



Compliance & Quality Assurance

2024



Testes de mesa

(Uso de testes de mesa para auxiliar a criação de casos de testes unitários, aplicando os diversos métodos estudados)

Prof. Silvio Macedo

Conteúdo

1. Conceitos Básicos
2. Objetivos do teste de mesa
3. Vantagens e Desvantagens
4. Exemplo prático
5. HandsOn

Conceitos Básicos

O que é teste de mesa?



Conceitos Básicos

O que é teste de mesa?

Teste de mesa, também conhecido como "*dry runs*" ou "*walkthroughs*", é uma técnica de teste de software realizada **manualmente**, sem a necessidade de executar o código em um computador.

A técnica envolve a simulação do comportamento do software, percorrendo passo a passo o código e seus fluxos de dados (*input* e *output*).

Encontrar falhas e pontos “cegos” no código:

Os testes de mesa podem ajudar a identificar falhas lógicas, erros de sintaxe, problemas de desempenho e outras falhas que podem ocorrer durante a execução do software.

Aprimorar a qualidade do código:

Ao encontrar e corrigir falhas no início do ciclo de desenvolvimento, os testes de mesa podem ajudar a melhorar a qualidade do código e evitar problemas mais tarde.

Vantagens x Desvantagens



Barato e fácil de realizar

Os testes de mesa não exigem nenhum equipamento especial ou software caro.

Realizado em qualquer fase do SDLC

Os testes de mesa podem ser realizados em qualquer fase do ciclo de desenvolvimento, desde o início até a fase de testes.

Análise mais profunda

Os testes de mesa permitem que a equipe analise o código mais profundamente do que seria possível com testes automatizados*.

Aumento do conhecimento técnico

O processo de realizar testes de mesa exige que os membros da equipe utilizem seus conhecimentos técnicos para analisar o código e identificar falhas.



Demorado para executar

Os testes de mesa podem ser demorados, especialmente para sistemas complexos.

Subjetividade

Os resultados dos testes de mesa podem ser subjetivos, pois dependem da experiência e do conhecimento da equipe que os realiza.

Baixa eficácia

Os testes de mesa podem não ser tão eficazes quanto os testes automatizados para encontrar alguns tipos de falhas.

Conhecimento técnico

Testes de mesa exigem um profissional com profundo conhecimento analítico e técnico.



O teste de mesa simula a execução de um algoritmo sem utilizar o computador.

Na prática!

```
total = 0;
```

```
for(x=0;x<3;x++)
```

```
{
```

```
total = total + x + 1;
```

```
}
```

```
Imprima total;
```

```
Imprima x;
```

#	Instrução	total	

Na prática!

```
total = 0;
```

```
for(x=0;x<3;x++)  
{  
total = total + x + 1;  
}
```

```
Imprima total;
```

```
Imprima x;
```

#	Instrução	total	
1	total = 0;	0	

Na prática!

```
total = 0;
```

```
for(x=0;x<3;x++)
```

```
{
```

```
total = total + x + 1;
```

```
}
```

```
Imprima total;
```

```
Imprima x;
```

#	Instrução	total	x
1	total = 0;	0	
2	for: x=0	0	0

Na prática!

```
total = 0;
```

```
for(x=0;x<3;x++)
```

```
{
```

```
total = total + x + 1;
```

```
}
```

```
Imprima total;
```

```
Imprima x;
```

#	Instrução	total	x
1	total = 0;	0	
2	for: x=0	0	0
3	total = total + x + 1 total = 0 + 0 + 1	1	0

Na prática!

```
total = 0;
```

```
for(x=0;x<3;x++)
```

```
{
```

```
total = total + x + 1;
```

```
}
```

```
Imprima total;
```

```
Imprima x;
```

#	Instrução	total	x
1	total = 0;	0	
2	for: x=0	0	0
3	total = total + x + 1 total = 0 + 0 + 1	1	0
4	for: x++	1	1

Na prática!

```
total = 0;
```

```
for(x=0;x<3;x++)
```

```
{
```

```
total = total + x + 1;
```

```
}
```

```
Imprima total;
```

```
Imprima x;
```

#	Instrução	total	x
1	total = 0;	0	
2	for: x=0	0	0
3	total = total + x+ 1 total = 0 + 0 + 1	1	0
4	for: x++	1	1
5	total = total + x+ 1 total = 1 + 1 + 1	3	1

Na prática!

```
total = 0;
```

```
for(x=0;x<3;x++)
```

```
{
```

```
total = total + x + 1;
```

```
}
```

```
Imprima total;
```

```
Imprima x;
```

#	Instrução	total	x
1	total = 0;	0	
2	for: x=0	0	0
3	total = total + x+ 1 total = 0 + 0 + 1	1	0
4	for: x++	1	1
5	total = total + x+ 1 total = 1 + 1 + 1	3	1
6	for: x++	3	2

Na prática!

```
total = 0;
```

```
for(x=0;x<3;x++)
```

```
{
```

```
total = total + x + 1;
```

```
}
```

```
Imprima total;
```

```
Imprima x;
```

#	Instrução	total	x
1	total = 0;	0	
2	for: x=0	0	0
3	total = total + x+ 1 total = 0 + 0 + 1	1	0
4	for: x++	1	1
5	total = total + x+ 1 total = 1 + 1 + 1	3	1
6	for: x++	3	2
7	total = total + x+ 1 total = 3 + 2 + 1	6	2

Na prática!

```
total = 0;
```

```
for(x=0;x<3;x++)
```

```
{
```

```
total = total + x + 1;
```

```
}
```

```
Imprima total;
```

```
Imprima x;
```

#	Instrução	total	x
1	total = 0;	0	
2	for: x=0	0	0
3	total = total + x+ 1 total = 0 + 0 + 1	1	0
4	for: x++	1	1
5	total = total + x+ 1 total = 1 + 1 + 1	3	1
6	for: x++	3	2
7	total = total + x+ 1 total = 3 + 2 + 1	6	2
8	for: x++	6	3

Na prática!

```
total = 0;
```

```
for(x=0;x<3;x++)  
{  
total = total + x + 1;  
}
```

```
Imprima total;
```

```
Imprima x;
```

#	Instrução	total	x
1	total = 0;	0	
2	for: x=0	0	0
3	total = total + x+ 1 total = 0 + 0 + 1	1	0
4	for: x++	1	1
5	total = total + x+ 1 total = 1 + 1 + 1	3	1
6	for: x++	3	2
7	total = total + x+ 1 total = 3 + 2 + 1	6	2
8	for: x++	6	3
9	total: 6 x: 3		



HandsOn

Referências

IBTQS Glossário ATC-NF: http://ibqts.com.br/downloads/Glossario_ATC-NF-IBQTS.pdf
[The Art of Software Testing, Second Edition by Glenford J. Myers, Corey Sandler, Tom Badgett, Todd M. Thomas](#)

OBRIGADO