



WBA0883_v1.0

Qualidade de software com Clean Code e técnicas de usabilidade



Qualidade de Software: fundamentos e conformidade de requisitos

Principais padrões de qualidade; requisitos de software; técnicas de levantamento de requisitos.

Bloco 1

Stella Marys Dornelas Lamounier

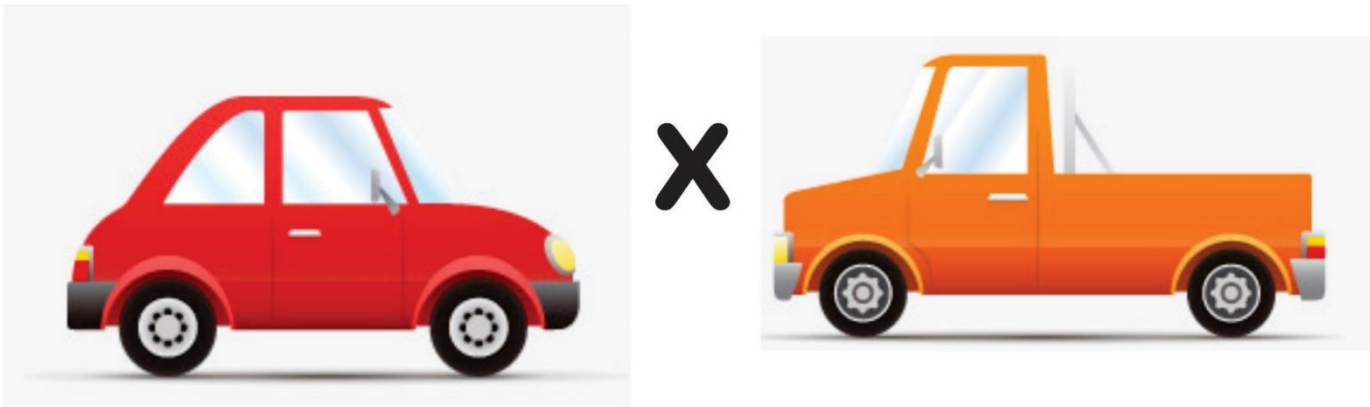


➤ Qualidade

“A qualidade é relativa. O que é qualidade para uma pessoa pode ser falta de qualidade para outra”.

(WEINBERG, 2005, p. 17).

Figura 1 – Qualidade em diferentes tipos de automóveis



Fonte: elaborada pela autora.



História da Qualidade

Figura 2 – Pirâmides do Egito



Fonte: http://www.fambrashalal.com.br/blog_arab/wp-content/uploads/2018/08/EGITO-520x245.jpg.
Acesso em: 29 out. 2021.

Figura 3 – O cúbito



Fonte: Machado (2000, p. 19).



➤ História da Qualidade

Figura 4 – Catedral de Milão



Fonte: <https://www.affaritaliani.it/milano/covid-23-milioni-in-meno-per-il-restauro-del-duomo-di-milano-707878.html>. Acesso em: 29 out. 2021.

Figura 5 – Muralha da China

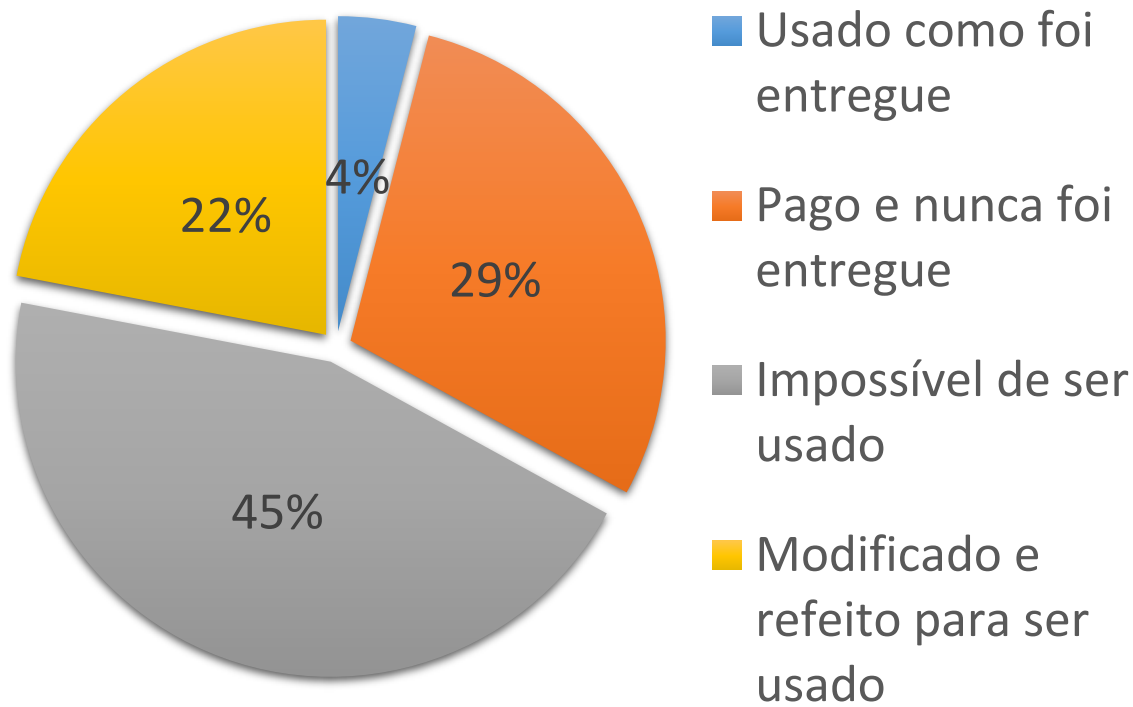


Fonte: <https://media.ceetiz.com/activity/BMIPEK004/thumbnails/642x450/badaling-4.jpg>. Acesso em: 29 out. 2021.



Crise do software

Figura 6 – Crise do Software



Fonte: Santo (2016, [s.p.]).

À medida que os softwares crescem em tamanho e complexidade e **ABSTRAÇÃO** e **COMPLEXIDADE**, conferem cada vez mais **DIFICULDADES** ao processo de desenvolvimento.



Software com erros fatais

Figura 7 – Foguete Ariane 5



Fonte: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRYpsb_MU7tbj7BMx6TqDfhXDQxz2NI4EVnbqtpeAYjE2MJ9bKtfQF4ql7juA_1WBNY_F0&usqp=CAU. Acesso em: 29 out. 2021.

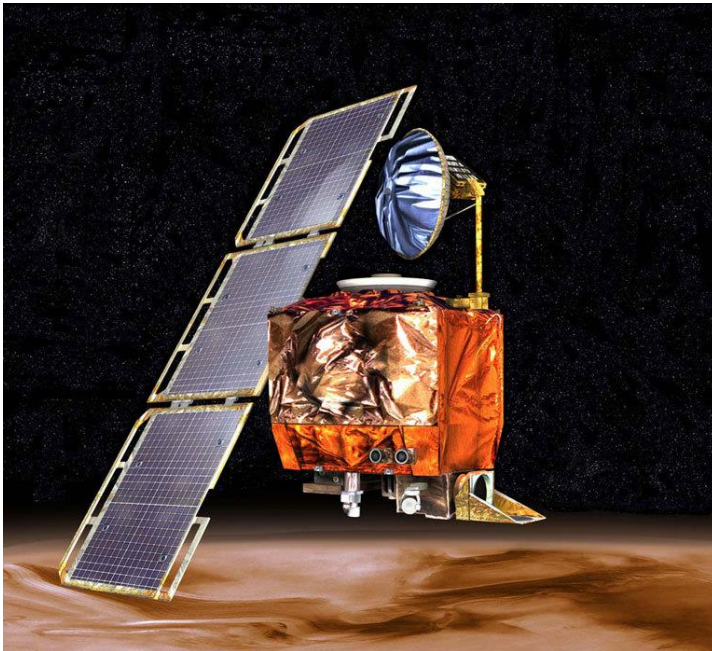
Figura 8 – Therac-25



Fonte: Caballero (2019, [s.p.]).

Software com erros fatais

Figura 9 – *Mars Climate Orbiter*



Fonte: Nasa (2018, [s.p.]).

Figura 10 – Processador Intel
Pentium A8050



Fonte: Arruda (2011, [s.p.]).

➤ Defeito, falha ou erro?

É qualquer imperfeição ou inconsistência no produto do software ou em seu processo.

Erros de cálculos

Falhas ocasionadas por cálculos matemáticos, como divisão por zero, arredondamento, excesso de casas decimais, algoritmos incorretos.

Erros de inicialização

Erros nas inicializações de variáveis.





Defeito, falha ou erro?

Manipulação de dados

Dados tratados ou manipulados erradamente ou não tratados, como datas inválidas ou dados *null*.

Hardware

Falta de tratamento de retorno dos códigos dos periféricos, como verificar se a impressora está ligada; e, ao fazer uma comunicação com outra máquina, verificar se a rede está ativa.

Controle do código fonte

Não controlar adequadamente o código fonte do sistema.

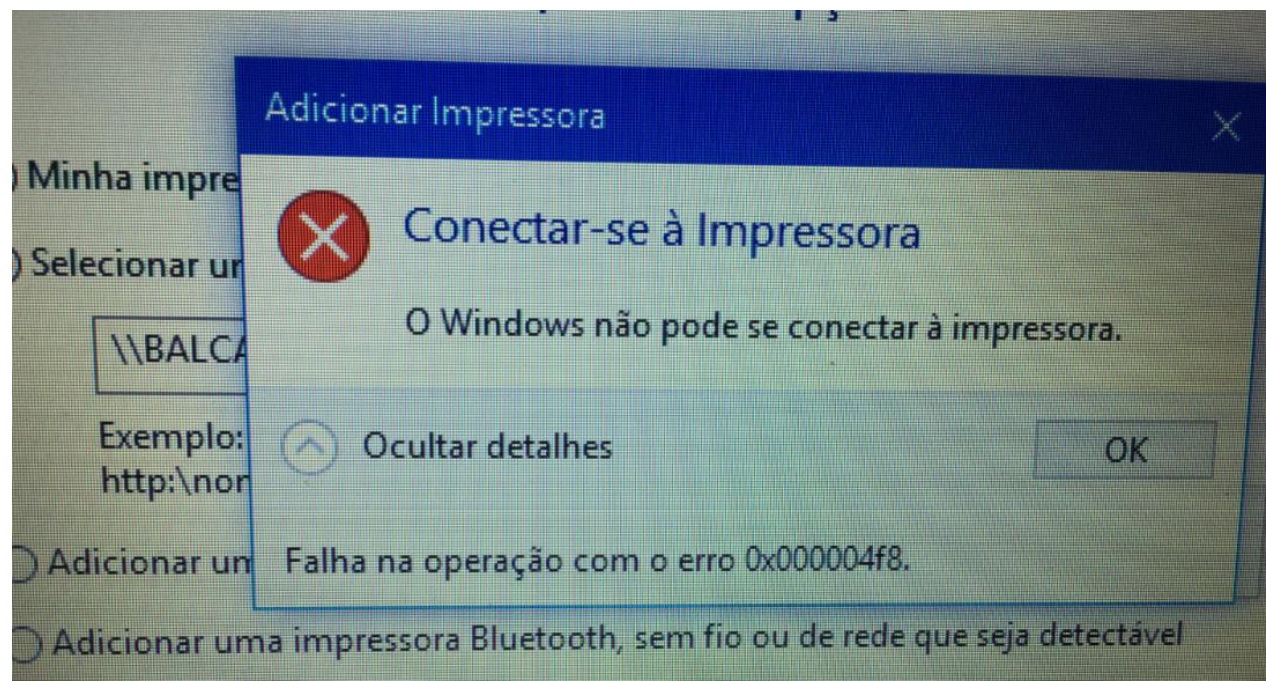


Defeito

Defeito, falha ou erro?

- É retornado ao usuário como comportamento inesperado do software (um aviso de tela).

Figura 11 – Falha de conexão – Impressora



➤ Defeito, falha ou erro?

- O erro é fruto da ação humana, que produz um resultado incorreto, como uma falha na escrita de um código.

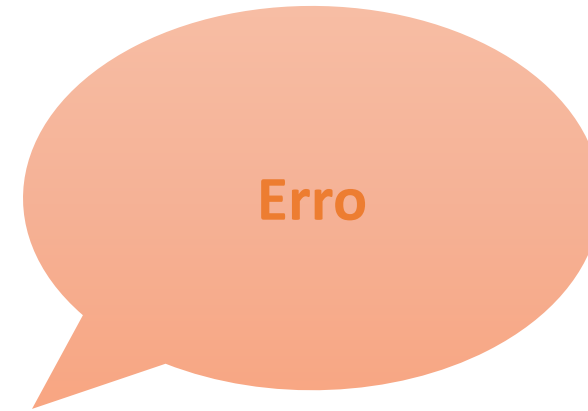


Figura 12 – Erro *versus* Defeito *versus* Falha

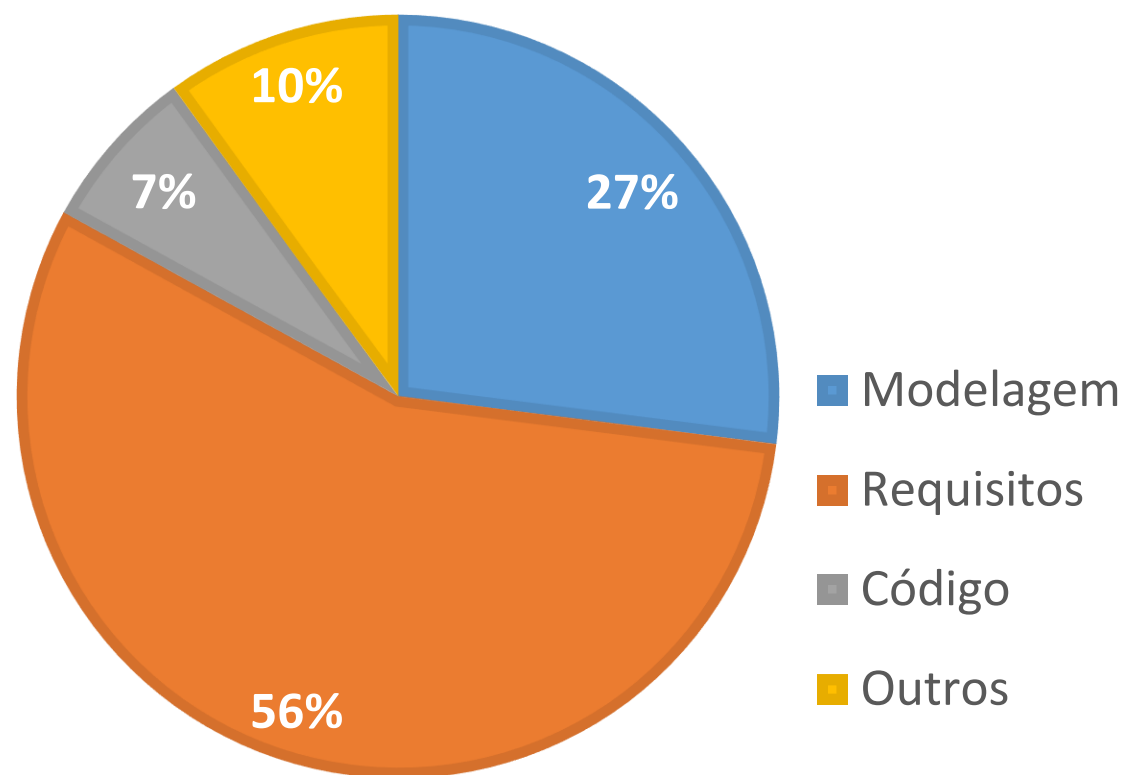


Fonte: elaborada pela autora.



➤ Maiores problemas para a construção de um software

Figura 13 – Problemas com software



Fonte: Bartié (2002, p. 26).

Qualidade de Software: fundamentos e conformidade de requisitos

Principais padrões de qualidade; requisitos de software; técnicas de levantamento de requisitos.

Bloco 2

Stella Marys Dornelas Lamounier

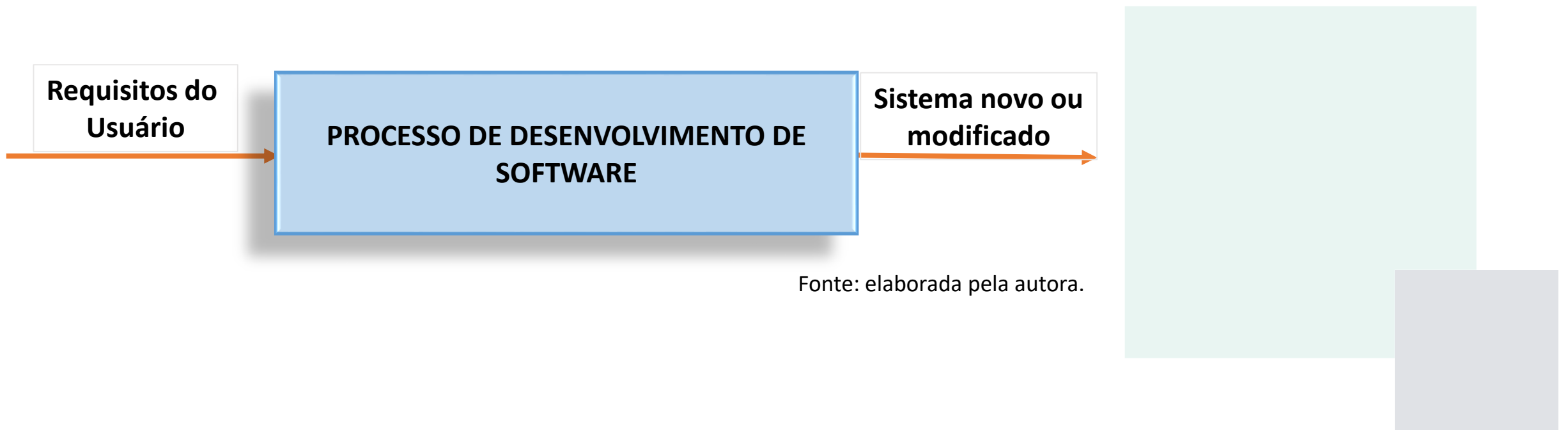


➤ Processos

“Um conjunto de atividades relacionadas entre si, que transforma entradas em saídas, para atingir um determinado propósito.”

(SOFTWARE ENGINEERING INSTITUTE, 2010, p. 449)

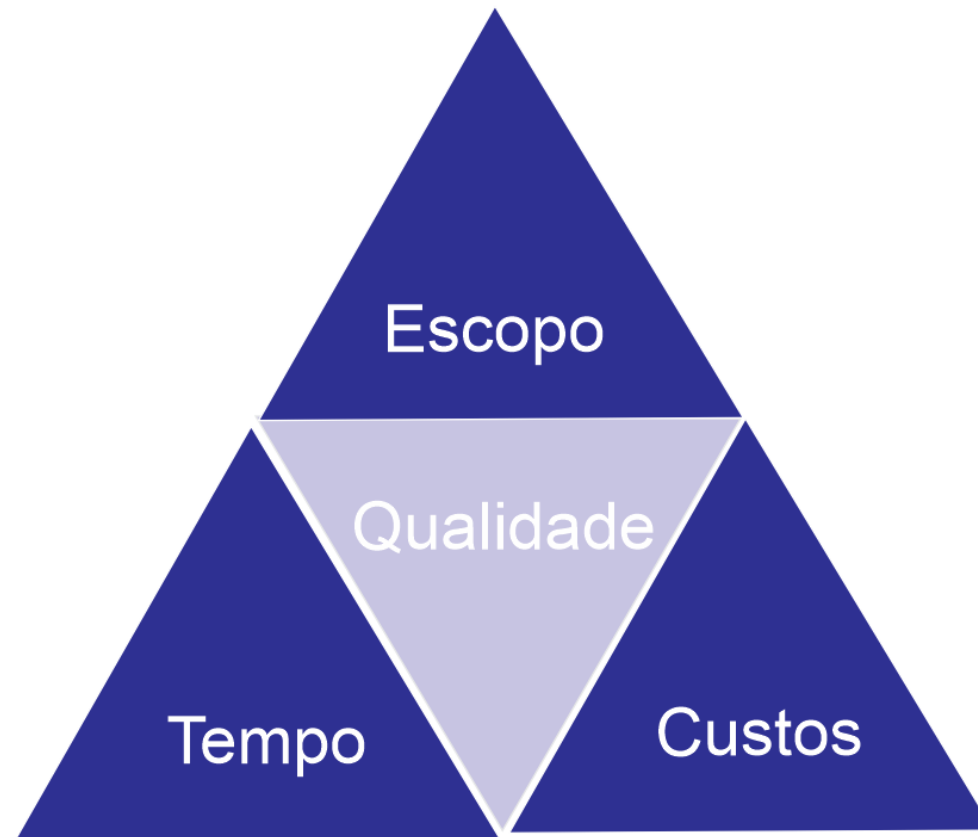
Figura 14 – Processo de Software



Fonte: elaborada pela autora.

➤ Padrões de Qualidade

Figura 15 – Restrição Tripla



Fonte: Montes (2020, [s.p.]).



► Padrões de Qualidade – ISO

- ISO – Significa igual.
- Criada em Genebra, na Suíça, em 1947.
- É uma empresa sem fins lucrativos.
- Presente em 130 países.
- No Brasil, é representada pela ABNT.
- As mais populares: ISO 9000 e ISO 9001.



Padrões de Qualidade – ISO

ISO 12207 – Define o processo de desenvolvimento de software.

ISO 9126 – Qualidade de produto de software.

ISO 12119 – Qualidade de software para pacotes de software.

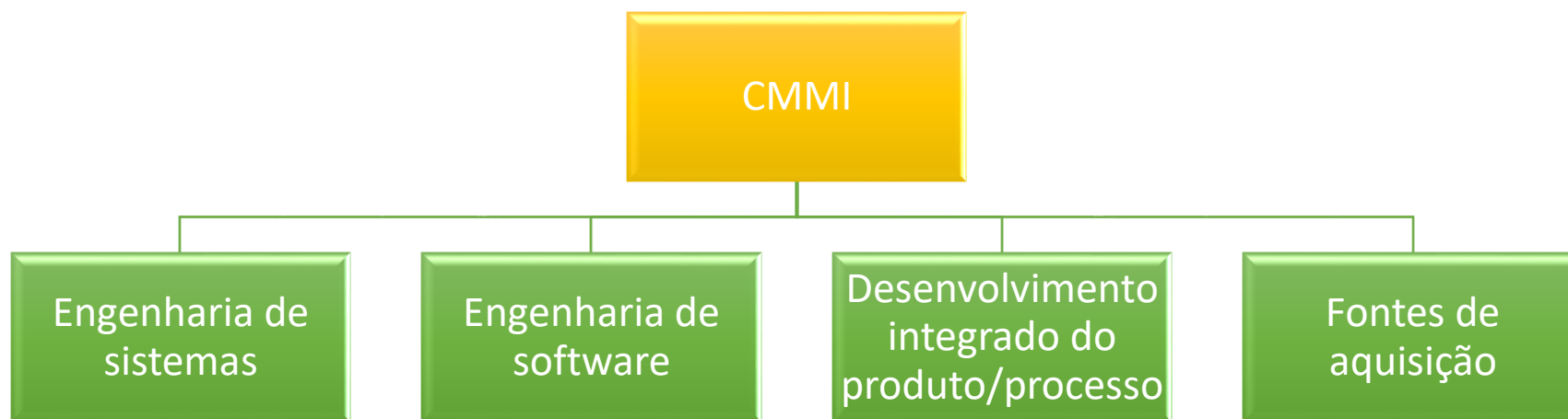
ISO 14598 – Qualidade do produto de software, propondo Atributos de Qualidade.



➤ CMMI (*Capability Maturity Model Integration* ou Modelo Integrado de Maturidade em Capacitação)

- Guia para melhoria de processos nas organizações.
- Gerenciar profissionais no desenvolvimento/aquisição e na manutenção de produtos/serviços.

Figura 16 – Disciplinas CMMI



Fonte: Koscianski e Soares (2007, p. 103).

Figura 17 – Restrição Tripla



Fonte: adaptada de Oliveira, Petrini e Pereira (2015, p. 47-48).

➤ MPS.BR – Melhoria de Processos do Software Brasileiro

- Criado em 2003.
- SOFTEX, COPPE/UFRJ, CESAR, CenPRA e CELEPAR.
- Foco principal: micro, pequenas e médias empresas de software brasileiras que detêm pouco recurso para melhoria de software brasileiro.
- Segue as principais abordagens internacionais pra definição, avaliação e melhoria dos processos de software.
- Mais **barato** (estima-se 40% do valor em comparação ao CMMI).

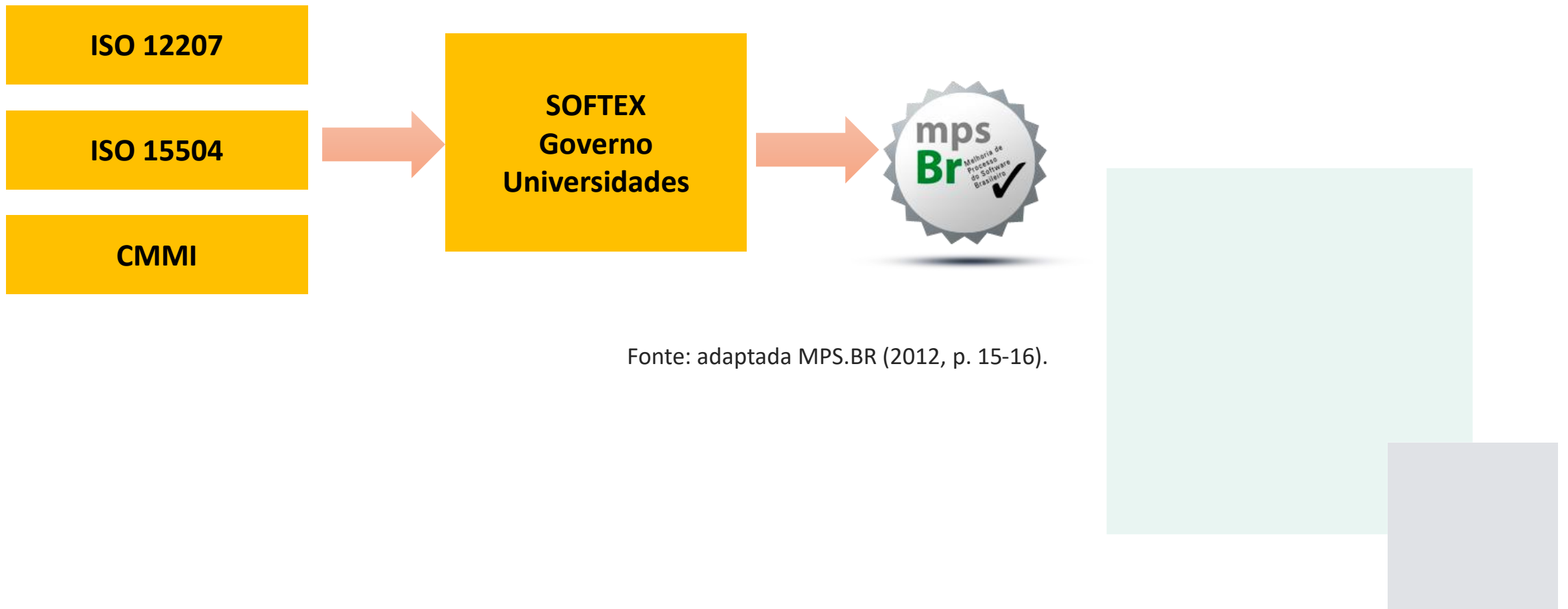
Figura 18 – Logo



Fonte: <http://www.rerum.com.br/rerum/?p=4030>.
Acesso em: 29 out. 2021.

➤ MPS.BR – Melhoria de Processos do Software Brasileiro

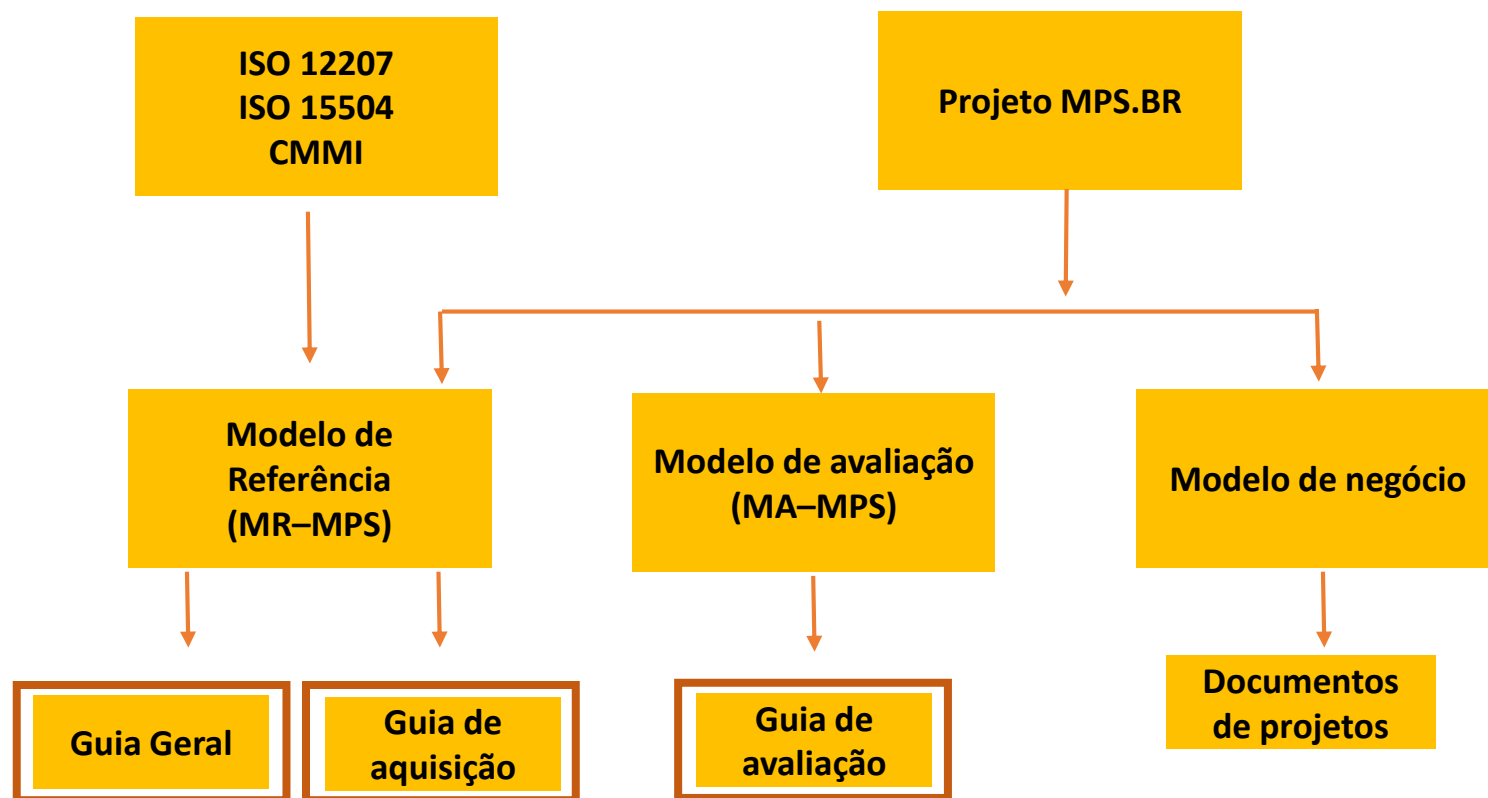
Figura 19 – MPS.BR



Fonte: adaptada MPS.BR (2012, p. 15-16).

Estrutura MPS.BR

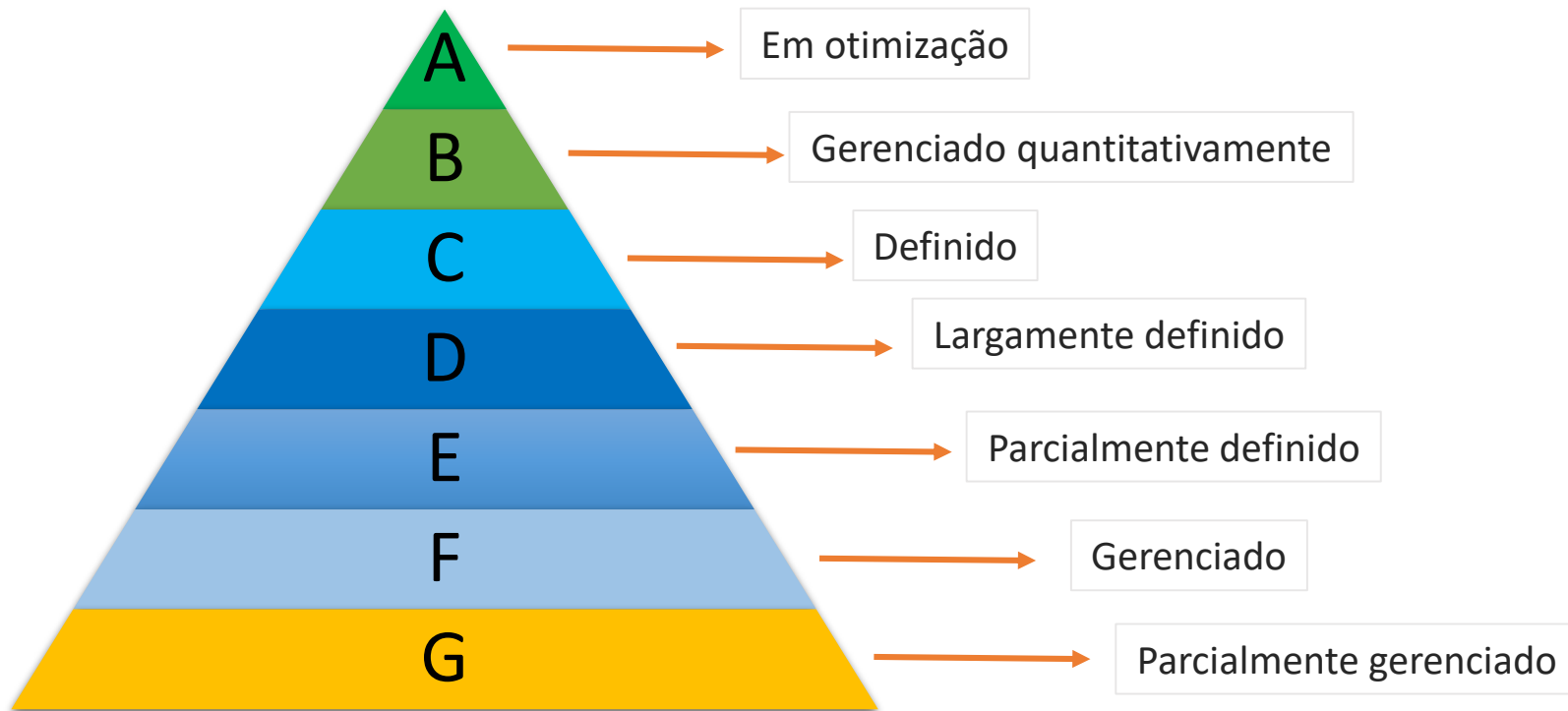
Figura 20 – Componentes do MPS



Fonte: adaptada de MPS.BR (2012, p. 14).

➤ MPS.BR – Modelo de referência MR–MPS

Figura 21 – Níveis de maturidade MPS.BR



Fonte: adaptada de MPS.BR (2012, p. 26-48).

Figura 18 – Logo



Fonte: <http://www.rerum.com.br/rerum/?p=4030>.
Acesso em: 29 out. 2021.

Qualidade de Software: fundamentos e conformidade de requisitos

Principais padrões de qualidade; requisitos de software; técnicas de levantamento de requisitos.

Bloco 3

Stella Marys Dornelas Lamounier



➤ Requisitos

- Ações e funcionalidades do sistema.
- Listagem de restrições, ou seja, o que o sistema não deve conter.
- Processos do sistema.
- Necessidade imposta pelo cliente.

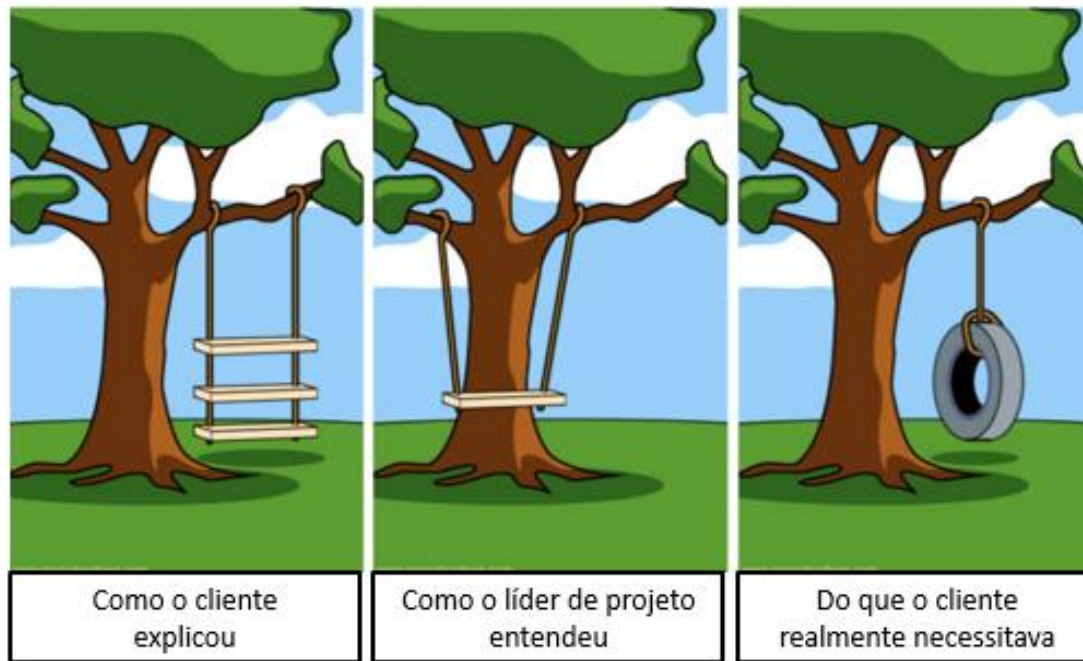
Tipos

- Requisitos funcionais.
- Requisitos não funcionais.



➤ Requisitos

Figura 22 – Problemas de comunicação para produção de software



Fonte: adaptada de Koscianski (2007, p. 173).



Requisitos funcionais

- Define uma função de um sistema de software ou seu componente. Funções que o sistema deve fornecer.
- Comportamento do sistema em uma determinada ação.
- O que o sistema não deve fazer.





Requisitos funcionais

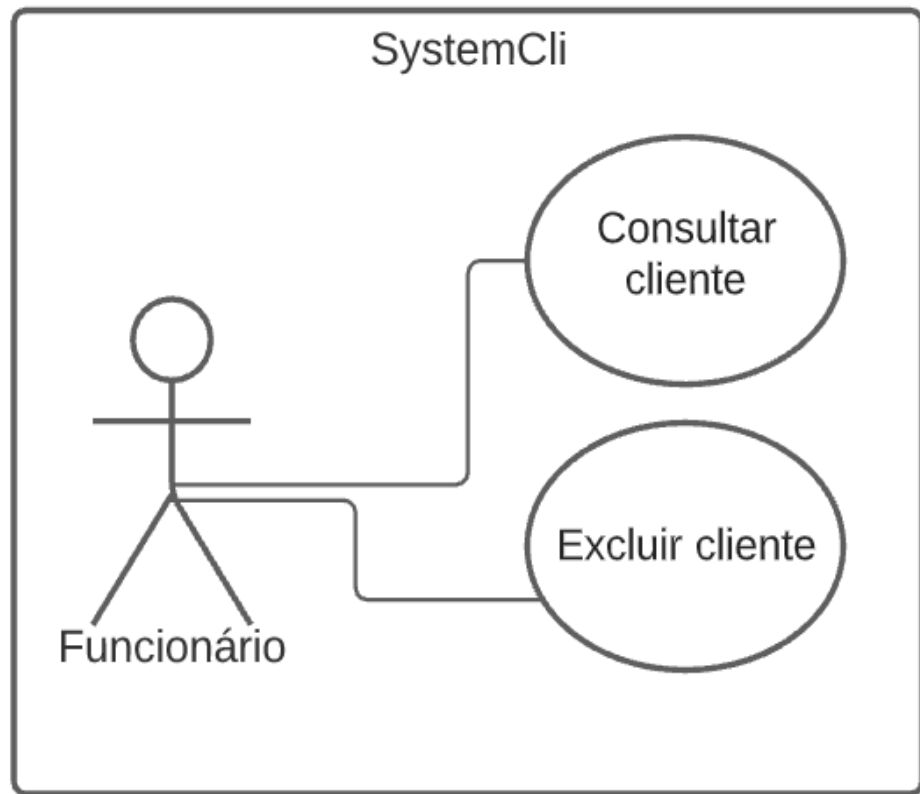
- **RF001** – O sistema deve permitir o cadastro dos fornecedores da loja.
- **RF002** – O sistema deve fornecer telas apropriadas para o usuário ler documentos disponíveis no repositório de documentos.
- **RF003** – Incluir cliente como pessoa física.
- **RF004** – Consultar cliente.
- **RF005** – Realizar pagamento por débito e crédito.
- **RF006** – Consultar estoque.
- **RF007** – Excluir cliente.





Requisitos funcionais

Figura 23 – Cenário SysteCLI



Fonte: elaborada pela autora.

➤ Requisitos não funcionais

- Qualidades específicas que o software deve ter.
- Definem características e impõem limites do sistema, como método de desenvolvimento, tempo, espaço, Sistema Operacional etc.



➤ Requisitos não funcionais

- **RNF001** – O tempo de resposta do sistema não deve ultrapassar 30 segundos.
- **RNF002** – Compatibilidade com sistemas operacionais Windows e Linux.
- **RNF003** – O sistema deve ser desenvolvido em C++.
- **RNF004** – O website deverá ser compatível com os navegadores Mozilla Firefox, Internet Explorer e Microsoft Edge, Chrome.
- **RNF005** – O backup será realizado em ambiente físico e em nuvem.



Documentação de requisitos

Figura 24 – Documentação de RF

| | | |
|--|-----------------------|-----------------------------|
| Sistema: SISTEMA VendasMilK2000 | | |
| Autor: | Data de criação: | Data da última atualização: |
| Detalhamento de requisitos | | |
| RF 001 – Registro de vendas | | |
| Módulo: Vendas | Prioridade: Essencial | |
| Descrição: Todos os produtos vendidos diariamente deverão ser registrados na memória da impressora fiscal. | | |
| Usuário: Funcionário responsável pelas vendas na empresa. | | |
| Informações de entrada: Todos os produtos aptos para a venda deverão ser registrados no sistema via leitura de código de barras. Cada produto deverá ter um código único, que possibilite sua leitura sem ambiguidade. Produtos que por ventura necessitarem de peso deverão ser pesados na presença do cliente para então serem registrados no sistema. | | |
| Informações de saída: Ao final da compra, é impresso um cupom fiscal com todos os produtos lidos pelo leitor de código de barras e suas especificações detalhadas. | | |
| Restrição: É proibido o registro de produtos sem código barras. | | |

Fonte: elaborada pela autora.

➤ Técnicas de levantamento de requisitos

- Entrevistas.
- *Benchmarking*.
- Observações.
- *Brainstorming*.
- Questionários e pesquisas.
- Análise dos documentos.



Teoria em Prática

Bloco 4

Stella Marys Dornelas Lamounier

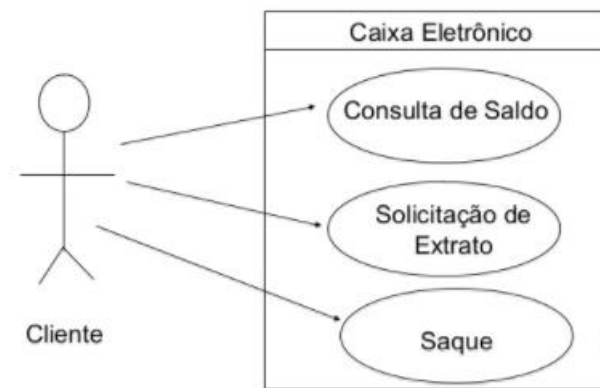


Reflita sobre a seguinte situação

Imagine uma situação fictícia para um cenário de software. Determinada empresa de desenvolvimento foi contratada para a confecção de um sistema para caixa eletrônico. A Figura 1 ilustra o cenário de transações que o cliente poderá realizar no sistema eletrônico (consultar saldo e solicitar extratos e saque em dinheiro).

Na figura do analista, você deverá criar um documento de levantamento de requisitos com os principais requisitos funcionais e não funcionais de acordo com o diagrama apresentado. Lembre-se de especificar de forma organizada cada um dos requisitos RF para funcionais e RNF para não funcionais. Não se limite a quantidades; quanto mais específico e objetivo for, mais qualidade o software apresentará. Lembre-se de abordar requisitos dotados de segurança, tempo de processamento, confiabilidade, facilidade de uso etc.

Figura 25 – Diagrama de Caso de Uso



Fonte: elaborada pela autora.



Norte para a resolução

- Levantar requisitos talvez não seja uma tarefa muito fácil, ainda mais quando se trata da criação de sistemas. Nesse sentido, descrever suas funções e especificações pode ser bem desafiador, tanto do ponto de vista do usuário quanto para quem vai desenvolver o sistema.
- Muitas vezes é difícil lidar com o desconhecido, uma vez que são informações vindas de diferentes fontes e incompletas na grande maioria das vezes. O cliente imagina o que quer, mas não tem o conhecimento para transmitir as informações necessárias; ao mesmo tempo, o desenvolvedor também pode não ter o conhecimento necessário para criar o sistema solicitado.





Norte para a resolução

- Para sanar os problemas vindos de todos os lados, torna-se imprescindível utilizar técnicas seguras de levantamento de requisitos. Nesse sentido, a equipe que está trabalhando com o projeto deverá saber distinguir os requisitos funcionais e não funcionais que abordam a criação do software. Além disso, deverá realizar uma documentação clara e objetiva, a fim de auxiliar o programador no desenvolvimento de um sistemas dotado de muita qualidade, satisfazendo assim as necessidades do cliente.
- Lembre-se de que os requisitos funcionais descrevem as funcionalidades do sistema, como as entradas e saídas desejadas, e o requisitos não funcionais são responsáveis pela qualidade específica do software, por suas características e por seus limites.



Dica do(a) Professor(a)

Bloco 5

Stella Marys Dornelas Lamounier





Ferramenta JIRA

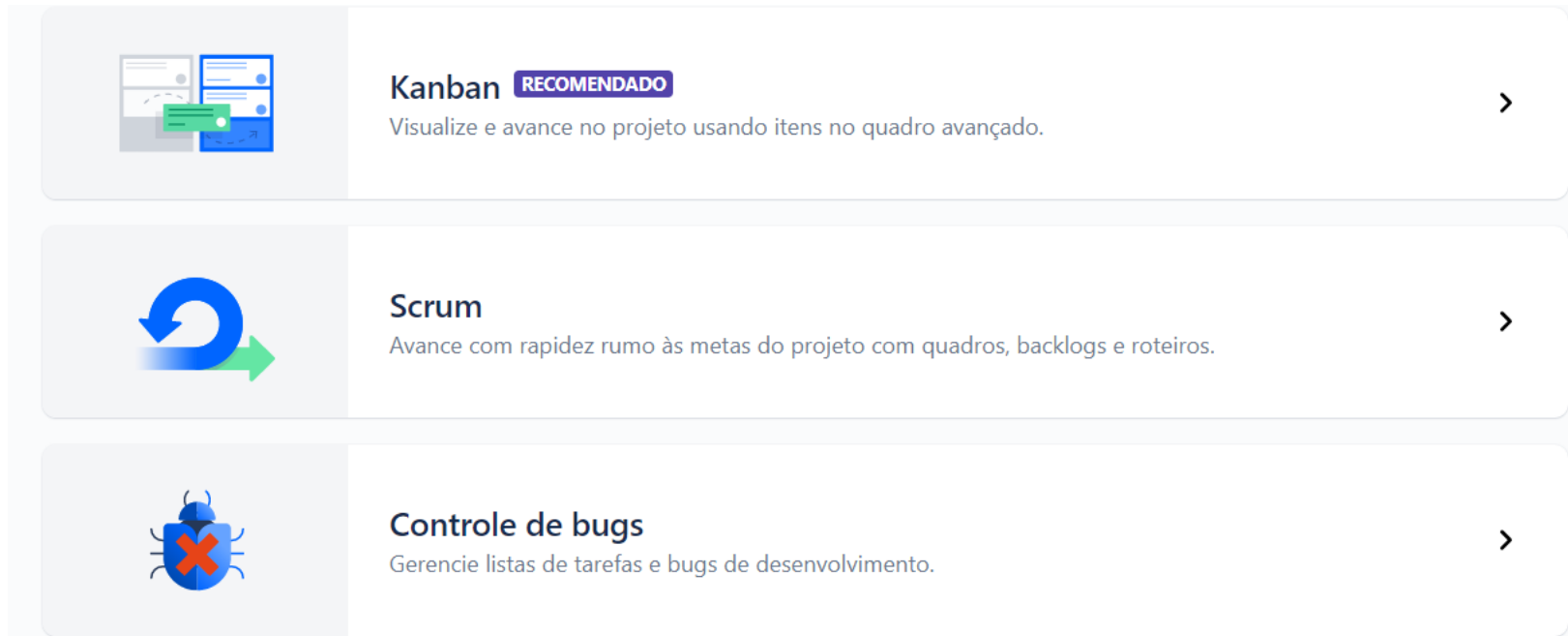
- A Ferramenta JIRA é umas das ferramentas de gestão de projetos mais utilizadas do mundo.
- É muito utilizada para o gerenciamento de requisitos em diferentes metodologias, principalmente ágeis.
- Você pode organizar e priorizar requisitos, preparar requisitos para validação etc.





Ferramenta JIRA

Figura 26 – Área de trabalho JIRA




Fonte: captura de tela da ferramenta JIRA.








Ferramenta JIRA


Figura 27 – Área de trabalho JIRA


 **Projeto1**
Projeto de software


 Roteiro


 **Backlog**

 Painel

 Código


 Páginas de projeto


 Adicionar item

 Configurações do proj...

Projetos / Projeto1

Backlog









Epic ▾

▼ Backlog (2 issues)

000

Criar sprint

| | | |
|---|---------------------|---|
|  EDD2019-1 | Cadastrar aluno |  |
|  EDD2019-2 | Cadastrar Professor |  |

+ Criar item

Fonte: captura de tela da ferramenta JIRA.



Referências

ARRUDA, Felipe. **A história dos processadores**. TecMundo, 2011 Disponível em: <https://www.tecmundo.com.br/historia/2157-a-historia-dos-processadores.htm>. Acesso em: 29 out. 2021.

BARTIÉ, Alexandre. **Garantia da qualidade de software**. [s.l.]: Gulf Professional Publishing, 2002.

CABALLERO, Carlos. **Software Architecture: Therac-25 the killer radiation machine**. 2019. Disponível em: <https://www.carloscaballero.io/software-architecture-therac-25/>. Acesso em: 29 out. 2021.

KOSCIANSKI, A.; SOARES, M. S. **Qualidade de Software**: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2. ed. São Paulo: Novatec, 2007.

MACHADO, N. J. **Medindo Comprimentos**. São Paulo: Scipione, 2000.

MPS.BR. **Guia Geral MPS de Software**. 2012. Disponível em: https://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_Geral_Software_2012-c-ISBN-1.pdf. Acesso em: 29 out. 2021.

NASA. Mars Climate Orbiter. 2018. Disponível em: <https://solarsystem.nasa.gov/missions/mars-climate-orbiter/in-depth/>. Acesso 09 nov. 2021.





Referências

MONTES, Eduardo. PMO Escritório de Projetos: Restrição Tripla. Disponível em: <https://escritoriodeprojetos.com.br/restricao-tripla> . Acesso: 17 mai 2021.

NAVARRO, Roberto. **Quantas pessoas foram necessárias para fazer a Muralha da China?** 2018. Disponível em: <https://super.abril.com.br/mundo-estranho/quantas-pessoas-foram-necessarias-para-fazer-a-muralha-da-china/>. Acesso em: 29 out. 2021.

OLIVEIRA, A.; PETRINI, M.; PEREIRA, D. L. Avaliação da adoção do CMMI considerando o custo de qualidade de software. **Gestão e Projetos: GeP**, [s.l.], v. 6, n. 1, p. 45-62, 2015.

SANTO, C. R. E. **Engenharia de software II**: Modelos de Ciclo de Vida e Processos de Software. 2016. Disponível em: <https://docplayer.com.br/10875057-Engenharia-de-software-ii-modelos-de-ciclo-de-vida-e-processos-de-software-aula-2.html>. Acesso em: 29 out. 2021.

SIQUEIRA, Ethevaldo. **Ariane 5 lança super satélites**. Estadão, 2012. Disponível em: <https://economia.estadao.com.br/blogs/ethevaldo-siqueira/ariane-5-lanca-super-satelites/>. Acesso em: 29 out. 2021.

SOFTWARE ENGINEERING INSTITUTE. **CMMI® for Development, Version 1.3**. 2010. Disponível em: <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9661>. Acesso em: 9 nov. 2021.

WEINBERG, G .M. **Software com Qualidade**. São Paulo: M.Books, 2005.



Bons estudos!

