

TESTE DE SOFTWARE, UM INSTRUMENTO PARA A QUALIDADE

Lucas Mariano Galdino de ALMEIDA (1); Rosemary Pessoa BORGES (2); Marília Aranha FREIRE (3); Erica de Lima GALLINDO (4); Adelson Luiz de LIMA (5)

(1) CEFET-RN, e-mail: lucasmgaldino@gmail.com (2) CEFET-RN, e-mail: maryborges1@yahoo.com.br
(3) CEFET-RN, e-mail: marilia@cefetrn.br (4) CEFET-RN, email: erica@cefetrn.br (5) CEFET-RN, email: adelsontads@gmail.com

RESUMO

Nos últimos anos temos observado um aumento vertiginoso do uso de sistemas computacionais em praticamente todas as áreas da sociedade, e junto com isso, têm-se visto uma busca incessante das organizações por produzir softwares de qualidade. À medida que a utilização de sistemas de informação cresce, fazendo com que cada vez mais os processos de negócio sejam dependentes de tecnologias da informação, eles tornam-se mais complexos. Nesse cenário, a atividade de testes, que muitas vezes não recebe a devida atenção durante o processo de produção de software, surge como um instrumento para proporcionar evidências da confiabilidade do software produzido. Este trabalho analisa o estado da arte em Qualidade de Software sob a ótica de testes, apresenta a motivação para a realização de testes de software e mostra como se consegue obter, a partir destes, parâmetros que indicam o aumento da confiabilidade e, conseqüentemente, o alcance da qualidade do software.

Palavras-chave: Teste de Software, Qualidade de Software

1. INTRODUÇÃO

Assim como houve uma explosão do mercado consumidor e da produção com o surgimento da industrialização, observa-se, hoje em dia, um crescimento quase que exponencial do número de sistemas de informação dentro das organizações. A cada dia que passa, as tecnologias da informação estão mais presentes e mais atuantes no cotidiano das empresas, seja para controlar transações bancárias entre cliente e empresa, para gerenciar as vendas ou controlar o estoque de um determinado produto. Os negócios das empresas estão cada vez mais complexos e necessitam de maior agilidade na obtenção das informações. Para alcançar a agilidade necessária às informações, as organizações fazem uso de uma grande quantidade de sistemas computacionais e, conseqüentemente, necessitam que eles cumpram corretamente as funções para as quais eles foram desenvolvidos, caso contrário, poderão sofrer grandes prejuízos.

Para atender a essas e outras necessidades, a indústria de software vem buscando incessantemente o aperfeiçoamento de ferramentas, métodos e técnicas para o desenvolvimento de softwares cada vez melhores e o mais importante: sem falhas.

2. IMPACTO DA FALTA DE QUALIDADE

A todo momento, o desenvolvimento de software está sujeito a erros, mesmo fazendo uso dos mais adequados métodos de desenvolvimento, ferramentas ou profissionais, eles permanecem presentes nos produtos, tornando assim, a atividade de teste fundamental durante todo o desenvolvimento de um software. Os erros são inerentes aos seres humanos que estão atuando no processo de desenvolvimento e podem ser injetados em qualquer fase do desenvolvimento.

Segundo Deutsch (1979 apud PRESSMAN, 2002, p. 786), o desenvolvimento de sistemas de software envolve uma série de atividades de produção em que as oportunidades de injeção de falhas humanas são enormes. Erros podem acontecer logo no começo do processo, podendo os objetivos estar errônea ou imperfeitamente especificados, como também em fases de projeto e desenvolvimento posteriores. Devido à incapacidade que os seres humanos têm de executar e comunicar com perfeição, o desenvolvimento de software deve ser acompanhado por uma atividade de garantia de qualidade.

São descritos abaixo, alguns dos casos mais famosos de erros de software que poderiam ser evitados caso tivesse sido utilizado um rigoroso e sistemático processo de testes.

Acidente: Therac 25

Therac-25 ministrou doses incorrectas de Raios X em pacientes entre 1985 e 1987 – 6 mortes. (adaptado de IEEE Computer – 07/07/93)

O Therac 25 era uma máquina utilizada para o tratamento de câncer por radioterapia, controlada por um software. Na época, foi uma grande evolução, pois gerenciava as dosagens adequadas de radiação aos seus pacientes, poupando o trabalho mecânico.

A máquina operou perfeitamente durante seis meses e era considerada livre de defeitos, até dar-se início a sucessivas falhas. Infelizmente, a descoberta foi tardia e irreparável. Devido a uma superdosagem causada pelo software, ocorreram quatro mortes e um total de quinze acidentes em locais diferentes.

O caso é considerado um dos mais graves em toda a história envolvendo falhas de computadores, justamente porque o prejuízo causado pelos erros foi incalculável: vidas humanas.

Acidente: Ariane 5

Ariane 5, após 10 anos de desenvolvimento, explodiu 40 segundos após lançamento. Módulo de software gerou evento não tratado. (adaptado de ESA – 1996)

O foguete espacial, Ariane 5, utilizado para levar satélites até suas orbitas, fazia uso de um sistema inercial e um de controle para o lançamento.

Certo dia, a aeronave se autodestruíu um minuto após seu lançamento. As condições de tempo estavam favoráveis para o lançamento, mas houve um desvio na trajetória, seguido pela explosão do foguete. A investigação que apurou as causas do acidente apontou uma série de falhas iniciais que incidiram em uma

anomalia no software controlador, exatamente no instante que este executava a conversão de dados de um número de 64 bits em ponto flutuante para um inteiro de 16 bits.

Intel

Em 1994, houve um erro de ponto flutuante no processador Pentium. A correção custou à empresa US\$ 475.000.000,00. O erro teria um custo insignificante se descoberto na fase de especificação. (adaptado de Computer Science, Springer Verlag – 1995)

Motorola

A empresa *PrimeCo Personal Communications* cancelou um contrato de US\$ 500.000.000,00 com a Motorola por causa de falhas. (adaptado de Wall Street Journal – 24/02/98)

2.1. Custo do defeito

Molinari (2003) demonstra que um defeito descoberto no início do projeto custa cerca de dez centavos de dólar. Quando o defeito é descoberto no final do projeto, pode chegar aos cem dólares ou mais. Veja a figura 1.

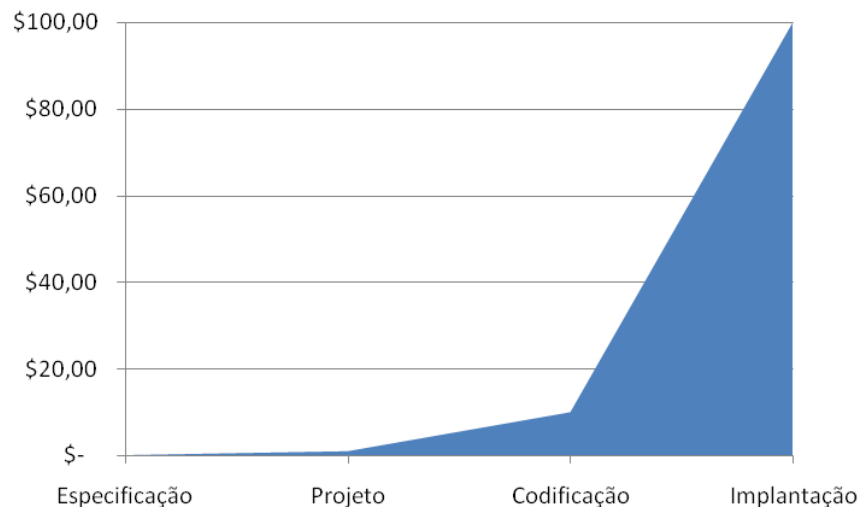


Figura 1 Custo de solução de defeitos por fase de desenvolvimento.

Fonte: Adatado de Moureira Filho e Rios (2003)

O gráfico acompanha o custo decorrente de um defeito, demonstrando que ele cresce em uma proporção média de dez vezes a cada etapa de desenvolvimento. Sendo assim, um erro encontrado na fase de especificação custa aproximadamente dez centavos de dólar para ser corrigido, para o caso do mesmo erro ser esquecido e detectado em fases posteriores, por exemplo, na fase de implantação, o custo aproximar-se de cem dólares.

3. QUALIDADE DE SOFTWARE

De acordo com Pressman (2002), qualidade de software é a conformidade a requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas esperadas de todo software desenvolvido profissionalmente.

Apesar de ser impossível provar que um software está absolutamente correto por meio de testes, a sua utilização fornece evidências da conformidade com as funcionalidades especificadas, segundo Pressman (2002) a atividade de teste não pode mostrar a ausência de *bugs*; ela só pode mostrar se defeitos de software estão presentes. Além disso, uma atividade de teste conduzida de forma sistemática e criteriosa auxilia no entendimento dos artefatos testados e evidencia as características mínimas desejadas do ponto de vista da qualidade do software.

3.1. O custo da qualidade

O Gráfico abaixo demonstra um modelo proposto por Black (2003), em que demonstra a relação da satisfação do cliente com custo da qualidade.

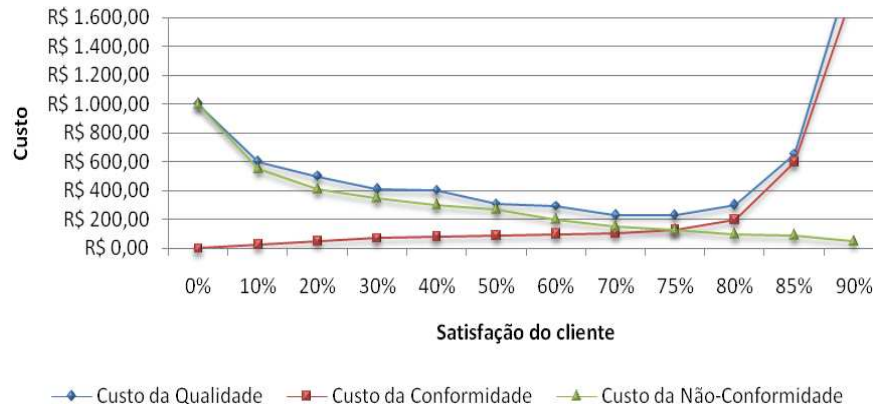


Figura 2 Custo da qualidade.
Fonte: Adaptado de Black (2003)

O Gráfico acima demonstra que a satisfação do cliente aumenta na medida em que ocorre o aumento do custo da qualidade.

Segundo Bartié (2002), o custo da qualidade pode ser entendido como o custo da conformidade mais o custo da não-conformidade. Os custos da conformidade são todos os investimentos realizados para planejar e manter toda uma infra-estrutura com o objetivo de prevenir e detectar erros do processo de desenvolvimento. Tudo que é realizado com a intenção de melhorar e garantir o processo deve ser considerado como custo da conformidade, ou seja, o custo para se obter e garantir qualidade.

Os custos da não-conformidade são todos os custos de atividades ligadas ao esforço de reparar falhas originadas no decorrer do processo de desenvolvimento. Todas as consequências financeiras causadas por estes defeitos devem ser computadas nos custos da não-conformidade. Tudo que é realizado ou gerado em função de defeitos produzidos durante os projetos de software devem ser encarados como não conformidades, ou seja, os custos provenientes da falta de qualidade.

4. TESTE DE SOFTWARE

O processo de desenvolvimento de software envolve uma série de atividades nas quais, apesar das técnicas, métodos e ferramentas empregados, erros no produto ainda podem ocorrer.

Muitos fatores podem ser identificados como causas de tais problemas, mas Atividades agregadas sob o nome de Garantia de Qualidade de Software têm sido introduzidas ao longo de todo o processo de desenvolvimento, entre elas atividades de VV&T – Verificação, Validação e Teste (PRESSMAN, 2002), com a finalidade de garantir que tanto o modo pelo qual o software está sendo construído quanto o produto em si estejam em conformidade com o especificado, visando assim minimizar a ocorrência de erros e riscos associados. Dentre as técnicas de verificação e validação, a atividade de teste é uma das mais utilizadas, constituindo-se em um dos elementos para fornecer evidências da confiabilidade do software em complemento a outras atividades, como por exemplo, o uso de revisões e de técnicas formais e rigorosas de especificação e de verificação.

Nesse cenário, teste de software é uma atividade crítica na garantia de qualidade de software e representa a última revisão de especificação, projeto e codificação. Segundo Moreira Filho; Rios (2003), teste de software é o processo que visa a sua execução de forma controlada, com o objetivo de avaliar seu comportamento com base no que foi especificado, dessa forma, a execução dos testes é considerada um tipo de validação.

4.1. Motivação

O Teste de Software é de fundamental importância para o controle da qualidade dos produtos desenvolvidos, pois certifica que os requisitos especificados estão completamente e corretamente implementados e que o produto irá atender às expectativas dos clientes. Diante desta realidade e tendência de mercado, que quer receber os seus projetos dentro do prazo e com qualidade satisfatória, temos o prazer de contar com esta disciplina.

Podemos citar alguns dados que podem ser determinantes para a adoção de uma prática sistemática e profissional da atividade de teste de software em um ambiente de desenvolvimento:

- Segundo dados do IDC (empresa que é referência na análise do mercado de tecnologia, com atuação em cerca de 90 países), desenvolvedores gastam 50% do seu tempo encontrando e corrigindo erros;
- 80% do custo de desenvolvimento são destinados à identificação e correção de erros – *National Institute of Standards and Technology* (NIST);
- Segundo estudos realizados pelo NIST, a economia norte-americana perde US\$ 60 bilhões por ano com erros de software, e desse total, 20 bilhões poderiam ser economizados com melhorias nos testes;
- Um processo de testes ajustado gera previsibilidade das ocorrências do suporte e uma comunicação mais precisa ao cliente/usuário final;

Uma estrutura formal de testes deve contar com mecanismos e ferramentas bem definidas, como:

- Testadores capacitados;
- Ferramentas de teste;
- Suporte gerencial;
- Estrutura de reporte independente;

As atividades de qualidade de software merecem ter a mesma atenção quanto às atividades de desenvolvimento e ao cronograma do projeto.

Outro fator determinante na adoção de uma estrutura formal de teste é que quanto mais cedo um defeito é identificado, menor será o custo para a correção o que torna vital a avaliação constante da qualidade dos produtos de software durante todo o ciclo de vida de desenvolvimento.

5. CONCLUSÃO

Nos dias de hoje o termo "qualidade de software" está cada vez mais presente como uma das preocupações das organizações. Isso se deve ao fato de que os usuários e os clientes estão mais exigentes e com isso desejam mais qualidade nos produtos de software.

Na medida em que o emprego de sistemas de software cresceu ao ponto em que boa parte de nossa vida depende cada vez mais de software e computadores, passa a ser de vital importância a existência de softwares confiáveis – softwares que forneçam resultados corretos quando alimentados com dados válidos e que identifiquem corretamente dados inválidos, principalmente quando se trata de aplicações onde um simples defeito pode causar um grande prejuízo ou simplesmente uma catástrofe, tais como software de aplicações espaciais, software de controle de processos na área médica, controle de processos em usinas nucleares.

Dessa forma, a atividade de teste de software vem para agregar valor aos processos de desenvolvimento, para que assim a indústria de software reduza os custos de desenvolvimento e manutenção de seus produtos como também aumentar a qualidade dos mesmos, uma vez que os testes têm por objetivo antecipar a manifestação de defeitos existentes, para que assim possam ser corrigidos o mais rapidamente possível com o mínimo de esforço, de preferência antes de chegar ao usuário final, e conseqüentemente, satisfazer os anseios do mercado de software.

REFERÊNCIAS

BARTIÉ, A. **Garantia da qualidade de software: adquirindo maturidade organizacional**. Rio de Janeiro: Campus, 2002.

BLACK, R. **Critical Testing Processes : Plan, Prepare, Perform, Perfect**. United States: Addison-Wesley, 2003.

DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. **Introdução ao Teste de Software**. Rio de Janeiro: Campos, 2007.

INTHURN, C. **Qualidade & teste de software**. Florianópolis: Visual Books, 2001.

LEVESON, N. **An Investigation of the Therac-25 Accidents**. Disponível em:
<http://courses.cs.vt.edu/~cs3604/lib/Therac_25/Therac_1.html> Acesso em 26 jun 2008.

LIONS, J. **Ariane 5 Flight 501 Failure**. Paris, 1996. Disponível em:
<<http://sunnyday.mit.edu/accidents/Ariane5accidentreport.html>> Acesso em 26 jun 2008.

MOLINARI, L.. **Testes de software: produzindo sistemas melhores e mais confiáveis**. São Paulo: Érica, 2003.

MOREIRA FILHO, T. R.; RIOS, E. **Projeto & engenharia de software: teste de software**. Rio de Janeiro: AltaBooks, 2003.

NEWMAN, M. E. **Software Errors Cost U.S. Economy \$59.5 Billion Annually**. Disponível em:
<http://www.nist.gov/public_affairs/releases/n02-10.htm> Acesso em: 1 ago 2008.

PAULA FILHO, W. P. **Engenharia de software: fundamentos, métodos e padrões**. Rio de Janeiro: LTC, 2001.

PETERS, J. F.; WITOLD, P. **Engenharia de software**. Rio de Janeiro: Campus, 2001.

PRESSMAN, R. S. **Engenharia de software**. São Paulo: Makron Books, 2002.