

Testes e Qualidade em Jogos

5º período

Professora: Michelle Hanne

Sumário

- Introdução aos Testes de Softwares

O que é um Teste de Software?

Teste é o processo de executar um programa

- com o intuito específico de encontrar erros;
 - antes de sua entrega ao cliente ou usuário final.
- Glenford J. Myers (1979)

Testar é uma atividade criativa;

- de geração de ideias;
- muito mais que uma atividade de implementação.
- Testar é uma busca heurística;
 - em um espaço aberto e multidimensional.
 - Cem Kaner (2001)

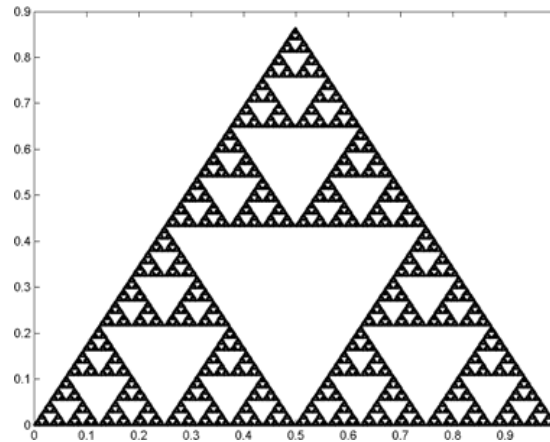
O que é um Teste de Software?

Teste é um elemento da garantia da qualidade do *software*.

É a única técnica de validação de requisitos não-funcionais.

Exercício

- Um programa lê 3 inteiros. Os três valores são interpretados como os comprimentos dos lados de um triângulo. O programa imprime uma mensagem que indica se o triângulo é escaleno, isósceles ou equilátero.*
- Qual é o conjunto de testes que você acha necessário realizar para verificar este programa?



* Extraído de Glenford J. Myers, *The Art of Software Testing*, Willey Interscience, 1979.

Algumas Respostas

1. Você tem um caso de teste que representa um triângulo escaleno válido?
(Note que casos de testes como 1,2,3 ou 2,5,10 não ganham um “sim” nesta pergunta, pois não existem triângulos com estes lados.)
2. Você tem um caso de teste que representa um triângulo equilátero válido?
3. Você tem um caso de teste que representa um triângulo isósceles válido?
(Um caso de teste como 2,2,4 não pode ser contado.)
4. Você tem no mínimo 3 casos de testes que representam um triângulo isósceles válido para os quais você tenha tentado as três permutações possíveis? (Ex.: 3,3,4; 3,4,3; 4,3,3)

Algumas Respostas

5. Você tem um caso de teste no qual um dos lados é zero?
6. Você tem um caso de teste no qual todos os lados são iguais a zero?
7. Você tem um caso de teste no qual um dos lados tem o valor negativo?
8. Você tem um caso de teste com 3 inteiros maiores que zero, onde a soma de dois lados é igual ao terceiro lado?
(Se o programa disser que 1,2,3 representa um triângulo escaleno, ele contém um erro.)
9. Você tem no mínimo 3 casos de teste da categoria 8 para os quais você tentou as três permutações? (Ex.: 1,2,3; 1,3,2 e 3,1,2)

Algumas Respostas

10. Você tem um caso de teste com 3 inteiros maiores que zero, onde a soma de dois lados é menor que o terceiro lado? (Ex.: 1,2,4 ou 12,15,30)
11. Você tem no mínimo 3 casos de teste da categoria 10 para os quais você tentou as três permutações? (Ex.: 1,2,4; 1,4,2 e 4,1,2)
12. Você tem um caso de teste especificando valores não-inteiros?
13. Você tem um caso de teste especificando uma quantidade errada de valores? (Ex.: 2 ou 4, ao invés de 3 inteiros)

Discussão Sobre o Exercício

- Como ponto de referência:
 - programadores altamente experientes obtêm, em média, somente 7,8 pontos do total de 13.
- Mesmo o teste de um programa trivial como este não é uma tarefa fácil;
 - imagine a dificuldade de se testar um sistema de controle aéreo com mais de 100.000 linhas de código!
- Obviamente, um conjunto de casos de testes que satisfaça às condições anteriores;
 - pode revelar a presença de erros;
 - mas não garante que todos os erros possíveis foram encontrados;
 - nem a ausência de defeitos no *software*.

O que é um Teste de Software?

Testar não é somente verificar e validar o *software*.

Um *software* deve funcionar não somente quando as coisas vão bem;

- ele deve funcionar quando as coisas vão mal também.

Deve-se testar a capacidade do *software* de manipular eventos inesperados;

- isto demonstra a qualidade do *software*.

O que é um Teste de Software?

Testar é o processo de executar um programa;

- com a intenção de descobrir uma falha.

Um bom caso de teste é aquele que tem elevada probabilidade de revelar uma falha ainda não descoberta.

Um teste bem-sucedido é aquele que revela uma falha ainda não descoberta.

A atividade de testes nunca está satisfeita com a resposta correta.

O que é um Teste de Software?

Em um primeiro momento, testar é uma atividade destrutiva;

- mas tem o mesmo objetivo do desenvolvimento.

O teste exaustivo é geralmente impossível;

- mesmo para produtos relativamente simples.
- A realização de testes, quase sempre, é limitada por restrições de cronograma e orçamento.
 - Por isto, é importante planejar e desenhar testes;
 - antes de executá-los;
 - que consigam o melhor proveito possível dos recursos alocados para eles.
- Focaliza-se o desenho de testes que tenham alta probabilidade de descobrir falhas;
 - com o mínimo esforço.

O que é um Teste de Software?

Em última instância, o objetivo da atividade de teste é:

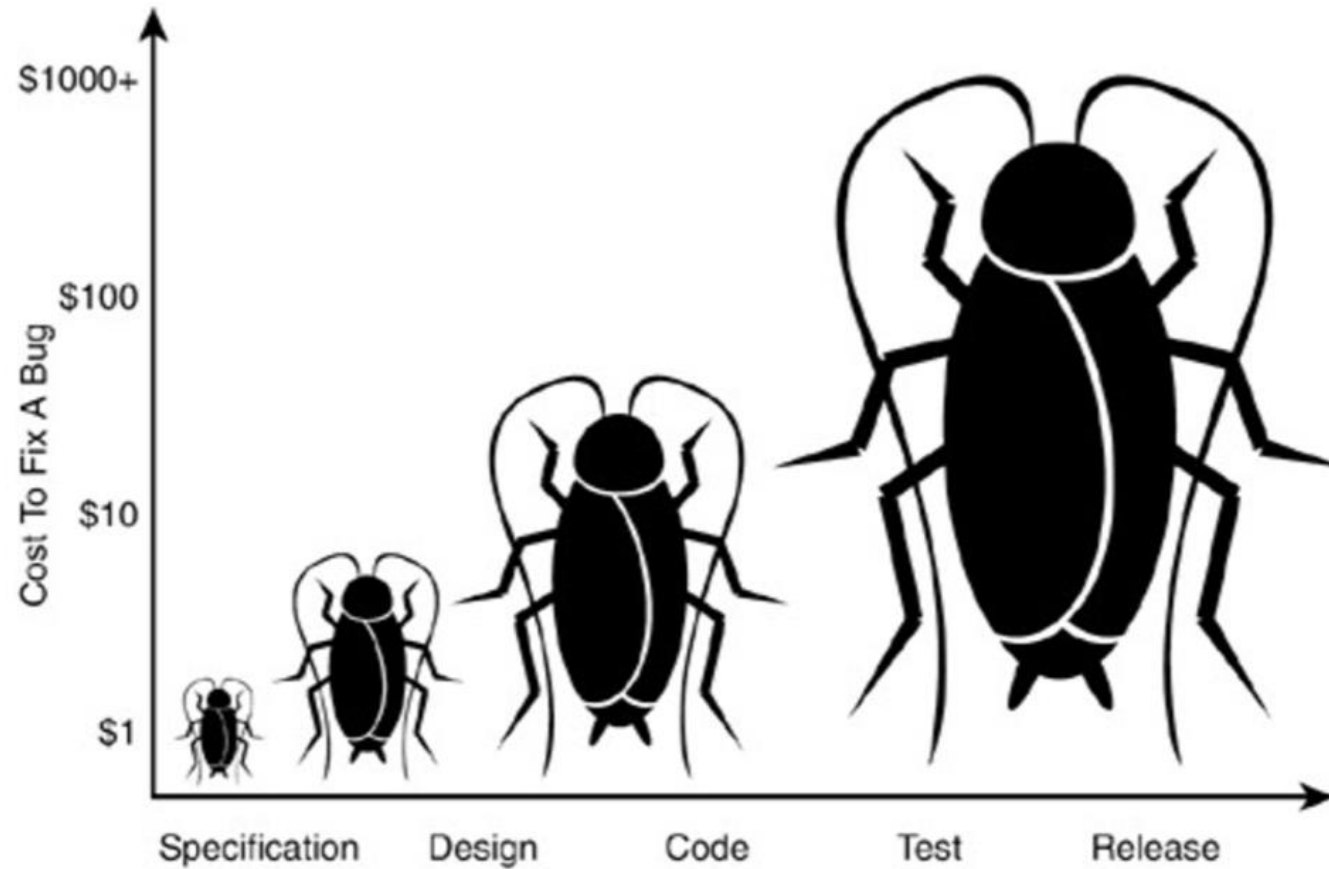
- planejar;
- desenhar;
- e realizar testes;
 - que descubram sistematicamente diferentes classes de falhas;
 - com uma quantidade mínima de tempo e esforço.

Encontrando Erros

Erros podem ser encontrados de várias formas (da melhor para a pior):

- pelo próprio desenvolvedor;
 - quando ele está avaliando e estudando o problema pela primeira vez;
- pelo próprio desenvolvedor;
 - quando ele está tentando fazer com que o código funcione;
- por outro desenvolvedor ou testador do grupo;
- por esforços de inspeções de código;
- como parte de um esforço de testes do produto;
- pelo usuário final.

Custo para se reparar um erro



- A estimativa é de que a cada 1000 linhas de código criadas, 70 novos bugs são criados e 15 serão descobertos pelo seu cliente.
- E se isso de fato acontecer, cada bug corrigido em sistemas que estão em produção custará 30 vezes mais (tempo e dinheiro) para serem corrigidos do que se tivessem sido descobertos na fase de desenvolvimento.

Custo para se reparar um erro

Se nós encontrarmos os erros na especificação;

- estes poderão ser corrigidos sem a necessidade de recodificação.

Programadores que encontram seus próprios erros;

- podem consertá-los sem gastar tempo registrando os mesmos ou explicando-os a alguém.
- Além disto, o grupo de testes não consegue desempenhar seu papel;
 - se houver erros básicos em componentes.

É extremamente caro lidar com erros em sistemas que já estão em produção.

Tempo para se Reparar um Erro

Testar o código o mais cedo possível;

- tende a diminuir;
- em muito;
- o retrabalho.

Estima-se que, em média, 40% a 50% do tempo total de desenvolvimento de *software*;

- seja gasto com retrabalho evitável.

Valor Agregado ao Teste

Um erro encontrado por testes de *software* tem quatro custos associados:

1º custo:

desenvolvimento do programa de forma incorreta;

2º custo:

realização de testes que detectaram os erros;

3º custo:

correção de especificação/modelos/códigos;

4º custo:

realização de testes novamente para verificar a correção.

Considerações

Testar é um estado de espírito:

- “Se o nosso objetivo é mostrar a ausência de erros;
 - nós vamos encontrar poucos deles.
- Se o nosso objetivo é mostrar a presença de erros;
 - nós vamos descobrir um grande número deles.”
- Glenford J. Myers (1979)

Considerações

Desta forma, não é interessante que os desenvolvedores sejam os únicos responsáveis pelos testes de seus códigos;

- eles têm um envolvimento “afetivo” com o código;
- testadores, por outro lado, não!

Envolvimento dos Usuários nos Testes do Software

O envolvimento dos usuários no processo de validação do *software*;

- é extremamente importante.

Quando o usuário é envolvido;

- há ganhos expressivos em termos do entendimento de suas necessidades;
- e da forma como ele tende a usar o *software*.

Testes Manuais

Uma das grandes vantagens de se realizar testes manuais;

- é a exploração de novas possibilidades de se encontrar erros.

É uma parte criativa da atividade de testes.

Não deve ser abandonada;

- mesmo porque existem situações em que a automação não pode ser feita;
- ou tem um custo que não a justifica.

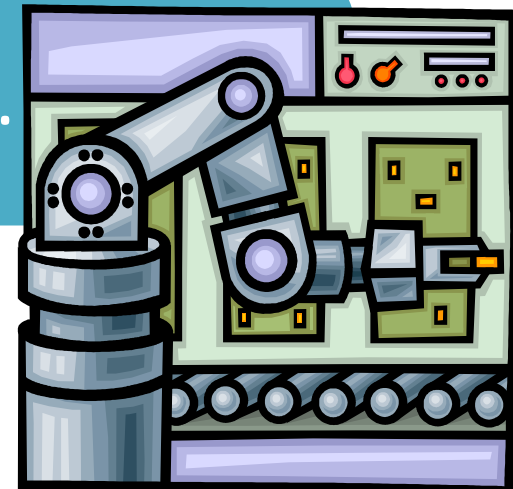
Automação de Testes de Software

A automação de testes tem um papel importante na garantia da qualidade do *software*.

A automação facilita a realização de testes de regressão.

Existem ferramentas para auxiliar na automação de testes.

- No mundo .NET, a mais popular é o *NUnit*.
- No mundo Java, a mais popular é o *JUnit*.



Baterias de Testes

Agrupam testes.

Devem ser confeccionadas de forma a agrupar testes que têm objetivos, importâncias e características similares;

- para que sejam executados em conjunto.

Referências

PAULA-FILHO, Wilson de Pádua. *Engenharia de Software: Fundamentos, Métodos e Padrões*. 2ª edição, Rio de Janeiro: LTC – Livros Técnicos e Científicos, 2003. Capítulo 8.

PRESSMAN, Roger S.. *Engenharia de Software*. 5ª edição, Rio de Janeiro: McGraw Hill, 2017. Capítulo 17.

MYERS, Glenford J.. *The Art of Software Testing*. Willey Interscience, 1979.

ROSE, Laura. *Myths and Realities of Iterative Testing*. <http://www-128.ibm.com/developerworks/rational>, Abril, 2006.

BOEHM, B. W.. *Software Engineering*. IEEE Transactions on Computers, C-25(12):1226–1241, 1976.

BAZIUK, W.. *Path to Improve Product Quality, Reliability, and Customer Satisfaction*. 6th International Symposium on Software Reliability Engineering: Toulouse, France, Outubro, 1995.