

Requisitos de *Software*

Prof.^a Ma. Jessica Oliveira

Aula 07 – 05/05/2025

Gerência de requisitos e controle de mudanças.

Fundamentos e importância.

- A gerência de requisitos é um conjunto de atividades fundamentais que garante que os requisitos de um sistema sejam documentados, controlados, modificados e rastreados ao longo de todo o ciclo de vida do *software*.
- Segundo Pressman e Maxim (2016), o objetivo da gerência de requisitos é assegurar que os requisitos do sistema reflitam, de forma contínua e precisa, as necessidades reais dos *stakeholders*, mesmo diante de mudanças naturais ao longo do projeto.

Fundamentos e importância.

- Na prática, os requisitos não são elementos estáticos. Conforme Sommerville (2019), eles tendem a evoluir em função de múltiplos fatores, tais como:
 - Alterações no ambiente de negócio;
 - Novas demandas dos usuários;
 - Descobertas técnicas ao longo do desenvolvimento;
 - Erros de interpretação ou omissões nos requisitos iniciais.

Fundamentos e importância.

- Quando essas mudanças ocorrem sem controle, surgem problemas graves, como requisitos conflitantes, requisitos obsoletos, retrabalho, e até mesmo fracasso do projeto.
- É por isso que a gerência de requisitos se torna uma atividade estratégica, não apenas técnica.
- Ela ajuda a manter o alinhamento contínuo entre o que é desenvolvido e o que realmente é necessário.

Fundamentos e importância.

- De acordo com o Guia BABOK® (IIBA, 2021), a gerência eficaz de requisitos também contribui para:
 - Aumento da rastreabilidade e transparência;
 - Facilidade na comunicação com stakeholders;
 - Maior controle sobre mudanças;
 - Minimização de riscos.
- Portanto, sem gerência de requisitos, há um alto risco de se construir o sistema certo de forma errada, ou até mesmo o sistema errado de forma correta.

Rastreabilidade.

- E definida como a habilidade de identificar e acompanhar cada requisito individualmente, desde sua origem até sua implementação final, testes e manutenção.
- Isso significa que é possível:
 - Saber de onde o requisito surgiu;
 - Ver quem o solicitou;
 - Acompanhar em quais documentos, modelos, códigos e testes o requisito impacta.

Rastreabilidade.

- Sommerville (2019) afirma que a rastreabilidade é um dos pilares da gerência de requisitos, pois ela:
 - Facilita a análise de impacto de mudanças;
 - Garante consistência entre os artefatos de desenvolvimento;
 - Apoia as atividades de validação e verificação;
 - Proporciona transparência entre os envolvidos.

Rastreabilidade.

- Existem dois tipos principais de rastreabilidade:
 - **Rastreabilidade para trás (*backward traceability*):** permite identificar a origem de cada requisito, ou seja, qual necessidade do cliente ou documento deu origem a ele.
 - **Rastreabilidade para frente (*forward traceability*):** permite rastrear um requisito até sua implementação e testes, garantindo que ele foi atendido no produto final.

Rastreabilidade.

- Um exemplo prático seria:
 - **Requisito R5:** "O sistema deve permitir que o paciente cancele uma consulta com 24h de antecedência."
 - **Origem:** Reunião com stakeholders em 10/03/2025.
 - **Artefatos afetados:** Documento de requisitos, Diagrama de Casos de Uso, Classes de controle de agendamento, Casos de teste T15 e T16.
- Esse controle detalhado só é possível quando há um processo organizado de documentação e uso de identificadores únicos para cada requisito.
- Conforme Wiegers e Beatty (2013), ferramentas modernas de gerência de requisitos, como Jama, DOORS, ReqView e Enterprise Architect, oferecem suporte nativo para rastreabilidade entre requisitos e demais artefatos.

Versionamento.

- É o processo de registrar e controlar diferentes versões de um mesmo requisito ao longo do tempo.
- Isso é necessário porque, em um projeto real, os requisitos:
 - São refinados com o tempo;
 - Podem ser corrigidos ou desdobrados;
 - Podem ser reescritos após testes ou feedbacks;
 - Precisam ser atualizados conforme novas regras de negócio.

Versionamento.

- Segundo Pressman e Maxim (2016), o versionamento é uma prática fundamental para manter a história evolutiva do sistema, sendo útil para auditorias, manutenção futura e análise de impacto.
- Cada alteração em um requisito deve gerar uma nova versão numerada, e todas as versões anteriores devem ser arquivadas, para consulta futura.
- Wiegers e Beatty (2013) destacam que uma política de versionamento bem definida reduz falhas de comunicação e aumenta a confiabilidade dos requisitos em projetos complexos.

Versionamento.

- Por exemplo:
 - R2_v1: “O sistema deve permitir o agendamento de consultas.”
 - R2_v2: “O sistema deve permitir o agendamento de consultas, com escolha de data e horário pelo paciente.”
 - R2_v3: “O sistema deve permitir o agendamento de consultas com seleção de médico, data e horário.”
- Se essas versões não forem controladas, os desenvolvedores e analistas podem trabalhar com informações divergentes, gerando **inconsistência, retrabalho e conflito**.

Versionamento.

- Além disso, práticas como controle de histórico, marcação de versões, uso de logs de mudanças e documentação comparativa (diff) são essenciais nesse processo.
- Ferramentas como Git e GitHub, embora voltadas para código-fonte, também permitem versionar documentos de requisitos e manter um histórico claro de alterações.
- Para documentos mais formais, é comum usar controles de versão como:
 - R1.0 – versão inicial;
 - R1.1 – correção de erro;
 - R2.0 – nova funcionalidade incorporada.

Controle de mudanças.

- O controle de mudanças é o processo de analisar, aprovar (ou rejeitar), documentar e implementar modificações nos requisitos ao longo do projeto.
- Como mudanças são inevitáveis, o importante é que elas sejam tratadas com métodos formais e critérios objetivos.
- Conforme o Guia PMBOK® (PMI, 2021), o controle de mudanças é parte integrante da Gestão da Integração e deve incluir:

Controle de mudanças.

- Registro formal da mudança (RFC – *Request for Change*);
 - Avaliação do impacto (em termos de tempo, custo, riscos e qualidade);
 - Decisão de aceite, rejeição ou adiamento;
 - Registro da decisão e atualização da documentação.
- Em projetos de *software*, especialmente os que seguem processos mais formais, é comum a criação de um *Change Control Board* (CCB), um comitê responsável por avaliar e aprovar mudanças nos requisitos. Esse comitê é composto por analistas, gerentes de projeto, desenvolvedores, representantes do cliente, entre outros.

Controle de mudanças.

- Segundo Sommerville (2019), o controle de mudanças deve responder às seguintes perguntas:
 - Qual é a natureza da mudança?
 - Quem solicitou a mudança?
 - Quais são os requisitos afetados?
 - Quais são os impactos esperados no sistema e no cronograma?
 - A mudança será implementada agora, depois ou será rejeitada?

Controle de mudanças.

- Wiegers e Beatty (2013) recomendam que toda solicitação de mudança seja tratada como um artefato formal, com campos como:
 - ID da mudança;
 - Descrição detalhada;
 - Requisitos afetados;
 - Decisão tomada;
 - Justificativa;
 - Nome do aprovador.
- Essa prática garante **transparência, rastreabilidade e previsibilidade**, permitindo uma evolução saudável do produto, sem comprometer a integridade do projeto.

Vamos para um desafio?

Dúvidas?

jessica.oliveira@p.ucb.br

Referências.

- IIBA. Guia BABOK®: **Guia para o Corpo de Conhecimento em Análise de Negócios**. 3. ed. Toronto: International Institute of Business Analysis, 2021.
- PMI. **Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos (Guia PMBOK®)**. 7. ed. Newtown Square: Project Management Institute, 2021.
- PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de Software**. 8. ed. São Paulo: McGraw-Hill, 2016.
- SOMMERVILLE, Ian. **Engenharia de Software**. 10. ed. São Paulo: Pearson, 2019.
- WIEGERS, K.; BEATTY, J. **Software Requirements**. 3. ed. Redmond: Microsoft Press, 2013.