

**CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“Dr. THOMAZ NOVELINO”**

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

JOÃO VITOR DE PAULA PEREIRA

GARANTIA DA QUALIDADE

Como pode contribuir com a qualidade e o sucesso dos sistemas
oferecidos aos usuários

FRANCA/SP

2022

JOÃO VITOR DE PAULA PEREIRA

GARANTIA DA QUALIDADE:

como pode contribuir com a qualidade e o sucesso dos sistemas
oferecidos aos usuários

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Carlos Eduardo de França Roland

FRANCA/SP

2022

JOÃO VITOR DE PAULA PEREIRA

GARANTIA DA QUALIDADE:

como pode contribuir com a qualidade e o sucesso dos sistemas
oferecidos aos usuários

Trabalho de Graduação apresentado à
Faculdade de Tecnologia de Franca – “Dr.
Thomaz Novelino”, como parte dos
requisitos obrigatórios para obtenção do
título de Tecnólogo em Análise e
Desenvolvimento de Sistemas.

Trabalho avaliado e aprovado pela seguinte Banca Examinadora:

Orientador(a) : _____

Nome..... : Prof Me Carlos Eduardo de França Roland

Instituição : Faculdade de Tecnologia de Franca – “Dr. Thomaz Novelino”

Examinador(a) 1 : _____

Nome..... : Prof. Esp. ALEXANDRE GOMES DA SILVA

Instituição : Faculdade de Tecnologia de Franca – “Dr. Thomaz Novelino”

Examinador(a) 2 : _____

Nome..... : Prof. Esp. DAVI MAXMILLAN SILVA

Instituição : Faculdade de Tecnologia de Franca – “Dr. Thomaz Novelino”

Franca, 3 de dezembro de 2022.

AGRADECIMENTO

Agradeço à minha família e amigos, principalmente meus pais por sempre me darem apoio e me incentivarem a estudar, por acreditarem em mim e no meu potencial, e registrar meu agradecimento aos meus colegas de trabalho que todos os dias me apoiam, me ensinam e certamente contribuíram muito na minha trajetória.

Dedico o presente Trabalho de Graduação aos meus amigos e familiares, em especial à minha mãe Alessandra, meu pai Ronaldo e meu irmão Alexandre.

Tudo aquilo que o homem ignora, não existe para ele. Por isso o universo de cada um, se resume no tamanho de seu saber.

Albert Einstein

RESUMO

Este trabalho tem como premissa apresentar a área que trata da qualidade dos softwares e os especialistas que nela atuam, visando chamar atenção para sua importância e relevância dentro de um ambiente de desenvolvimento de sistemas, e prover conhecimento para quem se interesse. Nas últimas décadas, a área de garantia da qualidade de software cresceu fortemente no mercado. Isso se deve ao fato de que a qualidade deixou de ser um diferencial que as empresas podem oferecer, e passou a ser um pré-requisito de qualquer produto digital disponibilizado aos usuários. Por essa razão os profissionais especializados nesse ramo são cada vez mais demandados nas organizações, mas ainda há pouco interesse por parte dos que estão ingressando no mercado da tecnologia. Os garantidores da qualidade de software são os responsáveis por verificar o cumprimento de normas e métodos específicos em todo o processo de desenvolvimento e operação do produto, e por propor melhorias nos processos. Devem atuar em cada uma das etapas do desenvolvimento de um software, desde a fase inicial do levantamento de requisitos, até a implantação da aplicação. Além disso, o Garantidor da Qualidade também é responsável por validar se o que foi desenvolvido está de acordo com os requisitos especificados e se está funcionando da maneira adequada, ou seja, testar o software. Essa é a principal atividade que esses especialistas executam, que visa prevenir que erros ocorram em produção, minimizando prejuízos e insatisfações de clientes. Os testes de software são bastante complexos, nos quais existem diversos níveis, técnicas e tipos. Há ainda uma grande quantidade de tecnologias que apoiam os testes, que podem ser realizados de forma manual ou automatizada, dependendo do contexto. O trabalho foi realizado com base nos conhecimentos obtidos através da vivência na área pelo autor, consultas com demais profissionais experientes, além de pesquisas bibliográficas principalmente em materiais de estudo voltados para certificações. Pôde-se concluir que apesar de sua relevância e importância, muitos profissionais da tecnologia não têm muita noção do trabalho que é desempenhado, e em algumas situações, acreditam que a presença da área não resulta em ganhos significativos e que podem até ser irrelevantes. De forma geral, a garantia da qualidade é muito bem-vinda na grande parte dos contextos organizacionais, e a tendência é que a área continue a crescer contribuindo para que softwares cada vez melhores estejam disponíveis no mercado.

Palavras-chave: Desenvolvimento. Garantia da Qualidade. Software. Tecnologia. Testes.

ABSTRACT

The premise of this work is to present the area that deals with software quality and the specialists who work in it, aiming to draw attention to its importance and relevance within a systems development environment, and provide knowledge to those who are interested. In recent decades, the area of software quality assurance has grown strongly in the market. This is due to the fact that quality is no longer a differential that companies can offer, and has become a prerequisite for any digital product made available to users. For this reason, professionals specializing in this field are increasingly in demand in organizations, but there is still little interest on the part of those entering the technology market. Software quality guarantors are responsible for verifying compliance with specific standards and methods throughout the product development and operation process, and for proposing process improvements. They must act in each of the stages of software development, from the initial phase of requirements gathering to application deployment. In addition, the Quality Assurance is also responsible for validating whether what was developed is in accordance with the specified requirements and is working properly, that is, testing the software. This is the main activity that these specialists perform, which aims to prevent errors from occurring in production, minimizing losses and customer dissatisfaction. Software testing is quite complex, in which there are several levels, techniques and types. There is also a large amount of technologies that support the tests, which can be performed manually or automatically, depending on the context. The work was carried out based on the knowledge obtained through the experience in the area by the author, consultations with other experienced professionals, in addition to bibliographic research mainly in study materials aimed at certifications. It could be concluded that despite its relevance and importance, many technology professionals are not very aware of the work that is performed, and in some situations, they believe that the presence of the area does not result in significant gains and that they may even be irrelevant. In general, quality assurance is very welcome in most organizational contexts, and the trend is for the area to continue to grow, contributing to increasingly better software being available on the market.

Keywords: *Development. Quality Assurance. Software. Technology. Testing.*

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Ciclo de trabalho do QA | 16 |
| Figura 2 – Custo para corrigir uma falha | 20 |
| Figura 3 – Fundamentos do teste de software | 27 |
| Figura 4 – Níveis de teste no ciclo de vida | 30 |

LISTA DE QUADROS

| | |
|--|----|
| Quadro 1 – Testes manuais x automáticos | 34 |
|--|----|

LISTA DE SIGLAS

API – Application Programming Interface

BDD – Behavior Driven Development

CVDS – Ciclo de Vida do Desenvolvimento de Software

KPI – Key Performance Indicators

QA – Quality Assurance

UI – User Interface

UX – User Experience

SUMÁRIO

| | |
|---|-----------|
| 1 INTRODUÇÃO | 13 |
| 2 O QUE É UM GARANTIDOR DA QUALIDADE | 15 |
| 3 A IMPORTÂNCIA DO QA | 16 |
| 4 CONHECIMENTOS QUE UM QA DEVE POSSUIR | 18 |
| 5 CUSTO DA FALHA | 20 |
| 6 DESAFIOS DE SE IMPLEMENTAR POLÍTICA DE QUALIDADE | 22 |
| 6.1 CONFLITOS ENTRE QUALIDADE E PRODUÇÃO | 23 |
| 6.2 GARANTIA DA QUALIDADE E TESTE | 24 |
| 7 TESTES DE SOFTWARE | 25 |
| 7.1 OBJETIVOS DO TESTE DE SOFTWARE | 25 |
| 7.2 A VISÃO E O PERFIL DO TESTADOR | 26 |
| 7.3 FUNDAMENTOS DO TESTE DE SOFTWARE | 27 |
| 7.3.1 Níveis de Teste | 27 |
| 7.3.2 Técnicas de Teste | 29 |
| 7.3.3 Tipos de Teste | 30 |
| 7.4 TESTES MANUAIS | 32 |
| 7.5 TESTES AUTOMATIZADOS | 33 |
| 7.6 ERROS COMUNS NA ATIVIDADE DE TESTES | 34 |
| 8 A REALIDADE E O FUTURO DA ÁREA | 36 |
| CONSIDERAÇÕES FINAIS | 37 |
| REFERÊNCIAS | 38 |

1 INTRODUÇÃO

Cada vez mais as empresas vêm se preocupando em melhorar a qualidade de seus softwares por diversos motivos, como oferecer maior segurança, evitando perdas financeiras, bloqueios na produção, sistemas apresentando erros operacionais, o que conseqüentemente melhora a satisfação dos clientes de software e resultando num melhor posicionamento da empresa fornecedora do produto no mercado.

Em consequência do aumento da busca pela qualidade, a área de *Quality Assurance* (QA - Garantia da Qualidade em português) vem crescendo a cada dia, e profissionais na área vêm sendo cada vez mais demandados e tendo ainda mais visibilidade.

No entanto, ainda sim poucas pessoas têm conhecimento do que de fato um QA faz e poucas se interessam pela área, principalmente quem está em formação, e por isso muitos profissionais que trabalham com tecnologia passam por uma transição para a carreira de QA.

O objetivo deste trabalho é demonstrar a atuação do QA dentro das organizações, quais são os desafios, as atividades desempenhadas e acima de tudo, a sua importância, que nem sempre as pessoas enxergam e dão valor. Portanto a ideia é despertar interesse na área, principalmente de estudantes que ainda estão na faculdade se preparando para ingressar no mundo da tecnologia e estão decidindo qual caminho seguir.

A metodologia utilizada foi exploratória através de pesquisas bibliográficas, com base em conteúdo de artigos, vídeos, materiais de estudo de certificações na área e pelo conhecimento previamente construído e experiência profissional do autor.

Os Capítulos 2 ao 6 abordam conceitos e fundamentos sobre Garantia da Qualidade, demonstrando o que é a disciplina, sua importância, os benefícios que pode trazer, qual perfil técnico os profissionais da área devem desenvolver, além de outras questões presentes no cotidiano do exercício profissional.

O capítulo 7 apresenta as atividades de testes de software, demonstrando o que de fato são, como devem ser usados na prática, as características relacionadas aos níveis, técnicas e tipos de testes existentes, como cada um deles funciona e de que forma se encaixam no ambiente de desenvolvimento.

Por fim, o Capítulo 8 trata de como é de fato a área de qualidade de software, o que se pode esperar para quem deseja ingressar e como esse ramo está em constante crescimento.

A relevância deste trabalho se deve ao fato de apresentar essa importante área da tecnologia, que as pessoas conhecem, mas não sabem de fato como funciona, e mostrar o quão expressivo e valioso é o trabalho de um *Quality Assurance*.

2 O QUE É UM GARANTIDOR DA QUALIDADE

Qualidade e eficiência são pontos importantes para o bom funcionamento das equipes de TI. Com o grande crescimento dos meios digitais, esse campo tornou-se estratégico e tem grande impacto nos negócios. O fato é que para um produto, especialmente os digitais, se manter no mercado a qualidade não é mais um diferencial e sim um pré-requisito, e isso cria a necessidade de especialistas que trabalhem totalmente focados na melhoria da qualidade.

Garantia da Qualidade é o termo traduzido do inglês *Quality Assurance*, e por isso os especialistas neste campo são geralmente chamados de QA. Diz respeito a profissionais ou equipes que têm a capacidade de garantir a qualidade no desenvolvimento de produtos ou serviços. Em suma, o trabalho é verificar o cumprimento de normas e métodos específicos em todo o processo de operação (SANTOS, 2019).

O plano de atuação do QA é identificar os ganhos operacionais com base na documentação e comunicação entre as demais áreas envolvidas. Seu objetivo é identificar *bugs* e problemas que podem afetar a qualidade final do que está em desenvolvimento, mas não só isso. O QA também tem a capacidade de levantar oportunidades, que podem ser uma melhoria direta no serviço ou aperfeiçoamento dos processos (SANTOS, 2019).

São pessoas que devem ter perfis analíticos e focados, conhecimento das atividades e métodos relacionados. Nem sempre a capacitação técnica desses profissionais é inteiramente na área de qualidade. É muito comum que alguns especialistas estejam relacionados ao desenvolvimento de software, mas isso não significa que todos os desenvolvedores possam assumir o papel de QA por estes serem perfis diferentes.

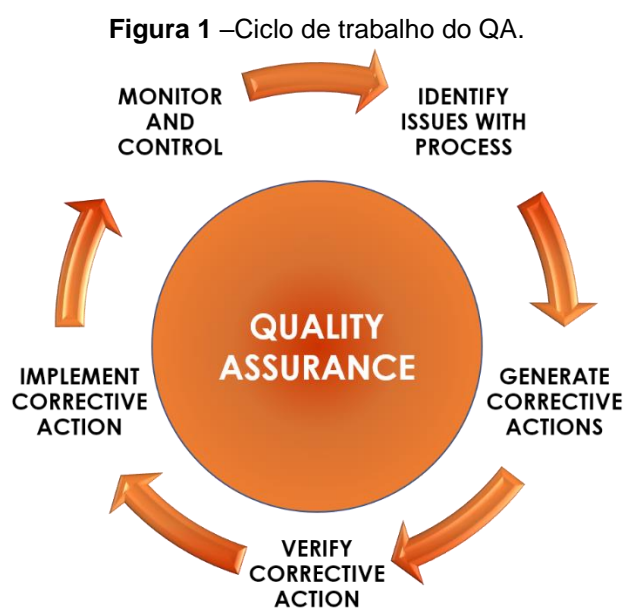
O profissional de QA tem papel relevante na equipe de desenvolvimento porque está interessado na qualidade do produto. E em uma equipe ágil é preciso resolver impedimentos no desenvolvimento rapidamente. Portanto, cabe a esse especialista testar o trabalho de todos, relatar os problemas que podem ocorrer e sugerir melhorias.

3 A IMPORTÂNCIA DO QA

O ritmo do mercado de software é muito dinâmico. As empresas devem ser ágeis e se adaptar aos novos requisitos dos clientes. Isso significa que é preciso ser flexível e sempre inovador.

Isso torna a TI uma importante ferramenta estratégica, pois permite identificar oportunidades no mercado e otimizar o trabalho em outras áreas da empresa. Porém, sua função não se baseia apenas no senso de gestão de especialistas.

A Garantia de Qualidade ajuda a garantir a conformidade com os processos estabelecidos, ao mesmo tempo em que aumenta a eficiência ao lidar com novos projetos. Ao participar de reuniões para tratar de novos produtos e serviços, os membros do time de QA podem resolver problemas que só surgiriam durante a entrega do produto (SANTOS, 2019). Na Figura 1 é mostrado o ciclo de trabalho do QA.



Fonte: Sofeast (2022)

Na era do desenvolvimento ágil, é arriscado desenvolver um projeto de TI sem o suporte da Garantia de Qualidade. Por exemplo, diferentemente do modelo em cascata que tem foco em outros processos, os métodos ágeis implementam testes de qualidade em toda a criação da solução (SANTOS, 2019).

O QA deve acompanhar o projeto desde a visão geral de projeto até a programação e a implementação no cliente. Eles são especialistas em mitigar os riscos operacionais e examinar minuciosamente as questões que impactam a entrega.

Diante disso, pode-se observar o quão importante são esses profissionais. Dentre os principais benefícios que se obtém, podem-se destacar (SANTOS, 2019):

- Aumento da satisfação do cliente;
- Custos operacionais reduzidos;
- Otimização do trabalho diário dos profissionais de TI.
- Redução do retrabalho;
- Eliminação de gargalos;
- Aumento da previsibilidade dos projetos.

4 CONHECIMENTOS QUE UM QA DEVE POSSUIR

O especialista em qualidade de software precisa ter uma abrangência de conhecimento ampla e, portanto, deve ser multidisciplinar. Isso acontece pois o QA pode estar envolvido em todas as etapas de um projeto e fazer os mais diversos tipos de teste e validações.

O mais importante de tudo, é entender a atividade de teste como um todo, que é a principal realizada. Para isso é necessário saber planejar, elaborar e executar casos de testes, escrever cenários em formato Desenvolvimento Guiado por Comportamento (em tradução livre do termo original em inglês *Behavior Driven Development* - BDD), documentar e reportar defeitos, além de compreender os fundamentos do teste de software. Até então, para isso, não é necessário dominar uma ferramenta, ainda que existam várias que apoiam o gerenciamento dos testes.

Um dos elementos de sistemas de informação mais utilizados por QAs são os Gerenciadores de Bancos de Dados, seja qual for adotado pela empresa de desenvolvimento. Muitas das validações em testes que precisam ser feitas necessitam realizar algum tipo de consulta, alterações manuais diretamente numa tabela do banco, ou popular tabelas dentre outras operações.

Também é fundamental ter um conhecimento prévio em programação, principalmente em lógica e, especialmente, na linguagem de desenvolvimento adotada pela empresa que o profissional atua. Esse domínio não precisa ser avançado a ponto de ser capaz de desenvolver código, mas um nível a partir do qual seja possível ler e entender o programa, para ter mais autonomia e saber reportar erros de forma mais assertiva, permitindo maior poder de argumentação.

Ainda na parte de testes de software, ferramentas para testar Interfaces de Programação de Aplicações (do termo em inglês *Application Programming Interfaces* - API), tais como Postman ou Insomnia são indispensáveis para o QA, pois independentemente da empresa, é muito provável que seja necessário fazer testes dessas funcionalidades. Via de regra essas são ferramentas fáceis de se aprender a usar.

Há também necessidade de conhecimento dos fundamentos da web, como o protocolo HTTP, HTML, CSS e DOM por exemplo, pois são eles que fornecem a base da maioria dos sistemas que o profissional deverá trabalhar (SILVA, 2022).

A capacidade de ler e entender *logs* também é uma habilidade útil no dia a dia, pois eles dizem muito sobre o que está acontecendo no sistema. Conseguir-se analisar a fundo todas as ações que estão sendo feitas pelos componentes do sistema e assim é possível localizar e entender por que determinadas falhas ocorrem (SILVA, 2022).

Conhecimentos que também podem ser interessantes, são em Experiência do Usuário (UX do termo em inglês *User Experience*) e Interface do Usuário (UI do termo *User Interface*), porque em algum momento pode ser necessário realizar testes de usabilidade, que envolvem validar se o produto implementa as melhores práticas de Interação Humano-Computador (IHC), se o uso é intuitivo, fácil, e se a interface visual é adequada ao usuário do produto. Sem esse tipo de conhecimento, o QA pode deixar de considerar defeitos relevantes e permitir que cheguem ao cliente. Além de, como foi citado anteriormente, ajudar na argumentação ao reportar falhas identificadas.

Ainda, para quem pretende atuar com automação de testes, é preciso conhecer algum *framework* específico para isso, e existem diversos deles, tais como Selenium, Cypress, Robot, Appium, Capybara dentre outros. Nesse caso deve-se dominar uma ou mais dessas ferramentas, pois normalmente uma única delas não é capaz de testar ou ser a mais adequada para todos os contextos. Algumas são melhores para automatizar testes em sistemas web, outras para *desktop* além das mais indicadas para aplicações *mobile*.

Estendendo para além dos testes, o QA precisa ter noções de análise de negócios, pois como estão diretamente envolvidos com o produto, terão condições de oferecer sugestões sobre o rumo do negócio, sugerir novos caminhos para antigos problemas, dentre outras contribuições (SILVA, 2022).

O entendimento sobre escrita de documentações e levantamento de requisitos também é interessante, já que são itens que devem passar pelas mãos do profissional da qualidade e este deve ser capaz de compreender e de auditar seus conteúdos antes de seguir para as próximas etapas de testes.

Como é possível observar, o trabalho do *Quality Assurance* não é simples. Existem muitas atividades presentes na rotina do profissional e uma grande quantidade de habilidades são demandadas. Apesar disso, não significa que esses especialistas precisam de fato dominar tudo. Na maioria dos casos um conhecimento superficial é o bastante, sem contar que em cada empresa, os QAs possuem atividades diferentes em cenários distintos.

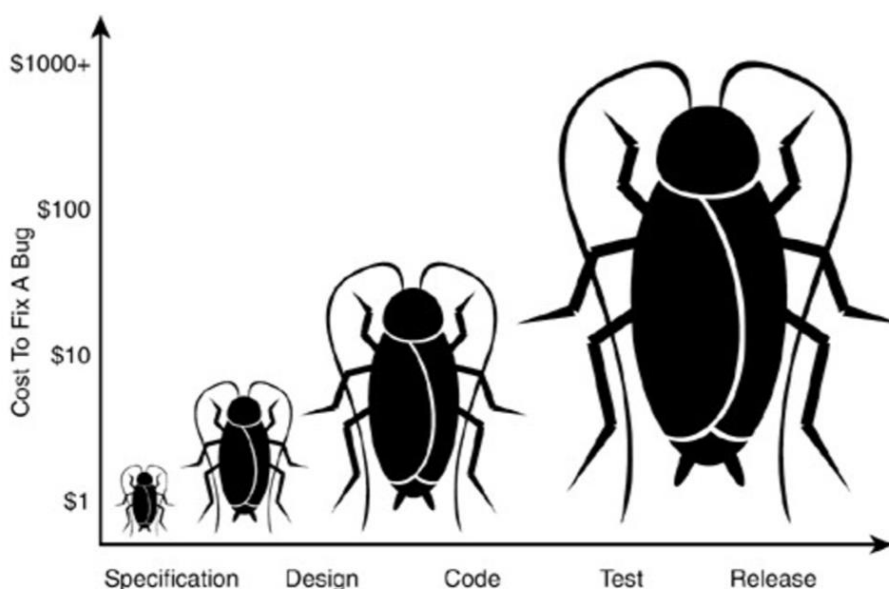
5 CUSTO DA FALHA

Durante o Ciclo de Vida do Desenvolvimento de Software (CVDS), quanto mais tarde uma falha é encontrada, maior o custo para corrigi-la. Portanto, isso mostra que o corrigir *bugs* é mais caro do que criar mecanismos para evitá-los (RODRIGUES, 2017).

Fazendo uma analogia com a medicina, curar uma doença é sempre mais custoso e problemático do que prevenir, e dentro da área de tecnologia acontece o mesmo, quanto mais cedo a falha for localizada, mais fácil e menos impactante é sua correção (RODRIGUES, 2017).

A Figura 2 ilustra o aumento da despesa com correções em softwares no decorrer do tempo.

Figura 2 – Custo para corrigir uma falha



Fonte: Rodrigues (2017)

Na fase de especificação, quando está sendo realizada a coleta de requisitos com o cliente, ou seja, tudo aquilo que o sistema precisa ter, caso uma falha seja encontrada, onde pode haver uma especificação de requisitos conflitante, basta alterar um parágrafo, uma frase, ou mesmo uma palavra.

Se um problema é encontrado na fase de codificação, já terá um custo de correção consideravelmente maior em relação à especificação, pois vários pontos precisam ser trabalhados. O código já desenvolvido deve ser alterado, os testes de

unidade precisam de atualização, pode haver necessidade de alterar os testes de integração entre outras coisas (FARIAS, 2019).

Considerando um nível além, na fase de testes sistêmicos propriamente dita, onde atua o QA, se um *bug* for encontrado, se torna ainda mais trabalhoso corrigir, pois juntamente com o desenvolvedor, será necessário analisar a falha para em seguida ser implementada a correção, que pode ter sido gerada durante o desenvolvimento, arquitetura, projeto ou na especificação, e não é fácil o trabalho de buscar a causa raiz do problema.

Por fim, se determinada falha só é encontrada pelo próprio usuário quando o sistema já se encontra em produção, além de todos os gastos e dificuldades mencionados nas etapas anteriores, é preciso lançar uma nova *release* o que, geralmente, é associado à insatisfação do cliente que pode gerar prejuízos ainda maiores, visto que dependendo da gravidade da falha, ou a incidência constante de erros, os usuários podem deixar de usar o sistema, rompendo contratos.

Por essas evidências fica ainda mais perceptível a responsabilidade que um garantidor da qualidade tem e o motivo pelo qual as organizações precisam estar atentas com a questão da qualidade daquilo que oferecem ao seu consumidor.

6 DESAFIOS DE SE IMPLEMENTAR POLÍTICA DE QUALIDADE

Implementar uma política de qualidade em empresas de desenvolvimento de software, vai além de montar uma equipe de QA e esperar que as coisas aconteçam, podendo ser um processo longo e desafiador.

Apesar de todos os benefícios que uma equipe voltada à qualidade pode alcançar, muitas empresas ainda não possuem. Isso acontece por vários motivos, seja por questões culturais ou porque o departamento de TI das organizações não está comprometido em garantir a qualidade de seus produtos digitais, porque consideram apenas como despesa e não investimento.

Aspectos financeiros, geralmente, são os maiores impeditivos e costumam ser complicados. Claro que se for analisado com mais detalhes, garantir a qualidade dos produtos é de fato um investimento, que poderá ser colhido a longo prazo, mas apesar disso, muitos gestores ainda não acreditam e não são capazes de perceber que cuidar da qualidade retorna os resultados no futuro (LIBÂNIO, 2020).

Não se pode deixar de notar que, como todo investimento, é preciso haver um aporte inicial na contratação de especialistas na área, adquirir licenças de ferramentas, mudar processos, capacitar pessoas e tudo isso gera um custo que, principalmente no caso empresas de pequeno porte, são difíceis de viabilizar.

A consideração estratégica da qualidade torna-se mais difícil em organizações que não possuem processos bem definidos ou burocratizam o desenvolvimento. No caso das que burocratizam, ainda é habitual encontrar equipes tomando decisões simplesmente porque uma metodologia ágil específica diz que deveriam, ignorando o contexto geral do negócio e o operacional do desenvolvimento (LIBÂNIO, 2020).

Mesmo que existam especialistas na função de controle de qualidade em sua equipe, é comum que a qualidade seja reduzida a apenas uma etapa do processo. Os testes (manuais ou automatizados) geralmente são feitos apenas no final do processo de desenvolvimento.

Em sua maioria, gerentes e gestores veem as questões de qualidade como um obstáculo para colocar seus produtos digitais em produção. E essas lideranças acabam delegando a responsabilidade da qualidade aos desenvolvedores, sem fornecer as ferramentas e técnicas adequadas para que os profissionais possam usar parte de seu tempo para a verificação da qualidade da entrega (LIBÂNIO, 2020).

Aprofundando na questão do tempo, líderes de projetos costumam exigir que os desenvolvedores entreguem código já com os testes feitos (testes unitários), mas não alocam no cronograma o tempo para executar essa atividade. Isso ocorre principalmente porque quanto mais tempo os desenvolvedores alocarem para criar testes, menos entregas ocorrem.

Apesar de tais considerações, uma boa maneira de se dar o primeiro passo em direção à qualidade é entender onde é possível obter mais resultados em menos tempo. Olhar para as métricas internas que indicam o surgimento de crises, como reclamações, e analisar quantas delas são causadas por problemas técnicos (LIBÂNIO, 2020).

Todas as empresas lidam com problemas sejam eles grandes ou pequenos. Por isso, entender a origem destes problemas é fundamental para visualizar o negócio como um todo e entender por onde começar.

É necessária uma compreensão mínima das dores do negócio para elencar processos e ações para se justificar como a qualidade pode ser estratégica. Este é um grande desafio, especialmente se os gestores não desenvolverem esta visão. Todavia, começar com uma análise mais aprofundada e direcionada dos Indicadores-chave de Performance (KPI do termo *Key Performance Indicator*) pode demonstrar que investir em qualidade é estratégico para o negócio da companhia (LIBÂNIO, 2020).

6.1 CONFLITOS ENTRE QUALIDADE E PRODUÇÃO

No dia a dia de uma empresa de software há conflitos entre a equipe de QA e a equipe de desenvolvimento, especialmente pelo fato de que os desenvolvedores se incomodam quando falhas são apontadas no seu trabalho. De fato, não é fácil ter seu trabalho questionado, mas é preciso entender que são questões que fazem parte de algo maior, que é a qualidade.

Isso eventualmente acontece pois estes são profissionais que têm perfis e visões bastante diferentes. Em casos em que, por exemplo, uma simples falha é localizada, mesmo que não impacte gravemente para o usuário final, para um QA é sempre algo relevante que precisa ser corrigido, e o desenvolvedor pode entender como algo de pouca relevância.

Esse tipo de atrito precisa ser resolvido o quanto antes caso esteja ocorrendo, pois isso acaba comprometendo a qualidade do produto final e conseqüentemente torna o ambiente desagradável para ambas as partes. Cabe ao gestor identificar essas situações e trabalhar para que isso não aconteça.

A ideia, é que cada um saiba executar sua função com maestria, mas também compreenda que em um ambiente organizacional, o trabalho é realizado com várias pessoas que dependem umas das outras e estão buscando um objetivo maior, portanto devem entender o papel de cada um e atuarem de fato como uma equipe.

6.2 GARANTIA DA QUALIDADE E TESTE

Um erro comum nessa área, é que as pessoas entendem que garantia da qualidade é a mesma coisa que teste de software, mas na realidade a atividade de testes é uma das diversas atribuições de um QA. Isso acontece pois de fato os testes normalmente são o que esses especialistas mais realizam no dia a dia das organizações.

O controle da qualidade engloba diversas atividades, incluindo os testes que contribuem para alcançar um nível adequado de qualidade. Os testes fazem parte do processo geral de desenvolvimento e manutenção de softwares. Neste caso, como a garantia da qualidade está relacionada com a execução adequada dos processos, então ela apoia o teste (ISTQB, 2019).

Via de regra, do ponto de vista do mercado, ambos os termos são habitualmente usados como uma coisa só, e não há problema algum, pois são completamente relacionadas e qualquer um consegue compreender. Mas é relevante entender que existe essa diferença e conceitualmente é errado considerá-los iguais.

7 TESTES DE SOFTWARE

Os testes de software consistem na verificação realizada em um sistema ou parte dele, garantindo que uma determinada entrada produza a saída esperada. É uma forma de validar a qualidade do código de uma aplicação, a fim de reduzir os riscos de falhas quando o sistema estiver em operação no cliente (FARIAS. 2019).

Apesar do que foi mencionado anteriormente sobre os testes não serem a única atribuição de um QA, essa é a principal atividade realizada por esses profissionais em qualquer empresa, e podem ser realizados tanto de forma manual, como também programando testes automáticos no caso de testes de aplicações.

É importante destacar que os testes vão muito além da execução, ou seja, executar e analisar os resultados, esta é uma das atividades presentes no processo de testes. Antes da execução em si, existe toda uma análise e planejamento do que será testado, além da modelagem e implementação dos testes, relatórios de progresso, e após a execução, também há a demonstração de resultados e avaliação da qualidade (MULLER, 2020).

Há também ainda outra visão equivocada a respeito dos testes, em que se acredita que eles se concentram inteiramente na verificação de requisitos e das especificações. No entanto, embora o teste envolva verificar se o sistema atende os requisitos especificados, eles também verificam se o software atenderá às necessidades do usuário e stakeholders em seus ambientes operacionais (ISTQB, 2019).

7.1 OBJETIVOS DO TESTE DE SOFTWARE

O teste tem diversos objetivos, mas independente do projeto tudo se resume em garantir a qualidade do sistema. De acordo com ISTQB (2019, p. 13), testar um software tem os seguintes objetivos:

- Evitar defeitos, avaliar os produtos de trabalho, como requisitos, histórias de usuários, modelagem e código;
- Verificar se todos os requisitos especificados foram cumpridos;
- Verificar se o objeto de teste está completo e validar se funciona como os usuários e stakeholders esperam;
- Criar confiança no nível de qualidade do objeto de teste;

- Encontrar defeitos e falhas reduz o nível de risco de qualidade inadequada do software;
- Fornecer informações suficientes aos stakeholders para que tomem decisões especialmente em relação ao nível de qualidade do objeto de teste;
- Cumprir os requisitos ou normas contratuais, legais ou regulamentares, ou verificar a conformidade do objeto de teste com esses requisitos ou normas.

Para que todos os objetivos sejam cumpridos, também é válido mencionar que o testador deve estar presente durante todo o Ciclo de Vida do Desenvolvimento do Software, desde a fase de levantamento de requisitos, até a fase da implementação, pois os objetivos envolvem esses diversos momentos, e o profissional também consegue compreender todo o contexto do que está sendo desenvolvido para validar cenários ainda mais assertivos a fim de localizar falhas (FARIAS, 2019).

7.2 A VISÃO E O PERFIL DO TESTADOR

As principais características do testador são a postura crítica, detalhista, calculista, pessimista e criteriosa. É preciso ter uma visão voltada para a qualidade e ter empatia também, pois durante os testes é sempre preciso se colocar no lugar do usuário final. A visão crítica é um dos pontos mais importantes e consiste em entender que as coisas precisam estar funcionando perfeitamente, pois sabe que o cliente é exigente, e que se determinada rotina falhar pode causar prejuízo financeiro entre outras consequências (FARIAS, 2019).

O testador também deve ser um profissional curioso, aquele que procura erros nas combinações mais inesperadas que podem comprometer o sistema. É alguém que investiga os comportamentos do software além do que a documentação descreve, tentando compreender possibilidades de erros a fim de criar cenários de testes para descobrir os defeitos.

O pessimismo, apesar de normalmente ser considerado algo negativo, não é o caso para um testador, visto que esse profissional deve sempre pensar o pior cenário possível, já que seu principal objetivo é localizar falhas. Este comportamento permite elaborar contextos de testes ainda mais abrangentes e assertivos.

Além de tudo, deve ser uma pessoa comunicativa e deve usar sua capacidade de negociação e persuasão em favor da qualidade, sugerindo melhorias e soluções (SILVA, 2022). Nem sempre é uma tarefa fácil convencer um desenvolvedor de que a lógica usada deve ser melhorada para prevenir eventuais ocorrências de erros. É um

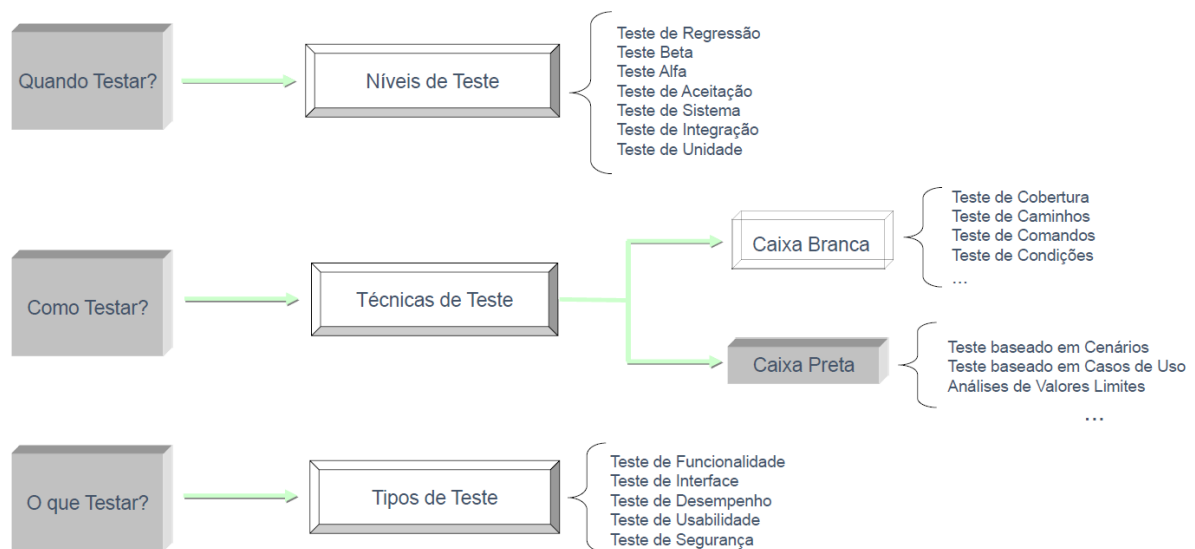
desafio, e é neste caso, especialmente, que o poder da comunicação pode fazer grande diferença.

7.3 FUNDAMENTOS DOS TESTES DE SOFTWARE

Os testes de software possuem três fundamentos que respondem às perguntas: quando, como, e o que testar? São conceitos básicos que qualquer profissional da área deve conhecer.

A Figura 3 apresenta como estão divididos os fundamentos e o que está presente em cada um deles.

Figura 3 – Fundamentos do teste de software.



Fonte: Farias (2019, p. 34)

7.3.1 Níveis de Testes

Os níveis de testes respondem à pergunta “quando testar?”. Existem 7 níveis que são realizados em diferentes momentos do CVDS (FARIAS, 2019), e são eles:

- **Teste de Unidade:** é o primeiro a ser realizado, pois ele não exige que se esteja em uma etapa avançada do sistema para poder ser aplicado. A partir do momento que é implementada a primeira classe, método ou função do sistema já é possível fazer um teste de unidade. Consiste em localizar falhas de funcionamento na menor parte de um software, funcionando independente do todo.

Normalmente são feitos pelos próprios programadores e são automatizados por meio de ferramentas como JUnit, PHPUnit entre outras. É importante que estejam sempre atualizados, coerentes com as regras de negócio elicitadas para o sistema, visto que estas estão em constante mudança.

- **Teste de Integração:** nesse ponto, os testes já não são feitos em um único método, e sim sobre um conjunto deles, validando a comunicação entre os componentes do sistema, focando o envio e o recebimento de dados e o foco é encontrar falhas nessas transações. Também são feitos por programadores durante o desenvolvimento e são automatizados utilizando as mesmas ferramentas usadas nos testes de unidade. São realizados antes do sistema ser concluído, ou seja, à medida que os componentes vão ficando prontos, os testes vão sendo desenvolvidos.
- **Teste de Sistema:** são testes realizados após a codificação estar concluída através da execução do sistema operando sobre o ponto de vista do usuário final, executando as funcionalidades em busca de falhas em relação aos objetivos originais. Aqui os testes já são planejados e executados pelo QA. Apesar de ser realizado simulando o usuário, é feito com a visão de um testador, pois o usuário não busca erros, ele quer apenas acessar o sistema e usá-lo para cumprir suas tarefas, enquanto o testador tem o intuito de encontrar falhas. Para realizar os testes, o responsável planeja e escreve cenários de teste coerentes com os requisitos especificados e faz as simulações desses cenários.
- **Teste de Aceitação:** também é executado em nível de sistema, mas desta vez são executados por um usuário ou um grupo restrito de usuários finais, que irão simular operações de rotina com intuito de verificar se o comportamento está de acordo com o solicitado, com o objetivo que o cliente determine se aceita ou não o sistema e ofereça *feedback* com sua experiência de uso.
- **Teste Alfa:** é semelhante ao teste de aceitação, porém, neste caso, é feito por uma quantidade maior de usuários, mas ainda um grupo pequeno. Tem como objetivo executar o sistema de forma não planejada e visa identificar erros não detectados até o momento, normalmente encontrados enquanto mais usuários finais utilizam o sistema de forma natural. Os Testes Alfa são feitos em um ambiente controlado, com representantes do time de programadores que acompanharão os testes para coletar as possíveis falhas a corrigir e melhorias a serem implementadas.

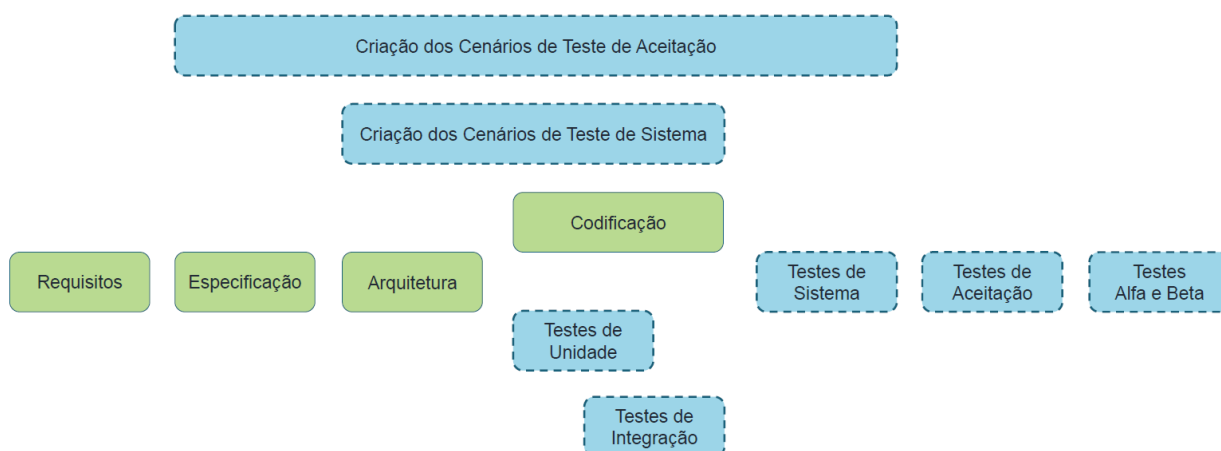
- **Teste Beta:** este teste também é semelhante ao Alfa e segue o mesmo princípio: deixar que os usuários testem o sistema de forma não planejada. No entanto, existem diferenças importantes, como a quantidade de usuários. Neste caso é aplicado por um número maior de pessoas que normalmente são desconhecidas e são ainda um subconjunto de todo o público-alvo. Por exemplo, de uma determinada localidade, que fale um idioma em específico ou que satisfaçam determinados critérios definidos pelo desenvolvedor do sistema. É como um pré-lançamento, mas dessa vez, os programadores não acompanham de perto fazendo o levantamento de erros. São os próprios usuários que reportam e podem ou não serem corrigidos.
- **Teste de Regressão:** não necessariamente é o último a ser realizado. Na verdade, ele pode acontecer a qualquer momento. Consiste em reexecutar testes que foram criados anteriormente, após alguma alteração ser realizada no sistema, com o objetivo de verificar se não houve impacto no que já funcionava e se tudo continua operando corretamente. São testes de extrema importância, visto que após uma correção ou implementação é muito comum a quebra de outras rotinas, e é exatamente por ser necessário o teste de regressão, que os testes automatizados fazem total diferença. Tendo sido desenvolvido uma única vez, pode ser executado inúmeras vezes com muito mais agilidade, demandando apenas atualizações recorrentes.

Na Figura 4 é possível visualizar onde cada nível de teste se encaixa dentro do CVDS.

7.3.2 Técnicas de Teste

As técnicas de teste respondem à pergunta “como testar?”. Existem 2 subconjuntos denominados teste de Caixa Branca, e teste de Caixa Preta que possuem diferentes técnicas dentro de cada um como descrito a seguir (FARIAS, 2019).

Figura 4 – Níveis de teste no ciclo de vida.



Fonte: Farias (2019, p. 100)

- **Caixa Branca:** também pode ser chamado de teste de Caixa Aberta, possui esse nome pois envolve técnicas que precisam avaliar o código desenvolvido ou codificar. Envolvem principalmente os testes de unidade, ou seja, teste de cobertura, caminhos, comandos, condições, funções, entre outros. Portanto normalmente ficam sob responsabilidade da equipe de desenvolvimento.
- **Caixa Preta:** são aqueles em que não é preciso analisar o interior do sistema, ou seja são testes baseados apenas em entradas e saídas, e envolvem testes baseado em cenários, casos de uso, análise de valores limite entre outros. Fazendo uma associação com os Níveis de Teste, os testes de Integração, Sistema, Aceitação, Alfa e Beta fazem parte dos testes Caixa Preta e neste caso, são de responsabilidade da equipe de testes ou do QA, usuários e até de desenvolvedores no caso de Integração.

7.3.3 Tipos de Teste

Os tipos de teste respondem à pergunta “o que testar?”, que dizem respeito a qual será o foco do teste. Existem 6 tipos principais, mas nem sempre todos são realizados nas empresas (FARIAS, 2019). São eles:

- **Teste de Funcionalidade:** tem como objetivo validar se as funcionalidades estão operando da forma correta, ou seja, quando se solicita uma ação no sistema, se ele realmente vai executá-la. Envolve níveis de teste e todos eles estão relacionados a testes de funcionalidade, desde os testes de Unidade, até o Alfa e Beta.
- **Teste de Desempenho:** neste caso o objetivo é validar se o desempenho do sistema em relação ao seu tempo de resposta para determinadas operações está de acordo com o que foi especificado, com o que é esperado e com a expectativa dos usuários. Por exemplo: ao processar um relatório em um sistema financeiro, espera-se que seja realizado em poucos segundos, e se levar muito mais do que isso, pode atrapalhar a produtividade do usuário e gerar insatisfação.
- **Teste de Usabilidade:** é um tipo de teste que tem como premissa validar aspectos que envolvem a experiência do usuário ao utilizar determinado sistema. Ou seja, o quão fácil é para o usuário utilizar, se a composição de cores da interface são agradáveis, se os menus são intuitivos, se o usuário consegue chegar à função que precisa executar sem dificuldade. Normalmente é pouco realizado, mas consiste em convidar pessoas que fazem parte do público-alvo ou o próprio cliente para fazer essa validação. É comum que o próprio profissional que atua em QA, valide esses aspectos com base em critérios de ergonomia, usabilidade e heurísticas enquanto realiza testes de funcionalidade.
- **Teste de Segurança:** este é um tipo de teste bastante crítico, trivial em softwares que armazenam e gerenciam dados sensíveis e informações sigilosas, tais como sistemas bancários, financeiros entre outros. Consiste em validar a proteção de um sistema contra invasões ou acessos não autorizados a informações, ou seja, se o tráfego das informações é criptografado e se não existem brechas para invasões. Para ser realizado, muitas vezes as empresas contratam *hackers* profissionais para tentarem invadir os sistemas e localizar possíveis vulnerabilidades, havendo empresas de consultoria que fazem esse tipo de trabalho. Qualquer aplicação pode, e deve, passar por esse tipo de teste, mas pelo fato de ter um alto custo, é comum que seja feito apenas quando realmente necessário.

- **Teste de Portabilidade:** são testes que visam validar o funcionamento do sistema em diferentes plataformas e dispositivos nos quais o sistema tem a proposta de funcionar, como por exemplo: se funciona nos sistemas operacionais Windows, Mac e Linux nos casos de aplicações *desktop* ou web, ou no Android e IOS nos casos de uso por *smartphones* e *tablets*. É um tipo de teste que sempre deve ser feito, pois pode gerar muitos transtornos e prejuízos se entrar em produção com algum tipo de falha nesse contexto. Um problema comum no desenvolvimento de sistemas web, é que pelo fato de existir uma grande variedade de *browsers*, os testes tendo sido realizados somente em um específico que a empresa utiliza como padrão, podem não funcionar em outros ambientes dos usuários.
- **Teste de Estresse:** consiste em levar o sistema a condições extremas de operação e validar o seu comportamento, verificando se ele é capaz de suportar uma grande carga de trabalho, como simular uma enorme quantidade de acessos simultâneos executando as principais funções, por exemplo. É muito confundido com os testes de desempenho, mas enquanto este submete o sistema a situações normais do dia a dia e mede o tempo de resposta, no teste de estresse são validadas situações extremas e atípicas para se certificar que o sistema continua a responder de forma aceitável.

7.4 TESTES MANUAIS

Os testes manuais significam testar uma aplicação manualmente por um ser humano, um especialista em garantia de qualidade, ou o profissional testador. Este executa os testes manuais e garante que o sistema esteja funcionando corretamente seguindo as condições descritas nos casos de testes (ANTUNES, 2021).

Nos testes manuais, o testador deverá avaliar o *design*, a funcionalidade e o desempenho do aplicativo verificando diversos elementos. Essa modalidade de teste é recomendada quando é preciso realizar testes exploratórios, de usabilidade e testes de aceite (ANTUNES, 2021).

São muito úteis quando não é possível implementar testes automáticos, pois realmente nem tudo é possível automatizar. Há também outros pontos positivos, o testador pode testar o software em condições semelhantes quando este estiver em

produção, pode identificar problemas relacionados à aparência visual do aplicativo e ajuda a descobrir problemas de usabilidade, itens que os automáticos não conseguem cobrir.

Outro ponto a se considerar, é que os testes manuais não exigem um alto investimento, já que não precisam de ferramentas caras ou habilidades de alto nível para serem executados. Mas vale lembrar que em certas ocasiões, como em testes de regressão, os automáticos acabam tendo vantagem inclusive na questão do custo (ANTUNES, 2021).

Entre os pontos negativos estão os fatos de serem bem mais lentos para se executar, e como existe uma interação totalmente humana, a probabilidade de *bugs* passarem despercebidos é alta.

7.5 TESTES AUTOMATIZADOS

Testes automatizados se refere a desenvolver testes programados para serem executados automaticamente e que comparam os resultados obtidos com os resultados esperados, pré-definidos, pelo testador. Também ajudam a saber se o software tem o desempenho previsto ou não.

Sua execução é feita com auxílio de ferramentas, *scripts* e programas, e são recomendados para realizar testes de carga, performance e o principal deles, o teste de regressão, visto que se programa uma única vez, e pode ser executado quantas vezes forem necessárias com agilidade (ANTUNES, 2021).

Os testes automatizados também são mais confiáveis por serem executados por programação, com chance de erro praticamente nula e sendo mais velozes que os testes manuais (FARIAS, 2019).

Ainda que tenham diversos pontos positivos, é preciso citar que existem alguns negativos também. São bem mais demorados para serem desenvolvidos, existem limitações em relação às ferramentas, pois para cada tipo e fase de teste é necessário um tipo de ferramenta e conhecimento sobre ela. Há ainda o aspecto do custo já que, dependendo da automação, é preciso uma infraestrutura adicional para sua execução e a participação de profissionais especializados que geralmente recebem salários mais altos (ANTUNES, 2021).

As vagas disponíveis no mercado atual para testadores ou QAs são voltadas para automação de testes, em sua maioria. Este é um tipo de habilidade que hoje

qualquer profissional de testes deve ter, mas vale lembrar que testes automáticos não substituem completamente os manuais pois, como citado anteriormente, eles têm suas limitações como serem incapazes de detectar problemas de usabilidade por exemplo.

O Quadro 1 apresenta uma breve comparação entre os testes manuais e automáticos.

Quadro 1 – Testes manuais x automáticos

| Testes Manuais | Testes Automáticos |
|---|--|
| Velocidade de execução baixa | Execução muito rápida |
| Repetitivo e Cansativo | Não cansam e não sentem preguiça. |
| Não exige tecnologias | Exige domínio de tecnologias específicas |
| Alto custo a cada execução | Alto custo apenas na criação |
| Possuem limitações quando o teste envolve situações de grande paralelismo | Permite testar situações impossíveis de testar manualmente |
| Podem explorar além do cenário de teste, quando necessário | Faz apenas o programado para fazer |
| Podem avaliar questões visuais como cores e formas | Não avaliam questões visuais |
| Podem avaliar questões de usabilidade | Não avaliam questões de usabilidade |

Fonte: Farias (2019, p. 73)

7.6 ERROS COMUNS NA ATIVIDADE DE TESTES

O entendimento de alguns desenvolvedores em relação aos testes de sistemas pode ser equivocado. O processo de testes no desenvolvimento de software às vezes não acontece de forma adequada, ocorrendo erros que podem comprometer a qualidade dos sistemas e a produtividade dos profissionais.

Colocar desenvolvedores para realizarem testes sistêmicos e atuar de fato como um testador é algo comum de acontecer, e pode ser que dê certo, mas não é o mais adequado, pois um desenvolvedor e um testador devem ter perfis diferentes. A forma como pensam e executam os cenários de teste não é a mesma. Sem contar que em alguns casos, o teste é feito pelo próprio desenvolvedor que implementou

aquelas modificações, o que pode ser bem perigoso e uma prática que deve ser evitada (FARIAS, 2019).

Outro equívoco é pensar que os testes têm como objetivo demonstrar a ausência de *bugs* em um software, mas na realidade a ideia é mostrar a existência dos erros para que sejam corrigidos antes de entrar em produção, e ainda que nenhum defeito seja encontrado, não significa que não possam existir (ISTQB, 2019).

Ainda relacionada a esta questão, a visão por parte de alguns gestores sobre os resultados dos testes às vezes é a de que não tenha sido realizado um bom trabalho já que todas as possibilidades de falhas deveriam ter sido cobertas. A questão é que o testador pode não ter feito um bom teste, mas esperar que tudo seja testado é um equívoco. Considerar a validação de todas as possíveis combinações de entradas e pré-condições é complexo e demorado, e nem se alcançam resultados significativos. O testador deve saber planejar e elaborar testes assertivos para concentrar esforços no que realmente importa.

Há, ainda, que se ressaltar um erro cometido pelos próprios profissionais da área. A repetição de testes, sem ajustes aos novos contextos de funcionalidades e operação não vai indicar novos defeitos. Para poder detectar falhas, os testes já existentes precisam ser atualizados, utilizando novos dados de entrada e realizando ações diferentes (ISTQB, 2019).

8 A REALIDADE E O FUTURO DA ÁREA

A transformação digital representa um mecanismo de inovação no setor de TI. Com a automação de testes, por exemplo, a equipe de QA pode trabalhar cada vez mais focada no desenvolvimento de planos para garantir a qualidade do produto digital, deixando o trabalho repetitivo dos testes para soluções tecnológicas que automatizem os processos, melhorando os resultados (SANTOS, 2019).

Apesar das melhorias que são obtidas com testes, os profissionais da qualidade ainda não têm sua relevância reconhecida. São percebidos como membros das equipes de desenvolvimento, mas sem o destaque merecido. Com a experiência adquirida, os testadores aumentam sua produtividade e passam a poder medir consistentemente os indicadores de resultados para comprovar os ganhos alcançados com investimentos em testes. Como citado anteriormente, há ainda a questão dos conflitos na equipe de desenvolvimento que deve ser considerada e tratada para minimizar seus efeitos.

Outra prática a ser evitada em testes de software, é a concentração de esforços nas etapas finais do processo de desenvolvimento, o que acaba dificultando a atividade dos QA especialmente para cumprimento do cronograma do projeto. Erros detectados tardiamente implicam retrabalho que impactam os prazos de entregas.

Em empresas maiores de desenvolvimento de software, com níveis elevados de maturidade, é mais comum a disseminação da cultura que todos os membros das equipes garantam a qualidade dos produtos, não se caracterizando nelas os cargos e funções de QA especificamente. Não significa, entretanto, que os processos de *Quality Assurance* não estejam presentes. A diferença é que as técnicas e processos de testes são conhecidos por todos os membros das equipes.

No geral, essa é uma área que está em constante crescimento com cada vez mais empresas adotando políticas de garantia da qualidade e os profissionais que trabalham na área são cada vez mais requisitados e melhor remunerados. Essa evolução é visível pois percebe-se a preocupação das empresas definirem cargos específicos e planos de carreira para profissionais da área.

CONSIDERAÇÕES FINAIS

Quando iniciado o trabalho de pesquisa, constatou-se que grande parte dos profissionais que estão ingressando na carreira de tecnologia, não conhecem a área da qualidade de software, e às vezes não se interessam por terem uma visão vaga das funções que são desempenhadas. Além disso, nas organizações de pequeno e médio portes, os gestores costumam não ter interesse em estruturar uma equipe focada na qualidade, pois não compreendem como o investimento pode agregar valor aos produtos que são desenvolvidos.

Diante disso, a pesquisa teve como objetivo identificar como a Garantia da Qualidade contribui com o sucesso dos softwares, o valor que essa área e os profissionais que trabalham nela agregam nos projetos e de que forma fazem isso, descrevendo suas atividades, atribuições e as competências que devem possuir.

Através deste estudo, foi possível comprovar a ideia inicial de que o investimento em qualidade é válido e que gera resultados positivos para a desenvolvedora de software.

Para o desenvolvimento deste projeto foram realizadas pesquisas em uma diversidade de fontes que apresentam estudos acerca do tema escolhido; consultas em documentos oficiais de treinamento para certificações na área da qualidade e testes de softwares; e materiais elaborados por profissionais com relevância de mercado que ministram cursos de capacitação. Além das pesquisas também foram levados em consideração depoimentos, opiniões e conversas com pessoas que trabalham como analistas de qualidade e possuem competência para tratar do assunto.

Diante da metodologia adotada, percebe-se que poderia ter sido realizada uma pesquisa mais ampla na bibliografia para encontrar referências que apresentassem um estudo de caso, por exemplo.

Como estudo futuro para entendimento mais aprofundado no assunto, recomenda-se um levantamento de campo, a fim de coletar dados e situações reais de falhas e incidentes, por exemplo, com identificação das consequências causadas e como poderiam ter sido evitadas com o trabalho de uma equipe voltada para Garantia de Qualidade. Outro benefício de um levantamento de campo é encontrar casos de sucesso nos quais a QA contribuiu no projeto.

REFERÊNCIAS

ANTUNES, Flávio. **Diferença entre fases de teste, tipos de teste e formas de execução**, 01/04/2021. Disponível em: <<https://www.zup.com.br/blog/fases-de-teste-tipos-de-teste#:~:text=Testes%20automatizados%20s%C3%A3o%20mais%20confi%C3%A1veis,testes%20de%20carga%20e%20desempenho>>. Acesso em: 30.set.2022.

FARIAS, Gustavo. **Testes de Software**. São Paulo, 2019.

ISTQB. **INTERNATIONAL SOFTWARE TESTING QUALIFICATIONS BOARD, Certified Tester Syllabus Foundation Level**. Edinburgh: ISTQB, 2019.

LIBÂNIO, Grace. **Os desafios de se implementar uma política de quality assurance dentro do desenvolvimento**, 02/10/2020. Disponível em: <<https://itforum.com.br/noticias/os-desafios-de-se-implementar-uma-politica-de-quality-assurance-dentro-do-desenvolvimento/>>. Acesso em: 04.out.2022.

MULLER, Guilherme. **O que é Teste de Software? Por que é necessário?**, 19/10/2020. Disponível em: <<https://cwi.com.br/blog/o-que-e-teste-de-software-por-que-e-necessario/>>. Acesso em: 21.set.2022.

RODRIGUES, André. **O custo efetivo dos bugs no desenvolvimento de softwares**, 05/05/2017. Disponível em: <<https://interaktiv.com.br/blog/post/o-custo-efetivo-dos-bugs-no-desenvolvimento-de-softwares>>. Acesso em: 02.out.2022.

SANTOS, Danilo. **Quality Assurance: entenda o que é e como aplicar na gestão de TI**, 19/03/2019. Disponível em: <<https://kalendae.com.br/blog/quality-assurance/>>. Acesso em: 30.set.2022.

SILVA, Gizele. **Pessoa QA: o que faz e qual a sua importância**. Disponível em <<https://coodesh.com/blog/candidates/carreiras/pessoa-qa-o-que-faz-e-qual-a-sua-importancia/#:~:text=Fun%C3%A7%C3%B5es%20da%20pessoa%20QA,conforme%20a%20expectativa%20do%20cliente>>. Acesso em: 01.out.2022.

SOFEAST. **What Is Quality Assurance?**. Disponível em <<https://www.sofeast.com/glossary/what-is-quality-assurance/>>. Acesso em: 01.out.2022.