**Testes de Software – Testes Unitários**

**André Luiz da Silva¹**

¹Análise e Desenvolvimento de Sistemas – Faculdades Integradas Camões  
Alameda Dr. Muricy, 707 - 80020-040 - Curitiba – PR – Brasil

anluiz.designer@gmail.com

***Resumo:*** *Um documento que apresenta os principais processos de testes a ser realizado durante o desenvolvimento de um software, colocando em mais detalhes o Teste Unitário. Este processo é importante para controlar e validar áreas isoladas do código de um software, em outras palavras é a área mais baixa dos tipos de testes.*

# 1. Introdução

Os softwares no mundo atual já desempenha um importante papel para auxiliar nos negócios de empresas, e sua construção passa por diversas etapas todas com sua devida prioridade e importância.

Sistemas atuais ainda devem conseguir ser integrados com outros sistemas já existentes, pois estão em constante evolução. Na crescente demanda nos dias atuais, boa parte dos negócios depende de software para manter o fluxo de trabalho, sendo assim a qualidade do sistema deve ser vital na sua execução. Um software de qualidade identifica quando recebe dados válidos e dados inválidos e age de maneira correta para cada situação.

Um software de qualidade deve passar por testes dentro do prazo e custos controlados e devendo ser compatível com o mercado, existem modelos que trazem ótimos resultados e são utilizados por grandes organizações, alguns destes modelos utilizados tem sido o ISO/IEC 12207, CMMI, SW-CMM. Testes de software é fundamental, porém é necessário ter habilidades, conhecimento e infraestrutura.

# 2. Revisões

O presente documento reúne informações de diversas fontes, podendo ser melhorado ao longo do tempo.

* *ver. 1.0 – Criado por André Luiz da Silva, 10/08/2017.*

# 3. Visão Geral do Processo de Testes

Na década de 1970 o tempo dedicado na codificação e testes unitários eram maiores pela parte dos desenvolvedores. Isto era feito para garantir ao Usuário que o sistema estava funcionando de maneira correta, nessa época era considerado um mal necessário e não eram tratados como um processo formal alinhadas com o processo de desenvolvimento.

A partir da década de 1980, foi dada mais importância na análise de requisitos e com isso as atividades de testes ganharam mais importância e foi criado Metodologias de Testes que continuam evoluindo até os dias de hoje. Com a utilização da Internet para realização dos negócios, houve mudanças significativas nas aplicações, pois agora fatores tais como, segurança e desempenho passaram a ser relevantes tornando assim as atividades de testes mais especializadas.

# 3.1 O que é um Teste de Software?

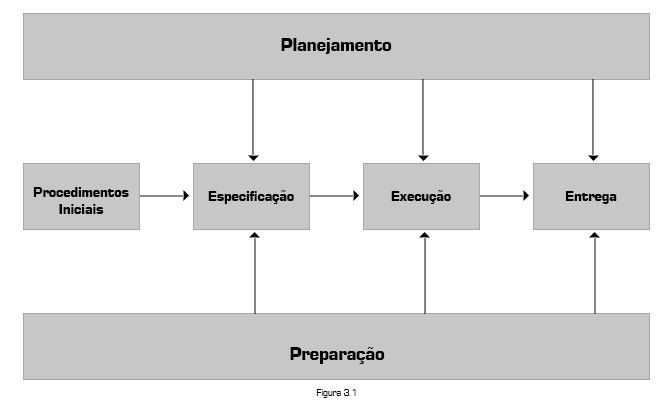
Existem diversas definições, mas na sua essência, teste de software é um processo feito de forma controlada para executar ações, com o objetivo de conferir o comportamento do que foi definido. Podemos dizer que é considerado como uma *validação*.

# 3.2 O processo de Testes

O processo de testes deve ser baseado em uma metodologia onde favorece o desenvolvimento, isto vai desde qualificação de pessoal, como também no ambiente e ferramentas adequadas.

A metodologia de testes deve ser um documento básico, da mesma forma que o desenvolvimento de um projeto é desejável haver uma metodologia adequada, o mesmo ocorre na área de testes.

A figura 3.1 apresenta o ciclo de fases de um processo de testes:



**Procedimentos Iniciais –** Elaboração do GOT (Guia Operacional de Testes), ou seja, um documento que contém um acordo entre as partes envolvidas (usuário, desenvolvimento, teste e produção), que define os assuntos e etapas do teste. Incluindo a responsabilidade de cada um, avaliação de riscos, níveis de esforço, entre outros.

**Planejamento –** Elaboração e revisão da estratégia e plano de teste.

**Preparação –** Preparação do ambiente de teste, incluindo equipamentos, pessoal, ferramentas e o software.

**Especificação –** Elaboração e revisão dos Casos de Teste.

**Execução –** Execução conforme elaborado o Caso de Teste.

**Entrega –** Conclusão do processo de teste, entregando o sistema para o ambiente de produção.

# 4. Tipos de testes

Adiante apresentamos a relação existente de testes, com suas respectivas definições e colocaremos neste documento com mais detalhes a respeito do Teste Unitário.

**Testes Caixa Preta (Black Box) –** Visa verificar as funcionalidades e comportamento do software visto pelo usuário ou alguém externo ao desenvolvimento, sem que eles tenham conhecimento do código ou da lógica do programa.

**Testes Caixa Branca (White Box) –** Avalia as lógica interna do componente, bem como as configurações e outros elementos técnicos.

**Testes Unitários –** Aplicado nos menores componentes dos códigos criados, visam garantir a funcionalidade e características conforme as especificações. Os testes unitários verificam uma parte isolada do sistema ou software afim de garantir sua funcionalidade. Este teste é na maioria das vezes realizado pelo próprio desenvolvedor.

**Testes de Integração –** Executados em uma combinação de componentes, que podem ser um código, módulos, aplicações distintas, etc. Este teste pode ser feito de forma incremental, onde cada módulo é adicionado conforme for sendo testado.

Este tipo de teste é exclusivo dos desenvolvedores ou equipe de testes.

**Testes de Sistema –** É realizado pela equipe de teste, buscando refletir o sistema no ambiente de produção mais próxima do real, utilizando cada função e módulos existentes. Aqui também é realizados os testes de cargas, performance, usabilidade, compatibilidade, segurança e recuperação.

**Testes de Aceitação –** Os testes finais onde é realizado pelos usuários, visando verificar se as soluções aplicadas atendem ao objeto de negócio e seus requisitos, mesmo assim a equipe de testes acompanham os usuários nesse período.

# 4.1 Outros tipos de testes

Além dos principais testes apresentado no tópico anterior, estes testes fazem parte dos processos, tais como *Testes de Regressão, Testes de Carga, Testes End-to-End, Testes de Configuração, Testes de Usabilidade, Testes de Instalação, Testes de Segurança, Testes de Recuperação, Testes de Compatibilidade, Testes de Desempenho, Testes Funcionais, Testes de Qualidade de Código, Testes de Alterações, Testes de Recuperação de Versões, Testes de Interoperabilidade, Testes de Sobrevivência, Testes Estáticos, Testes Embutidos*.