

Teste de Software

3ª Edição revisada e atualizada

Emerson Rios
Trayahú Moreira Filho



ALTA BOOKS
E D I T O R A

Rio de Janeiro, 2013



Nota dos autores

Quando decidimos focar o nosso trabalho em testes de software, a primeira dificuldade que enfrentamos foi a falta de uma bibliografia em língua portuguesa sobre este assunto. Isso ocorreu quando publicamos a primeira edição deste livro. Evidentemente que, de 2002 até a data da publicação desta 3ª edição, a situação mudou muito no Brasil, mas optamos por incluir esse esclarecimento, com o intuito de manter o registro de como esse trabalho iniciou. Tudo que conseguimos, na época, foram livros em inglês relatando situações muitas vezes estranhas à nossa realidade.

Um dos pontos de partida para qualquer projeto de teste de software é a adoção de uma metodologia de testes consistente e adequada ao ambiente de desenvolvimento da empresa. Como não encontramos em 2002 nada em português sobre este assunto, resolvemos, após um longo estudo das principais metodologias usadas na Europa e nos Estados Unidos, desenvolver uma metodologia de testes que fosse aderente à nossa experiência e ao que conhecemos das organizações onde trabalhamos, e de outras nas quais tivemos contato durante a nossa vida profissional. Essa metodologia foi o ponto de partida para este livro.

Decidimos também que o livro seria direcionado àqueles que estivessem se iniciando no assunto de teste de software. Mais tarde, desde que o interesse se justifique, poderemos, em outros livros, expandir alguns tópicos que pela sua abrangência deixaram de ser abordados neste livro de forma mais detalhada, pois, se assim fossem, descaracterizaria sua finalidade de ser um instrumento para quem inicia o assunto. Tópicos como Estimativas, Medições, Melhorias do Processo de Testes e Revisões Técnicas estão nessa situação; apesar de terem sido abordados com a consistência suficiente para o seu uso nos projetos de testes das organizações.

No estágio atual do mercado brasileiro, por ocasião da primeira edição deste livro, o assunto de teste de software ainda não era tratado com a seriedade que já vinha sendo dada na Europa e nos Estados Unidos, onde vários eventos específicos são promovidos anualmente. Esse quadro, evidentemente, mudou muito no Brasil, que atualmente pode ser listado como uma das referências em teste de software no mundo, principalmente em comparação com os países do chamado BRICS (Brasil, Rússia, Índia, China e África do Sul).

A primeira edição deste livro, lançado de forma pioneira no Brasil, certamente ajudou muito na mudança do quadro que havia na época. Dessa forma, esperamos com este livro pioneiro, estar dando o primeiro passo para que a qualidade dos softwares e sistemas de aplicação desenvolvidos no Brasil atinja patamares mais elevados.

Para a terceira edição deste livro resolvemos incluir um capítulo tratando a Análise de Riscos em projetos de teste de software, pois testar sempre será um exercício de evitar riscos.

Os autores

Emerson Rios

Economista formado pela UFF e pós-graduado em Engenharia de Sistemas pela COPPE/UFRJ. Possui, ainda, o título de Master of Information Technologies obtido na AIIM/EUA. Trabalha com desenvolvimento de aplicações desde 1971, tendo exercido diversos cargos de gerência no BNH e, posteriormente, na Caixa Econômica Federal, atuando sempre na Área de TI.

No BNH — Banco Nacional de Habitação, foi Chefe Adjunto do Departamento de Informática e Chefe da Área de Desenvolvimento de Sistemas. Na Caixa, foi Coordenador do Projeto Internet, tendo exercido a Gerência de Desenvolvimento de Aplicações para a Internet no Rio de Janeiro. Ocupou, ainda na Caixa Econômica, os cargos de Gerente de Suporte e Gerente de Desenvolvimento de Sistemas. Foi fundador e Diretor do Brazilian Function Point Users Group — BFPUG, e chegou a ser Diretor do Instituto Brasileiro de Simplificação de Procedimentos Mercantis — SIMPRO-Brasil, junto ao qual representou o Brasil na ONU em Genebra, de 1997 a 2001, por ocasião do Fórum Anual sobre Normatização de Comércio Eletrônico (UN-CEFACT).

Ministrou aulas nos cursos de MBA da Fundação Getúlio Vargas, onde lecionou Análise de Pontos de Função. Emerson também foi professor do curso de especialização em Gerência Estratégica da Informação — DEL/UFRJ. Trabalhou como consultor técnico da Associação Latino-Americana de Desenvolvimento em Lima, Peru. Possui três livros publicados na área de informática: *Processamento de Dados e Informática: Conceitos Básicos* (Ática, 1987); *Organização da Informática na Empresa* (Campus, 1985); e *Reconhecimento Inteligente de Caracteres (ICR/OCR): Imagens de Documentos* (CENADEM, 1997). Publicou, também, os seguintes livros na área de Teste de Software: *Documentação de Teste* (Editora Imagem Art Studio), *Caratê Aplicado ao Teste de Software* (Editora Imagem Art Studio), *Análise de Riscos em Projetos de Teste de Software* (Editora Alta Books), *Base de Conhecimento em Teste de Software* (Editora Martins Fontes). Após carreira de trinta e três anos no BNH e na Caixa Eco-

nômica Federal, resolveu concentrar suas atividades na área de Teste de Software. Trabalhou como Gerente de Produto, na área de Testes de Aplicações, na SQA — System Quality Assurance.

É Coordenador Regional para o Brasil da STAR — Software Testing Analysis and Review e foi presidente e fundador da Associação Latino Americana de Teste de Sistemas — ALATS (2002/2010). Atualmente, é consultor da empresa Riosoft/Softex na área de testes e de melhoria de processos e Diretor do Instituto de Teste de Software — ITSiTEste. Ocupa também a presidência do QAI Rio de Janeiro Chapter. É professor do MBA sobre Qualidade de Software da UFRJ.

Fale com o autor no e-mail: **`emersonrios@alats.orgiteste.com.br`**

Trayahú Rodrigues Moreira Filho

Graduado em Estatística pela ENCE e com especialização em Análise de Sistemas pela PUC-RJ.

Participou em Londres do Programa de Desenvolvimento Gerencial (PDG) para executivos seniores da BAT-Industries. Atua na área de informática desde 1968, tendo exercido cargos técnicos e gerenciais na Marinha de Guerra, Vulcan Material Plásticos, Souza Cruz, Origin C&P, SQA do Brasil Ltda, Path ITTS e iTeste. Até 1993, dedicou-se à Gerência de Desenvolvimento de Sistemas, Planejamento Estratégico de Informática, Metodologias e Ferramentas de Desenvolvimento de Sistemas, Projetos de Introdução de Novas Tecnologias e Modificações Organizacionais da Área de Informática.

Trayahú foi professor da cadeira de Estatística da Escola Naval. Desenvolveu e gerenciou projetos de desenvolvimento de sistemas em diversas plataformas e para todas as áreas de negócio das empresas onde trabalhou, tendo liderado o projeto que desenvolveu e implantou o software do primeiro armazém automático comandado por computador da América Latina. Coordenou a preparação dos PDIs 90/92 e 93/95 da Souza Cruz. Criou e implantou a Metodologia de Desenvolvimento de Sistemas daquela empresa, participou da equipe que realizou o diagnóstico de utilização de informática, cujas recomendações alteraram os paradigmas tecnológicos e organizacionais até então praticados. Em 1994, iniciou novas atividades como Consultor, tendo se dedicado prioritariamente às atividades relacionadas com Gerência de Projetos e Programas, Escritório de Projetos, Desenvolvimento de Sistemas e Qualidade e Teste de Software.

Além disso, gerenciou o Projeto de Estruturação e Implementação dos Escritórios de Projetos da Caixa Econômica Federal (Rio, São Paulo e Brasília). Coordenou a estruturação do Escritório de Projetos da Golden Cross e elaborou a Metodo-

logia de Gerenciamento de Projetos da CETIP e Mitsubishi do Brasil. Participou da preparação do Planejamento Estratégico de TI do BNDES. Foi editor da coluna de Gerência de Projeto da revista DEVELOPERS e coautor dos livros Testes de Software (Editora Alta Books) e Base de Conhecimento de Teste de Software (Editora Martins Fontes). Participou como palestrante da COMDEX/RJ sobre o tema Planejamento Estratégico de Informática. Ministrou diversos cursos sobre o tema Gerência de Projetos, Escritório de Projetos, Planejamento Estratégico de TI, Manutenção e Teste de Software. Foi Diretor da ALATS – Associação Latino Americana de Teste de Software. Atualmente é Diretor de Operações da iTeste.

Fale com o autor no e-mail: **`tmoreira@alatsiteste.org.com.br`**



Sumário

1: Introdução	1
2: Visão geral do processo de testes	9
3: Processo de testes	27
4: Gestão do processo de testes	81
5: Estimativas	91
6: Infraestrutura	125
7: Melhoria do processo de teste	133
8: Erros de Programas	155
9: Automação do processo de testes	161
10: Revisões Técnicas	173
11: Teste em Aplicações Web na Internet	179
12: Análise de riscos	195
13: Documentação de Teste	215
Anexo 1: Peça teatral tecnológica sobre testes	267
Anexo 2: Modelo de Avaliação de Maturidade	273
Anexo 3: Relatório de defeitos	279
Anexo 4: Roteiro para instalação de um Centro de Testes	287
Anexo 5: Agência de Viagens	291
Bibliografia	295



1: Introdução

No mundo atual, já há algum tempo, o software desempenha um importante papel no apoio aos negócios das empresas, sendo em algumas delas parte intrínseca. As empresas que focam na internet não conseguem sobreviver sem o apoio de um software. Essa importância tende a crescer em um mundo em que as atividades e produtos tendem a depender cada vez mais do software.

Por outro lado, desde os primeiros computadores comerciais, os softwares implantados ou lançados no mercado têm se caracterizado, na sua maioria, por apresentarem um grande número de defeitos que afetam a usabilidade, a funcionalidade, a segurança e a confiabilidade dos mesmos, com impacto decisivo nos negócios, resultando, muitas vezes, em enormes prejuízos pela perda de participação de mercado ou por danos na imagem dos produtos lançados.

Em um mundo globalizado, onde a internet é um importante suporte aos negócios, um teste malfeito pode significar um caminho aberto para diversos problemas graves, como por exemplo: fraudes, incorreções e bloqueios do site. Muitas vezes os usuários acabam abandonando o site, insatisfeitos com a sua usabilidade e/ou funcionalidade, causando enormes prejuízos à imagem da organização ou financeiros.



Neste livro, empregaremos o termo Usuário para definir a pessoa (interna ou externa à organização) que utiliza o sistema e a pessoa (interna à organização) responsável pelo apoio ao projeto de desenvolvimento e testes de um sistema.

Esse termo foi escolhido para tais finalidades, apesar de existirem outros que também atendem a essas mesmas atividades.

Informações de mercado dizem que mais de 90% dos sistemas são liberados com graves defeitos.

Softwares com problemas de performance e com defeitos na execução são custosos. Os livros e revistas relatam dezenas de casos de softwares com falhas e os seus impactos, sendo que o relatório *The Economic Impact of Inadequate Infrastructure for Software Testing* (NIST, May 2002), na época, estimava que o custo total dos

softwares com defeitos para as organizações nos EUA correspondia, em termos aproximados, a um valor um pouco abaixo de 1% do Produto Interno Bruto (PIB).

Por este mesmo relatório, o impacto na indústria produtora de software pela falta de tecnologia de testes robusta e padronizada, pode ser agrupado nas seguintes categorias:

- ▶ Aumento de falhas devido à qualidade pobre.
- ▶ Aumento dos custos de desenvolvimento.
- ▶ Aumento do time to market devido à ineficiência do processo de testes.
- ▶ Aumento do custo de suporte aos produtos no mercado.

Informações atualizadas deste estudo apontam para o mesmo resultado, mostrando que muito pouco foi alterado na situação vigente da época. As razões que encontramos é que as aplicações estão ficando cada vez mais complexas. Atualmente, um software tem que se integrar com muitos ambientes e possui inúmeros componentes que precisam se comunicar entre si. Desenvolver software está cada vez mais difícil e, como consequência disso, a atividade de testar está cada vez mais complexa.

Quando tratamos de testes, também nos referimos à qualidade do software. Não que sejam sinônimos, mas certamente o nível de qualidade dos testes de um software é um fator importante, entre outros, para definir a qualidade do produto final, que depende do processo de desenvolvimento desse software.

Apesar do conceito de qualidade ser subjetivo e depender de cada organização e/ou usuário, considera-se um software com qualidade quando:

- ▶ o número e a severidade dos defeitos residuais do processo de testes é aceitável pela organização;
- ▶ o software é entregue dentro do prazo e custo, atende aos requisitos e/ou às expectativas;
- ▶ foi construído de tal maneira que possa ser mantido de forma eficiente após sua implantação.

Dentro deste contexto, em épocas passadas, as atividades do processo de testes faziam parte do desenvolvimento de sistemas de informações e influenciavam a qualidade dos softwares, porém, devido às pressões crescentes para a liberação desses sistemas, tais atividades nem sempre eram executadas com a eficiência requerida. Além disso, tradicionalmente, os responsáveis por quase todos os estágios de testes eram os próprios desenvolvedores e os usuários, os quais, via de regra e por diversas razões, faziam com que sistemas mal testados fossem liberados para a operação.

Por incrível que possa parecer, grande parte das empresas ainda usam os seus desenvolvedores para testarem os seus softwares. O que, em resumo, significa dizer que apesar de todas as mudanças ocorridas no ambiente de desenvolvimento, ainda testam as suas aplicações de forma inadequada. Não podemos afirmar que todas as empresas que desenvolvem aplicações têm uma equipe de teste, pois isso ainda não é a realidade do mercado.

Por que essas falhas têm ocorrido com o processo de teste?

- 1) A atividade de testar o software não tem sido tratada pelas organizações como uma verdadeira prática formal. Não é dada a devida importância à essa atividade, que muitas vezes, ainda é informal, sem metodologia e com funções/responsabilidades não definidas e sem um processo adequado.

Não existe uma gestão formal da atividade, sendo esta confundida com a própria gestão do processo de desenvolvimento.

- 2) Os testes são incompletos durante o desenvolvimento, implicando em problemas ocorridos após sua implantação. Os custos para identificar e corrigir esses problemas podem ser de 100 a 1.000 vezes maiores do que se fossem realizados logo após a sua introdução. A maioria das empresas não considera o teste um projeto.

A *insuficiência dos testes* é indicada por Capers Jones em *Patterns of Software System Failure and Success* como um dos principais motivos de falha nos Projetos de Desenvolvimento de Software e, segundo o Gartner Group (2001), apenas uma pequena parcela das aplicações entregues (7%) atendem aos requisitos de tempo de resposta e performance.

- 3) A abordagem de testes não foi adequada para as novas tecnologias. Na realidade, pouco esforço foi feito nas organizações para adequar os procedimentos e reciclar o pessoal técnico de testes para tratar as novas tecnologias.
- 4) A estrutura organizacional para testes não tem se modificado. Quase todos os estágios de testes ainda são feitos pelos desenvolvedores. Nem todos eles gostam de testar o software. Outros não possuem o perfil de testador e/ou não são qualificados para executar as atividades de testes. A maioria dos desenvolvedores é muito bem treinada para desenvolver software, mas pouca atenção é dada para a capacitação desses profissionais no processo e nas técnicas de testes. Por outro lado, poucas organizações adotam equipes/organizações de testes independentes.

- 5) Pouca utilização de ferramentas de automação de testes. Por exemplo, certos tipos de testes em ambiente Cliente/Servidor e web não podem ser feitos de maneira eficaz sem o auxílio das ferramentas de testes.

Reduzir os custos do desenvolvimento de sistemas e melhorar a qualidade dos produtos finais são objetivos importantes das organizações. Por outro lado, atingir esses objetivos é um tremendo desafio devido ao crescente aumento da complexidade dos negócios que obriga a um correspondente aumento da complexidade dos sistemas e uma diminuição do seu ciclo de vida produtivo, ou seja, as atualizações são muito mais frequentes. O tamanho dos softwares passou da ordem de milhares de linhas de código para milhões ou até bilhões.

Reduzir os custos de desenvolvimento e melhorar a qualidade dos produtos finais estão intimamente relacionados. Segundo Capers Jones (1986) e B. Boehm (1987), 44% do esforço de um projeto são dedicados ao retrabalho, ou seja, refazer o trabalho já feito antes. Reduzir o retrabalho implica em realizar processos que identifiquem e corrijam defeitos o mais cedo possível de forma a evitar retrabalhar esses defeitos mais adiante no projeto, quando o custo será muito maior. Tais ações são de melhoria de qualidade.

Da mesma forma, quando são realizados testes adequados e dentro das melhores práticas, contribui-se também para a melhoria da qualidade e redução dos custos dos produtos finais, pela diminuição do retrabalho nos estágios posteriores de testes, durante a implantação e na manutenção.

No livro, abordaremos de forma objetiva os principais aspectos da atividade de testes, visando permitir que profissionais e organizações envolvidas possam se estruturar e se capacitar para a execução dessa atividade.

Além disso, abordaremos, ainda que de forma superficial, o processo de Revisão Técnica (ou Inspeção), também chamado de *Teste de Verificação Estático* por alguns autores, que é um processo relevante e complementar ao de testes e cuja utilização conjunta proporciona o melhor índice de identificação e eliminação de defeitos, tanto nos documentos produzidos durante os testes como no código produzido pelos desenvolvedores.

Com isso, esperamos que o nosso trabalho possa contribuir para melhorar a qualidade dos softwares produzidos e como consequência reduzir os custos de desenvolvimento.

Chamamos a atenção dos leitores para os seguintes pontos:

- a) A Metodologia de Testes contida no livro foi confeccionada para servir de elemento básico para a elaboração e acompanhamento/monitoração do processo de testes, independentemente do fato da equipe de testes trabalhar ou não com ferramentas de automação. Embora muitas vezes possa parecer que foi baseada exclusivamente no modelo de desenvolvimento de aplicações em cascata (waterfall), pode também ser utilizada em outros modelos, como RUP e aplicada aos ciclos de interação nas metodologias ágeis. Para facilitar e tornar mais fácil o entendimento, utilizou-se o modelo em cascata, em várias passagens do livro.
- b) A Metodologia de Testes deve ser aderente à Metodologia de Desenvolvimento de Sistemas utilizada pela organização. O processo de testes, quando usado, deve ter o seu início nas primeiras fases do processo de desenvolvimento e serem realizados de forma simultânea, até a implantação dos softwares, para que os resultados esperados sejam atingidos.

A Metodologia apresentada neste livro não deve ser autossuficiente e muitas vezes implicará na elaboração de normas complementares para dar suporte a sua implantação.

- c) As melhores práticas sugerem que a Metodologia de Testes, qualquer que seja a escolhida, deve ser implantada e consolidada na organização antes da adoção da automação do processo.
- d) O Processo de Testes está inserido no contexto das atividades de Garantia e Controle da Qualidade de Software (Software Quality Assurance) das organizações. As atividades do processo de testes atuam na detecção dos defeitos durante a execução do processo de desenvolvimento de sistemas, porém ambos devem ser definidos, alinhados e refinados sistematicamente visando à atuação preventiva dos problemas. A qualidade dos softwares só pode ser conhecida e melhorada se todos os processos forem trabalhados desta maneira.

1.1 ROTEIRO PARA LEITURA DO LIVRO

O livro introduz o assunto de testes de software como um processo de vital importância para as organizações que produzem software para uso interno ou para comercialização no mercado. Ele foi escrito com este objetivo e é também dirigido aos gestores e executores das atividades de testes, sendo eles da área de desenvolvimento de sistemas ou da área de testes.

Desta forma, procuramos dar um nível de profundidade intermediário aos diversos assuntos, às vezes, mais detalhado em alguns e menos em outros. Esta abordagem foi escolhida devido aos seguintes objetivos primordiais que pretendemos alcançar:


- ▶ mostrar as atividades de testes como um processo com suas próprias características e integradas ao processo de desenvolvimento;
- ▶ chamar a atenção para as características intrínsecas do processo de testes e dar uma visão metodológica e macro;
- ▶ motivar os gestores e executores a se aprofundarem mais nas técnicas de avaliação das atividades, nas metodologias de execução e nos métodos de gestão;
- ▶ Recomendamos que o livro não seja lido de forma sequencial, e sim de acordo com o interesse em cada assunto.

Desta forma, se o leitor:

- ▶ for uma pessoa leiga no assunto de testes de software, comece com os capítulos:
 - ▶▶ 1 — INTRODUÇÃO
 - ▶▶ 2 — VISÃO GERAL DO PROCESSO DE TESTES
 - ▶▶ 8 — ERROS DE PROGRAMAS
- ▶ já estiver familiarizado com testes e quiser ter uma macrovisão da execução através da metodologia de testes que apresentamos, passe direto para o capítulo:
 - ▶▶ 3 — PROCESSO DE TESTES
- ▶ Desejar tratar de tópicos associados à execução, incluímos os seguintes capítulos:
 - ▶▶ 5 — ESTIMATIVAS, para aqueles que quiserem entender e se aprofundarem nas técnicas de cálculo do tamanho de um projeto de testes e de estimativa das horas requeridas;
 - ▶▶ 10 — REVISÃO TÉCNICA, para aqueles que tiverem interesse em melhorar o nível de qualidade dos produtos resultantes das atividades do processo de testes, incluímos uma descrição resumida dessa técnica de suma importância como complemento aos testes;
 - ▶▶ 11 — TESTES DE APLICAÇÕES WEB, para aqueles que estiverem participando de um projeto de desenvolvimento e/ou de teste de sistema para o ambiente Web, dedicamos este capítulo para descrever as características importantes dos testes aplicados a esse ambiente.

- ▶ tiver interesse na gestão do processo, vá direto para os capítulos
 - ▶▶ 4 — GESTÃO DO PROCESSO DE TESTES
 - ▶▶ 6 — INFRAESTRUTURA
 - ▶▶ 12 — ANÁLISE DE RISCOS EM PROJETOS DE TESTE DE SOFTWARE
- ▶ for responsável pela gestão do processo nas organizações e estiver desejoso em conhecer os métodos de avaliação e de preparação dos planos de melhoria do processo, visando o aumento do nível de qualidade dos produtos resultantes das atividades do processo de testes e a eficiência do processo como um todo, recomendamos a leitura dos capítulos
 - ▶▶ 7 — MELHORIA DO PROCESSO DE TESTES
 - ▶▶ 9 — AUTOMAÇÃO DO PROCESSO DE TESTES

Finalmente, desejamos que os leitores aproveitem este livro e nos deem esperanças de que a partir daí, testar software passe a ter uma melhor apreciação contribuindo para um incremento na qualidade dos produtos. Essa ordem de leitura não impede o leitor de escolher o seu próprio caminho conforme o seu interesse no assunto. Alguns capítulos foram excluídos da lista pois podem ser lidos a qualquer momento ou dentro de qualquer ordem, e o mesmo é válido para os anexos.



2: Visão geral do processo de testes

Nas décadas de 1960 e 1970, os desenvolvedores dedicavam a maior parcela dos seus esforços nas atividades de codificação e nos testes unitários. Estima-se que 80% destes esforços eram despendidos nestas atividades.

Uma parcela menor era dedicada à integração dos programas e nos testes dos sistemas. As atividades de testes eram consideradas um mal necessário para provar aos usuários que os produtos funcionavam e não eram tratadas como um processo formal alinhado com as atividades do processo de desenvolvimento de sistemas (também poucas vezes tratado como processo).

A partir dos anos 1980, durante o processo de desenvolvimento, passou a ser dada maior importância à análise dos requisitos ao desenho funcional e técnico dos novos sistemas. Um esforço maior passou a ser dedicado à integração das diversas peças que compunham os softwares e ao teste destes para funcionarem como um sistema. As atividades de testes passaram a ser tratadas como processo formal, aparecendo as Metodologias de Testes que evoluíram até os dias de hoje.



Nota

No livro, utilizaremos o termo Requisitos para definir o conjunto de especificações que explicitam as funcionalidades e as necessidades de caráter técnico da aplicação para apoio a uma função de negócio da organização.

Nos últimos anos, com a utilização da internet para a realização de negócios, houve uma mudança significativa na abrangência e complexidade das aplicações, onde fatores, tais como segurança e performance passam a ser relevantes, tornando a atividade de testar cada vez mais especializada.



Nota

Atualmente, um site fora do ar por defeitos de programa poderá comprometer seriamente o negócio e/ou a imagem da organização.

2.1 O QUE É TESTE DE SOFTWARE

Existem diversas definições para teste de software:

- ▶ verificar se o software está fazendo o que deveria fazer, de acordo com os seus requisitos, e não o que não deveria fazer;
- ▶ processo de executar um programa ou sistema com a intenção de encontrar defeitos (teste negativo) (Glen Myers — 1979);
- ▶ 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 (Bill Hetzel — 1988).

Muitas outras definições poderiam ser ainda citadas, porém em essência, teste de software é o processo que visa a sua execução de forma controlada, com o objetivo de avaliar o seu comportamento baseado no que foi especificado. A execução dos testes é considerada um tipo de *validação*.

Na prática, não se pode testar um programa por completo e garantir que ele ficará livre de bugs. É quase impossível testar todas as possibilidades de formas e alternativas de entrada de dados, bem como testar as diversas possibilidades e condições criadas pela lógica do programador.

Segundo uma estimativa de Beizer (1990), a média do número de defeitos em programas liberados para testes é de 1 a 3 por 100 instruções executáveis. Claro, existem diferenças entre os programadores, entretanto uma coisa é certa, todos eles cometem erros seja em grau maior ou menor.

Se não se podem descobrir todos os defeitos de um programa e em decorrência disso nunca se pode afirmar que ele está 100% correto, por que testar? Porque o propósito dos testes é descobrir e corrigir os problemas e com isto melhorar a sua qualidade. O quanto se quer melhorar dependerá de quanto se deseja investir.



A qualidade do software depende do investimento feito no processo de testes.

Um software mal testado poderá custar caro (e muito) para a organização.

Dentro do processo de testes existem também as técnicas de *verificação*, tais como: inspeção, revisão de produtos e walkthroughs. Estas técnicas baseadas em reuniões e check-lists servem para identificar defeitos de elaboração, descumprimento de padrões e das boas práticas. Devem ser realizados em documentos produzidos, planos,