

Projeto Guru99 – Dia 4

Casos de teste dentro!

Muito bem!

Aqui estão os casos de teste criados pelos nossos especialistas para os 8 módulos do Manager discutidos na última mensagem de correio electrónico.

Pode argumentar que os casos de teste acima se parecem mais com casos de teste de unidade e devem ser executados pelos programadores.

Em teoria, isso está correto. Na prática, os programadores estão quase sempre com falta de tempo e raramente testam o sistema de forma exaustiva. Além disso, pode haver erros resultantes de diferenças entre o ambiente de desenvolvimento e o ambiente de teste (que geralmente é uma cópia do ambiente de produção). Por isso, como melhor prática, é melhor verificar alguns ou todos os casos de testes unitários. No entanto, esta abordagem pode variar de empresa para empresa.

No nosso projeto Guru99 Bank, iremos incluir casos de teste unitários.

Teste de Integração

Nós recomendamos fortemente que você leia este tutorial sobre testes de integração antes de seguir em frente.

Um projeto de software típico consiste de vários módulos de software, codificados por diferentes programadores. O teste de integração foca na verificação da comunicação de dados entre esses módulos.

Aqui está o Plano de Teste de Integração preparado por nossos especialistas. [link](#)

Muito bem, hoje vamos fazer a linha de base de todos os nossos casos de teste. Amanhã começamos com a execução dos testes!

Abraço.Equipe Guru99

DICA Profissional

- 1) O teste de unidade também é chamado de teste de módulo ou teste de componente
- 2) O formato dos planos de teste de módulo e de integração NÃO é fixo. NÃO existem formatos universais. É decidido pela sua organização de testes para se adequar aos requisitos do projeto. Não se preocupe se os seus formatos forem diferentes dos nossos. Mas verifique se inclui todos os casos/cenários de teste documentados pelos nossos especialistas

No email enviado um link onde contem um documento no excel onde continha os casos de testes e scripts de testes dos especialistas e também um link onde tem a informação sobre Testes de Integração que é você realizar integrar os módulos e realizar os testes, onde verifica a comunicação entre os módulos . Porque realizar esse teste ?

- Um Módulo, em geral, é projetado por um desenvolvedor de software individual cuja compreensão e lógica de programação podem diferir de outros

programadores. O Teste de Integração torna-se necessário para verificar se os módulos de software funcionam em unidade

- No momento do desenvolvimento do módulo, há grandes chances de mudança de requisitos por parte dos clientes. Esses novos requisitos podem não ser testados na unidade e, portanto, o teste de integração do sistema torna-se necessário.
- As interfaces dos módulos de software com o banco de dados podem estar erradas
- As interfaces de hardware externo, se houver, podem estar incorretas
- O tratamento inadequado de exceções pode causar problemas

E segundo a engenharia de software existe dentro do testes de integração alguns tipos:

Testing Big Bang

O Big Bang Testing é uma abordagem de teste de integração na qual todos os componentes ou módulos são integrados de uma só vez e depois testados como uma unidade. Este conjunto combinado de componentes é considerado como uma entidade durante o teste. Se todos os componentes da unidade não forem concluídos, o processo de integração não será executado.

Vantagens:

- Conveniente para sistemas pequenos.

Desvantagens:

- A localização de falhas é difícil.
- Dado o grande número de interfaces que precisam ser testadas nessa abordagem, alguns links de interfaces a serem testados podem ser perdidos facilmente.
- Como o teste de Integração só pode começar depois que “todos” os módulos forem projetados, a equipe de teste terá menos tempo para execução na fase de teste.
- Como todos os módulos são testados ao mesmo tempo, os módulos críticos de alto risco não são isolados e testados com prioridade. Módulos periféricos que lidam com interfaces de usuário também não são isolados e testados em prioridade.

Teste Incremental

Na abordagem **de Teste Incremental**, o teste é feito integrando dois ou mais módulos logicamente relacionados entre si e, em seguida, testados quanto ao funcionamento adequado do aplicativo. Em seguida, os outros módulos relacionados são integrados de forma incremental e o processo continua até que todos os módulos logicamente relacionados sejam integrados e testados com sucesso.

A Abordagem Incremental, por sua vez, é realizada por dois Métodos distintos:

- Baixo para cima
- Careca

Stubs e Drivers

Stubs e Drivers são os programas fictícios no teste de integração usados para facilitar a atividade de teste de software. Esses programas agem como substitutos dos modelos ausentes no teste. Eles não implementam toda a lógica de programação do módulo de software, mas simulam a comunicação de dados com o módulo de chamada durante o teste.

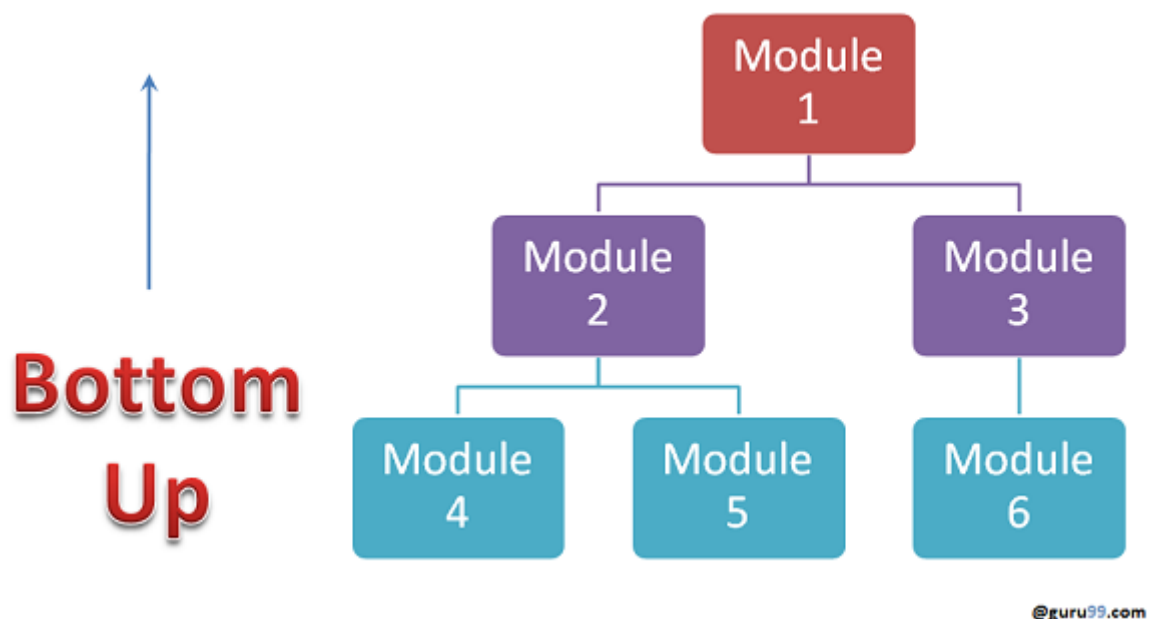
Stub : É chamado pelo Módulo em Teste.

Driver : Chama o Módulo a ser testado.

Teste de integração de baixo para cima

O teste de integração de baixo para cima é uma estratégia na qual os módulos de nível inferior são testados primeiro. Esses módulos testados são usados posteriormente para facilitar o teste de módulos de nível superior. O processo continua até que todos os módulos no nível superior sejam testados. Depois que os módulos de nível inferior são testados e integrados, o próximo nível de módulos é formado.

Representação esquemática :



Vantagens:

- A localização de falhas é mais fácil.
- Não se perde tempo esperando que todos os módulos sejam desenvolvidos, ao contrário da abordagem Big-bang

Desvantagens:

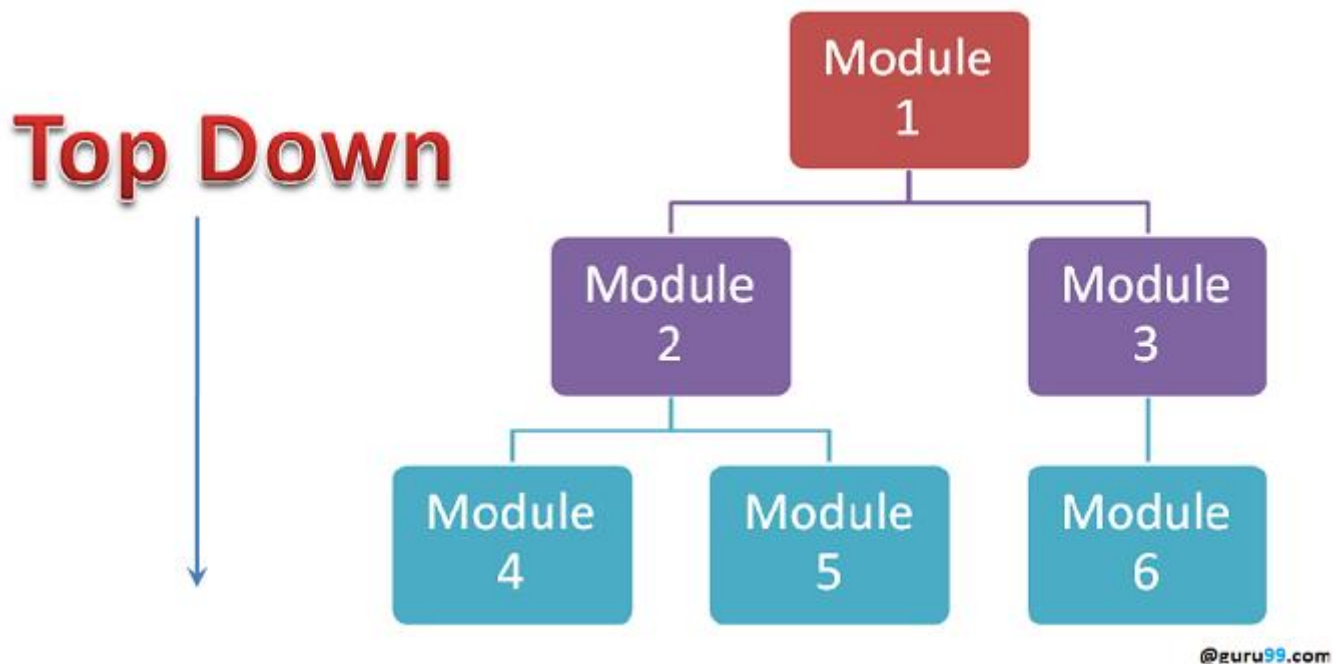
- Os módulos críticos (no nível superior da arquitetura de software) que controlam o fluxo do aplicativo são testados por último e podem estar sujeitos a defeitos.

- Um protótipo inicial não é possível

Teste de integração de cima para baixo

O teste de integração de cima para baixo é um método no qual o teste de integração ocorre de cima para baixo, seguindo o fluxo de controle do sistema de software. Os módulos de nível superior são testados primeiro e, em seguida, os módulos de nível inferior são testados e integrados para verificar a funcionalidade do software. Os stubs são usados para testar se alguns módulos não estiverem prontos.

Representação Diagramática:



Vantagens:

- A localização de falhas é mais fácil.
- Possibilidade de obter um protótipo inicial.
- Módulos Críticos são testados em prioridade; as principais falhas de design podem ser encontradas e corrigidas primeiro.

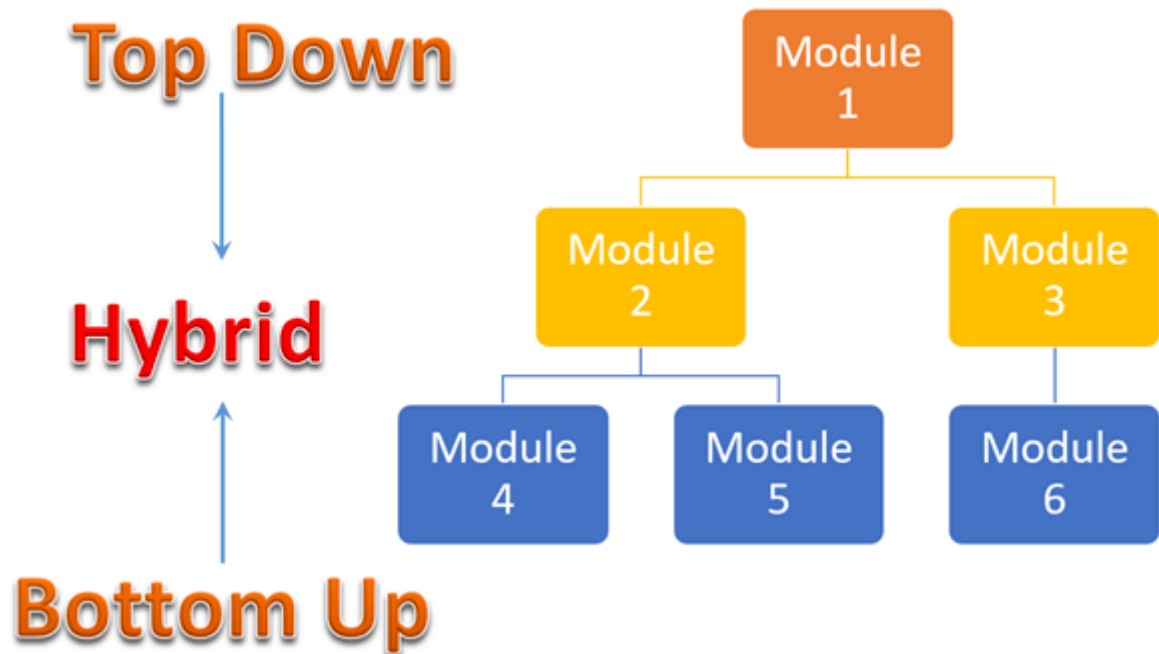
Desvantagens:

- Precisa de muitos Stubs.
- Módulos em um nível inferior são testados inadequadamente.

Teste Sanduíche

O teste sanduíche é uma estratégia na qual os módulos de nível superior são testados com módulos de nível inferior, ao mesmo tempo em que os módulos inferiores são integrados aos módulos superiores e testados como um sistema. É uma combinação de

abordagens de cima para baixo e de baixo para cima, portanto, é chamado de **Teste de Integração Híbrida**. Ele faz uso de stubs e drivers.



Como fazer testes de integração?

O procedimento de teste de integração, independentemente das estratégias de teste de software (discutidas acima):

1. Preparar o Plano de Testes de Integração
2. Projete os cenários, casos e scripts de teste.
3. Execução dos Casos de teste seguido de reporte dos defeitos.
4. Rastrear e testar novamente os defeitos.
5. As etapas 3 e 4 são repetidas até que a integração seja concluída com sucesso.

Breve Descrição dos Planos de Teste de Integração

Inclui os seguintes atributos:

- Métodos/abordagens para testes (conforme discutido acima).
- Escopos e itens fora dos escopos do teste de integração.
- Papéis e responsabilidades.
- Pré-requisitos para testes de integração.
- Ambiente de teste.
- Planos de Risco e Mitigação.

CrITÉRIOS de Entrada e Saída do Teste de Integração

CrITÉRIOS de entrada e saída para a fase de testes de integração em qualquer modelo de desenvolvimento de software

Critério de entrada:

- Componentes/módulos testados na unidade
- Todos os bugs de alta prioridade corrigidos e fechados
- Todos os módulos devem ser codificados e integrados com sucesso.
- Testes de integração Plano, caso de teste, cenários a serem assinados e documentados.
- [Ambiente de teste](#) necessário para ser configurado para teste de integração

Critério de saída:

- Teste bem-sucedido de aplicativo integrado.
- Os casos de teste executados são documentados
- Todos os bugs de alta prioridade corrigidos e fechados
- Documentos técnicos a serem enviados seguidos de Notas de versão.

Melhores Práticas/Diretrizes para Teste de Integração

- Primeiro, determine a [estratégia de teste de](#) integração que pode ser adotada e, posteriormente, prepare os casos de teste e os dados de teste de acordo.
- Estude o desenho da Arquitetura do Aplicativo e identifique os Módulos Críticos. Estes precisam ser testados em prioridade.
- Obtenha os designs de interface da equipe de arquitetura e crie casos de teste para verificar todas as interfaces em detalhes. A interface com o banco de dados/aplicativo externo de hardware/software deve ser testada detalhadamente.
- Após os casos de teste, são os dados de teste que desempenham o papel crítico.
- Sempre tenha os dados fictícios preparados, antes de executar. Não selecione dados de teste durante a execução dos casos de teste.

Font: <https://www.guru99.com/integration-testing.html>