

Curso: Testes Automatizados

Micro 2 - Aula 1 –Testes Ágeis

Olá, seja bem-vindo(a)!

Esta aula tem como objetivo definir para você as regras e princípios do desenvolvimento ágil de software, focado nos testes. Você já conheceu algumas características do teste automatizado, suas boas práticas, vantagens, desvantagens e aplicações. Agora, nesta aula, você vai aprender como os testes são construídos, segundo a metodologia ágil.

Antes da cultura ágil ser difundida, os testes, na maioria dos casos, faziam parte apenas de uma única fase no ciclo de desenvolvimento de software. Logo após a codificação, os testes só eram realizados na conclusão do processo, depois que as funcionalidades já estavam prontas. Seguindo esse procedimento, os softwares, comumente, tinham baixa qualidade, falhas e erros, que eram descobertos apenas na fase final. A partir do desenvolvimento da estrutura ágil de software, esse problema começou a ser mitigado, posto que o teste passou a ser empregado desde o início do processo, atravessando a fase de análise e levantamento de requisitos, envolvendo, inclusive, o cliente neste andamento.

O teste ágil, portanto, não é sequencial (no sentido em que é executado somente após a fase de codificação), mas contínuo. Então, este pode começar no início do projeto, com integração contínua entre desenvolvimento e teste. Você pode perceber que, diferentemente do modelo em cascata, em um modelo ágil, o plano de teste é desenvolvido e atualizado para cada versão do sistema. O modelo ágil inclui tipos de teste realizados em cada iteração, como requisitos de dados de teste, infraestrutura, ambientes de teste e resultados de teste. Os planos de teste típicos no ágil possuem o escopo do teste; novas funcionalidades que estão sendo testadas; nível ou tipos de teste com base na complexidade dos recursos; teste de carga e desempenho; infraestrutura; plano de mitigação ou riscos; recursos; e resultados finais.

Portanto, se você deseja levar a sério os testes automatizados para o seu software, existe um conceito-chave que você não pode deixar de conhecer: a pirâmide de teste. Ficou curioso(a), não é? Então, vamos lá!

O programador americano Mike Cohn, visando possibilitar às empresas um melhor uso de processos e técnicas ágeis, criando organizações de desenvolvimento com alto desempenho, apresentou em seu livro, “Succeeding with Agile”, o conceito de pirâmide de teste. A metáfora da pirâmide facilita para você pensar em diferentes camadas de teste e informa o que deve ser feito em cada camada.

Devido à sua simplicidade, a essência da pirâmide de teste serve como uma boa regra geral quando se trata de estabelecer seu próprio conjunto de testes. O modelo divide os tipos de teste em três camadas, com base no retorno do investimento oferecido pela automação. Os componentes de cada camada podem variar de um software para outro, mas a camada inferior (a maior parte da estrutura da pirâmide), normalmente, inclui testes de unidade, representando a ideia de que a automação oferece o melhor ganho para o cliente. Deu para entender?

Então, se você já tem alguma experiência com testes, vai ser muito fácil. E mesmo que você não tenha experiências prévias, não se preocupe, você vai aprender o passo a passo necessário para dominar o uso de testes automatizados.

Dando continuidade, você agora vai entender mais detalhadamente como funciona a pirâmide, está bem? A seguir, a imagem de uma pirâmide apresenta as informações sobre os níveis de teste. A pirâmide de teste é dividida em três partes. A base da pirâmide, representa os “Unit Tests” (Testes Unitários ou Testes de Unidade). O centro, os “Service Tests (Testes de Serviço)” e o topo representa os “UI Tests” (Testes de Interface do Usuário). À esquerda da pirâmide, há uma seta dupla que vai da base até seu topo. Na extremidade inferior da seta, está escrito: “more isolation” (maior isolamento); na superior, está escrito “more integration” (maior integração). À direita da pirâmide, há uma seta dupla que vai da base até seu topo. Na extremidade inferior da seta, está escrito: “faster” (mais rápido) e na superior, está escrito “slower” (mais devagar).

Para você compreender melhor o funcionamento destes níveis, tenha em mente que os testes de unidade formam a base sólida da pirâmide, importante para demonstrar se o novo código está funcionando corretamente: o desenvolvedor que escreveu o código é qualificado, exclusivamente, para verificar se seus testes cobrem o código. A cada linha, método ou classe criada, um teste também é implementado. Na camada do meio, temos os testes de serviço ou de integração: eles, geralmente, garantem que o código está corretamente integrado com serviços externos, além de qualquer alteração na lógica de negócios poder ser rapidamente convertida em código de teste equivalente, são mais lentos que os testes de unidade e podem depender de recursos externos. No último nível, os testes de aceitação ou testes de UI, fornecem o próximo maior benefício e esta é a camada em que você testaria o produto da perspectiva do usuário final.

Como testador, sua principal prioridade seria garantir que os fluxos de testes estivessem alinhados com os requisitos. Logo, as equipes ágeis e eficientes podem acelerar todo o processo de teste, minimizando o tempo necessário para converter os critérios de aceitação de histórias de usuários em testes de automação.

Ficou mais claro para você? Nesta aula, entendemos o que é um teste ágil e como ele é organizado dentro da estrutura de desenvolvimento de software, a partir da Pirâmide de Cohn, que nos traz uma visão facilitada e completa sobre o produto e/ou software em questão.

É importante ressaltar que para uma melhor fixação do conteúdo, é essencial que você coloque em prática o que aprendeu na aula. Fique à vontade para explorar as possibilidades de testes automatizados e proceder da melhor forma que se adeque às suas necessidades. Por hoje é só!

Bons estudos!

Referências

COHN, M. **Succeeding with Agile: Software Development Using Scrum**. Addison-Wesley Professional, 2009 – First Edition.

CRISPIN, L. et al. **Agile Testing: A practical guide for testers and agile teams**. Pearson Education, 2009 – Second Edition.

Página da Martin Fowler. Disponível em: <<https://martinfowler.com/articles/practical-test-pyramid.html>>.

The Practical Test Pyramid. Vocke, H. 2018. Disponível em: <<https://martinfowler.com/articles/practical-test-pyramid.html>>. Acesso em: 04 mai. 2020.