



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS V – DIVINÓPOLIS - TÉCNICO EM INFORMÁTICA – 3º B
DISCIPLINA PROJETO DE SISTEMAS

Teste de Software

Bárbara Emmanuelle Lacerda de Moura
Iann Gabriel de Lima
Lorna Carolina Araújo de Sousa

Professor: Michel Pires

Divinópolis, 17 de Setembro, 2015.

Testes de Software

O teste de software é um processo de execução de um produto para verificar se ele está funcionando corretamente e se atingiu todas os requisitos para o qual foi projetado. Com esses testes, é possível encontrar falhas, de maneira forma simples, que futuramente serão corrigidas a fim de manter o funcionamento correto do produto.

Nos softwares podem ser encontrados defeitos, um ato cometido pelo indivíduo ao tentar entender uma determinada informação, erros, um estado intermediário incorreto ou um resultado não esperado ao executar o software e falhas, quando o software funciona de maneira diferente ao esperado pelo usuário.

E para detectar os erros, várias técnicas podem ser usadas, entre elas a técnica estrutural e funcional. A primeira consiste em uma avaliação do comportamento interno do software. Já a segunda, testa o funcionamento do software, sem considerar qualquer comportamento interno.

1. Teste Unitário

O Teste Unitário, é um tipo de atividade que visa testar pequenas partes ou unidades do sistema. O ponto foco desse tipo de teste, são os métodos dos objetos ou mesmo pequenos trechos de código. Assim, ele procura encontrar falhas dentro de uma pequena parte do sistema funcionando independentemente do todo.

2. Teste de Componente

O Teste de Componente, possui um universo um pouco maior ao teste unitário. Seu propósito é testar o componente como um todo e não apenas as suas funções ou métodos. Mesmo assim, o teste continua a ser executado sem considerar a

Interação com outras partes do sistema, ou seja, leva-se apenas em consideração o componente a ser testado e nenhuma outra entidade do sistema.

3. Teste de Sistema

O terceiro é o Teste de Sistema, que pelo qual têm grande importância para o sucesso do software. Sua principal filosofia é varrer o sistema em busca de falhas através da sua própria utilização, como se ele fosse o usuário final. Assim, ele busca falhas no uso do dia-a-dia.

4. Teste de Funcionalidade

Já o Teste de Funcionalidade verifica se as funcionalidades descritas na especificação de requisitos serão executadas pelo sistema integrado.

5. Teste de Aceitação

Outro teste é o Teste de Aceitação. Que são realizados geralmente por um restrito grupo de usuários finais do sistema. Esses simulam operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado.

6. Teste de Integração

Outro teste é o Teste de Integração, que visa encontrar falhas provenientes da integração e componentes do sistema. Geralmente as falhas encontradas são de envio e recebimento de dados. Um exemplo sobre isso é quando A está esperando o retorno de Y e recebe o retorno X.

Dentre todos os testes citados acima, o Teste de Integração foi escolhido para ser utilizado na verificação do projeto SALVA - VIDAS.

Teste de Integração

Os testes de integração são os testes das interações entre componentes de software, ou seja, é o caminho no qual o teste é conduzido para integrar componentes no sistema. É o processo que verifica se os componentes do sistema trabalham juntos conforme descrito nas especificações do sistema e do projeto do programa.

O objetivo deste teste é encontrar falhas provenientes da integração interna dos componentes de um sistema. Não faz parte do escopo deste teste, o tratamento de interfaces com outros sistemas, que são testadas na fase de teste de sistema.

JUnit

O JUnit é mais conhecido nos testes unitários, ou seja, testar uma única funcionalidade do seu sistema. Mas também pode ser utilizado para testes de integração.

JUnit é um framework de código aberto inicialmente desenvolvido por Kent Beck e Erick Gamma, que permite a criação de testes automatizados utilizando Java, facilita a criação de código para os testes, além de apresentar os resultados do teste de forma bastante clara e visível. Caso os testes forem executados com sucesso, apresenta-se uma

barra verde ao desenvolvedor, senão uma barra vermelha com uma indicação de onde ocorreu o erro no teste será mostrada.

Para utilizá-lo, basta criar uma classe que estenda `junit.framework.TestCase` e definir cada método a ser testado, sendo que este seja público, sem retorno de argumentos e que seu nome tenha o prefixo `test`, como por exemplo `testXxx()`. Utiliza-se também, a notação `@test` em cima do método tornando-o assim um método testável. Por exemplo:

```
import junit.framework.*;

class SuaClasseDeTeste extends TestCase {
    public int testSeuTeste() {
        ...
    }
}
```

Além disso, o JUnit tem um excelente suporte, uma vez escritos, todos os testes são executados rapidamente sem atrapalhar o processo de desenvolvimento. Checando os resultados dos testes e fornecendo as respostas ao programador.

Para a instalação do Junit, basta adicionar o arquivo `junit.jar` no classpath para compilar e rodar os programas de teste. Mas caso utilize o Eclipse, Jbuilder ou Netbeans, já estarão configurados.

Para se fazer o teste é necessário levar em conta qual funcionalidade do sistema precisa de uma verificação.

Conclusão:

Pode-se concluir, portanto, que os testes de software são muito importantes para encontrar erros, defeitos ou falhas no sistema e devem ser considerados como última etapa em qualquer projeto de desenvolvimento de software.

Levando em consideração as funcionalidades propostas no projeto SALVA-VIDAS, o teste de software que mais se adapta as necessidades do projeto, é o Teste de Integração.

Referências

- JUnit: <<http://junit.org/index.html>>. Acessado em 12/09/2015.
- Spinola, Eduardo Oliveira. **Testes de unidade com JUnit**. Disponível em: <<http://www.devmedia.com.br/testes-de-unidade-com-junit/4637>>. Acessado em 12/09/15.
- Spinola, Eduardo Oliveira. **Testes de Integração com Java e JUnit**. Disponível em: <<http://www.devmedia.com.br/testes-de-integracao-com-java-e-junit/25662>>. Acessado em 12/09/15.
- Artigo Engenharia de Software – Introdução a Teste de Softwar. Disponível em: <<http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-teste-de-software/8035>>. Acessado em 15/09/15.