

- 1 - Introdução
- 2 - Revisão Bibliográfica
 - 2.1 - Considerações iniciais
 - 2.2 - Teste de software
 - 2.2.1 - Técnicas de teste
 - 2.2.1.1 - Técnica Funcional
 - 2.2.1.2 - Técnica Estrutural
 - 2.2.1.3 - Técnica Baseada em Erros
 - 2.2.2 - Ferramentas de teste

1 – Introdução

Capítulo 2 – Revisão Bibliográfica

2.1 – Considerações iniciais

Nesta seção serão apresentados os conceitos básicos para o bom entendimento deste trabalho. Na seção 2.2 serão apresentados os principais conceitos sobre teste de software, onde serão abordadas também as técnicas de teste, na seção 2.2.1, e as ferramentas utilizadas para o teste de software, na seção 2.2.2.

2.2 – Teste de software

Atualmente, na engenharia de software, existem diversas metodologias e técnicas que visam o bom desenvolvimento de um produto de software em todas as suas etapas. Porém, mesmo com o apoio de todas essas metodologias e técnicas, alguns erros podem ser inseridos nesse produto. E tendo em vista a qualidade e confiabilidade do produto, algumas atividades para detectar esses erros se fazem necessárias. Dentre essas atividades para garantia de qualidade, podemos citar a atividade de V & V, Verificação e Validação. Essa atividade verifica a correta construção do software. O teste de software é uma das atividades de verificação e validação mais utilizadas. O objetivo do teste é detectar a maior quantidade de erros e defeitos contidos no software. Um bom teste de software é aquele que consegue fazer com que o produto testado falhe.

O teste de software é dividido em três etapas:

1. Teste unitário ou de unidade: Onde serão verificados as menores unidades testáveis de um software, onde o foco é encontrar erros de programação e de lógica.
2. Teste de integração: Nessa etapa são verificadas as integrações entre as unidades do programa, detectando principalmente erros de interface e comunicação entre eles.
3. Teste de sistema: O sistema como um todo será testado, verificando se os módulos se adequam entre si e se as expectativas do usuário serão alcançadas.

Cada uma dessas etapas, serão divididas ainda em outras quatro etapas:

1. Planejamento das atividades do teste e dos recursos utilizados.
2. Criação dos casos de teste com base nas técnicas e critérios selecionados.
3. Execução dos casos de teste.
4. Análise dos resultados obtidos com os casos de teste.

Uma atividade de teste deveria ser executada em conjunto com todas as entradas possíveis para esse software, ou seja, o domínio de entrada, caracterizando o teste exaustivo. Porém, até mesmo em softwares muito simples, a quantidade de entradas possíveis seria tão grande que a atividade de teste seria inviável. Assim, foram definidas técnicas e critérios para a seleção de subconjuntos dos domínios de entrada, de forma que esses subconjuntos representem as entradas como um todo. **Mesmo não sendo possível garantir que o software esteja inteiramente correto e livre de falhas, a partir da utilização desses critérios e técnicas, ainda assim um grau de qualidade de confiança pode ser obtido.**

Essas técnicas e critérios podem ainda ser utilizadas em conjunto uma com as outras, de modo a contemplar um domínio maior de testes, obtendo assim uma medida ainda mais confiável e precisa da qualidade do software.

Ainda assim, mesmo com atividades de teste bem elaboradas e planejadas, essas atividades são uma das mais onerosas no processo de criação de uma aplicação. Devido a seu grau de importância e ao seu custo no processo de criação do produto de software, ferramentas de teste automatizado se fizeram necessário, como forma de diminuir os custos e manter o grau de qualidade do mesmo. Ressalta-se ainda, que sem essas ferramentas, os testes estariam detidos ao teste manual que é extremamente custosa e bastante propensa a falhas, visto que é feita por humanos.

Também é importante ressaltar que o teste de software, apesar de ser uma atividade extremamente importante, é apenas uma das técnicas de engenharia de software, e sozinha não é capaz de garantir o bom desenvolvimento do software. Faz-se então necessário que para a boa produção de um software outras técnicas da engenharia de software sejam utilizadas em conjunto.