



O novo ciclo de vida dos testes de software

O que é o Ciclo de Vida de um Software?

O ciclo de vida de um software compreende todas as atividades que ocorrem desde o momento em que ele é idealizado até sua entrega final. Assim como o desenvolvimento tem seu ciclo, o teste também segue uma série de fases que dependem da maturidade e do momento do projeto.

A principal importância de entender esse ciclo, especialmente no contexto ágil, é a **antecedência** com que as atividades de teste podem e devem acontecer. Quanto antes o teste é envolvido no processo, mais rápido o software fica pronto e com mais qualidade, pois ajuda a preencher lacunas e a responder questionamentos que não foram pensados inicialmente.

O Modelo Tradicional em Cascata (Sequencial)

Antigamente, o desenvolvimento de software seguia um modelo sequencial, conhecido como "Cascata", com fases bem definidas e rígidas:

1. **Análise:** Um analista de requisitos levantava as necessidades do cliente por meio de entrevistas e documentação.
2. **Design:** Com os requisitos aprovados, a equipe de arquitetura desenhava a solução, definindo componentes e integrações.
3. **Implementação:** Os desenvolvedores começavam a codificar, transformando o design em um produto funcional.
4. **Teste:** A equipe de testes recebia toda a documentação das fases anteriores e começava seu trabalho.
5. **Manutenção:** Após a entrega, qualquer correção ou ajuste era feito nesta fase.

O Problema do Teste no Fim do Processo

Nesse modelo, o teste era a última etapa antes da manutenção. A equipe de testes enfrentava um grande desafio:

- **Falta de Tempo:** Grande parte do tempo era consumida apenas para tentar entender o que foi feito nas fases anteriores (requisitos, arquitetura, etc.).
- **Defeitos Básicos:** Muitas perguntas se transformavam em defeitos que poderiam ter sido evitados se a equipe de testes estivesse envolvida antes.
- **Fase de Manutenção Eterna:** Como muitos problemas eram descobertos tardiamente, a fase de manutenção se tornava longa e custosa.

O Modelo Ágil: Uma Abordagem Integrada e Dinâmica

O modelo ágil não elimina as fases de análise, design e implementação, mas as executa de forma diferente:

- **Ciclos Curtos (Sprints):** As atividades acontecem repetidamente em ciclos menores e controláveis.
- **Envolvimento Contínuo do Testador (QA):** O testador participa de **todas as fases** desde o início. Ele não espera o desenvolvimento terminar para começar a trabalhar.

Enquanto a definição do produto ocorre, o testador já está:

- **Fazendo perguntas** e esclarecendo dúvidas.
- **Planejando os testes** e o ambiente.
- **Entendendo o design** para decidir se testes de componentes isolados fazem sentido.
- **Criando simuladores (mocks)** para adiantar a validação de integrações.

A Mudança no Papel e no Objetivo do Teste

Essa mudança de processo alterou fundamentalmente o objetivo e a responsabilidade do teste:

- **Objetivo:** Não é mais "achar o defeito" ou "quebrar o software". O objetivo agora é **ajudar a entregar um produto de qualidade o mais rápido possível**.
- **Responsabilidade:** O modelo de "passar o bastão" entre equipes foi substituído por uma **responsabilidade compartilhada**. A qualidade é um compromisso de todos no time.

Com essa abordagem integrada, o processo se torna mais dinâmico, o backlog de defeitos diminui e a equipe de testes já sabe o que e como testar quando a funcionalidade é

entregue.

Questões sobre o Material

1. Qual era o principal problema enfrentado pela equipe de testes no modelo tradicional em Cascata, segundo o texto?

- a) A falta de documentação clara sobre os requisitos do cliente.
- b) O excesso de reuniões com a equipe de desenvolvimento.
- c) Envolver-se apenas no final do processo, gastando muito tempo para entender o que foi feito e tendo pouco tempo para testar.
- d) A dificuldade em automatizar os testes devido à arquitetura do sistema.

2. Qual é a diferença fundamental no envolvimento do testador (QA) ao comparar o modelo Ágil com o modelo tradicional?

- a) No modelo Ágil, o testador é responsável por codificar parte do sistema.
- b) No modelo Ágil, o testador participa de todas as fases desde o início, enquanto no tradicional ele atuava principalmente no final.
- c) No modelo tradicional, o testador definia os requisitos, e no Ágil ele apenas os valida.
- d) No modelo Ágil, o testador não precisa se preocupar com o ambiente de testes.

3. Como o objetivo principal do teste de software mudou com a adoção de metodologias ágeis, de acordo com o texto?

- a) O objetivo mudou de "garantir zero defeitos" para "documentar todos os bugs".
- b) O objetivo mudou de "validar a arquitetura" para "testar apenas a interface do usuário".
- c) O objetivo mudou de "achar o defeito" para "ajudar a entregar um produto de qualidade o mais rápido possível".
- d) O objetivo continua o mesmo, apenas o método de execução foi alterado.

4. O que um testador faz durante as fases de definição e design em um time ágil, segundo o texto?

- a) Espera a implementação terminar para receber a documentação final.
- b) Apenas observa as discussões sem interagir com a equipe.

c) Escreve o código junto com a equipe de desenvolvimento.

d) Faz perguntas, planeja os testes, entende os componentes e pode criar simuladores (mocks).

5. **Como o modelo de responsabilidade pela qualidade do software é descrito no modelo Ágil em comparação ao tradicional?**

a) No modelo Ágil, a responsabilidade é exclusiva do testador, enquanto no tradicional era do desenvolvedor.

b) No modelo tradicional havia uma divisão de responsabilidades por fase, enquanto no Ágil a responsabilidade é compartilhada por toda a equipe.

c) A responsabilidade não muda, continuando a ser do gerente de projetos em ambos os modelos. d) No modelo Ágil, o cliente se torna o único responsável pela qualidade final do produto.

Gabarito

1. **c)** Envolver-se apenas no final do processo, gastando muito tempo para entender o que foi feito e tendo pouco tempo para testar.

2. **b)** No modelo Ágil, o testador participa de todas as fases desde o início, enquanto no tradicional ele atuava principalmente no final.

3. **c)** O objetivo mudou de "achar o defeito" para "ajudar a entregar um produto de qualidade o mais rápido possível".

4. **d)** Faz perguntas, planeja os testes, entende os componentes e pode criar simuladores (mocks).

5. **b)** No modelo tradicional havia uma divisão de responsabilidades por fase, enquanto no Ágil a responsabilidade é compartilhada por toda a equipe.