Projeto Guru99 - Dia 8

Próxima fase de testes

Olá colega testador, Atualizações sobre a versão 2 do site

============================

Abaixo estão os links atualizados

Casos de teste de integração

Casos de teste de unidade

Rastreador de erros

Consulte o Changelog para determinar as alterações efetuadas nos casos de teste de acordo com o SRS v2.

============================

Atualizações da versão 3 do sítio

============================

A equipa de desenvolvimento informou que a versão 3 estará disponível nos próximos 2 dias

Esta versão incluirá:

Todos os módulos para a função de Gestor

Todos os módulos para a função de cliente

Idealmente, é necessário criar e executar casos de teste de módulo para

Função de gestor:

Alterar palavra-passe

Depósito

Retirada

Transferência de fundos

Função de cliente:

Consulta de saldo

Transferência de fundos

Mini extracto

Extrato personalizado

Alterar palavra-passe

Também é necessário efectuar testes de integração para todos os módulos.

Isto requer um esforço gigantesco e é normalmente realizado por uma equipa mínima de 5 membros.Para fins de auto-aprendizagem, vamos ignorar esta etapa.

Passando à fase seguinte, crie um plano de teste do sistema. Se é novo no domínio dos testes do sistema, consulte este tutorial

**Tarefa do dia**

**Criar um plano de teste do sistema**

Consulte o SRS\_v3. O cliente quer agora que seja criado um serviço Web. Crie casos de teste para verificar esse webservice.

Parabéns

Equipa Guru99

Dica Profissional:

1) Se você verificar o Defeito # 10 no Bug Tracker. O seu estado é diferido. Isso acontece frequentemente durante o ciclo de vida de um projeto de teste, onde bugs de baixa prioridade e severidade que estão levando muito tempo para serem resolvidos são adiados e corrigidos na próxima fase do projeto. Estes erros são também designados por "problemas conhecidos"

2) Qualquer grande pedido de alteração (como o pedido de serviço Web acima) altera o âmbito do projecto e o orçamento tem de ser renegociado. No entanto, é melhor deixar esta tarefa para os seus executivos de desenvolvimento de negócios e gestores de testes

-Foi disponibilizado um documento(planilha) de gestão de bug onde contem o bus e o test case referente ,os passos para reprodução,resultado esperado,resultado atual,a prioridade e a severidade .

-Um documento (planilha)de casos de testes que são os caso de teste da versão1 que falharam e agora foram retestados .

-Um documento (planilha )de teste de integração onde contem os testes da versão 2

- esses documentos foram disponibilizados para nós qa realizarmos comparações e vermos se fizemos os passos parecidos ou se deixamos algum teste passar

-Agora vou criar um plano de teste de sistema para aversão 3

-Documento sobre teste de sistema

**O que é Teste de Sistema?**

**O teste do sistema** é um nível de teste que valida o produto de software completo e totalmente integrado. O objetivo de um teste de sistema é avaliar as especificações do sistema de ponta a ponta. Normalmente, o software é apenas um elemento de um sistema computadorizado maior. Por fim, o software é conectado a outros sistemas de software/hardware. O Teste do Sistema é definido como uma série de testes diferentes cujo único propósito é exercitar o sistema completo baseado em computador.

## O teste do sistema é Blackbox

Duas categorias de teste de software

* Teste de caixa preta
* Teste de caixa branca

O teste do sistema se enquadra na categoria **de teste de caixa preta** de teste de software.

**O teste de caixa branca** é o teste do funcionamento interno ou do código de um aplicativo de software. Em contraste, caixa preta ou teste de sistema é o oposto. O teste do sistema envolve o funcionamento externo do software da perspectiva do usuário.

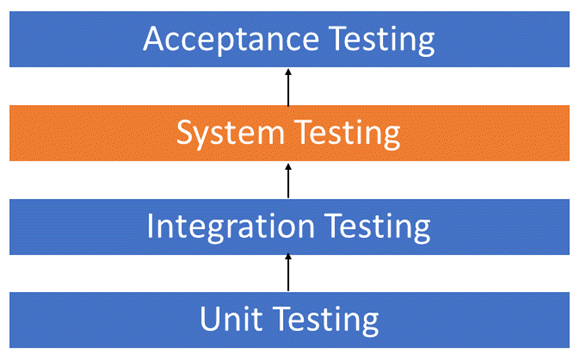
## O que você verifica no teste do sistema?

O teste do sistema envolve testar o código do software para seguir

* Testar os aplicativos totalmente integrados, incluindo periféricos externos, para verificar como os componentes interagem entre si e com o sistema como um todo. Isso também é chamado de cenário de teste de ponta a ponta.
* Verifique o teste completo de cada entrada no aplicativo para verificar as saídas desejadas.
* Teste da experiência do usuário com o aplicativo.

Essa é uma descrição muito básica do que está envolvido no teste do sistema. Você precisa criar casos de teste detalhados e suítes de teste que testem cada aspecto do aplicativo visto de fora, sem examinar o código-fonte real. Para saber mais sobre uma abordagem abrangente desse processo, leia sobre [o teste de ponta a ponta](https://www.guru99.com/end-to-end-testing.html) .

**Hierarquia de teste de software**



Como em quase todos os processos de engenharia de software, o teste de software tem uma ordem prescrita na qual as coisas devem ser feitas. A seguir está uma lista de categorias de teste de software organizadas em ordem cronológica. Estas são as etapas tomadas para testar totalmente o novo software em preparação para comercializá-lo:

* Teste de unidade realizado em cada módulo ou bloco de código durante o desenvolvimento. [O teste de unidade](https://www.guru99.com/unit-testing-guide.html) normalmente é feito pelo programador que escreve o código.
* Teste de integração feito antes, durante e depois da integração de um novo módulo no pacote de software principal. Isso envolve o teste de cada módulo de código individual. Um software pode conter vários módulos que geralmente são criados por vários programadores diferentes. É crucial testar o efeito de cada módulo em todo o modelo de programa.
* Teste de sistema feito por um agente de teste profissional no produto de software completo antes de ser introduzido no mercado.
* Teste de aceitação – teste beta do produto feito pelos usuários finais reais.

**Tipos de teste do sistema**

Existem mais de 50 tipos de teste de sistema. Para obter uma lista exaustiva dos tipos de teste de software, clique [aqui](https://www.guru99.com/types-of-software-testing.html) . Abaixo, listamos os tipos de teste de sistema que uma grande empresa de desenvolvimento de software normalmente usaria

1. [**Teste de Usabilidade**](https://www.guru99.com/usability-testing-tutorial.html)**–** concentra-se principalmente na facilidade do usuário em usar o aplicativo, na flexibilidade no manuseio de controles e na capacidade do sistema de atender aos seus objetivos
2. [**Teste de carga**](https://www.guru99.com/load-testing-tutorial.html)**–** é necessário saber que uma solução de software funcionará sob cargas da vida real.
3. [**Teste de regressão**](https://www.guru99.com/regression-testing.html)**–** envolve testes feitos para garantir que nenhuma das alterações feitas ao longo do processo de desenvolvimento tenha causado novos bugs. Ele também garante que nenhum bug antigo apareça devido à adição de novos módulos de software ao longo do tempo.
4. [**Teste de Recuperação**](https://www.guru99.com/recovery-testing.html)**–** é feito para demonstrar que uma solução de software é confiável, confiável e pode se recuperar com sucesso de possíveis falhas.
5. **Teste de migração -** é feito para garantir que o software possa ser movido de infraestruturas de sistema mais antigas para infraestruturas de sistema atuais sem problemas.
6. **Teste Funcional –** Também conhecido como teste de integridade funcional, [o Teste Funcional](https://www.guru99.com/functional-testing.html) envolve tentar pensar em possíveis funções ausentes. Os testadores podem fazer uma lista de funcionalidades adicionais que um produto pode ter para melhorá-lo durante o teste funcional.
7. **Teste de Hardware/Software –** A IBM refere-se ao teste de Hardware/Software como “Teste de HW/SW”. É quando o testador concentra sua atenção nas interações entre o hardware e o software durante o teste do sistema.

**Quais tipos de teste de sistema os testadores devem usar?**

Existem mais de 50 tipos diferentes de teste de sistema. Os tipos específicos usados ​​por um testador dependem de diversas variáveis. Essas variáveis ​​incluem:

* Para quem o testador trabalha – Este é um fator importante na determinação dos tipos de teste de sistema que um testador usará. Os métodos utilizados pelas grandes empresas são diferentes dos utilizados pelas médias e pequenas empresas.
* Tempo disponível para teste – Em última análise, todos os 50 tipos de teste podem ser usados. Muitas vezes, o tempo é o que nos limita a usar apenas os tipos mais relevantes para o projeto de software.
* Recursos disponíveis para o testador – É claro que alguns testadores não terão os recursos necessários para conduzir um tipo de teste. Por exemplo, se você é um testador que trabalha para uma grande empresa de desenvolvimento de software, é provável que você tenha um software de teste automatizado caro que não está disponível para outras pessoas.
* Educação do testador de software - Existe uma certa curva de aprendizado para cada tipo de teste de software disponível. Para usar parte do software envolvido, um testador precisa aprender como usá-lo.
* Orçamento de teste – O dinheiro se torna um fator não apenas para empresas menores e desenvolvedores de software individuais, mas também para grandes empresas.

Font: https://www.guru99.com/system-testing.html