Aula 02 - Engenharia de Software II

Prof. Me. Deivison S. Takatu

deivison.takatu@fatec.sp.gov.br

Engenharia de Requisitos

- São as descrições das características, funcionalidades e restrições que um sistema deve possuir para atender às necessidades do usuário.
- Representam a ponte entre o cliente e a equipe de desenvolvimento.
- A fase de engenharia de requisitos é crítica para o sucesso de um projeto.

Referência: KOTONYA e SOMMERVILLE (1998).

Engenharia de Requisitos

- **Requisitos Funcionais**: Descrevem o que o sistema deve fazer. São as funcionalidades do sistema, as tarefas que ele precisa executar.
    - Exemplos: "O sistema deve permitir o cadastro de novos usuários." ou "O sistema deve calcular o valor total do pedido."
- **Requisitos Não Funcionais**: Descrevem como o sistema deve se comportar. São as restrições e qualidades do sistema.
    - Exemplos: **Usabilidade**: "O sistema deve ter uma interface intuitiva."; **Desempenho**: "O sistema deve responder a uma requisição de busca em no máximo 3 segundos."

Exemplo de Requisitos Funcionais

- RF1: O sistema deve permitir o cadastro de novos livros.
- RF2: O sistema deve permitir a busca de livros por título, autor e ISBN.
- RF3: O sistema deve permitir o empréstimo e devolução de livros.
- RF4: O sistema deve notificar usuários sobre atrasos na devolução.

Definem o "O QUE" o sistema deve fazer, especificando as funções que os usuários poderão realizar no sistema.

Exemplo de Requisitos Não Funcionais

- RNF1 (Desempenho): O sistema deve responder a consultas de busca em menos de 2 segundos.
- RNF2 (Segurança): O sistema deve autenticar usuários com senha criptografada.
- RNF3 (Usabilidade): A interface do usuário deve ser intuitiva.
- RNF4 (Confiabilidade): O sistema deve estar disponível 99,5% do tempo.

Categorias comuns: Desempenho, Segurança, Usabilidade, Confiabilidade, Escalabilidade, Manutenibilidade, Portabilidade, entre outros.

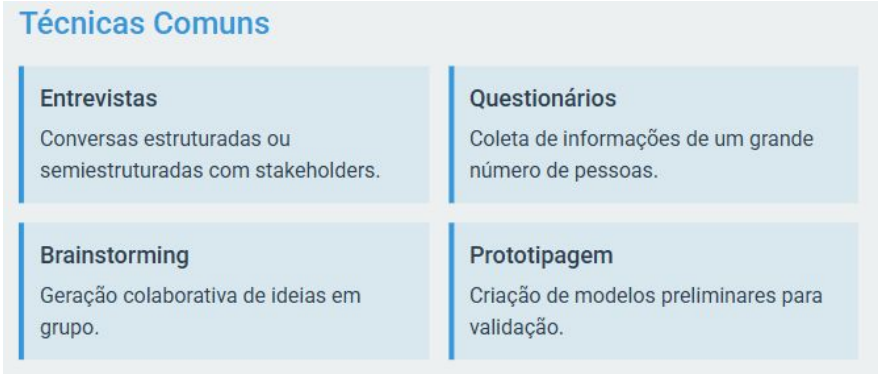
Processo de Engenharia de Requisitos

O processo de Engenharia de Requisitos é iterativo e contínuo, envolvendo comunicação constante com stakeholders para garantir que o software atenda às necessidades reais. As fases principais são:

1. Elicitação Coleta e descoberta de requisitos junto aos stakeholders.
2. Análise Refinamento e organização dos requisitos coletados.
3. Negociação Resolução de conflitos e priorização de requisitos.
4. Especificação Documentação formal dos requisitos acordados.

Fase 1: Elicitação de Requisitos

Processo de descoberta, captura e levantamento de requisitos junto aos stakeholders, buscando entender suas necessidades, expectativas e restrições para o sistema.

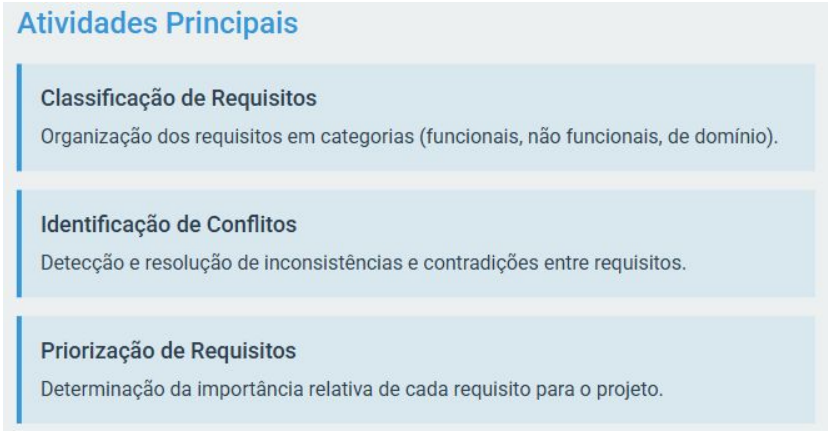


Referência: KOTONYA e SOMMERVILLE (1998).

Fonte: Elaboração própria.

Fase 2: Análise de Requisitos

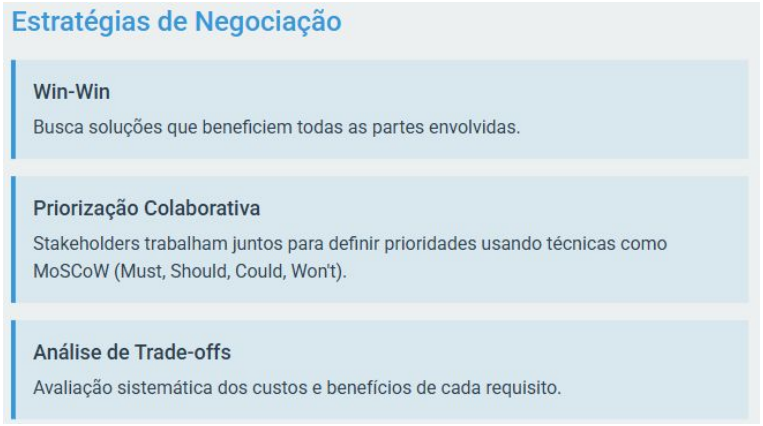
Processo de examinar, refinar e estruturar os requisitos coletados durante a elicitação, garantindo que sejam completos, consistentes e não ambíguos.



Fonte: Elaboração própria.

Fase 3: Negociação de Requisitos

Processo de resolução de conflitos entre requisitos concorrentes, considerando restrições de recursos, tempo e orçamento para chegar a um conjunto viável de requisitos.



Fonte: Elaboração própria.

Fase 4: Especificação de Requisitos

Processo de documentação formal e detalhada dos requisitos acordados, criando uma base para o desenvolvimento, validação e gerenciamento de mudanças.

Fonte: Elaboração própria.

Conceitos de Requisitos de Software

A Necessidade do Cliente: Requisitos traduzem necessidades e expectativas dos usuários e stakeholders em funcionalidades e características do sistema, estabelecendo o que o software deve fazer.

Ponte entre Negócio e Tecnologia: Requisitos conectam objetivos de negócio com soluções técnicas, garantindo que o software desenvolvido agregue valor real à organização e aos usuários.

Requisitos como visão de produto

- Requisitos representam necessidades do usuário traduzidas em funcionalidades do produto.
- Na gestão de produto, é comum descrever os requisitos como histórias de usuário, épicos e features, conectando-os à estratégia de negócio.
- **Exemplo**: “Como cliente, quero salvar meus pedidos favoritos para comprá-los novamente com rapidez.”

Priorização e valor de negócio

- Nem todos os requisitos têm o mesmo peso: Por isso, é papel do Product Owner, Product Manager é definir prioridade.
- Técnicas comuns: *MoSCoW (Must, Should, Could, Won't)*, Matriz Valor x Esforço.
- Requisitos são vistos como itens de backlog que devem gerar valor ao usuário e vantagem competitiva.

Comunicação e especificação ágil

- Na gestão de produto, requisitos não precisam ser documentos extensos, mas sim artefatos vivos. O foco é manter uma visão compartilhada entre stakeholders.

Fonte: Somosadd.

Requisitos e User Stories

Requisito Funcional (RF): O sistema deve exibir um gráfico de evolução patrimonial.

User Story: Como investidor, quero visualizar um gráfico da evolução patrimonial, para acompanhar o crescimento do meu dinheiro ao longo do tempo.

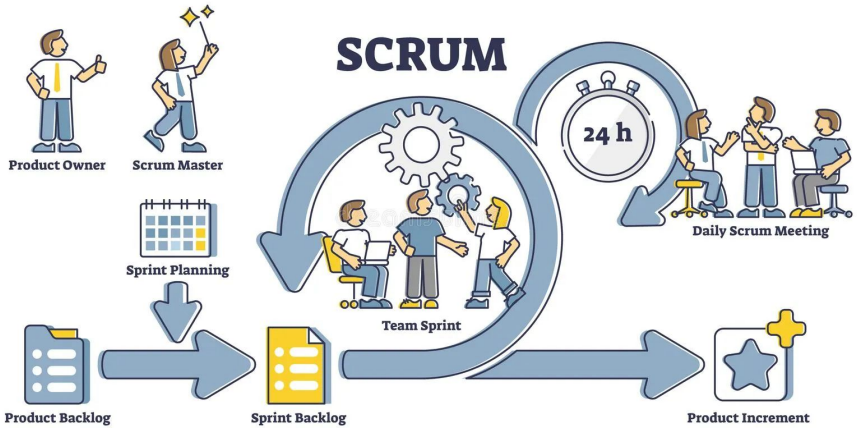
Requisitos e User Stories

O Product Backlog reúne User Stories, organizadas e priorizadas conforme valor de negócio e esforço. Isso permite dividir funcionalidades em atividades menores e garantir entregas incrementais de maior impacto.

Com Sprints curtas, o time entrega pequenos incrementos, revisa com stakeholders e adapta o planejamento a partir do feedback. Essa prática ágil aumenta a colaboração, reduz riscos e mantém o produto alinhado às necessidades reais dos usuários.

Requisitos em Agilidade e Scrum

No Scrum, os requisitos são descritos como user stories no backlog, refinados continuamente com feedback. O Product Owner assegura alinhamento com a visão, e cada sprint entrega valor com transparência e adaptação.



Fonte: dio.

Estudo de Caso - Lovable

Lovable.dev é uma plataforma baseada em inteligência artificial que permite a criação de aplicações web completas sem a necessidade de codificação. Ao descrever sua ideia em linguagem natural, a IA gera automaticamente o frontend, backend, autenticação, banco de dados e integrações necessárias.

Fonte: lovable.dev/

Atividade

Nesta atividade, cada grupo deverá elaborar um backlog inicial com 10 requisitos utilizando o projeto da matéria Engenharia de Software I. Os requisitos devem ser adaptados utilizando a técnica de User Story. Cada requisito deverá ser classificado utilizando a técnica de priorização MoSCoW. Documentem esse backlog no Trello, criando um quadro específico para o projeto e registrando cada requisito como um cartão.

Referências

1. PRESSMAN, Roger S.; MAXIM, Bruce R. Engenharia de Software: uma abordagem profissional. 8. ed. McGraw-Hill, 2016.
2. SOMMERVILLE, Ian. Engenharia de Software. 10. ed. Pearson, 2019.
3. LARMAN, Craig. Utilizando UML e Padrões: Uma introdução à análise e ao projeto orientados a objetos. 3. ed. Bookman, 2007.
4. PFLEEGER, Shari Lawrence; ATLEE, Joanne M. Engenharia de Software: teoria e prática. 4. ed. Pearson, 2010.
5. JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James. UML: Guia do Usuário. 2. ed. Bookman, 2000.
6. KOTONYA, Gerald; SOMMERVILLE, Ian. Engenharia de Requisitos. 1. ed. LTC, 1998.
7. IEEE Computer Society. Guide to the Software Engineering Body of Knowledge (SWEBOK V3.0). IEEE, 2014.
8. ISO/IEC/IEEE. 29148:2018 – Systems and software engineering — Life cycle processes — Requirements engineering. ISO/IEC/IEEE, 2018.
9. Object Management Group (OMG). Unified Modeling Language (UML) Specification. Versão 2.5.1, 2017.

Aula 02 - Engenharia de Software II

Prof. Me. Deivison S. Takatu

deivison.takatu@fatec.sp.gov.br