

### Análise de requisitos:

Realização da contextualização do software para escolha dos requisitos.  
Definição de prioridade e foco da validação.

### Fase de planejamento:

Criar um plano que esteja compatível com o orçamento e prazo do projeto disponíveis.  
Divisão de responsabilidades e atividades dentro do squad.

### Integração do caso de teste:

Um primeiro esqueleto/prévia do que vai ser testado.

### Configuração do ambiente de teste:

Deixar a máquina pronta para poder rodar o sistema assim como funcionaria no cliente, se há uma compatibilidade com os requisitos (ex. Banco de Dados, Linguagem de Programação, etc).

### Fase de implementação:

Fazer os registros do que aconteceu com os testes implementados na arquitetura preparada.

### Encerramento:

Discutir os resultados obtidos pelo teste, criando um relatório para a equipe de desenvolvimento e interessados.



## Exercício de fixação

### Relacione cada princípio de teste com sua definição.

1	Não se pode provar que não há defeitos. Ele reduz a probabilidade de defeitos não encontrados no software, mas mesmo se eles não forem encontrados, o processo de teste não é uma demonstração de correção.	Teste mostra a presença de defeitos e não a sua ausência.	▼
2	Não é possível testar tudo, exceto em casos triviais. Em vez de tentar realizar testes exaustivos, devemos utilizar análises de riscos, técnicas de teste e de prioridades para centralizar os esforços do teste.	Testes exaustivos são impossíveis.	▼
3	Para detectar defeitos de forma prematura, as atividades de teste, tanto estáticas quanto dinâmicas, devem ser iniciadas o mais cedo possível no ciclo de vida do desenvolvimento de software para ajudar a reduzir ou eliminar alterações caras.	Teste prematuro economiza tempo e dinheiro.	▼
4	Em geral, um pequeno número de módulos contém a maioria dos defeitos encontrados durante o teste de pré-lançamento, ou é responsável pela maioria das falhas operativas.	Os defeitos são agrupados.	▼
5	Se os mesmos testes forem repetidos inúmeras vezes, eventualmente esses testes não encontrarão mais nenhum novo defeito. E para detectar um novo defeito, é importante revisar e atualizar regularmente os testes e os dados de teste existentes para adaptá-los e potencialmente encontrar mais falhas.	Paradoxo do pesticida.	▼
6	Dependendo do propósito do sistema ou da indústria, diferentes aplicações devem ser testadas diferentemente.	O teste depende do contexto.	▼
7	O sucesso de um sistema não depende apenas de encontrar erros e corrigi-los até que desapareçam, pois pode não haver erros mas surgir outros problemas. Existem outras variáveis que devem ser consideradas no momento de medir o sucesso.	A ausência de erros é uma falácia.	▼

**Mesa 03:**

Micheli Pinheiro

Diego Reis

Alcilene Reis

Lika Oyama

Jorge Silva