

# A IMPORTÂNCIA DO TESTE DE SOFTWARE PARA A QUALIDADE DO PROJETO

**Autora: LUCIANA DE BARROS ARAÚJO<sup>1</sup>**

**Professor Orientador: LUIZ CLAUDIO DE F. PIMENTA<sup>2</sup>**

## RESUMO

O mercado atual está cada vez mais exigente com relação à qualidade dos produtos e serviços oferecidos. Para Projetos de Software a etapa de testes é de fundamental importância, pois busca garantir a qualidade do Software desenvolvido. Diante disso, a aplicação do Gerenciamento da Qualidade em Projetos de Software é importante para o aumento das perspectivas de sucesso do projeto. Os testes de software alinhados aos conceitos de qualidade, modelos de melhoria de processos de teste e as boas práticas do Gerenciamento da Qualidade contribuem para otimizar as chances de sucesso do projeto.

**Palavras-chaves:** Testes de Software; Qualidade.

## 1 INTRODUÇÃO

Com o aumento crescente de novas tecnologias as Organizações passaram a investir em desenvolvimento de Softwares. Em função da grande concorrência do mercado, as empresas passaram a exigir maior qualidade nos sistemas desenvolvidos pra aumentar a eficiência e diminuir custos.

Para acompanhar as exigências do mercado e atingir a qualidade do produto solicitado, as empresas de desenvolvimento de software têm aplicado novas metodologias para controlar os processos além de investir em uma equipe qualificada de testes, o que há algum tempo não se era visto com importância, principalmente empresas de pequeno porte, pois além do custo alto de manter uma equipe focada nesta etapa do Projeto ainda há a cultura que os testes podem ser feitos pelos desenvolvedores.

---

<sup>1</sup> Aluna concluinte do curso MBA Gestão Estratégica de Projetos  
Centro Universitário UNA  
lu.barros@ymail.com

<sup>2</sup> Professor da disciplina Gerenciamento da Qualidade do curso MBA Gestão Estratégica de Projetos  
Centro Universitário UNA  
pimenta-luiz@ig.com.br

Com o objetivo de melhorar a qualidade dos testes algumas organizações consideram as atividades de teste de software como um projeto paralelo e integrado ao projeto de desenvolvimento.

O teste de software tem como objetivo garantir a qualidade dos sistemas desenvolvidos. Durante esta atividade é possível reduzir riscos para o negócio, diminuir custos, reduzir a probabilidade de defeitos, identificar falhas no desenvolvimento do software e corrigi-las antes que o produto seja entregue para o Cliente. A correção do software após implantação no ambiente de produção tem o custo mais elevado do que durante a fase de testes.

Diante disso, os testes de software alinhados a conceitos de qualidade, modelos de melhoria de processos de teste e as boas práticas do Gerenciamento de Qualidade podem aumentar as perspectivas de sucesso do projeto.

## **1.1 OBJETIVO GERAL**

Demonstrar a importância do teste de software para a qualidade do projeto.

## **1.2 OBJETIVOS ESPECÍFICOS**

- Apresentar conceitos e técnicas do teste de software
- Descrever as características e técnicas da área de conhecimento do Gerenciamento de Qualidade com base no PMBOK
- Apresentar modelos de melhoria de processos de teste

## **2 REFERENCIAL TEÓRICO**

### **2.1 Qualidade**

Apesar de o termo qualidade ter diversas definições elas se complementam.

Juran (1992) define a qualidade como ausência de deficiências, ou seja, quanto menos defeitos, melhor a qualidade.

## Segundo Feigenbaum (1994)

Qualidade é a correção dos problemas e de suas causas ao longo de toda a série de fatores relacionados com marketing, projetos, engenharia, produção e manutenção, que exercem influência sobre a satisfação do usuário.

Para Crosby (1986), a Qualidade é a conformidade do produto às suas especificações. As necessidades devem ser especificadas, e a qualidade é possível quando essas especificações são obedecidas sem ocorrência de defeito.

Qualidade é tudo aquilo que melhora o produto do ponto de vista do cliente segundo Deming (1993).

## Segundo Pressman (1995)

Qualidade de software é definida como Conformidade a requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo software profissionalmente desenvolvido.

Ishikawa (1993) define que a Qualidade é desenvolver, projetar, produzir e comercializar um produto de qualidade que é mais econômico, mais útil e sempre satisfatório para o consumidor.

Para Gilmore (1974) a Qualidade é o grau em que um produto específico está de acordo com o projeto ou especificação.

## Segundo Vargas (2007)

Qualidade é satisfazer a necessidade do cliente. Se o cliente deseja algo elementar, ter qualidade é conseguir entregar esse algo elementar. O entendimento de qual é a necessidade do cliente é que vai orientar como fazer o trabalho, é entender o que o cliente quer e entregar o que o cliente quer.

A qualidade é definida como o grau em que o projeto cumpre os requisitos. Planejar a qualidade é o processo de identificar os requisitos e/ou padrões de qualidade do projeto e do produto, bem como documentar de que modo o projeto demonstrará a conformidade (PMBOK, 2008).

## 2.2 Testes de software

### 2.2.1 Conceito

As definições de teste de software são diversas, mas independente do autor pode-se concluir que o objetivo dos testes de software é garantir a qualidade.

Segundo Myers (1979)

Testar um software é um processo de executar um programa com o objetivo de encontrar defeitos. Teste de software é um processo, ou um grupo de processos, definidos para garantir que um código faz o que ele foi desenhado para fazer, e não faz nada que não foi especificado para fazer.

Para Hetzel (1988), é qualquer atividade que a partir da avaliação de um atributo ou capacidade de um programa ou sistema seja possível determinar se ele alcança os resultados desejados.

Bons testadores necessitam de um conjunto especial de habilidades. Um testador deve abordar um software com a atitude de questionar tudo sobre ele (McGregor e Sykes, 2001).

Testar software é a atividade de comparar o que um item de teste faz com o que é esperado que faça (ISO/IEC-29119).

Rios (2011) conclui sobre testes de software:

- Os testes são feitos para verificar se o software está fazendo o que foi pedido que ele fizesse num requisito;
- O teste deve ser considerado um projeto;
- É importante ter um processo de teste;
- Testar para garantir que o negócio não vai correr riscos provocados por defeitos em produção;
- Testar para assegurar a qualidade do software.

### **2.2.2 Fases de testes:**

Segundo descrito o ISTQB (2007), o teste de software pode ser dividido em quatro fases:

- **Teste Unitário ou de Componentes:** Tem como objetivo procurar defeitos e verificar o funcionamento do software que são testáveis separadamente.
- **Teste de Integridade:** É caracterizado por testar as interfaces entre os componentes, interação de diferentes partes de um sistema.
- **Teste de Sistema:** Refere-se ao comportamento de todo o sistema/produto definido pelo escopo de um projeto ou programa desenvolvido.
- **Teste de Aceite:** Tem como objetivo estabelecer a confiança no sistema, antes que ele entre em produção.

### **2.2.3 Técnicas de testes:**

O teste de software tem como objetivo garantir a qualidade dos sistemas desenvolvidos. Para atingir esse objetivo várias técnicas podem ser utilizadas.

De acordo com ISTQB (2007), as técnicas de teste são:

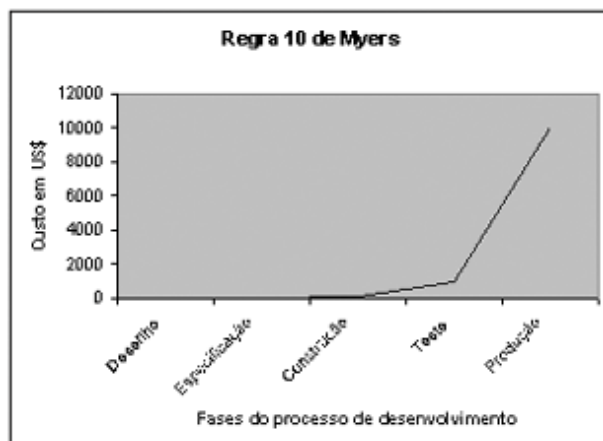
- **Estrutural ou Caixa Branca:** Utilizado quando o objetivo é garantir que o produto desenvolvido está estruturado e funciona corretamente. Pode ser dividido em teste de stress, contingência, segurança, performance e conformidade.
- **Funcional ou Caixa Preta:** Utilizado quando o objetivo é verificar se os requisitos do sistema e as especificações foram atendidos, pode ser dividido em teste de funcionalidade, regressão, interconexão e usabilidade.

## 2.2.4 Erro, Defeito ou Falha:

Estes termos são utilizados na Engenharia de Software e definidos por Rios (2011):

- Erro: Problema introduzido no software pelo programador.
- Defeito: Problema encontrado no software pelos testadores. É decorrente de um erro.
- Falha: Problema ocorrido no software por um erro não detectado pelos testes.

Segundo Myers (1979), o custo de correção de defeitos tende a aumentar quanto mais tarde o defeito ocorre. Defeitos encontrados nas fases iniciais da etapa de desenvolvimento do software são mais baratos de serem corrigidos do que aqueles encontrados na produção.



Fonte: Rios 2007

## 2.2.5 Teste Estruturado: Ciclo de Vida de Teste

Segundo Rios (2007), quanto antes os testes iniciarem, tanto mais barato será corrigir os defeitos encontrados. Para que isso seja possível é preciso que o processo de teste, assim como o processo de desenvolvimento, tenha também um ciclo de vida, conforme detalhado no quadro abaixo:

<b>Ciclo de vida de testes</b>
<ol style="list-style-type: none"> <li>1. Início</li> <li>2. Planejamento</li> <li>3. Especificação</li> <li>4. Execução</li> <li>5. Término</li> </ol>
<p><b>Atividades desenvolvidas no ciclo de vida dos testes</b></p> <p>Para que o projeto de testes seja desenvolvido adequadamente, foram definidas as atividades que compõem este ciclo de vida:</p>
<ol style="list-style-type: none"> <li>1. Definição do escopo do projeto</li> <li>2. Participação na revisão da documentação de desenvolvimento;</li> <li>3. Elaboração da lista de requisitos de teste usando os requisitos do sistema;</li> <li>4. Elaboração do planejamento dos testes (plano de testes e estratégia de testes);</li> <li>5. Preparação do ambiente de testes</li> <li>6. Elaboração dos casos de teste;</li> <li>7. Preparação da massa de teste;</li> <li>8. Execução dos testes (programas e scripts);</li> <li>9. Registro dos defeitos encontrados;</li> <li>10. Documentação dos resultados do teste;</li> <li>11. Revisão dos resultados do teste;</li> <li>12. Manutenção da documentação de testes (bibliotecas de casos de testes, etc.)</li> </ol>
<b>Documentos do processo de testes</b>
<ol style="list-style-type: none"> <li>1. Estratégia de testes</li> <li>2. Análise de riscos</li> <li>3. Plano de testes</li> <li>4. Casos de teste</li> <li>5. Script de teste</li> <li>6. Registro de defeitos</li> </ol>

Modelo de teste estruturado

Fonte: Rios 2007

### 2.2.6 Melhoria dos testes

Escolher um (ou mais) processos de testes adequado e integrado ao processo de desenvolvimento de software:

- Estruturar uma área orientada para testes e com pessoal capacitado;
- Utilizar um ambiente de testes adequado;
- Utilizar ferramentas de automação dos testes;
- Utilizar regras de medição do projeto de testes;

- Utilizar mecanismos de aferição do nível de maturidade do processo de testes visando conseguir melhorias contínuas;
- Usar técnicas de gerência de projetos (PMI);
- Identificar, documentar e compartilhar as lições aprendidas.

### 2.3 Normas e Modelos

A engenharia de software evoluiu ao longo dos anos procurando estabelecer normas, critérios, métodos, técnicas e ferramentas para a produção do software. Para a garantia da qualidade do software foram desenvolvidos métodos para todo o processo de desenvolvimento do software, inclusive atividades relacionadas aos testes.

As normas têm o foco no cliente e as organizações dependem de seus clientes. É importante o entendimento das necessidades atuais e futuras do cliente, atender aos requisitos e exceder as expectativas.

A norma ISO/IEC 25010 define as características de qualidade que todos os softwares devem ter. Cumprir o que a norma determina, significa alcançar um nível muito alto de qualidade no software que será entregue.

<b>NORMA ISO/IEC 25010</b>	
<b>Características de Qualidade</b>	<b>Descrição resumida da característica</b>
<b>Funcionalidade</b>	Este nível de qualidade é atingido quando as funcionalidades do software atendem ao que foi solicitado nos seus requisitos
<b>Performance</b>	Este nível de qualidade é atingido quando o software mantém o desempenho previsto nos requisitos
<b>Compatibilidade</b>	Este nível de qualidade é atingido quando o software consegue trocar informações com outros sistemas no mesmo ambiente operacional.
<b>Usabilidade</b>	Este nível de qualidade é atingido quando o software pode ser usado por usuários específicos com níveis determinados de eficácia, eficiência e satisfação.



<b>Confiabilidade</b>	Este nível de qualidade é atingido quando o software, sob determinadas condições, (exemplo, escassez de recursos) consegue executar as suas funcionalidades de forma confiável.
<b>Segurança</b>	Este nível de qualidade é atingido quando o software protege as suas informações e dados de acordo com níveis de autorização.
<b>Manutenibilidade</b>	Este nível de qualidade é atingido quando o software pode ser mantido ou modificado com eficácia e eficiência.
<b>Portabilidade</b>	Este nível de qualidade é atingido quando o software pode ser transferido para outro ambiente operacional definido nos seus requisitos com eficiência e eficácia.

Fonte iteste

A norma ISO / IEC / IEEE 29119 Teste de Software é uma entidade internacional com um conjunto de normas para testes de software que podem ser usados em qualquer ciclo de vida de desenvolvimento de software ou organização. Com a implementação destas normas, a organização adota os padrões reconhecidos internacionalmente e acordados para testes de software, que irá fornecer uma abordagem de alta qualidade para testes que podem ser comunicados em todo o mundo.

O Modelo de Maturidade em Capacitação - Integração CMMI é um conjunto de práticas de gerenciamento e de melhoria de qualidade a ser aplicado no processo de desenvolvimento de software, um modelo para a melhoria contínua de processos da Organização.

Devido aos testes de softwares terem ganhado importância para obtenção da qualidade desejada foram desenvolvidos alguns modelos específicos:

TMM - Modelo de Maturidade em Teste. O objetivo do TMM é dar suporte às organizações na melhoria do processo de testes e tomou como base os seguintes itens:

- É um modelo complementar ao CMMI com o qual mantém compatibilidade;
- É baseado na avaliação da situação atual do processo de testes através de regras claras e objetivas;
- É uma linha para a melhoria contínua do processo de testes;
- É um modelo baseado nas melhores práticas de teste existentes no mercado.

TPI - O modelo TPI foca na melhoria do processo de testes e ajuda a definir gradualmente os passos para sua evolução, levando em consideração o tempo, o custo e a qualidade.

O MPT.BR é um modelo para melhoria de processo de Teste de Software tomando como referência a norma ISO-29119 e o CMMI. Uma empresa certificada comprova o conhecimento e aplicação do uso das melhores práticas de mercado, promovendo alto grau de confiança e controle na elaboração dos projetos. Ao atingir a maturidade comprova a excelência no processo de gestão de teste, levando segurança a seus clientes por reconhecerem na maturidade de processos, sinônimos de confiabilidade e qualidade.

O MPT.Br trata a melhoria do processo de teste através das melhores práticas relativas às atividades desenvolvidas ao longo do ciclo de vida de teste do produto. O objetivo do modelo é auxiliar às organizações a alcançarem os resultados almejados através da melhor execução das atividades planejadas, minimizando os impactos quando da introdução e uso de novas tecnologias. Os pontos críticos do modelo, a base referencial em teste e a evolução da engenharia de software são sempre levados em consideração para consolidação e implementação da melhoria contínua do modelo.

## **2.4 Guia PMBOK: Gerenciamento da qualidade**

O Guia PMBOK (2008) é um livro que relata as boas práticas no gerenciamento de projetos. Este guia é uma base para muitas empresas se nortearem no momento de planejarem e executarem seus projetos.

Guia PMBOK (2008) ainda expõe que cada projeto existe para obter resultados, e apesar de todos os projetos usarem essencialmente matérias primas iguais ou semelhantes, objetivam efeitos exclusivos, fato que demanda adaptação, planejamento do gerente de projeto e das equipes responsáveis.

O gerenciamento do projeto acontece por meio da integração e aplicação de cinco etapas de processos: iniciação; planejamento; execução; monitoramento e controle e encerramento. Tais etapas são encadeadas lógica e estrategicamente são

compostas de 42 processos cujos fins são: a aplicação do conhecimento habilidades, ferramentas e técnicas.

O gerenciamento de projetos é dividido, conforme o Guia PMBOK (2008), em áreas de conhecimento. São elas: Gerenciamento de integração do projeto; gerenciamento do escopo do projeto; gerenciamento de tempo do projeto; gerenciamento de custos do projeto; gerenciamento da qualidade do projeto; gerenciamento de recursos humanos do projeto; gerenciamento das comunicações do projeto; gerenciamento de risco do projeto; gerenciamento de aquisições do projeto.

Neste artigo, o enfoque é a área de gerenciamento de qualidade por haver maior relação com a atividade teste de software.

O gerenciamento da qualidade inclui criar e seguir políticas e procedimentos para garantir que um projeto atenda as necessidades definidas que deveriam atender. Também pode significar concluir o projeto sem desvios em relação aos requisitos do projeto. Inclui planejar, realizar a garantia e realizar o controle da qualidade conforme PMBOK (2008).

Segundo o guia PMBOK (2008), o gerenciamento da qualidade do projeto possui três processos: Planejamento, Execução e Controle.

Planejar a qualidade é identificar os recursos e padrões de qualidade do projeto e do produto para atender aos requisitos do cliente.

Em relação ao planejamento da qualidade de acordo com o guia PMBOK (2008), existem 9 técnicas/ferramentas a serem utilizadas, são elas: Análise de custo-benefício, Custo da qualidade, Gráficos de controle, Benchmarking, Projeto de experimentos, Amostragem estatística, Elaboração de fluxogramas, Metodologias proprietárias de gerenciamento da qualidade, Ferramentas adicionais.

Para cada projeto deve ser avaliada qual a ferramenta é mais adequada. Abaixo estão descritas algumas dessas ferramentas.

Benchmarking:

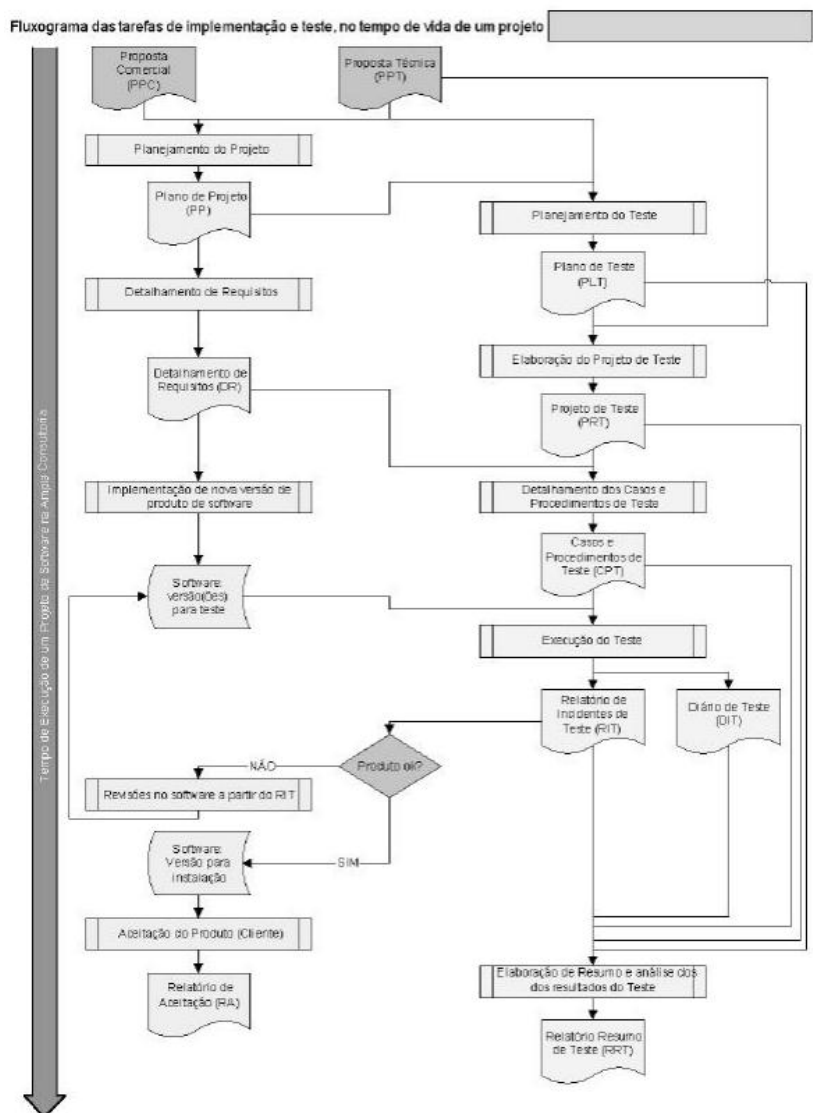
Comparação de práticas de projetos reais ou planejadas às de outros projetos para gerar ideias de melhoria e para fornecer uma base pela qual deve ser medido o desempenho;

Elaboração de fluxogramas:

Mostra como um processo ou sistema flui do início ao fim, e como os elementos se inter-relacionam. É usado no controle da qualidade para analisar problemas de qualidade, mas também pode ser usado no planejamento para analisar problemas de qualidade futuros potenciais e determinar padrões da qualidade. O fluxograma torna mais fácil a análise de um processo e a sua correta identificação das entradas e saídas e dos pontos críticos do processo.

O fluxograma utiliza um conjunto de símbolos para representar as etapas do processo, as pessoas ou os setores envolvidos, a sequência das operações e a circulação dos dados e dos documentos.

Exemplo de fluxograma: Processo de teste e seu relacionamento com o processo de desenvolvimento.



Fonte: Artigo "Uma Metodologia para Teste de Software no Contexto da Melhoria de Processo."

Metodologias proprietárias de gerenciamento da qualidade:

São metodologias existentes no mercado com regras próprias para a qualidade de um projeto que podem agregar valor ao projeto. O PMBOK (2008) cita, por exemplo, CMMI, Six Sigma, porém outras metodologias podem ser utilizadas.

Ferramentas adicionais de planejamento da qualidade:

Brainstorm, análise de campo de força, diagramas matriciais, matrizes de priorização, reuniões.

O processo Realizar a garantia da qualidade (Execução) é determinar se os padrões estão sendo cumpridos, se o trabalho é aprimorado continuamente e se as falhas são corrigidas. É fundamental para a melhoria contínua da execução projeto.

De acordo com o guia PMBOK (2008), existem 3 técnicas/ferramentas a serem utilizadas, são elas: Ferramentas e técnicas de planejamento da qualidade e realizar o controle da qualidade, Auditorias da qualidade e Análise de processos. Para cada projeto deve ser avaliada qual a ferramenta é mais adequada. Abaixo estão descritas algumas dessas ferramentas.

Auditorias da qualidade: Verificar o cumprimento das políticas, dos padrões e dos procedimentos da empresa, e determinar se as políticas, os padrões e os procedimentos usados são eficientes e eficazes. Em geral, as auditorias da qualidade são realizadas pelo departamento de garantia da qualidade, mas o gerente de projetos pode realizar esse trabalho se a organização executora não tem um departamento de garantia da qualidade.

Análise de processos: É parte da melhoria de processos.

Realizar o controle da qualidade é o processo de monitoramento e registro dos resultados da execução das atividades de qualidade para avaliar o desempenho e recomendar as mudanças necessárias. De acordo com o guia PMBOK (2008), existem 3 técnicas/ferramentas a serem utilizadas, são elas: Diagrama de Causa e Efeito, Gráficos de Controle, Fluxogramas, Histograma, Diagrama de Pareto, Gráfico de Execução, Diagrama de Dispersão, Amostragem Estatística, Inspeção, Revisão das solicitações de mudanças aprovadas.

Para cada projeto deve ser avaliada qual a ferramenta é mais adequada. Abaixo estão descritas algumas dessas ferramentas:

O Diagrama de Causa e Efeito (ou Espinha de peixe) é uma técnica largamente utilizada, que mostra a relação entre um efeito e as possíveis causas que podem estar contribuindo para que ele ocorra.

Histograma: São gráficos de barras que mostram a variação sobre uma faixa específica. (JURAN, 1989). Segundo Paladini (1994), a maneira como esses dados se distribuem contribui de uma forma decisiva na identificação dos dados. Eles descrevem a freqüência com que variam os processos e a forma de distribuição dos dados como um todo.

O Diagrama de Pareto, ou gráfico de Pareto, é um tipo específico de histograma ordenado por freqüência de ocorrência. Mostra quantos defeitos foram gerados por tipo ou categoria de causa identificada. A ordem de classificação é usada para direcionar a ação corretiva. O princípio de Pareto se aplica aos testes de software. Implica que 80% de todos os defeitos descobertos durante o teste podem ser relacionados a 20% de todos os componentes do programa. O problema, sem dúvida, é isolar os componentes suspeitos e testá-los rigorosamente. (MYERS, 1979)

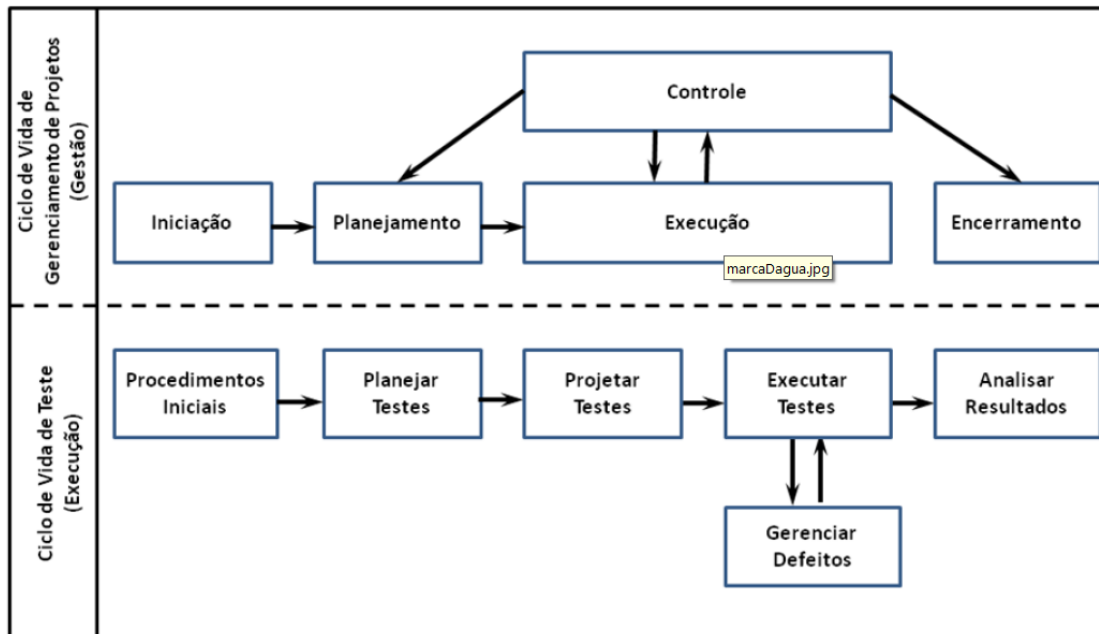
Revisão das solicitações de mudanças aprovadas: Todas as mudanças devem ser revisadas para verificar se foram implementadas corretamente.

Projetos para serem bem conduzidos precisam de processos. Segundo o PMBOK (2008), um processo é uma sequência repetitiva de operações, com o objetivo de produzir resultados padronizados, podendo ser definido também como um conjunto de ações e atividades inter-relacionadas realizadas para se obter um conjunto pré-especificado de produtos, resultados ou serviços.

Segundo o PMBOK (2008), os ciclos de vida normalmente definem:

- Qual trabalho deve ser realizado em cada fase;
- Quais entregas (deliverables) devem ser geradas em cada fase e como devem ser validadas e verificadas;
- Os Stakeholders envolvidos em cada fase;
- Como controlar e aprovar cada fase.

No quadro abaixo é possível verificar o alinhamento entre o ciclo de vida dos testes e o ciclo de vida do Gerenciamento de Projetos:



Fonte Iteste

### 3 PROCEDIMENTOS METODOLÓGICOS

O estudo foi elaborado através da leitura de artigos, trabalhos acadêmicos e livros específicos da área. O objetivo do estudo foi identificar as metodologias, normas, processos e técnicas para a utilização nos testes de software.

De acordo com Morano (2003), a pesquisa bibliográfica consiste num estudo sistematizado desenvolvido com base em material publicado em livros, revistas, jornais, redes eletrônicas, isto é, material acessível ao público em geral. Desenvolvida a partir de material já publicado, como livros, artigos, periódicos, internet e outros.

### 4 CONSIDERAÇÕES FINAIS

É possível concluir através da pesquisa bibliográfica que os testes de software agregam valor ao produto final e que estão em conformidade com os conceitos de qualidade. Se a qualidade é atender as necessidades do cliente, testar um software é contribuir para que o software desenvolvido esteja de acordo com as expectativas do cliente.



Embora muitas melhorias tenham ocorrido na área de teste de software, muitas empresas não consideram esta atividade como um processo importante na Organização. Quando os testes são feitos considerando processos bem estruturados através de metodologias de testes podem evitar o retrabalho, prejuízo e insatisfação do cliente.

Conforme citado neste artigo existem várias metodologias disponíveis no mercado que podem contribuir para que o software seja entregue atendendo as necessidades do cliente. Os modelos de melhoria de processos permitem através de um processo eficiente que os defeitos sejam encontrados no momento em que custo seja menor que o da correção.

O primeiro passo para a busca pela qualidade é a aceitação pela alta administração da Organização em investir em novos processos na área de testes. A escolha da metodologia a ser implementada vai de encontro com os objetivos estratégicos da Organização. Não importa qual a metodologia escolhida e aplicada, o importante é que a busca constante pela qualidade façam parte da cultura da Organização.

## 5 REFERÊNCIAS

CRESPO, Adalberto Nobiato, *et al.* Uma Metodologia para Teste de Software no Contexto da Melhoria de Processo. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2004/024.pdf>. Acesso em 02/11/2013.

ITESTE. Teste na Qualidade. Disponível em: <http://www.iteste.com.br/Servi%C3%A7os/TestenaQualidade.aspx>. Acesso em 27/10/2013.

ISO/IEC/IEEE 29119-1:2013. Disponível em: [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=45142](http://www.iso.org/iso/catalogue_detail.htm?csnumber=45142). Acesso em 29/10/2013.

ISTQB, Comissão Internacional para Qualificação de Teste de Software, 2007.

MPT.BR, Melhoria de Processo de Teste Brasileiro. Disponível em: <http://mpt.org.br/>. Acesso em 29/10/2013.

PMI. Um guia do conhecimento em gerenciamento de projeto. Guia PMBOK 4ª ed. \_EUA Project Management Institute, 2008.

RAMOS, Rogério. Definições de Qualidade. Disponível em: [http://www.infoescola.com/administracao\\_/definicoes-de-qualidade/](http://www.infoescola.com/administracao_/definicoes-de-qualidade/). Acesso em 01/11/2013.

RIOS, Emerson. Análise de Riscos em Projetos de Teste de Software, Editora Alta books, 2005.

RIOS, Emerson; CRISTALLI, Ricardo. Introdução ao TMM – Test Maturity Model. [http://www.testset.com.br/material/introducao\\_ao\\_tmm.pdf](http://www.testset.com.br/material/introducao_ao_tmm.pdf). Acesso em 02/11/2013

RIOS, Emerson. Gerenciando Projetos de Teste de Software, <http://www.emersonrios.eti.br/Artigos/Clinica%20de%20Teste%20de%20Software%20-%202%20hs%20e%2030%20min.pdf>. Acesso em 29/10/2013.