O PROCESSO DE TESTE DE SOFTWARE

Josenilson dos Santos Silva
Grazielle Bandeira Viana
Giselle Barbosa Gomes Machado
Rogério Oliveira da Silva

Resumo: Teste de Software é o processo de executar programas, com o objetivo de achar defeitos. Um projeto de teste tem que ser bem organizado, terá como participação no plano do projeto de teste, o gerente de projeto, engenheiros de software, ou especialistas em testes. O Teste de Software não é a garantia de um software totalmente livre de defeitos, porém é necessário e muito importante que se tenha teste em todo o projeto de um software, pois minimiza bastantes os defeitos e ajuda na parte de custos. Quanto mais um software for testado, melhor será o seu desempenho e menor será o custo e provavelmente terá menos riscos de ocorrer problemas na execução, mais claro que, para um teste ser bem-sucedido, precisa de uma equipe bem organizada e um plano de projeto bem elaborado. Esse artigo tem como objetivo, mostrar as técnicas de software, as práticas de teste e a parte de documentação do projeto e etc.

Palavras Chave: Teste; Qualidade; Software; Engenharia.

Abstract:: Software testing is the process of running programs in order to find defects. A test *project has to be well organized, will participate in the test project plan, the project manager, software engineers, and testers. Software testing is not guaranteed a completely defect free software, but it is necessary and very important to make testing throughout the design of a software because it minimizes enough defects and help in the costs. The more software is tested, the better the performance and lower the cost and probably will have less risk of problems occur in the implementation, clear that, for a test to be successful, you need a well-organized team and a plan project well prepared. This article aims to show the software techniques, test practices and project documentation part and etc.*

Keywords*: Testing; Quality; Software; Engineering.*

1. Introdução

Teste de software é o processo de execução de um produto para determinar se ele atingiu suas especificações e funcionou corretamente no ambiente para o qual foi projetado. O seu objetivo é revelar falhas em um produto, para que as causas dessas falhas sejam identificadas e possam ser corrigidas pela equipe de desenvolvimento antes da entrega final. Por conta dessa característica das atividades de teste, dizemos que sua natureza é "destrutiva", e não "construtiva", pois visa ao aumento da confiança de um produto através da exposição de seus problemas, porém antes de sua entrega ao usuário final.

Conceitualmente, o teste de software pode ser compreendido através de uma visão intuitiva ou mesmo de uma maneira formal. Existem atualmente várias definições para esse conceito. De uma forma simples, testar um software significa verificar através de uma execução controlada se o seu comportamento corre de acordo com o especificado. O objetivo principal desta tarefa é revelar o número máximo de falhas dispondo do mínimo de esforço, ou seja, mostrar aos que desenvolvem se os resultados estão ou não de acordo com os padrões estabelecidos.

Com continuidade sobre o assunto,destaca-se um problema que está muito relacionado, conhecido como Bug. Em 1947, surgiu o primeiro Bug encontrado em computadores, onde os engenheiros encontraram um inseto nos circuitos da máquina Harvard Mark I. Este inseto estava causando um erro nos cálculos da máquina. Ao ser encontrado, o inseto foi retirado e colado no livro de registro com a intenção de registrá-lo como o primeiro Bug encontrado.

Com o livro "The art of software testing" Glenford Myers, trouxe ao leitor um conhecimento mais avançado sobre testes, abordando sobre as técnicas de software no ano de 1979. Este livro é muito conhecido pelos profissionais, podendo ser tratado como uma referência teórica, com o ano de 1979 um grande passo para a área. O livro possui uma coletânea de materiais sobre testes de software. Atualmente este livro se encontra em sua 2º edição e é utilizado até os dias de hoje, onde muitos os consideram como o marco inicial de Testes de Software.

Philippe Kruchten, em seu livro "The rational unified process: Na introduction (Boston: Addison-Wesley, 1999) ", propõe que existem pelo menos três dimensões de qualidade que precisam ser consideradas antes de se iniciar qualquer ciclo de teste: Confiança, Funcionalidade e Performance. Estas três dimensões proporcionam um sistema que resiste a falhas durante a execução, se comporta conforme foi especificado e realiza um tempo de resposta adequado e aceitável.

Antigamente os testes eram feitos pelos próprios desenvolvedores do sistema. A tarefa de efetuar testes em software foi considerada pouco importante por muito tempo. Geralmente a prática de teste era vista como um castigo para os programadores, ou como uma tarefa onde não se deveria gastar muito com tempo e investimentos. Porém as coisas mudaram, com a globalização e a competição entre as empresas de software está havendo uma grande preocupação em aprimorar e aperfeiçoar os processos de testes em desenvolvimento de software com vistas a reduzir custos com manutenção e produzir um produto de boa qualidade.

Então, vimos que o teste não é só mais uma atividade dentro do ciclo de vida de um projeto, o teste passa a ser uma atividade muito importante, e levando em consideração essas afirmações, as empresas passam então a, montar uma equipe capacitada para os testes e a atenderem todos os critérios de qualidade.

2. Os processos de teste

O Processo de Testes de Software representa uma estruturação de fases, atividades, artefatos, papéis e responsabilidades que tem o objetivo de padronizar os trabalhos, além de maximizar a organização e monitoramento dos projetos de testes. Portanto, o objetivo do processo de teste é fornecer informação para garantir a qualidade do produto, decisões e processos para uma atividade de teste. Além disso, o processo de teste busca garantir que nenhum passo crítico do processo seja esquecido, ou seja, que todas as atividades sejam realizadas na ordem correta.

Vimos que o teste é muito importante para um software, porém á condições em que o teste precisa ser interrompido. A interrupção deve acontecer assim que o custo dos testes superar o custo da falha para o negócio.Sempre haverá um ponto de equilíbrio para definir a interrupção dos testes, nem sempre é fácil de encontrar esse ponto, mas muitos especialistas informam que o melhor a serfeito, é interromper os testes quando o intervalo de ocorrência de defeitos começarem a aumentar muito, por exemplo, de horas para dias. Para isso usa-se os termos under test e over test, que significa "falta de teste" e "excesso de teste".

Os testes sempre reduzem a probabilidade de defeitos encontrados no software, mesmos que nenhum erro seja encontrado, isso não garante a conformidade. Os testes podem ser bem exaustivos, mesmo utilizando as ferramentas de automação, com isso é impossível se afirmar que tudo foi testado. Muitas vezes as empresas usam como testador os analistas de sistema, um erro, pois para testar um software deve ter profissionais capacitados, usando metodologia apropriada, em ambiente adequado e ferramentas de automação na maioria dos casos.

Quanto mais rápido se inicia o teste menor será o custo, Barry Boehm, em seu livro Software engineering, publicado em 1976, afirma que o custo de um erro detectado nos requisitos é X; se encontrado na fase de teste, seu custo passa a ser 10X; e se encontrado na produção, 100X. Com isso, como mencionado antes, quanto mais rápido encontrarmos erros, mais barato fica para corrigir. Estudos estimam que de 80% dos defeitos, 20% pertence ao código, ao identificar essa parte de erro, pode se priorizar o teste enquanto outros erros são encontrados.

É importante ressaltar que os testes precisam ser revisados com frequência, pois se os mesmos testes sejam aplicados repetidamente, em determinado momento eles deixaram de ser úteis, com isso não se encontrará mais erros, prejudicando então a eficiência que deverá ter o software. Porém com todos esses princípios é importante sempre que o software atenda sempre as necessidades do usuário, resumindo, um software deve ser testado sempre.

Há também quem diga que o testador é um inimigo do desenvolvedor, porém não existe essa teoria no mundo de desenvolvimento de software, onde uns querem mostrar que tem mais capacidade ou entende mais quer o outro, a questão é, os mesmos deverão ter entendimento que a empresa precisa de desenvolver produto com a melhor qualidade possível, em outras palavras, passíveis erros.

3 Ciclo de Vida do Processo de Teste

Como mencionado antes, quanto antes os testes iniciarem, mais barato será corrigir os defeitos encontrados.Para que isso aconteça todo teste deve ter um ciclo de vida.

Etapas do teste:

1- **Procedimentos inicias**

   Nesta etapa deverá ser aprofundado um estudo dos requisitos do negócio que dará origem ao sistema de informação a ser testado, de modo a garantir que o mesmo esteja completo e sem nenhuma ambiguidade. No final desta etapa é elaborado o GOT – Guia Operacional de Teste.

2- **Planejamento**

   Consiste em elaborar a Estratégia de Teste e o Plano de Teste a ser utilizados de modo a minimizar os principais riscos do negócio e fornecer os caminhos para as próximas etapas.

3- **Preparação**

   Nesta etapa, o objetivo básico é preparar o ambiente de teste (equipamentos, pessoal, ferramentas de automação, hardware e software), para que os testes sejam executados corretamente.

4- **Especificação**

É elaborado e revisado os casos de teste e os de roteiros teste.Os casos de teste e os roteiros de teste devem ser elaborados dinamicamente durante o decorrer do projeto de teste.

5- **Execução**

Os testes deverão ser executados de acordo com os casos de teste e os roteiros de teste.Devem ser usados scripts de teste, caso seja empregada alguma ferramenta de automação de testes.

6- **Entrega**

Esta fase é onde acontece a entrega do software depois de ter passado por todas as fases, com objetivo de minimizar todas as chances possível de erros (bugs).

Essa representação de etapas de teste é uma estrutura básica, existem outras, porém a ideia é sempre a mesma mostrada à cima. Para que o teste der certo, desde o início é preciso ter uma equipe alocada para o trabalho, a equipe deve ser dividida em cada fase de teste para que o teste seja bem-sucedido.

4. Técnicas de Teste

Existem várias maneiras de testar um software, todas as maneiras terão um mesmo objetivo. Técnica é o processo que vai assegurar perfeito funcionamento de alguns aspectos de software ou de sua unidade. Separado em dois grupos temos as principais técnicas de testes, são elas, Estrutural e Funcional.

Estrutural

O Teste estrutural também conhecido como teste da caixa branca, tem por objetivo testar o código fonte, testar cada linha de código possível, testar os fluxos básicos e os alternativos. Nesse teste podemos classificá-los em alguns itens como:

- **Teste de estresse:** Verifica como o sistema é executado com determinados volumes de dados.
- **Teste de execução:** Se o sistema atinge o nível desejado de eficiência.
- **Teste de recuperação (contingência):** Se o sistema é capaz de retornar o nível anterior antes da falha.
- **Teste de operação:** Se o sistema opera conforme sua documentação.
- **Teste de conformidade:** Se o sistema foi desenvolvido conforme padrões e procedimentos.
- **Teste de segurança:** Se o sistema está protegido conforme normas e políticas da organização.

Funcional

O teste funcional conhecido como teste da caixa preta, é baseado na análise funcional do software ele garante que os requisitos funcionem conforme o especificado, ele não se preocupa na forma como ele foi implementado, é inserido alguns dados e espera-se na saída o

resultado de como foi projetado os requisitos. Nesse teste podemos classificá-los em alguns itens como:

- **Teste de requisitos**: Se o sistema é efetuado conforme suas especificações
- **Teste de regressão**: Se o sistema ou alguma parte dele foi afetado por alguma alteração.
- **Teste de tratamento de erros**: Se os possíveis erros que possam acontecer têm tratamento antes de acontecer suas falhas.
- **Teste de suporte manual**: Interação entre homem e máquina.
- **Teste de interconexão com outros softwares**: Verifica se o sistema troca informações com outros sistemas.
- **Teste de controle**: Verifica se o sistema tem algum controle de dados, validações e integridade, logs de auditoria.
- **Testes paralelos**: Se a versão atual ou a antiga gera os mesmos resultados

5. Ambiente de teste

Ambiente de teste é toda a infraestrutura onde o teste será executado, compreendendo configurações de hardware, software, ferramentas de automação, equipe envolvida, aspectos organizacionais, suprimentos, rede e documentação. Sua finalidade é proporcionar a realização de testes em condições conhecidas e controladas. A figura a baixo representa o ambiente e seus componentes:

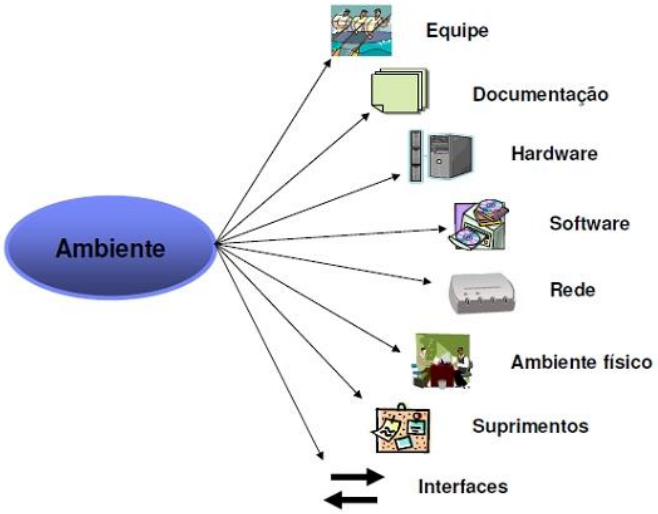

Figura 1 - Ambiente de Teste[1]

5.1 Preparação do ambiente

Temos a seguir a distribuição do ambiente por atributos:

- Escopo
- Equipe

[1] FONTE: matera.com – Google imagens.

- Volume de dados
- Origem dos dados
- Interfaces
- Ambiente

6. Ferramentas de suporte ao teste de software

Ao longo dos anos, à medida que o teste vem ganhando mais espaço no mercado, diversas ferramentas têm surgido visando à melhoria no gerenciamento e na execução das atividades voltadas para o Teste de Software.

A utilização de ferramentas é um importante aliado para avaliar se o software está exposto a vulnerabilidades. Independente da função, o objetivo gira em torno da simplificação das atividades, controle, e, principalmente, garantia da qualidade em cada uma das etapas de um Processo de Teste. Outro fator relevante é que elas permitem sobretudo, minimizar os riscos, inconformidades e uma série de falhas que podem ser detectadas durante a execução dos testes.

Além de simplificar as atividades, a adoção dessas ferramentas pode aumentar consideravelmente o número de defeitos encontrados, e consequentemente agilizar os testes, aprimorando a qualidade das aplicações. Elas também podem levar a melhorias na confiabilidade do software, tornando os testes mais eficazes e produtivos.

As ferramentas são classificadas de acordo com as atividades de teste que elas suportam. Seja para o gerenciamento ou para execução de um teste, elas são categorizadas de acordo com os tipos de teste que executam. Os tipos de teste, unitário, segurança, sistema, ambiente, integração e desempenho, possuem diferentes funções e cada um deles visa avaliar uma determinada característica do software. Um fator interessante é que há ferramentas de suporte que contemplam praticamente todos os tipos de teste existentes.

Nos dias atuais, algumas ferramentas suportam várias atividades, como gestão dos testes, automação e controle dos defeitos, enquanto outras suportam apenas uma. Mesmo assim, apresentá-las intitulando uma ou outra como “a melhor” não é possível. Esse título vai depender de uma série de fatores como, por exemplo, investimento disponível, aplicabilidade, tipo de software a ser testado, tamanho da equipe e expectativa de retorno. Em função disso, é interessante analisar as opções disponíveis para que seja possível definir qual se enquadra melhor às suas expectativas.

7. Diferença entre ferramentas e técnicas de teste

Ferramentas e técnicas de testes são muito importantes para o desempenho do processo de teste. A pessoa que irá testar precisar conhecer primeiro as técnicas de teste para depois entender quais ferramentas devem ser usadas com cada técnica. É tão importante o testador conhecer as técnicas de teste, o quanto para um médico ter primeiro o conhecimento da doença, para só então, poder fazer medicação correta conforme o problema apresentado ao paciente. Assim funcionam com o testador, primeiramente obter os conhecimentos necessários das técnicas, para então usar a ferramenta certa.

Algumas técnicas são manuais e outras automáticas; algumas fazem testes estáticos, e outras, os dinâmicos.Outras avaliam a estrutura do sistema e sua função, na questão das ferramentas algumas são extremamente técnicas e demandam uma noção aprofundada de programação e do sistema a ser testado. Outras são genéricas por natureza e podem ser úteis

para quase todos os profissionais da área de teste. Existem ferramentas para criar dados de teste e para a validação dos testes.

8. ELABORAÇÃO DO TESTE

Na fase de elaboração do teste, a atividade principal é a elaboração dos cenários de testes, que serão testados na fase de execução. O cenário de teste é um caminho que deve ser seguido ou a situação que deve ser testada, o cenário que serve de base para o teste é descrito numa especificação do sistema, por exemplo, na Unified Modeing Languge (UML), é o caso de uso. O caso de teste é o cenário a ser executado para verificar se o que foi especificado está devidamente implantado.

8.1 Documentação do Teste

Um analista de teste gasta em média 50% a 60% do seu tempo em documentação do teste. A norma IEEE 829-1998[2]é um padrão que define a estrutura de vários documentos relativos a testes de software. A escolha dos documentos que darão o suporte suficiente aos testes depende da complexidade deste, da equipe de testes disponível e do tempo dedicado aos testes durante o desenvolvimento do software. Os documentos definidos pela norma cobrem as tarefas de planejamento, especificação ou elaboração e análise dos resultados.

Plano de Teste

O plano de testes é o documento principal dos testes de software. Nele estão contidas informações importantes sobre as partes envolvidas no teste, os objetivos do teste, as partes do software a serem testados, os critérios de aceitação e os passos necessários para executar os testes.

Cada plano de testes possui descrição de um ou mais casos de teste. Um caso de teste compreende no mínimo um conjunto de ações a serem executadas e um critério de aceitação, que diz se o teste foi bem-sucedido ou não. Na maioria das vezes os casos de teste possuem também critérios de entrada ou requisitos, e vem acompanhado de uma breve descrição do item a ser testado e dos objetivos que ele busca atingir.

À medida que a disciplina de testes de software foi se profissionalizando nos últimos anos, novas seções foram adicionadas aos planos de teste, como sumário de alto nível e lições aprendidas. Estas seções tem o intuito de situar o plano de testes e a equipe de testes com relação a outras equipes e ao planejamento do projeto como um todo.

Especificação de Projeto de Teste

É o detalhamento da abordagem apresentada no plano de teste que identifica as funcionalidades e as características que serão testadas pelo projeto. Estes documentos também apontam os casos e os procedimentos de teste, se existirem, e apresenta os critérios de aprovação.

Especificação de Caso de Teste

[2] FONTE: Institute of Electrical and Electronics Engineers IEEE STD 829 Standard for software testing documentation (Nova York:IEEE Computer Society; 1998).

Uma Especificação de Casos de Testes deve contemplar os casos de teste válidos e inválidos. Casos de teste inválidos são aqueles que exercitam a entrada de valores inválidos nos campos do sistema em teste. Segundo Bastos et al. (2007), a utilização dos casos de uso como referência para a geração de casos de testes é uma regra bem comum no mercado. Nem sempre existe tempo/recursos para utilização das técnicas e os testes são focados na validação de regras de negócio.

Especificação do procedimento de teste

Identifica todos os passos necessários para a operação do sistema e o exercício dos casos de teste especificados, onde cobre o projeto de teste planejado. Os procedimentos de teste formam um documento separado, que se espera que seja seguido passo a passo, sem imprevistos.

9. Execução dos testes

Conforme mencionado ao longo desse artigo que o teste é extremante importante e devem ser testados em todas as etapas do ciclo de vida do processo de desenvolvimento de software, desde os requisitos até o teste de aceitação, na fase de homologação e liberar o software para produção. O projeto de teste deve ser desenvolvido em paralelo e está integrado ao projeto de desenvolvimento.

A responsabilidade de cada um na execução dos testes deve ser documentada no Plano de Teste. Abaixo mostra os responsáveis pelo teste:

- **Testes Unidade** (*Programador*) -também conhecido como testes unitários. Tem por objetivo explorar a menor unidade do projeto, procurando provocar falhas ocasionadas por defeitos de lógica e de implementação em cada módulo, separadamente. O universo alvo desse tipo de teste são os métodos dos objetos ou mesmo pequenos trechos de código.
- **Testes de Integração** (*Analista de Sistemas*) –visa provocar falhas associadas às interfaces entre os módulos quando esses são integrados para construir a estrutura do software que foi estabelecida na fase de projeto.
- **Testes de Sistema** (*Analista de Teste*) –avalia o software em busca de falhas por meio da utilização do mesmo, como se fosse um usuário final. Dessa maneira, os testes são executados nos mesmos ambientes, com as mesmas condições e com os mesmos dados de entrada que um usuário utilizaria no seu dia-a-dia de manipulação do software. Verifica se o produto satisfaz seus requisitos.
- **Testes de Aceitação** (*Usuários com a ajuda dos analistas de teste*) – 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.

O Plano de Teste deve incluir todos os elementos necessários para que os testes sejam executados corretamente.Em cada uma das etapas do processo de teste, é preciso executar os testes e analisar os resultados esperados, e todos os registros da execução dos testes devem ser guardados sob ferramentas de gestão dos testes, o que permite o acompanhamento do progresso dessa execução.

10. Teste de aceitação

O usuário faz parte deste teste, pois é de sua responsabilidade. Mas tem também os testadores que trabalham no teste de aceitação para ajudar o usuário. Os testadores podem incluir um plano para o teste de aceitação no Plano de Teste, porém muitos autores acham que esse plano foge do escopo do trabalho dos testadores e que não poderia ser considerado para definição do tipo de teste executado pela equipe de teste.

Uma boa ideia seria se os usuários participassem de todo o processo de desenvolvimento e teste de software, o teste de aceitação seria executado muito mais simples e rápido, levando em consideração que os problemas principais já terão sido detectados.

O teste de aceitação é projetado para determinar se o software que foi desenvolvido ou a parte que está liberada para teste estão aptos a serem utilizados pelo usuário em ambiente de produção.

Apto é uma expressão muito importante para garantir que o software foi desenvolvido de modo a atender aos requisitos de negócio inicialmente definidos. “Estar Apto” envolve quatro componentes: dados, pessoas, estrutura e regras.

Resumindo essas informações mostradas no artigo, não basta o software ser desenvolvido, testado e liberado para a produção. O bom funcionamento e o bom uso dependem também de outros fatores importantes, como foram mostrados aqui. Tem o ambiente onde o software irá operar. Bem, de qualquer forma, o ambiente de teste deve ser adequado e equivalente ao ambiente de produção.

Definição dos critérios de aceitação

Para definir os critérios de aceitação o usuário deve considerar alguns pontos importantes, ter pleno conhecimento das regras de negócio, familiarizar-se com a aplicação que está sendo implantada na produção, entender os riscos e benefícios da metodologia de desenvolvimento e de teste, e entender as consequências de adicionar uma nova funcionalidade para melhorar ou completar o sistema.Os requisitos que o software deve atender: requisitos de funcionalidades, requisitos de performance e requisitos de qualidades.

Antes dos critérios de aceitação do usuário, é preciso definir o nível de criticidade do software. Alguns fatores que definem o nível de criticidade só software são: importância do sistema para a organização, consequências das falhas, complexidade do projeto, riscos de tecnologia, complexidade do ambiente do usuário. Temos ainda o desenvolvimento do plano de aceitação e a execução do teste de aceitação, essas informações detalhadas vocês podem encontrar no livro BASTOS, Anderson et al. Base de conhecimento em teste de software. 3. Ed. São Paulo: Martins Fontes, selo Martins, 2012. 208 p.

11. Considerações finais

Esse artigo possibilitou de modo rápido e simples o conhecimento em teste de software, foi abordado aqui a área de teste de software**,** onde abrangemos processos, ambiente de teste**,** elaboração do teste**,** execução dos testes, e aceitação dos testes.

O objetivo do artigo foi alcançado com sucesso, mostrando aos usuários uma forma fácil de compreender o teste de software. Como vimos nesse artigo o teste de software é muito importante para a elaboração de um software, por isso foi escolhido esse artigo para a matéria de engenharia de software do meu curso acadêmico.

Para um bom entendimento desse artigo vale a pena pesquisar o livro Base de Conhecimento em Teste de Software, referenciado nesse artigo, esse livro é muito bom para iniciantes e abrange o tema de forma bem fácil de compreender, temos outros livros, artigos e etc. Vale a pena se aprofundar nesse tema.

12. Referências

BASTOS, A.; RIOS, E.; CRISTALLI R.; MOREIRA, T. Base de Conhecimento em Teste de Software. São Paulo: Martins, 2007

BASTOS, Anderson et al. Base de conhecimento em teste de software. 3. ed São Paulo: Martins Fontes, selo Martins, 2012.

CARRERA, Jose. Os 7 princípios do teste de software, 2012. Disponível em:<https://bytesdontbite.com/2012/06/04/os-7-principios-do-teste-de-software//>. Acesso em: 16 de junho. 2016.

ANDRADE, Ana Paula, Criação e geração de planos de teste de software, 2012. Disponível em <http://www.ibm.com/developerworks/br/local/rational/criacao_geracao_planos_testes_software/. >. Acesso em: 16 de junho. 2016.

NOGUEIRA, Admilson. Casos de uso(cenários), 2006. Disponível em<http://imasters.com.br/artigo/3811/uml/casos-de-uso-cenarios?trace=1519021197&source=single>. Acesso em: 17 de junho. 2016.