

## **APLICAÇÕES DE CONCEITOS DE QUALIDADE DE SOFTWARE EM EMPRESAS DO NOROESTE PAULISTA**

### *APPLICATIONS OF SOFTWARE QUALITY CONCEPTS IN COMPANIES IN NORTHWEST PAULISTA*

**Bruno H. S. Francisco<sup>1</sup>, Gabriela H. D. Mendes<sup>2</sup>, Fabiana P. M. Caravieri<sup>3</sup>**

<sup>1</sup>Faculdade de Tecnologia Prof. José Camargo – Fatec Jales, bruno.francisco@fatec.sp.gov.br

<sup>2</sup>Faculdade de Tecnologia Prof. José Camargo – Fatec Jales, gabriela.mendes7@fatec.s.gov.br

<sup>3</sup>Faculdade de Tecnologia Prof. José Camargo – Fatec Jales, fabiana.caravieri@fatec.sp.gov.br

#### **Informação e Comunicação**

#### **Subárea: Banco de Dados, Engenharia e Desenvolvimento de Software**

#### **RESUMO**

A partir da década de 1990, muitas empresas de software perceberam que gastavam bilhões de dólares com softwares que não entregavam as funcionalidades prometidas e com qualidade abaixo do esperado. Com o crescimento do mercado tecnológico e a demanda por sistemas que automatizem tarefas, a qualidade de *software* se tornou um fator indispensável para o processo de desenvolvimento. Nesse sentido, este trabalho propõe a realização de um estudo sobre os conceitos de qualidade de softwares, principais modelos e normas de controle de qualidade, assim como uma pesquisa de campo em *softwarehouses* do noroeste paulista. O objetivo desse trabalho visa identificar essas empresas pesquisadas que utilizam modelos de qualidade de *software*, como a norma ISO-9126-1, modelos CMMI e MPS.BR. Desse modo, pode-se identificar a utilização de conceitos e modelos de qualidade de *software* e suas aplicações durante o processo de desenvolvimento, bem como os papéis e responsáveis por realizarem esse controle de qualidade, além de identificar e analisar o processo de qualidade nas empresas pesquisadas. De acordo com os resultados obtidos através da pesquisa de campo, pode-se constatar que o conceito de qualidade nas empresas analisadas se aplica apenas na fase de execução de testes, não existindo, assim, qualidade de ponta a ponta em seu cenário atual. Palavras-chave: qualidade; software; processo de desenvolvimento; gerenciamento da qualidade.

#### **ABSTRACT**

*Since the 1990s, many software companies realized that they were spending billions of dollars on software that did not deliver the promised functionality and with lower quality than expected. With the growth of the technological market and the demand for systems that automate tasks, software quality has become an indispensable factor for the development process. In this sense, this paper proposes to carry out a study on the concepts of software quality, main models and standards of quality control, as well as field research in software houses in the northwest of São Paulo. The aim of this paper is to identify those surveyed companies that use software quality models, such as ISO-9126-1, CMMI and MPS.BR models. In this way, it is possible to identify the use of software quality concepts and models and their applications during the development process, as well as the roles and responsible for carrying out this quality control, in addition to identifying and analysing the quality process in researched companies. According to the results obtained through field research, it can be seen that the concept of quality in the analyzed companies applies only in the test execution phase, thus not existing end-to-end quality in its current scenario.*

*Keywords: quality; software; development process; quality management.*

## 1 INTRODUÇÃO

A partir da década de 90, muitas empresas começaram a perceber que gastavam bilhões de dólares anuais em softwares que não entregavam as características e funcionalidades que prometiam.

Na década de 1990, as principais empresas reconheciam que, por ano, bilhões de dólares eram desperdiçados em softwares que não entregavam as características e funcionalidades que prometiam. Conforme a tecnologia foi se desenvolvendo ao longo dos anos, as empresas se tornaram cada vez mais dependentes de sistemas automatizados (PRESSMAN, 2011).

Entretanto, esses sistemas apresentavam falhas recorrentes ocasionadas pela falta de qualidade no seu desenvolvimento, como exemplo, pode-se citar o foguete “Ariane 5”.

Uma falha no software ocasionou a perda total do foguete e prejuízos financeiros. Assim como o foguete “Ariane”, pode-se citar o caso da máquina de radioterapia “Therac-25” na década de 80. Novamente, uma falha de software gerou danos irreparáveis após essas máquinas enviarem mais radiação que o necessário.

À medida em que a demanda por sistemas complexos aumenta, a qualidade desponta como um fator essencial no desenvolvimento de software. Entretanto, uma das dificuldades encontradas na implantação de um programa de qualidade está em entender o que, de fato, significa qualidade de software (LENILDO, 2010).

Nesse contexto, este trabalho tem como objetivo realizar a abordagem de alguns conceitos sobre a qualidade de software por meio de um estudo de caso realizado em empresas de tecnologia presentes na região do noroeste paulista. Além de apresentar dados sobre a aplicação de metodologias que englobam qualidade no PDS (Processo de desenvolvimento de Software) e como isso impacta o desenvolvimento e o desempenho do sistema.

## 2 REFERENCIAL TEÓRICO

O processo de desenvolvimento de software (PDS) é uma metodologia para o cumprimento de metas para o desenvolvimento de um software com alta qualidade. No PDS é definido qual será a abordagem adotada acerca de como o software será elaborado, bem como um conjunto de atividades metodológicas genéricas (PRESSMAN, 2011).

Pressman (2011) destaca que uma metodologia genérica define atividades padrões para o processo de desenvolvimento do software, como: comunicação, planejamento, modelagem, construção e entrega, além de um conjunto de atividades de apoio (*umbrella activities*). Sendo a garantia de qualidade uma dessas atividades de apoio, embora a definição de qualidade possa ser interpretada de diversas maneiras dependendo da ótica de quem a avalia, é unânime que é possível identificá-la independente da situação.

Embasando-se nos conceitos de Pressman (2011), foi possível traçar uma visão de apoio sobre qualidade de software, podendo ser definida como: uma gestão de qualidade efetiva aplicada de modo a criar um produto útil, que forneça valor mensurável para aqueles que o produzem e para aqueles que o utilizam.

De igual modo, Sommerville (2011) define, no processo de qualidade de software, a relevância da atuação dos times de QA (*Quality Assurance*) nas empresas de desenvolvimento, assegurando softwares confiáveis que atendam as expectativas de seus desenvolvedores e usuários.

Segundo Singh e Kassie (2018), uma das formas mais fundamentais de se aumentar a qualidade de um software é a diminuição de seus defeitos, que podem vir a causar falhas enquanto o software estiver em produção.

Uma das principais formas de diminuir esses defeitos é focar na prevenção durante o desenvolvimento, utilizando-se de um grande investimento em conhecimentos e habilidades do time.

A respeito da perspectiva do usuário final, Singh e Kassie (2018) apresentam dez principais fatores que influenciam na percepção de qualidade em um software. Esses fatores estão em conformidade com a Norma ISO/IEC 9126-1, definidos da seguinte maneira: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade, portabilidade, compreensibilidade, interoperabilidade, operabilidade e estética.

## 2.1 IMPORTÂNCIA DA GESTÃO DE QUALIDADE NAS EMPRESAS DE SOFTWARE

De acordo com Pressman (2011), ainda hoje, muitos programadores acreditam que a qualidade é algo que deve se preocupar apenas depois do código ter sido escrito. No entanto, a garantia da qualidade de software é uma atividade universal, devendo ser aplicada durante todo o PDS, não somente na fase de documentação.

Uma gestão de qualidade bem-feita dá suporte para que seja possível construir softwares com cada vez mais qualidade. Os fatores e atividades dessa gestão evitam que se crie um caos no projeto e permite ao desenvolvedor pensar em uma solução mais consistente, aumentando cada vez mais a qualidade do produto, conforme descrito por Pressman (2011).

Ao criar um produto com alto nível de qualidade, é gerado um valor tanto para a empresa desenvolvedora quanto para o usuário. Uma vez que o usuário terá para si um produto que atende e satisfaz suas necessidades, e a empresa tem um produto que necessita de menos correções de erros e falhas, tem-se menor demanda de suporte ao usuário permitindo que ela possa focar em entregar novos recursos e produtos.

## 2.2 QUALIDADE DE SOFTWARE

Durante o início da era da computação, a qualidade de software era responsabilidade exclusiva do desenvolvedor, tendo, na década de 1970, os primeiros padrões desenvolvidos pela indústria militar, e logo em seguida, se espalhado para empresas de software comercial.

Nesta seção, é apresentada a Norma ISO/IEC sobre qualidade de software, assim como os tipos de modelos utilizados.

### 2.2.1 Norma Sobre Qualidade de Software

A norma ISO/IEC 9126-1 é um padrão internacional para teste, qualidade e validação de software criado em 1991 que define as características e subcaracterísticas de qualidade, conforme ilustrado pela Figura 1 (ALIT; SUGIARTO, 2020).

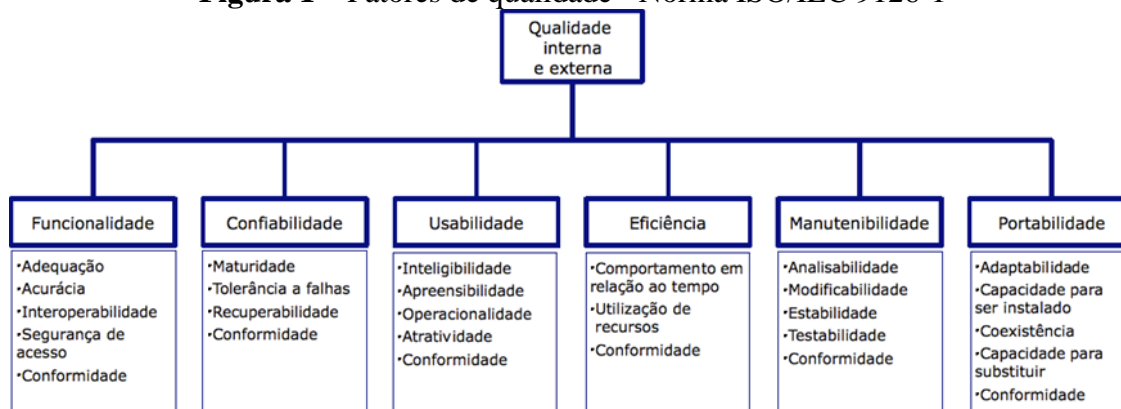
Fazem parte desta norma os atributos de funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. De acordo com a norma ISO/IEC 9126-1, esses atributos de qualidade podem ser definidos da seguinte forma:

**1) Funcionalidade:** considerada como um fator crucial para a qualidade do produto de software, pois está diretamente relacionada ao cumprimento dos requisitos funcionais definidos pelo usuário. Esse atributo possui as subcaracterísticas:

- Adequação: indica que o software atende aquilo que foi definido nos requisitos funcionais do projeto.
- Acurácia: é a capacidade do software em entregar os dados e os resultados de forma exata.
- Interoperabilidade: capacidade de interação e integração com outros produtos, como softwares ou hardwares.

- **Segurança:** capacidade do software em controlar acessos não autorizados a ele ou a suas funcionalidades.

**Figura 1** – Fatores de qualidade - Norma ISO/IEC 9126-1



Fonte: Adaptada de SINGH; KASSIE, 2018.

**2) Confiabilidade:** essa característica está associada ao funcionamento do software sem a presença de alguma falha ou erro durante sua execução e uso, e possui as seguintes subcaracterísticas:

- **Maturidade:** quantidade de falhas apresentadas pelo software.
- **Tolerância a falhas:** capacidade do software continuar operando, mesmo que parcialmente, em caso de alguma falha.
- **Recuperabilidade:** tempo e esforço necessário para reestabelecer as funcionalidades do software após uma falha.

**3) Usabilidade:** é a facilidade na operação do software por parte de seu usuário.

- **Inteligibilidade:** indica o esforço do usuário para entender e reconhecer os elementos presentes no software.
- **Apreensibilidade:** tempo e esforço necessário para que uma pessoa seja treinada para operar corretamente o software.
- **Operacionalidade:** esforço necessário para operar o produto e suas funções.

**4) Eficiência:** define um conjunto de atributos que está relacionado a performance do software.

- **Comportamento em relação ao tempo:** tempo necessário para que o software realize suas tarefas.
- **Utilização de recursos:** capacidade do software em utilizar de forma adequada os recursos disponíveis, como memória e armazenamento.

**5) Manutenibilidade:** facilidade com que o software pode receber manutenção e mudanças.

- **Analísabilidade:** tempo e esforços necessários para identificar e diagnosticar problemas.
- **Modificabilidade:** tempo e esforços necessários para corrigir, modificar e adaptar os problemas encontrados.
- **Estabilidade:** risco de as alterações causarem um efeito inesperado no software.
- **Testabilidade:** esforço necessário para realizar a verificação e validação do software após modificações.

**6) Portabilidade:** facilidade com que o software pode ser executado em diferentes ambientes, como mobile, web e desktop.

- Adaptabilidade: capacidade do software em se adaptar em ambientes específicos.
- Capacidade para ser instalado: tempo e esforço necessários para instalar o software em um ambiente.
- Coexistência: capacidade do software de coexistir e compartilhar recursos com outros sistemas.
- Capacidade para substituir: esforço necessário para substituir outro software já existente.

### 2.2.2 Tipos de Modelos de Qualidade de Software

Os processos de melhoria de software são um conjunto de práticas de processo baseadas em modelos de qualidade e na experiência da Engenharia de Software sob o conceito de maturidade e capacidade (MELLUZI NETO et al., 2018).

O objetivo dos processos de melhoria é promover a efetividade do PDS buscando trazer melhores resultados referentes ao custo, qualidade, produtividade, satisfação ao cliente, lucratividade e vantagem de mercado. São utilizados níveis de maturidade para identificar qual a situação atual da organização e identificar pontos de evolução. Quanto maior o nível de maturidade, maior é a capacidade de aplicação de processos, o que resulta em melhores resultados (MELLUZI NETO et al., 2018).

Neste trabalho é apresentado os principais conceitos dos modelos CMMI e MPS.BR, descritos nos tópicos 2.2.2.1 e 2.2.2.2.

#### 2.2.2.1 Modelo CMMI

A princípio, o CMMI começou como CMM (*Capability Maturity Model*) em 1987. Um projeto do Instituto de Engenharia de Software (*SEI - Software Engineering Institute*). Foi publicado pela primeira vez em 1991 o *CMM for Software*, que teve início como uma lista de verificação de fatores críticos de sucesso. O modelo também possuiu como base pesquisas na IBM (*International Business Machines*) Corporation e demais líderes de garantia de qualidade do século XX, como Philip Crosby e W. Edwards Deming (MICROSOFT, 2022).

O nome e os cinco níveis de representação em etapas foram inspirados nos modelos de maturidade de fabricação de Philip Crosby<sup>1</sup>. O modelo CMM obteve uma adoção considerável e passou por várias revisões e se tornou tão abrangente que levou ao desenvolvimento de CMMs por uma gama de diferentes ramos além do software. Devido a esses fatores, a proliferação de novos modelos se tornou confusa, dessa maneira, o governo americano financiou um projeto para criar uma estrutura única que integrasse as engenharias de sistemas, software e o desenvolvimento de produtos. O resultado desse projeto foi o CMMI (MICROSOFT, 2022).

O modelo CMM visa avaliar a maturidade dos processos de uma organização e apresenta diretrizes sobre como melhorar os processos, com o objetivo de melhorar o produto. O modelo CMMI também é um modelo de gerenciamento de riscos fornecendo uma maneira de medir a capacidade de uma organização de gerenciar riscos (MICROSOFT, 2022).

O CMMI foi projetado para ser usado como base para uma iniciativa de aprimoramento de processo, com seu uso na avaliação somente para um sistema de suporte e aprimoramento de medidas. A base fundamental é uma área de processo que define os objetivos e diversas atividades que são geralmente usadas para atendê-los (MICROSOFT, 2022).

---

<sup>1</sup> Empresário e escritor estadunidense que contribuiu para a teoria da gestão e métodos de gestão da qualidade

A avaliação organizacional mostra o nível em que se está operando e isso é um indicador da capacidade da organização de gerenciar riscos.

Conforme descrito no *CMMI Institute*, o CMMI está dividido em cinco níveis de maturidade (Figura 2), sendo eles:

**1) Inicial:** em que o processo é caracterizado como imprevisível e reativo. O trabalho geralmente é concluído, porém com atrasos e acima do orçamento (CMMI INSTITUTE, 2022).

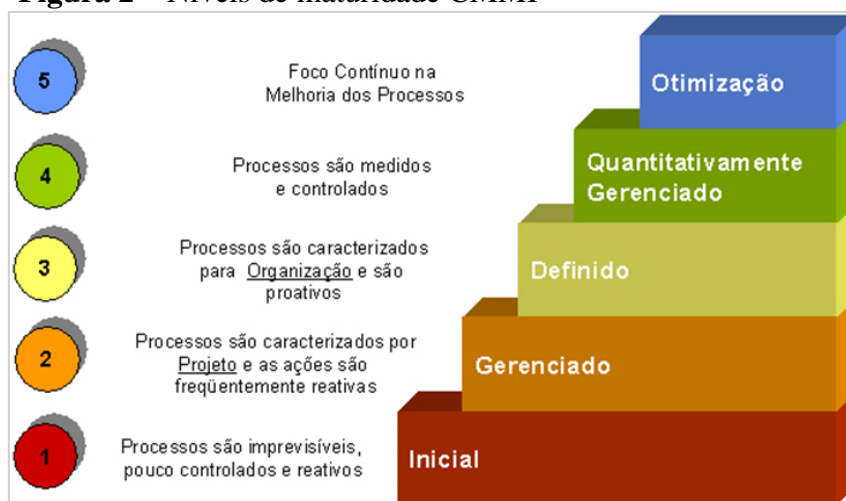
**2) Gerenciado:** há gerenciamento no nível do projeto. O trabalho é planejado, executado, medido e controlado. (CMMI INSTITUTE, 2022).

**3) Definido:** o processo é proativo ao invés de reativo. Os padrões da organização fornecem orientação entre projetos, programas e portfólios (CMMI INSTITUTE, 2022).

**4) Quantitativamente Gerenciado:** o nível de maturidade é médio e controlado. A organização é guiada por dados, com objetivos quantitativos de melhoria de desempenho (CMMI INSTITUTE, 2022).

**5) Otimizado:** o processo já é considerado estável e flexível. A organização está focada em melhoria contínua para responder a oportunidade e mudanças (CMMI INSTITUTE, 2022).

**Figura 2 – Níveis de maturidade CMMI**



Fonte: Adaptada de ISD BRASIL, 2022.

Segundo Melluzi Neto et al. (2018), o modelo CMMI conta com três vertentes, são elas: *CMMI for Development* (CMMI-DEV), que engloba desenvolvimento de software, *CMMI for Services* (CMMI-SCV) para prestação de serviços, e *CMMI for Acquisition* (CMMI-ACQ) que trata de contratações e aquisições, definidas da seguinte maneira:

- **CMMI for Development** (CMMI para Desenvolvimento): é um conjunto integrado de melhores práticas que melhora o desempenho e os principais recursos para organizações que desenvolvem produtos, componentes e serviços.

- **CMMI for Services** (CMMI para Serviços): por sua vez, o *CMMI Services* é um conjunto integrado de práticas recomendadas que aprimora o desempenho e os principais recursos para organizações que fornecem serviços, incluindo B2B, B2C, serviços independentes e aqueles que fazem parte de uma oferta de produto.

- **CMMI for Acquisition** (CMMI para Aquisição): por fim, o *CMMI for Acquisition* é um conjunto integrado de melhores práticas que melhora o desempenho e os principais recursos para organizações que adquirem componentes, bens ou serviços de outra empresa.

### 2.2.2.2 Modelo MPS.BR

O MPS.BR é um modelo brasileiro, criado pela Associação para a promoção da Excelência do Software Brasileiro (SOFTEX), apoiado pelo Ministério da Ciência, Tecnologia e Inovações. Teve seu início em 2003 com objetivo de promover a melhoria na capacidade de desenvolvimento de software, serviços e recursos humanos em empresas de TI (MELLUZI NETO et al., 2018).

O modelo MPS.BR é dividido em sete níveis de maturidade, em que cada um deles apresenta qual o estado atual do PDS. Esses níveis são apresentados na Figura 3. De acordo com Softex (2021), os níveis propostos pelo modelo MPS.BR são descritos como:

**1) Nível G:** Com a implementação desse nível, a execução do processo passa a ser gerenciada, e são produzidos resultados definidos, planejamento e monitoramento a respeito da execução do projeto e as pessoas estão preparadas para executar suas responsabilidades no processo.

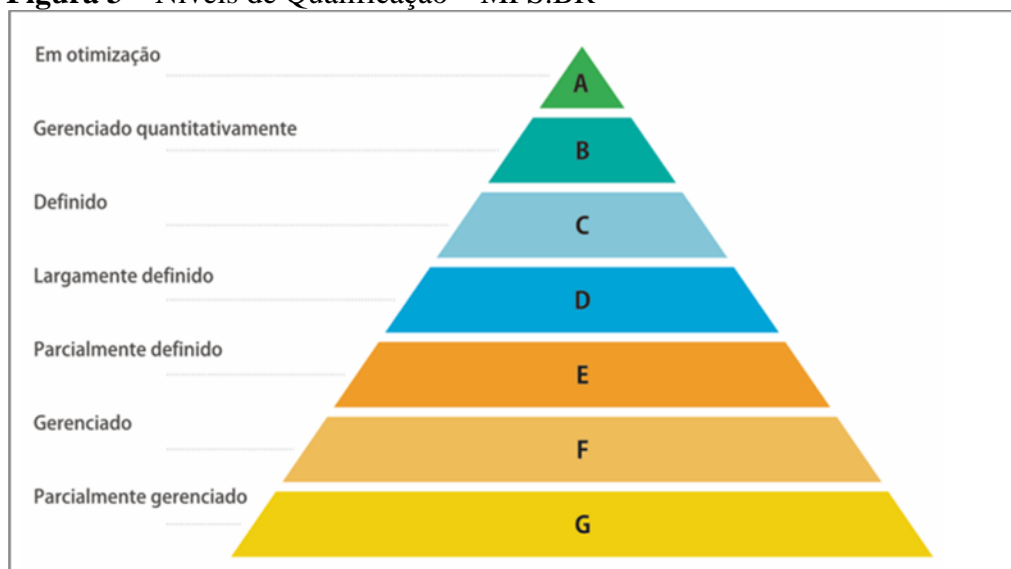
**2) Nível F:** A execução do processo e produtos de trabalho são gerenciados. Após a implementação desse nível, tem como resultado uma verificação objetiva de que o processo é seguido.

**3) Níveis E, D e C:** O processo é definido. Neste nível, o processo que anteriormente era gerenciado, passa a ser definido e adaptável. Tem como resultado uma avaliação objetiva do processo e padrões, registro dos resultados e não conformidades são asseguradas de correções. Melhorias no processo são identificadas durante a execução das atividades de garantia da qualidade e são disponibilizadas para a organização, com informações relacionadas ao processo e seus ativos.

**4) Nível B:** O processo é previsível. O processo que anteriormente era definido e adaptável, passa a ser previsível dentro dos limites definidos, buscando a obtenção de seus resultados. Tem como resultado de sua implantação técnicas estatísticas e quantitativas para prever a qualidade e o desempenho dos processos.

**5) Nível A:** O processo é melhorado continuamente. Ao atingir esse nível de maturidade, os processos selecionados e previsíveis são objetos de melhoria contínua e estão de acordo com os objetivos da organização. Propostas de melhoria têm seus efeitos nos objetivos, qualidade e desempenho analisadas com base em técnicas estatísticas e quantitativas.

**Figura 3 – Níveis de Qualificação – MPS.BR**



Fonte: FUMSOFT apud ASSUNÇÃO; VASCONCELLOS, 2013.

Enquanto os níveis do CMMI foram desenvolvidos tendo como foco os requisitos de uma empresa ideal, o MPS.BR, baseando-se nesses níveis, foi desenvolvido com um foco maior no mercado de software brasileiro (FERREIRA et al.; 2007).

Em uma comparação aos níveis de maturidade do MPS.BR com os níveis do CMMI, é identificado que o MPS.BR visa possibilitar uma melhoria mais gradual ao processo de desenvolvimento, principalmente ao ser aplicado em pequenas e médias empresas brasileiras, além de ter um custo de implantação menor (FERREIRA et al.; 2007).

### 3 METODOLOGIA

Para fins de embasamento teórico sobre processos de qualidade, foram realizadas pesquisas bibliográficas na área de Engenharia de Software, destacando Sommerville (2011) e Pressman (2011), artigos acadêmicos publicados em bases científicas, como: Google Scholar, IEEE Xplore Library e em sites de empresas que dão suporte a guias oficiais dos modelos citados no presente artigo, como: Microsoft Docs, Softex, CMMI *Institute* e o ISD Brasil.

Após o levantamento bibliográfico, também foi realizada uma pesquisa de campo, no período de 08 a 21 de setembro de 2022, com empresas de desenvolvimento de software da região de Jales, do Estado de São Paulo. Essa pesquisa teve o objetivo de levantar dados a respeito do conhecimento e aplicação dos conceitos de qualidade de software nessas empresas de desenvolvimento. Para essa pesquisa, utilizou-se a ferramenta *Microsoft Forms* e enviada por e-mail para os profissionais responsáveis pelo setor de desenvolvimento de software.

### 4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Levando em consideração os dados discutidos na *Seção 2* sobre conceitos e modelos de qualidade dentro do ambiente de desenvolvimento de um sistema, a pesquisa de campo foi realizada com 12 empresas de desenvolvimento situadas na região de Jales e Fernandópolis.

Desse modo, a *Seção 4.1* apresenta a discussão e a análise dos resultados com o cenário atual das empresas a respeito da existência de um processo em vigor de controle da qualidade, independente da forma que é aplicado. Logo, a *Seção 4.2* aborda a visão das empresas pesquisadas sobre qualidade de software, como são definidos os processos no ambiente de trabalho e na vida útil de seu produto.

Finalizando a análise, a *Seção 4.3* apresenta a visão das empresas pesquisadas sobre os modelos de melhoria de processo abordados na *Seção 2* e suas normas de qualidade.

#### 4.1 CONTROLE DE QUALIDADE NO CENÁRIO ATUAL NAS EMPRESAS

Embasando-se nas pesquisas que este trabalho aborda sobre qualidade de software e sua relevância durante o PDS, buscou-se compreender na prática a realidade das empresas, se elas aplicam algum processo de qualidade de software em seu dia a dia.

A pesquisa foi aplicada no mês de setembro de 2022 e constatou-se que 92% das empresas pesquisadas fazem uso do gerenciamento da qualidade no processo de desenvolvimento do seu produto, conforme é ilustrado no Gráfico 1.

Buscou-se entender também, a disposição das empresas em adotarem um modelo de qualidade, mesmo que isso exija uma reestruturação ou mudança de seus processos internos.

Os resultados foram positivos, com 100% dos entrevistados afirmando que estão dispostos a implantar um modelo de qualidade. Dessa maneira, pode-se constatar que as empresas atuais entendem a importância do gerenciamento da qualidade no mercado contemporâneo.



Quando questionado sobre financeiro em qualidade de software, a pesquisa apontou que 100% dos empresários estão dispostos a investir em qualidade para o cenário da empresa.

**Gráfico 1** – O gerenciamento da qualidade incluso no processo de desenvolvimento do produto



Fonte: Elaborado pelos autores.

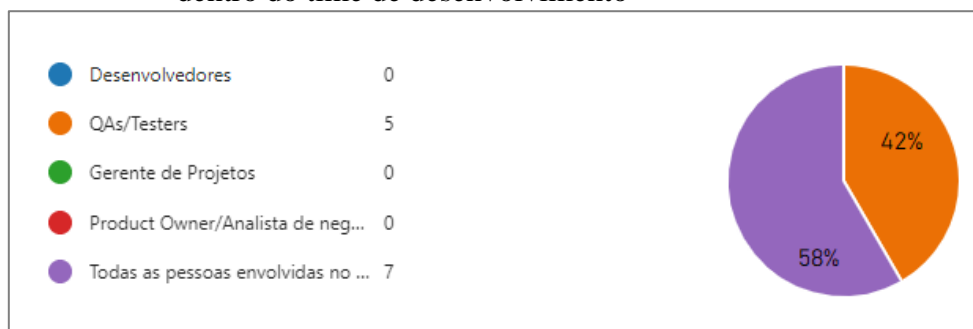
Os resultados coletados na pesquisa evidenciam que as empresas entendem a importância de aplicar conceitos e práticas de qualidade em seu processo de desenvolvimento, mostrando-se abertas para a sua implantação. A implantação de processos de qualidade traz inúmeros benefícios tanto para empresários quanto para os clientes que fazem uso desses produtos, além de tornar esses mais competitivos no mercado.

#### 4.2 VISÃO DAS EMPRESAS SOBRE PROCESSO DE QUALIDADE

Por intermédio da pesquisa de campo, foi questionada a visão das empresas a respeito do processo de qualidade, para que fosse possível mensurar o que as empresas entendem por processo de qualidade, não especificamente a respeito dos modelos de qualidade apresentados anteriormente na *Seção 2* deste trabalho.

De acordo com os resultados obtidos na pesquisa de campo, 58% das empresas afirmam que, em sua visão, todo o time envolvido no desenvolvimento do produto de software é responsável pela qualidade dele. Enquanto 42% responderam que a responsabilidade pertence única e exclusivamente ao time de QAs (Qualidade), conforme representado pelo Gráfico 2.

**Gráfico 2** – Visão da empresa sobre o responsável pela qualidade do software dentro do time de desenvolvimento



Fonte: Elaborado pelos autores.

Das empresas entrevistadas, constatou-se que 67% têm hoje alguém responsável pela qualidade do software, enquanto 33% disseram não possuir alguém responsável por cuidar dessa área (Gráfico 3).

**Gráfico 3** – Existência de pessoas responsáveis pela qualidade de software nas empresas



Fonte: Elaborado pelos autores.

Nessa pesquisa, buscou-se saber também, o papel que essas pessoas assumem atualmente dentro da organização, e identificou-se que 48% das respostas incluem cargos dentro do time de qualidade (Quadro 1).

**Quadro 1** – Cargos exercidos pelos responsáveis pela qualidade do software nas empresas

Respostas
QA
Gerente da qualidade
Desenvolve software.
Análise de teste de software e Testers
Qualidade
Não sei informar
Quality assurance
QA

Fonte: Elaborado pelos autores.

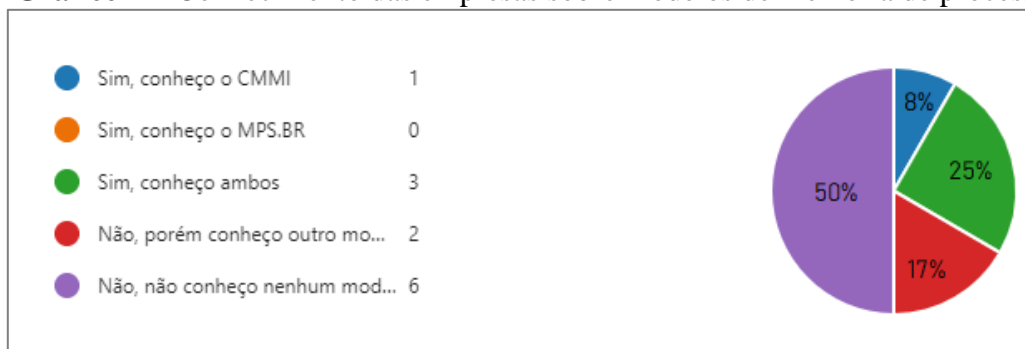
O grupo de colaboradores responsáveis pela qualidade do software reforça a visão das empresas sobre a qualidade durante o processo de desenvolvimento. Pode-se concluir, a partir desses dados, que a qualidade de software ainda é vista como responsabilidade de apenas uma parte do time e não como incumbência de todos.

#### 4.3 CONHECIMENTO TEÓRICO DAS EMPRESAS SOBRE QUALIDADE

Durante a pesquisa, também foi abordado sobre alguns modelos de qualidade para compreender o conhecimento teórico das empresas. Foi realizado um levantamento tendo como foco dois modelos de melhoria de processo: CMMI e MPS.BR; e a norma ISO/IEC 9126-1, ambos citados na *Seção 2* deste trabalho.

Quando os entrevistados foram questionados sobre os modelos de melhoria de processo: CMMI e MPS.BR, 25% dos entrevistados responderam que conhecem ambos os modelos, 8% conhecem o CMMI, 17% alegaram que não conhecem os referidos modelos, mas conhecem outros, e 50% informaram não conhecer nenhum modelo de melhoria de processo. Essas informações podem ser observadas no Gráfico 4.

**Gráfico 4** – Conhecimento das empresas sobre modelos de melhoria de processo



Fonte: Elaborado pelos autores.

Quando questionado a respeito da Norma ISO/IEC 9126-1, 25% das empresas conhecem a norma, 17% informaram não conhecer, porém tem ciência de outros modelos, e por fim, 58% dos entrevistados afirmaram não conhecer nenhum modelo de qualidade (Gráfico 5).

**Gráfico 5** – Conhecimento das empresas sobre modelos de qualidade de software



Fonte: Elaborado pelos autores.

Finalizando a pesquisa, foi questionado se as empresas fazem uso de algum modelo de qualidade, como, por exemplo, a Norma ISO/IEC 9126-1. Das empresas entrevistadas, 8% utilizam a norma como modelo de qualidade, 25% alegaram utilizar outro modelo, porém não informado, e 67% afirmaram não utilizar nenhum modelo de qualidade (Gráfico 6).

**Gráfico 6** - Utilização de algum modelo de qualidade nas empresas



Fonte: Elaborado pelos autores.

De acordo com os dados coletados na pesquisa, as empresas compreendem que a qualidade tem um papel de suma importância dentro do Processo de Desenvolvimento de Software, porém não o exploram como um todo, concentrando a responsabilidade da qualidade do produto no time de QAs.

Diante do cenário apresentado baseado na pesquisa de campo, este trabalho evidenciou que as empresas aplicam o controle de qualidade na fase final do PDS, ou seja, na fase de testes, não sendo aplicada garantia da qualidade desde o planejamento com a definição dos “Itens de Trabalho”<sup>2</sup> até o momento da implantação do software.

A visão de testes e qualidade está limitada aos testes realizados no processo, em que o “Item de Trabalho” já está desenvolvido. No entanto, os modelos de melhoria de processo, tais como o CMMI e MPS.BR, citados anteriormente na *Seção 2*, propõe não apenas o controle da qualidade, mas o gerenciamento dos projetos que tem início no “Plano Global do Projeto” e tem seu término no “Plano de Gerenciamento de Contratos”, conforme definido pelo *Project Management Institute* (PMI).

Neste contexto, para que o software possa ser desenvolvido adequadamente, a etapa de testes deve ser considerada como uma das fases do PDS e não a única responsável pela qualidade do software (BASTOS et al., 2012).

O tipo mais comum de teste de software ocorre de forma manual, realizado por seres humanos, que averiguam se o que foi pedido no “Item de Trabalho” foi de fato atendido e está funcionando conforme o esperado. A responsabilidade de realizar esses testes e dar o aval para que o item vá para produção é do Analista de Qualidade (QA), que hoje, são comuns nas empresas de desenvolvimento de software (*softwarehouse*).

Para os testes, foi identificado através de um estudo de caso em uma das empresas participantes da pesquisa de campo, que são escritos “*Casos de Teste*” para averiguar o comportamento do sistema. A forma como são organizados pode variar de acordo com a técnica que o Analista responsável prefere trabalhar, em alguns casos são utilizados *Checklists*.

A Figura 4 demonstra um trecho do roteiro de testes que são executados no software antes de realizar a liberação de uma nova versão em ambiente de produção. Esses testes têm como objetivo verificar as principais funções do sistema, garantindo que ele não apresente falhas graves nesses pontos. Cada analista de testes executa um roteiro (Trilhas) que contém áreas diferentes do sistema para serem testadas.

Todo esse processo executado pelos analistas de testes são apenas uma parte do gerenciamento da qualidade (BASTOS et al., 2012). A partir dos dados adquiridos por meio da pesquisa de campo, as empresas na região noroeste do estado de São Paulo não possuem qualidade de ponta a ponta, mas focam apenas no setor de execução de testes como garantia de qualidade. Essas empresas não seguem modelos de qualidade citados neste trabalho, mas cada uma a sua maneira, implanta em seu processo de desenvolvimento aquilo que compreende como qualidade.

---

<sup>2</sup> Work Items (Itens de Trabalho) é como são chamados os cards nas empresas que fazem uso de ferramentas da abordagem DevOps, como o Azure DevOps e metodologias ágil como Scrum e Kanban. São utilizados para controle de erros (BUGs); PBIs (*Product Backlog Items*); Épicas, que são agrupamentos de novas funcionalidades (*Epic*), Funcionalidades implementadas (*Features*); Histórias de usuários, entre outros dentro de um quadro (*Board*).

**Figura 4** – Checklist de testes funcionais



 Add cover Add comment

## TRILHA 1

► Movimento

▼ Cadastros

▼ Convênio

- ☒ Cadastrar
- ☒ Consultar
- ☐ Editar
- ☐ Desativar
- ☐ Excluir (não permitir Excluir se houver vínculo no sistema)

► Gerenciar produtos

► Balanço de estoque

Fonte: Elaborada pelos autores.

## 5 CONSIDERAÇÕES FINAIS

No contexto dos modelos de melhoria de processo estudados neste trabalho, foi possível identificar que o processo de testes é apenas uma fase do gerenciamento da qualidade. Em comparação com os resultados obtidos com a pesquisa de campo, observou-se o cenário real das empresas que atuam no noroeste paulista. Pode-se perceber que existe uma noção, por parte dos entrevistados, a respeito dos modelos de qualidade de software, no entanto, essa visão a respeito de qualidade de software ainda é muito limitada, não concedendo, dessa maneira, espaço para que as práticas de qualidade sejam exploradas no PDS como um todo.

Em sua maioria, as empresas participantes da pesquisa demonstraram que o seu conhecimento teórico em modelos de melhoria de processos é escasso e baixo, e a implantação da qualidade está exclusivamente voltada para os testes de software.

A qualidade de software é de suma importância para o desenvolvimento de um bom produto. As empresas têm consciência de que é estritamente necessário que exista controle da qualidade dentro de seu PDS, no entanto, não praticam o gerenciamento da qualidade. Dessa maneira, não existe controle de qualidade de ponta a ponta, apenas a parte de testes que são executados após o incremento no sistema.

Conforme pautado pelas pesquisas, as empresas do noroeste paulista da microrregião de Jales/SP estão preocupadas com a qualidade do seu produto e em inserir o processo de qualidade dentro de seu PDS. Entretanto, ainda é necessário compreender que a qualidade de software não se inicia e não se resume na fase de testes. Ela precisa ser aplicada desde o planejamento do que será implementado até o momento em que o software ou a funcionalidade desenvolvida seja disponibilizada em ambiente de produção. Dessa maneira, se faz possível o gerenciamento da qualidade de ponta a ponta no PDS. Um processo de qualidade bem definido, consequentemente, é eficaz para que o software entregue ao seu cliente aquilo que lhe foi proposto.

## REFERÊNCIAS

ALIT, R.; SUGIARTO; A. H. Quality analysis of SIRUP on functionality and usability characteristics using ISO 9126. *In: INFORMATION TECHNOLOGY INTERNATIONAL SEMINAR – ITIS*, 6th., 2020, Surabaya, Indonesia. **Proceddings** [...]. Surabaya, Indonesia, 2020. Disponível em: <https://ieeexplore.ieee.org/document/9321042>. Acesso em: 21 set. 2022.

ASSUNÇÃO, R. P.; VASCONCELLOS, F. P. **MPS.BR**: a experiência e os benefícios em implantar o modelo nos níveis G E F: um estudo de caso. 2013. Disponível em: <https://docplayer.com.br/2068085-Mps-br-a-experiencia-e-os-beneficios-em-implantar-o-modelo-nos-niveis-g-e-f.html>. Acesso em: 21 set. 2022.

BASTOS, A. *et al.* **Base de conhecimento em testes de software**. 3. ed. São Paulo: Martins Fontes, 2012.

CMMI INSTITUTE. **CMMI levels of capability and performance**. Disponível em: <https://cmmiinstitute.com/learning/appraisals/level>. Acesso em: 21 set. 2022.

FERREIRA, A. *et al.* Applying ISO 9001:2000, MPS.BR and CMMI to achieve software process maturity: BL Informatica's Pathway. *In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING – ICSE'07*, 29th., 2007, Minneapolis, MN, USA. **Proceddings** [...]. Minneapolis, MN, USA, 2007. Disponível em: <https://ieeexplore.ieee.org/document/4222625>. Acesso em: 21 set. 2022.

ISD BRASIL. **O que é CMMI?** Disponível em: <http://www.isdbrasil.com.br/o-que-e-cmmi.php>. Acesso em: 21 set. 2022.

LENILDO. **Qualidade de software**: engenharia de software 29. 2010. Disponível em: <https://www.devmedia.com.br/qualidade-de-software-engenharia-de-software-29/18209>. Acesso em: 21 set. 2022.

MELLUZI NETO, G. *et al.* Resultados da implantação de CMMI e MPS-BR em empresas de desenvolvimento e manutenção de software: a visão da alta gestão. **Revista Brasileira de Computação Aplicada**, v. 10, n. 1, p. 2-10, maio 2018.

MICROSOFT. **Integração de modelo de maturidade de tela de fundo para capacidade (CMMI)**. 2022. Disponível em: <https://learn.microsoft.com/pt-br/azure/devops/boards/work-items/guidance/cmmi/guidance-background-to-cmmi?view=azure-devops>. Acesso em: 21 set. 2022.

PRESSMAN, S. R. **Engenharia de software**: uma abordagem profissional. 7. ed. São Paulo: AMGH, 2011.

SINGH, J.; KASSIE, N. B. User's perspective of software quality. *In: INTERNATIONAL CONFERENCE ON ELECTRONICS, COMMUNICATION AND AEROSPACE TECHNOLOGY – ICECA*, 2th, 2018, Coimbatore, India. **Proceddings** [...]. Coimbatore, India, 2018. Disponível em: <https://ieeexplore.ieee.org/document/8474755>. Acesso em: 21 set. 2022.

SOFTEX. **MPS.BR**: melhoria de processo do software brasileiro: 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](https://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_Geral_Software_2012-c-ISBN-1.pdf). Acesso em: 21 set. 2022.

SOMMERVILLE, I. **Engenharia de software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.