

GESTÃO DA QUALIDADE NO DESENVOLVIMENTO DE PRODUTO DE SOFTWARE

QUALITY MANAGEMENT IN SOFTWARE PRODUCT DEVELOPMENT

Jéssica Bruna da Costa Aguiar – jessicaguiar345@gmail.com
Faculdade de tecnologia de Taquaritinga – Taquaritinga – São Paulo – Brasil

Fernando Cesar Graciano – Fernando.graciano@fatec.sp.gov.br
Faculdade de Tecnologia de Taquaritinga – Taquaritinga – São Paulo – Brasil

DOI: 10.31510/inf.v18i2.1314

Data de submissão: 14/09/2021

Data do aceite: 03/11/2021

Data da publicação: 30/12/2021

RESUMO

A exigência e a competitividade estão tornando desafiador para a produção de produtos de boa qualidade nos tempos atuais. Sendo assim a alta em criatividade e investimento faz com que seus produtores tenham que ter qualidade e eficiência. Para o processo de um produto de software existe requisitos básicos para uma boa qualidade, como por exemplo: bom desempenho, adaptável as necessidades específicas, fácil de usar e sem defeitos. Por isso esse artigo tem como objetivo mostrar que com um bom gerenciamento de projeto com prazo e cronogramas desde o começo ate o final da produção tende a ter qualidade, pois é fundamental que o software seja eficaz, confiável e siga as exigências.

Palavras-chave: Qualidade. Software. Processo. Produto

ABSTRACT

Demand and competitiveness are making it challenging to produce good quality products in current times. Thus, the high in creativity and investment makes its producers have to have quality and efficiency. For the process of a software product there are basic requirements for good quality, such as: good performance, adaptable to specific needs, easy to use and without defects. So, this article aims to show that with a good project management with deadlines and schedules from the beginning to the end of production tends to have quality, because it is essential that the software is effective, reliable and follows the requirements.

Keywords: Quality. Software. Process. Product

1 INTRODUÇÃO

Quando a Associação Brasileira de Normas Técnicas (ABNT) foi criada no Brasil, foram definidos um conjunto de normas afim de garantir mais padronização e qualidade garantindo aprimorar a fabricação de produtos.

Já a perspectiva do produto software esse assunto de qualidade é bem extenso. Conforme Bezerra, Araújo e Chalegre (2011) para cada fase de desenvolvimento de um produto existem várias ferramentas e técnicas que podem garantir um melhor aspecto para o produto final.

Sendo assim um problema da qualidade de software é definir objetivos ao qual pretende atingir no projeto. Para isso é preciso enumerar as características que se deseja do software e incluir valores qualitativos e quantitativos a ser respeitados. Produto medido a uma série de características seus diagnósticos de qualidade são mais precisos.

Este artigo tem como objetivo realizar uma pesquisa bibliográfica e apresentar algumas técnicas para melhor o desenvolvimento de produto de software.

2 FUNDAMENTAÇÃO TEÓRICA

Para o desenvolvimento deste artigo é necessário que esta seção apresente conceitos relacionados ao tema afim do entendimento de alguns pontos essenciais para a realização da pesquisa bibliográfica.

2.1 Sistemas de informação

Segundo Carpon e Johnson (2004), o sistema de informação é composto por um conjunto de comandos com a finalidade ordenar que os componentes eletrônicos realizem operações matemáticas, grave informações no discos rígido, e exiba os resultados dos cálculos de maneira rápida e de simples entendimento.

Dutra, Barbosa (2019) alegam que a informação é valiosa para qualquer tomada de decisão organizacional, e para que seja eficaz necessita ser bem administrada. Sabendo então que a sobrevivência das organizações depende de habilidades para geri-las, o acompanhamento adequado nas etapas faz com que eficiência e eficácia seja decisório para uma boa gestão da informação.

A sua importância nas organizações deveria ser como um recurso fundamental. Com o passar dos anos essa ideia foi se aperfeiçoando. De acordo com Vance (2000), a administração reconhece a importância que a informação exerce sobre uma empresa.

2.2 Requisitos software

Requisito de sistema são descrições dos serviços que um sistema devesse prestar e as restrições a sua operação (SOMMERVILLE,2009). Esses requisitos mostram as necessidades dos clientes a um determinado propósito, como controlar um dispositivo, um pedido ou encontrar informações.

Segundo Pádua (2009), os requisitos reúnem as atividades que buscam resolver, os processos enunciados de forma clara e objetiva em produto software. Como exemplos, a condição ou potencialidade que o usuário precisa para resolver problema ou atingir um objetivo; condição ou potencialidade que o sistema, componente ou produto precisa ter para ser aceito, ou seja, satisfaça o que for formalmente imposto; Características documentada.

Pádua (2009) ainda afirma que os requisitos necessitam ser levantados pela equipe de projeto, em conjunto com os usuários finais, clientes solicitantes e, posteriormente, especialista da área de aplicação. O conjunto de técnicas empregadas para levantar, detalhar, documentar e validar os requisitos de um produto forma a disciplina de Engenharia de Requisitos, geralmente chamada de Requisitos.

Sabemos então que a identificação de requisitos no começo de um projeto é muito importante. Os clientes muitas vezes não sabem do que realmente necessitam, fazendo com que dificulte a criação de um projeto que satisfaça suas necessidades. (MORAIS; ZANIN, 2017) ele é crucial para todas as demais etapas e essa etapa também tem um cunho investigativo, que os desenvolvedores além de criar o software, tem que extrair o máximo de informação do que ele precisa desse cliente.

2.2.1 Requisitos funcionais.

Segundo Sommerville (2007) os requisitos funcionais são todos os serviços que o produto de software deve realizar, desde a entrada dos dados, como eles devem ser processados, armazenados e exibidos em tela ou em relatórios. Em alguns casos, também podem explicar o que o sistema não deve fazer.

Os requisitos funcionais são fundamentais para o desenvolvimento de um software pois são exatamente o que os clientes ou usuários finais necessitam automatizar em seu processo de negócio.

2.2.2 Requisitos não funcionais

Os requisitos não funcionais não interferem no desenvolvimento produto de software, ou seja, não tem nenhuma informação do processo de negócio a ser automatizado. Em vez disso, os requisitos que estabelecem o comportamento do sistema depois de desenvolvido.

Em outras palavras, apesar de não interferirem diretamente nas automatizações dos processos de negócio, são essenciais para alcançar o objetivo final do software, ou seja, são relacionados com uso do sistema, como por exemplo, usabilidade, segurança, disponibilidade, desempenho, manutenção e tecnologias envolvidas.

2.3 Processos de Negócios

Segundo PAIM (2009) um processo de negócio é sinalizado como um conjunto de atividades estruturadas, realizadas em sequência e relacionadas entre si, cujo objetivo final é produzir um produto ou prestar um serviço. Este resultado então pode ser vendido para um ou mais clientes.

Toda empresa ou organização, independentemente do tamanho, possui um conjunto de tarefas e atividades que colaboram e interagem para a entrega de algum tipo de produto ou serviço. Estes conjuntos de atividades e ou tarefas podem conter envolvidos pessoas, máquinas, sistemas, sensores, departamentos, etc. (BECKER; ROSEMANN; VON UTHMANN, 2000).

O processo de negócio é onde o produto de software será crucial, pois o software deve garantir um melhor desempenho dentro deste conjunto de atividades afim de aumentar a produção, diminuir o custo e aumentar a lucratividade.

2.4 Ciclo de vida do software

Modelos de ciclos de vida de um software mostram técnicas que são utilizadas para desenvolver um sistema na prática. Com os avanços tecnológicos notamos a evolução que eles passaram e com isso os impactos que eles trouxeram para o software, ate porque temos

como um produto final o sistema funcional no qual atende as necessidades do cliente. (MORAIS, 2017).

Para Morais (2017), “o desenvolvimento de software é considerado diferente na prática por duas razões. Primeiramente, os profissionais de software são seres humanos e, portanto, cometem erros. Em segundo lugar, as necessidades do cliente podem mudar enquanto o produto de software está sendo desenvolvido”. São esses uns dos motivos pelo qual o produto de software passa por diversos processos para classificar como pronto. Outro fator importante desse assunto é que um software nunca é finalizado, porque este sempre vai passar por modificações, na verdade ele sempre estará em constante evolução.

Quando um software está sendo criado seu processo pode ser definido como um conjunto de atividade de trabalho, ação e tarefas.

2.5 Gestão da qualidade

Com o aumento da globalização e avanços da tecnologia e mudanças em todas as áreas, torna a competição no mercado mais acirrado. Com isso consumidores se tornaram mais exigentes dado ao fato de se ter cada vez mais opção no mercado. Mais do que nunca a qualidade ocupa uma posição importante em qualquer organização.

Segundo Lélis (2011), a décadas antes da internet e globalização a gestão da qualidade era um bicho de sete cabeças para empresas principalmente no Brasil.

“O conceito da qualidade é amplo e tem variado ao longo do tempo. No campo da gestão, a qualidade pode ser definida como um conjunto de propriedades e características de um produto, processo ou serviço que lhe fornecem a capacidade de satisfazer as necessidades explícitas e implícitas do cliente.” (Lizardo & Ribeiro, 2020)

Lélis (2011) ainda ressalta que nos últimos 20 anos os consumidores estão mais de olho nas organizações, levando em conta como se comportam em relação ao meio ambiente e a sociedade. Para ela gestão da qualidade tem três exigências:

- **Ética:** meio ambiente, sociedade;
- **Não gerar custos, reduzi-los:** evitando desperdício de mão-de- obra e matérias, redução de tempo na produção, mais satisfação do trabalhador e menos estresse;

- **Planejamento da qualidade:** Como o velho ditado “prevenir é melhor que remediar”, afinal é prevenir os problemas planejamento com cuidado cada etapa de um produto para evitar futuras dor de cabeça.

Sabemos então que qualidade é importante para qualquer produto e para o produto de software não é diferente.

Pressman (2011) afirma:

“Software de alta qualidade é um objetivo importante..., qualidade de software pode ser definida como uma gestão de qualidade efetiva aplicada de modo a criar um produto útil que forneça valor mensurável para aqueles que o utilizam.”

Essa é uma definição que pode ser modificada e debatida interminavelmente. Temos três pontos importante:

1. **Gestão da qualidade efetiva:** infraestrutura que do suporte para qualquer tentativa de se construir um software de alta qualidade, sendo então um mecanismo de controle e equilíbrio que ajudam a evitar o caos no projeto.
2. **Produto útil:** que fornece conteúdo, as funções e recursos que o usuário final deseja e não menos importante fornecer confiabilidade e isenção de erros.
3. **Agregar valor, tanto ao fabricante quanto ao usuário:** Com um software de qualidade gera benefícios para empresa e comunidade de usuário final

2.5.1 Dimensões de qualidade de Garvin

David Garvin (1987) sugere que qualidade é considerada multidimensional onde começa com avaliação de conformidade terminando com visão transcendental (estética). Mesmo que as oitos dimensões de qualidade de Garvin não foi criada especificamente para software, pode-se aplicada quando considerada qualidade de software, são eles:

- ✓ Qualidade de desempenho;
- ✓ Qualidade dos recursos;
- ✓ Confiabilidade;
- ✓ Conformidade;
- ✓ Durabilidade;
- ✓ Facilidade de manutenção;
- ✓ Estética;
- ✓ Percepção.

Essas dimensões de Garvin nos dão a uma visão indulgente da qualidade de software, podem considerar apenas subjetivamente. (PRESSMAN,2011)

2.5.2 Fatores de qualidade de McCall

A proposta de categorização dos fatores que afetam a qualidade de software foi criada por MacCall, Richards e Walters (1977), fatores esses que tem três importantes aspectos do produto de software. São eles: características operacionais, habilidade de suportar mudanças e adaptabilidade a novos ambientes. Com esses fatores MacCall(1977) e seus colegas fizeram as seguintes descrições:

- Correção: O quanto o programa satisfaz as especificações e os objetivos da missão do cliente;
- Confiabilidade: O quanto pode esperar de um programa na realização da função pretendida com precisão exigida;
- Eficiência: O quanto de recursos computacionais e códigos exigidos por um programa para desempenhar a função;
- Integridade: O quanto se pode controlar o acesso de pessoas não autorizadas a mexer no software e dados.
- Usabilidade: Esforço necessário para se aprender, operar, preparar entrada de dados e interpretar a saída de um programa;
- Facilidade de manutenção: Esforço necessário para localizar e corrigir erros em um programa (Definição muito limitada);
- Flexibilidade: Esforço necessário para modificar um programa em ação;
- Testabilidade: Esforço necessário para testar um programa garantindo que ele desempenha a função destinada;
- Portabilidade: Esforço necessário para transferir um programa de ambiente hardware e/ou software para outro;
- Reusabilidade: Quanto de um programa ou partes dele pode ser reutilizado para outras aplicações- relacionado ao empacotamento e o escopo das funções que ele executa;
- Interoperabilidade: Esforço necessário para integrar um sistema a outro.

Desenvolver essas medidas diretas desse fator de qualidade é difícil e em alguns casos impossível. Medidas essas definida por MacCall (1977), na realidade podem ser medidas

indiretamente, no entanto avaliar a qualidade de uma aplicação utilizando esses fatores possibilita uma solida indicação de qualidade de um software. (PRESSMAN, 2011).

2.5.3 Fatores de qualidade ISO 9126

A ISO 9126 foi desenvolvida com o intuito de identificar atributos fundamentais da qualidade para software de computador. Esse padrão tem seis atributos fundamentais de qualidade:

- **Funcionabilidade:** Grau que o software satisfaz as necessidades declarada pelos seguintes subatributos; adequabilidade, exatidão, interoperabilidade, conformidade e segurança;
- **Confiabilidade:** Quanto tempo que o software fica disponível para uso como indica os seguintes subatributos: maturidade, tolerância a falhas, facilidade de recuperação.
- **Usabilidade:** Grau de facilidade de utilização do software como indicado no subatributos: facilidade de compreensão, facilidade de aprendizado, operabilidade.
- **Eficiência:** Grau de otimização de uso do software, dos recursos do sistema como indica os seguintes subatributos: comportamento em relação ao tempo, comportamento em relação aos recursos.
- **Facilidade de manutenção:** Facilidade de correção no software como indicado no subatributos: facilidade de análise, facilidade de realização de mudanças, estabilidade, testabilidade.
- **Portabilidade:** Facilidade que o software pode ser transporte de um ambiente a outro como indicado no subatributos: adaptabilidade, facilidade de instalação, conformidade, facilidade de substituição. (PRESSMAN, 2011).

Pressman (2011), afirma que: “A ISO 9126 assim como outros fatores mencionado anteriormente, não levam necessariamente a medida direta, no entanto eles fornece uma base razoável para medidas indiretas e uma excelente lista de verificação para avaliar a qualidade do sistema.”

2.6 Qualidade no processo de desenvolvimento de software

Pádua (2009) define qualidade como: “Grau de conformidade de um sistema, componente ou processo com os respectivos requisitos, ou , alternativamente , como grau de

conformidade de um sistema, componente ou processo com as necessidades e expectativas de clientes ou usuários ...qualidade é definida por uma coleção de atributos; funcionalidade, confiabilidade, satisfação do usuário e desempenho são aspectos importantes.”

Ele ainda nos mostra que qualidade na produção de software implicam atitude preventiva no surgimento de problemas da qualidade deste produto. (PADUA,2009)

Sugere alguns mecanismos de prevenção para reduzir defeitos injetados ao longo do projeto, fazendo com que assim diminua quantidade de defeitos que serão removidos posteriormente, seja eles no final do projeto ou durante a vida útil do produto.

Segundo Sommerville (2009), afirma que: “A suposição fundamental da gerencia de qualidade é que a qualidade do processo de desenvolvimento afeta diretamente a qualidade dos produtos entregues.” Ou seja, esta direcionada ao processo de produção. Como software é projetado e não manufatura o seu desenvolvimento é um processo mais criativo do que mecânico. Sendo assim a influência de habilidades e experiências é significativa. Novidade de uma aplicação ou pressão comercial para um release antecipado de produto também afeta a qualidade independente do seu processo.

Portanto segundo Sommerville (2009), “softwares o relacionamento entre qualidade de processo e qualidade de produto é mais complexa.” É difícil medir seus atributos de qualidade mesmo após o uso de longo período do software, conseqüentemente, difícil dizer que as características do processo influenciam esses atributos. Por causa do projeto e da criatividade não tem como prever como as mudanças influenciaram na qualidade do produto.

O gerenciamento de qualidade de processo e o aprimoramento certamente conduz a um software com poucos defeitos. Este gerenciamento envolve:

1. Definir padrão de processo, “como” e “quando” conduzir revisão;
2. Monitorar o processo de desenvolvimento, para assegurar que estão seguindo os padrões;
3. Relatar o processo do software tanto para gerencia de projeto quanto pro comparador do software.

3 METODOLOGIA

Pode ser definido o método como um conjunto de etapas e instrumentos onde o pesquisador direciona seu projeto com critérios e caráter científico onde pode alcançar dados que suportam ou não a teoria inicial. (PRAÇA,2015).

Praça (2015) nos afirma que:” Metodologia científica é capaz de proporcionar uma compreensão e análise do mundo através da construção do conhecimento.”

3.1 Procedimentos metodológicos

Para a realização desse artigo, foram realizadas pesquisas bibliográficas compostas de dados obtidos através de livros e artigos relacionados ao tema em questão. Seu objetivo foi de levantar estudos disponível com a finalidade de avaliar, produzir e explicar o tema abordado.

4 RESULTADOS E DISCUSSÃO

A qualidade de um produto deve ser analisada em todo o seu processo de fabricação, um produto de software não é diferente. Baseado nas pesquisas feitas para obter esse trabalho analisamos que todas as etapas de desenvolvimento de um software são importantes para não ocorrer erros, desperdício de dinheiro e tempo e vida útil do produto.

Software é um produto complexo, onde esse assunto está longe de ser encerrado, pois profissionais da área estão constantemente avançando conforme a globalização no que diz respeito a conceitos, métodos e processos para se ter um produto de software de qualidade.

No entanto analisamos que os requisitos básicos para um bom produto software está relacionado com um bom gerenciamento na qualidade de processo e qualidade de produto.

Qualidade de processo podemos organizar melhor o projeto assim otimizando tempo, é nessa área que está a responsabilidade da definição da metodologia e ciclo de vida de um software e a uma boa comunicação entre a equipe de desenvolvedores.

Qualidade de produto garante qualidade durante todo o processo de desenvolvimento, é onde entra os testes feitos em cada ciclo de criação. Estes testes são importantes para evitar problemas e quando feito desde o início o projeto e feito junto com os clientes diminui os erros, fazendo assim com que diminua os erros durante a criação.

Os benefícios que as empresas tem ao dar atenção na área da qualidade do software são inúmeras vão além de financeiros. Por isso ele é benéfico e importante nos dias de hoje, evitando que os usuários sofram falhas, empresas fornecendo produtos de qualidade. Entre outros benefícios podemos citar: economia, impede emergências corporativas, inspira confiança do cliente, mantém nível de experiência do usuário, traz mais lucro e aumenta a satisfação do cliente.

5 CONCLUSÃO

Este artigo teve como finalidade de mostrar a importância da qualidade em qualquer tipo de serviço, especificamente no produto de software. Mostrando de forma sintetizada a estruturação e os pontos mais importante para um produto de qualidade.

Pode se observar no decorrer da pesquisa que não tem uma forma certa de não se cometer erros durante o desenvolvimento de um software, mas que sim pode ser evitado se bem planejados e sempre averiguando etapa por etapa.

REFERÊNCIAS

ARAÚJO, Renata Bezerra e Silva; CHALEGRE, Virgínia Carvalho. Qualidade de Produtos de Software, 2011.

DUTRA, Frederico Giffoni de Carvalho. Modelos e etapas para a gestão da informação: uma revisão sistemática de literatura. 2019. Tese (Doutorando) - Universidade Federal de Minas Gerais, Belo Horizonte, 2019.

LIZARDO, Celina; RIBEIRO, Paulo. A IMPORTÂNCIA DA GESTÃO DA QUALIDADE E APLICAÇÃO DAS SUAS FERRAMENTAS NA LOGÍSTICA COM VISTA À SATISFAÇÃO DOS CLIENTES, 2020

PADUA, Wilson De Paula. Engenharia de Software, fundamentos, métodos e padrões, 2009.

PRAÇA, Fabíola Silva Garcia. METODOLOGIA DA PESQUISA CIENTÍFICA: ORGANIZAÇÃO ESTRUTURAL E OS DESAFIOS PARA REDIGIR O TRABALHO DE CONCLUSÃO. Revista Eletrônica “Diálogos Acadêmicos”. Disponível em <<http://uniesp.edu.br/sites/biblioteca/revistas/20170627112856.pdf>>. Acesso em: 31/08/2021

PRESSMAN, Roger S. Engenharia de Software, uma abordagem profissional. [S. l.: s. n.], 2011.

SOMMERVILLE, Ian. Engenharia de Software, 2009.

ZANIN, Aline; MORAIS, Izabelly Soares. Engenharia de software, 2017.