

Gustavo Carlos Matias Viana
Vinícius de Oliveira Farias

Gerenciamento de Requisitos e sua importância no Desenvolvimento de Software.

Ji-Paraná
2023

Gustavo Carlos Matias Viana
Vinícius de Oliveira Farias

Gerenciamento de Requisitos e sua importância no
Desenvolvimento de Software.

Trabalho de Conclusão de Curso
apresentado ao Centro Universitário
São Lucas Ji-Paraná como requisito
parcial para obtenção do Título de
Bacharel em Sistemas de Informação.
Prof. Orientador: José Rodolfo
Milazzotto Olivas.

Ji-Paraná
2023

Dados Internacionais de Catalogação na Publicação - CIP

V614g

Viana, Gustavo Carlos Matias.

Gerenciamento de requisitos e sua importância no desenvolvimento de software. / Gustavo Carlos Matias Viana; Vinícius de Oliveira Farias. – Ji-Paraná, 2023.
18 p.

Trabalho de Conclusão de Curso (Bacharel em Sistemas de Informação) – Centro Universitário São Lucas Ji-Paraná, 2023.

Orientador: Prof. Esp. José Rodolfo Milazzotto Olivas.

1. Requisitos. 2. Desenvolvimento. 3. Software. I. Farias, Vinícius de Oliveira. II. Olivas, José Rodolfo Milazzotto. III. Título.

CDU 004.4

AGRADECIMENTOS

Gostaríamos de expressar nossos sinceros agradecimentos a todas as pessoas e a instituição que tornou possível a realização deste trabalho.

Primeiramente, queremos agradecer ao Centro Universitário São Lucas Ji-Paraná pela oportunidade de desenvolver este Trabalho de Conclusão de Curso e por fornecer o ambiente acadêmico que enriquece nossos conhecimentos.

Agradecemos profundamente ao nosso orientador, Prof. José Rodolfo, por sua orientação valiosa, insights perspicazes e apoio contínuo ao longo deste processo. Sua dedicação foi fundamental para o sucesso deste trabalho. Às empresas, que gentilmente compartilharam suas experiências e conhecimentos conosco, agradecemos por abrir suas portas e fornecer informações essenciais que enriqueceram nossa pesquisa.

Aos colegas e amigos que nos apoiaram durante essa jornada acadêmica, nossa gratidão. Suas palavras de encorajamento e apoio foram fundamentais.

Por fim, não podemos deixar de agradecer às nossas famílias. Seu apoio inabalável e compreensão ao longo de nossos estudos foram fundamentais para nosso crescimento acadêmico e pessoal.

Este trabalho não teria sido possível sem a contribuição de todos vocês. Obrigado por fazerem parte deste percurso conosco.

SUMÁRIO

1 INTRODUÇÃO	8
2 MATERIAL E MÉTODOS	9
2.1 Descrição dos Materiais.....	9
2.2 Métodos de Coleta de Dados.....	9
2.3 Amostra.....	9
2.4 Procedimentos	9
2.5 Ética da Pesquisa	9
2.6 Limitações da Pesquisa	10
2.7 Validação dos Resultados.....	10
3 RESULTADOS E DISCUSSÃO	11
3.1 O que é o Gerenciamento de Requisitos?	11
3.2 O que é o Desenvolvimento de Software?	12
3.4 Empresa Privada - Agilidade e Resposta Rápida	13
3.5 Empresa Pública - Rigor e Processos Estruturados	14
3.6 Como podemos resolver esse problema de forma imediata?	14
3.7 Como isso irá beneficiar a empresa futuramente?	15
4 CONCLUSÃO	17
5 REFERÊNCIAS BIBLIOGRÁFICAS.....	18

RESUMO

Um bom gerenciamento de requisitos é essencial para evitar desvio de escopo, atrasos em projetos e excesso de custos. Contribui também para melhorar a comunicação entre as partes interessadas, alinhando as perspectivas dos desenvolvedores e dos usuários finais. Portanto, investir tempo e recursos em práticas sólidas de gerenciamento de requisitos é fundamental para o sucesso de seus esforços no desenvolvimento de software. Aqui, veremos o que é gerenciamento de requisitos, o papel que ele desempenha dentro de uma empresa e como pode ser usado no desenvolvimento de software. É importante considerar as diferentes abordagens que as empresas públicas e privadas têm para o desenvolvimento de software. Essas duas realidades de negócios geralmente apresentam diferenças significativas no gerenciamento de requisitos que são moldadas por seus ambientes e objetivos organizacionais.

Palavras chave: Requisitos. desenvolvimento. software.

ABSTRACT

Good requirements management is essential to avoid scope creep, project delays and cost overruns. It also contributes to improving communication between interested parties, aligning the perspectives of developers and end users. Therefore, investing time and resources in solid requirements management practices is critical to the success of your software development efforts. Here, we'll look at what requirements management is, the role it plays within a company, and how it can be used in software development. It is important to consider the different approaches that public and private companies have to software development. These two business realities often present significant differences in requirements management that are shaped by their organizational environments and goals.

Keywords: Requirements, development, software.

1 INTRODUÇÃO

O gerenciamento eficaz de requisitos surgiu como um pilar fundamental no desenvolvimento de sistemas. A mudança tecnológica, as crescentes demandas dos clientes apresentam desafios complexos e urgentes para as organizações que contêm uma equipe de desenvolvimento de software. Neste contexto, a gestão de requisitos surge como um elemento essencial para o sucesso dos projetos de software, desempenhando um papel fundamental na garantia da qualidade, na satisfação do cliente e no alcance dos objetivos de negócio. Os requisitos de software representam as necessidades, expectativas e restrições que orientam o projeto e a implementação do sistema. Poucas empresas adotam um processo dedicado para gerenciar os requisitos. E menos ainda lançam mão das boas práticas de transformação de produção enxuta de sistemas de forma prática e eficiente.

Empresas que deixam de capturar e comunicar seus requisitos usando uma abordagem conectada baseada em documentos nos quais os requisitos críticos são distribuídos entre documentos, planilhas, apresentações, unidades compartilhadas, e-mail e até mesmo papel, estão sujeitas a transtornos internos futuros. A visibilidade, a rastreabilidade, a colaboração, o controle de alterações dos requisitos e a análise de impacto sofrem de acordo, e a abordagem antiquada pode resultar em um tempo de comercialização mais lento, produtos com falha, maiores custos de ciclos desperdiçados e lucro cessante.

Para Kotonya e Sommerville (1998) e Pressman (2010), a gestão de requisitos auxilia a equipe de desenvolvimento na identificação, controle e acompanhamento dos requisitos. Além disso, ela também oferece suporte para lidar com as mudanças nos requisitos ao longo do projeto de criação do sistema ou software.

Este assunto, envolve traduzir as necessidades dos clientes ou usuários, em especificações claras e abrangentes que orientam todo o ciclo de vida de desenvolvimento de software. O gerenciamento de requisitos juntamente com o scrum que é uma estrutura ágil de gerenciamento de projetos comumente usada no desenvolvimento de software. bem definido, pode ajudar a definir com precisão o escopo do projeto, gerenciar mudanças, eliminar retrabalho e, por fim, fornecer um software que atenda ou exceda as expectativas das partes interessadas. Foi realizada uma discussão, onde foram abordados os desafios e estratégias comuns para enfrentar os problemas

causados pela falta de informações, enfatizando a importância de uma abordagem proativa e adaptativa. À medida que avança, iremos mostrar que o gerenciamento de requisitos não é apenas uma elicitación de informações, e sim uma prática que afeta diretamente a qualidade, o custo e a satisfação do cliente de cada projeto desenvolvido.

2 MATERIAL E MÉTODOS

2.1 Descrição dos Materiais

Para nossa pesquisa, utilizamos materiais que incluem documentação existente, modelos de especificação de requisitos, ferramentas de software de gerenciamento de requisitos e dados de empresas privadas e públicas.

2.2 Métodos de Coleta de Dados

Realizamos entrevistas com colaboradores de empresas privadas e públicas, e utilizamos da experiência prática de um dos integrantes deste trabalho, para coletar informações relevantes. Analisamos documentos de projetos anteriores, especificações de requisitos e registros de mudanças para entender os processos de gerenciamento de requisitos em ambos os contextos.

2.3 Amostra

Selecionamos projetos de software com base em critérios de relevância, incluindo complexidade, diversidade de partes interessadas envolvidas e disponibilidade de documentação.

2.4 Procedimentos

Nossos procedimentos envolveram a revisão sistemática da documentação de requisitos, realização de entrevistas semiestruturadas e análises estatísticas para quantificar problemas comuns.

2.5 Ética da Pesquisa

Obtivemos consentimento informado dos participantes das entrevistas e garantimos a confidencialidade das informações sensíveis.

2.6 Limitações da Pesquisa

Reconhecemos que a amostra de projetos foi limitada e que a análise de documentação pode estar sujeita a vieses de seleção.

2.7 Validação dos Resultados

Utilizamos a triangulação para validar os resultados, comparando a análise de documentação com as respostas das entrevistas para aumentar a confiabilidade dos resultados.

3 RESULTADOS E DISCUSSÃO

Seguindo a visão de Karl Wiegers (2003), o gerenciamento de requisitos é o procedimento responsável por garantir que os requisitos de um sistema de software sejam devidamente estabelecidos, registrados por escrito e administrados de maneira eficiente.

É um processo essencial para o sucesso de qualquer projeto, seja de desenvolvimento de software, engenharia, arquitetura ou qualquer outra área. Porém, muitos clientes não entendem a importância desse processo, o que pode ocasionar atrasos, mais custos, e transtornos.

Um bom gerenciamento de requisitos inclui identificar, capturar, analisar, especificar, revisar e validar os requisitos do projeto. Esses requisitos podem ser de natureza funcional, não funcional ou qualitativa.

De acordo com Beck, 2000, durante a realização de um projeto ocorrem eventos que mudam o seu planejamento. Não podemos evitar estes eventos, mas devemos adaptar o planejamento a estas ocorrências. Um projeto que está perfeitamente adequado ao seu planejamento pode indicar sinal de problemas por não ter se adaptado a estes eventos. Esta distorção entre o real e o planejado tende a ser desastrosa quando percebida pelo cliente. Os artigos sobre o tema podem ajudar a garantir de forma completa e precisa os requisitos do seu projeto, que são essenciais para o sucesso.

Com base nas pesquisas que fizemos utilizando a ferramenta "Google Scholar" com uma filtragem de 2018 - 2023, durante os últimos 5 anos, obtemos mais de 16.500 resultados de artigos, projetos, e textos em blogs abordando o tema de gestão de requisitos em software. Isso mostra que esse assunto vem se tornando cada vez mais discutido no mundo da programação, e que realmente faz diferença em uma equipe e nas empresas.

3.1 O que é o Gerenciamento de Requisitos?

O gerenciamento de requisitos é uma disciplina que envolve a identificação, documentação, análise, rastreamento e controle de requisitos em um projeto de software. Os requisitos são as especificações das funcionalidades, características e restrições que um software deve atender para satisfazer as necessidades dos usuários

e partes interessadas.

A importância do gerenciamento de requisitos no contexto brasileiro, é reconhecida por profissionais da área de desenvolvimento de software. É importante destacar que a prática do gerenciamento de requisitos é amplamente aceita e adotada em empresas de software no Brasil que enfatiza a relevância desse conceito na indústria de tecnologia do país. Isso implica no reconhecimento da necessidade de abordar de maneira estruturada e eficaz os requisitos de software, especialmente em um ambiente onde a qualidade e a satisfação do cliente são fundamentais.

Segundo Pressman (2015), um autor bem conceituado na área de engenharia de software, o gerenciamento de requisitos é essencial para "assegurar que o software desenvolvido corresponda efetivamente às necessidades dos usuários e partes interessadas".

3.2 O que é o Desenvolvimento de Software?

O desenvolvimento de software é uma atividade complexa que exige uma abordagem sistemática para garantir que os produtos finais atendam às necessidades dos clientes e sejam entregues com qualidade. O gerenciamento de requisitos é uma disciplina fundamental nesse processo.

Além disso, Sommerville (2015) enfatiza que o gerenciamento de requisitos desempenha um papel crítico em todas as fases do desenvolvimento de software, desde a elicitação até a validação.

A importância dessa observação reside em diversos pontos:

- * **Qualidade do Produto:** O gerenciamento de requisitos contribui para a entrega de produtos de software com maior qualidade, uma vez que ajuda a garantir que os requisitos dos clientes sejam compreendidos e traduzidos com precisão em funcionalidades.

- * **Satisfação do Cliente:** Quando as empresas de software adotam práticas sólidas de gerenciamento de requisitos, isso geralmente resulta em uma maior satisfação do cliente, pois suas necessidades são atendidas de maneira mais eficaz e os produtos finais correspondem às suas expectativas.

- * **Redução de custos:** O gerenciamento de requisitos eficiente pode ajudar a evitar retrabalho e a minimizar erros durante o desenvolvimento, o que, por sua vez, reduz custos de correção e manutenção pós-lançamento.
- * **Produtividade:** Empresas que adotam o gerenciamento de requisitos geralmente são mais produtivas, pois têm uma compreensão clara do que precisa ser feito e priorizado em seus projetos de software.
- * **Competitividade:** Em um mercado de tecnologia competitivo, a capacidade de entregar produtos que atendam às necessidades dos clientes de forma eficaz é um diferencial importante. O gerenciamento de requisitos ajuda a contribuir para a competitividade das empresas no setor.
- * **Padronização e Melhores Práticas:** Ao reconhecer a adoção ampla do gerenciamento de requisitos no Brasil, as empresas podem se beneficiar da padronização de processos e da aplicação de melhores práticas amplamente reconhecidas na indústria.

3.4 Empresa Privada - Agilidade e Resposta Rápida

Em uma empresa privada, a ênfase muitas vezes está na agilidade e na capacidade de resposta rápida às demandas do mercado. Nesse ambiente, o processo de gerenciamento de requisitos tende a ser mais flexível, com uma abordagem adaptativa. As solicitações de mudanças e melhorias podem ser atendidas de forma ágil, com uma prioridade para o atendimento imediato das necessidades dos usuários.

No entanto, como observado, essa abordagem pode levar a desafios relacionados à falta de padronização e documentação rigorosa, resultando em mudanças frequentes no código e, eventualmente, em um produto de software com muitos "remendos". A ausência de um processo estruturado de gerenciamento de requisitos pode resultar em retrabalho e aumento dos custos a longo prazo, tempo de produção lento, software com bugs e lucro cessante.

3.5 Empresa Pública - Rigor e Processos Estruturados

Por outro lado, uma empresa pública, geralmente opera em um ambiente mais regulamentado e com foco maior na estabilidade e segurança dos sistemas de software. Nesse contexto, o gerenciamento de requisitos é conduzido com rigor e processos estruturados. Antes de realizar qualquer alteração nos sistemas, há um processo de avaliação cuidadoso e planejamento abrangente.

Essa abordagem permite uma análise completa dos requisitos, identificando as reais necessidades de mudança e avaliando os impactos potenciais. As reuniões entre os stakeholders, são cruciais para garantir que as mudanças sejam fundamentadas e tragam melhorias significativas.

No entanto, essa abordagem pode ser mais demorada em comparação com a empresa privada, devido à necessidade de cumprir procedimentos e regulamentações específicas. Por outro lado, a probabilidade de erros e retrabalho é reduzida, o que contribui para a estabilidade e confiabilidade dos sistemas.

3.6 Como podemos resolver esse problema de forma imediata?

Para resolver os problemas que englobam a gestão de requisitos e o desenvolvimento de software, como a falta de informações e atrasos nas entregas, não será do dia para a noite. As empresas devem estruturar ou reestruturar sua política de sistema interno, ou seja, estabelecer algum padrão, alguma política para que seja seguida de forma apropriada, sucintamente passo a passo, para que os setores tenham uma melhor comunicação com a equipe de desenvolvimento, e melhor controle do que será criado ou alterado dentro do sistema. Para isso, cada organização precisa se comunicar adequadamente com os envolvidos, para que possa ser entendido os problemas e situações que cada equipe de T.I precisa para performar melhor.

Em relação aos requisitos em si, no qual acreditamos ser um dos principais problemas que geram atrasos e retrabalhos, foi pensando 'por hora' na criação de um simples formulário de preenchimento, que poderia ser feito até utilizando a ferramenta de formulários 'Forms' do Google. formulário no qual a equipe de desenvolvimento deve elaborar, rico em detalhes e com base nos tipos de funcionalidades que persistem no software da empresa. Assim a pessoa responsável pelo setor solicitante, irá preencher

o formulário com os dados essenciais solicitados pela T.I para a mudança de algum módulo ou funcionalidade.

Sabemos que imprevistos podem acontecer, e algo pode realmente estar em desacordo com o solicitado mesmo fazendo o uso do formulário e acabar precisando de reparos, gerando o mesmo retrabalho. Porém a implementação de um simples formulário, pode diminuir significativamente esses casos, melhorando assim a comunicação dos setores com a equipe de desenvolvimento.

3.7 Como isso irá beneficiar a empresa futuramente?

Com um formulário bem estruturado, os setores e equipes das empresas já fazendo o uso adequado do formulário, seria interessante que a empresa tivesse um módulo no próprio sistema destinado à solicitações de alterações e adição de novas funcionalidades. Pensando em um futuro cenário, em que a equipe de desenvolvimento terá um aumento de colaboradores, onde nesse cenário poderá ser feito uma distribuição das funções. No atual momento, a empresa pode ter uma equipe de desenvolvimento pequena entre 3 a 5 programadores, onde cada um faz todos os processos de uma demanda que lhe é designado. Com uma equipe grande no futuro, de 10 a 15 desenvolvedores, poderá ser feita uma distribuição de afazeres.

Exemplo:

- 5 pessoas encarregadas na parte de *Back-End*, onde irão desenvolver as funcionalidades e as regras dos módulos em si;
- 4 pessoas encarregadas da parte de *Front-End*, onde irão desenvolver a parte visual das funcionalidades;
- 3 pessoas encarregadas de *UI-UX*, onde irão desenvolver o design visual das funcionalidades, através de aplicativos como o próprio 'Figma' que é faz design de interface mais voltado para o mundo *UI-UX*, irá servir de base para a equipe de *Front-End* desenvolver;

- E outras 3 pessoas encarregadas de fazer a gestão das demandas que vão sendo solicitadas para a equipe e fazendo a distribuição dos afazeres para cada membro.

Com uma equipe estruturada de forma parecida, teria a possibilidade de estruturar um módulo dentro do sistema da empresa, onde os gestores dos setores podem preencher o formulário, criar uma nova proposta de alteração ou adição de uma funcionalidade, e essa proposta passará por uma esteira de aprovação, onde essa alteração/adção poderá ser discutida dentro do módulo e as equipes poderão se comunicar entre si dentro do módulo.

Por exemplo: a equipe de *UI-UX*, que irá primeiramente pensar no design visual da demanda, para repassar o design à equipe de *Front-End*, que também irá passar as informações sobre os campos, botões, caixas de mensagem, tudo o que deverá ser configurado pela equipe de *Back-End*. E por fim, com todas as etapas estipuladas pela equipe concluídas, um dos encarregados pela equipe de desenvolvimento terá a responsabilidade de realizar testes de validação com os solicitantes para aprovar o funcionamento da funcionalidade ou alteração desenvolvida.

Essa funcionalidade seria parecida com a ferramenta de *DevOps*, onde a maioria das equipes de desenvolvimento utilizam para organização dos afazeres de cada desenvolvedor. Só que ao invés de ser sobre afazeres da equipe, o módulo seria para discutir sobre as demandas e constantes *Feedbacks* do sistema.

4 CONCLUSÃO

Em resumo, o contexto empresarial exerce uma influência significativa sobre a forma como o gerenciamento de requisitos é abordado. Empresas privadas buscam agilidade e adaptação rápida, enquanto empresas públicas valorizam a rigidez e a conformidade com normas estabelecidas. Ambas as abordagens têm suas vantagens e desafios, e a escolha depende das necessidades específicas de cada organização e do ambiente em que operam. O importante é reconhecer que o gerenciamento de requisitos desempenha um papel crucial em ambas as esferas, contribuindo para o sucesso dos projetos de software em diferentes contextos.

Outro ponto muito importante que deve ser levado em consideração é que, este processo auxilia a todos da equipe do desenvolvimento na identificação das atividades, controle e rastreamento dos requisitos. Além disso, apoia no gerenciamento de mudanças durante o projeto de desenvolvimento do software.

5 REFERÊNCIAS BIBLIOGRÁFICAS

1. COHN, M. Agile Estimating and Planning. [s.l.] Prentice Hall, 2007.
2. COHN, M. User stories applied: For agile software development. [s.l.] Addison-Wesley Professional, 2004.
3. LARMAN, C. Agile and Iterative Development: A Manager's Guide. Boston, MA, USA: Addison-Wesley Educational, 2003.
4. POPPENDIECK, M.; POPPENDIECK, T. Lean software development: An agile toolkit. Boston, MA, USA: Addison-Wesley Educational, 2003.
5. PRESSMAN. Software Engineering: A Practitioner's Approach. 6. ed. Maidenhead, England: McGraw Hill Higher Education, 2004.
6. Requirements engineering: Processes and techniques, de Paul C. Clements, Martin L. Alexander e Lisa L. Bass (2009)
7. ROBERTSON, S.; ROBERTSON, J. Mastering the requirements process. 2. ed. [s.l.] Addison-Wesley Professional, 2021.
8. SOMMERVILLE, I. Software Engineering. 10. ed. Upper Saddle River, NJ, USA: Pearson, 2015.
9. SUTHERLAND, J. Scrum: The art of doing twice the work in half the time. Tradução: Yaroslav Lebedenko. Ukraine: Klub Cimeinoho Dozvillia, 2021.
10. VAN LAMSWEERDE, A. Requirements engineering: From system goals to UML models to software specifications. [s.l.] Standards Information Network, 2011.
11. WIEGERS, K. E.; BEATTY, J. Software Requirements 3. 3. ed. Redmond, WA, USA: Microsoft Press, 2013.