

Teste de Software

Apresentação

O teste de software é um processo que está inserido no desenvolvimento de um software. Seu objetivo é encontrar/revelar falhas e problemas no software, para que sejam corrigidas até que o produto final atinja a qualidade desejada. Entende-se como "qualidade" o encontro do que está sendo entregue com os requisitos de software levantados no início do projeto, ou seja, o software de qualidade é aquele que atende aos requisitos e às necessidades do cliente. Os profissionais que trabalham nesta área estão habituados a realizar um conjunto de testes de diferentes naturezas e propósitos, envolvendo não apenas os testes funcionais da aplicação, mas também diversas outras atividades, como avaliação da especificação de requisitos, avaliação de projeto técnico, verificações em outros documentos, testes de performance e capacidade e avaliação de interface.

Nesta Unidade de Aprendizagem, você vai adquirir conhecimentos fundamentais sobre testes de software na Engenharia de software. Vai estudar conceitos básicos sobre o que são testes de software e seus tipos, além de saber indicar os recursos técnicos que atuam nos testes.

Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Identificar os conceitos de planejamento de testes.
- Reconhecer os conceitos de testes (unitários, de integração, sistemas e de aceitação).
- Indicar os recursos técnicos que atuam nos testes.

Padrão de resposta esperado

Testes unitários: garantem que pequenas partes funcionais do sistema estarão respondendo aos requisitos de software, ou seja, que funções, métodos, classes e outros elementos de código estarão em funcionamento. Neste tipo de teste, serão verificados os pequenos cálculos que devem ter assertividade em 100% dos casos.

Testes de integração: na entrega de cada incremento de software, deverão ser realizados testes de integração para garantir que cada incremento não irá prejudicar o funcionamento dos softwares de forma geral.

Teste de sistemas e de aceitação: o teste de sistema irá garantir o funcionamento como um todo, sendo possível verificar também todas as questões funcionais. Esse teste pode ser realizado como forma de aceitação pelos clientes e testadores ao final do projeto como forma de analisar se os requisitos estão atendidos.

Desafio

Considere a seguinte situação: Pedro está desenvolvendo um sistema de controle financeiro para uma grande instituição. Este sistema deve ter a garantia de realizar cálculos financeiros de forma correta, uma vez que pequenos erros podem causar prejuízos de milhões de reais para a instituição. O desenvolvimento deste sistema será dividido em fases bem definidas, de acordo com o ciclo de vida incremental da Engenharia de software. Cada módulo desta divisão deve ser incrementado no seu momento do ciclo de vida.

Imagine que você é o parceiro de Pedro na criação deste *software* e está responsável pela parte de testes. Pedro está na dúvida sobre a utilidade da realização da fase de teste. Como cada um dos tipos de testes (unitários, de integração, sistemas e de aceitação) pode ajudar Pedro a entregar um *software* de qualidade para o usuário?

Os diferentes tipos de testes desempenham papéis importantes no processo de desenvolvimento de software e podem ajudar Pedro a entregar um software de qualidade para o usuário da seguinte maneira:

Testes Unitários:

Os testes unitários visam verificar a corretude do código em nível de componentes individuais, como funções ou métodos.

Eles ajudam Pedro a identificar e corrigir erros no código de forma precoce, o que é fundamental para evitar que pequenos erros se propaguem para fases posteriores do desenvolvimento.

Garantem que cada unidade funcione conforme o esperado antes de ser integrada com outras partes do sistema.

Testes de Integração:

Os testes de integração têm como objetivo verificar se os diferentes componentes ou módulos do sistema funcionam corretamente quando são combinados.

Eles ajudam Pedro a identificar problemas de interoperabilidade entre os módulos, garantindo que a comunicação entre eles ocorra conforme o planejado.

Evitam problemas de integração que podem levar a erros financeiros graves quando o sistema estiver em uso.

Testes de Sistema:

Os testes de sistema avaliam o sistema como um todo, verificando se ele atende aos requisitos funcionais e de desempenho.

Eles ajudam Pedro a garantir que o sistema completo funcione de acordo com as especificações e que todos os cálculos financeiros sejam precisos.

Identificam problemas sistêmicos que podem causar prejuízos financeiros se não forem corrigidos antes do uso em produção.

Testes de Aceitação:

Os testes de aceitação são realizados com a participação do usuário final ou do cliente para garantir que o sistema atenda aos requisitos do negócio.

Eles ajudam Pedro a confirmar que o software é adequado para o uso pretendido e que os cálculos financeiros são precisos de acordo com os critérios do usuário.

Garantem que o sistema atende às expectativas do usuário e ajuda a evitar prejuízos financeiros decorrentes de insatisfação ou erros não detectados.

Em resumo, os diferentes tipos de testes desempenham papéis complementares no processo de desenvolvimento de

Infográfico

Teste de *software* é a investigação do *software* a fim de fornecer informações sobre sua qualidade em relação ao contexto em que ele deve operar. O infográfico mostra os diferentes tipos de testes e as técnicas utilizadas em cada um deles. Confira!

TESTES DE SOFTWARE

É a **investigação** do *software* a fim de fornecer informações sobre sua **qualidade** em relação ao contexto em que ele deve operar.

ARTEFATOS

Plano de testes
Casos de testes
Relatórios de bugs
...

FASES

Testes unitários
Testes de integração
Testes de sistemas
Testes de aceitação

TÉCNICAS

Caixa-branca	Técnicas não funcionais
Caixa-preta	Teste de performance
Régressão	Teste de carga
Caixa-cinza	...

Conteúdo do Livro

Os testes de *software* permitem que sejam detectados problemas entre o que foi especificado e o resultado final da implementação. A criação de uma estratégia de teste de *software* pode fornecer um roteiro que descreve as etapas que devem ser executadas como parte de um teste, além de definir quando os passos são planejados e executados. Além disso, a estratégia define o montante de trabalho, o tempo e os recursos que serão utilizados. Geralmente, uma estratégia de teste de *software* inclui um planejamento dos testes, projeto de casos de teste, execução de testes e coleta e avaliação dos dados.

Faça a leitura do capítulo Teste de *software*, da obra Engenharia de *software*, para ver conceitos de testes de *software* e seus tipos, além de identificar os recursos técnicos que atuam nos testes.

Boa leitura.

ENGENHARIA DE *SOFTWARE*



Aline Zanin

Izabelly Soares
de Moraes

Revisão técnica:

Jeferson Faleiro Leon

Desenvolvimento de Sistemas

Especialista em Formação Pedagógica de Professores



M827e Morais, Izabelly Soares de.

Engenharia de software [recurso eletrônico] / Izabelly Soares de Morais, Aline Zanin ; revisão técnica : Jeferson Faleiro Leon. – Porto Alegre : SAGAH, 2017.

ISBN 978-85-9502-253-9

Engenharia. 2. Engenharia de software auxiliada por computador. I. Zanin, Aline. II. Título.

CDU 004.41

Teste de software

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Identificar os conceitos de planejamento de testes.
- Reconhecer os conceitos de testes (unitários, de integração, sistemas e de aceitação).
- Indicar os recursos técnicos que atuam nos testes.

Introdução

O teste de software é um processo que está inserido no desenvolvimento de um software. Seu objetivo é encontrar/revelar falhas e problemas no software, para que sejam corrigidos até que o produto final atinja a qualidade desejada. Entende-se como “qualidade” o encontro do que está sendo entregue com os requisitos de software levantados no início do projeto, ou seja, o software de qualidade é aquele que atende aos requisitos e às necessidades do cliente. Os profissionais que trabalham nessa área estão habituados a realizar um conjunto de testes de diferentes naturezas e propósitos, envolvendo não apenas os testes funcionais da aplicação, mas também diversas outras atividades, como avaliação da especificação de requisitos, avaliação de projeto técnico, verificações em outros documentos, testes de performance e capacidade e avaliação de interface.

Neste capítulo, você vai adquirir conhecimentos fundamentais sobre testes de software na Engenharia de Software. Vai estudar conceitos básicos sobre o que são testes de software e seus tipos, além de saber indicar os recursos técnicos que atuam nos testes.

Planejamento de testes de software

Teste de software é uma das atividades envolvidas no processo de desenvolvimento de sistemas. O diferencial é que todas as ações envolvidas na atividade de testes se preocupam exclusivamente em garantir a qualidade do software. O processo de teste básico consiste das seguintes atividades básicas: planejamento e controle; análise e modelagem; implementação e execução; avaliação dos critérios de saída e relatórios; e atividades de encerramento de teste.

A atividade de **planejamento** é a primeira a ser executada e ela é a base de todo o processo de testes, uma vez que estrutura e organiza as partes e os artefatos envolvidos com os testes. Essa etapa tem por finalidade principal definir os objetivos e especificar as atividades envoltas no projeto de testes e definir uma forma de alcançá-los. É nessa etapa que é criado um documento chamado “plano de teste”, que contém a sumarização de todas as definições processuais de um projeto de testes, como: pessoas envolvidas, tipos de testes, técnicas, riscos, artefatos, etc. A atividade de controle acontece durante todo o processo e visa a verificar se o que foi definido no planejamento está sendo cumprido.

Na etapa de **análise e modelagem**, são criados os artefatos necessários para cumprir o planejamento. Nessa etapa, por exemplo, são criados os casos de teste e os *scripts* de teste. Além disso, é verificado o ambiente de testes e são feitos os ajustes necessários. São priorizados os casos de testes ou *scripts* de teste e é criada uma rastreabilidade bidirecional entre os requisitos e os casos de teste.

Na etapa de **implementação e execução**, os artefatos projetados são organizados em suítes de acordo com sua prioridade, ou por funcionalidade, e são executados. Na etapa de **avaliação dos critérios de saída e relatórios**, a execução do teste é avaliada por meio dos objetivos definidos.

Por fim, na etapa de **atividades de encerramento de teste**, são coletados os dados de todas as atividades para registrar experiência, *testware*, fatos e números. Por exemplo, quando um software é lançado, um projeto de teste é completado (ou cancelado), um marco do projeto foi alcançado, ou a implantação de uma demanda de manutenção foi completada (Figura 1) (BSTQB, 2011).

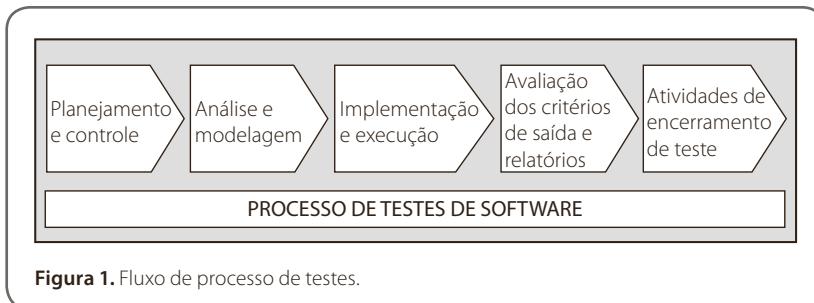


Figura 1. Fluxo de processo de testes.

Níveis de testes: unitários, integração, sistema e aceitação

Para a execução de um processo de testes completo, diversos são os níveis de testes que podem ser empregados, sendo eles: unitários, de integração, sistema e de aceitação. Vamos analisar a diferença entre eles:

- **Unitários:** este nível de teste geralmente é executado pelo programador, durante o processo de desenvolvimento do software. Isso porque esse nível de teste considera o teste de pequenas partes do sistema. Um exemplo de técnica utilizada para realização desse nível de teste é o Test Driven Development (TDD) ou, em português, Desenvolvimento Guiado por Testes, sendo possível empregá-lo em diversas linguagens de programação.
- **Teste de integração:** o teste de integração tem por objetivo verificar se uma ou mais unidades “se encaixam perfeitamente”, ou seja, se as peças que de forma separada funcionam perfeitamente, quando unificadas, irão se comunicar e estar em pleno funcionamento.
- **Teste de sistema:** este tipo de teste é o mais comumente empregado nas empresas de tecnologia e ele visa a testar o sistema ou uma funcionalidade como um todo. Esse teste pode ser realizado de forma manual ou de forma automatizada e fazendo uso de diversas ferramentas.
- **Teste de aceitação:** este é o último nível de testes e recomenda-se que ele seja feito com a presença do cliente ou de um representante dele. Nesse nível de teste, o objetivo não é encontrar falhas, e sim confirmar que tudo está implementado conforme foi solicitado, para que a entrega seja aceita.



Saiba mais

TDD é uma técnica de testes em que, antes de se construir a funcionalidade, se constrói o teste. Isso garante que a funcionalidade implementada esteja em pleno funcionamento e de acordo com os requisitos solicitados, uma vez que o teste é criado visando a atender os requisitos. Essa técnica considera os ciclos *red*, *green* e *refactor*. No ciclo *red* (vermelho), escreve-se um teste que irá falhar porque a funcionalidade não foi programada ainda; no ciclo *green* (verde), implementa-se a funcionalidade que fará o teste passar, e no ciclo *refactor* (refatore), o código é aprimorado.

Recursos técnicos utilizados para testes de software

A fim de tornar mais fácil a tarefa de testes de software, diversos são os recursos técnicos que podem ser utilizados, sejam eles ferramentas, frameworks, linguagens de programação, etc. Vamos listar a seguir, separados por níveis de testes, alguns recursos que podem ser utilizados

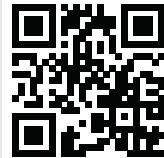
1. Unitários: conforme supracitado, a principal técnica para a realização de testes unitários é TDD. São exemplos de recursos para aplicação de TDD os seguintes:
 - a) JUnit: framework utilizado para realização de TDD, utilização de linguagem Java.
 - b) PHPUnit: framework utilizado para realização de TDD, utilização de linguagem PHP.
 - c) Flask: framework utilizado para realização de TDD, utilização de linguagem Python.
4. Sistema e aceitação: por terem funcionamento similar, no contexto de testarem a ferramenta já desenvolvida, são os que dispõem da maior gama de ferramentas de testes, sendo algumas delas:
 - a) Sikuli: ferramenta para automação de qualquer sistema que tenha interface gráfica, inclusive sistemas desktop.
 - b) Selenium WebDriver: ferramenta para automação de testes de sistemas web.
 - c) Robotium: ferramenta para automação de testes de aplicativos móveis.



Link

Aprofunde seus conhecimentos sobre testes unitários assistindo ao vídeo disponível no link ou código a seguir.

<https://goo.gl/421r8c>



Referências

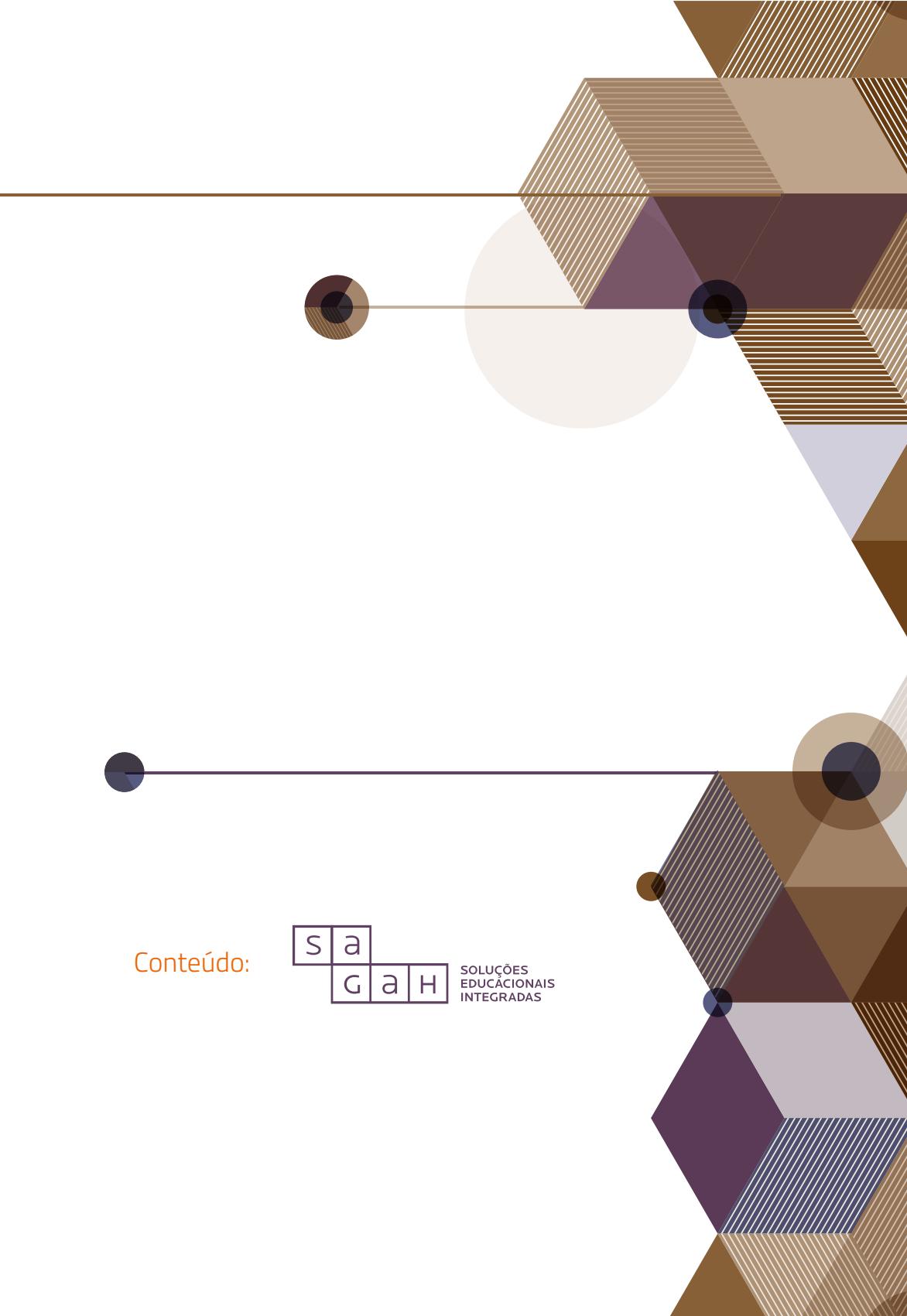
BSTQB, Brazilian Software Testing Qualifications Board. *Certified Tester Foundation Level Syllabus*. BSTQB, 2001. Disponível em: <http://bstqb.org.br/uploads/syllabus/syllabus_ctfl_2011br.pdf>. Acesso em: 16 set. 2017

Leituras recomendadas

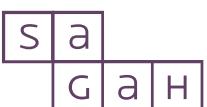
DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. (Ed.). *Introdução ao teste de software*. Rio de Janeiro: Elsevier, 2007.

RIOS, E.; MOREIRA, T. *Teste de software*. Rio de Janeiro: Alta Books, 2006.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.



Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS

Dica do Professor

Para garantir a qualidade de um programa, com a detecção de falhas antes do *software* ser comercializados, são realizados testes, que vão desde a análise da estrutura interna até a avaliação da interface do *software*. O vídeo a seguir mostra os tipos de testes de *software* e o ciclo de vida dos testes. Assista!



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Exercícios

- 1) Qual destas definições representa um teste de *software*?
 - A) O teste do *software* é a investigação do *software* a fim de fornecer informações sobre sua qualidade em relação ao contexto em que ele deve operar. Isso inclui o processo de utilizar o produto para encontrar seus defeitos.
 - B) Os testes de *software* são ações tomadas para que um projeto demore mais do que o esperado, não sendo necessário em projetos críticos.
 - C) Os testes de *software* são realizados no início do projeto, antes da implementação do *software*.
 - D) Testes de *software* buscam demonstrar a incapacidade dos desenvolvedores na criação de soluções de qualidade.
 - E) Testes de *software* são um conjunto de atividades realizadas somente pelo cliente quando recebe o produto.
- 2) O comportamento interno do componente de *software* trabalha diretamente sobre o código-fonte do componente de *software* para avaliar aspectos como teste de condição, teste de fluxo de dados, teste de ciclos, teste de caminhos lógicos e códigos nunca executados. Das técnicas apresentadas a seguir, qual avalia esse comportamento interno do componente de *software*?
 - A) Teste de caixa-preta.
 - B) Teste de caixa-branca.
 - C) Regressão.
 - D) Técnicas não funcionais.
 - E) Caixa-cinza.
- 3) Qual é a fase de teste em que se testam as menores unidades de *software* desenvolvidas (pequenas partes ou unidades do sistema)?

- A) Teste de aceitação.
- B) Teste de sistema.
- C) Teste de operação.
- D) Teste de integração.
- E) Teste de unidade.

4) O que é TDD na Engenharia de *Software*?

- A) É um ciclo de vida desenvolvido para facilitar a criação de testes.
- B) TDD é uma técnica de testes de *software* utilizada pelos usuários dos sistemas.
- C) TDD é um modelo clássico da Engenharia de *Software* e não é mais utilizado.
- D) TDD é um sistema usado para testar *software* de forma automatizada.
- E) O TDD é uma técnica de desenvolvimento de *software* que se baseia na criação inicial de testes de *software* antes mesmo do desenvolvimento.

O Test Driven Development (TDD), ou Desenvolvimento Guiado por Testes, é uma técnica na qual o desenvolvedor cria o software para atender apenas aos requisitos. Esse desenvolvimento é guiado e controlado pelos testes unitários previamente criados.

- 5) O ciclo de vida de testes de *software* tem cinco fases: planejamento, preparação, especificação, execução e entrega. Qual fase tem como objetivo elaborar e revisar os casos e roteiros de testes?
- A) Planejamento.
 - B) Preparação.
 - C) Execução.
 - D) Especificação.
 - E) Entrega.

Na prática

Acompanhe o planejamento de testes para um *software* de sistema de controle de voos, a ser utilizado em todo o País. O sistema é muito crítico, pois pequenos erros no *software* podem causar acidentes aéreos, colocando em risco a vida de muitas pessoas. Sendo assim, existe uma forte necessidade de esse sistema ser testado de diversas formas, a fim de garantir o pleno funcionamento e a assertividade nas respostas.

Conteúdo interativo disponível na plataforma de ensino!

Ao final do projeto, foi entregue um sistema totalmente funcional e coberto por testes. Qualquer falha possível seria descoberta, primeiro, através dos testes unitários. Além disso, os testes de sistema e integração garantiram a qualidade do sistema, pois os requisitos foram atendidos.



Paulo é gerente do time de desenvolvimento de *software* que criou o sistema de controle de voos. Ele seguiu o ciclo de vida dos testes para garantir que o *software* criado seja devidamente testado.

- Inicialmente Paulo realizou o **planejamento da estratégia de testes** e também criou um **plano de teste**.
- Após, Paulo iniciou a **fase de Preparação**. Nesta fase ele preparou o ambiente de teste (equipamentos, pessoal, ferramentas para automatizar os teste) para tudo fosse realizado da forma que foi planejado.



Nesta fase de planejamento, Paulo definiu que utilizaria os seguintes tipos de técnicas:

- **unitários**
- **integração**
- **sistemas e de aceitação**.

Porém, Paulo achou importante dar ênfase nos testes unitários para garantir que pequenos módulos do sistema não iriam falhar, já que o sistema não poderia apresentar erros nos cálculos e dados processados.



Além disso, testes de sistema e integração foram planejados com foco maior em desempenho e usabilidade. O time de testes iniciou a fase de especificação onde os casos e roteiros de testes foram elaborados e revisados. Como sequência, foram executados todos os roteiros e casos de teste (testaram o *software*) e por fim houve a entrega da documentação de testes.

Obviamente, foram encontrados problemas no *software*. Estes problemas (*reports*) foram passados para os desenvolvedores que ajustaram e corrigiram os *bugs*.

Saiba mais

Para ampliar o seu conhecimento a respeito desse assunto, veja abaixo as sugestões do professor:

Implantação de um Processo de Teste de Software

Acompanhe uma palestra com a demonstração da implantação de um processo de teste de software, com a apresentação das dificuldades, desafios e potenciais melhorias com a adoção de um processo.



Aponte a câmera para o código e accese o link do conteúdo ou clique no código para accesar.

Teste de Software - Plano de Teste

No próximo vídeo você poderá saber mais sobre o caso de testes e o plano de teste, que é um documento que possibilita uma visão macro do sistema, com cronograma, possíveis riscos e outras informações importantes.



Aponte a câmera para o código e accese o link do conteúdo ou clique no código para accesar.