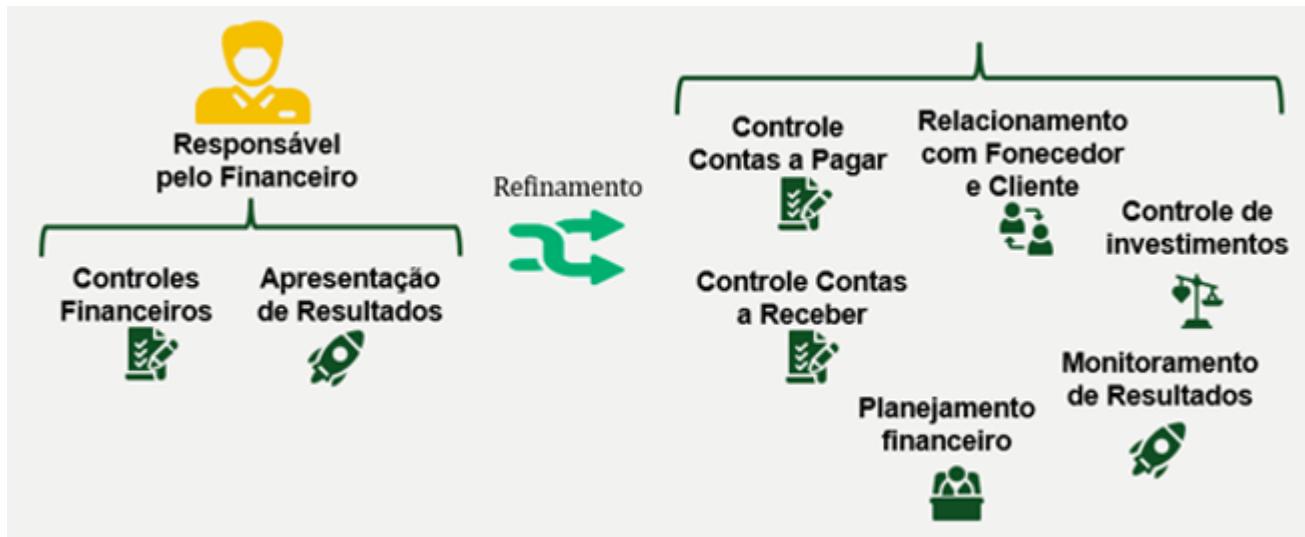


Figura 1 | Escopo em refinamento.

Após algumas iterações do levantamento de requisitos, conforme esse exemplo, o escopo sofreu refinamento, com detalhamento no processo: inicialmente eram compostos de dois processos e, após refinamentos, foram identificados seis subprocessos relacionados ao mesmo escopo. Seguindo o mesmo conceito de refinamento, podem ocorrer inúmeras composições de processos, subprocessos e atividades, até mesmo o desmembramento em outro escopo, conforme o contexto e cenário de negócio.

## Papel do analista de sistemas e do engenheiro de software na engenharia de requisitos

Os conhecimentos da engenharia de requisitos conduzem para obter requisitos de software de qualidade, com nível suficientemente capaz de gerar sistemas de informações conforme esperado pelos clientes, usuários finais e ao proprietário do software.

Tanto o analista de sistemas quanto o engenheiro de software são os responsáveis pela gestão das atividades relacionadas aos requisitos de software. Eles desempenham papéis complementares, juntos, contribuem para a definição, análise e gestão dos requisitos. Ambos são responsáveis pela avaliação e viabilidade dos Requisitos de Software, suficientes para a sua implementação.

Com focos distintos, o analista de sistemas é responsável em garantir que o sistema atenda às expectativas de negócio. Esse profissional tem a incumbência pelo entendimento aprofundado das necessidades dos usuários e do negócio e, ainda, pela tradução (especificação) dessas necessidades em requisitos claros e compreensíveis para o time de desenvolvimento. Pela visão da comunicação e interação, o analista de sistemas é o elo entre os stakeholders e o time de desenvolvimento (equipe de TI).

Com foco mais técnico, o engenheiro de software, é envolvido na engenharia de requisitos para garantir que os requisitos sejam tecnicamente viáveis e alinhados às melhores práticas de

desenvolvimento e arquitetura de software. A colaboração do engenheiro está em valorizar os recursos tecnológicos que podem agregar valor às soluções em software.

A engenharia de requisitos tem a responsabilidade de organizar o processo completo no tratamento do requisito de negócios ao requisito de software, desde o estudo de viabilidade até a validação. Segundo Sommerville (2011, p. 70), a construção dos requisitos passa por iterações que evoluem o conhecimento do requisito a partir da regra de negócios de uma organização até o requisito de sistemas. As principais interações acontecem de duas maneiras: (1) entre o analista de sistemas e os usuários finais ou cliente, em busca da compreensão das necessidades reais da organização; (2) entre o engenheiro de software e o analista de sistemas, em prol da construção do design do software.

Vejamos a Figura 2, ilustrando o ciclo espiral evolucionário desde o início da elicitação de requisitos até a documentação final do requisito de software. Destaca-se atividades com maior ênfase técnica para ampliar o conhecimento do time de tecnologia, com a atuação direta do engenheiro de software com as atividades de: estudo de viabilidade, prototipação, revisões e validação de requisito. Porém, em todas elas, ocorrerão em comum acordo do Analista de Sistemas; por conseguinte, contarão com participações pontuais pelos demais Stakeholders.

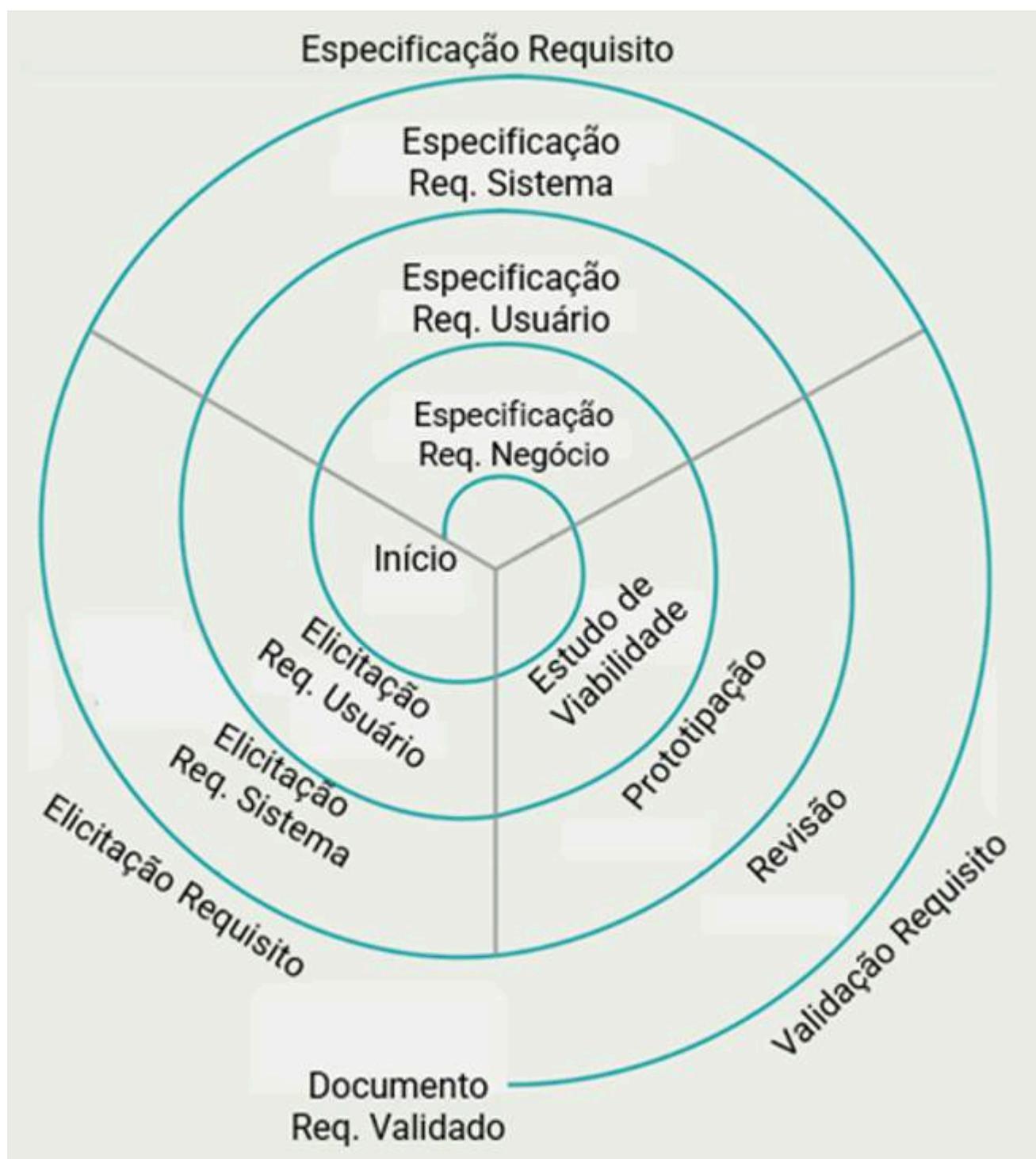


Figura 2 | Construindo requisitos de software em iterações. Fonte: adaptado de Sommerville (2011, p. 70).

Essas atividades da engenharia de requisitos são desempenhadas sob a responsabilidade do analista de sistemas e do engenheiro de software, ocorrendo de maneira intercalada, porém os dois profissionais podem participar em todas elas.

Por formação e preparação profissional, o analista de sistemas é o responsável pelas atividades iniciais, juntamente com os demais stakeholders, em elicitaciones e especificações; mantendo as demais atividades com a participação ativa também do engenheiro de software.

## Siga em Frente...

## Técnicas de levantamento de requisitos

Entende-se por técnicas de levantamento as maneiras ou ferramentas utilizadas para otimizar os recursos e o tempo para desempenhar atividades da análise de sistemas.

Percebemos que o requisito de software passa a ser especificado após várias iterações entre analista de sistemas e stakeholders, por meio de interações, nos ambientes (ou camadas) de negócios, de usuários e de sistemas, até obter a visão do software em requisitos.

Respectivamente, essas áreas compreendem:

Negócio: problemas, inovações ou evoluções em regras de negócio para mudar a estratégia ou objetivos da organização.

- **Usuário:** visão do usuário final (gestores ou especialistas) e suas experiências para executar atividades e decidir por melhores resultados.
- **Sistema:** relação com os processos operacionais, gerenciais e estratégicos, inclusive com as respectivas integrações com outros sistemas (fornecedores, clientes, sociedade, parceiros de negócios, entre outros).
- **Software:** perspectivas da informatização, fluxo de dados e informações, lógica dos processos, funcionalidades e manipulações de dados, segmentações ou agrupamento de dados ou processos, envio ou recebimento de dados entre aplicações e serviços digitais e as otimizações de tarefas até disponibilizar novas possibilidades inovadoras por meio da TI.

Percebida a evolução do conhecimento até o esperado requisitos de software validado, o analista de sistemas tem a responsabilidade de aplicar as técnicas de acordo com situações particulares. Conheçamos:

- **Técnica da análise de documentos:** analisar documentações utilizadas atualmente pelo usuário, possibilitando a identificação de informação relevante para definir os requisitos do sistema. Podemos citar alguns documentos: planos de negócio, literatura de marketing, acordos entre parceiros de negócio, pedidos de proposta (RFP – Request for Proposal), fluxos de processos atuais, modelos de dados lógicos, regras de negócio, documentação de software de aplicação, processos de negócio, casos de uso, outras especificações de requisitos, registros de problemas ou questões em aberto (issues), políticas, procedimentos e normas, como leis, códigos ou ordens etc.

Certifique se os documentos existentes foram previamente analisados e se estão de alguma maneira em uso, dando maior veracidade ao processo de negócio atual. Porém, alguns procedimentos podem ter sido atualizados na prática, enquanto a documentação continua como definido (sem a devida modificação); portanto, deve confirmar com os envolvidos.

Os documentos também podem delimitar mais claramente o escopo do problema em análise, inclusive permitindo maior aprofundamento do assunto objeto de análise e estudo.

- **Técnica do glossário:** artefato reconhecido como recurso importante na gestão do conhecimento, bastante útil na produção de software. Os termos são base fundamental para a comunicação e contribui para a elicitação de requisitos, por tratar das dimensões produto e processo, que identifica e define termos-chave para o domínio do problema, capturando o vocabulário das partes interessadas.

Toda vez que se deparar em termos com ambiguidade, ou conflitantes, é momento de analisar a palavra e defini-la por senso comum.

- **Técnica da etnografia:** também conhecido como técnica da observação em campo, pode complementar amplamente o que não ficou limitado aos meios verbais ou escritos. Cada vez mais usada em áreas diversas para obter o entendimento das interações entre pessoas com outras pessoas, instituições, máquinas (e sistemas de software), ou com o próprio ambiente. A observação é uma ação involuntária que qualquer aprendiz utiliza, reforçando a intenção do descobrimento ou da exploração dos processos.

A etnografia pode ser passiva, na qual o observador assiste ao procedimento sendo realizado e toma nota dos detalhes, sem interferir na atividade com perguntas; ou ativa, na qual se fazem intervenções durante o processo de observação.

Uma vantagem dessa técnica é a observação do trabalho de um grupo de pessoas, é possível identificar a interação entre elas e os pontos em comum ou diferentes.

Podemos elencar como vantagem, também: alguns requisitos implícitos (não narrados, ou não explanados), que podem ser registrados por meio da etnografia; por outro lado, como desvantagem, é necessário muito tempo de observação por existirem processos esporádicos ou sazonais, e restritos somente a procedimentos existentes; aqueles processos em criação ou em implantação não serão possíveis observar.

- **Técnica da entrevista:** é um diálogo formal ou informal, entre duas ou mais pessoas, trilhado por um conjunto de questões. Busca-se respostas pelo entrevistador junto aos entrevistados (fonte de informação).

Nessa técnica, o grau de confiança e sintonia entre as partes é o diferencial para obter um levantamento de requisitos completo e de sucesso.

Caso o entrevistado tenha interesse no projeto em questão, a colaboração será natural, rica em detalhes, porém, ao contrário, o entrevistado será menos engajado ou poderá sentir medo do que está por vir com um novo sistema.

O entrevistador experiente com os problemas do negócio objeto de estudo deve se preparar elaborando um roteiro, conforme ilustrado na Figura 3, na qual algumas perguntas inter-relacionadas serão aplicadas durante a entrevista; também pode haver perguntas abertas fora do roteiro.

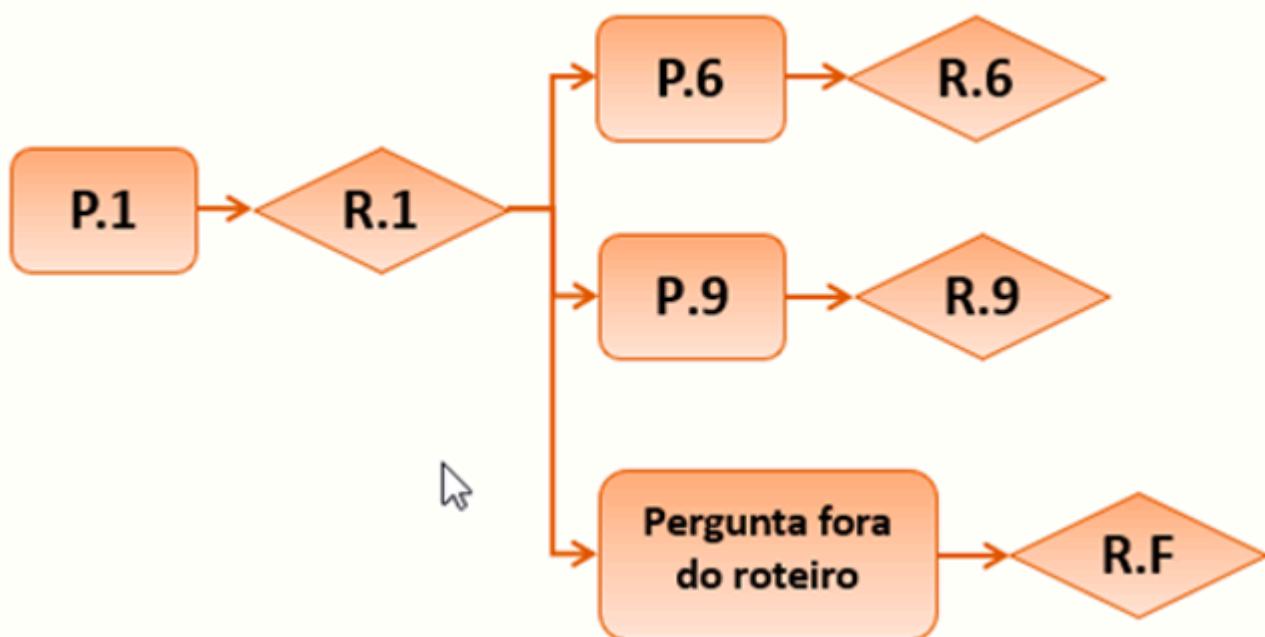


Figura 3 –| Roteiro de entrevista não estruturado. Fonte: adaptado de Vazquez (2016).

As entrevistas têm como desvantagem ser onerosas a um conjunto grande de partes interessadas. Porém, é válida por estabelecer um vínculo pessoal com o entrevistado podendo realizar novas intervenções para tirar dúvidas ou aprofundar o problema.

O analista ainda deve certificar se está entrevistando a pessoa de maior capacidade para repassar os conhecimentos do escopo; é comum gravar a entrevista, desde que tenha permissão expressa; planejar horários, tempo de entrevista, local, modo presencial ou remoto, participantes, tema, entre outros. Identificar o máximo de integrações possíveis com outros sistemas ou processos. Lembramos que a entrevista é uma técnica eficiente, porém deve ser complementada por outras.

- **Técnica de pesquisa:** também chamado de questionário, o objetivo é colher informações da experiência do pesquisado quanto ao problema em análise, para analisar após o questionário respondido. Muito utilizada para aplicar antes da entrevista, quando o entrevistador não tem o domínio do negócio ou do problema em estudo; com as respostas, facilitará a criação do roteiro das perguntas a serem realizadas na entrevista.

Quando aplicada à pesquisa após a entrevista, terá cunho de validação ou complementação de informações que não ficaram claras durante a entrevista.

A pesquisa é viável para projetos nos quais os entrevistados estão em vários lugares diferentes (cidades ou países distantes), o que tornaria custoso o trabalho pelo deslocamento em viagens.

Bons estudos!

## Vamos Exercitar?

Preparado para demonstrar uma situação possível da realidade na CaféSaúde aos gestores da área de identificação de grãos de café arábica de qualidade? Então, demonstre a sua capacidade de utilizar técnicas de levantamento de requisitos, como também explanar sobre as atividades do analista de sistemas e do engenheiro de software no projeto da CaféSaúde.

Para iniciar o levantamento de requisitos da CaféSaúde, é necessário escolher uma técnica (ou mais) e a preparação inicial.

Adotemos a entrevista para coletar as necessidades do sistema de mapeamento de produtores de café arábica de qualidade. Porém, devemos preparar um breve questionário para direcionar os agendamentos das entrevistas.

No questionário, é importante citar o objetivo pelo qual a pesquisa está sendo aplicada e em que a entrevista será realizada. Deve-se escolher algumas pessoas para serem pesquisadas, por indicação do gestor da área.

Vamos ao exemplo de um questionário aos colaboradores do segmento do café de qualidade, conforme Quadro 1.

<b>Processo: mapeamento de produtos de café arábica de qualidade</b>		Analista de sistemas: Hisa
Nome do entrevistado:		Contato:
1.	Qual é a sua responsabilidade no processo?	*espaço para resposta
2.	Quais documentos você utiliza para desempenhar suas atividades?	
3.	Quais períodos você poderá ser entrevistado?	
4.	Tem conhecimento em sistemas de informações?	

5.	Se respondeu sim no item 4. Descreva rapidamente o processo que utiliza no sistema.	
----	--	--

Quadro 1 | Questionário CaféSaúde.

A entrevista será realizada com os agentes de campo, com duração máxima de duas horas, mas podendo ser realizadas mais de uma entrevista para cada agente. O detalhamento dos processos de negócio e os requisitos de sistemas devem ser oferecidos durante a entrevista, para que o analista de sistema tenha fundamento para especificar os requisitos de negócio, de usuário e de sistemas.

Os requisitos de negócios serão acompanhados pelo engenheiro de software, Hasegawa, responsável pelo estudo de viabilidade. Após a especificação dos requisitos de usuários coletados pelo analista junto ao time de agentes de campo e ao supervisor do setor, o engenheiro elabora o protótipo para validar juntamente com os entrevistados.

Para qualquer dúvida que surja ao longo do período de levantamento de requisitos, o analista poderá submeter um questionário complementar ou solicitar nova entrevista para esclarecimentos.

Assim, sucessivamente, será gerado um documento com os requisitos de software que servirá de insumo para o time de desenvolvimento, juntamente com o engenheiro.

Estudante, com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos. Bons estudos!

## Saiba mais

I. **Saiba mais sobre: entrevistas**, páginas 71 e 72.

[Gerenciamento de requisitos](#).

II. **Saiba mais sobre: questionários**, página 73.

[Gerenciamento de requisitos](#).

III. **Saiba mais sobre: stakeholders**: influência sobre os requisitos, páginas 69 a 71.

[Gerenciamento de requisitos](#).

## Referências

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024.

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos**: software orientado ao negócio. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

## Aula 3

Domínio do Problema, Restrições e Premissas de Requisitos

### Domínio do problema, restrições e premissas de requisitos

#### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

#### Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante!

Nesta videoaula, você irá conhecer o domínio do problema, restrições e premissas de requisitos. Esse conteúdo é importante para a sua prática profissional, pois explicará o domínio do problema e os tipos de requisitos, processos de especificação de requisitos e restrições e premissas de requisitos.

Prepare-se para essa jornada de conhecimento! Vamos lá!!!

## Ponto de Partida

Estamos nas primeiras iterações das atividades do analista de sistemas desempenhando suas atividades da engenharia de requisitos, portanto, é importante apresentar aos gestores da CaféSaúde como será a especificação de requisitos.

O seu desafio é descrever o procedimento que será praticado no projeto da CaféSaúde para o sistema de rastreamento de café arábica nas propriedades do norte do Paraná, com o detalhamento dos tipos de requisitos, restrições e premissas.

Com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos. Porém, continue pesquisando e atualizando seu conhecimento sobre especificação, restrições e premissas de requisitos para aplicar no seu cotidiano profissional.

Bons estudos!

## Vamos Começar!

Estudante, bem-vindo à especificação de requisitos, um dos artefatos mais esperado após a árdua tarefa da elicitação de requisitos: Domínio do problema, especificação, restrições e premissas de requisitos!

O propósito é conceber um documento suficiente de conteúdo para futuras validações, porém essencial para as próximas etapas do desenvolvimento de software, análise, projeto, implementação e teste.

## Domínio do problema e os tipos de requisitos

Imagine você inserido nos negócios da produção de café, com plantações à vista do cafezal, aquelas filas de plantas verdes e frutos vermelhos brilhantes. Em instantes, você percebe a colheita acontecendo com os cuidados grão a grão, escolhidos e levados à secagem. Após algumas horas, começa a sentir o cheiro inconfundível de café torrado e moído invadindo todo espaço da propriedade. Agora, você já pode visualizar a passagem da água a 95°C, no pó de café (imagine, há pouco era um grão vermelho), sobe aquele aroma que estimula saborear o puro sabor da bebida.

Contextualizada a rotina da produção de café, temos o que podemos chamar de macro ambiente do qual extrairemos os processos informacionais para o desenvolvimento de um sistema, pelo qual podemos chamar de domínio do problema.

O domínio da aplicação é caracterizado pelo ambiente no qual ela está inserida. Ou seja, se o software será desenvolvido para controlar a colheita e beneficiamento do café, seu domínio é o campo de plantações, pessoas, máquinas, ferramentas, técnicas, recursos financeiros, armazém,

materiais, entre outros elementos, tudo o que deve ser considerados no momento da especificação de requisitos para a nova aplicação.

A partir da delimitação do domínio, o analista de sistemas e o engenheiro de software poderá identificar objetos, atributos e processos, stakeholders que serão fundamentais para o desenvolvimento de um software. Nesse cenário, serão construídas as funcionalidades do sistema, refletindo o que é praticado na rotina do domínio. É pelo domínio definido que saberemos delimitar o escopo do sistema, as integrações com outros sistemas, quais dados serão compartilhados, quais informações são relevantes para as tomadas de decisões.

Um analista de sistemas experiente no domínio do problema tem vantagens para compreender as necessidades do negócio, terá facilidade com os termos, jargões e conceitos. Com poucas explanações, será possível captar a real situação. Porém, lembremos que as soluções dadas anteriormente, em situações similares, podem ser insucessos em sua organização. Contudo, tais experiências serão extremamente benéficas para propor alternativas em sistemas.

A definição dos tipos de requisitos de negócio e de software são determinados ou influenciados pelo domínio do problema. Tanto requisitos funcionais quanto requisitos não funcionais estão intimamente associados ao domínio. O analista de sistemas especificará os requisitos que representarão o comportamento, o objetivo, as funcionalidades e os benefícios que o software proporcionará aos negócios ou ao domínio do problema. Vejamos como os requisitos podem ser mais bem compreendidos e gerenciados, após a sua classificação, conforme ilustra a Figura 1:



Figura 1 | Tipos de requisitos no domínio do problema.

- **Requisitos de domínio:** especificam necessidades que o próprio domínio estabelece como imprescindíveis e devem ser seguidas, tais como padrões e regulamentações. Naturalmente, os softwares devem implementar funcionalidades seguindo esse tipo de requisito.
- **Requisitos de interface:** a maioria dos sistemas fazem integração com outros, acessam dados de outras fontes, disponibilizam dados para outras organizações, portanto esses requisitos são aplicáveis nessas comunicações provendo maior abrangência e utilidade.
- **Requisitos funcionais:** compreende as funcionalidades essenciais do domínio, podem seguir técnicas e padrões conhecidos ou adaptados à cultura da organização. Especifica o que o sistema deve fazer, intimamente relacionado às regras de negócios do domínio.
- **Requisitos não funcionais:** são aspectos relacionados à qualidade do software que está sendo desenvolvido, atendem aos critérios de confiabilidade, desempenho, escalabilidade, usabilidade e segurança.

Em nosso cenário, produção de café, o cadastramento de lotes (talhões) da plantação de café, bem como as variedades das mudas plantadas, fazem parte dos requisitos funcionais.

## Processos de especificação de requisitos

Ao definir o contexto, cenário ou domínio do problema, o analista de sistemas tem ótimos motivos para especificar os requisitos de software com maior segurança. A descrição ou diagramação do requisito é uma tarefa técnica e requer habilidades específicas para identificação dos objetos, procedimentos, dados, estruturação, dependências, bem como o estado de cada objeto em função dos processos.

O processo de especificação pode ocorrer numa fase do projeto de um desenvolvimento de software, quando é escolhida a metodologia Waterfall, bem no início do ciclo de vida de sistemas. Ultimamente, com a adoção da metodologia Ágil, por muitas empresas e time de desenvolvimento, a especificação ocorre a cada iteração. Compreendendo a tarefa de coleta e análise dos problemas e mudanças, descrição ou diagramação do requisito e a validação do documento que será utilizado em processos posteriores, da engenharia de software.

## Especificação de requisitos

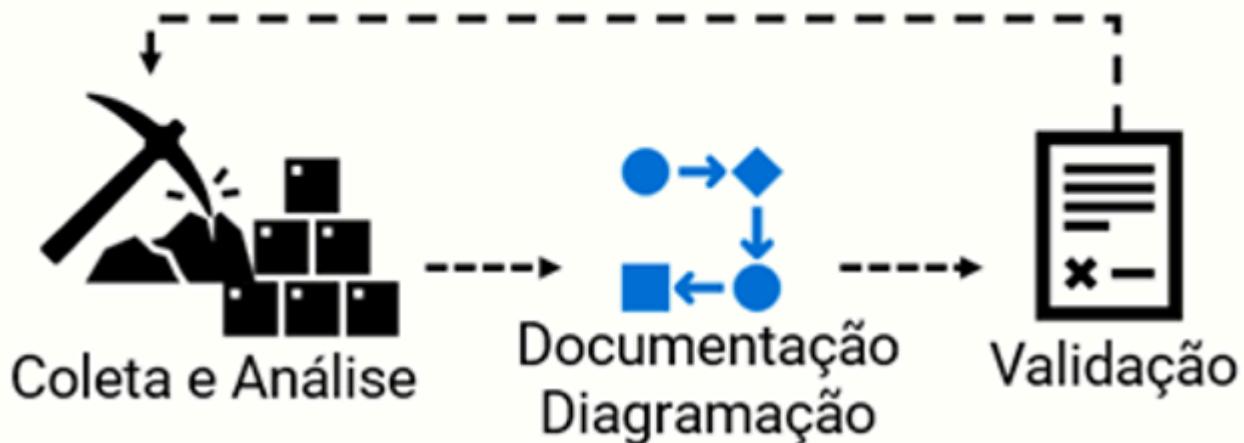


Figura 2 | Processo de especificação de requisitos.

As organizações têm facilidades adotando um ciclo curto do processo de desenvolvimento, ou seja, em algumas semanas, é possível especificar modificações nos requisitos, assim o sistema se torna aderente aos objetivos de negócio. A metodologia incremental oportuniza mudar o sistema, alinhando as estratégias de negócio, em velocidade adequada, demandadas pela dinâmica e competitividade organizacional pós-modernas.

Fazem parte da especificação do requisito as ferramentas para detalhamento, desde editores de textos até aplicativos de diagramação. A técnica utilizada para a especificação de requisitos não

deve alterar o objetivo da funcionalidade, ou das regras, nem mesmo deixar de contemplar o domínio do problema. Essas variações de técnicas são escolhidas pela facilidade de uso ou pela cultura organizacional.

Ao final da especificação de requisitos, é importante obter um documento com características que orientem os engenheiros de software nos projetos e os desenvolvedores nas implementações conforme esperado pelo cliente e pelos usuários finais. São estas as características:

- **Completude:** elementos que definem a funcionalidade, os aspectos não funcionais, abordam todo o domínio e as integrações, o fluxo dos procedimentos definitivo e/ou alternativos, ou de exceção, inclusive casos especiais.
- **Compreensibilidade:** com redação clara e sucinta, em linguagem adequada a todos os stakeholders, sem ambiguidades. Quando utilizar linguagem técnica deve existir explicações objetivas.
- **Viabilidade:** a funcionalidade deve ser tecnicamente viável sob a avaliação de recurso disponível ou orçado no projeto, tecnologicamente ao alcance do time de desenvolvimento. Porém, em algumas metodologias de desenvolvimento, pode-se prever um passo específico de estudo e viabilidades, para evitar que a especificação seja elaborada.
- **Verificabilidade e testabilidade:** seja possível verificar se foi implementado corretamente e deve conter critérios claros de aceitação.
- **Rastreabilidade:** que possa ser rastreado de volta ao seu objetivo de negócio ou necessidade específica. Possibilita compreender a justificativa da existência do requisito e facilita a análise de impacto no caso de alterações.

Um nível adequado na qualidade da especificação de requisitos evitará custos adicionais por esforços em resolver conflitos ou necessidade de reanálise dos requisitos. Logo, contribuirá com a entrega de valor ao cliente e aumentará a eficiência no desenvolvimento do software.

**Siga em Frente...**

## Restrições e premissas de requisitos

O analista de sistemas deve se prevenir com as restrições, pois são limitações impostas ao desenvolvimento do sistema. Nem sempre está claramente explicita uma condição pela qual saibamos que é uma restrição ao projeto ou até mesmo na operação do sistema. São fatores que devem ser respeitados durante o projeto e implementação e podem afetar o escopo e as soluções possíveis para o sistema.

Vejamos alguns exemplos de restrições que podem impactar em software:

- **Orçamento:** o valor total de gastos previstos não poderá passar de um teto, portanto são imitações de custos que podem afetar as funcionalidades e o alcance do projeto.
- **Prazo:** requisitos temporais que impõem prazos específicos para entrega do sistema.

- **Tecnologia:** tecnologias ou padrões específicos que são exigidos. Exemplo: para usar um banco de dados específico ou aderir a um framework existente.
- **Legislação e regulamentação:** exigências legais que o sistema deve atender, como normas de segurança de dados (LGPD), ou restrições relacionadas ao negócio. Ex: legislação da Anvisa (Agência Nacional de Vigilância Sanitária) que regulamentam os sistemas da área da saúde, alimento, cosmético etc.

As restrições devem ser identificadas logo no início do projeto e tratadas com rigor, durante todo tempo em projetos, pois têm impacto significativo nas decisões de arquitetura, design e implementação do sistema.

Enquanto as premissas são suposições sobre o projeto e o ambiente em que o sistema será desenvolvido e operado. Elas podem ser fatores externos que, se mudarem, podem afetar o sistema ou a viabilidade do projeto.

Consideremos alguns exemplos de premissas em desenvolvimento de software:

- **Disponibilidade de recursos:** a premissa de que uma equipe específica estará disponível para o desenvolvimento, em tempo hábil, que não afetará em custo, qualidade e prazo do projeto.
- **Acesso a tecnologias:** suposição de que o ambiente de desenvolvimento de software estará adequado para implementar e testar os novos equipamentos de integração acoplados em máquinas de homologação.
- **Infraestrutura:** à medida que as novas funcionalidades e requisições de IoT forem entregues ao cliente, o ambiente operacional do cliente estará apto a receber essas tecnologias.

As premissas também devem ser monitoradas e revisadas ao longo do projeto, pois qualquer mudança de requisitos pode afetar o escopo, o prazo, a estratégia ou o cronograma do desenvolvimento.

## Vamos Exercitar?

Na CaféSaúde, os requisitos devem estar especificados dando ênfase aos controles de status de cada produto colhido. À medida que o café (fruto vermelho) é colhido, deve prever a rastreabilidade pelas embalagens identificadas com o talhão (lote plantado), especificações das variedades (subclasse do café arábica), até a disponibilização do café (embalado a vácuo) em pó, pronto para coar.

Nessa fase de planejamento das atividades de levantamento de requisitos de software, os gestores devem prever o desembolso de recursos, por um período, até que seja implementado o software, ou seja, sem receber o software. Todavia, ele acompanhará todas as atividades, visualizando os documentos que estarão em construção, atualização e aprovação, por meio de ferramentas apropriadas e configuradas para tal finalidade.

Requisito de sistema #001:

- As informações de colheita devem ser apontadas assim que a embalagem receber a identificação da variedade e localização do talhão de colheita. Os requisitos devem estar em condições mínimas de qualidade, sob o aspecto da verificabilidade e testabilidade, especificando que o sistema deve processar até 1.000 transações por segundo. Na identificação deve constar os dados da pessoa que efetuou a colheita, o peso aferido logo que lacrar (lacre identificado) a embalagem, medição da umidade do produto, data e hora, localização (GPS) e temperatura do local.

Requisito de restrição #X001:

- Restrições, todos os safristas serão cadastrados antes do início do trabalho, sendo obrigatória a identificação biométrica, ciente dos processos controlados de LGDP e autorizando o compartilhamento de dados pessoais, conforme os processos internos da organização CaféSaúde.

Bons estudos!

## Saiba mais

I. **Saiba mais sobre: requisitos não funcionais**, lendo as páginas 35 a 39.

[Gerenciamento de requisitos](#).

II. **Saiba mais sobre: tipos de requisitos**

Leia sobre classificação de requisitos, classificação mais clássica, mas que não impede que existam sistemas de classificação diferentes. Nas páginas 47 a 52.

[Gerenciamento de requisitos](#).

III. **Saiba mais sobre: domínio do problema**

Leia o texto sobre a técnica do questionário, dando ênfase ao domínio do problema, na página 73.

KERR, Eduardo Santos (org.). [Gerenciamento de requisitos](#).

## Referências

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024.

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos**: software orientado ao negócio. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

## Aula 4

Negociação de Requisitos

### Negociação de requisitos

#### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer a negociação de requisitos. Esse conteúdo é importante para a sua prática profissional, pois, com ele, compreenderá o processo de negociação de requisitos, requisitos mal especificados: causas e efeitos e técnicas de negociação de requisitos. Prepare-se para essa jornada de conhecimento! Vamos lá!!!

### Ponto de Partida

Continuando a proposta de prática do aprendizado, nesta aula, apesar das inúmeras iterações em especificação de requisitos, o analista de sistemas deve submeter os requisitos da CaféSaúde pela negociação de requisitos.

O seu desafio é descrever o processo, objetivo e técnicas para identificação de requisitos mal especificados. Deve demonstrar aos gestores da CaféSaúde a importância da negociação de requisitos.

Com o aprendizado deste conteúdo, você se prepara para atuar em engenharia de requisitos, mas continue pesquisando e atualizando seu conhecimento sobre a área para aplicar no seu cotidiano profissional.

Bons estudos!

## Vamos Começar!

A negociação de requisitos é um passo importante para que cada stakeholder faça uma reavaliação das prioridades de seus requisitos e para que conheçam as necessidades dos demais; inclusive, é nesse passo que eles ficam sabendo das limitações técnicas ou de algum outro recurso que pode afetar diretamente o requisito solicitado.

## Processos de negociação de requisitos

Sabemos que os requisitos de software foram elaborados com base em regras de negócios e que muitos stakeholders participam nas atividades e elicitação dos requisitos, por questionário ou entrevista.

Portanto, é o momento de apresentar os requisitos aos stakeholders para trabalharem em conjunto a fim de estabelecer um consenso sobre quais requisitos serão implementados, como serão implementados e qual a prioridade entre eles. Assim, o analista de sistemas deve conduzir a apresentação para que as expectativas estejam alinhadas entre todos os stakeholders, explanando também as limitações que vão impactar nos requisitos.

Segundo Pressman (2021), quando todos os envolvidos (stakeholders) conquistam sistemas que satisfaçam suas necessidades, o analista de sistemas (time de desenvolvimento) também conquista o plano de desenvolvimento com prazos e orçamentos reais, conforme ilustrado na Figura 1, na qual ocorre o resultado de ganho mútuo (win-win) e acontece o processo de comunicação bidirecional chamado de handshake (aperto de mãos).

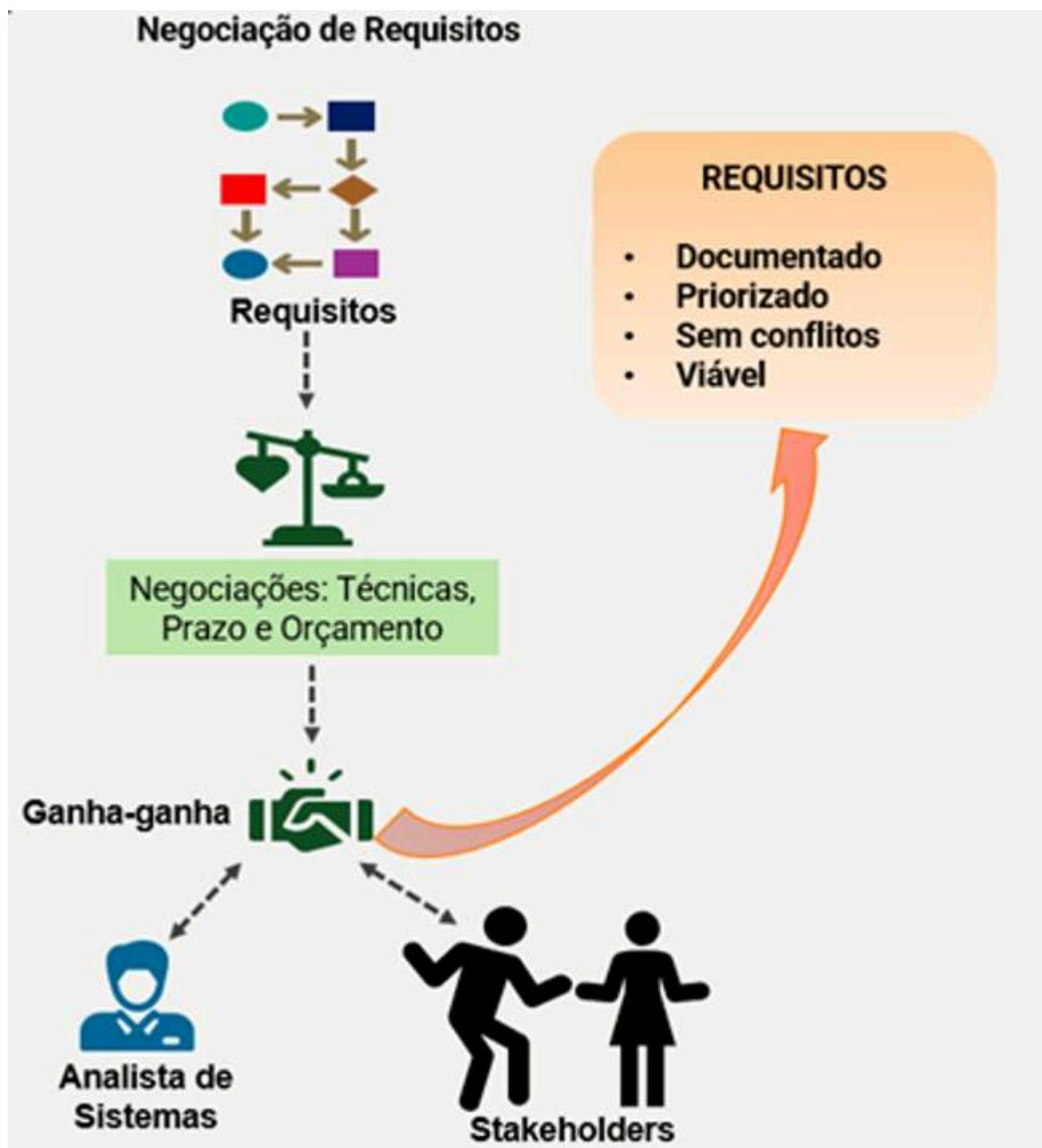


Figura 1 | Negociação de requisitos (cenário ideal).

Para o handshake, comprehende-se que tanto os stakeholders (cliente e usuários finais) quanto a equipe de software ficam satisfeitos. O analista de sistemas propôs soluções para os requisitos, descreveu seu impacto e comunicou seus objetivos para os representantes do cliente; eles examinaram e aceitaram as soluções propostas, por considerarem boas o suficiente.

Em uma negociação de requisitos, a identificação dos stakeholders é o primeiro passo: usuários finais, patrocinadores, clientes e membros da equipe de desenvolvimento. Assim, os passos da negociação podem seguir:

- **Definição das prioridades:** os requisitos levantados passam por uma análise de priorização, após identificar os requisitos essenciais, desejáveis e aqueles que podem ser postergados ou descartados. Eventualmente, podem ser usadas ferramentas como a análise MoSCoW (Must have, Should have, Could have, e Won't have) para definir essas prioridades.
- **Análise de viabilidade técnica e de impacto:** o time de desenvolvimento (incluindo o engenheiro de software) deve analisar os requisitos selecionados quanto à viabilidade e o impacto de cada requisito na arquitetura, no design e no cronograma do projeto. Assim, o time deve sugerir ajustes, quando necessário, que tornem os requisitos viáveis e sustentáveis dentro do projeto.
- **Identificação de conflitos:** a equipe de análise de sistemas identifica possíveis conflitos entre as três direções: os objetivos de negócio, as necessidades dos stakeholders e as restrições de projeto. São exemplos de conflitos comuns: funcionalidades incompatíveis, requisitos com altos custos de implementação, restrições técnicas e prazos conflitantes.
- **Resolução de conflitos:** deve buscar o entendimento das restrições e da proposta de soluções alternativas, podendo ser a redefinição dos requisitos, os ajustes no escopo ou a priorização de requisitos de acordo com o prazo e o orçamento. Para facilitar o entendimento dessa fase, as técnicas de prototipagem e simulação podem ser úteis para que os stakeholders visualizem e avaliem alternativas.
- **Documentação do acordo de requisitos:** os requisitos finais são documentados em uma especificação de requisitos de software, na qual cada requisito deve estar claramente descrito, com critérios de aceitação (com justificativas) e a sua prioridade. Esse passará a ser o documento formal dos requisitos negociados; em alguns casos, servirá de contrato entre stakeholders e time de desenvolvimento.
- **Gestão de mudanças:** é comum ocorrerem modificações nos requisitos ao longo do ciclo de vida do projeto, portanto deve prever novas renegociações de requisitos por necessidades do negócio, alterações tecnológicas ou descobertas técnicas. Por isso, o processo de negociação é considerado contínuo, garantindo que qualquer novo requisito passe pelo processo de análise, priorização e impacto.

## Requisitos mal especificados: causas e efeitos

São considerados requisitos mal especificados aqueles requisitos que apresentam ambiguidades, inconsistências ou incompletudes, tornando-os difíceis de interpretar e implementar. Para evitar insucesso, falhas no software, atrasos ou alto custo em projetos de softwares, os requisitos devem estar especificados adequadamente à compreensão de todos os envolvidos (representantes do cliente e time de desenvolvimento).

Podemos elencar algumas causas para requisitos estarem mal especificados:

1. Comunicação ineficaz entre stakeholders: faltou clareza na explanação dos stakeholders devido a:

- Diferentes níveis de conhecimento técnico entre stakeholders e time de desenvolvedores.
- Diferença de expectativas sobre o escopo do sistema.

2. Falta de entendimento do domínio do problema: se o time de desenvolvimento não compreender o domínio do problema, acabam definindo requisitos que não correspondem às necessidades reais do cliente. Podendo faltar atenção aos processos específicos do cliente, normas regulatórias ou contextos operacionais.
3. Pressão por prazos apertados: se há pressão para finalizar a fase de definição de requisitos e iniciar o desenvolvimento;
4. Ambiguidade na linguagem usada: o uso de linguagem ambígua ou subjetiva na documentação dos requisitos, como "rápido", "fácil de usar" ou "seguro", gera múltiplas interpretações. A escrita deve possuir especificidade para a compreensão do que realmente é esperado pelo sistema.
5. Mudanças constantes e falta de controle de versão: as mudanças frequentes nos requisitos, sem um processo formal de controle, podem levar à perda de informações ou à incorporação de requisitos obsoletos.

Vamos compreender alguns efeitos dos requisitos estarem mal especificados:

- 1. Retrabalho e aumento de custos:** requisitos mal especificados levam a interpretações erradas, resultando na necessidade de desenvolver novamente. Isso aumenta o tempo e o custo do projeto, impactando o orçamento planejado e, em alguns casos, inviabilizando o projeto.
- 2. Atrasos no projeto:** a necessidade de rever e corrigir requisitos durante o desenvolvimento causa atrasos, pois a equipe não possui outros membros para ajustar a implementação causados pelos requisitos mal especificados.
- 3. Insatisfação dos stakeholders:** atrasos, retrabalho, ou falhas no software, cria grande insatisfação dos stakeholders, inclusive abala a confiança do projeto como um todo.
- 4. Riscos à qualidade e à usabilidade do sistema:** imprecisão na implementação pode prejudicar na qualidade em geral do software ou afetar a usabilidade da aplicação. Especialmente prejudicial em sistemas críticos, nos quais a falta de qualidade pode resultar em falhas graves ou impactos significativos no negócio.
- 5. Efeito cascata na manutenção e evolução:** os requisitos mal especificados podem ficar sem os ajustes necessários, causando maior quantidade de manutenção no software, menores prazos para implementação, maior retrabalho em análise, projeto e implementação. Inclusive, dificulta a adição de novas funcionalidades ou adaptação do sistema a novas demandas.

## Siga em Frente...

## Técnicas de negociação de requisitos

A melhor estratégia para a negociação de requisitos é a abordagem colaborativa que ajudam a alcançar o consenso entre stakeholders, considerando os objetivos de negócio da organização, também, priorizar demandas e ajustar o escopo de acordo com as limitações e objetivos do projeto.

Para facilitar as decisões e o planejamento da implementação, podem utilizar técnicas que analisam o impacto e a importância para o projeto. Vejamos algumas:

**1. Análise MoSCoW (Must have, Should have, Could have, Won't have):** que faz a classificação dos requisitos em quatro categorias:

- **Must Have (Essenciais):** são os requisitos críticos para o sistema, considerados os fundamentais para o fluxo do processo que proporcionam resultados organizacionais.
- **Should Have (Importantes):** os requisitos importantes, não essenciais, porém devem ser desenvolvidos logo após a estabilização dos requisitos essenciais.
- **Could Have (Desejáveis):** aqueles requisitos que agregam valor, mas podem ser postergados. Eles auxiliam na execução dos requisitos importantes e essenciais.
- **Won't Have (Fora do Escopo):** são requisitos que não serão implementados nesse projeto.

**2. Técnica de prototipagem:** os protótipos ajudam os stakeholders a visualizar as funcionalidades e a interagir com uma versão preliminar do sistema; proporcionando maior segurança sobre a implantação, inclusive permite efetuar ajustes nos requisitos. A prototipagem é efetiva na redução da ambiguidade.

**3. Workshop de requisitos:** organizados entre stakeholders e equipe técnica para discutir, priorizar e negociar requisitos em conjunto. Com estratégia colaborativa, permite o alinhamento entre as expectativas dos stakeholders e a estratégia do time de desenvolvimento. O brainstorming é utilizado para estimular a comunicação e a geração de ideias.

**4. Matriz de priorização de requisitos:** cada requisito é classificado com base em critérios como importância para o negócio, urgência, valor agregado e complexidade de implementação. A matriz organiza os requisitos visualmente e facilita a tomada de decisão sobre quais devem ser abordados primeiro.

**5. Técnica de escalonamento de priorização:** após a distribuição, pelos stakeholders, de pontos ou valores a cada requisito com base em sua importância. Os requisitos com maior pontuação são priorizados.

Essas técnicas de negociação de requisitos visam o equilíbrio das necessidades dos stakeholders com as limitações do projeto, auxiliando na priorização e no consenso. Elas podem ser aplicadas, conforme necessário ou em combinação de duas ou mais técnicas.

## Vamos Exercitar?

Na CaféSaúde, os requisitos devem estar especificados adequadamente para os procedimentos de coleta das informações de campo durante a colheita. Imaginem um gestor do nível tático da empresa analisando a performance da colheita na safra 2025/26, comparando à safra 2024/25. De repente, ele percebe que falta o dado da umidade do grão colhido e ensacado ainda quando está em campo.

A primeira reação do gestor foi um pedido de correção no sistema, indicando a falta de implementação do dado obrigatório na fase da colheita. Quando o analista de sistema analisou essa solicitação de correção, ele verificou que não se trata de uma correção no software, pois, na especificação do requisito, não previa o input desse dado.

Essa situação inusitada pode ocorrer naturalmente por falha ou inexistência da fase de negociação de requisitos. Poderia ter sido verificada essa falha na documentação do requisito quando da análise de viabilidade do requisito por requerer um aparelho de medição da umidade.

Durante a identificação de conflitos o analista de sistemas identificaria possíveis conflitos entre as três direções: os objetivos de negócio, as necessidades dos stakeholders e as restrições de projeto. Vejamos alguns exemplos:

- **Dos objetivos de negócio:** um dos requisitos deve estar prevendo que as informações sejam guardadas por cinco anos, no mínimo, para gerar um histórico de ocorrências. Se a CaféSaúde tem metas de melhoramento da bebida, evidentemente os dados históricos proporcionarão melhores resultados.
- **Das necessidades dos stakeholders:** as prioridades são para requisitos que geram relatórios em lugar dos requisitos que validam os dados de integração com outros sistemas. Os gestores necessitam de dados válidos para serem analisados, pois a tomada de decisão deve ocorrer com base em dados verdadeiros e reais.
- **Das restrições de projeto:** as funcionalidades que estavam previstas para serem desenvolvidas e entregues até o final do plantio de café estão sendo programadas para o final da colheita. As funcionalidades de manejo da cultura do café devem ficar prontas até o final do plantio para coletar dados e gerenciar o manejo visando eficiência no gerenciamento da lavoura.

Com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos, com foco nas negociações de requisitos.

## Saiba mais

I. **Saiba mais sobre: negociação**, leia o item 7.6, Negociação de requisitos, página 122.

[Engenharia de software](#).

II. **Saiba mais sobre: colaboração**. Leia o item 7.2.3, Trabalho em busca da colaboração, página 108.

[Engenharia de software](#).

III. **Saiba mais sobre: qualidade em requisitos**. Leia o item 15.2, Qualidade de software, páginas 312 a 313.

## Gerenciamento de requisitos.

IV. **Saiba mais sobre: validade de requisitos.** Leia o texto sobre Gestão da Qualidade, página 479.

## Princípios de Sistemas de Informação.

V. **Saiba mais sobre: defeitos em software.** Leia o item 3.3.2, Uma das principais causas de defeitos em software, páginas 62 a 64.

## Engenharia de requisitos: software orientado ao negócio.

## Referências

KERR, E. S. (org.). **Gerenciamento de requisitos.** São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software.** 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos.** Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos:** software orientado ao negócio. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

## Aula 5

Encerramento da Unidade

## Videoaula de Encerramento

**Este conteúdo é um vídeo!**

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

**Dica para você**

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer as definições de requisitos de Software. Este conteúdo é importante para a sua prática profissional, pois você conhecerá os fundamentos de requisitos de software e como levantar requisitos e compreenderá o processo de especificação e de negociação de requisitos. Prepare-se para esta jornada de conhecimento! Vamos lá!!!

## Ponto de Chegada

Olá, estudante! Para desenvolver a competência desta unidade, que é:

- Conhecer os fundamentos de requisitos de software.
- Explicar como efetuar o levantamento e elicitação de requisitos de software.
- Demonstrar o processo de especificação, restrições e premissas de requisitos.
- Explicar o processo de negociação de requisitos.

Você deverá primeiramente conhecer os conceitos fundamentais de iniciação de um projeto de desenvolvimento de software, envolvimento das pessoas interessadas no sistema (stakeholders) e as responsabilidades do analista de sistemas.

## É Hora de Praticar!



### Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Para evitar que os requisitos causem conflitos, a especificação deve estar consistente e completa, sempre solucionando um problema da organização, ou esteja proporcionando resultados que elevarão a empresa a estágios melhores.

Vamos compreender como os gestores da CaféSaúde podem conduzir as atividades do levantamento de requisitos para o desenvolvimento de software. Em função do grande volume de dados a serem coletados no campo, em uma área extensa, mais de quinhentos hectares de plantio de café, para o gerenciamento da colheita.

Os gestores precisam conhecer melhor as responsabilidades do analista de sistemas e do engenheiro de software.

Como você poderia orientar os gestores a controlar as atividades do levantamento de requisitos na CaféSaúde?

Está apto para explicar os fundamentos de requisitos de software e como efetuar o levantamento e elicitação de requisitos de software?

Você consegue explicar o processo de especificação, restrições e premissas de requisitos e o processo de negociação de requisitos?

Compreendamos as atividades de cada iteração na engenharia de requisitos, na fase de levantamento e elicitação de requisitos até a apresentação do documento com os requisitos validados.

A maior parte do tempo, as pesquisas em TI para solução do problema acontecem em paralelo ao processo de levantamento e elicitação de requisitos. Enquanto os analistas de sistemas estão conhecendo os processos da fazenda, os engenheiros de software estão pesquisando alternativas em TI (Tecnologia da Informação) que possam suprir tais dificuldades atuais. Conforme ilustra a Figura 1, inicia-se pelos conhecimentos dos requisitos de negócio e os problemas apresentados pelos stakeholders.



Figura 1 | Ciclo dos requisitos.

Após a identificação dos requisitos de negócios, principalmente pela descoberta do problema, o analista continua o processo de especificação dos requisitos de usuário, subsequentemente, com o requisito de sistemas. Nessas iterações, é comum que os engenheiros de software apresentem alternativas por experiência em TI para a solução do problema.

Assim, o último procedimento trata-se do requisito de software, juntamente com as soluções dos problemas encontrados, com a participação do engenheiro de software, desenvolvido com elementos de TI.

Estrategicamente, a CaféSaúde prioriza o sistema de colheita do café. Você deve apresentar os requisitos aos gestores, sendo eles:

**Requisito de negócio:** a colheita deve ser controlada pela região (localização) e demais dados relacionados à colheita, resultando na produtividade e qualidade alcançada na safra. Dentre os problemas, o principal é como coletar os dados da colheita, de maneira segura e confiável, prevendo chuva e vento, regiões remotas, entre outros fatores que dificultam a ação humana.

**Requisito de usuário:** os operadores de máquinas e demais recursos utilizados nas colheitas serão os usuários finais do sistema, portanto a aplicação deve facilitar a disponibilização de comandos de fácil compreensão e uso no campo, sob sol e chuva.

Requisito de sistemas: as transações eletrônicas devem transmitir os dados em tempo real para a base do servidor situado em nuvem, sempre devolvendo respostas ao usuário pelo sucesso da transferência e conclusão das operações.

Requisito de software: as operações de entrada de dados da colheita devem ser desenvolvidas em interface com uma quantidade reduzida de comandos, apenas os essenciais, a fim de facilitar o aprendizado para operar a aplicação. Por meio dos recursos de GPS, entre outros sensores, deve apresentar o mínimo de comandos para coleta de dados necessários da colheita. A qualidade da documentação dos requisitos deve ser apresentada com elementos que possam responder aos questionamentos, segundo Stair (2021, p. 479):

- O requisito descreve algo realmente necessário para o cliente?
- O requisito está definido corretamente?
- O requisito é consistente com outros requisitos?
- O requisito está completamente definido?
- O requisito é verificável (testável)?
- O requisito pode ser rastreado até a necessidade do usuário?

Ao final dessa fase da engenharia de requisitos, você poderá efetuar a negociação de requisitos: um processo colaborativo e iterativo que envolve comunicação, priorização e alinhamento entre os stakeholders. Ação que permite o envolvimento maior do time e que possa desenvolver um sistema que atenda às necessidades de negócio e expectativas dos usuários, ao mesmo tempo em que considera as limitações práticas de implementação.

Dentre os benefícios do processo de negociação de requisitos, estão:

- alinhamento de expectativas: auxiliar a alinhar as expectativas dos stakeholders e do time de desenvolvimento, reduzindo conflitos e promovendo uma compreensão mútua.
- Redução de riscos: identificar e tratar potenciais problemas e conflitos logo no início, reduzindo o risco de falhas, retrabalho e estouro de orçamento.
- Foco nos objetivos do negócio: assegurar que os requisitos atendam às necessidades reais do negócio e dos usuários, priorizando entre os mais importantes e os secundários.
- Maior controle de escopo: garantir que o escopo do projeto seja mais bem definido e controlado, evitando aumentos desnecessários (scope creep).

O objetivo deste estudo de caso foi facilitar o entendimento da elicição e especificação de requisitos, em seus diversos estágios da especificidade técnica da engenharia de requisitos. Continue aprimorando a sua habilidade em definição de requisitos de software, atualizando-se sobre novas técnicas e abordagens.

Bons estudos!

Acesse o infográfico utilizando o QRCode por qualquer dispositivo para compreender a extensão do conteúdo desta unidade da disciplina de engenharia de requisitos.

**ENGENHARIA DE REQUISITOS**

prof. Marco Hisatomi

[https://bit.ly/engreq\\_info1\\_definicoesRequisitos](https://bit.ly/engreq_info1_definicoesRequisitos)

**DEFINIÇÕES DE REQUISITOS DE SOFTWARE – UN1**

Compreender a delimitação de escopo. Compreender e reconhecer os principais requisitos de softwares. Reconhecer o domínio do problema, restrições e premissas de requisitos.

Compreender a negociação de requisitos.

---

**Fundamentos da Engenharia de Requisitos**



Inicie pelos Fundamentos de Requisitos de Software, em seguida conheça sobre os requisitos de Software, delimitação do escopo do sistema e a Engenharia de Requisitos na Engenharia de Software.

**Levantamento e Elicitação de Requisitos**



Conheça a Elicitação de Requisitos, em seguida compreenda sobre a Análise e Refinamento do Escopo, o Papel do Analista de Sistemas e Engenheiro de Software na Engenharia de Requisitos e Técnicas de Levantamento de Requisitos.

**Domínio do Problema, Restrições e Premissas de Requisitos**



Conheça o Domínio do Problema, Restrições e Premissas de Requisitos, em seguida compreenderá sobre os Tipos de requisitos no domínio do problema, Processos de Especificação de Requisitos e Restrições e Premissas de Requisitos.

**Negociação de Requisitos**



Conheça a Negociação de Requisitos, em seguida compreenderá o Processos de Negociação de Requisitos, Requisitos Mal Especificados: Causas e Efeitos e Técnicas de Negociação de Requisitos



**Avalie!**

[https://bit.ly/engreq\\_aval\\_un1](https://bit.ly/engreq_aval_un1)

VOCÊ APRENDE MAIS QUANDO ESTUDA DENTRO DO SEU ESTILO

Figura 1 | Engenharia de requisitos. Fonte: elaborada pelo autor.

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos: software orientado ao negócio**. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

## Unidade 2

### Classificação de Requisitos de Software

#### Aula 1

##### Classificação de Requisitos

#### Classificação de requisitos

##### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer a classificação de requisitos. Este conteúdo é importante para a sua prática profissional, pois explica sobre os processos de classificação de requisitos, requisitos funcionais e requisitos não funcionais. Prepare-se para essa jornada de conhecimento! Vamos lá!!!

## Ponto de Partida

Os gestores da CaféSaúde estão empenhados no desenvolvimento do sistema de gerenciamento da cadeia produtiva de café Standard, Premium e High Special. Para tanto, você ficará responsável por explicar a eles como se aplicará a classificação de requisitos, seus processos, a classificação de requisitos funcionais e não funcionais.

Com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos. Porém, continue pesquisando e atualizando seu conhecimento sobre os fundamentos da classificação de requisitos para aplicar no cotidiano profissional.

Bons estudos!

## Vamos Começar!

Estudante, bem-vindo ao conteúdo do processo de classificação de requisitos!

Para organizar, categorizar e priorizar os requisitos coletados, a fim de facilitar o entendimento e orientar o desenvolvimento do sistema, usamos a classificação de requisitos.

A classificação dos requisitos proporciona ao time de desenvolvimento e aos stakeholders melhor entendimento das necessidades críticas do sistema e determina como atendê-las, além de auxiliar no gerenciamento de mudanças e na alocação de recursos.

No desenvolvimento de software devemos primar pela entrega de funcionalidades eficientes e úteis, derivadas dos requisitos de negócios definidos pelos stakeholders. Existe um longo caminho para chegar a esse objetivo, que passa pela classificação de requisitos, uma das ações primordiais para organizar e priorizar os requisitos.

Pela classificação de requisitos, é possível obter resultados importantes para a engenharia de requisitos e para a engenharia de software:

- **Comunicação:** garantir que os stakeholders compartilhem uma compreensão comum dos requisitos e suas classificações.
- **Estruturação:** com a identificação segura, possibilita a rastreabilidade e visualização da função de cada requisito.
- **Organização:** nomenclatura adequada de cada requisito, bem como a descrição completa e artefatos relacionados ao requisito.
- **Priorização:** determinar aqueles requisitos que devem ser desenvolvidos em sequência específica para atender às necessidades do negócio.

Vejamos as categorias de requisitos, segundo SWEBOK® GUIDE V4.0 (2024).



Figura 1 | Categorias de requisitos de software. Fonte: adaptado de SWEBOK® GUIDE V4.0 (2024, p. 46).

A classificação de requisitos pode ser realizada em várias dimensões de categorias:

- **Os requisitos de software (produto):** especificam a forma, a função esperada do software (pode ser uma modificação).
- **Os requisitos do projeto de software (requisitos de processo):** restringem ao projeto como construir o software. Geralmente, restringem custos, cronograma e/ou pessoal, podem restringir aspectos como ambientes de teste, migração de dados, treinamento de usuários e manutenção.
- **Os requisitos funcionais:** especificam comportamentos observáveis que o software deve fornecer, que podem ser especificadas em forma de políticas a serem aplicadas e processos a serem executados, como processamento de dados, relatórios e fluxos de trabalho.
- **Requisitos não funcionais:** restrições tecnológicas: obrigam, ou proíbem, a utilização de tecnologias de automatização específicas ou de infraestruturas definidas, tais como segurança, usabilidade, disponibilidade, entre outras.
- **Requisitos não funcionais:** restrições de qualidade de serviço: especificam níveis de desempenho aceitáveis que uma solução automatizada deve exibir. Exemplos são tempo de resposta, taxa de transferência, precisão, confiabilidade e escalabilidade.

Alguns autores ainda incluem uma classificação que obriga os analistas de sistemas e demais stakeholders a pesquisar e optar em atender aos requisitos de domínio que estabelecem características específicas do contexto ou domínio do sistema, como regulamentos legais ou normas da indústria que precisam ser seguidos.

## Requisitos funcionais

Alguns aspectos devem ser considerados consoante o contexto do software em desenvolvimento. Entendamos aplicativos altamente técnicos, aqueles que atendem ao segmento não-negócios, como exemplo apresentado pelo SWEBOK® GUIDE V4.0 (2024, p. 46):

O software que implementa o protocolo de comunicação de rede TCP/IP, também devem ter políticas e processos: "uma porta deve ser capaz de existir com zero, uma ou muitas conexões associadas, mas uma conexão deve existir em exatamente uma porta associada", "os estados aceitáveis de uma conexão devem ser 'ouvir', 'syn sent', 'estabelecido', 'fechamento'... ", e "se o tempo de vida de um segmento chegar a zero, esse segmento será excluído".

O analista de sistemas deve estar atento quanto à criticidade de um requisito funcional, em uma organização de negócios, portanto é indispensável determinar a volatilidade/estabilidade do requisito. Alguns requisitos podem mudar durante o ciclo de vida do software (durante o próprio processo de desenvolvimento da primeira versão).

Por exemplo, em uma aplicação de relacionamento de clientes, os requisitos (1) para as funções de cálculo e crédito de juros nas contas a pagar dos clientes são provavelmente mais estáveis do que um requisito (2) para criar critérios de abordagens no contato, relacionamento, monitoramento do cliente. O primeiro reflete uma característica fundamental do domínio econômico (que as dívidas podem sofrer interferências por juros), enquanto o último pode se tornar obsoleto por uma mudança estratégica da empresa, conforme percepções no mercado tão competitivo.

Essa classificação, volatilidade/estabilidade do requisito, realizada pelo analista de sistemas, contribuirá potencialmente ao projeto; pode auxiliar o engenheiro de software a estabelecer um design (projeto de software) mais tolerante a mudanças.

Uma vantagem da classificação volatilidade/estabilidade do requisito é auxiliar na priorização (muito raramente os requisitos têm graus de importância iguais); a especificação de requisitos apresenta maior qualidade quando prioriza explicitamente os requisitos em função de sua estabilidade (chances que ele tem de ser alterado), segundo Kerr (2015).

A priorização é dada por um estudo, podendo ser submetido à matriz de priorização, na qual são atribuídos valores (ou pesos) com base em critérios: importância ao negócio, impacto e complexidade, obrigatória, facultativa, desejável, e até ponderada em relação ao custo de desenvolvimento e implementação, gerando uma lista organizada de requisitos prioritários.

**Siga em Frente...**

## Requisitos não funcionais

Os requisitos não funcionais, normalmente, têm um escopo global, pois sua satisfação não pode ser alocada a um componente discreto. Portanto, esse tipo de requisito pode afetar fortemente a arquitetura de software, como também o design de muitos componentes. Ao contrário de requisito com escopo restrito, impactará menos na satisfação de outros requisitos.

Compreendamos as características e decisões que o analista de sistemas deverá analisar e desempenhar em função da classificação do requisito não funcional.

Na fase de testes e validações, para efetuar o tratamento de erros, tratamento de exceções e tolerância a falhas, o detalhamento dos requisitos não funcionais, torna-se fundamental.

A forma como os erros são tratados afeta a capacidade do software de atender aos requisitos relacionados à correção, robustez e outros atributos não funcionais. Às vezes, as asserções são usadas para verificar se há erros. Outras técnicas de tratamento de erros – como retornar um valor neutro, substituir o próximo dado válido, registrar uma mensagem de aviso, retornar um código de erro ou desligar o software – também são usadas.

Para SWEBOK® GUIDE V4.0 (2024), o especialista em negócios não consegue compreender os requisitos quando ele recebe requisitos funcionais e não funcionais intercalados para revisão ou validação, pois as descrições ou definições em tecnologia não são conhecidas por ele; mais um motivo para classificar os requisitos.

Conheçamos aspectos não funcionais que um software poderá atender:

- **Escalabilidade:** capacidade do software de aumentar as características não funcionais: carga, número de transações e volume de dados; está relacionada à complexidade da plataforma e do ambiente de execução, como redes distribuídas, sem fio e ambientes virtualizados, clusters de grande escala e nuvens móveis.
- **Disponibilidade, continuidade e níveis de serviço de software:** devem ser gerenciadas para garantir que os compromissos do cliente sejam atendidos. Nesse caso, os engenheiros de operações garantirão que a infraestrutura adequada seja planejada, projetada, implementada e testada. A disponibilidade do software é medida e registrada, e a indisponibilidade não planejada é investigada e as ações apropriadas são tomadas. Os relatórios de serviço produzem indicadores de disponibilidade e continuidade dos serviços de operações em relação às metas de nível de serviço.

Em resumo, os processos de classificação de requisitos ajudam a organizar e priorizar as necessidades do sistema, orientando o desenvolvimento e garantindo que o sistema entregue valor conforme as expectativas dos stakeholders e as restrições do projeto.

Estudante, continue pesquisando e estudando sobre a importância da classificação de requisitos.

Bons estudos!

## Vamos Exercitar?

Por ser um projeto de campo, o software deve ser preparado com o apoio de infraestrutura com sistema de conexão a longo alcance de Internet, mas deve prever um sistema que faça atualização em segundo plano, visto que poderá ficar sem comunicação síncrona por horas ou até dias.

Os aplicativos de coleta de dados nos talhões do cafezal podem ser em celular, mesmo que não tenha conexão de Internet, pois esse tipo de equipamento possui bateria de longa duração e poder de armazenamento de grande volume de dados.

Vejamos como o analista de sistemas classificará os requisitos de software nessa fase do projeto:

Requisitos funcionais:

1. Cadastro de novos colhedores de café, com o escaneamento do documento ou recebimento de carteira de identificação, via sistema de compartilhamento, em nuvem ou com sinal NFC ou Bluetooth.
2. Ao efetuar a inserção de dados do colhedor de café, uma assinatura deve ser coletada dando ciência da Lei LGPD.
3. O sistema deve mostrar os talhões, seguindo a localização GPS do celular.
4. Os sistemas de coleta de dados de temperatura e umidade devem transmitir esses dados sincronizando com a movimentação inserida pelo operador no smartphone.

Requisitos não funcionais:

1. A comunicação entre os equipamentos deve ser assegurada via sistema de compartilhamento, em nuvem ou com sinal NFC ou Bluetooth.
2. Os equipamentos IoT, leitores de temperatura e umidade, devem prever a comunicação com o smartphone.

Requisito de domínio:

1. O sistema deve prever a exibição e o cumprimento da LGPD, sempre que a pessoa proprietária dos dados solicitar.

Estudante, com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos.

Bons estudos!

**Saiba mais**

# ENGENHARIA DE REQUISITOS

## I. Saiba mais sobre: categorias

Leia o item 1.2. *Categorias de Requisitos de Software*, página 45.

[Guide to the Software Engineering Body of Knowledge v4.0](#)



## II. Saiba mais sobre: requisito de produto e de projeto

Leia o item 1.3. *Requisitos do produto de software e requisitos do projeto de software*, página 45.

[Guide to the Software Engineering Body of Knowledge v4.0](#)



### III. Saiba mais sobre: requisitos funcionais e não funcionais

Leia na página 46 o item 1.4 *Requisitos funcionais* e 1.5 *Requisitos não funcionais*.

[Guide to the Software Engineering Body of Knowledge v4.0](#)



#### IV. Saiba mais sobre: requisitos não funcionais

Leia na página 109 o item 7.2.5 *Requisitos não funcionais*.

[Engenharia de software](#)

#### V. Saiba mais sobre: requisitos com estabilidade ou volatilidade

Leia nas páginas 94 e 95 sobre a volatilidade e risco de requisitos.

## Engenharia de requisitos

### Referências

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: [https://integrada\[minhabiblioteca.com.br\]/reader/books/9786558040118/](https://integrada[minhabiblioteca.com.br]/reader/books/9786558040118/) Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: [https://integrada\[minhabiblioteca.com.br\]/reader/books/9786556900674/](https://integrada[minhabiblioteca.com.br]/reader/books/9786556900674/) Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos: software orientado ao negócio**. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

Washizaki, H. **Guide to the Software Engineering Body of Knowledge v4.0**. Waseda University: 2024. Disponível em: [https://bit.ly/engreq\\_swebokv4](https://bit.ly/engreq_swebokv4) Acesso em: 15 nov. 2024.

## Aula 2

Priorização de Requisitos

### Priorização de requisitos



#### Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer a **priorização de requisitos**.

Esse conteúdo é importante para a sua prática profissional, pois compreende os fundamentos de priorização de requisitos, os critérios de priorização de requisitos e as técnicas de priorização de requisitos.

Prepare-se para esta jornada de conhecimento! Vamos lá!!!

## Ponto de Partida

Os gestores da CaféSaúde estão empenhados no desenvolvimento dos sistemas de gerenciamento da cadeia produtiva de café Standard, Premium e High Special. Para tanto, você ficará responsável por explicar a eles como se aplicará a priorização de requisitos, quais critérios e técnicas para escolher os requisitos mais importantes, para estabelecer uma ordem de desenvolvimento.

Estudante, com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos. Porém, continue pesquisando e atualizando seu conhecimento sobre os fundamentos da priorização de requisitos para aplicar no cotidiano profissional.

Bons estudos!

## Vamos Começar!

Estudante, bem-vindo ao conteúdo do processo de priorização de requisitos!

Com os estudos, percebemos que o gerenciamento de mudanças de requisitos deve ocorrer em qualquer metodologia de desenvolvimento, com critérios ajustados, para atingir os objetivos estratégicos de negócio e para otimizar os recursos.

Porém, no ciclo de vida ágil, acontece implicitamente pelo curto tempo da iteração de cada entrega, pois podem ocorrer iterações com a entrega de apenas um requisito novo do backlog. Ou seja, sabemos que todas as solicitações de mudanças de requisitos previamente acordados, tornam-se mais um item no backup do produto. Somente ao receber uma prioridade é que se tornará um item aceito para entrar nas próximas iterações (em scrum é a sprint).

## Fundamentos de priorização de requisitos

Segundo SWEBOK® GUIDE V4.0 (2024), a priorização de requisitos de software é útil em projetos de software porque auxilia os engenheiros de software na concentração de esforço na entrega da funcionalidade mais valiosa o mais breve possível. Também, auxiliará muito em casos de decisões inteligentes de trade-off (troca) envolvendo resolução de conflitos de escopo.

Os requisitos priorizados no desenvolvimento do projeto inicial também serão priorizados em caso de alteração/correção deles, no processo de manutenção. Ou seja, os defeitos identificados

em relação aos requisitos de prioridade mais alta (no projeto inicial), provavelmente, devem ser reparados antes dos defeitos levantados em relação aos de menor prioridade.

Outros fatores primordiais na priorização de requisitos para software estão tanto na amplitude quanto na profundidade dele. Os requisitos que abrangem muitas funcionalidades (sem muito detalhamento) podem satisfazer usuários que não exigem complexidade dos sistemas, conforme ilustrado na Figura 1. Enquanto requisitos que atendem às especificidades em uma regra de negócios requer conhecimento aprofundado dessas regras pelo analista de sistemas e engenheiro de software.

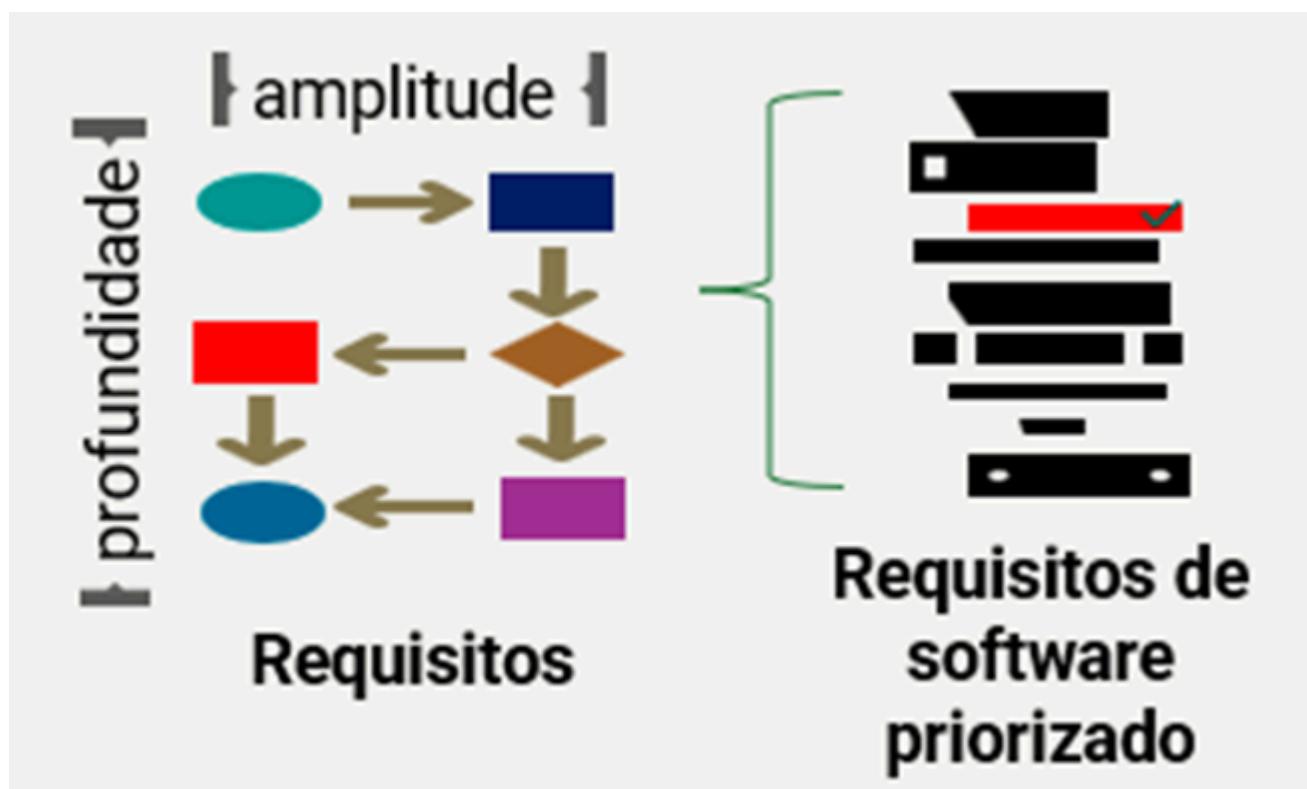


Figura 1 | Detalhamento com profundidade e expansão da amplitude.

Sendo assim, a metodologia de entregas regulares com atividades iterativas em curto ciclo de desenvolvimento de software proporciona melhores resultados na priorização de requisitos, em detrimento da necessidade de expansão da amplitude e a especificação da profundidade a cada iteração.

## Critérios de priorização de requisitos

A priorização envolve a classificação de alternativas com base em critérios comuns para oferecer o melhor valor, para SWEBOK® GUIDE V4.0 (2024). Por exemplo, em projetos de engenharia de software, os requisitos de software são frequentemente priorizados para agregar o máximo valor ao cliente dentro das restrições de cronograma, orçamento, recursos e

tecnologia, ou para permitir que a equipe construa o produto em incrementos - os primeiros incrementos fornecem o maior valor para o cliente.

Se o único critério de seleção for dinheiro, será escolhida a alternativa com o maior valor presente, valor futuro etc. Quando existem vários critérios, podem ser utilizadas várias técnicas para avaliar os critérios em conjunto. São fatores que determinam as prioridades de requisitos: maior ou menor urgência e importância em um projeto da organização. Sempre, os critérios são aplicados visando otimizar os recursos; ao mesmo tempo, alinhar as entregas com os objetivos do negócio e as expectativas dos stakeholders.

Ressaltamos critérios mais utilizados, as respectivas questões, para priorização de requisitos de software:

- a. **Importância para o negócio:** o requisito agrega valor significativo e é essencial para o cumprimento dos objetivos do negócio.
- b. **Valor para o cliente:** atende às expectativas e necessidades do cliente ou usuário final.
- c. **Urgente:** mede quão rapidamente o requisito de software precisa estar pronto.
- d. **Impacto no sistema:** o requisito afetará outras partes do sistema, negativamente a arquitetura ou depende de outros elementos críticos.
- e. **Complexidade técnica:** refere-se ao nível de dificuldade para implementar o requisito, considerando tecnologias envolvidas e possíveis riscos, conforme o prazo esperado.
- f. **Risco:** considera os riscos técnicos, financeiros ou regulatórios envolvidos.
- g. **Frequência de uso:** quão frequentemente o requisito será usado no sistema, são mais prioritários que funcionalidades raramente utilizadas.
- h. **Dependências:** requisito pré-requisito para outros, deve anteceder outros.

Outros critérios podem ser aplicáveis quando o contexto do sistema exigir. Estudante, fique atento para quesitos demandados pela alta administração ou órgãos reguladores e fiscais. Ou seja, existem muitos outros critérios que podem se tornar mais importantes ou urgentes do que listamos aqui.

**Siga em Frente...**

## Técnicas de priorização de requisitos

Uma variedade de esquemas de priorização está disponível. Para qualquer técnica utilizada, o analista de sistemas e o engenheiro de software devem avaliar a sua eficácia ao longo do tempo, após algumas iterações, compreendendo os resultados e os impactos gerados positivamente ou não, conforme o contexto da organização e da cultura organizacional.

Alguns profissionais adotam algumas perguntas-chave para auxiliar na escolha da melhor abordagem, segundo SWEBOK® GUIDE V4.0 (2024). Vejamos as perguntas para saber quais fatores são relevantes para determinar a prioridade de um requisito sobre outro?

- O valor agregado, ou a conveniência, ou quão satisfeito o cliente ficou com o requisito de software (para cliente e usuário).
- Qual a indesejabilidade do requisito ou a insatisfação de clientes (para clientes e usuários).
- Qual o custo de entrega do requisito.
- Qual o custo de manutenção ao longo da vida útil do software.
- Qual o risco técnico de implementação.
- Qual o risco de que os usuários não o utilizem, mesmo que implementado.

Por esses questionamentos, o analista de sistemas e o engenheiro de software, com os demais stakeholders, poderá elencar os requisitos de softwares prioritários.

Seguindo critérios semelhantes, de acordo com SWEBOK® GUIDE V4.0 (2024), o modelo de Kano mostra que considerar apenas valor, desejo ou satisfação pode levar a prioridades errôneas. Quando é considerado o grau de insatisfação dos stakeholders, se o requisito de software não é atendido, temos melhor compreensão das prioridades. A priorização deve pesar tanto à satisfação que os usuários experimentarão por certos recursos quanto à insatisfação que experimentarão se não tiverem:

Por exemplo, considere num projeto para desenvolver um cliente de e-mail, analisando a prioridade entre dois requisitos: (1) ter um filtro de spam eficaz, ou (2) manipular os anexos em e-mails. É provável que os usuários fiquem satisfeitos com um filtro de spam eficaz do que com a capacidade de lidar com anexos, portanto, o filtro de spam receberia uma prioridade mais alta com base no critério de satisfação. Por outro lado, a incapacidade de manusear anexos deixam muitos usuários extremamente insatisfeitos - muito mais do que não ter um filtro de spam eficaz.

Para SWEBOK® GUIDE V4.0 (2024), a segunda questão-chave é

"Como podemos converter o conjunto de fatores relevantes em uma expressão de prioridade?"

Temos uma fórmula da prioridade,

$$\text{Prioridade} = \frac{(\text{Valor} * (1 - \text{Risco}))}{\text{Custo}}$$

É apenas um exemplo de uma função matemática, mais objetiva, para obter valor numérico, o que levaria a uma decisão mais coerente, desde que os dados utilizados em "Valor agregado" e "Risco" pudessem ser apurados com rigorosos critérios. Assim, a escolha de esquemas de medição para os fatores relevantes poderá trazer benefícios.

Uma vez calculada a prioridade dos requisitos, devem ser especificadas de forma que possam ser comunicadas a todos os stakeholders, conforme ilustra a Figura 2. Verifiquemos algumas opções:

- I. Escala rotulada (por exemplo: deve fazer, pode fazer, desejo fazer).
- II. Escala numérica (por exemplo: 1 . . . 10).
- III. Listas que classificam os requisitos em ordem decrescente de prioridade.

A priorização eficaz de requisitos concentra-se em encontrar grupos de requisitos com prioridades semelhantes, em vez de criar escalas de medição excessivamente rígidas ou debater pequenas diferenças. Porém, deve-se sempre agrupar requisitos que estejam alinhados aos objetivos estratégicos da organização.

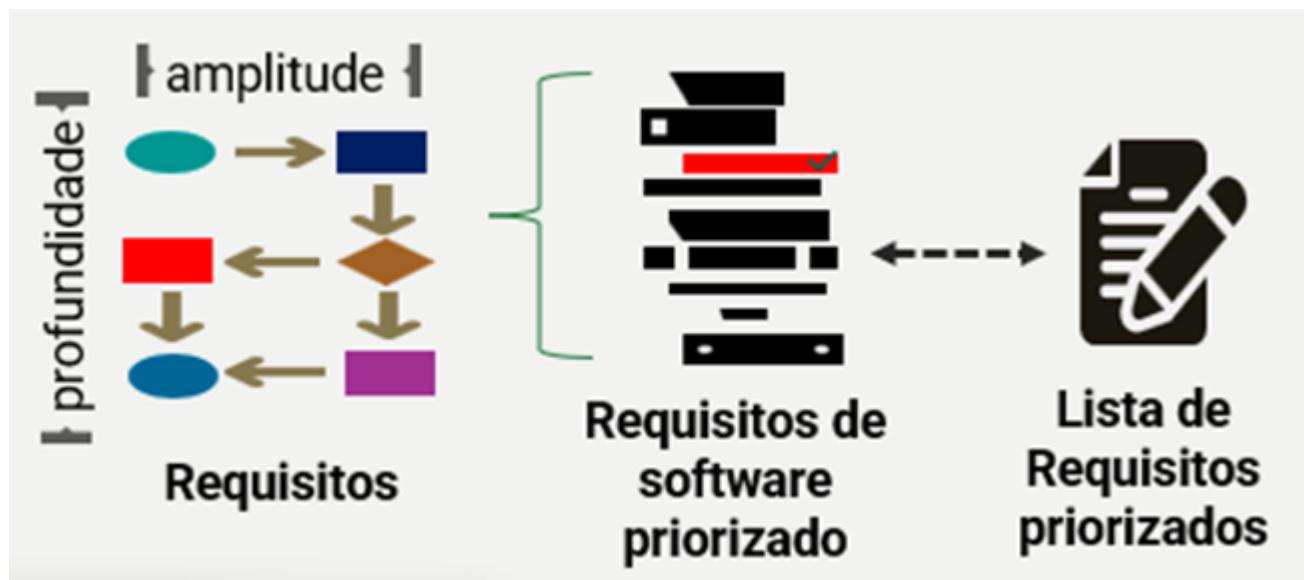


Figura 2 | Lista de requisitos priorizados.

Conheçamos a técnica de matriz de priorização de requisitos, ferramenta útil para priorizar os requisitos de software, considerando critérios como importância para o negócio, urgência, custo e impacto no sistema. A matriz auxilia a determinação de quais requisitos devem ser implementados com maior prioridade, em uma visão clara aos stakeholders e ao time de desenvolvimento.

Vamos ao passo a passo para elaborar a matriz de priorização, lembrando que adaptará cada organização essa matriz conforme o seu contexto:

1. Definir os critérios de priorização: escolher os critérios que sejam relevantes para o projeto. Alguns dos mais comuns incluem: importância para o negócio, urgência, custo de implementação, impacto no sistema, dependências e risco.
2. Atribuir peso a cada critério: cada critério deve receber um peso para refletir a relevância em relação aos objetivos do projeto; uma escala comum para pesos é de 1 a 5, onde 1 representa menor peso e 5 representa maior peso. Por exemplo, se o "custo de implementação" é mais relevante que a "urgência", ele pode receber um peso maior.
3. Avaliar os requisitos para cada critério: os requisitos considerados críticos para o negócio deve receber pontuação maior (5) para o critério "importância para o negócio" e assim sucessivamente, atribuindo uma pontuação de 1 a 5 para cada requisito em relação a cada critério.

4. Calcular a pontuação total de cada requisito: para cada requisito e para cada critério, multiplique a pontuação atribuída pelo peso respectivo e obtenha uma somatória de valores, para compor a pontuação total, seguindo a fórmula:

$$\text{Pontuação Total} = (\text{Pontuação Critério}_1 \times \text{Peso}_1) + \dots + (\text{Pontuação}_n \times \text{Peso}_n)$$

- Envolver stakeholders: discutir e validar os critérios e as pontuações com os stakeholders, para assegurar que as prioridades estão alinhadas visando negócio.
- Revisar e atualizar regularmente: reavaliar a matriz conforme novos requisitos surgem ou as condições do projeto mudam.
- Manter a simplicidade: não incluir muitos critérios, para evitar complexidade excessiva no cálculo e na interpretação dos resultados.

A matriz de priorização é uma abordagem flexível que pode ser adaptada a diferentes projetos, conforme o cenário da organização, ajudará a garantir que os recursos sejam direcionados para as funcionalidades mais importantes e de maior valor para o negócio.

Saibamos que a técnica utilizada para priorização de requisitos tem o conceito derivado da [Matriz de Eisenhower](#), fundamento para o gerenciamento de atividades e do tempo, com base na análise de importância e urgência.

Em resumo, os processos de priorização de requisitos, critérios e técnicas de priorização de requisitos auxiliam a organizar as necessidades do sistema, orientando o desenvolvimento e garantindo que o sistema entregue valor conforme as expectativas dos stakeholders e as restrições do projeto.

Estudante, continue pesquisando e estudando sobre a importância da priorização de requisitos.

Bons estudos!

## Vamos Exercitar?

Estudante, agora é dada a oportunidade para você exercitar a análise de sistemas com maior rigor, principalmente por tratar de um assunto extremamente importante em um desenvolvimento de software, a priorização dos requisitos.

Vejamos que uma decisão de priorizar um requisito pode ter consequências graves para o projeto da CaféSaúde, pois os recursos estão limitados e as necessidades de sistemas em funcionamento são urgentes. Por exemplo, se um requisito é escolhido inadequadamente, antes de outro que deveria ter sido implementado, e poderá ter atraso de sessenta dias e um esforço que nem sempre podemos recuperar num projeto de dez meses, num ciclo usando Scrum. Porém, se a metodologia utilizada contemplar ciclo de build com maior tempo, o prejuízo será muito maior.

Alguns critérios que elencamos para você explicar aos gestores como priorizar os requisitos de software, são:

- Importância para o negócio: a “integração das informações de colheita com a unidade de beneficiamento do café” (secagem e seleção de grãos), tem impacto direto para tomada de decisões nos negócios.
- Urgente: o “registro de informações de rastreamento do produto”, integrado por IoT, para atender às legislações internacionais de alimentos, Lei Alimentar Geral (EC 178/2002).
- Dependências: as funcionalidades para “cadastro de talhões”, “cadastro de colhedor”, “cadastro de máquinas e equipamentos”, devem ser priorizadas em relação aos de movimentação que dependerão dos dados cadastrais.

Exemplifiquemos o uso de matriz de priorização de requisitos. Imaginemos cinco critérios (com os respectivos pesos) para priorizar os requisitos de software da CaféSaúde:

Importância para o negócio (peso = 4); urgência (peso = 3); custo de implementação (peso = 2); impacto no sistema (peso = 2) e risco (peso = 1).

#### Bloco 1

Requisitos	Importância	Urgência	Custo
Cadastro do colhedor	5 = 20	4 = 12	3 = 6
Relatório de colheita	3 = 12	2 = 6	1 = 2
Integração dos dados	5 = 20	4 = 12	2 = 4
Entrada de dados	5 = 20	4 = 12	4 = 8

#### Bloco 2

Impacto	Risco	Pontos
2 = 4	2 = 2	44
1 = 2	1 = 1	23
3 = 6	1 = 1	43
2 = 4	1 = 1	45

Quadro 1 | Matriz de prioridades de requisitos

Analizando a priorização dos quatro requisitos, temos, em ordem de pontuação:

1. “Cadastro do colhedor”, com a pontuação intermediária (44), esta deve receber maior prioridade por conter dependência (entrada de dados).

2. "Entrada de dados", com a maior pontuação (45), porém fica abaixo por ser dependente de "cadastro do colhedor";
3. "Integração dos dados", com pontuação intermediária (43).
4. "Relatório de colheita", com pontuação mais baixa (23).

Estudante, com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos.

Bons estudos!

## Saiba mais

### I. Saiba mais sobre: priorização de requisitos

Leia o item 7.1, *Natureza iterativa do processo de requisitos* e 7.2 *Priorização de requisitos*, páginas 59 e 60.

[Guide to the Software Engineering Body of Knowledge v4.0](#)



## II. Saiba mais sobre: priorização de requisitos

Leia o item 2., *Priorização de requisitos*, páginas 39 a 41.

[Engenharia de requisitos](#)

## III. Saiba mais sobre: matriz de importância e urgência

Leia o artigo sobre a Matriz de Eisenhower: como funciona e como fazer.

[Matriz de Eisenhower](#)



#### IV. Saiba mais sobre: priorização é algo contínuo

Leia sobre continuar analisando as prioridades, item 9.7. *Priorizar Requisitos*, nas páginas 264 e 265.

[Engenharia de requisitos: software orientado ao negócio](#)

#### V. Saiba mais sobre: técnica de priorização

Leia sobre a técnica de Moscow, página 267 a 268.

## [Engenharia de requisitos: software orientado ao negócio](#)

### Referências

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: [https://integrada\[minhabiblioteca\].com.br/reader/books/9786558040118/](https://integrada[minhabiblioteca].com.br/reader/books/9786558040118/) Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: [https://integrada\[minhabiblioteca\].com.br/reader/books/9786556900674/](https://integrada[minhabiblioteca].com.br/reader/books/9786556900674/) Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos: software orientado ao negócio**. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

WASHIZAKI, H. **Guide to the Software Engineering Body of Knowledge v4.0**. Waseda University: 2024. Disponível em: ou [https://bit.ly/engreq\\_swebokv4](https://bit.ly/engreq_swebokv4) Acesso em: 15 nov. 2024.

## Aula 3

Análise de Requisitos

### Análise de requisitos

#### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula você irá conhecer a **análise de requisitos**.

Esse conteúdo é importante para a sua prática profissional, pois envolve as características de caso de uso de requisitos, dos cenários de uso de requisitos e de requisitos no ciclo de vida do projeto.

Prepare-se para esta jornada de conhecimento! Vamos lá!!!

## Ponto de Partida

Os gestores da CaféSaúde estão empenhados no desenvolvimento dos sistemas de gerenciamento da cadeia produtiva de café Standard, Premium e High Special. Você ficará responsável por explicar a eles como acontece a análise de requisitos e os benefícios dos caso de uso de requisitos e o ciclo de vida de requisitos com ciclo de vida do projeto.

Com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos. Continue pesquisando e atualizando seu conhecimento sobre os fundamentos da análise de requisitos para aplicar no cotidiano profissional.

Bons estudos!

## Vamos Começar!

Estudante, seja bem-vindo ao conteúdo do processo de análise de requisitos!

Entendamos a análise de requisitos como principal atividade resultante na especificação das características operacionais do software, que indica uma solução das necessidades de mudanças ou resolução de problemas dos usuários finais.

Os resultados da análise de requisitos dão ao projetista de software (ou engenheiro de software) informações úteis e imprescindíveis para serem transformadas em projetos de arquitetura, de dados, de interfaces e de componentes. As descrições (em modelagem ou em linguagem natural) da especificação de requisitos de software fornecem ao desenvolvedor e ao cliente os meios para construir o software e validar a qualidade após sua construção.

Vamos nos concentrar na análise baseada em cenários, uma técnica cada vez mais popular em toda a comunidade de engenharia de software.

## Caso de uso de requisitos

Na engenharia de requisitos é comum se beneficiar do caso de uso para documentar as funcionalidades do sistema. Uma ferramenta estruturada e orientada às necessidades dos usuários, flexível por utilizar linguagem natural, fornece o início sólido para representar os

requisitos de um sistema e, inclusive, facilitam a comunicação entre stakeholders e o time de desenvolvimento.

Além de ser um dos primeiros documentos, o caso de uso será referência para o design do projeto de software, para a implementação e validação do sistema.

Vejamos um exemplo descrito em SWEBOK® GUIDE V4.0 (p. 54, 2024), no Quadro 1, no qual as descrições expressam os requisitos em linguagem natural, no contexto da organização produtora de café em grãos.

Caso de Uso #87	Cadastramento de colhedores de café
Eventos de acionamento	<ul style="list-style-type: none"> <li>Ao apontar o nome de uma pessoa sem cadastro.</li> <li>Quando o recrutador enviar o nome de novo colhedor.</li> </ul>
Parâmetros	Nome, CPF, telefone, data_nascimento, cart_trabalho.
Requisitos mínimos	Dados completos para o cadastramento.
Garantias	A pessoa tem o seu cadastro efetivado.
Fluxo Principal	Inclusão do cadastro do colhedor com todos os dados obrigatórios.
Fluxos alternativos	<ul style="list-style-type: none"> <li>O colhedor de café com idade entre 16 e 18 anos deve ter o cadastro autorizado pelo responsável civil.</li> <li>A pessoa sem a Cart_trabalho pode ter o cadastro validado com o MEI.</li> </ul>
Exceções	<ul style="list-style-type: none"> <li>Não permitir cadastro de pessoas menores de 16 anos completos.</li> <li>Não permitir cadastro de pessoas entre 16 e 18 anos, sem o responsável civil.</li> <li>Não cadastrar pessoa sem a Cart_trabalho ou MEI.</li> <li>Não cadastrar pessoa sem o CPF válido.</li> </ul>

Quadro 1 | Caso de uso estruturado para cadastro de colhedor. Fonte: adaptado de SWEBOK Guide v4.0 (2024, p. 54).

O caso de uso poderá ser descrito em linguagem natural não estruturada, sendo uma coleção de declarações. Por exemplo: no contexto de uma faculdade, "um aluno não pode se inscrever nos cursos do próximo semestre se houver alguma taxa de matrícula não paga" é um exemplo de regra de negócios que serve como requisito.

## Cenários de uso de requisitos

Para que o analista de sistemas e o engenheiro de software elaborem o design de projeto, ou protótipos de sistemas para compreender o quadro geral de uma organização cliente, apostando numa visão holística, eles podem usar metodologias de pensamento sistêmico, de acordo com SWEBOK Guide v4.0 (2024, p. 306).

Imagine que você pretenda fornecer um software que poderá ser útil para vários segmentos, essa abordagem o ajudará a criar um conjunto completo de cenários de uso do software. Essa abordagem auxilia os desenvolvedores de software a criarem um conjunto completo de cenários possíveis em que o software seja útil. Por meio de vários cenários de uso e de dados caracterizando esses cenários, possibilita ao cliente e ao usuário perceberem potenciais uso do software.

## Siga em Frente...

## Requisitos no ciclo de vida do projeto

Em SWEBOK® GUIDE V4.0 (2024, p. 308),

"Um Ciclo de Vida do Produto de Software (CVPS) inclui todas as atividades necessárias para definir, construir, operar, manter e aposentar um produto ou serviço de software e suas variantes".

Compreendemos que os requisitos de software fazem parte do produto do software, portanto devemos estar cientes de que, antes das primeiras atividades do projeto de desenvolvimento de software, temos muitas informações sobre os requisitos e sobre o produto de software. Percebemos a afirmação em SWEBOK® GUIDE V4.0 (2024, p. 308), que diz: "as atividades do CVPS de 'operar', 'manter' e 'reformar' ocorrem num período muito mais longo do que o desenvolvimento inicial do software, ou seja do Ciclo de Vida do Desenvolvimento de Software (CVDS)", conforme ilustramos na Figura 1.



Ciclo de Vida do Requisito

Ciclo de Vida do Desenvolvimento

Figura 1 | Ciclo de vida do requisito.

Os requisitos de um produto de software começam logo que as regras de negócios são estabelecidas para a realização dos processos de negócio de uma organização, antes que as atividades de projeto de software se iniciem. Assim que se inicia o projeto de software, todos os artefatos de requisitos até então confeccionados serão aproveitados integralmente, conforme metodologia adotada pelo time de desenvolvimento de software.

Ainda que o projeto de software (especificamente para um produto de software) se encerre por contrato ou força maior, as mudanças são naturalmente necessárias para atender às necessidades de negócio ou adaptações tecnológicas. Portanto, um novo projeto de software será aberto, ainda que seja somente para manutenção desse sistema.

Lembramos que as modificações realizadas em requisitos de software e em seus artefatos devem ser gerenciadas com os mesmos cuidados e rigor, sem se associar a um projeto de projeto. Ou seja, continuam mantendo o gerenciamento das atividades de elicitação, especificação, revisão, validação e documentação, inclusive garantindo a rastreabilidade de requisitos.

Continue pesquisando e estudando sobre a importância da análise de requisitos.

Bons estudos!

## Vamos Exercitar?

Estudante, agora é dada a oportunidade para você exercitar a análise de sistemas com maior rigor, principalmente, por tratar de um assunto extremamente importante do desenvolvimento de software, a análise de requisitos.

Você deverá aproveitar o momento para demonstrar cenários que podem ser úteis aos gestores da CaféSaúde, garantindo maior segurança, como também ilustrando aos gestores, novas formas de uso do sistema. Percebemos a inserção de novas funcionalidades com identificação de cenário de uso 2, conforme o Quadro 2.

Caso de Uso #87	Cadastramento de colhedores de café
Eventos de acionamento	Ao apontar o nome de uma pessoa sem cadastro. Quando o recrutador enviar o nome de novo colhedor.
Parâmetros	Nome, CPF, telefone, data_nascimento, cart_trabalho.
Requisitos mínimos	Dados completo para o cadastramento.

Garantias	A pessoa tem o seu cadastro efetivado. Cenário de uso 2: se parametrizado, recrutador e trabalhador avisados.
Fluxo Principal	Inclusão do cadastro do colhedor com todos os dados obrigatórios. Cenário de uso 2: Se parametrizado, envio da confirmação do cadastro ao recrutador por e-mail e ao trabalhador por mensagem de WhatsApp.
Fluxos alternativos	<ul style="list-style-type: none"> <li>• O colhedor de café com idade entre 16 e 18 anos, deve ter o cadastro autorizado pelo responsável civil.</li> <li>• A pessoa sem a Cart_trabalho pode ter o cadastro validado com o MEI.</li> </ul>
Exceções	<ul style="list-style-type: none"> <li>• Não permitir cadastro de pessoas menores de 16 anos completos.</li> <li>• Não permitir cadastro de pessoas entre 16 e 18 anos, sem o responsável civil.</li> <li>• Não cadastrar pessoa sem a Cart_trabalho ou MEI.</li> <li>• Não cadastrar pessoa sem o CPF válido.</li> </ul>

Quadro 2 | Cenário de uso – com cenário de uso 2.

Lembremo-nos que os gestores da CaféSaúde também saíram que para todos os sistemas desenvolvidos, mesmo que tenha sido concluído o projeto do software ou que tenha encerrado o contrato de desenvolvimento (caso tenha sido dessa forma), o gerenciamento dos requisitos deve continuar, as modificações devem ser analisadas e previstas para futuras implementações ou manutenções.

Estudante, com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos.

Bons estudos!

## Saiba mais

### I. Saiba mais sobre: especificação de requisitos – caso de uso

Leia o item 4.1, *Especificação de requisitos de linguagem natural não estruturada*, páginas 53 e 54.

[Guide to the Software Engineering Body of Knowledge v4.0](#)



## II. Saiba mais sobre: análise de requisitos

Leia o item 8.1, *Análise de requisitos*, página 127.

[Engenharia de software](#)

## III. Saiba mais sobre: ciclo de vida do produto de software

Leia o item 10.11, *Ciclo de vida do produto*, página 308.

[Guide to the Software Engineering Body of Knowledge v4.0](#)



#### IV. Saiba mais sobre: análise de requisitos

Leia o item 3, *Especificação de requisitos* (documentação), na página 19.

[Gerenciamento de requisitos](#)

## Referências

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos**: software orientado ao negócio. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

WASHIZAKI, H. **Guide to the Software Engineering Body of Knowledge v4.0**. Waseda University: 2024. Disponível em: ou [https://bit.ly/engreq\\_swebokv4](https://bit.ly/engreq_swebokv4) Acesso em: 15 nov. 2024.

## Aula 4

Modelagem de Requisitos

### Modelagem de requisitos

#### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer a **modelagem de requisitos**. Esse conteúdo é importante para a sua prática profissional, pois compreende os artefatos do elicição de requisitos, levantamento ágil de requisitos e sistemas autoadaptativos. Prepare-se para esta jornada de conhecimento! Vamos lá!!!

## Ponto de Partida

Os gestores da CaféSaúde estão empenhados no desenvolvimento dos sistemas de gerenciamento da cadeia produtiva de café Standard, Premium e High Special.

Como preparar requisitos que possam ser implementados para se tornar autoadaptativos? Fica aqui a sua oportunidade para explicar como esse conceito fará parte da história da CaféSaúde.

Dentre os métodos de elicitação de requisitos de software, você fará a demonstração do levantamento ágil de requisitos e dos artefatos da elicitação de requisitos.

Com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos. Continue pesquisando e atualizando seus conhecimentos sobre os fundamentos da modelagem de requisitos para aplicar no cotidiano profissional.

Bons estudos!

## Vamos Começar!

Estudante, bem-vindo ao conteúdo do processo de modelagem de requisitos!

Lembremos que a construção de modelos nos permite o uso intenso de cenários. Sempre que os analistas de sistemas e os engenheiros de software se concentram para encontrar uma solução sistêmica ou abstrair resoluções para um problema, baseiam-se em cenários; uma técnica que se tornou eficiente em toda a comunidade de engenharia de software.

## Artefatos da elicitação de requisitos

Entendamos a modelagem de requisitos como o principal resultado da especificação das características operacionais do software, por meio da análise de sistemas, que indica uma solução das necessidades de mudanças ou resolução de problemas dos usuários finais, com a participação do engenheiro de software e demais stakeholders.

Para Pressman (2021, p. 37), “o artefato mais importante é um ‘incremento de software’ operacional entregue para o cliente na data combinada. Os documentos mais importantes criados são as histórias de usuário e os casos de teste relacionados”.

Dentre muitos artefatos resultantes da análise de requisitos, elencamos os primeiros fundamentais para o projetista de software (ou engenheiro de software) contendo informações úteis e imprescindíveis para serem transformadas em projetos de arquitetura de software, de dados, de interfaces e de componentes. As descrições (em modelagem ou em linguagem natural) da especificação de requisitos de software fornecem ao desenvolvedor e ao cliente os meios para construir o software e validar a qualidade após ele ser construído.

A modelagem faz parte do processo de abstração, segundo SWEBOK® GUIDE V4.0 (2024), usado para representar características e elementos de um sistema, enquanto a simulação é utilizada para conduzir experimentos planejados, com base em um modelo de sistema, para entender melhor o seu comportamento, as relações entre os subsistemas e para analisar aspectos do projeto. A modelagem e simulação, juntas, beneficiam os engenheiros de software na construção de previsões por meio de teorias ou hipóteses sobre o comportamento do sistema. A prototipagem é outro processo de abstração para representar o sistema, com aspectos comportamentais; o protótipo pode ser uma versão inicial de um sistema, mesmo que não possua todas as funcionalidades da versão final.

O objetivo da modelagem é construir uma solução de um problema por meio da abstração, imaginando o que seria um sistema que atenderia às necessidades dos stakeholders. Ao analisar o modelo construído, é possível entender melhor o sistema e seus problemas, pois compreendem os limites de seu conhecimento sobre a situação.

## Levantamento ágil de requisitos

Até que as equipes de desenvolvimento de software mudassem a cultura da documentação em primeiro lugar, antes da construção do código-fonte, muitas horas eram dispensadas em burilamento das descrições e diagramação dos requisitos de software.

O método ágil trouxe o conceito fundamental para eliminar deficiências da engenharia de software convencional: fraquezas, ou seja, esforços que não fossem para entregar a nova versão/build do software.

Logicamente, não é indicado o uso do método ágil para todos os projetos de softwares; alguns produtos ou situações dependem de outras metodologias.

Percebemos que o mundo dos negócios é ágil e complexo! Diríamos que é quase impossível saber como as regras de negócio evoluirão nos próximos meses. Tanto a economia quanto as condições de mercado mudam rapidamente, alteram consequentemente as necessidades dos usuários e, sem nenhum alerta, novas ameaças competitivas surgem.

Portanto, em muitas situações, definir os requisitos completamente antes que se inicie a implementação é improvável; a agilidade na construção de software está na dinâmica da interação diretamente com os usuários para responder com o software funcionando para atender o novo cenário tão fluido e volátil.

Fluidez implica mudança, e mudança é cara, segundo Pressman (2021), principalmente sem eficiência no gerenciamento dos requisitos e do projeto de software. No método ágil, prima-se pela habilidade de reduzir os custos da mudança no processo de software.

Em princípio, o método ágil estabelece o diálogo e a interação entre os stakeholders com muita frequência e objetividade, antes e durante o processo de construção do software.

Para algumas pessoas, comunicar-se na forma escrita pode ser eficiente; para outros, não. Porém, os custos ficam por conta do esforço para escrever (ou da diagramação), na leitura do que foi descrito, nas horas dispensadas para os esclarecimentos pelo mal-entendido do texto, e assim sucessivamente quando então, retoma-se o entendimento após um bom diálogo entre as partes.

Entendamos como é a dinâmica do desenvolvimento de software com o framework Scrum. O time Scrum é interdisciplinar e se auto-organiza. É composto por um Product owner, um Scrum master e uma pequena equipe de desenvolvimento (três a seis pessoas).

Os principais artefatos do Scrum são: (1) o backlog do produto, (2) o backlog do sprint e (3) o incremento de código.

1. O backlog do produto é uma lista priorizada de requisitos ou características do artefato que agregam valor de negócios para o cliente. Sempre que surgirem novos itens, podem ser adicionados ao backlog a qualquer momento com a aprovação do Product owner e o consentimento da equipe de desenvolvimento. O Product owner ordena os itens para cumprir as metas mais importantes de todos os envolvidos; e somente ele pode decidir encerrar um sprint antecipadamente ou estendê-lo, caso o incremento não seja aceito.
2. O backlog do sprint é um subconjunto do backlog do produto, escolhido pela equipe de desenvolvimento, para gerar o incremento de código durante o sprint atual.
3. O incremento representa a união de todos os itens dos sprints anteriores e todos os itens do backlog do sprint atual. É nesse período que desenvolvedores e stakeholders devem obter o entendimento dos requisitos de software.

Os sprints têm uma limitação de tempo (ou seja, são time-boxed, de 3 a 4 semanas). O modo como o time de desenvolvimento implementa o incremento é uma responsabilidade do próprio time, que também decide quando está pronto para demonstrar ao Product owner.

O Scrum master atua como facilitador para todos os membros da equipe Scrum; e juntamente com o Product owner identificam técnicas para gerenciar os itens do backlog do produto e garantir que os itens estejam enunciados de forma clara e concisa.

O feedback, liderado pelo Product owner, da reunião de revisão do sprint, é o ponto alto da validação dos requisitos e do incremento (versão) do produto de software, com a participação do Scrum master, time de desenvolvimento e demais stakeholders. Se existirem características de requisitos a serem modificados, eliminados ou inseridos no incremento do software, esses ajustes serão realizados no planejamento do backlog do próximo Sprint.

**Siga em Frente...**

**Sistemas autoadaptativos**

Um manifesto alertou para uma iminente crise de software, publicado pela IBM em 2003, em função do aumento de complexidade em suas fases (instalação, ajuste, configuração e manutenção). Incertezas foram geradas logo após a implantação do software, pois os requisitos haviam sido modificados durante a sua implementação.

Weyns (2019) cita que uma classe de sistemas de software autoadaptativos foi concebida com mecanismos de autogerenciamento, capazes de se adaptar sem ou com o mínimo de intervenção humana.

Esses sistemas são projetados para responder às mudanças no ambiente operacional, com ajustes automáticos no seu comportamento, nos objetivos do sistema ou nas condições de uso. Assim, essa classe de sistemas garante a funcionalidade, eficiência e resiliência em cenários dinâmicos e imprevisíveis, com pouca intervenção humana. Entendamos algumas características de sistemas autoadaptativos:

1. **Capacidade de reação:** realiza ajustes automáticos com base nas informações monitoradas. Exemplo: um sistema de rede de comunicação digital ajusta a largura de banda alocada com base no tráfego em tempo real.
2. **Monitoramento contínuo:** capacidade de monitorar continuamente seu estado interno e o ambiente externo. Exemplo: um sistema de ventilação monitora a temperatura ambiente para ajustar sua eficiência.
3. **Objetivo dirigido:** segue as metas pré-definidas, como desempenho, economia de recursos ou segurança. Exemplo: um carro autônomo ajusta a rota para minimizar o tempo de viagem.
4. **Resiliência:** continua funcional, mesmo em condições adversas ou inesperadas. Exemplo: sistemas bancários que se adaptam a picos de acessos durante horários de alta demanda.
5. **Feedback e controle:** utiliza loops de feedback para avaliar os resultados das ações de adaptação e ajustar conforme necessário. Exemplo: sistemas de controle de trânsito ajustam os tempos de semáforos com base em padrões de trânsito.

Os sistemas autoadaptativos são úteis em cenários adversos, que requerem capacidade de adaptação contínua, oferecendo eficiência, resiliência e inovação, que podem ser fundamentais em IoT, computação em nuvem, segurança e saúde.

Continue pesquisando e estudando sobre a importância da análise de requisitos.

Bons estudos!

## Vamos Exercitar?

Estudante, agora é dada a oportunidade de você exercitar a análise e modelagem de sistemas juntamente com os gestores e usuários da empresa CaféSaúde, cujo cenário está baseado em um sistema de automação de secagem de café.

Devido à complexidade caracterizada pelas altas temperaturas exigidas pelo processo de secagem de café, o sistema apropriado para gerenciar todo processo é autoadaptativo. Vejamos, no Quadro 1, como seria a especificação dos requisitos para o sistema autoadaptável.

Nível	Propriedade	Exemplo do cenário
Geral	Autoadaptação	O self-adaptiveness, preparado para efetuar o autogerenciamento, autogoverno, automanutenção, autocontrole e autoavaliação. O sistema avalia o grão antes de iniciar a secagem, configura os parâmetros preestabelecidos, gerencia e otimiza o consumo de energia elétrica, devolve parte da quantidade de produtos, caso um imprevisto seja detectado, para evitar danos
Principal	Autoconfiguração	O self-configuration: ao receber produtos de diferentes situações, tamanho do grão, ou peso, o sistema faz a leitura dessas características e efetuar a regulagem para iniciar a secagem.
	Autocura	O sistema monitora integralmente os recursos para detectar mal funcionamento da entrada de ar frio ou quente, faz o ajuste necessário para manter a temperatura adequada ao produto.
	Auto-otimização	O self-tuning é um recurso que regula a saída de ar frio para aliviar o sistema de aquecimento.
	Autoproteção	Estar prevenido de recursos para detectar e avisar quando

		existe uma fuga abrupta da temperatura.
Primitivo	Autoconsciência	Regular a temperatura e a umidade, conforme parametrização exigida pelo produto em secagem.
	Ciência do contexto	Regulagem da temperatura e umidade do ambiente de secagem do café.

Quadro 1 | Hierarquia do sistema autoadaptativo.

Lembremo-nos, e que os gestores da CaféSaúde também saibam: o sistema em desenvolvimento será continuamente adequado, conforme as mudanças estratégicas do negócio, pois, a cada iteração (sprint), novos requisitos podem ser incluídos, alterados ou eliminados. Esses requisitos serão elicitados ao longo da etapa de projeto, design e implementação, respeitando o time-boxed da Sprint, com a interação diária com o Product Owner e demais stakeholders.

Bons estudos!

## Saiba mais

### I. Saiba mais sobre: artefatos no scrum

Leia o item 3.4.1, *Equipes e artefatos do Scrum*, páginas 42 a 44.

### Engenharia de software

### II. Saiba mais sobre: sistemas autoadaptativos

Leia o item 2.3, *Sistemas autoadaptativos*, páginas 18 a 21.

### III. Saiba mais sobre: gestão de artefatos

Leia o item 7, *Software Configuration Management Tools*, páginas 200 e 201.

### Guide to the Software Engineering Body of Knowledge v4.0



## Referências

CAMARGO, M. P. O. **Estabelecimento de uma arquitetura de referência para sistemas ciber-físicos autoadaptativos.** 2023, 125 f. Dissertação (Mestrado em Ciência da Computação ) – Unesp, Rio Claro, 2023. Disponível em: chrome-extension://efaidnbmnnibpcajpcglclefindmkaj/https://repositorio.unesp.br/server/api/core/bitstreams/824e6814-cb65-4b6b-9975-0b9a8c416805/contente. Acesso em: 27 já. 2025.

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos: software orientado ao negócio**. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

WASHIZAKI, H. **Guide to the Software Engineering Body of Knowledge v4.0**. Waseda University: 2024. Disponível em: ou [https://bit.ly/engreq\\_swebokv4](https://bit.ly/engreq_swebokv4) Acesso em: 15 nov. 2024.

WEYNS, D. **Software Engineering of Self-adaptive Systems**. Springer: 2019. Disponível em: [https://link.springer.com/chapter/10.1007/978-3-030-00262-6\\_11](https://link.springer.com/chapter/10.1007/978-3-030-00262-6_11) Acesso em: 18 nov. 2024.

## Aula 5

Encerramento da Unidade

### Videoaula de Encerramento

#### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer os fundamentos de classificação e priorização de requisitos de software.

Este conteúdo é importante para a sua prática profissional, pois compreenderá o processo de análise e modelagem de requisitos.

Prepare-se para essa jornada de conhecimento! Vamos lá!!!

## Ponto de Chegada

Olá, estudante! Vamos desenvolver a competência desta unidade, que é:

- Identificar e classificar requisitos de softwares.
- Explicar a priorização de requisitos.
- Descrever a análise de requisitos.
- Explicar os artefatos de requisitos.

Você deverá compreender e reconhecer a classificação dos principais tipos de requisitos de softwares, incluindo os critérios de priorização de requisitos, o caso de uso de requisitos e os artefatos dos requisitos.

## É Hora de Praticar!



### Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dentre vários requisitos de software na CaféSaúde, a sua responsabilidade é classificar cada um deles para organizar uma prioridade de desenvolvimento. Em todos os cenários de desenvolvimento de software, existem inúmeros requisitos, porém, na maioria das vezes, é impossível desenvolver todos os requisitos elicitados.

Os envolvidos, juntamente com os líderes do time de desenvolvimento, devem priorizar cada requisito para organizar uma ordem lógica e aceita para atingir os objetivos de negócios da empresa. Portanto, você deve explicar aos gestores da CaféSaúde como efetuar essa priorização de requisitos.

Para as metodologias de desenvolvimento de software, ao realizarem as tarefas da análise de sistemas, são gerados artefatos para documentar os requisitos de software. Você deve explanar aos gestores como serão as tarefas de análise de sistemas na CaféSaúde. Ou seja, os gestores precisam conhecer melhor as responsabilidades do analista de sistemas, do engenheiro de software e do time de operações.

Como você orientará os gestores a controlar as atividades de classificação, priorização de requisitos na CaféSaúde?

Você está apto para explicar a importância da classificação e priorização de requisitos de software?

Você consegue explicar a análise de caso de uso de requisitos e o levantamento ágil de requisitos?

Algumas documentações são extensas, outras são custosas para confeccionar. Porém, para o método ágil, são descrições mínimas, pois grande parte do conhecimento é compartilhado entre os stakeholders por meio de reuniões, entrevistas, questionários, revisões, entre outros encontros.

O importante para o ágil é ter o sistema em funcionamento, como o mais desejado dos artefatos, uma coleção de código-fonte, banco de dados, componentes lógicos ou eletrônicos, que sejam úteis aos negócios e conforme expectativas dos stakeholders.

Para o time de desenvolvimento de software compreender a realidade da organização, portanto, serão demonstrados os cenários com as atividades da colheita, simulando as anotações dos dados na realização dessa tarefa. À medida que atividades reais estão acontecendo, os analistas e engenheiros de software devem assimilar o máximo possível, anotando apenas os dados essenciais para dar continuidade. À análise do sistema para propor uma solução ou melhorias em suas rotinas.

De posse de uma lista com todos os requisitos, os envolvidos devem se reunir para priorizar cada requisito de software. Cada requisito recebe uma classificação, conforme ilustrado na Figura 1.

		URGENTE	POUCO URGENTE
		FAZER IMEDIATAMENTE	AGENDAR PARA FAZER
IMPORTANTES	IMPORTANTES		
	POUCO IMPORTANTES		
POUCO IMPORTANTES	IMPORTANTES		
	POUCO IMPORTANTES		

Matriz de Eisenhower - Priorização

Fonte: adaptado de <https://g4educacao.com/blog/matriz-de-eisenhower>

Figura 1 | Matriz de priorização. Fonte: adaptado de Matriz de Eisenhower – Priorização.

Na CaféSaúde, temos alguns requisitos listados. Compreendemos como ficará a priorização entre eles, conforme as argumentações de cada envolvido:

- Req#01 – cadastro de colhedores de café.
- Req#03 – relação de colhedores ativos.
- Req#05 – classificação e atualização dos dados do colhedor de café.
- Req#08 – entrada de dados da colheita de café, formando um lote colhido.
- Req#09 – atualização dos dados da colheita de café de cada lote colhido.
- Req#13 – transferência do lote colhido da fase de colheita para fase de secagem.

Vejamos a adaptação elaborada para atender aos requisitos de software, demonstrada na Figura 2. O Req#01 deve ser um dos primeiros, por ser a base dos demais requisitos; os Req#08 e Req#09 são diretamente dependentes do Req#01. Conforme o foco do negócio para esse projeto é formar um histórico de dados da colheita, o Req#08 passa a ser prioridade, logo que o Req#01 estiver implantado.

Percebemos que a análise de requisitos para a priorização é realizada com todos os requisitos compreendidos e aprovados pelos stakeholders.

	URGENTE	POUCO URGENTE
IMPORTANT	<b>IMPORTANTE NOS NEGÓCIOS</b>	<b>VALOR PARA O CLIENTE</b>
	 Req#01 – cadastro de colhedores de café   Req#08 – entrada de dados da colheita de café, formando um lote colhido	 Req#13 – transferência do lote colhido da fase de colheita para fase de secagem
POUCO IMPORTANTE	<b>IMPACTO E URGENTE</b>	<b>COMPLEXO E DEPENDENCIA</b>
	 Req#09 – atualização dos dados da colheita de café de cada lote colhido	 Req#03 – relação de colhedores ativos   Req#05 – classificação e atualização dos dados do colhedor de café

Figura 2 | Matriz de prioridades de requisitos de software CaféSaúde.

Imaginemos que, três dias após a rodada de priorização, surge o requisito de software Req#06 – Classificação do lote colhido para secagem. Pensemos juntos: esse requisito Req#06 não é novo, ele estava sendo elicitado, aliás vários outros estão sendo elicitados e aprovados pelos stakeholders, concomitantemente aos demais que estão na “fila” para implementação. Assim, agora, com o Req#06 na lista, será necessário analisá-lo e priorizá-lo frente aos demais. Essa é a dinâmica comum no ambiente de desenvolvimento com o método ágil.

Todos os requisitos de negócios – aqueles que atendem diretamente aos objetivos estratégicos – são analisados para se transformarem em funcionalidades de um software, para os quais

nomeamos como requisitos funcionais. Eles serão úteis nas operações dos usuários finais, bem como percebidos pelos resultados proporcionados, por representarem as regras de negócios da organização.

Relembremos dos fatores proporcionados pela tecnologia da informação (TI), chamados de **requisitos não funcionais**. São eles os responsáveis por impulsionar a eficiência, facilitar operações dos usuários, garantir conformidade com as normas e qualificar as execuções dos requisitos funcionais, prolongar vida útil do sistema. Para o sistema na CaféSaúde podemos ter os requisitos não funcionais:

- **Desempenho:** eficiência do sistema. Na integração dos dados da colheita, do processo de "transferência do lote colhido da fase de colheita para fase de secagem" em 500 transações por segundo.
- **Confiabilidade:** a disponibilidade deve estar em 99% em épocas de colheita do café, sendo necessário recuperar de falhas críticas em 3 minutos.
- **Usabilidade:** usuários devem realizar uma transação em até 3 cliques, principalmente, na operação entrada de dados de colheitas. Oferecer suporte a deficientes visuais seguindo as diretrizes de acessibilidade [WCAG](#).
- **Auditabilidade:** possuir capacidade de rastrear e garantir conformidade das operações da CaféSaúde, registradas com o carimbo de data e hora. Com Logs armazenados por um ano, no mínimo.

Como sabemos, em qualquer metodologia, é importante recorrer aos processos de classificação de requisitos, pois ele permite a compreensão do papel de cada requisito de software, consequentemente permite decidir quais funcionalidades serão priorizadas; auxilia a antecipar e gerenciar requisitos voláteis; ainda, torna claro o que será implementado em cada fase, alinhando as expectativas dos stakeholders e da equipe técnica.

Compreendamos uma das técnicas largamente utilizada para a priorização de requisitos (MoSCoW):

- **Essenciais (Must-have):** imprescindíveis para o sistema funcionar, diretamente relacionado ao controle de colheitas, processo de secagem e aqueles requisitos que controlarão a qualidade do café.
- **Importantes (Should-have):** agregam valor considerável aos negócios, no fluxo do produto desde o talhão plantação do cafezal até o empacotamento do café moído, mesmo não sendo os críticos; eles podem ser postergados, somente se houver limitações de prazo ou orçamento.
- **Desejáveis (Could-have):** benéficos, mas que possuem baixa prioridade. Auxiliam os gestores da CaféSaúde na gestão do processo.
- **Não necessários (Won't-have):** requisitos que estão sendo coletados ou que não fazem parte do escopo atual do projeto; um dia poderão ser reclassificados como sendo mais importantes.

Estudante, o objetivo desse estudo de caso foi facilitar o entendimento da classificação e priorização de requisitos, as diversas abordagens e cuidados para fazer a escolha adequada aos objetivos de negócios da CaféSaúde ou qualquer organização. Continue aprimorando a sua habilidade em compreensão de requisitos de software, atualizando-se com novas técnicas e abordagens.

# ENGENHARIA DE REQUISITOS

Bons estudos!

Acesse o infográfico utilizando o QRCode por qualquer dispositivo para compreender a extensão do conteúdo desta unidade da disciplina de engenharia de requisitos.

## ENGENHARIA DE REQUISITOS

Prof. Marco Hisatomi



[https://bit.ly/engreq\\_info2\\_classificarequisitos](https://bit.ly/engreq_info2_classificarequisitos)

### CLASSIFICAÇÃO DE REQUISITOS DE SOFTWARE – UN2

- Identificar e classificar requisitos de softwares.
- Explicar a Priorização de Requisitos.
- Descrever a Análise de Requisitos.
- Explicar os Artefatos de Requisitos.

#### Classificação de Requisitos



Conhecerá a Classificação de Requisitos, conteúdo importante para a sua prática profissional, compreendendo os Processos de Classificação de Requisitos, Requisitos Funcionais e Requisitos Não Funcionais

#### Priorização de Requisitos

Conheça a Priorização de Requisitos. Conteúdo importante para a sua prática profissional, pois compreenderá sobre os Fundamentos de Priorização de Requisitos, os Critérios de Priorização de Requisitos e as Técnicas de Priorização de Requisitos



#### Análise de Requisitos



Conheça a Análise de Requisitos, conteúdo importante, pois compreenderá sobre a características de Caso de Uso de Requisitos, dos Cenários de Uso de Requisitos e de Requisitos no Ciclo de Vida do Projeto

#### Modelagem de Requisitos

Conhecerá a Modelagem de Requisitos, conteúdo importante para atuar profissionalmente, pois compreenderá sobre os Artefatos do Elicitação de Requisitos, Levantamento Ágil de Requisitos e Sistemas Autoadaptativos



Avalie!

[https://bit.ly/engreq\\_aval\\_un2](https://bit.ly/engreq_aval_un2)

VOCÊ APRENDE MAIS QUANDO ESTUDA DENTRO DO SEU ESTILO

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos**: software orientado ao negócio. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

## Unidade 3

### Gerenciamento de Requisitos

#### Aula 1

##### Processos de Gerenciamento de Requisitos

##### Processos de gerenciamento de requisitos

###### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer os processos de gerenciamento de requisitos. Esse conteúdo é importante para a sua prática profissional, pois contempla os riscos em mudanças de requisitos, processos de qualidade na engenharia de requisitos e a revisão de requisitos.

Prepare-se para esta jornada de conhecimento! Vamos lá!!!

#### Ponto de Partida

Os gestores da CaféSaúde estão empenhados no desenvolvimento do sistema de gerenciamento da cadeia produtiva de café Standard, Premium e High Special. Para tanto, você ficará responsável por explicar a eles como se aplicará o processo de gerenciamento de requisitos, envolvendo riscos em mudanças de requisitos (impactos no processo de desenvolvimento); processos de qualidade na engenharia de requisitos; e revisão de requisitos.

Estudante, continue pesquisando e atualizando seu conhecimento sobre os processos de gerenciamento de requisitos para aplicar no cotidiano profissional.

Bons estudos!

## Vamos Começar!

Estudante, bem-vindo ao conteúdo do processo de gerenciamento de requisitos!

O gerenciamento de requisitos de software visa a realização de atividades, ao longo do ciclo de vida do projeto de software, interconectadas e organizadas para assegurar que o produto seja entregue conforme esperado pelos stakeholders.

Normalmente, as atividades que tratam de requisitos são: elicitação, análise, documentação, priorização, validação, rastreabilidade, entre outras, dependendo da metodologia adotada. O gerenciamento de requisitos tem como objetivo planejar e controlar os recursos a serem utilizados, bem como providenciar os softwares/ferramentas quando necessário, e remover os obstáculos que surgirem durante a atividade.

## Riscos em mudanças de requisitos

As novas tecnologias estão à disposição para agregar valores aos negócios de uma organização, causando impacto positivo assim que um novo sistema é colocado em operação. Porém, elas são renovadas (ou inovadas) rapidamente, levando à necessidade de adequações no sistema, ou seja, devemos calcular os riscos e seus impactos em função dessas mudanças.

Além da atualização tecnológica, podem ser causadores de mudanças de requisitos as necessidades específicas dos usuários finais, as demandas pelas condições impostas pelo mercado e a aderência às conformidades com os padrões, as legislações ou as normatizações.

Os gestores de projetos de softwares estão preparados para gerenciar as mudanças de requisitos por processos de análise de riscos, principalmente aqueles quem impactam negativamente o sucesso do projeto. Compreendamos os riscos mais comuns na mudança de requisitos:

- 1. Aumento do escopo (scope creep):** quando novos requisitos são acrescentados ao projeto, podendo impactar em atrasos nas entregas, sobrecarga no time de desenvolvimento ou diminuição da qualidade do software.

2. **Replanejamento do cronograma:** podem ocorrer mudanças de requisitos já definidos, causando alteração no planejamento original, principalmente em cronograma, podendo impactar em atrasos nas entregas, sobrecarga no time de desenvolvimento, ou diminuição da qualidade do software.
3. **Acréscimo nos custos do projeto:** os requisitos alterados que provocam revisão na arquitetura/design do software, podendo impactar em sobrecarga no time de desenvolvimento, aumento de custos com retrabalho, ou novas implementações (não planejadas anteriormente) e atrasos nas entregas.
4. **Inconsistência em requisitos:** as mudanças mal gerenciadas em requisitos com alterações frequentes podem tornar os requisitos inconsistentes, provocando erros na implementação, introdução de bugs no sistema e produto final invalidado.
5. **Resistência do time de desenvolvimento:** as mudanças constantes e recorrentes nos mesmos requisitos geram frustrações ou resistência pelos membros do desenvolvimento, provando queda de produtividade e qualidade do sistema, distanciamento na comunicação entre equipe técnica e stakeholders.

Conhecendo as possíveis causas de riscos indesejados no projeto, tanto o analista de sistemas quanto os engenheiros de software devem elaborar um processo para evitar tais riscos.

A metodologia ágil, por si, ameniza bastante os riscos. Em virtude dos princípios que mantêm a comunicação diária e aproximação com os responsáveis pelos requisitos de software, o time de desenvolvimento recebe as novas informações rapidamente e com muita precisão. Os processos de revisão contínua dos requisitos, classificados e priorizados, facilitam a compreensão e análise de impactos dos novos processos e das mudanças de requisitos. O método incremental e iterativo, com ciclo rápido, permite gerenciar melhor as mudanças, mantendo o backlog adaptado aos negócios continuamente, conforme necessário. As entregas de ciclo curto também contribuem na implementação de requisitos consistentes e na análise contínua do código.

Vejamos alguns procedimentos para o gerenciamento de riscos que podem contribuir para evitar os impactos negativos:

- **Planejamento para controlar mudanças:** formar e definir especialistas para analisar, aprovar (ou rejeitar) as mudanças de requisitos.
- **Análise da mudança:** utilizar ferramentas de rastreamento de requisitos para analisar o impacto das mudanças, quanto à extensão e complexidade da alteração.
- **Avaliação de impacto:** avaliar os impactos técnicos, financeiros e no cronograma, envolvendo todos os stakeholders para essa avaliação.
- **Priorização das mudanças:** foco em requisitos mais críticos para o sucesso dos negócios/projeto, baseado em classificação das mudanças em urgente, impactante e viável.
- **Gerenciamento das expectativas:** obter a ciência dos stakeholders quanto ao impacto das mudanças, inclusive clientes de limitações ou restrições de ajustes inviáveis devido ao prazo ou orçamento.
- **Comunicação eficaz:** informar todos os impactos percebidos após análise e avaliação de impacto e obter aprovação formal para iniciar as implementações de mudanças.

Os riscos são introduzidos ao projeto proveniente das inevitáveis mudanças em requisitos de software, portanto o gerenciamento adequado evitará impactos negativos ou prejuízos no projeto e aos negócios. Assim, devemos adotar processos com rastreabilidade e comunicação clara e eficiente com os stakeholders para auxiliar na mitigação dos riscos, contribuindo com o sucesso do projeto, mesmo com as mudanças constantes.

## Processos de qualidade na engenharia de requisitos

A qualidade na engenharia de requisitos é percebida de maneira subjetiva, pois dependemos do nível de satisfação das pessoas, dos stakeholders e dos usuários finais, que avaliam um software após a entrega e com critérios próprios.

Para garantirmos um mínimo de satisfação nas entregas de sistemas, devemos definir processos claros a todos os envolvidos no processo de desenvolvimento, quanto aos procedimentos de elicitação, modelagem, validação e análise de requisitos de software, diretamente relacionados à engenharia de requisitos. Assim, minimiza-se os riscos de mal entendidos sobre o que será desenvolvido e entregue.

Lembremos os destaques, de Pressman e Maxim (2021, p. 312), sobre os fatores relevantes em processos de desenvolvimento de software:

“uma gestão de qualidade efetiva estabelece a infraestrutura que dá suporte a qualquer tentativa de construir um produto de software de alta qualidade”.

Sobre requisitos, especificamente, Pressman e Maxim (2021, p. 312) complementam:

“ele satisfaz a um conjunto de requisitos implícitos (p. ex., facilidade de uso) que se espera de todo software de alta qualidade”.

Dentre os pontos da qualidade, complementam: “isso permite que os engenheiros de software despendam mais tempo criando aplicações novas e menos tempo em manutenções” e, ainda, “a comunidade de usuários ganha um valor agregado, pois a aplicação fornece a capacidade de agilizar algum processo de negócio”.

Abordaremos o processo visando uma contínua melhoria na engenharia de requisitos, conforme ilustra a Figura 1.

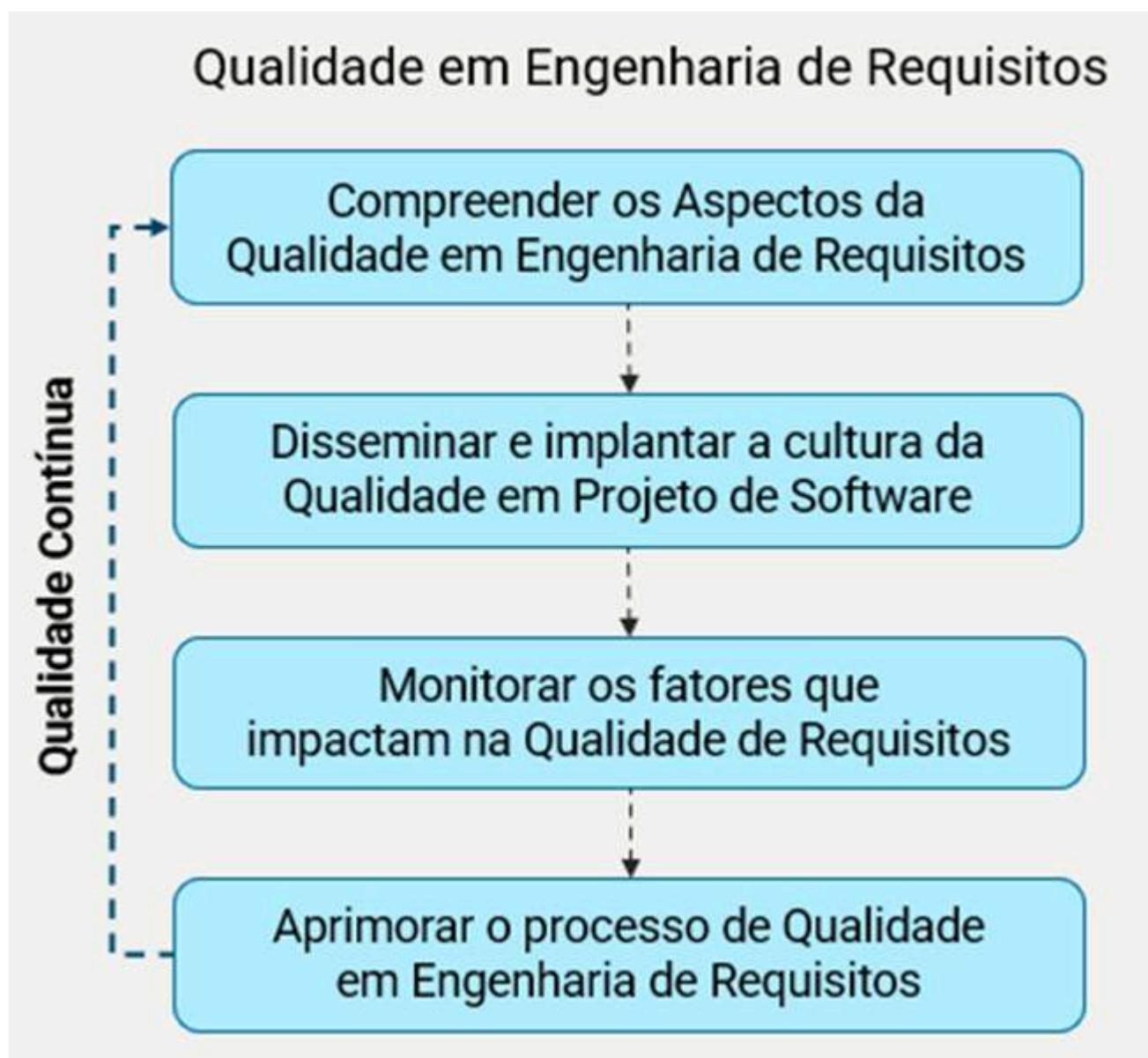


Figura 1 | Qualidade em engenharia de requisitos.

Os principais aspectos da qualidade estão relacionados aos objetivos que devemos perseguir na gestão das atividades da engenharia de software, envolvendo artefatos de requisitos de software, o time de desenvolvimento e demais stakeholders:

- **Compreensibilidade de requisitos:** quando as pessoas envolvidas compreendem o que foi definido, como será validado e percebem valor agregado ao software, ou seja, não existe ambiguidade, mas são objetivos mensuráveis.
- **Consistência entre requisitos:** não devem conter conflitos internos ou com outras especificações do projeto e, ao contrário, devem estar alinhados aos objetivos do negócio, sob o escopo do sistema, sem deixar lacunas nos processos de negócio.
- **Prioridade no desenvolvimento:** devem ser classificados conforme a sua importância, relevância e urgência, auxiliando na alocação de recursos e no planejamento incremental

da implementação.

- **Rastreabilidade:** o requisito deve ser rastreável, permitindo sua conexão com o código-fonte e demais artefatos: modelagem do requisito, testes, design e registros de validação/revisão; apoiado em ferramentas de rastreamento (p. ex., matrizes de rastreabilidade).

Todos os principais aspectos da qualidade definidos devem ser mantidos em funcionamento, até que eles sejam redefinidos. Portanto, os membros do time de desenvolvimento devem conhecer o processo de monitoramento do processo que impactam na qualidade do requisito:

Comunicação eficiente: garantir que o time de desenvolvimento, representantes do produto de software e demais stakeholders se comuniquem suficientemente, para manter todos os requisitos conforme os aspectos da qualidade.

Gerenciamento das mudanças: a frequência e extensão da mudança em requisitos não devem impactar na qualidade do requisito e do projeto em desenvolvimento; devem manter os aspectos da qualidade sempre que alterar um requisito e quando adicionar novo requisito devem classificá-lo e priorizá-lo.

Atualização das ferramentas: os processos baseados em ferramentas devem estar atualizados tecnologicamente e operacionalmente; as pessoas devem se manter aptas a usá-las, conforme especificado no processo.

Atendimento às expectativas dos stakeholders: o monitoramento das expectativas dos clientes e usuários finais e as respostas aos feedbacks dos stakeholders são fundamentais para reconhecer e manter o nível de qualidade dos requisitos.

## Siga em Frente...

## Revisão de requisitos

A revisão de requisitos de software é fundamental para conferir se cada requisito se encontra alinhado aos objetivos de negócio, com as especificações necessárias ao projeto e conforme às expectativas dos stakeholders. É considerada uma etapa crítica, pois deve garantir que os requisitos estejam claros, completos, consistentes e validados.

Essa prática deve reduzir riscos, evitar retrabalho e assegurar que todos os envolvidos tenham uma compreensão compartilhada do que será desenvolvido.

A revisão técnica formal (RTF) é uma atividade de controle da qualidade de software e envolve todos os stakeholders que definiram os requisitos, construíram o software e aqueles que utilizarão o sistema, incluindo o time da Tecnologia de Informação (TI), conforme Pressman (2021, p. 332) explica:

Os objetivos de uma RTF são: (1) descobrir erros na função, lógica ou implementação para qualquer representação do software; (2) verificar se o software que está sendo revisado atende aos requisitos; (3) garantir que o software foi representado de acordo com padrões predefinidos; (4) obter software desenvolvido de maneira uniforme; e (5) tornar os projetos mais gerenciáveis.

Na engenharia de software, as RTF são consideradas, de certa forma, um compartilhamento de conhecimento aos engenheiros de software e analistas de sistemas menos experientes, além de ser um momento para observar as diferentes estratégias para a análise, projeto design e implementação.

Compreendamos como você pode se preparar para uma revisão, conforme algumas diretrizes para a revisão de requisitos:

- **Engajamento dos stakeholders:** nem sempre todos os envolvidos estão engajados como se espera no método ágil, incluindo o representante do produto (p.ex., Product Owner no Scrum), mas a participação ativa dos stakeholders garantirá maior qualidade dos requisitos de software.
- **Controle do cronograma:** o tamanho de requisito, as alterações que ele sofre e a complexidade das inúmeras modificações podem comprometer o prazo do projeto, portanto nessa revisão deve-se perceber se o prazo continua condizente ao planejado.
- **Controle de riscos:** a revisão deve ter foco nos atributos da qualidade dos requisitos de software, assim minimizará os riscos indesejáveis no projeto de software.

Concluindo, a revisão de requisitos executada adequadamente garantirá que os resultados de software estejam alinhados aos objetivos do negócio e viáveis de serem implementadas, além de reduzir os riscos, melhorar a qualidade do produto final e promover a satisfação de stakeholders, sendo uma prática indispensável na engenharia de requisitos.

Estudante, os processos de gerenciamento de requisitos auxiliam na condução do projeto de desenvolvimento, apoiados pela gestão de riscos, do gerenciamento da qualidade e da revisão de requisitos; continue pesquisando e estudando sobre o tema.

Bons estudos!

## Vamos Exercitar?

Percebemos a dinâmica das atividades em um projeto de software, na CaféSaúde, quanto ao gerenciamento de projetos, com foco em riscos em mudanças de requisitos, processos da qualidade e revisão de requisitos.

Segue o comunicado do analista de sistemas da CaféSaúde: “Fica agendada a reunião de revisão de requisitos para a próxima segunda-feira, logo após a reunião diária, quando serão apresentados os detalhes em relação aos riscos percebidos durante as atividades de análise de requisitos com o time de desenvolvimento”.

Para comunicar a todos envolvidos sobre os riscos na mudança do requisito de apontamento da colheita, acompanhemos os passos:

1. O usuário final, supervisor de campo, percebeu que, ao incluir os dados da colheita, em especial a temperatura do ambiente, ele precisava digitar o dado a cada lote-colhido. Portanto, fez a solicitação de alteração na entrada de dados (Req#127) para preservar o mesmo valor apontado no primeiro lote-colhido do dia.
2. Ao receber a solicitação de mudança, o analista de sistemas verificou que essa característica faz sentido para o procedimento de entrada de dados do lote-colhido. No entanto, argumentou com os envolvidos, que esse comportamento não funcional (Não funcional - aqui o sentido é de requisito não funcional por se referir a um aspecto de produtividade (não de uma regra de negócio), apenas uma facilidade promovida pelo recurso da tecnologia da informação) do sistema poderia permitir informar a temperatura incorreta, uma vez que, por volta das onze horas, no apontamento de N-ésimo lote-colhido, a temperatura poderia estar diferente da apontada no primeiro momento do dia.
3. Em seguida, a análise do engenheiro de software ponderou a possibilidade de conectar o aplicativo diretamente a um sensor de temperatura (Sobre o sensor de temperatura. Os sensores de temperatura de ambiente podem ser utilizados em diversas aplicações, como: monitoramento de ambientes externos, monitoramento de procedimentos de produção, ou sistemas de climatização). O que poderia prevenir erros de dados informados pelas pessoas e evitar a implementação de funcionalidades inconsistentes com os objetivos organizacionais, nesse caso, inefficientes.

Percebemos que o envolvimento das pessoas (time de desenvolvimento e stakeholders) podem influenciar diretamente na qualidade do sistema em desenvolvimento, bem como no próprio processo de desenvolvimento. Ao executar a atividade de “monitorar os fatores que impactam na qualidade de requisitos” foi detectado que o Req#127 poderia impactar negativamente o projeto.

As consequências de uma implementação mal elaborada e mal analisada podem ser prejudiciais para o sistema por impactar em retrabalho, falta de engajamento dos stakeholders, entre outros fatores.

Assim, a reunião de revisão foi útil para continuar analisando os requisitos funcionais e não funcionais, de maneira aprofundada, envolvendo todos os stakeholders, acrescido de pensamentos estratégicos dos negócios e do foco em qualidade na engenharia de requisitos.

Estudante, com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos, com foco em gerenciamento de requisitos.

Bons estudos!

**Saiba mais**

## I. Saiba mais sobre: riscos em planejamento de projetos

Leia sobre “O grau de incerteza estrutural” e “escopo do projeto é mal entendido”, na página 505, no item 25.1, *Comentários sobre as estimativas*.

### [Engenharia de software](#)

## II. Saiba mais sobre: revisão

Leia o item 16.6, *Revisões técnicas formais*, na página 332.

### [Engenharia de software](#)

## III. Saiba mais sobre: revisão de requisitos

Leia, na página 210, o item 1.3: *Processo de revisão e revisão de requisitos*.

### [Guide to the Software Engineering Body of Knowledge v4.0](#)



## Referências

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos: software orientado ao negócio**. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

WASHIZAKI, H. **Guide to the Software Engineering Body of Knowledge v4.0**. Waseda University: 2024. Disponível em: [https://bit.ly/engreq\\_swebokv4](https://bit.ly/engreq_swebokv4) Acesso em: 15 nov. 2024.

## Aula 2

Gestão de Alteração de Requisitos

### Gestão de alteração de requisitos

#### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

##### Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer a gestão de alteração de requisitos. Esse conteúdo é importante para a sua prática profissional, pois você vai entender como gerenciar os relacionamentos entre os requisitos, a padronização na gestão de alteração de requisitos e os impactos no projeto.

Prepare-se para esta jornada de conhecimento! Vamos lá!!!

### Ponto de Partida

Os gestores da CaféSaúde estão empenhados no desenvolvimento do sistema de gerenciamento da cadeia produtiva de café Standard, Premium e High Special. Para tanto, você ficará responsável por explicar a eles como será a gestão de alteração de requisitos, o gerenciamento

dos relacionamentos entre os requisitos, padrões na gestão de alteração requisitos e os impactos no projeto.

Estudante, com o aprendizado deste conteúdo, você estará cada vez mais preparado para atuar em engenharia de requisitos. Continue pesquisando e atualizando seu conhecimento sobre os processos de gerenciamento e alterações de requisitos para aplicar no seu cotidiano profissional.

Bons estudos!

## Vamos Começar!

Estudante, bem-vindo ao conteúdo de gestão de alteração de requisitos, que visa eficiência em melhoria na qualidade do produto final, redução de retrabalho, controle de custos e prazos. As mudanças gerenciadas estruturadamente evitam inconsistências e prejuízos por esforços indevidos e proporcionam o aumento da satisfação dos stakeholders.

## Gerenciar os relacionamentos entre os requisitos

O gerenciamento dos relacionamentos entre os requisitos pode ser realizado conforme as prioridades entre os requisitos e pela obrigatoriedade do relacionamento entre eles. Vejamos algumas perspectivas que justificam os relacionamentos entre os requisitos, ilustrado na Figura 1.

**Requisitos derivados.** Segundo SWEBOK® GUIDE V4.0 (2024), na prática, os requisitos podem ser sensíveis ao contexto. P. ex., um stakeholder externo pode impor um requisito de escopo, que seria um projeto de um requisito, podendo envolver dezenas de engenheiros de software; poderíamos obter vários requisitos derivados menores e gerenciáveis, ao invés de tentar gerenciar apenas um requisito com várias funcionalidades.

**Requisitos conflitantes.** Quando um projeto possui stakeholders diversos, existe probabilidade de ocorrer conflitos entre requisitos. Pela abordagem de conflito de requisitos é possível gerenciar o relacionamento entre os requisitos, segundo SWEBOK® GUIDE V4.0 (2024). Geralmente, os engenheiros de software utilizam-se da negociação para resolução dos requisitos conflitantes diretamente com os envolvidos. Assim, é provável encontrar uma decisão em consenso, mas poderia ser resolvida por questões contratuais. Outra possibilidade de conflito seria entre o requisito desejado para o sistema e o requisito que pode ser realizado (restritos por custo, cronograma, entre outros). Um exemplo de requisito conflitante, seria o uso de unidade de medida para temperatura, em graus °C (Celsius) e em °F (Fahrenheit).

**Correspondência ao escopo.** Em SWEBOK® GUIDE V4.0 (2024), a correspondência ao escopo obriga os responsáveis a compreender e reconhecer os requisitos relacionados. Portanto, quando ocorrer restrições em projetos por custo, cronograma, tecnologia, entre outros, evidentemente haverá uma análise detalhada dos requisitos de software quanto às prioridades.

Possivelmente serão removidos os requisitos de prioridade mais baixa quanto à correspondência, como escopo, removendo também aqueles requisitos dependentes em profundidade ou correlatos pela amplitude.

**Natureza iterativa.** O gerenciamento do relacionamento entre requisitos é importante para garantir a eficiência na gestão. Ao utilizarmos a natureza iterativa, conforme o SWEBOK® GUIDE V4.0 (2024): quando possuem características de amplitude entre os requisitos correlatos, ou de profundidade entre os requisitos dependentes, a cada iteração do desenvolvimento, otimizaremos recursos em esforços de pessoal e em prazo, quando efetuarmos a expansão na amplitude dos requisitos. Essa mesma otimização pode ocorrer quando efetuarmos a expansão por requisitos dependentes.

**Técnicas de elicitação.** Os requisitos podem sofrer interpretações diferentes sob a perspectiva das técnicas de elicitação, explanado em SWEBOK® GUIDE V4.0 (2024), portanto, quando stakeholder participa em várias tarefas da elicitação, daria a ele uma visão holística, com a possibilidade de identificar requisitos relacionados ou dependentes entre si. Por exemplo, a entrevista garante uma visão aprofundada, porém de apenas uma das partes; o brainstorming traz perspectivas diversas e aspectos que podem ser verificados somente com várias pessoas juntas; ou o planejamento conjunto de requisitos (JRP - joint requirements planning), possibilitando requisitos relacionados.

**Rastreamento de requisitos.** Bastante útil no gerenciamento do projeto e no gerenciamento de mudanças de requisitos ou no gerenciamento de requisitos relacionados. Os requisitos podem ser rastreados entre si quando estiver especificado em um artefato, evidenciando o relacionamento entre os requisitos. Da mesma forma, podemos identificar se existem componentes de design destinados a satisfazer o requisito, como também podemos identificar quais requisitos são atendidos por um componente de design.

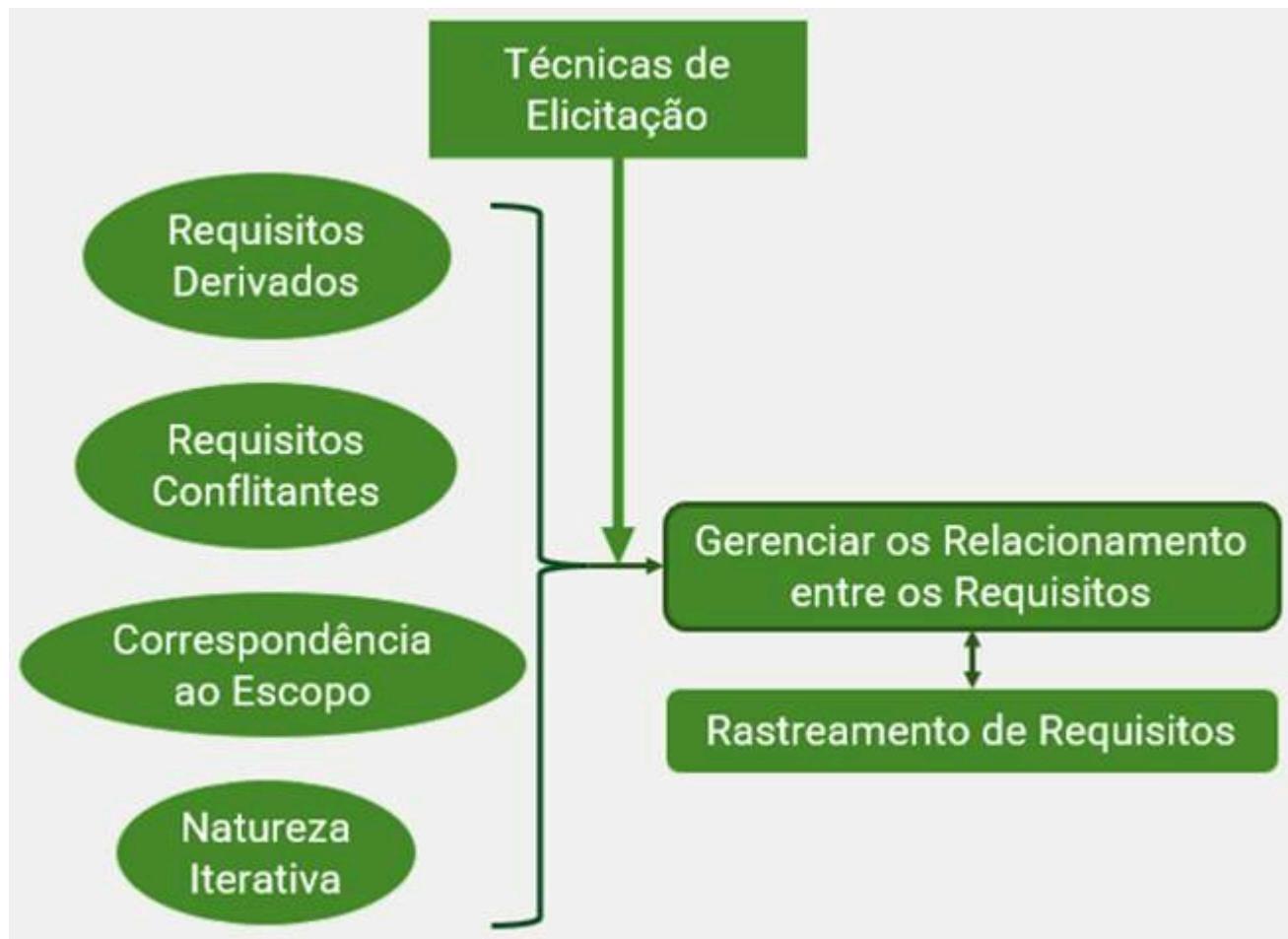


Figura 1 | Gerenciar o relacionamento entre requisitos.

Aqui, alertamos que se faz necessário o gerenciamento dos requisitos relacionados quanto às perspectivas elencadas para garantir que estão sendo implementados seguindo os critérios da priorização e do alinhamento aos objetivos do negócio. Ressaltamos que podem surgir outras perspectivas a depender do contexto.

O relacionamento entre requisitos de software pode ser refletido na gestão de configuração de software, quando os artefatos estão identificados e controlados por sua relação de dependência, derivação e sucessão. Nesse caso, o analista de sistemas deve consultar o repositório de artefatos da gestão de configuração para descobrir possíveis relacionamentos entre requisitos.

## Padronização na gestão de alteração requisitos

O gerenciamento de alteração de requisitos é essencial e a sua eficiência será proporcional ao sucesso de projetos de software. O objetivo, ao longo do ciclo de vida do projeto, é que as alterações sejam tratadas de maneira controlada e estratégica, minimizando riscos e maximizando a qualidade do produto final (SWEBOK® GUIDE V4.0, 2024).

Deve ficar esclarecido aos membros do time de desenvolvimento todos os passos do gerenciamento de alteração de requisitos, como um padrão de processos para gerenciar tais mudanças. Vejamos:

- **Definir um processo formal:** estabelecer atividades claramente definidas para registrar, avaliar e implementar as mudanças; qualquer alteração de requisito de software, seja dos requisitos já implementados ou de outros que continuam aguardando a codificação, devem ser submetidas ao processo.
- **Examinar alteração e prioridade:** definir rigor nos critérios de alterações de requisitos, quanto às prioridades definidas em cada um dos requisitos. A alteração de um requisito poderá ter a sua prioridade trocada, o que acarretará trocas dos demais requisitos.
- **Envolver stakeholders:** a responsabilidade da alteração de requisitos e da troca de prioridade deve ser um processo decidido por todas as partes interessadas, mesmo que a solicitação não tenha sido iniciada pelo usuário final.
- **Automatizar rastreamento:** o uso de ferramentas (softwares) para rastrear alterações e documentar seu impacto é fundamental para agilizar futuras consultas, proporcionando maior eficiência no gerenciamento de alterações de requisitos.
- **Comunicação eficiente:** mesmo que os envolvidos com o requisito aprovem as alterações no requisito, é fundamental comunicar regularmente aos stakeholders sobre mudanças, seu status e impactos analisados e esperados.

Outro passo necessário em resposta às alterações de requisito é a concordância dos stakeholders e do cliente sempre que ocorrer impacto no cronograma, recursos e/ou mudança proporcional no escopo em outras partes do projeto, orçamento do projeto, entre outros.

Lembremo-nos de que a gerenciamento de alteração de requisitos é percebido pelo método ágil de maneira natural, segundo Pressman (2021):

- O fundamental do ágil: “é difícil prever quais requisitos de software vão persistir e quais sofrerão alterações”, esclarecendo que as alterações são inevitáveis. Ainda, segue “é igualmente difícil prever de que maneira as prioridades do cliente sofrerão alterações conforme o projeto avança”, fortalecendo uma cultura de disposição para atender aos desafios das mudanças de requisitos.
- O time ágil deve ficar atento aos feedbacks do cliente, pronto para as adaptações apropriadas, incrementalmente.
- O sistema funcionando a cada iteração, sob rigoroso crivo do time de operações, é um catalisador eficaz para o feedback do cliente.
- Em ritmo constante, as mudanças são imprevisíveis, a cada iteração, exigindo os incrementos em períodos curtos (três a quatro semanas).
- A rigidez do timeboxing, período fixo planejado para a iteração com entrega do executável, capacita o cliente a avaliar o incremento de software regularmente e a dar o feedback necessário ao time ágil.

Em SWEBOK® GUIDE V4.0 (2024), no ciclo de vida curto e constante do ágil, a solicitação de alteração de requisitos, previamente acordados torna-se mais um item no backlog do produto, e

o “aceite” é dado com a priorização desse item, quando, então, o requisito poderá ser escolhido para ser implementado.

## Siga em Frente...

## Impactos no projeto

Avaliar o impacto das mudanças permite que decisões sejam tomadas com base em informações completas, evitando surpresas no cronograma, no orçamento ou na qualidade do produto final.

Vejamos alguns “gatilhos” que motivam as alterações de requisitos em projeto de software:

- **Mudanças nos requisitos de negócio:** stakeholders revisam suas prioridades, identificam novas necessidades de negócios e pressões da concorrência ou surgem problemas operacionais ainda não previstos.
- **Novas regulamentações ou conformidades:** precisa atender a requisitos legais ou normativos que surgiram (ou se alteraram) após o início do projeto.
- **Feedbacks:** testes, protótipos ou avaliações iniciais podem revelar a necessidade de ajustes de requisitos.
- **Condições externas:** avanços tecnológicos, ou atualizações em tecnologia implementadas.

As alterações de requisitos podem provocar reanálise em demais requisitos. Além desse impacto, os impactos no projeto de software podem ser em vários aspectos:

- **Arquitetônico:** sendo um dos mais importantes, pois se afeta a arquitetura do software, a manutenção se torna custosa e demanda esforço em demasia. Para SWEBOK® GUIDE V4.0 (2024), as decisões diferidas podem comprometer sua sustentabilidade ou a evolução futura do software, entre outros: refinamento em larga escala do sistema em componentes-chave, comunicação e interação entre componentes, interfaces de componentes.
- **Satisfação do cliente:** quando gerenciadas adequadamente, mudanças podem aumentar a satisfação do cliente; caso contrário, podem causar frustração.
- **Escopo:** podem ampliar ou reduzir o escopo original do projeto, afetando as entregas e os resultados esperados.
- **Cronograma:** podem atrasar prazos planejados, especialmente se envolvem atividades não previstas.
- **Orçamento:** pode aumentar custos, seja por necessidade de mais recursos, ferramentas ou tempo adicional.
- **Qualidade:** alterações mal gerenciadas podem comprometer a qualidade do produto final, resultando em retrabalho ou falhas.
- **Recursos humanos:** novas habilidades ou mais esforço podem ser necessários, gerando sobrecarga ou necessidade de treinamento.
- **Comunicação e alinhamento:** alterações frequentes ou mal comunicadas podem gerar desalinhamento entre equipes e stakeholders.

Estudante, os processos de gerenciamento das alterações de requisitos auxiliam na condução do projeto de desenvolvimento, por gerenciar os relacionamentos entre os requisitos, na padronização na gestão de alteração requisitos e impactos no projeto; continue pesquisando e estudando sobre a importância do gerenciamento de alterações de requisitos.

Bons estudos!

## Vamos Exercitar?

Na CaféSaúde, as alterações de requisitos podem ser impactantes por prever tecnologias emergentes e para auxiliar pessoas com pouca experiência, tanto no time de desenvolvimento quanto no time de operações.

Portanto, os gestores da CaféSaúde estão solicitando um planejamento de atividades para gerenciar as alterações de requisitos.

Ao perceber a necessidade de alteração em requisitos, utiliza-se de um processo formal:

- a. Registrar a alteração para levar ao time de desenvolvimento para ser analisada.
  - b. Avaliar a alteração com a presença de todos os envolvidos, avaliando também possíveis impactos no projeto (prazo, escopo, orçamento).
  - c. Em alterações de integração com equipamento IoT, deve-se revisar o detalhamento das interfaces entre o equipamento e os componentes do sistema.
  - d. Via ferramenta automatizada, identificar os requisitos relacionados e, se necessário, efetuar as devidas adaptações.
  - e. Revisar a prioridade do novo requisito e dos mais requisitos.
  - f. Replanejar o projeto com o novo requisito, reunindo todos os envolvidos.
  - g. Comunicar o novo plano com as prioridades e previsão de entregas das iterações atual e futuras.
  - h. Implementar as mudanças.
- **Automatizar rastreamento:** escolher uma ferramenta (software) específica para fornecer a rastreabilidade em projetos ágeis; conectando os artefatos de requisitos de software, código-fonte e casos de testes.
  - **Comunicação eficiente:** mesmo que os envolvidos com o requisito aprovem as alterações, é fundamental comunicar regularmente aos stakeholders sobre mudanças, seu status e impactos analisados e esperados.

Bons estudos!

## Saiba mais

- I. **Saiba mais sobre: projeto de arquitetura.** Leia na página 75.

[Guide to the Software Engineering Body of Knowledge v4.0](#)



II. Saiba mais sobre: controle de mudanças. Leia na página 59.

[Guide to the Software Engineering Body of Knowledge v4.0](#)



III. Saiba mais sobre: **rastreabilidade**. Leia o item 7.2.6, na página 109.

[Engenharia de software](#)

## Referências

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 04 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos: software orientado ao negócio**. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

WASHIZAKI, H. **Guide to the Software Engineering Body of Knowledge v4.0**. Waseda University: 2024. Disponível em: [https://bit.ly/engreq\\_swebokv4](https://bit.ly/engreq_swebokv4) Acesso em: 15 nov. 2024.

## Aula 3

Monitoramento de Requisitos

### Monitoramento de requisitos

#### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante!

Nesta videoaula, você irá conhecer o monitoramento de requisitos.

Esse conteúdo é importante para a sua prática profissional, pois comprehende os processos de monitoramento de requisitos, influência dos stakeholders, interesses e requisitos e o alinhamento com os objetivos de negócios.

Prepare-se para esta jornada de conhecimento! Vamos lá!

### Ponto de Partida

Os gestores da CaféSaúde estão empenhados no desenvolvimento do sistema de gerenciamento da cadeia produtiva de café Standard, Premium e High Special. Para tanto, você ficará responsável por explicar a eles como serão os processos de monitoramento de requisitos, principalmente pela influência dos stakeholders, interesses e requisitos.

Continue pesquisando e atualizando seu conhecimento sobre os processos de monitoramento de requisitos para aplicar no seu cotidiano profissional.

Bons estudos!

## Vamos Começar!

### Monitoramento de requisitos

Este é um processo que visa eficiência na manutenção da qualidade do produto final, acompanhar, controlar e verificar continuamente o status e a evolução dos requisitos ao longo do ciclo de vida do projeto.

O monitoramento deve garantir que os requisitos sejam implementados conforme as necessidades dos stakeholders, alinhados com os objetivos do negócio e que mantenham a integridade do sistema, mesmo com as mudanças ou ajustes nos requisitos. Deve, ainda, visar a redução de retrabalho, controlar os custos e gerenciar os prazos de entrega, mantendo o grau de satisfação dos stakeholders.

### Processos de monitoramento de requisitos

A motivação para um analista de sistemas (e seus pares do time de desenvolvimento) investir esforços em monitoramento de requisitos é baseada nas características de estabilidade e volatilidade dos requisitos. Segundo SWEBOK® GUIDE V4.0 (2024):

**Estáveis:** são requisitos que provavelmente nunca mudarão ao longo da vida útil do software, por exemplo, tratamento com datas (vencimento de produtos, nascimento de pessoas, referências geográficas latilong, entre outros).

**Menos estáveis:** quando podem mudar ao longo do ciclo de vida do sistema; por exemplo, num sistema de bancos, os requisitos para suportar diferentes contas isentas de impostos, requisito de software que podem se tornar obsoletos devido a uma alteração da legislação governamental.

**Volátil:** requisitos podem ser muito instáveis, como aqueles que mudam durante o projeto de desenvolvimento (possivelmente mais de uma vez); dinâmicos e suscetíveis a mudanças frequentes, uma consequência de fatores internos ou externos; correntemente dependem do ambiente, dos stakeholders ou do contexto tecnológico.

Um dos processos no monitoramento de requisitos é avaliar a probabilidade de mudar requisitos, identificando os potencialmente voláteis. Ou seja, a correta classificação de um requisito pelo aspecto da volatilidade proporcionará maior eficiência no gerenciamento de requisitos.

Vejamos alguns desafios para o monitoramento de requisitos: (1) grande quantidade de requisitos para monitorar, em projetos extensos, requer ferramentas automatizadas para aumentar a eficiência; (2) alta frequência em alterações de requisitos dificulta em análise das dependências entre requisitos, requer ferramenta e gestão de configuração para rastreabilidade e eficiência na manutenção dos requisitos e demais artefatos relacionados.

Para SWEBOK® GUIDE V4.0 (2024), o processo de solicitação de alteração de software (Software Change Request - SCR) eficaz deve estar baseado em ferramentas de software e procedimentos orientados para aplicar o processo de mudança, incluindo o registro de justificativa do Comitê de Controle de Configuração (Configuration Control Board - CCB) quanto ao processo de mudança.

As alterações de requisitos de software são facilitadas quando existe o controle de versão de artefatos (de requisitos de software e demais itens relacionados entre eles). Vejamos um exemplo de uma lista de itens de configuração (Configuration Item - CI) que mostra o status de uma alteração realizada, em uma determinada data, ilustrada no Quadro 1.

Lista de itens de configuração (IC)			
Item de Configuração	Versão anterior	Versão atual	Data
IC#002	-	1	22/10/2023
IC#007	-	1	26/02/2024
IC#011	-	1	17/04/2024
IC#002	1	2	27/04/2024

Quadro 1 | Lista da versão de requisitos de software. Fonte: adaptado de SWEBOK® GUIDE V4.0 (2024, p. 194).

Assim, os recursos do sistema gerando o relatório do histórico de alterações pode facilitar o rastreamento da resolução de problemas e propiciar maior rapidez para o desenvolvimento das soluções, demonstrando um fluxo adequado para monitoramento de requisitos.

## Stakeholders, interesses e requisitos (influências e metas)

Entendamos as diretrizes para obter um plano essencial da participação das pessoas no processo de análise de requisitos de uma proposta de mudanças, tanto para tratamento de problemas quanto para construção de soluções.

O planejamento de engajamento do stakeholder conterá os stakeholders relevantes para a mudança pretendida e a maneira como ocorrerá a colaboração entre analista de sistemas e stakeholders, conforme ilustrado na Figura 1.



Figura 1 | Planejando o engajamento de stakeholders.

Como entrada do planejamento, entendem-se necessidades organizacionais, compreendendo: (1) problema a resolver ou oportunidade de melhoria, que a organização necessita de soluções; e (2) a abordagem, como as atividades/tarefas serão administradas pelo analista de sistemas.

Como saída, obteremos o plano de engajamento de stakeholder, composto pela lista de pessoas, características individuais, os respectivos papéis e responsabilidades nas mudanças pretendidas, inclusive com as abordagens de comunicação e colaboração a serem utilizadas pelo analista de negócios.

Cada pessoa envolvida na fase de análise da mudança tem autoridade ou exerce influências, sendo que os resultados dessas forças são imprescindíveis para definir a solução do problema ou o aproveitamento da oportunidade de negócio. Embora seja complexo identificar o nível de influência de cada indivíduo, espera-se que a lista contenha: o nome, o papel na organização, dados de contato, autoridade e responsabilidade, complementada por:

- As tarefas, com as entregas previstas, atribuídas ao stakeholder ou papel organizacional.
- O nível de interesse de cada stakeholder.
- O poder de influência de cada stakeholder.
- Identificação dos stakeholders e usuário final (que interagirá diretamente com o produto/solução), que terá participação fundamental em experiência do usuário (UX).

## Siga em Frente...

## Alinhamento com os objetivos de negócios

Os requisitos de softwares são descritos para atender a um determinado problema reclamado por um perfil de usuário. Aqueles alinhados aos objetivos estratégicos da organização atendem diretamente às metas e objetivos organizacionais, podem atender a missão e visão ou agregam valores da empresa em funcionalidades e características do sistema.

Vejamos um contexto de visão de negócio: se o requisito proporcionar resultados que agregam benefícios relacionados à visão da empresa, é considerado requisito alinhado aos Objetivos de Negócios (ON). Supondo que foi definido que a visão é “se tornar referência no mercado de ração para PET, em dois anos, aumentando sua participação de trinta por cento para sessenta por cento, na região metropolitana de Curitiba-PR”. Os requisitos definidos que contribuírem diretamente para essa meta, serão classificados como requisitos de objetivos de negócios.

No processo de desenvolvimento de software, existem ações que podem ser executadas sob foco de otimização dos resultados alinhados com os objetivos organizacionais, segundo SWEBOK® GUIDE V4.0 (2024), e as prioridades do negócio, clientes e usuários. As organizações iniciam a implementação de aplicações, em alinhamento aos objetivos organizacionais, baseadas em projetos, em uma escala de tempo e orçamento definidos.

Por um lado, os projetos são de natureza temporária, entregando funcionalidades com produto de software que atendam às necessidades do usuário, em um prazo definitivo e dentro do orçamento. Enquanto a manutenção de software visa prolongar a vida útil do produto de software, mantendo-o operacional e útil pelo maior tempo possível, normalmente, esses esforços atendem à demanda de usuários e stakeholders por atualizações e aprimoramentos de software.

Percebemos as duas fases distintas em produto de software: (1) desenvolvimento de aplicações com funcionalidades que entregam resultados alinhados aos ON e (2) manutenção das funcionalidades que agregam as mudanças e evoluções, e continuam entregando resultados alinhados aos ON.

Em ambos os casos, a economia da manutenção de software não é tão visível quanto a do desenvolvimento de software.

Contudo, segundo SWEBOK® GUIDE V4.0 (2024, p. 175), no nível organizacional, as fases de desenvolvimento e de manutenção são percebidas de maneiras divergentes quanto ao

investimento e resultado. A organização considera a manutenção consumindo recursos significativos com pouco benefício claro e quantificável. Por isso, é comum o desenvolvimento e adição de novos recursos receberem prioridade mais alta do que outras atividades de manutenção.

No entanto, atividades da engenharia de software são primordiais para manutenibilidade do software e na eficiência no gerenciamento de mudanças de sistemas, tais como: refatoração, segurança ou melhoria de desempenho, pois, sem executar essas ações, dificilmente alcançariam as metas e objetivos dos clientes de software, bem como às restrições como tempo e orçamento.

Sendo assim, recomenda-se o equilíbrio entre os objetivos e restrições organizacionais e de negócios e a capacidade de manutenção do software e os padrões de engenharia de software para evitar a deterioração do código-fonte, o que causaria uma dívida técnica, consequentemente, a falta de manutenibilidade, encaminhando-o à desativação do software, conforme ilustra na Figura 2.

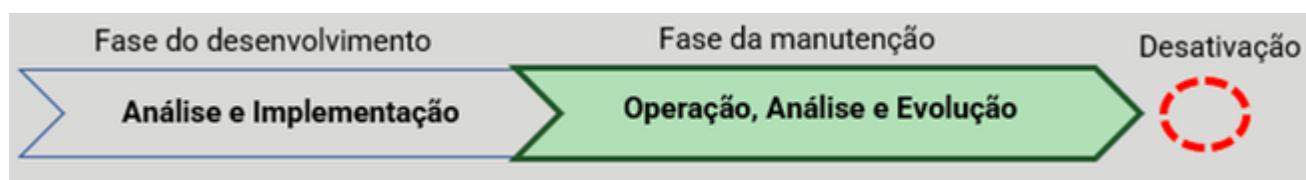


Figura 2 | Três fases de um sistema. Fonte: adaptada de Wiley (2015).

Ao investir continuamente em abordagens de gerenciamento de desenvolvimento e da gestão da manutenção de software, pode-se auxiliar as organizações a (SWEBOK® GUIDE V4.0, 2024, p. 175):

- Entender o custo total do software, em todo o seu ciclo de vida, do desenvolvimento à manutenção.
- Comparar os custos e os benefícios entre o desenvolvimento de novo software e o aprimoramento do software existente.
- Organizar os recursos de pessoas, papéis e habilidades, definindo a responsabilidade pela manutenção e desenvolvimento para a mesma equipe.
- Concentrar esforços em requisitos de manutenção desde o início, pois a mesma equipe é responsável pelo desenvolvimento e pela manutenção do software.

Para você compreender a caracterização desses requisitos, observemos exemplos alinhados aos objetivos estratégicos que atendem diretamente às metas corporativas, no Quadro 2, ilustrando com exemplos de requisitos de software derivados de um objetivo de negócio, demonstrando três níveis, objetivo de negócio (ON), projeto e requisito de software.

Objetivo de negócio	Projeto	Requisitos de software
<ul style="list-style-type: none"> <li>• Tornar-se referência no mercado de ração para</li> </ul>	<ul style="list-style-type: none"> <li>• Desenvolver sistema de planejamento e</li> </ul>	<ul style="list-style-type: none"> <li>• Manter recursos de materiais, tecnologia,</li> </ul>

<p>PET, em dois anos, aumentando sua participação de trinta por cento para sessenta por cento, na região metropolitana de Curitiba-PR.</p>	<p>produção.</p>	<p>equipamentos, métodos produtivos de ração.</p> <ul style="list-style-type: none"> <li>• Controlar parametrização de produção automatizada de ração.</li> <li>• Apresentar produtividade e qualidade da produção da indústria.</li> <li>• Comunicar situações de risco e oportunidades da produção na região metropolitana de Curitiba-PR.</li> </ul>
--	------------------	---

Quadro 2 | Exemplos de requisitos alinhados ao ON.

Os requisitos alinhados aos ON podem ser diferenciados conforme as necessidades de negócios ou estrategicamente ao processo transformador da organização. Lembremo-nos que todos os objetivos a serem alcançados são necessários, como planejamento, engajamento e recursos decompostos em projetos ou programas empresariais tecnológicos, que, por vezes, são decompostos em vários requisitos de software.

Podem existir inúmeras classificações, conforme o ramo de atividades do setor da economia ou da natureza dos negócios, com nomenclaturas específicas, como exemplificaremos a seguir, alguns dos requisitos alinhados aos ON, facilitando o entendimento e engajamento dos envolvidos:

**Metas corporativas:** relacionados à estratégia a médio e longo prazo, com visão clara do alvo ou condição pela qual a organização será referenciada ou reconhecida.

**Foco no ROI:** atendem ao plano de recebimento dos recursos investidos em um período especificado em planejamento financeiro, considerando todos os projetos e os custos da infraestrutura da organização.

**Conformidade e regulatório:** cumprir com as restrições impostas por organismos regulamentadores ou evidenciar os registros das operações que garantem a conformidade com as instituições normatizadoras.

**Mercado internacional:** em fase transitória, algumas metas grandiosas podem ser subdivididas para maior visibilidade e foco, por exemplo, expandir para o mercado europeu nos próximos cinco anos, com implementação de plataforma multilingue com suporte para diferentes moedas e regulamentações locais.

Estudante, os processos de gerenciamento das alterações de requisitos auxiliam na condução do projeto de desenvolvimento, os relacionamentos entre os requisitos, padronização na gestão de alteração requisitos e impactos no projeto. Continue pesquisando e estudando a importância do gerenciamento de alterações de requisitos.

Bons estudos!

## Vamos Exercitar?

Na CaféSaúde, os gestores passam a cuidar das plantações de café e das construções de secadores de café, em função da época das chuvas, adequada ao plantio, mas um problema sério em construções civis.

Eles estão ocupados nessas atividades dos projetos físicos primordiais, porém deixaram os projetos de desenvolvimento de software, ficando você responsável em alocar esforços também no projeto de desenvolvimento dos sistemas, alinhados à estratégia da CaféSaúde.

O primeiro passo para reverter essa situação de menor interesse pelos stakeholders em conceber e desenvolver sistemas do que providenciar a infraestrutura de atividades primárias, é mostrar que os resultados em sistemas são muito importantes também. Apesar de sabermos que a prioridade em atividades primárias são maiores que a das organizacionais, com pouco esforço dos stakeholders e a participação em momentos programados, as participações são imprescindíveis para os negócios de qualquer organização.

Podemos refletir sobre o nível de influência de cada pessoa nas definições ou em negociações de requisitos de software, mas será difícil encontrar uma resposta clara. Porém, é evidente que a participação de cada pessoa do time operacional pode ser decisiva para estabelecer alguns aspectos:

- Nas tarefas, com as entregas previstas, atribuídas ao stakeholder ou papel organizacional, quanto ao processo de secagem temos:  
O supervisor da secagem de café conhece como são as configurações de cada processo, pelo nível de umidade do grão, considerando todas as condições do ambiente e do potencial dos equipamentos; portanto, ele é a pessoa mais apropriada para contribuir com dados e fluxo do processo para essa etapa na produção do CaféSaúde, ou seja, quanto às definições de requisitos de negócio para registrar e monitorar a qualidade o café em produção.
- O nível de interesse de cada stakeholder, no processo de secagem de café:  
O operador do equipamento de secagem, Sr. Cho-Kim, está conhecendo o processo nesse ano, recebendo os primeiros treinamentos, enquanto o Sr. Okubo, experiente na região há mais de quinze anos, teria maior proficiência em configurar os equipamentos de secagem de café e também definir os requisitos do sistema. Porém, o iniciante colocou-se à disposição com maior facilidade que o sênior.

- O poder de influência de cada stakeholder, entre o iniciante e o sênior no processo de secagem:  
As duas experiências são importantes para definir o design do software, uma vez que o iniciante poderá contribuir pela falta de experiência viciosa no processo, enquanto o sênior contribuirá pelas suas inúmeras dificuldades enfrentadas com diversos tipos de equipamentos de secagem.
- Identificar qual dos stakeholders é usuário final (que interagirá diretamente com o produto/solução), este que terá participação fundamental em experiência do usuário (UX), no processo de secagem de café:  
A interface de configuração do equipamento de secagem e a integração com os demais processos da produção será de responsabilidade do supervisor do setor, enquanto a interface de registro de ocorrências e de pequenas decisões será do operador júnior.

Estudante, pelo exemplo vivenciado nesse caso da CaféSaúde, você percebeu que ambos os níveis de usuários podem influenciar na elicitação e especificação de requisitos de software. Fique preparado, observando pesquisas e experiências em alinhamento com as diretrizes organizacionais.

Bons estudos!

## Saiba mais

I. **Saiba mais sobre: alinhamento ao negócio**, lendo o item 2.2.1. *Alinhamento com os Objetivos Organizacionais*, página 175, onde percebemos que as prioridades devem ser analisadas de forma contínua.

[Guide to the Software Engineering Body of Knowledge v4.0.](#)



II. Saiba mais sobre: interesse, lendo a página 55.

### Gerenciamento de requisitos

III. Saiba mais sobre: requisitos de negócio. Leia as páginas 86 a 88.

### Engenharia de requisitos: software orientado ao negócio

## Referências

IIBA - International Institute of Business Analysis. **Padrão Essencial Global do IIBA para Análise de Negócios.** Toronto, Canada. Disponível em: [https://bit.ly/anasis\\_bacorestd](https://bit.ly/anasis_bacorestd) Acesso em: 28 set. 2024.

KERR, E. S. (org.). **Gerenciamento de requisitos.** São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software.** 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos.** Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos:** software orientado ao negócio. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

WASHIZAKI, H. **Guide to the Software Engineering Body of Knowledge v4.0.** Waseda University: 2024. Disponível em: ou [https://bit.ly/engreq\\_swebokv4](https://bit.ly/engreq_swebokv4) Acesso em: 15 nov. 2024.

## Aula 4

Rastreabilidade de Requisitos

### Rastreabilidade de requisitos



#### Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer o rastreamento de requisitos.

Este conteúdo é importante para a sua prática profissional, pois trata das finalidades do uso da matriz de rastreabilidade, os tipos de matriz de rastreabilidade e a elaboração da matriz de rastreabilidade.

Prepare-se para esta jornada de conhecimento! Vamos lá!!!

## Ponto de Partida

Os gestores da CaféSaúde estão empenhados no desenvolvimento do sistema de gerenciamento da cadeia produtiva de café Standard, Premium e High Special. Para tanto, você ficará responsável por explicar a eles como será a rastreabilidade de requisitos e suas respectivas vantagens para garantir maior segurança no desenvolvimento de software.

Com o aprendizado deste conteúdo, você estará mais preparado para atuar em engenharia de requisitos. Porém, continue pesquisando e atualizando seu conhecimento sobre os procedimentos de rastreabilidade de requisitos para aplicar no cotidiano profissional.

Bons estudos!

## Vamos Começar!

## Rastreabilidade de requisitos

Na engenharia de software, rastreabilidade e requisitos têm relação com mapeamentos documentados entre os artefatos (por exemplo, requisitos e casos de teste). Esse recurso permite ao engenheiro de requisitos representar a relação entre os requisitos e demais artefatos do projeto.

### Finalidades do uso da matriz de rastreabilidade

O uso da matriz de rastreabilidade está relacionado com a análise do impacto causado pelas mudanças nos requisitos de software. Usando a matriz de rastreabilidade, o analista de sistemas identificará quais requisitos estão relacionados, dependentes ou derivados, detalhará as alterações necessárias para cada um deles e quanto de trabalho (esforço e tempo) é necessário para realizá-las. Lembramos que esses itens a serem alterados podem ser design, scripts, diagramas, código-fonte, caso de testes, entre outros artefatos.

É comum em projetos em conformidades com padrões de qualidade de software o uso de matriz de rastreabilidade, juntamente com outras ferramentas simples, como os formulários e listas de verificação, até mesmo uma lista de verificação de revisão de código.

As ferramentas automatizadas que permitem controle de versão/ramificação de artefatos (por exemplo, Git) são fundamentadas em rastreamento de item de configuração (IC). Vejamos

alguns dos benefícios gerados pela rastreabilidade:

**Controle de mudanças:** auxilia diretamente na obtenção dos itens que sofrerão impacto em caso de alteração em um dos requisitos de software ou para rastrear quais especificações de software foram responsáveis pela implementação.

**Gestão de qualidade:** garantir que os requisitos sejam implementados e testados; incluindo a padronização de documentação em artefatos gerados ao longo da implementação.

**Redução de riscos:** uma das principais vantagens para evitar as omissões de requisitos, ou, ao contrário, a criação de requisitos de software desnecessários.

**Facilidade de auditoria:** favorece aos implementadores documentar conforme padrões definidos e oferece uma trilha clara para auditorias e conformidade regulatória.

**Melhora na comunicação:** promove alinhamento entre equipes técnicas e de negócios.

As ferramentas e técnicas da rastreabilidade, destinadas à qualidade de software são bem-vindas aos engenheiros de requisitos. Elas garantem que os artefatos sejam rapidamente localizados, exibam o histórico da sua criação até a última versão com as respectivas modificações.

Pensemos em ferramentas simples, como as listas de verificação (para revisão de código), muito empregadas para facilitar o controle de código-fonte revisado ou em revisão. Elas organizam as atividades dos desenvolvedores e melhoram a qualidade do software.

O desenvolvimento de software envolve normalmente o uso, criação e modificação de muitos produtos de trabalho, como documentos de planejamento, especificações de processo, requisitos de software, diagramas, projetos e pseudocódigo, código manuscrito e gerado por ferramentas, casos de teste e relatórios manuais e automatizados, arquivos e dados.

À medida que o software é desenvolvido, gerenciado, mantido ou estendido, essas relações de rastreabilidade devem ser mapeadas e controladas para demonstrar a consistência dos requisitos de software com o modelo de software.

O uso da rastreabilidade normalmente melhora o gerenciamento de produtos de trabalho de software e a qualidade do processo de software e garante às partes interessadas que todos os requisitos sejam atendidos.

A rastreabilidade permite a análise de mudanças na fase de produção e manutenção porque as relações com os produtos de trabalho do software podem ser facilmente percorridas para avaliar o impacto da mudança.

## Tipos de matriz de rastreabilidade

Uma vez que o requisito de software torna-se identificável (por um nome), torna-se também uma peça importante no desenvolvimento de software, pois pode sofrer modificações (alteração, derivação, sucessão ou suspensão). Ou seja, o requisito de software deve ser controlado em um repositório gerenciável pelos envolvidos no projeto.

Uma biblioteca de software é uma coleção controlada de artefatos, como código-fonte, scripts, código-objeto, documentação e demais artefatos relacionados. Cada item controlado deve ser vinculado entre eles. Os requisitos de software e os seus casos de teste são armazenados nesse repositório e devem ser vinculados às linhas de base de uma versão.

Lembremos que a linha base de uma versão numa biblioteca de software representa um conjunto de artefatos que configuram um estado definitivo, em qualquer instante do projeto. É importante por ser uma referência da qual pode ser recuperado o estado e conteúdo de qualquer item dessa versão, em qualquer momento do projeto.

Inclusive, a linha base de um repositório é um conceito fundamental que contribui de fato na rastreabilidade dos requisitos de software (e demais artefatos controlados), os relacionamentos entre eles, compondo as versões de cada um dos artefatos. Um código-fonte (ou qualquer outro artefato, inclusive os binários) construído é inserido no repositório, ficando à disposição para fazer parte de uma linha base.

Todo IC (item de configuração) que faz parte de uma linha base, tem o seu estado “congelado”, ou seja, não poderá sofrer alteração para essa linha base. Caso sofra alteração, uma nova versão é criada para esse IC, podendo fazer parte de uma nova linha base. Ilustremos essa situação na Figura 1.

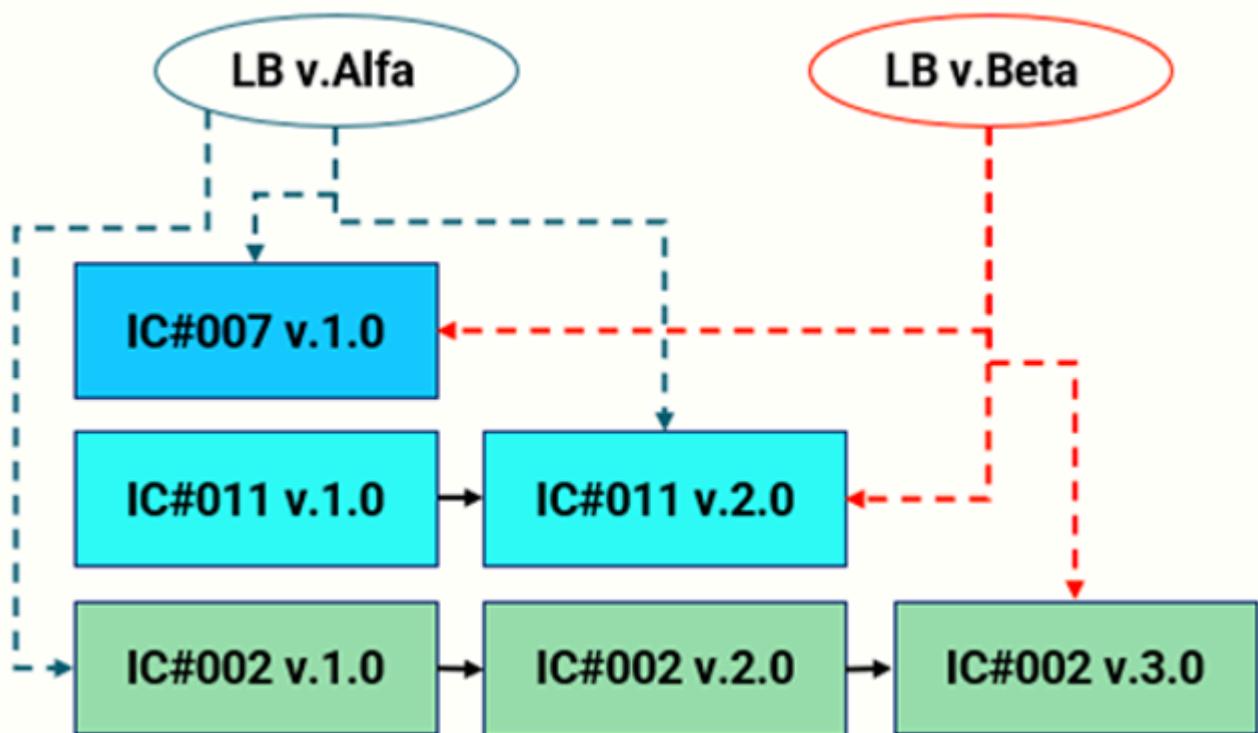


Figura 1 | Linha base e seus IC.

Compreendamos os fluxos de desenvolvimento, quanto aos itens de configuração. Quando o desenvolvimento de software acontece, alterando ou evoluindo, um IC passa de uma versão para outra a cada alteração recebida, como verificado na Figura 1, o IC#002 que sofreu duas alterações (da v.1.0 até v.3.0), no fluxo principal do desenvolvimento.

Os tipos de rastreabilidade são, segundo Kerr (2015):

- Bidirecional (Bidirectional Traceability): é possível rastrear um requisito de software a partir do artefato implementado até o seu requisito fonte, como também, ou na ordem inversa, passando por todos os níveis intermediários; usual nas análises de impactos de alterações por fornecer informações sobre as dependências com outros requisitos e causas de sua especificação.
- Na ordem direta (Forward Traceability): rastreamento de um requisito de software até determinados componentes de análise ou de implementação, casos de teste, ou seja, todos os artefatos gerados para atendê-lo.
- Na ordem inversa (Backward Traceability): rastreamento de um requisito de software (todos os artefatos gerados por causa dele) até sua fonte; podendo ser de um stakeholder, uma legislação, um padrão organizacional, ou a linguagem de programação de um sistema que deve operar em conjunto com o que está sendo desenvolvido.
- Interna aos requisitos: rastreamento das relações somente entre os requisitos de software, ou seja, sem a correlação com outros artefatos.
- Externa aos requisitos: rastreamento das relações entre requisitos e outros artefatos (produtos, fluxograma, diagramas, documentos de projeto etc.).

- Pré-requisitos: referência a características do requisito anteriores à sua especificação.
- Pós-requisitos: referência a características do requisito posteriores à sua especificação, utilizada frequentemente para descobrir os modelos de análise e procedimentos de teste criados para sua verificação.
- Funcional: rastreamento de requisitos relacionados com funcionalidades do sistema, normalmente estão diretamente relacionados aos requisitos funcionais.
- Não funcional: rastreamento de requisitos relacionados com características gerenciais como propósito, desempenho e responsabilidades, entre outros.

**Siga em Frente...**

## Elaboração da matriz de rastreabilidade

Uma matriz de rastreabilidade de requisitos ajuda a visualizar quantos e quais requisitos são influenciados pela mudança, facilitando a análise de impacto.

O tipo mais usado é a matriz requisito-requisito, indicado com os pontos de intersecção dos relacionamentos entre os requisitos, representados em uma tabela, conforme ilustra o quadro 1, mostrando o relacionamento de dependência entre os Requisitos. Por exemplo, o Req#001 tem relação com os Req#002 e Req#003.

### Bloco 1

Requisitos e Dependências	Req#001	Req#002	Req#003
Req#001		X	X
Req#002			X
Req#003	X		
Req#004		X	

### Bloco 2

Req#004
X
X

Quadro 1 | Matriz de rastreabilidade requisito-requisito. Fonte: adaptado de Kerr (2015, p. 143).

Verifica-se que o Req#001 tem relação com os Req#002 e Req#003, portanto, qualquer alteração no Req#001, o analista de sistemas deve analisar o impacto nos Req#002 e Req#003, assim garantido que todos os requisitos relacionados ou dependentes possam ser analisados e adequados.

Esse tipo de matriz é bastante simples, porém torna-se inviável caso seja praticado usando em documento ou planilha eletrônica para os projetos com grande volume de requisitos ou apresente alguma complexidade de relacionamento. Se tornará viável somente por meio de aplicativo de software específico, que pode montar e atualizar automaticamente.

Alguns problemas podem surgir por possuírem conteúdos diversos ou por falhas em apontamentos do relacionamento, prejudicando a rastreabilidade: (1) em casos de itens com informação não estruturada, por exemplo, gravações de entrevistas, vídeos, figuras, entre outros; (2) quando depende da documentação pelo desenvolvedor, causando falha na atualização das inter-relações entre os requisitos; (3) quando não tem critério apropriado para atualização de um requisito; deixando o requisito evoluído/modificado na mesma versão, pois poderá causar inconsistência com os demais requisitos.

Normalmente, o procedimento para elaborar uma matriz de rastreabilidade é simples, porém há critérios claros a todos que utilizarão a ferramenta:

**Estrutura:** identificar e definir que tipos de artefatos serão controlados e as relações a serem rastreadas (requisitos de software ☐ artefatos de design ☐ código-fonte ☐ arquivos de testes ☐ objetos de verificação).

**Ferramentas:** utilizar ferramentas de software para cadastrar os itens de controle de rastreabilidade, como Excel, Jira, IBM DOORS, Trello, entre outras.

**Pessoal:** capacitar todos os envolvidos no processo do desenvolvimento para gerar os artefatos com as documentações padrão.

**Manutenção Contínua:** atualizar a matriz regularmente para refletir mudanças nos requisitos e no projeto.

Segundo SWEBOK® GUIDE V4.0 (2024, p. 268), alguns modelos de ciclo de vida podem afetar a ordem das atividades de projeto de software. Uma revisão do produto de software implica em revisitar os requisitos de segurança, depende da rastreabilidade dos requisitos de segurança ao longo do processo de desenvolvimento. Assim, é importante que a equipe de segurança inclua pessoas especializadas para revisão dos requisitos de segurança, quando dos artefatos identificados na matriz de rastreabilidade com a classificação de segurança.

Estudante, os processos de rastreabilidade de requisitos auxiliam na condução do projeto de desenvolvimento. Continue pesquisando e estudando a importância do rastreamento de requisitos.

Bons estudos!

## Vamos Exercitar?

Avançando em design e implementação do software, os gestores solicitaram uma garantia de que todos os requisitos de softwares estão sendo devidamente implementados, no projeto da CaféSaúde. O analista de sistemas reuniu com o engenheiro de software para preparar uma demonstração de uma matriz de rastreabilidade, mostrando os artefatos gerados a partir de um requisito de software.

Você é o responsável por alocar esforços também no projeto de desenvolvimento dos sistemas, que estão alinhados à estratégia da CaféSaúde.

Portanto, foi apresentado pelo time de desenvolvimento uma matriz de rastreabilidade bidirecional simples, envolvendo elementos da análise e da implementação, ilustrada no Quadro 2:

### Bloco 1

Requisito	Design	Código-fonte	Caso Teste
RF#033 Login do usuário de campo	D#091 Tela de Login	CF Login.js	TM#021 TestLogin
RF#010 Movimentação da Colheita	D#022 Tela Colheita	CF Colheita.js	TM#220 TestColheita
RF#309 Integração IoT temperatura	D#702 Interface	CF CompoTemp	TA#713 TestTemper
RN#211 Segurança colheita	D#111 painel segurança	CF CompoSegur	TA#850 Invasor

### Bloco 2

Status
Aprov
Aprov
Dsv
NãoIn

Quadro 2 | Relatório da matriz de rastreabilidade

No caso da CaféSaúde, foram usadas siglas para caracterizar os tipos de requisitos e demais artefatos referências na matriz de rastreabilidade, extraída da ferramenta repositório dos artefatos, com suporte ao gerenciamento de desenvolvimento, que apresenta o status do requisito de software. O RF são requisitos funcionais, RN são requisitos não funcionais, D item do design, CF código-fonte de programa ou de componente, TM/TA identificação de testes manuais ou gerados automaticamente, Status Aprov (aprovado), Dsv (em desenvolvimento), Nãoin (não iniciado).

Estudante, pelo exemplo vivenciado nesse caso da CaféSaúde, você percebeu que a rastreabilidade pode ser composta de vários níveis, conforme a natureza do software. Fique preparado cada vez mais com pesquisas e experiências em rastreamento de requisitos de software.

Bons estudos!

## Saiba mais

I. **Saiba mais sobre: rastreabilidade.** Leia a página 210, para compreender a rastreabilidade.

[Guide to the Software Engineering Body of Knowledge v4.0.](#)



II. **Saiba mais sobre: rastreabilidade.** Leia as páginas 142 e 146, para compreender a rastreabilidade e os seus tipos.

[Gerenciamento de requisitos.](#)

III. **Saiba mais sobre: uso da rastreabilidade.** Leia as páginas 256 e 257, para compreender uma aplicabilidade.

[Engenharia de requisitos: software orientado ao negócio.](#)

## Referências

IIBA - International Institute of Business Analysis. **Padrão Essencial Global do IIBA para Análise de Negócios**. Toronto, Canada. Disponível em: [https://bit.ly/anasis\\_bacorestd](https://bit.ly/anasis_bacorestd) Acesso em: 28 set. 2024.

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos: software orientado ao negócio**. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

WASHIZAKI, H. **Guide to the Software Engineering Body of Knowledge v4.0**. Waseda University: 2024. Disponível em: ou [https://bit.ly/engreq\\_swebokv4](https://bit.ly/engreq_swebokv4) Acesso em: 15 nov. 2024.

## Aula 5

Encerramento da Unidade

### Videoaula de Encerramento

#### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante!

Nesta videoaula, você irá conhecer o processo de gerenciamento de requisitos.

Esse conteúdo é importante para a sua prática profissional, pois traz conhecimentos fundamentais para entregar sistemas com qualidade, baseados em requisitos de softwares gerenciados, rastreáveis e alinhados aos objetivos de negócio.

Prepare-se para essa jornada de conhecimento! Vamos lá!!!

## Ponto de Chegada

Olá, estudante! A competência desta unidade é:

- Compreender o gerenciamento de requisitos.
- Explicar a gestão de alteração de requisitos.
- Descrever o monitoramento de requisitos e alinhamento com os objetivos de negócio.
- Explicar a rastreabilidade de requisitos.

Você deverá compreender e reconhecer os mecanismos fundamentais que competem ao gestor de projetos de softwares, conhecer e aprimorar para entregar sistemas com qualidade baseados em requisitos de softwares gerenciados, rastreáveis e alinhados aos objetivos de negócio.

## É Hora de Praticar!



### Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

A implementação está plena, indicando que existem requisitos de software funcionais e não funcionais codificados, outros em planejamento e especificação; também existem aqueles que estão sofrendo mudanças.

Essa dinâmica, aparentemente, é crítica, porém devemos nos organizar e gerenciar adequadamente para a realização do desenvolvimento adequado, atendendo a todos os requisitos do sistema, conforme as expectativas do cliente contratante.

Vamos explanar a dinâmica na CaféSaúde, conforme os requisitos estão sendo alterados e implementados, ao longo do ciclo de vida do desenvolvimento de sistemas.

Lembremo-nos que os envolvidos da CaféSaúde e os líderes do time de desenvolvimento devem priorizar os requisitos em ordem lógica e aceita para atingir os objetivos de negócios da empresa, gerenciando e monitorando os requisitos de software, garantido que se mantenham rastreáveis para possíveis auditorias.

Está apto para explicar a importância do gerenciamento das mudanças e os riscos que ela provoca em requisitos de software?

Qual é a vantagem do alinhamento dos requisitos de software com os objetivos de negócios no monitoramento de requisitos?

Você consegue explicar os benefícios da rastreabilidade de requisitos de software?

Para responder à reflexão da importância do gerenciamento das mudanças e os riscos que ela provoca em requisitos de software, percebemos a situação na CaféSaúde.

Na fase de design do sistema de gerenciamento de colheitas do café, ao classificar os requisitos de software, foram identificadas características de dependência, derivações e sucessão de requisitos, o que facilitará a implementação, gestão e validação das funcionalidades que compõem esse módulo. Vejamos as situações:

**Derivação:** o requisito de software deriva de outro, normalmente em um relacionamento sequencial no desenvolvimento, porém uma restrição exige que seja desenvolvido em sequência, pela arquitetura, mesmo que existam recursos para desenvolver juntos. Compreendamos:

- Considerando os dois requisitos: RF#028 Manter o cadastro de talhões de café e RF#721.3 Integrar os dados de localização do talhão de café.
- Considerando que existe uma restrição na fase de implementação de componentes: aproveitamento de recursos na implementação dos requisitos de integração, pelo desenvolvimento de componentes especializados em comunicação com o IoT (Internet of Things).
- Existirão duas tarefas na implementação: (1) codificação para o requisito RF#028 e (2) adequação do código-fonte para atender ao requisito RF#721.3

RF#028	←	RF#721.3
--------	---	----------

- As implementações serão executadas por dois desenvolvedores distintos, sendo que aquele que assumir a implementação de um dos requisitos, será o revisor da implementação do outro requisito.
- Cada requisito poderá sofrer alterações, futuramente, independentemente, porém sempre que ocorrer mudança em um deles, o outro deverá ser revisado, para manter o funcionamento correto.

**Dependência:** um requisito de software depende mutuamente de outro. Sempre que ocorrer a alteração em um deles, o outro também poderá ser alterado, portanto devem ser revisados. Compreendamos:

- Considerando os dois requisitos: RF#028 Manter o cadastro de talhões de café e RF#237 Manter dados históricos de alteração de talhões de café.
- Existirão duas tarefas na implementação: (1) codificação para o requisito RF#028 e (2) codificação para o requisito RF#237

RF#028    ↔    RF#237

- As implementações serão executadas por dois desenvolvedores distintos, sendo que aquele que assumir a implementação de um dos requisitos, será o revisor da implementação do outro requisito.
- Cada requisito poderá sofrer alterações, futuramente, independentemente, porém sempre que ocorrer mudança em um deles, o outro deverá ser revisado, para manter o funcionamento correto.

**Sucessão:** um requisito de software evolui à medida que sofre modificação, ou seja, a cada nova versão dele, é uma instância identificada de sucessão. Compreendemos:

Considerando os dois requisitos: RF#028 Manter o cadastro de talhões de café, sempre ocorrer alteração o requisito receberá novo rótulo, passando de RF#028 v.1 para RF#028 v.2

RF#028 v.1    →    RF#028 v.2

- As implementações serão executadas em momentos distintos, sendo que a versão N+1 ocorrerá sempre após a versão N.

Vejamos as características do requisito **alinhados aos objetivos estratégicos** na CaféSaúde, sendo um diferencial da organização:

Os requisitos de software traduzem objetivos estratégicos em soluções tecnológicas porque especificam o uso de equipamento de tecnologia conectados entre si para diminuir ou evitar a transcrição de dados, efetuando as integrações em tempo real.

RF#018 Os dados da colheita devem ser integrados por transmissão, em tempo real, ao servidor principal, exceto em situações de queda do link, quando são acumulados em armazenamento local, em duas unidades de armazenamento simultaneamente e distintas, para minimizar os riscos de perda de informações.

Esse requisito (RF#018) também atende ao objetivo da CaféSaúde, alinhando a:

- Automação dos processos para aumentar eficiência.
- Priorização de funcionalidades que agreguem valor ao negócio.
- Suporte ao crescimento da organização, com funcionalidades preparadas para escala.
- Garantia de flexibilidade por conceber alternativas em caso de falta de conexão com o servidor.

Alinhar os requisitos de software aos objetivos estratégicos da organização garante a relevância do sistema para o negócio e maximiza o impacto do investimento em tecnologia. Lembrando que esse alinhamento deve ser contínuo, envolvendo todos os stakeholders e adaptando-se às mudanças nas metas e no mercado.

Vamos compreender os benefícios da rastreabilidade de requisitos de software, conforme ilustrado na Figura 1, exibindo uma situação em que a linha base (LB) é formada pelos itens de configuração (IC) da versão Beta, conforme os IC: IC#007 v.1.0, IC#011 v.2.0 e IC#002 v.3.0.

Verifica-se nessa figura que existe uma indicação de mudança no requisito IC#011, na sua última versão v.2.0.

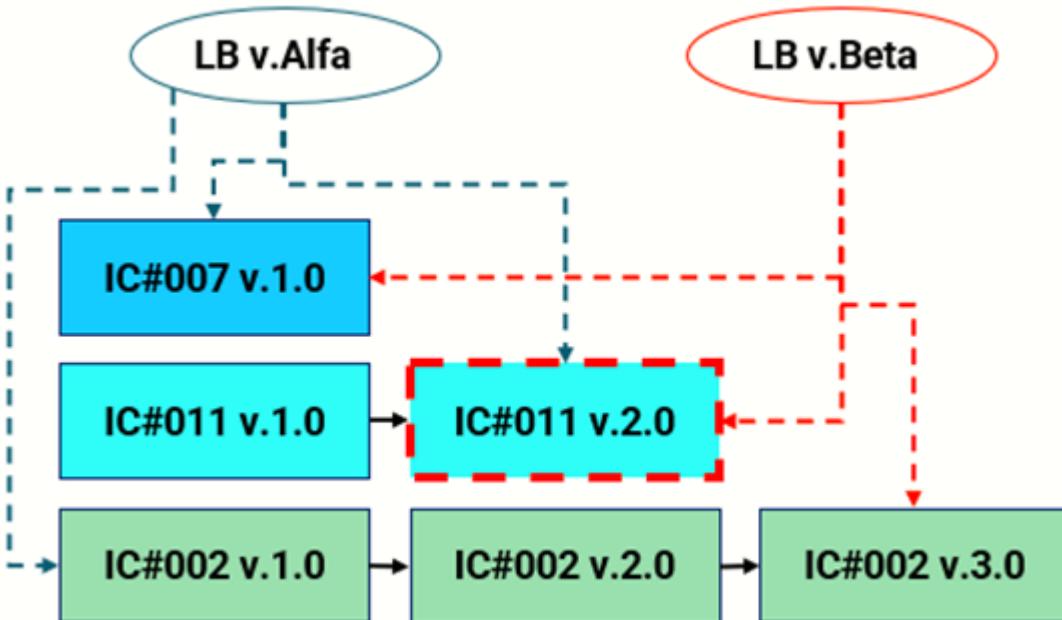


Figura 1 | Requisitos em alteração.

Ao efetuar alterações no requisito do artefato IC#011, ele recebe uma versão de sucessão v.3.0, conforme ilustra a Figura 2; porém esse IC ainda não está presente em nenhuma linha base, possivelmente por estar em testes finais. Caso essa versão v.3.0 do IC#011 seja aprovada para incorporar uma nova release do sistema, na próxima geração de executável ou build, ou entrega, o IC será indicado.

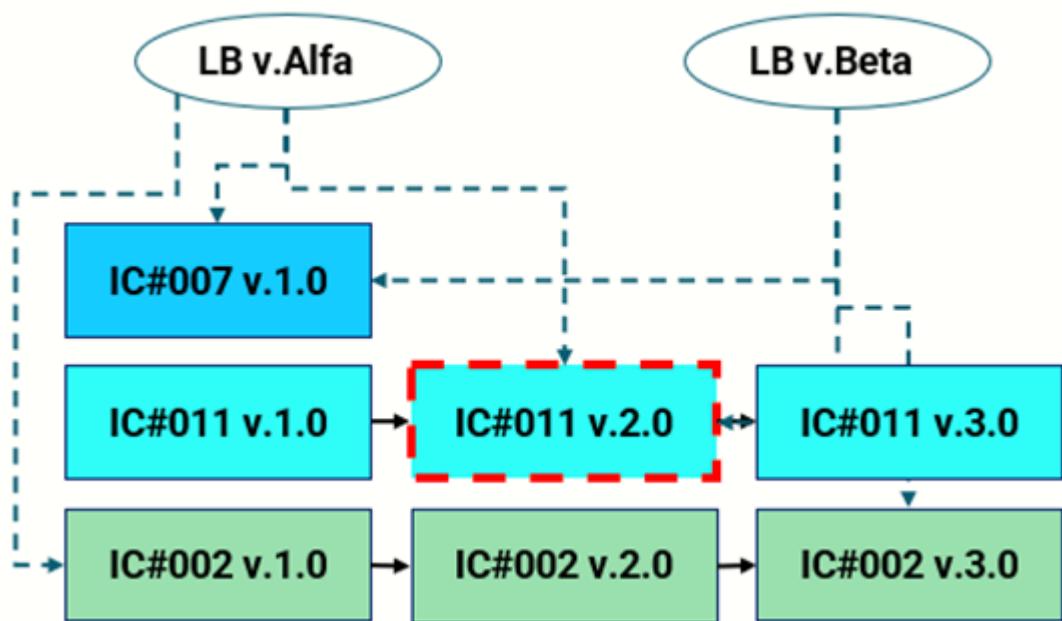


Figura 2 | IC com alterações realizadas.

Ao receber aprovação para disponibilizar a alteração realizada no IC#011, com a versão v.3.0, ele fará parte da versão Gama, ou seja, uma nova linha base será gerada LV v.Gama, conforme ilustra a Figura 3, compondo os itens: IC: IC#007 v.1.0, IC#011 v.3.0 e IC#002 v.3.0.

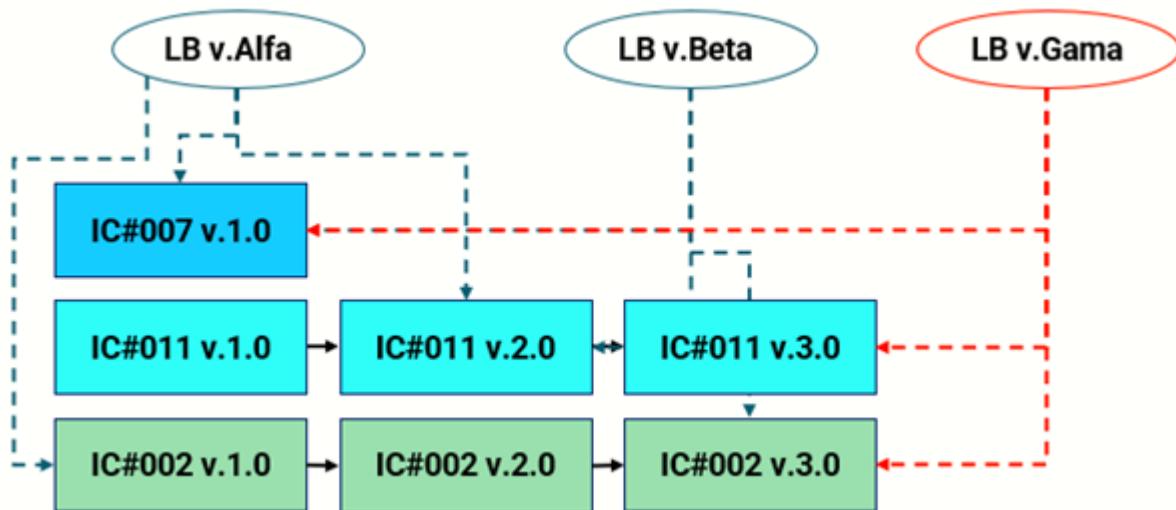


Figura 3 – IC na versão LB v.Gama.

O gerenciamento de itens de configuração, utilizando o versionamento por meio de linha base (LB), traz uma garantia de que todos os artefatos relacionados estão na sua melhor versão. Nesse caso, percebemos que é possível identificar os itens anteriores à modificação, o item IC#011 v.2.0, que continua intacto para consultas ou para testes de regressão, quando necessário.

A caracterização desse cenário controlado dará suporte suficiente aos auditores, quando investigarem o histórico de evolução de cada item. As três versões do IC#011, até mesmo a versão v.1.0 do item IC#011 que não fez parte de uma LB, não foi colocada em ambiente de produção em nenhum momento.

Estudante, o objetivo desse estudo de caso foi facilitar o entendimento do rastreamento de artefatos, inclusive dos requisitos de software, em uma abordagem do controle de versão por meio da matriz de rastreabilidade da gestão de configuração no ambiente da CaféSaúde. Continue aprimorando a sua habilidade e compreensão do gerenciamento de requisitos de software se atualizando com novas técnicas e abordagens.

Bons estudos!

Acesse o infográfico utilizando o QRCode por qualquer dispositivo para compreender a extensão do conteúdo desta unidade da disciplina de Engenharia de Requisitos.

## ENGENHARIA DE REQUISITOS

prof. Marco Hisatomi

[https://bit.ly/engreq\\_info3\\_gerencrequisitos](https://bit.ly/engreq_info3_gerencrequisitos)

### GERENCIAMENTO DE REQUISITOS – UN3

Compreender o Gerenciamento de Requisitos;  
Explicar a Gestão de Alteração de Requisitos;  
Descrever o Monitoramento de Requisitos e  
alinhamento com o Objetivos de Negócio;  
Explicar a Rastreabilidade de Requisitos.

#### Processos de Gerenciamento de Requisitos



Compreenderá os Riscos em Mudanças de Requisitos, Processos de Qualidade na Engenharia de Requisitos e a Revisão de Requisitos.

#### Gestão de Alteração de Requisitos

Compreenderá como Gerenciar os Relacionamentos entre os Requisitos, a Padronização na Gestão de Alteração Requisitos e os Impactos no Projeto.



#### Monitoramento de Requisitos



Compreenderá sobre os Processos de Monitoramento de Requisitos, influência dos Stakeholders, Interesses e Requisitos, Alinhamento com os Objetivos de Negócios.



#### Rastreabilidade de Requisitos

Compreenderá sobre a Finalidades do Uso da Matriz de Rastreabilidade, os Tipos de Matriz de Rastreabilidade e a Elaboração da Matriz de Rastreabilidade.



Avalie!

[https://bit.ly/engreq\\_avali\\_un3](https://bit.ly/engreq_avali_un3)

VOCÊ APRENDE MAIS QUANDO ESTUDA DENTRO DO SEU ESTILO

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos**: software orientado ao negócio. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

WASHIZAKI, H. **Guide to the Software Engineering Body of Knowledge v4.0**. Waseda University: 2024. Disponível em: ou [https://bit.ly/engreq\\_swebokv4](https://bit.ly/engreq_swebokv4) Acesso em: 15 nov. 2024.

## Unidade 4

### Verificação, Validação e Documentação de Requisitos

#### Aula 1

##### Verificação de Requisitos

#### Verificação de requisitos

##### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

##### Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer a verificação de requisitos. Este conteúdo é importante para a sua prática profissional, pois ensina a você a averiguação de requisitos ambíguos, os conflitos e consistência de requisitos e a revisão da especificação de requisitos.

Prepare-se para essa jornada de conhecimento! Vamos lá!!!

## Ponto de Partida

Os gestores da CaféSaúde estão empenhados no desenvolvimento do sistema de gerenciamento da cadeia produtiva de café Standard, Premium e High Special. Para tanto, você ficará responsável por explicar a eles como será a verificação, validação e documentação de requisitos de software.

Para ampliar a qualidade dos requisitos, eles podem passar por um processo de revisão de especificação de requisitos, pelos quais serão verificados todos os detalhes.

Continue pesquisando e atualizando seu conhecimento sobre os processos de verificação de requisitos para aplicar no seu cotidiano profissional.

Bons estudos!

## Vamos Começar!

Estudante, seja bem-vindo ao conteúdo sobre a verificação de requisitos!

A verificação de requisitos é uma tarefa em que os envolvidos (analistas de sistemas, stakeholders, engenheiros de software e demais pessoas) concluem se o requisito de software está definido corretamente, conforme IIBA BABOK v.3 (2015). Lembremo-nos que existe diferença entre requisito “definido” e “definido corretamente”, portanto vamos compreender esse último estado para obter o requisito de software verificado.

Para garantir que o requisito tenha sido definido corretamente, a participação do analista de negócio é fundamental, este que identificou a necessidade real da organização ou o problema vivenciado pelos usuários finais. Os artefatos que compõem os requisitos de software devem ser facilmente compreendidos pelo time de desenvolvimento (arquiteto, engenheiro de software e demais membros), que executará as atividades de desenvolvimento do software.

Normalmente, obtém-se boa qualidade ao seguir um padrão de notação representando a realidade da organização, em texto, descrito com linguagem formal ou informal ao nível de compreensão de todos os envolvidos. Se adequado a uma finalidade específica, com a história do usuário ou documentos, diagramas, protótipos, anexos, entre outros, conforme a metodologia e cultura organizacional, permitirá o desenvolvimento do software com qualidade satisfatória.

## Averiguação de requisitos ambíguos

Devemos ter ciência de que requisitos ambíguos são um risco total ou parcial do projeto de desenvolvimento de software, segundo SWEBOK® GUIDE V4.0 (2024), caso não seja superada a incerteza ou a ambiguidade em requisitos de software. Muitas vezes, a incerteza reflete a falta de conhecimento, portanto, quando verificar esse fato, a investigação é o caminho, estudando fontes formais, livros didáticos e periódicos profissionais. Interagir com os envolvidos é fundamental, entrevistando as partes interessadas ou consultando outros membros do time.

Percebemos, a dinâmica da interação e o crescimento da certeza de requisitos definidos corretamente, na Figura 1.

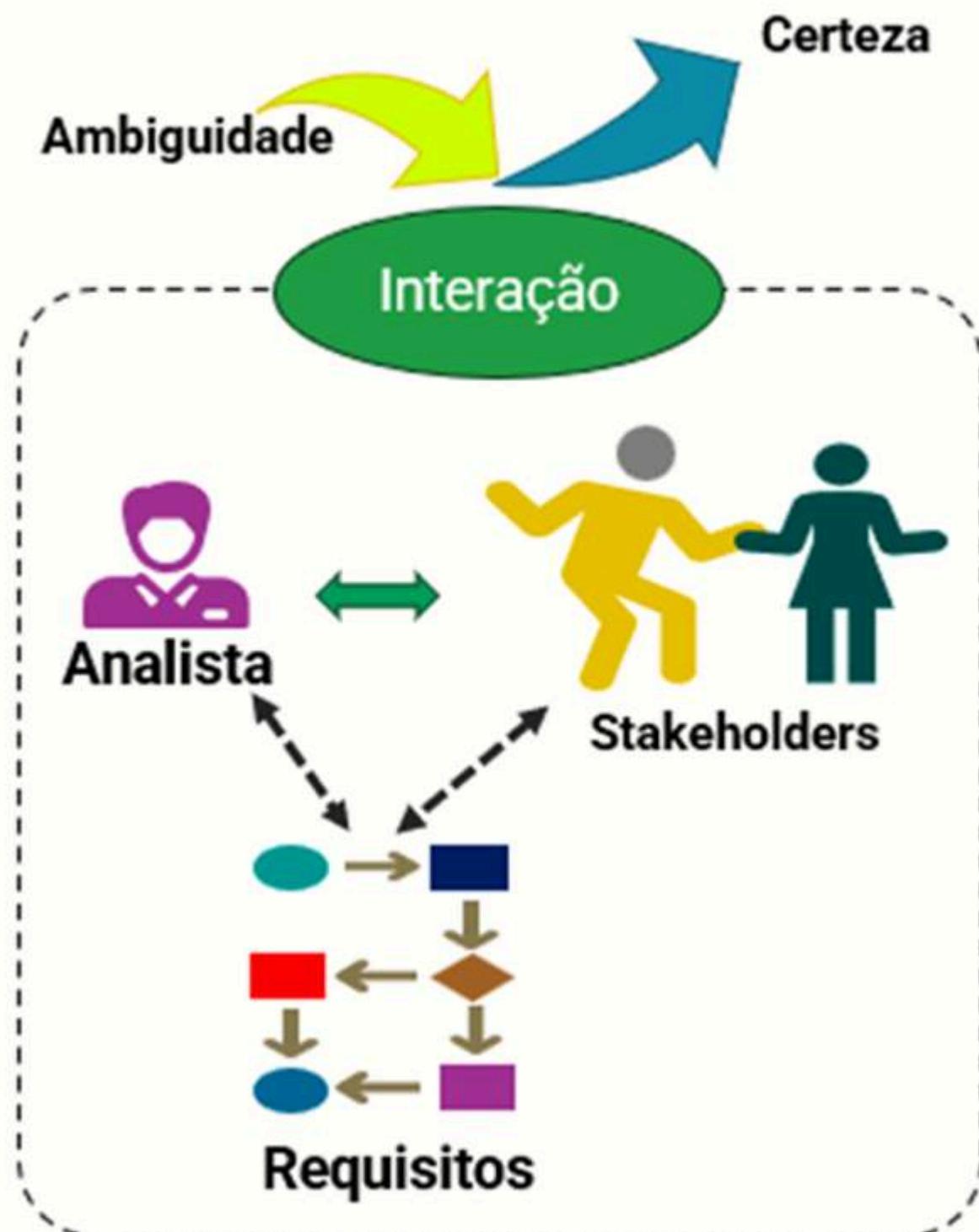


Figura 1 | Aumentando o nível de certeza.

Estudante, saibamos como investigar os requisitos de software para que possamos classificá-los como “definido corretamente”, verificando características de requisitos e aspectos de qualidade, baseado em IIBA BABOK v.3 (2015), embora a qualidade seja, em última análise, determinada pelas necessidades das partes interessadas que usarão os requisitos:

- Atômico: quando a escrita, por si, do requisito é capaz de ser compreendida pelos envolvidos, independentemente de outros requisitos.
- Completo: contém o detalhe adequado para que o trabalho continue, com as orientações para o trabalho futuro, porém o nível de completude é relativo à perspectiva ou metodologia, bem como a fase do desenvolvimento em que o requisito está sendo examinado ou representado.
- Consistente: o resultado deve estar alinhado às necessidades dos stakeholders e não conflitante com outros requisitos.
- Conciso: a descrição deve ter conteúdo sucinto, apenas o necessário.
- Viável: tecnicamente possível ser desenvolvido, economicamente dentro do cronograma e do orçamento, sob análise de riscos.
- Testável: possível de ser verificado, os seus resultados são mensuráveis ou de valor percebido.
- Priorizado: classificado, agrupado ou negociado em termos de importância e valor em relação a todos os outros requisitos.

Percebemos como os envolvidos podem averiguar ambiguidade na especificação de requisitos de software. Ao revisar anotações, podem procurar por termos que nem sempre são específicos tecnicamente, por exemplo: “rápido”, “eficiente” ou “adequado”, que podem ter significados totalmente distintos, dependendo da experiência ou conhecimento da pessoa.

Nesses termos, a verificação de requisitos poderá apontar a necessidade de discernimento para expressões com critérios que definam o que é “rápido”, trocando por “se comparado com o processo atual, deve ser cinco vezes mais rápido para processar o lote de entregas”. Fica, assim, claro e objetivo, inclusive mensurável, para análise e testes futuros.

Portanto, estudante, ao ser responsável pela tarefa de verificação deve-se ter em mente encontrar exemplos concretos, quantificáveis, diretos e objetivos para substituir as descrições vagas e evasivas, ou inquantificáveis.

Por exemplo: “quais critérios definem que a janela de movimentação é adequada para preencher?” Poderia estar especificando o tamanho, a fonte e cores das letras na janela de movimentação, também explicitando qual é a ordem dos dados de preenchimento e uso de mouse, acionamento por teclas ou touch screen (ecrã sensível ao toque ou pressão dos dedos), entre outros aspectos.

Após minuciosa verificação, os requisitos podem seguir o fluxo do desenvolvimento de software. Os engenheiros de software saberão que, após a verificação, os requisitos são viáveis economicamente e tecnicamente para serem implementados, portanto estarão seguros para arquitetar os componentes e módulos do sistema de maneira mais eficiente em sua estrutura.

## Conflitos e consistência de requisitos

Podemos refletir continuamente sobre os adjetivos “conflitantes” ou “consistentes” para obter maior qualidade na especificação de requisitos de software. No papel de analista de negócios ou de sistemas, devemos entregar requisitos ausentes de conflitos (diferenciar e explanar tais diferenças com detalhes criteriosos e perceptíveis). Concomitantemente, esse requisito apresente consistência (compreende-se por ser estável). Devemos especificar os requisitos funcionais de um software compreensível aos envolvidos:

**Sem conflitos:** as entradas e saídas de um procedimento devem ser completas e esclarecidas, não deixando dúvidas aos envolvidos e ao time de desenvolvimento. Por exemplo, se iniciar o uso de uma unidade de medidas em quilograma, deverá manter essa unidade em todo o processo, exceto quanto queira especificar a transformação de quilograma para onça, conforme esperado pelo usuário final.

**Consistente:** não deve haver definições contraditórias entre si; embora saibamos que é praticamente impossível especificar todos os requisitos com esse critério em sistemas complexos ou de grande porte.

Os motivos que levam a cometer tais erros na especificação de requisitos são inúmeros, podemos citar alguns:

- Erros na compreensão em reuniões e entrevistas, por utilizar termos técnicos ou de origem desconhecida (fabricante de equipamentos, regionalismo, entre outros); até mesmo na escrita de documentos ou na diagramação dos fluxos de processos.
- Em questionários respondidos por várias pessoas, em perguntas abertas (descritivas ou dissertativas) em que prevalece o vocabulário regional ou cultural; por exemplo, “como você classifica, em sua avaliação, a mercadoria que está sendo industrializada?”
- Lacunas em projetos, nos quais se determinam limites entre um processo e outro, tecnicamente recomendado restringir o escopo dos processos para viabilizar o cronograma, recursos e orçamentos. Por exemplo, em dois projetos em desenvolvimento, em um deles o termo é “fabricante” e, no outro, “fornecedor”, sendo indiferente para essa organização a utilização desses dois vocábulos. Em outra organização é imprescindível a distinção entre uma entidade que comercializa mercadorias prontas (fornecedor) e outra que transforma matéria prima em mercadoria para uso final (fabricante).
- Grande número de stakeholders envolvidos, com necessidades diversas, eventualmente inconsistentes ou conflitantes. Por exemplo, quando há um projeto extenso envolvendo fornecedores e clientes, ao mesmo tempo, no desenvolvimento de sistema de [SCM](#) (Supply Chain Management - SCM é um termo padrão em Sistemas de Informações, conheça mais sobre o assunto), podemos nos deparar com diferentes nomes de produtos ou de unidade de medidas; o fornecedor trata como macaxeira, a indústria local conhece por mandioca e o cliente procura pelo aipim.

Compreendemos que o trabalho do analista de sistemas é aprofundado, minucioso, com critérios definitivos para entregar requisitos de software apropriados para dar sequência à

implementação. Ou seja, todos os envolvidos no projeto vão interpretar o requisito funcional ou não funcional e desenvolverão suas tarefas produzindo resultados exatamente como o cliente e usuário final necessitam.

## Siga em Frente...

### Revisão da especificação de requisitos

A revisão de requisitos de software é uma atividade continua, em virtude das mudanças que um requisito pode sofrer, pois a dinâmica dos processos de negócios interrompe a estabilidade dos requisitos funcionais, sem cadência ou intensidade definida.

Outro fator relevante no mundo real é a complexidade que aumenta por causa das correlações de dependência ou de derivação entre os requisitos de software. Sempre que um requisito é modificado outros requerem revisão em suas especificações.

Para Sommerville (2011), as inconsistências e conflitos nas especificações de requisitos de software podem aparecer em qualquer momento do projeto. Desde a fase inicial ou durante a implementação do software, até na fase de operações, exatamente por causa das mudanças em requisitos de negócios que refletem diretamente nos requisitos funcionais. Assim, o time de desenvolvimento deve ser capaz de gerenciar as mudanças nas especificações e, consequentemente, revisá-las.

Ao revisar um requisito de software, espera-se que estejamos averiguando possíveis ambiguidades e inconsistências, ou até mesmo conflitos entre si. Trata-se de exercer um olhar técnico e crítico para manter a especificação revisada em um nível que seja compreendido pelos demais envolvidos que darão continuidade às tarefas do desenvolvimento.

Vejamos um exemplo de especificação de requisito de software que apresenta problemas e como a revisão poderá ter objetivo claro:

Em uma metrópole, as escolas enfrentam dificuldade em definir quem serão seus alunos no próximo ano; por outro lado, os alunos não sabem em qual escola haverá vagas para se matricularem. Vamos escrever alguns requisitos do sistema de vagas abertas para compreender o processo de revisão da especificação de requisitos:

Requisito inicial: “o estudante do ensino fundamental deve ser capaz de pesquisar se existe uma vaga para o próximo ano”.

Em primeira análise, percebemos que: o requisito está conciso, porém faltando detalhes para se tornar comprehensível aos desenvolvedores.

- Seria uma consulta à base de todas as escolas, independente da região onde o estudante reside?

- Qual seria a faixa etária do estudante que procura uma vaga?
- Qual ano ou nível do ensino fundamental o estudante frequentará?
- Teria escolha ou preferência em período do dia para frequentar as aulas?
- Teria opções para escola municipal, estadual ou federal?

O requisito necessita de melhorias para contemplar os esclarecimentos em parte das dúvidas apresentadas. Elementos que não foram especificados podem gerar conflito ou ambiguidade em tarefas no desenvolvimento, tais como:

- a. Quais parâmetros o estudante poderá ou deverá fornecer no momento da consulta às vagas?
- b. Como será apresentada a lista de informações consultada pelo estudante?
- c. Para a análise dos dados, faltam as especificações de cada dado, porém poderiam estar em documento anexo.
- d. Será necessária a criação de um perfil de usuário para efetuar a consulta de vagas? Nesse caso, deverá anexar um documento com as especificações dos dados do perfil.
- e. Poderia implementar parte desse requisito, até que sejam esclarecidos ou complementados os demais dados?

Estudante, você percebeu que os requisitos funcionais podem ser escritos em diferentes níveis de detalhamento. Lembre-se que a imprecisão na especificação de requisitos é a causa de muitos problemas da engenharia de software.

Continue pesquisando e estudando sobre a importância da revisão de especificação de requisitos.

Bons estudos!

## Vamos Exercitar?

Agora que os requisitos estão em plena especificação, muitos já estão concluídos para serem revisados, está na hora de conferir se são compreensíveis aos envolvidos.

Na CaféSaúde os gestores entendem que os requisitos de software serão as principais fontes de definições de procedimentos operacionais e diretrizes para o desenvolvimento das aplicações, desde os cuidados com as plantações de café, processos de secagem, torra e moagem, até as distribuições ao consumidor.

Compreendemos como você efetuará a revisão de requisitos com o analista de sistemas na CaféSaúde. Reúna os documentos base para descrição de requisitos, manual de controle de secagem de café, orientações para colheita de café, outros dados históricos dos processos manuais registrados pelos supervisores de campo e de máquinas, enfim, fontes seguras e confiáveis para consultas e verificação do conteúdo.

O checklist, ferramenta da gestão da qualidade, segundo IIBA BABOK v.3 (2015), pode ser útil para a tarefa de verificação de requisitos. Poderá conter itens padrões de elementos de qualificação, adicionando outros apropriados aos negócios da CaféSaúde, que podem indicar problema ou preocupação aos gestores.

Um aspecto a ser lembrado, nessa revisão os desenvolvedores do sistema podem interpretar o requisito de maneira diferente e implementar uma funcionalidade. Ao especificar a temperatura do ambiente poderá definir como oF, no entanto deveria ser oC, o que pode causar falhas graves em análise futura, caso não façam as devidas conversões. Outra inconsistência de imprecisão na especificação seria a implementação de valores inteiros (sem casa decimal), sendo que os padrões indicam o uso de uma, para os casos de produtos da cadeia alimentar (por exemplo, 24,7°C).

Outro exemplo aplicado aos requisitos da CaféSaúde, ao revisar a especificação do requisito: “ao concluir a inclusão da movimentação de lote colhido em campo, o sistema deve ser eficiente para integrar os dados aos demais processos da colheita”, sugira que o termo “eficiente” seja compreensível pelos engenheiros de software.

Nesse caso, poderia ser substituído para “o sistema deve exibir, num quadro de informações, a quantidade do café colhido sendo acumulada à quantidade total na safra e a média do percentual de umidade do café na safra”.

Estudante, pelo exemplo vivenciado nesse caso da CaféSaúde, você percebeu a importância na revisão de especificação de requisitos de software. Aprenda mais com pesquisas e experiências em alinhamento com as diretrizes organizacionais.

Bons estudos!

## Saiba mais

I. **Saiba mais sobre: características de requisitos.** Leia a página 143.

[Padrão Essencial Global do IIBA para Análise de Negócios](#)

II. **Saiba mais sobre: atividades de verificação.** Leia as páginas 143 e 144.

[Padrão Essencial Global do IIBA para Análise de Negócios](#)

III. **Saiba mais sobre: visão geral da verificação .**Leia o item “2.2.1, páginas 82 a 84.

[Engenharia de Software - Projetos e Processos](#)

## Referências

IIBA - International Institute of Business Analysis. **Padrão Essencial Global do IIBA para Análise de Negócios.** Toronto, Canadá. Disponível em: [https://bit.ly/anasis\\_bacorestd](https://bit.ly/anasis_bacorestd) Acesso em: 28 set. 2024.

KERR, E. D. (org.). **Gerenciamento de requisitos.** São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PAULA FILHO, W. de P. **Engenharia de Software - Projetos e Processos - V. 2.** 4. ed. Rio de Janeiro: LTC, 2019. E-book. Disponível em:  
<https://integrada.minhabiblioteca.com.br/reader/books/9788521636748/> Acesso em: 7 dez. 2024. [Minha Biblioteca]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software.** 9t. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos.** Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos:** software orientado ao negócio. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

WASHIZAKI, H. **Guide to the Software Engineering Body of Knowledge v4.0.** Waseda University: 2024. Disponível em: [https://bit.ly/engreq\\_swebokv4](https://bit.ly/engreq_swebokv4) Acesso em: 15 nov. 2024.

## Aula 2

Validação de Requisitos

### Validação de requisitos



## Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

### Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula você irá conhecer a **validação de requisitos**, conteúdo importante para a sua prática profissional. Você verá os métodos, os processos e as técnicas de validação de requisitos.

## Ponto de Partida

Os gestores da CaféSaúde estão empenhados no desenvolvimento do sistema de gerenciamento da cadeia produtiva de café Standard, Premium e High Special. Para tanto, você ficará responsável por explicar a eles como será a validação de requisitos.

Para ampliar a qualidade dos requisitos, eles podem passar por um processo de validação de requisitos, métodos de validação de requisitos e técnicas de validação de requisitos.

Continue pesquisando e atualizando seu conhecimento sobre os processos de validação de requisitos para aplicar no cotidiano profissional.

Bons estudos!

## Vamos Começar!

Estudante, bem-vindo ao conteúdo sobre a validação de requisitos!

A validação de requisitos é uma tarefa em que os envolvidos (analistas de sistemas, stakeholders, engenheiros de software e demais pessoas) concluirão que o requisito de software foi implementado corretamente, garantindo que o projeto se alinha aos requisitos de negócios e suportam a entrega do valor necessário, conforme IIBA (2015).

O alinhamento deve ser percebido e garantido desde os requisitos de negócios, os designs de projeto, os requisitos das partes interessadas e a solução, até a transição. Vejamos uma situação comum, em que dois stakeholders têm expectativas diferentes diante de uma necessidade (algumas vezes chegam ser conflitantes). Tais diferenças vão aparecer somente na validação dos requisitos porque são validados simultaneamente. Fica evidente, quando o estado

futuro desejado é exposto, que todas as partes interessadas podem defender a sua posição para chegar a esse estado futuro.

Iniciando a validação de requisitos, segundo SWEBOK® GUIDE V4.0 (2024), deve-se estar preparado para encontrar requisitos que não fazem parte do escopo, ou do interesse do usuário final. Um questionamento importante é: existem necessidades ou problemas relatados pelo usuário final sem requisito especificado? Algum requisito especificado não representa as necessidades das partes interessadas?

Ao final do processo de validação de requisitos, espera-se obter a confiança no trabalho, pois os stakeholders sabem que as suas necessidades foram especificadas em requisitos de software.

## Métodos de validação de requisitos

O fluxo das tarefas de validação de requisitos é uma escolha natural para seguir um processo de trabalho, segundo IIBA (2015), nomeando vários métodos para validação de requisitos, conforme podemos mostrar:

1.Identificar suposições (Identify Assumptions): única forma de validar um produto ou serviço inéditos (sem precedentes), as organizações se beneficiam, mesmo sem experiências anteriores confiáveis, por meio de deduções sobre a resposta do cliente ou das partes interessadas

- a. Pode ser difícil ou impossível provar a origem de problemas, em situações específicas, derivadas de alguma causa raiz.
- b. Quando as partes interessadas assumem benefícios que resultarão da implementação de requisitos, tais premissas são identificadas e definidas para que os riscos associados possam ser gerenciados.

2.Definir critérios de avaliação mensuráveis (Define Measurable Evaluation Criteria): os analistas de negócios definem os critérios de validação que serão usados para avaliar o sucesso do resultado após a implementação dos requisitos.

- a. As métricas de linha de base podem ser estabelecidas com parâmetros do estado atual e as métricas-alvo podem ser desenvolvidas para refletir a realização dos objetivos de negócios ou alguma outra medida de sucesso.
- b. Normalmente, os benefícios esperados são definidos como parte do estado futuro, porém nem sempre podem ter estabelecidos os critérios de medição específicos e o processo de validação.

3.Avaliar o alinhamento com o escopo da solução (Evaluate Alignment with Solution Scope): os benefícios de uma implementação nem sempre serão aceitos como solução desejada pelo stakeholder.

- a.Um requisito pode ser eliminado por não trazer benefícios para um stakeholder.

b. Quanto aos requisitos que não se alinham ao escopo, o estado futuro deve ser reavaliado:

O escopo da solução deve ser alterado; ou

c. O requisito deve ser removido do escopo da solução.

Se um projeto não puder ser validado para dar suporte a um requisito, pode haver um requisito mal compreendido ou ausente ou o projeto deve ser alterado.

Para IIBA (2015), o analista de negócios deve manter o foco nas diretrizes que auxiliam nas atividades de validação de requisitos:

- Objetivos de negócios: garantir que os requisitos estejam alinhados e oferecerão os benefícios de negócios desejados.
- Descrição do estado futuro: auxiliar na garantia de que os requisitos, que fazem parte do escopo da solução, alcançarão o estado futuro desejado.
- Valor potencial: uma referência de valor, a ser fornecido pela implementação por um requisito, que possa ser validado.
- Escopo da solução: garantir que os requisitos que fornecerão benefícios estejam dentro do escopo da solução desejada.

## Processos de validação de requisitos

O processo de validação está relacionado diretamente aos objetivos da gestão da qualidade, segundo Paula Filho (2019), pela qual as atividades seguem um fluxo com várias atividades controladas e resultados esperados conforme padrões de conformidades (Consultar modelos de maturidades como: CMMI e MPS-BR).

Vejamos um processo de validação comum, mas que deve ser adaptado para projetos especiais, quando essa necessidade for percebida pelos seus gestores:

1. Identificar os produtos de trabalho a serem validados.

2. Desenvolver e implementar uma estratégia de validação.

Por meio da estratégia na validação vai proporcionar redução de custos, pois os problemas identificados em fases iniciais são menos dispendiosos para corrigir; as sessões de validação promovem alinhamento e consenso entre os stakeholders; a determinação de uma abordagem a ser utilizada (revisões formais, ou sessões colaborativas); a definição de cronograma e os marcos para a validação de requisitos.

3. Identificar os critérios e procedimentos para validação e estabelecimento de um ambiente para validação.

Cada requisito poderá exigir critérios específicos pela característica, porém é comum utilizar a aprovação de usuários e demais stakeholders; um resultado mensurável para validar o requisito (ser verificável); deve estar diretamente alinhado aos objetivos estratégicos do negócio.

4. Executar a validação para garantir que o produto esteja pronto para uso no ambiente pretendido.

5. Identificar e registrar os problemas

Uma documentação organizada para o entendimento e a resolução eficaz, contendo características dos problemas, tais como: identificador único para rastreamento futuro; descrição com detalhes claros sobre o problema; classificação do problema (ambiguidade, inconsistência, incompletude etc.); stakeholders envolvidos; sugestões para resolução do problema; e data, status, e o responsável pela correção.

6. Disponibilizar dos resultados da validação.

Os envolvidos devem ser comunicados com o detalhamento do resultado da validação.

7. Demonstrar a prontidão do produto para o uso pretendido.

Assim que os apontamentos da validação forem resolvidos, deve-se concluir e divulgar a resolução, atualizando o status do identificador único do problema registrado.

O processo de validação, segundo SWEBOK v.4 (2014), em métodos ágeis são praticados continuamente, a cada iteração com a entrega das implementações, visando melhorias no produto (exemplo do Scrum):

- Denominado revisão do sprint pela qual é demonstrado o que foi alcançado para o proprietário do produto (product owner) e outros stakeholders.
- Atividade em que o time de desenvolvimento recebe os primeiros feedbacks sobre a aplicação e dos requisitos de software.
- São discutidas as mudanças e as novas ideias/soluções em função da validação realizada.
- Utiliza-se os critérios de "prontos" para que o software receba o status de validado.
- Nessa revisão espera-se a colaboração, transparência, melhoria contínua, cultura de comunicação aberta e a responsabilidade compartilhada.

Considera-se a validação terminada quando as funcionalidades validadas do produto foram aprovadas tanto nos testes de sistema, indicando que estão de acordo com os requisitos de software, segundo Paula Filho (2019), incluindo os casos de testes de sistemas, a avaliação de uso em conformidade com os requisitos esperados e a usabilidade do produto, considerando-se inclusive a documentação para uso operacional.

**Siga em Frente...**

## Técnicas de validação de requisitos

Segundo IIBA BABOK v.3 (2015), deve-se considerar que os benefícios das técnicas de validação de requisitos estão relacionados diretamente com a participação de todos os stakeholders, ou seja, eles têm a responsabilidade primária de determinar se os requisitos estão validados ou não. Entende-se por partes interessadas: o analista de negócios (e demais membros do time de desenvolvimento), o cliente, os usuários finais e os patrocinadores.

Elencamos algumas técnicas, conforme IIBA BABOK v.3 (2015), para validação de requisitos de software:

- **Critérios de aceitação e avaliação:** usados para definir as métricas de qualidade que devem ser atendidas para alcançar a aceitação por uma parte interessada. Quando se definem critérios mensuráveis e testáveis permitem a avaliação objetiva e consistente de soluções, em todos os níveis, desde o nível alto até um nível mais detalhado, ilustrado na Figura 1.

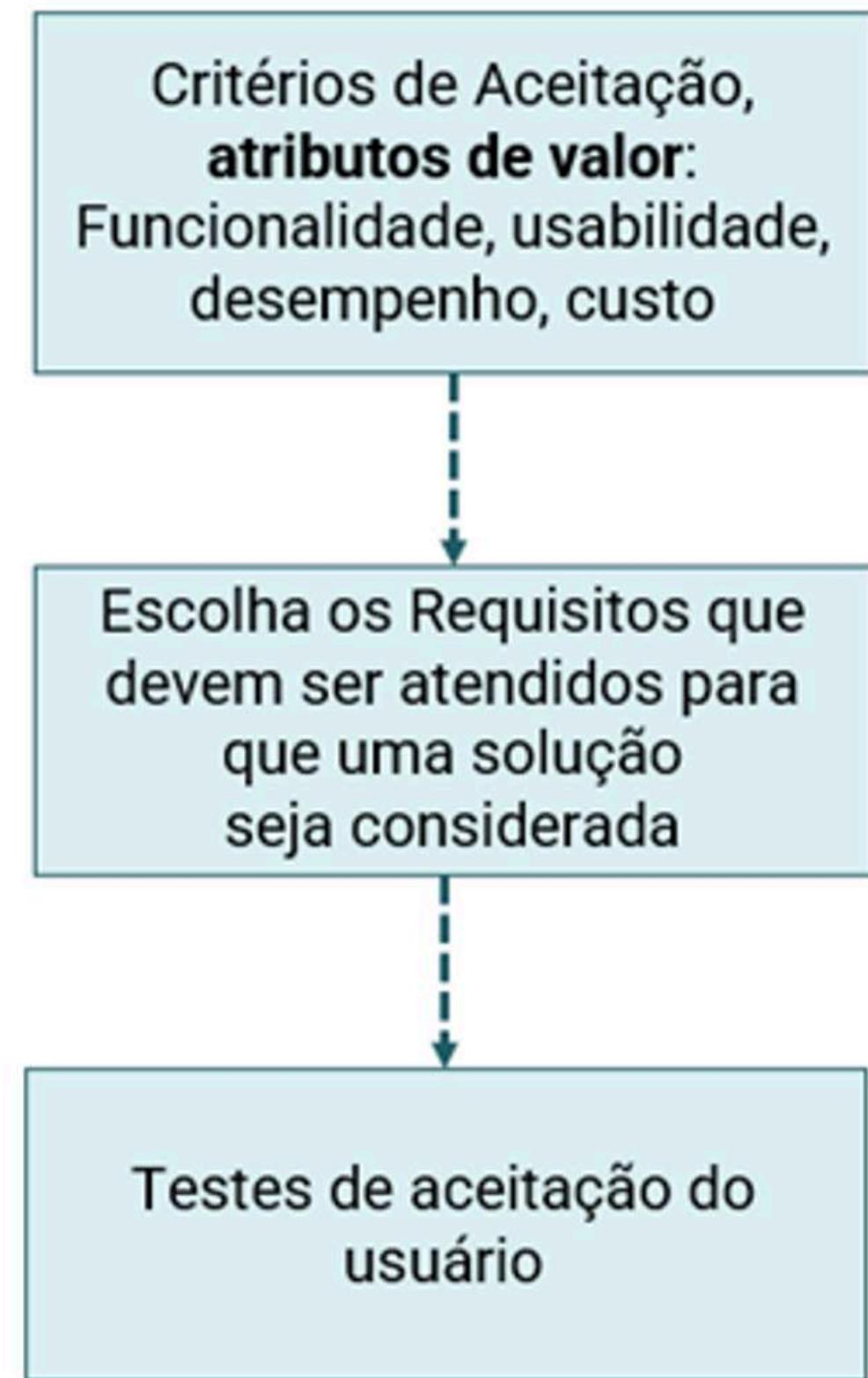


Figura 1 | Critérios de aceitação e avaliação. Fonte: adaptado de IIBA BABOK v.3 (2015).

Os critérios de aceitação e avaliação definem medidas de atributos de valor a serem usados para avaliar e comparar soluções:

1. Os critérios de aceitação descrevem o conjunto mínimo de requisitos que devem ser atendidos para implementar uma determinada solução ou componente de uma solução. Normalmente, esses critérios de aceitação quando se avalia apenas uma solução possível e, geralmente, são expressos como aprovado ou reprovado.
2. Os critérios de avaliação definem um conjunto de medidas que permitem classificar soluções de acordo com seu valor aos stakeholders. Cada critério de avaliação representa uma escala contínua ou discreta para medir um atributo específico da solução, como: custo, desempenho, usabilidade e quão bem a funcionalidade representa as necessidades das partes interessadas. Os atributos que não podem ser medidos diretamente são avaliados usando julgamento de especialistas ou várias técnicas de pontuação.

Tanto os critérios de avaliação quanto de aceitação podem ser definidos com os mesmos atributos de valor. Ao aceitar uma solução, normalmente, os critérios são escritos usando requisitos mínimos de desempenho e limites máximos de custo em acordos contratuais e testes de aceitação do usuário. Registre o resultado da validação.

- **Análise de documentos:** identificar necessidades de negócios previamente documentadas para validar os requisitos, conforme ilustrado na Figura 2. Usada para obter informações de análise de negócios, incluindo compreensão e requisitos contextuais, examinando os materiais (correlacionados) disponíveis que descrevem o ambiente de negócios ou os ativos organizacionais existentes.

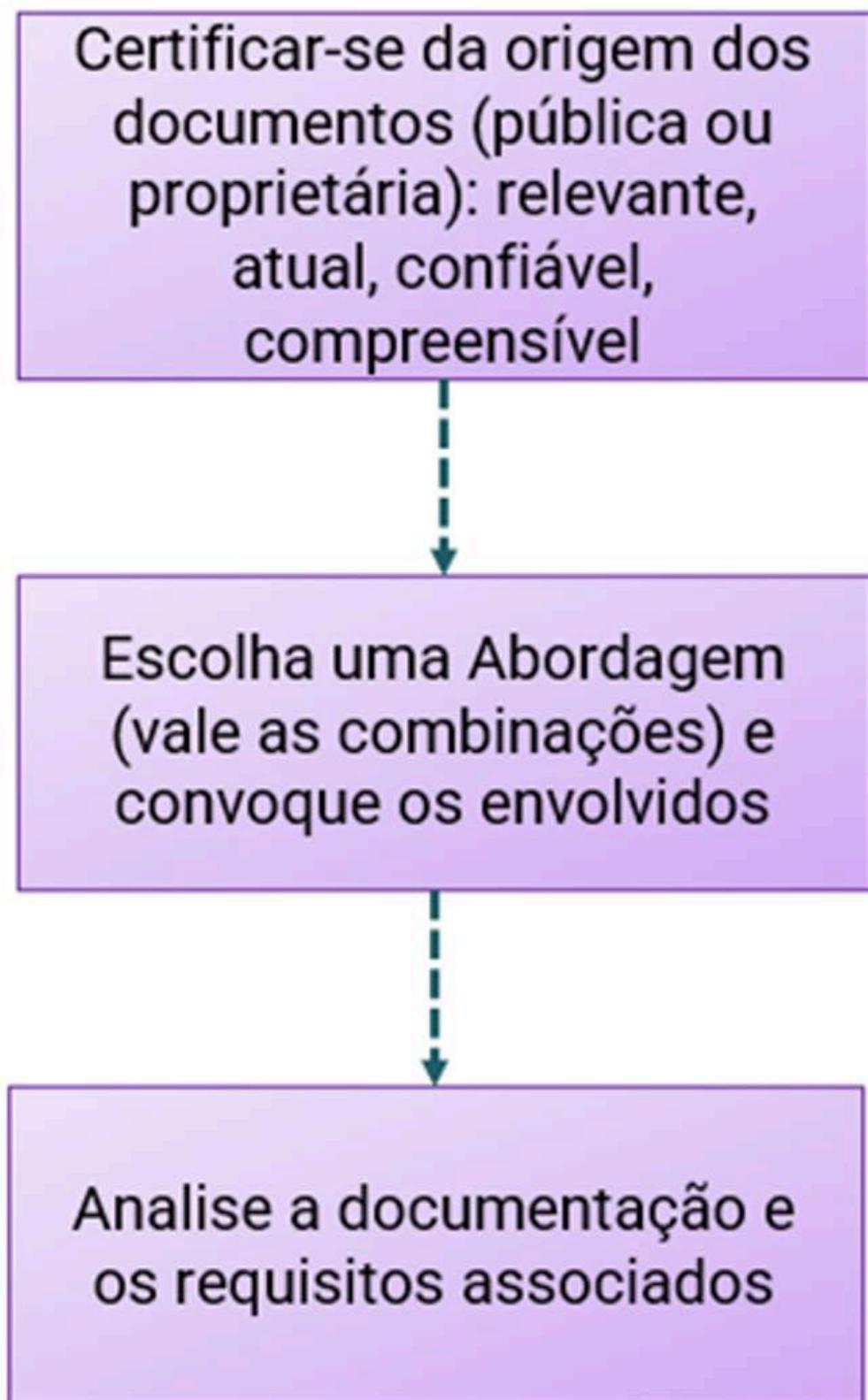


Figura 2 | Análise de documentos. Fonte: adaptado de IIBA BABOK v.3 (2015).

São inúmeras as abordagens, para aproveitar ao máximo a análise de documentos, com informações relevantes aos requisitos de negócios e, consequentemente, suporte direto aos requisitos de software. Elencamos algumas abordagens, com os respectivos objetivos dessa técnica:

- a. Coleta de informações básicas: apenas para compreender melhor o contexto de uma necessidade de negócios, ou aproveitar as soluções existentes (lições aprendidas).
- b. Confronto de informações elicitadas: comparando com especificações, entrevistas, observações, brainstorming, etnografia, entre outras realizadas, para dirimir dúvidas.
- c. Mineração de dados: uma análise de dados a fim de determinar padrões, agrupamento de dados em categorias e identificar/determinar oportunidades de mudança.
- d. Revisão metódica: analistas de negócios podem registrar as associações de informações determinantes com outros artefatos do trabalho de especificação de requisitos, documentação de treinamento, diagramação de requisitos, ou relatório de problemas.
- e. Lacunas entre requisitos: auxiliar na resolução de lacunas de informações, na falta de especialistas no assunto (SME- subject matter experts), para completar atributos de soluções.

Ao planejar a análise de documentos, registre todos os passos para eventuais consultas e conformidades, principalmente o resultado da análise, conforme a abordagem escolhida, em cada requisitos de software associado ao documento. Ainda, o analista de negócios pode indicar:

- O nível de detalhe do conteúdo ser suficiente e apropriado aos desenvolvedores de software e demais envolvidos.
- O documento é um recurso visual importante, portanto deve ser transformado em gráficos, modelos, fluxos de processos ou tabelas de decisão, a fim de auxiliar na compreensão.

Estudante, você percebeu que os requisitos funcionais devem ser validados para certificar que o produto de software está sendo entregue, conforme as expectativas do cliente e demais stakeholders.

Continue pesquisando e estudando sobre a importância da validação de requisitos de software.

Bons estudos!

## Vamos Exercitar?

Agora que os requisitos de software foram efetivamente utilizados pelo time de desenvolvimento, as especificações das funcionalidades foram implementadas em produto de software, é chegada a hora da validação dos requisitos.

Na CaféSaúde, os gestores entendem que os requisitos de software foram responsáveis pela construção do sistema, sendo necessária uma atividade de grande importância no processo de

desenvolvimento de software.

Vamos praticar como a validação de requisitos será útil para aumentar a qualidade do software, alinhado às necessidades dos usuários da CaféSaúde.

Vejamos o requisito de software: "o sistema deve permitir que os usuários de campo redefinam suas senhas".

Embora seja um requisito de software com pouca complexidade, vamos elencar alguns critérios de aceitação, como um exemplo da prática da validação de requisitos:

- a. O usuário poderá solicitar a redefinição de senha informando seu e-mail cadastrado.
- b. O sistema deve enviar um link de redefinição de senha ao e-mail do usuário em até 5 minutos.
- c. O link de redefinição deve expirar após 10 minutos, por segurança.
- d. A nova senha deve ter pelo menos 8 caracteres, incluindo uma letra maiúscula, pelo menos um número e um caractere especial.
- e. O sistema deve confirmar a alteração da senha ao usuário, inclusive enviando e-mail informando a ação do usuário.

Analisando que o sistema desenvolvido reflete o que os usuários esperam, conforme as especificações do requisito de software, alinhados aos objetivos de negócios, evidenciado que atendem aos critérios de aceite e pronto para ser colocado em operação, concluímos que os requisitos de software e o sistema estão validados.

Experimentemos métodos para validação de requisitos, conforme SWEBOK (2024), usados pelos analistas de negócios: inspeção de requisitos, simulação e execução e prototipagem.

1. Inspeção de requisitos: inspecionando a especificação do requisito, pessoas preparadas para procurar erros, omissões, suposições inválidas, falta de clareza e desvio de aceitação, podendo ser inspecionado sob várias perspectivas:

Diretrizes para validação	Resultado da validação
As necessidades de clientes, usuários e consumidores estão representados de forma completa e precisa, sob o crivo do analista de sistemas.	Os colhedores de café recebem as informações e suas devidas participações conforme combinado. Os supervisores de campo executam suas atividades apoiados pelo sistema. Operadores de máquinas são orientados pelo sistema com informações corretas.
Solicitar uma inspeção a engenheiros de software mais experientes, quanto ao documento estar claro e em conformidade com os padrões aplicáveis.	Os responsáveis pela qualidade do projeto verificam os documentos e especificações dos requisitos do processo da colheita e secagem

	de café estão em conformidade com os padrões definidos pelo CMMI.
Arquitetos de software, responsáveis pelo projeto e a construção do software inspecionam se o documento é suficiente para apoiar seu trabalho.	As especificações de requisitos dos processos de colheita e secagem do café possuem todos os elementos para o design do projeto e demais detalhes das regras de negócios para implementar o sistema.

2. Simulação e execução: aos stakeholders é melhor perceber os requisitos em execução do que a inspeção da especificação do requisito; muitas pessoas acreditam que inspecionar uma especificação em detalhes é demasiado superficial.

Especificação do requisito	Execução do requisito
O usuário pode redefinir sua senha.	<p>Existe uma função sempre ativa para redefinir a senha, quando acionada pelo usuário, é exibida uma janela para informar o e-mail do usuário.</p> <p>O sistema prepara um formulário para redefinição de senha e envia o link para este formulário através do e-mail informado.</p> <p>O usuário aciona o link recebido em seu e-mail, o sistema abre uma janela para que o usuário informe uma nova senha.</p> <p>O sistema atualiza a nova senha assim que o usuário confirmar e enviar a redefinição da senha.</p>
O link de redefinição deve expirar após 10 minutos, por segurança.	O sistema impede o uso do formulário quando o usuário não acionar o link de redefinição de senha após 10 minutos.
A nova senha deve ter pelo menos 8 caracteres, incluindo uma letra maiúscula, pelo menos um número e um caractere especial.	O sistema mostra uma mensagem com a instrução para criação da nova senha: “a palavra-chave não atende ao requisito”, “a senha deve ter pelo menos 8 caracteres, incluindo uma letra maiúscula, pelo menos um número e um caractere especial”

Fornecer checklist (listas de verificação), critérios de qualidade ou uma "definição de pronto" aos inspecionadores pode orientá-los a se concentrar em aspectos peculiares da especificação de requisitos.

Estudante, pelo exemplo vivenciado nesse caso da CaféSaúde, você percebeu a importância na validação de requisitos de software. Esteja preparado cada vez mais, consultando pesquisas e experiências em alinhamento com as diretrizes organizacionais.

Bons estudos!

## Saiba mais

I. Saiba mais sobre: **validação de requisitos de software**. Leia as páginas 246 a 250.

[Engenharia de requisitos](#)

II. Saiba mais sobre: **validação de requisitos e as partes interessadas**. Leia as páginas 250 a 251.

[Engenharia de requisitos](#)

III. Saiba mais sobre: **validação de requisitos**. Leia as páginas 123 e 124.

[Engenharia de software](#)

IV. Saiba mais sobre: **validação de requisitos**. Leia a página 57.

[Guide to the Software Engineering Body of Knowledge v4.0](#)

## Referências

IIBA - International Institute of Business Analysis. **Padrão Essencial Global do IIBA para Análise de Negócios**. Toronto, Canadá, 2015. Disponível em: [https://bit.ly/anasis\\_bacorestd](https://bit.ly/anasis_bacorestd) Acesso em: 28 set. 2024.

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

MASCHIETTO, L. G. et al. **Desenvolvimento de software com metodologias ágeis**. Porto Alegre: SAGAH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556901824/> Acesso em: 9 dez. 2024. [Minha Biblioteca]

PAULA FILHO, W. de P. **Engenharia de Software - Projetos e Processos** - V. 2. 4. ed. Rio de Janeiro: LTC, 2019. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788521636748/> Acesso em: 7 dez. 2024. [Minha Biblioteca]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos: software orientado ao negócio**. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

WASHIZAKI, H. **Guide to the Software Engineering Body of Knowledge v4.0**. Waseda University: 2024. Disponível em: ou [https://bit.ly/engreq\\_sweboky4](https://bit.ly/engreq_sweboky4) Acesso em: 15 nov. 2024.

## Aula 3

Documentação de Requisitos

### Documentação de requisitos

#### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer a **documentação de requisitos**. Este conteúdo é importante para a sua prática profissional, pois você aprenderá sobre os documentos da elicitação, da especificação e da estruturação do documento de requisitos. Prepare-se para essa jornada de conhecimento! Vamos lá!!!

### Ponto de Partida

A documentação de requisitos de software é um dos maiores desafios na CaféSaúde, pela sua complexidade de estruturação, falta de pessoal especializado em criar e manter um padrão de documentos formais, entre outros fatores inerentes ao desenvolvimento de software.

Os gestores da CaféSaúde estão muito preocupados com o futuro do sistema na fase de manutenção do software que está sendo desenvolvido agora. Embora, o principal motivador para elaborar a documentação da especificação de requisitos seja o desenvolvimento do software, sabemos que a base para a comunicação das regras de negócios, em formato de requisitos de software formalizado, será importante após a implantação.

Estudante, o seu desafio referente à documentação é explanar, com alguns exemplos, a importância das documentações de elicitação e especificação de requisitos por meio de uma estrutura padronizada e organizada para requisitos de software.

Assim, os gestores poderão manter o seu foco nos negócios da CaféSaúde, no gerenciamento da cadeia produtiva de café Standard, Premium e High Special.

Este exercício prático aumentará a sua competência em engenharia de requisitos, auxiliando nas habilidades em elaboração de documentações de requisitos de software.

Bons estudos!

## Vamos Começar!

Estudante, bem-vindo ao conteúdo sobre a documentação de requisitos!

Os documentos de requisitos são fundamentais em momentos diferentes do ciclo de vida de um sistema. A curto prazo, desempenha papel tipicamente para a construção do software, por possuir diretrizes e definições das soluções em sistemas. A longo prazo, nem sempre conhecido, um documento de requisito tem papel importante para a manutenção do sistema.

## Documentação da elicitação e da especificação de requisitos

Percebemos maior utilidade da documentação de requisitos na fase de projeto e construção de software, sendo a principal fonte de conhecimento do sistema para tal fim. Na fase de manutenção e evolução do software, o engenheiro de software conhecerá as funcionalidades do sistema por meio da execução, debugging ou teste de caixa-branca, contudo, há uma questão: o software está consistente e alinhado aos objetivos de negócios e conforme as especificações originais aprovadas?

Explanemos alguns dos dilemas em software, segundo SWEBOK® GUIDE V4.0 (2024), quando se trata de documentação de requisitos e operação do sistema:

- O que esse código faz é o que pretende fazer: pode se chamar de bug ou até de defeito, apenas porque existe uma diferença observável entre o que o software se destina a fazer e o que ele faz.
- Se os engenheiros de software da fase de manutenção/evolução não são os mesmos do desenvolvimento; o papel da documentação de requisitos torna-se a principal base de conhecimentos formal, pois as intenções para o software foram capturadas e comunicadas por ela.

SWEBOK® GUIDE V4.0 (2024) ainda complementa que os documentos dizem respeito a requisitos de software e ao gerenciamento desses requisitos ao longo da vida útil do software, portanto às inúmeras mudanças que ocorreram. Compreendamos dois benefícios das documentações:

- A documentação de requisitos de software não é uma atividade discreta de front-end do ciclo de vida de desenvolvimento de software, mas sim um processo desde o início de um projeto que continua a ser refinado ao longo de toda a vida útil do software.
- Essas documentações precisam ser adaptadas à organização, ao contexto dos negócios e ao projeto de software.

Pensemos o quanto é benéfico um documento de requisitos de software para esclarecer dúvidas e recuperar o conhecimento em requisitos de software. Segundo SWEBOK® GUIDE V4.0 (2024), quando o resultado da execução do sistema apresenta diferenças às expectativas do stakeholder, temos os documentos das especificações de requisitos elaboradas por meio das atividades da elicitação com os stakeholders da época, conforme ilustra a Figura 1.

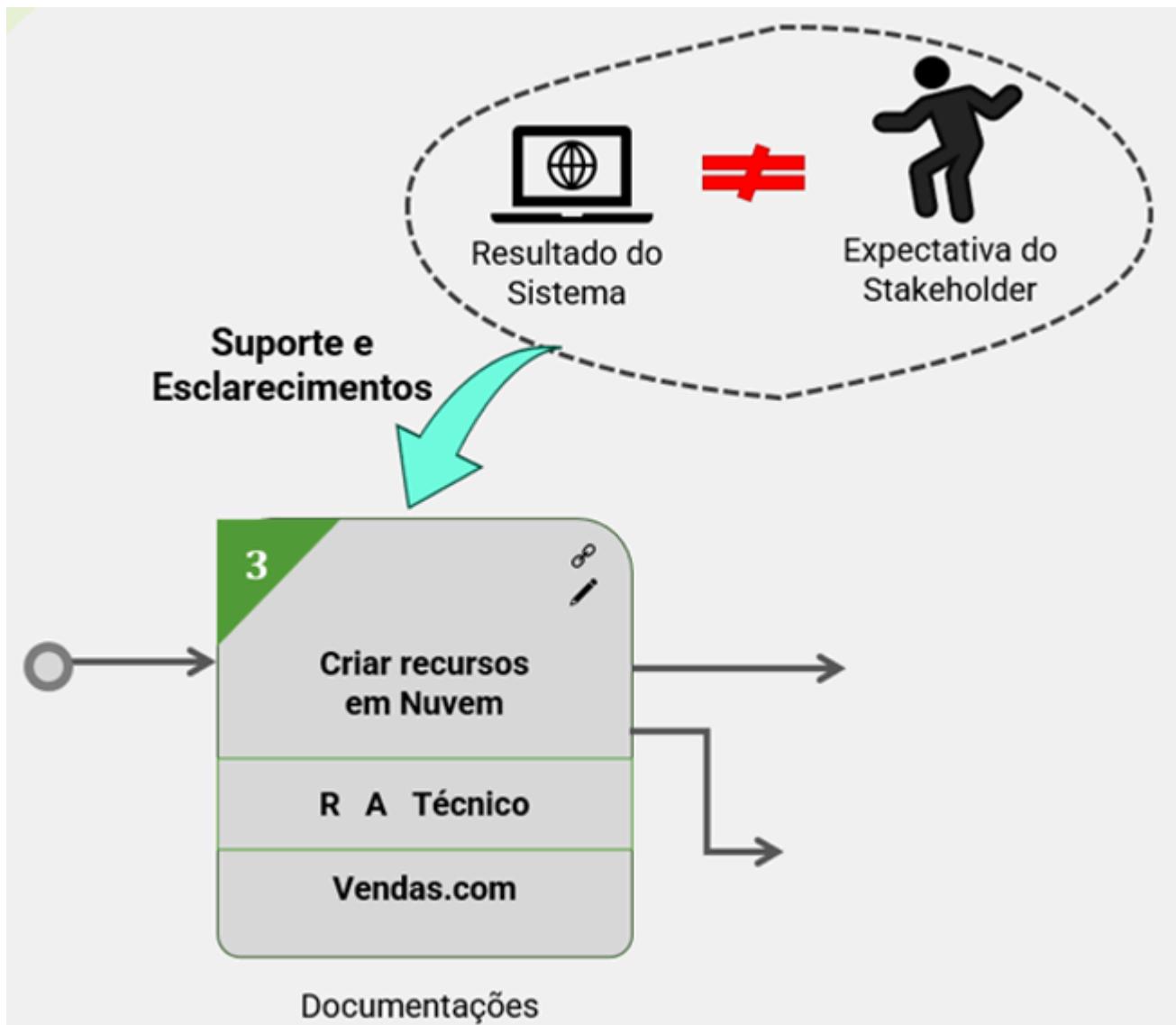


Figura 1 | Divergências esclarecidas pelas especificações de requisitos.

O questionamento é muito forte: sem a documentação de requisitos de software, como poderíamos esclarecer se o produto de software foi desenvolvido corretamente? Portanto, é primordial que a documentação da especificação de requisitos esteja fiel aos desejos dos stakeholders.

Segundo SWEBOK® GUIDE V4.0 (2024), a engenharia de requisitos caracteriza-se pelo tratamento sistemático de requisitos, sendo a sua área de conhecimento relacionada diretamente a arquitetura, design, implementação, teste e manutenção de software, ficando mais próxima aos modelos e métodos de engenharia de software por oferecer alto valor na especificação de requisitos, em forma de modelos.

Assim, as especificações são parte formal das necessidades dos stakeholders, suportando a comunicação entre os envolvidos e possibilitando o gerenciamento dos requisitos de software.

ao longo do ciclo de vida do desenvolvimento do sistema. Os documentos das especificações podem ser modificados a qualquer momento, pois sabemos que depende das alterações dos requisitos de negócios. Contudo, o seu uso está associado ao ciclo de vida, processo da engenharia de software que determina quando e como os requisitos passam a ter funcionalidade no software. Vejamos:

- No ciclo de vida em cascata, todos os requisitos são essencialmente elicitados e especificados em uma fase de requisitos discreta. Quando eles estiverem substancialmente concluídos, os documentos de requisitos são utilizados nas tarefas de arquitetura, concepção e construção, e assim sucessivamente.
- Em alguns ciclos de vida iterativos, a construção da documentação de requisitos ocorre em duas etapas: (a) inicialmente, os requisitos são especificados em alto nível, (b) na fase de elaboração de requisitos são completados com o detalhamento necessário das necessidades dos usuários e dos negócios.
- No ciclo de vida ágil, os requisitos de software são especificados de forma incremental, *just in time*; à medida que cada elemento da funcionalidade é adicionado, a documentação sofre complemento e atualização com o detalhamento do incremento.

Por outro lado, se analisarmos um documento de requisitos (a sua especificação, a modelagem e seus anexos), não somos capazes de discernir com exatidão qual o ciclo de vida usado no desenvolvimento. Todavia, os aspectos inerentes à documentação de elicitação e de especificação de requisitos diferem apenas pela transição, entre a coleta/captura dos requisitos e o detalhamento/refinamento deles. Acompanhemos o Quadro 1 com os aspectos que caracterizam as atividades de elicitação e de especificação e seus respectivos documentos.

Aspectos	Documentação da elicitação	Documentação da especificação
Objetivo	Anotações da captura e compreensão	Refinar e formalizar em documento
Detalhamento	Baixo e informal	Alto e estruturado
Conteúdo	Necessidades do negócio, problemas e ideias pessoais	Requisitos funcionais e não funcionais refinados, com critérios de aceitação
Formato	Informal ou semiformal	Padronizado e formal
Técnicas	Entrevistas, workshops, brainstorming, etnografia	Protótipos, histórias, modelagem, caso de uso
Artefato	Material bruto, base para pesquisas	Gerenciável, rastreável, documento principal da comunicação
Participantes	Analista de sistemas e stakeholders	Engenheiro de requisitos e de software, analista de

sistemas, stakeholders,  
gerente de projeto e  
time de desenvolvimento

Quadro 1 | Caracterização da elicitação e especificação quanto aos documentos.

Revisando o entendimento dos documentos, temos que: enquanto os requisitos estão sendo elicitados, em uma dinâmica informal, exploratória e idealizadora, aos responsáveis pela especificação de requisitos compete a tarefa a jusante (de refinamento), em formato ou modelo documental, para suportar as atividades de arquitetura e design do software.

## Siga em Frente...

## Estruturação do documento de requisitos

Sabemos que a documentação de requisitos de software é fundamental para assegurar que todos os envolvidos com o desenvolvimento de sistemas compreendam e concordem com o que está sendo desenvolvido, segundo SWEBOK® GUIDE V4.0 (2024).

Em muitos casos, a documentação de requisitos faz parte de um contrato entre a equipe de desenvolvimento e os stakeholders, pois tem o detalhamento do que o sistema deve entregar e as restrições sob as quais deve operar.

Espera-se de documentações de software garantam a eficiência no desenvolvimento de software por oferecer:

- Claridade dos requisitos: com especificação clara e precisa do que o sistema deve entregar.
- Comunicabilidade: facilitar a comunicação entre stakeholders, desenvolvedores, testadores e demais partes interessadas.
- Base para o desenvolvimento e testes: possuir diretrizes, restrições, detalhamento necessário para a implementação do sistema e a criação de casos de teste.
- Gerenciamento de mudanças: auxiliar nas mudanças de requisitos de negócios e de software, ao longo do ciclo de vida do projeto.
- Gestão da qualidade e de riscos: dando suporte para a rastreabilidade e gerenciamento de configuração de software.

Uma estrutura da documentação de requisitos deve ser definida durante a fase de planejamento das atividades de elicitação e especificação de requisitos. Elencamos alguns elementos importantes a constar na documentação de requisitos.

1. Iniciar com uma introdução sobre o objetivo do documento, a sua importância e uma visão geral das funções do sistema a ser desenvolvido.
2. Definir o escopo do sistema, explicitando o que pertence e o que está fora do escopo.

3. Descrever as restrições e premissas:
4. Restrições técnicas: as limitações relacionadas à tecnologia que afetam o desenvolvimento do sistema.
5. Restrições de negócio: as restrições de negócio que devem ser consideradas em fase de especificação e entrega do software.
6. Premissas: declaração das condições feita durante a elicitação de requisitos que devem ser seguidas no projeto.
7. Dos requisitos funcionais: cada funcionalidade do sistema deve ser identificada, especificada e mantida, conforme mudanças que ocorrem durante do ciclo de vida.
  - Deve conter elementos essenciais para auxiliar na compreensão da funcionalidade para um requisito funcional e dependerá da metodologia adotada pelo time de desenvolvimento de software.
  - Especificados ou modelados: por caso de uso, UPN (Notação Universal do Processo), BPMN (notação de gerenciamento de processos de negócio), entre outros.
  - Identificar atores: os diferentes tipos de usuários, sistemas ou equipamentos que interagem com o sistema.
  - Normalmente representados por um fluxo principal (com as ações em um caso de uso típico e comum) e complementados por fluxos alternativos (detalhando quaisquer desvios ou exceções que possam ocorrer).
  - Quando escolher as histórias de usuário, devem demonstrar o ponto de vista dos usuários finais, em um formato básico "como [tipo de usuário], eu quero [ação] para que [benefício]".
  - Critérios de aceitação: todo requisito de software deve possuir um ou mais critérios específicos para determinar se a funcionalidade atende ao propósito, com parte mensurável ou observável (com as devidas referências) pelos envolvidos na verificação e validação do requisito.

5. Dos requisitos não funcionais: as diretrizes do software, especificações especiais caracterizada pela natureza do sistema ou contexto do negócio, devem ser explicitadas para serem contempladas a partir do design de projeto do software, tais como:

- Desempenho: relacionadas à performance durante o funcionamento do software como tempo de resposta, taxa de transferência e capacidade de operação.
- Segurança: associada às restrições em acesso ao dado ou ao recurso do sistema, como autenticação, autorização e conformidade.
- Usabilidade: o quanto a aplicação deverá satisfazer o usuário (pela sua experiência) garantindo que o sistema seja intuitivo, otimizado e fácil de usar.
- Compatibilidade: para quais plataformas o software será suportado, ou em quais ambientes operacionais acontecerão as integrações.
- Manutenibilidade: descrever os caminhos e técnicas que contribuirão para facilitar a manutenção e atualização do sistema, com responsáveis, ferramentas, técnicas, entre outros.

6. Anexos e referências: os documentos de requisitos podem ser complementados com outros documentos organizacionais pertinentes como glossário, normatização, regulamentação, leis, artigos, diagramas, fluxogramas, gráficos, manuais, design de interação, entre outros.

A documentação de requisitos é um componente vital para o sucesso de qualquer projeto de software, inclusive é o suporte ideal na fase de manutenção do software. Contudo, deve seguir uma abordagem estruturada, organizada, aplicando as melhores práticas, de maneira clara, precisa e útil a todos os envolvidos no desenvolvimento, teste e manutenção do sistema.

Os padrões em documentação de especificação de requisitos têm outros propósitos, além da facilidade de interpretação e comunicação, a de identificação como um item de configuração (IC) na matriz de rastreabilidade. Um artefato IC é o menor elemento rastreável na área do conhecimento do gerenciamento de configuração de software (GCS). As abordagens de GCS são aplicadas também para gerenciar requisitos e todas as mudanças que ele recebe ao longo do ciclo de vida do software.

Estudante, com os conhecimentos sobre documentação de requisitos, aumenta a sua competência em gerenciamento de requisitos de software, por isso você deve conhecer a gestão desses artefatos para o seu desempenho profissional em análise de sistemas ou como engenheiro de requisitos.

Continue pesquisando e estudando sobre o uso e organização das documentações de requisitos de software.

Bons estudos!

## Vamos Exercitar?

Estudante, sabendo das dificuldades em documentação na CaféSaúde, vamos explanar os principais motivadores para que os envolvidos no desenvolvimento de sistemas possam garantir a criação e manutenção da documentação de especificação de requisitos.

Os objetivos do processo de secagem do principal produto na CaféSaúde são:

- **Qualidade do café:** manter características uniformes dos grãos: tamanho, perfil sensorial, cor; e tornar o grão saudável para evitar a fermentação ou danos térmicos.
- **Eficiência:** o secador rotativo reduz o tempo de secagem comparado ao método exclusivamente ao sol, proporcionando maior controle e previsibilidade.
- **Capacidade:** grandes volumes podem ser processados em um curto intervalo, otimizando custo/recurso e a logística da colheita.

O sistema deverá controlar o processo de secagem, dando possibilidade de, futuramente, analisar os resultados ou até tomada de decisões em tempo real. Portanto, os requisitos de software devem especificar funcionalidades do processo com o detalhamento necessário aos operadores e gestores da CaféSaúde.

Da elicitação	Da especificação
---------------	------------------

Dados importantes na secagem de grãos do café	Registrar: a umidade do grão vindo do campo, umidade na entrada do secador, umidade na saída do processo.
Dados do processo de secagem	Registrar a quantidade de grãos (em quilograma) na entrada do processo, quantidade na saída.
Configuração do equipamento de secagem	Para cada ciclo de secagem: configuração da temperatura, velocidade de rotação e o tempo do ciclo, tempo de resfriamento.
Relatório do processo de secagem	Possibilitar consultar os dados históricos: volume, tempo, temperatura, grau de umidade. Criar gráficos comparativos por período.

Quadro 1 | Documentação da elicitação e da especificação.

Compreendamos as características de um documento de requisito de software, com base no processo de secagem do grão de café da CaféSaúde, com as características de detalhamento, descritas no Quadro 2.

Estrutura	Da documentação
Como introdução sobre o objetivo do documento	Especificar quais funções do sistema devem ser implementadas no software.
Escopo	Cabe ao sistema manter os dados do processo de secagem do grão de café, compreendendo a configuração da secagem, dados de entrada e saída do grão de café.
Restrições e premissas	Deve manter uma unidade de medida padrão para temperatura, peso, demais medidas.  As operações manuais serão substituídas gradativamente para leitores automatizados.  O processo de secagem poderá ser utilizado para outros produtos (diferente do café).
Requisitos funcionais	O sistema deve dar suporte ao operador do secador para informar os dados de configurações do ciclo de secagem, os dados do estado inicial e estado final do grão do café.

Requisitos não funcionais	Manter os dados registrados por dez anos; o armazenamento de dados deve ser em tempo real em nuvem.  Prever uma infraestrutura mínima de funcionamento por um dia, em caso de falta da conexão para o sistema em nuvem.
Anexos e referências	Imagens do equipamento de secagem.

Quadro 2 | Documentação de requisitos de software.

Estudante, percebemos que a descrição detalhada de um processo poderá ter variações na redação pela experiência do analista de sistemas ou pelo estilo individual, porém a estrutura de um documento de requisito deve ser definida num padrão com o time de desenvolvimento. Essa padronização deve ser compreendida tanto pelo time de desenvolvedores quanto pelos demais stakeholders.

Bons estudos!

## Saiba mais

I. **Saiba mais sobre: UPN modelagem de requisitos.** Leia sobre Introdução à UPN.

[Entender a notação de processo universal.](#)

II. **Saiba mais sobre: documentação.** Leia as páginas 283 e 284.

[Guide to the Software Engineering Body of Knowledge v4.0.](#)

III. **Saiba mais sobre: documentação pesada ou intuitiva.** Leia o item Manifesto Ágil, páginas 13 a 15.

[Desenvolvimento de Software com Metodologias Ágeis.](#)

## Referências

IIBA - International Institute of Business Analysis. **Padrão Essencial Global do IIBA para Análise de Negócios.** Toronto, Canadá, 2015. Disponível em: [https://bit.ly/anasis\\_bacorestd](https://bit.ly/anasis_bacorestd) Acesso em: 28 set. 2024.

KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

MASCHIETTO, L. G. *et al.* **Desenvolvimento de software com metodologias ágeis**. Porto Alegre: SAGAH, 2021. E-book. Disponível em: [https://integrada\[minhabiblioteca.com.br\]/reader/books/9786556901824/](https://integrada[minhabiblioteca.com.br]/reader/books/9786556901824/) Acesso em: 9 dez. 2024. [Minha Biblioteca]

PAULA FILHO, W. de P. **Engenharia de Software - Projetos e Processos** - V. 2. 4. ed. Rio de Janeiro: LTC, 2019. E-book. Disponível em: [https://integrada\[minhabiblioteca.com.br\]/reader/books/9788521636748/](https://integrada[minhabiblioteca.com.br]/reader/books/9788521636748/) Acesso em: 7 dez. 2024. [Minha Biblioteca]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. E-book. Disponível em: [https://integrada\[minhabiblioteca.com.br\]/reader/books/9786558040118/](https://integrada[minhabiblioteca.com.br]/reader/books/9786558040118/) Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. E-book. Disponível em: [https://integrada\[minhabiblioteca.com.br\]/reader/books/9786556900674/](https://integrada[minhabiblioteca.com.br]/reader/books/9786556900674/) Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos**: software orientado ao negócio. 1. ed. Rio de Janeiro: Brasport, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

WASHIZAKI, H. **Guide to the Software Engineering Body of Knowledge v4.0**. Waseda University: 2024. Disponível em: [https://bit.ly/engreq\\_swebokv4](https://bit.ly/engreq_swebokv4) Acesso em: 15 nov. 2024.

## Aula 4

Testes de Requisitos

### Testes de requisitos

**Este conteúdo é um vídeo!**



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer os **testes de requisitos**.

Esse conteúdo é importante para a sua prática profissional, pois envolve gerenciar as dependências entre os documentos de requisitos, o plano de testes de requisitos e as técnicas de testes de requisitos.

Prepare-se para esta jornada de conhecimento! Vamos lá!!!

## Ponto de Partida

Para gerenciar as dependências entre os documentos de requisitos, vamos apresentar aos gestores da CaféSaúde as técnicas adequadas inerentes ao processo de desenvolvimento de software (PDS) que garantem a construção do plano de testes de requisitos e suportam a administração das técnicas de testes de requisitos.

Dessa maneira, os gestores da CaféSaúde serão tranquilizados nessa fase final do desenvolvimento do sistema de gerenciamento da cadeia produtiva de café Standard, Premium e High Special. Sendo assim, novamente, você será o responsável por explicar como serão os Testes de Requisitos.

Estudante, com o aprendizado deste conteúdo você estará cada vez mais preparado para atuar em engenharia de requisitos. Porém, continue pesquisando e atualizando seu conhecimento sobre os processos de testes de requisitos para aplicar no seu cotidiano profissional.

Bons estudos!

## Vamos Começar!

Estudante, seja bem-vindo ao conteúdo sobre testes de requisitos!

O envolvimento dos desenvolvedores desde as primeiras elicitações de requisitos facilita o aprendizado das regras de negócios e, consequentemente, a construção do sistema nas fases de projeto e de implementação. Porém, é fundamental o uso das técnicas e ferramentas da gestão inerentes ao processo de desenvolvimento de software, como apresentaremos no gerenciamento de documentos de requisitos e na gestão de testes de software.

## Gerenciar as dependências entre os documentos de requisitos

Inicialmente, relembraremos as informações sobre a rastreabilidade dos artefatos do projeto de software, pela qual o time de desenvolvimento encontra as dependências entre os documentos

de requisitos para a realização da gestão de mudanças de requisitos. Segundo Pressman (2021), a matriz de rastreabilidade é um modo de manter as dependências entre documentação de requisitos, que consequentemente auxiliam nas decisões de arquitetura e na investigação de causas de defeito.

Pressman (2021) complementa que a capacidade de gerenciamento de documentos de requisitos de software está baseada no conhecimento de ferramentas de gestão da configuração de software (GCS) que implementa a matriz de rastreabilidade. Ou seja, tendo suporte de uma ferramenta de GCS, é possível aprofundar nos processos com base em documentos e, ainda, tornam-se gerenciáveis as políticas e estratégias que regem as atividades de testes de requisitos de software.

Essas dependências dos documentos de requisitos são determinadas pela relação que existe entre os requisitos por derivação ou por dependência e também por sucessão, que precisam ser apreciados ao se determinar o impacto de uma alteração proposta (SWEBOK, 2024). Vejamos como ocorre a relação entre requisitos, ilustrado na Figura 1:

- Supondo que o requisito representado no documento IC#002 está especificado e aprovado para o desenvolvimento, porém, por uma questão de negócios do cliente, altera-se a prioridade de uma parte do IC#002, quando então cria-se o IC#011; portanto, este é um requisito derivado do IC#002. Nesse contexto, estabelece-se maior prioridade de desenvolvimento para o requisito IC#011 no objetivo de atender a uma regra específica de cálculo do valor de prestações de compradores de um determinado segmento do mercado.

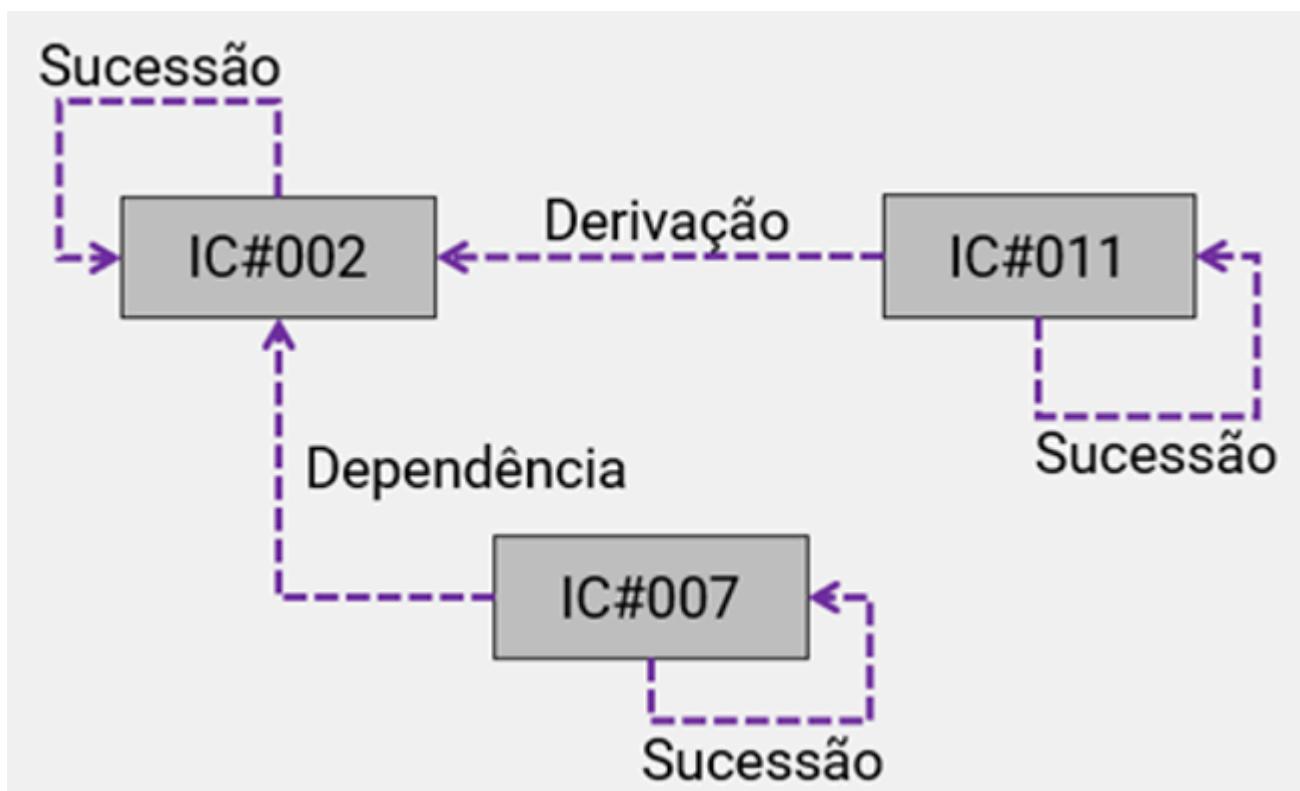


Figura 1 | Dependência e derivação de requisitos. Fonte: adaptado de SWEBOK® GUIDE V4.0 (2024).

Complementando o ilustrado na Figura 1, verifica-se a dependência do requisito IC#007 em relação ao IC#002 (por motivos técnicos ou por prioridade imposta pelos stakeholders, entre outros). Ainda, a relação de sucessão que pode existir para qualquer documento de requisitos (ou outro tipo de artefato) por mudanças, ou correção de conteúdo, caracterizando as versões do artefato controlado pela GCS.

Lembremo-nos que na alteração ocorrida no IC#002, essas mudanças em regras monetárias/fiscais não são comuns, porém, quando a legislação é alterada, as organizações têm prazo definido para cumprir tais determinações econômicas. Sendo assim, todos os requisitos de software relacionados ou dependentes devem ser priorizados na construção, na manutenção e na validação.

Segundo Sousa e Redmiles (2008) *apud* Pressman (2021), considerar a gestão de dependências como gestão de impacto ampara os desenvolvedores a se concentrarem no modo como as alterações ocorridas afetam o seu trabalho.

## Plano de testes de requisitos

Primeiramente, citamos que as documentações de testes estão, hierarquicamente, classificadas em três categorias, segundo SWEBOK (2024, p. 136): documentação de testes organizacional, de gerenciamento e dinâmico, conforme ilustra a Figura 2. Todas as documentações de testes são parte integrante da formalização do processo de teste.

## **Documentação de teste organizacional**

## **Documentação de gerenciamento de teste**

## **Documentação de teste dinâmico**



Figura 2 | Categorias de documentação de testes. Fonte: adaptada de SWEBOK (2024).

Para gerenciar as dependências entre os documentos de requisitos, com foco em processo de testes de sistemas, utilizamos a estrutura com as documentações de testes organizacional, gerenciamento e dinâmico, juntamente com o gerenciamento de configuração de software (GCS).

Por meio do GCS, assegura-se a rastreabilidade entre os requisitos e o monitoramento das mudanças de todos os artefatos relacionados aos testes de um sistema, em todas as suas versões, conforme SWEBOK (2024). Assim, ao realizar as tarefas do processo de testes de sistemas, uma das mais importantes da validação dos requisitos, por meio de execução (teste dinâmico) do sistema, será possível evidenciar o alinhamento estratégico dos requisitos e a satisfação dos usuários.

Compreendemos a estrutura de documentações de testes que estabelecem o que deve ser realizado no processo de testes de requisitos para o gerenciamento dos documentos de

requisitos, desde o planejamento dos testes até o registro dos resultados, para cada requisito de software.

1. Documentação de teste organizacional: inclui as informações necessárias para documentar a política de teste e as estratégias de teste organizacional, incluindo estratégia para atender aos critérios de aceite dos requisitos de software.

O profissional responsável pelos testes, na organização deve determinar uma política para criação de plano de testes, com a alocação de ferramentas que podem auxiliar na padronização de documentação de testes e aumentar a produtividade na elaboração e execução dos testes da organização.

A estratégia de teste deve estar descrita em um plano, adotando uma abordagem com as definições dos objetivos, dos estágios de testes de um projeto, incluindo os tipos de teste a serem executados, as orientações de utilização de recursos internos e externos.

2. A documentação de gerenciamento de teste: inclui o plano de teste, o relatório de status do teste e o relatório de conclusão do teste, vinculado a um ou mais requisitos de software. Tendo como foco a garantia que os testes, serão executados eficientemente, completa, rastreável e controlada, visando a qualidade do produto de software.

Essa documentação deve compor um conjunto estruturado de artefatos gerados e mantidos durante o processo de teste de software. Dentre eles, citamos:

- Plano de teste: com o escopo, abordagem, os recursos e responsáveis, o cronograma das atividades de teste, critérios de entrada/saída e critérios de aceitação do requisito de software.
- Estratégia: detalhamento das técnicas de testes, quando necessário, para cada critério de aceitação do requisito de software e do uso da automação de teste.
- Caso de teste: detalhamento dos cenários de testes, entradas, ações de teste, resultados esperados e critérios de aprovação.
- Mapeamento do requisito: vinculando os requisitos de software atendidos pelo plano de testes, garantindo a rastreabilidade desse artefato e de outros, caso existam, a arquitetura/design do software, documentação técnica de código-fonte, entre outros.

3. Documentação de teste dinâmico: inclui os documentos de registro durante e após a execução dos testes, com as respectivas orientações para cada ação do teste. Lembramos que testes dinâmicos contemplam a execução do sistema em um ambiente controlado, verificando a funcionalidade, desempenho, usabilidade e outros aspectos operacionais esperados pelo requisito de software.

Especificação de teste: arquitetura/design dos itens do teste, especificação de caso de teste e de procedimento de teste.

Dados de teste: requisitos de ambiente de teste, entrada dos testes, resultados esperados, resultado de execução de teste (resultados reais, resultados de teste, log de execução de teste e relatório de incidente).

Dos aspectos não funcionais: os aspectos que interferem na operação do sistema devem ser registrados, tais como: tempo de resposta, utilização de recursos da TI (tecnologia da informação).

É importante lembrarmos que uma documentação de teste deve ficar atualizada progressivamente, seguindo o mesmo nível de qualidade, dando suporte para a análise de melhoria no processo de testes. Ao analisar essa documentação, será possível ter uma visão clara sobre o comportamento do sistema testado e a identificação de defeitos.

## Siga em Frente...

## Técnicas de testes de requisitos

Os testes em sistemas visam entregar um software de qualidade, com foco em validar o que será entregue e verificar como foi implementado, sempre com referência aos requisitos de software. Uma das metas para alcançar a qualidade desejada é encontrar a maior quantidade possível de falhas (e saná-las antes de entregar). Por isso, ao longo de décadas, as técnicas de testes estão sendo melhoradas.

Diante ao cenário, sempre em transição, SWEBOK (2024) propõe três aspectos-chave para os testes: especificação, estrutura e experiência. Com o produto de software – Sistema em Teste (SUT System Under Test) em foco, juntamente com os critérios de aceitação dos requisitos de software, temos o refinamento dos aspectos-chave:

1. Especificação: conhecidas por técnicas de caixa-preta, dependente da execução do sistema, para aceitação dos critérios determinados e preparados com casos de testes em função do comportamento e de entrada/saída do sistema em teste (SUT System Under Test);
2. Estrutura: técnica praticada por desenvolvedores capacitados em testes, conhecido como caixa-branca; para essa técnica, os casos de testes são gerados compreendendo como o SUT foi codificado.
3. Experiência: está relacionada à decisão por utilizar duas ou mais técnicas de testes. Por exemplo, o teste baseado em modelo comprehende-se por utilizar várias técnicas (uma sobreposição) sob o conceito de um modelo que representa a especificação comportamental do SUT.

Podemos considerar várias abordagens de testes, caso tenha o foco em procurar falhas e seguir a documentação de requisitos, principalmente quanto aos critérios de aceitação do requisito. Com essa diretriz, SWEBOK (2024) descreve técnicas de testes comumente usada pelos times de desenvolvimento de software. **Conheceremos algumas:**

- Particionamento de equivalência ou domínio: a técnica é criar dados de entrada em coleções de subconjuntos (ou classes de equivalência) com base em um critério ou relação específica, sendo que cada coleção deve ser tratada da mesma forma pelo sistema. Essa técnica reduz a quantidade de casos de teste, pois cada domínio é representado por ao menos um teste.
- Forçando exceção: o testador deve encontrar exceções e provocar o sistema para encontrar falhas, ou seja, perceber o comportamento do sistema sob condições adversas, quando encontra dados ou cenários anômalos, inesperados ou inválidos. O objetivo é garantir que o sistema funcione corretamente com essas situações, exibindo mensagens de erro, registrando logs e, principalmente, mantendo a integridade do sistema.
- Baseadas na experiência: o sistema no estado da prática são abordagens baseadas em Ad Hoc (para determinada situação), em conhecimento e em ML (Machine Learning). Um dos benefícios nessa técnica é orientar na definição dos objetivos de teste e na geração dos casos de teste; eles são gerados na habilidade e intuição do engenheiro de software, por experiência anterior. É uma opção quando são gerados casos de teste que não são facilmente gerados por técnicas formais; porém, não é viável quando não possui experiência em projeto semelhante. Das técnicas baseadas na experiência, segundo SWEBOK (2024), temos:
  - Teste de pares (Buddy): dois companheiros, um elabora os casos de testes e executa o teste, o outro observa e analisa o processo de teste. Nessa técnica, é comum gerar casos de teste com maior cobertura.
  - Gamificação: visa converter tarefas de teste em componentes de jogabilidade – capacidade do jogador em interagir e controlar o jogo; ao aplicar técnicas que envolvem profissionais ou fazem crowdsourcing (terceirização) de tarefas de testes complexas, a gamificação pode melhorar substancialmente a prática de teste de software.
  - Teste rápido: capacidade de selecionar e executar um conjunto de testes muito pequeno para identificar rapidamente problemas críticos no SUT, para detectar falhas no início do processo de desenvolvimento.
  - Teste de fumaça (ou Build Verification Testing): pela intuição do engenheiro de software, antes dos testes planejados, as principais funcionalidades do SUT são operacionalizadas para se comportar adequadamente. O objetivo é evitar falhas causadas pelo ambiente de teste.
- Baseado em cenários: principal técnica para validar os requisitos de software, com foco em gerar casos de teste baseados nos aspectos comportamentais do SUT, a partir de modelos formais dos requisitos de software do SUT. Na execução do teste, os resultados são avaliados em comparação aos resultados esperados. Pela complexidade das técnicas, as baseadas em cenários são frequentemente suportadas em automação de teste. Utilizando o fluxo de trabalho, da especificação do requisito de software, o processo de negócios é testado usando a técnica baseada em cenários, cobrindo o fluxo principal e os secundários do processo.

Ao final dos testes, as documentações devem ser complementadas com o registro de resultados, incluindo os defeitos e as variações necessárias, quando existir, do plano inicial.

Juntamente com a lista de defeitos podem ser descritos detalhes que auxiliarão o desenvolvedor no ajuste e correção do código-fonte.

Estudante, concluindo esse tema, você se sentirá capacitado a desenvolver os planos de testes com foco em qualidade de software, seguindo os requisitos de software. Porém, recomendamos continuar pesquisando e estudando a importância dos testes de requisitos de software.

Bons estudos!

## Vamos Exercitar?

À medida que o projeto de desenvolvimento de software chega próximo à entrega, os gestores da CaféSaúde ficam satisfeitos com os resultados, porém a expectativa está na fase de teste de requisitos, ou seja, em colocar à prova o produto de software criado a partir das especificações de requisitos de software. Para tanto, os engenheiros de software contam com várias técnicas de testes, incluindo as abordagens para garantir a entrega adequada dos sistemas de colheita e secagem de café.

No detalhamento dos processos de testes, os usuários ficarão seguros ao saber que estão sendo aplicadas técnicas de testes eficientes que garantirão as operações sem problemas em suas responsabilidades.

Nos módulos em que a temperatura do grão do café é informada pelo operador, poderá estar parametrizado para receber apenas entre 18 e 25 graus, evitando efetuar o input de dados fora do contexto do negócio gerenciado pela CaféSaúde.

Utilizando a técnica de particionamento de equivalência ou domínio, percebemos um exemplo de dados para os casos de teste; entrando com o valor abaixo de 18 ou acima de 25, o sistema não deve prosseguir. Todavia, informando qualquer valor entre 18 e 25, será considerada uma entrada válida.

Neste exemplo, verifiquemos os casos de testes, baseado nos critérios de aceitação do requisito de software, conforme ilustrado no Quadro 1.

Temperatura	Comportamento	Aceitação
17	Fora do limite	Reprovado
18	No limite	Aprovado
22	Dentro do limite	Aprovado
25	No limite	Aprovado
25,1	Fora do limite	Reprovado

### Quadro 1 | Dados de entrada para o critério de aceitação.

Utilizando a técnica de forçar exceção, percebemos os dados de peso da amostragem para a secagem do grão de café. Foi estabelecido no critério de aceitação, conforme o requisito de software, apenas números positivos para o peso. Analisemos o Quadro 2 com os casos de testes para a técnica de forçar exceção.

Peso	Comportamento	Aceitação
abc	Caractere inválido	Reprovado
-14	Valor Negativo	Reprovado
8%	Número e caractere especial	Reprovado
25	Dentro do esperado	Aprovado

### Quadro 1 | Dados de entrada para o critério de aceitação.

Em todos os casos de testes submetido aos testes de requisitos devem haver as instruções para auxiliar os desenvolvedores na correção do sistema.

Os testes dinâmicos são fundamentais e mais importantes por executar o sistema em suas funcionalidades, conforme especificado nos requisitos e gerenciado pelos planos de testes. Portanto, esses testes devem ser documentados para auxiliar no monitoramento do progresso do teste, fornecendo um registro histórico das atividades realizadas. Por fim, será útil para avaliar a qualidade do software.

Ela complementa os testes estáticos (como revisões e análises) ao capturar o comportamento real do sistema em execução.

Lembramos que o uso de checklist (listas de verificação), integrado ao sistema de gestão do projeto, elaborado para orientar o roteiro do planejamento na execução de testes, auxiliará nas atividades de testes de requisitos.

Estudante, pelo exemplo vivenciado nesse caso da CaféSaúde, você perceberá a cada novo conhecimento que a sua participação em projetos de softwares é fundamental. Atualize-se, cada vez mais, com pesquisas e experiências em gerenciamento de documentos de requisitos e de testes.

Bons estudos!

## Saiba mais

I. **Saiba mais sobre: gestão de dependências e alterações.** Leia as páginas 442 e 443.

## Engenharia de software

II. **Saiba mais sobre: 5.1.4. Documentação de testes.** Leia as páginas 136 a 137.

### Guide to the Software Engineering Body of Knowledge v4.0

III. **Saiba mais sobre: testes baseados em cenários.** Leia a página 129.

WASHIZAKI, Hironori. Guide to the Software Engineering Body of Knowledge v4.0

IV. **Saiba mais sobre: relacionamento de documentos.** Leia a página 193.

### Guide to the Software Engineering Body of Knowledge v4.0

## Referências

IIBA - International Institute of Business Analysis. **Padrão Essencial Global do IIBA para Análise de Negócios.** Toronto, Canadá, 2015. Disponível em: [https://bit.ly/anasis\\_bacorestd](https://bit.ly/anasis_bacorestd) Acesso em: 28 set. 2024.

KERR, E. S. (org.). **Gerenciamento de requisitos.** São Paulo: Pearson, 2015. *E-book*. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

MASCHIETTO, L. G. et al. **Desenvolvimento de software com metodologias ágeis.** Porto Alegre: SAGAH, 2021. *E-book*. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556901824/> Acesso em: 9 dez. 2024. [Minha Biblioteca]

PAULA FILHO, W. de P. **Engenharia de Software - Projetos e Processos - V. 2. 4. ed.** Rio de Janeiro: LTC, 2019. *E-book*. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788521636748/> Acesso em: 7 dez. 2024. [Minha Biblioteca]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software.** 9. ed. Porto Alegre: AMGH, 2021. *E-book*. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos.** Porto Alegre: SAGAH, 2020. *E-book*. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

SBROCCO, J. H. T. de C.; MACEDO, P. C. de. **Metodologias ágeis - engenharia de software sob medida.** Rio de Janeiro: Érica, 2012. *E-book*. Disponível em:

[https://integrada\[minhabiblioteca.com.br\]/reader/books/9788536519418/](https://integrada[minhabiblioteca.com.br]/reader/books/9788536519418/) Acesso em: 1 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos: software orientado ao negócio.** 1. ed. Rio de Janeiro: Brasport, 2016. *E-book*. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

WASHIZAKI, H. **Guide to the Software Engineering Body of Knowledge v4.0.** Waseda University: 2024. Disponível em: [https://bit.ly/engreq\\_swebokv4](https://bit.ly/engreq_swebokv4) Acesso em: 15 nov. 2024.

## Aula 5

Encerramento da Unidade

### Videoaula de Encerramento

#### Este conteúdo é um vídeo!



Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

#### Dica para você

Aproveite o acesso para baixar os slides do vídeo, isso pode deixar sua aprendizagem ainda mais completa.

Olá, estudante! Nesta videoaula, você irá conhecer as atividades de verificação, validação e o gerenciamento da documentação de requisitos.

Esse conteúdo é importante para a sua prática profissional, pois são conhecimentos fundamentais para entregar sistemas com qualidade baseados em requisitos de softwares verificados, validados, gerenciados e testados.

Prepare-se para esta jornada de conhecimento! Vamos lá!!!

### Ponto de Chegada

Olá, estudante! São competências desta unidade:

- Compreender a verificação de requisitos.
- Compreender a validação de requisitos.

- Compreender a estrutura da documentação de requisitos.
- Explicar os testes de requisitos.

Você compreenderá e reconhecerá os mecanismos fundamentais que competem ao engenheiro de softwares conhecer e aprimorar para entregar sistemas com qualidade, baseados em requisitos de softwares gerenciados, validados, verificados e testados, sempre alinhados aos objetivos de negócio.

## É Hora de Praticar!



### Este conteúdo é um vídeo!

Para assistir este conteúdo é necessário que você acesse o AVA pelo computador ou pelo aplicativo. Você pode baixar os vídeos direto no aplicativo para assistir mesmo sem conexão à internet.

Com o desenvolvimento de software alcançando o estado de pronto, os gestores da CaféSaúde estão animados em saber como o time de desenvolvimento está atuando em verificação, validação e documentação dos requisitos de softwares. É, também, a fase de testar o produto de software com base nos requisitos, uma etapa crucial que garantirá a melhoria contínua dos sistemas.

Portanto, vamos explanar como ocorre a verificação, a validação e o gerenciamento dos documentos de requisitos.

Você está apto a desempenhar as atividades de verificação e validação de requisitos de software?

Você consegue explicar as vantagens da documentação de requisitos de software?

Quais cuidados devemos tomar no planejamento dos testes de requisitos de software?

Ao depararmo-nos com a realização da verificação e validação de requisitos, necessitamos das especificações e demais artefatos que compõem os requisitos de software. Porém, reunir a documentação completa dos requisitos de software não é apenas acessar o repositório do conteúdo do projeto de software da CaféSaúde.

Percebemos como um requisito de software pode ser representado parcial ou totalmente por documentações, pelas quais pode-se aumentar a garantia da veracidade do conteúdo e a rastreabilidade.

Compreendemos a importância de um IC (Item de Configuração) para o desenvolvimento de software:

- Pensemos nas atividades em que o analista de sistemas realiza no início do projeto, com os operadores de equipamentos da colheita do café, supervisores das equipes de

colhedores de café, entre outros, pelas quais geram (a) documentos registrando a reunião ou documentos anexados.

Ao longo das atividades de elicitações, podem surgir (b) outros documentos que serão anexados aos requisitos de negócio, evidenciando o alinhamento com a estratégia organizacional. Essas elicitações são especificações que estabelecem o (c) descritivo definitivo sobre o requisito de software, contando com possíveis (d) diagramas, fluxogramas ou modelagens.

Assim, pelo ciclo de vida do desenvolvimento, os primeiros (e) designs do sistema são elaborados para representar a arquitetura do software, (f) diagramas da análise dos dados são fundamentais para a criação do banco de dados. Ao criar os (g) casos de testes e scripts, artefatos que acompanham a construção do (h) código-fonte do sistema, também são IC do projeto.

Sucessivamente, novos documentos ou artefatos farão parte do projeto e os anteriores podem ser atualizados com uma nova versão com alterações; consequentemente, todos eles são considerados item de configuração na gestão de configuração de software (GCS).

- Um IC (Item de configuração de software) é um artefato identificado e gerenciável projetado para ser gerenciado como uma única entidade, mediante uma política de gestão de configuração de software (GCS).

Sendo o IC uma entidade de software, será controlado pela GCS, juntamente com os códigos-fonte, gerados sob a especificação de requisitos de software.

Entendamos o GCS, segundo SWEBOK® GUIDE V4.0 (2024), como uma ferramenta que controla vários IC. Citemos alguns dos mais comuns em um projeto de software: planos, documentação de requisitos, documentação de design, materiais de testes, ferramentas de software, código-fonte e executável, dados e dicionários de dados e documentação para instalação, manutenção, operações e uso de software.

Fato prático para o GCS, segundo Pressman (2021), o conhecimento de ferramentas de GCS é imprescindível para conseguir gerenciar os documentos de requisitos de software. É a ferramenta de GCS que possibilita a implementação da matriz de rastreabilidade, sendo possível aprofundar os processos baseados em documentos e, ainda, tornam-se gerenciáveis as políticas e estratégicas que regem as atividades de todo o ciclo de vida do software.

Com os artefatos gerenciados por uma ferramenta de GCS, o time de desenvolvimento de software estará confiante para desempenhar as atividades de verificação e validação de requisitos de software.

Acompanhemos algumas abordagens da verificação e validação de requisitos: uma linguagem clara, adequada à realidade dos engenheiros de software e demais envolvidos no projeto de software; com o detalhamento adequado.

As documentações relacionadas aos requisitos de softwares devem ter características que satisfaçam às necessidades de todos os envolvidos no desenvolvimento do software. Segundo SWEBOK® GUIDE V4.0 (2024), assim podem ser construídas:

- Linguagem adequada ao contexto do negócio e das pessoas envolvidas, utilizar uma linguagem simples e clara, evitando ambiguidade.
- Detalhamento adequado à compreensão dos stakeholders (pelo time de tecnologia da informação, pelo time operacional e demais envolvidos), detalhes suficientes para que

todos possam verificar e validar, pelo conteúdo e pela documentação.

- Sobretudo estar com a consistência necessária para atender às necessidades, resolvendo os problemas e/ou implementando inovações desejadas pelos gestores de negócios.

Entendamos algumas das principais atividades e aspectos que podem aumentar a possibilidade de serem realizadas com sucesso pelos engenheiros de software ou o responsável direto pela entrega. Para Pressman (2021), os testes são cruciais na obtenção de um produto de software com qualidade e alinhado aos objetivos dos negócios.

Portanto, ao especificar os requisitos de software, a clareza e determinação dos critérios de aceitação do requisito serão fundamentais para compreensão de todos os envolvidos na construção do sistema. Pois, através desses requisitos serão gerados componentes da tecnologia da informação capazes de executar as funcionalidades ou realizar comportamentos satisfatórios aos objetivos do projeto de software.

Compreendamos como a documentação gerenciada adequadamente favorecerá na análise de projeto e construção de software minimizando riscos de falhas ou retrabalho:

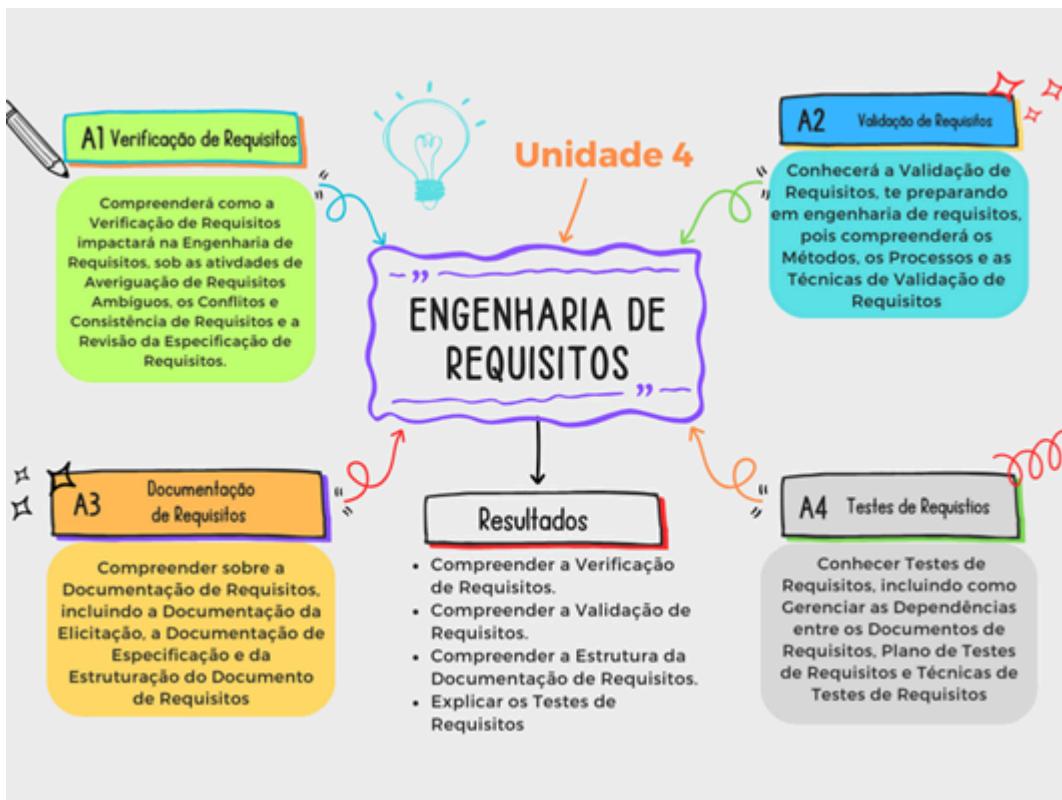
Sob a visão de impactos, as pessoas se concentrarão em objetivos estabelecidos pelas especificações de requisitos, envolvendo-se na verificação e validação constante dos artefatos que estão criando, alterando ou atualizando. Ou seja, empregarão estratégias e comportamentos que possam minimizar impactos negativos de suas ações. Por exemplo, descrever os casos de testes e gerar dados de testes condizentes com a realidade, guiados pelos critérios de aceitação, tornarão os testes de requisitos eficientes e adequados para atender às expectativas do projeto.

Observemos: quando um desenvolvedor minimiza o impacto de seu trabalho sobre o de outros, ele também está diminuindo o trabalho necessário para minimizar o impacto de seu próprio trabalho no deles. Se tornará uma cultura organizacional no desenvolvimento de software de qualidade utilizando a engenharia.

Estudante, ao concluir este conteúdo, você estará capacitado a desempenhar suas atividades de testes de requisitos com maior segurança e compreensão das necessidades de um projeto de software. Continue aprimorando a sua habilidade em compreensão do gerenciamento de requisitos, teste de requisitos e validação de requisitos.

Bons estudos!

Acesse o infográfico utilizando o QRCode por qualquer dispositivo para compreender a extensão do conteúdo desta unidade da disciplina de Engenharia de Requisitos.



KERR, E. S. (org.). **Gerenciamento de requisitos**. São Paulo: Pearson, 2015. *E-book*. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**. 9. ed. Porto Alegre: AMGH, 2021. *E-book*. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/> Acesso em: 5 nov. 2024. [Minha Biblioteca]

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020. *E-book*. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/> Acesso em: 4 nov. 2024. [Minha Biblioteca]

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos: software orientado ao negócio**. 1. ed. Rio de Janeiro: Brasport, 2016. *E-book*. Disponível em: <https://plataforma.bvirtual.com.br> Acesso em: 4 nov. 2024. [Biblioteca Virtual 3.0]

WASHIZAKI, H. **Guide to the Software Engineering Body of Knowledge v4.0**. Waseda University: 2024. Disponível em: [https://bit.ly/engreq\\_swebokv4](https://bit.ly/engreq_swebokv4) Acesso em: 15 nov. 2024.