

A ascensão do software de gerenciamento de teste

Se a sua empresa é como a maioria das organizações que tem realizado o teste automatizado há um tempo, você provavelmente está observando um aumento das linguagens de programação. Graças a formas mais específicas de abstração nas atuais linguagens de programação de nível mais elevado, esse problema não desaparecerá tão cedo. Na verdade, está se intensificando..

Como, então, chegamos a esse estágio? A evolução está totalmente baseada na história das linguagens de programação e no anseio por maiores níveis de abstração e começa com a primeira linguagem de programação de alto nível, FORTRAN. Desenvolvida em 1953 por John Backus, o FORTRAN atendeu à necessidade de um maior nível de abstração nos

vieram níveis mais poderosos de abstração, como a programação orientada a objeto que é hoje uma das estruturas de programação mais utilizadas. Além disso, esses modelos de computação mais novos muitas vezes são desenvolvidos para solucionar problemas comuns. Alguns deles são desenvolvidos para tarefas de programação de uso geral e outros para uma aplicação específica. Por exemplo, o LabVIEW foi desenvolvido para aplicações de controle, teste e medição, e o Python para tarefas rápidas de script.

Esses níveis crescentes de abstração resultam em linguagens mais adequadas para tarefas específicas. Hoje, os melhores gerentes de teste projetam sistemas de teste que aproveitam a capacidade de diversas linguagens e economizam o tempo de desenvolvimento usando um software de gerenciamento de teste.

Desenvolvimento de sistemas de teste tradicionais

Considerando que o software é a base da automação ao desenvolver um sistema de teste, muitas empresas preferem padronizar uma única linguagem usada para todos os aspectos do projeto do sistema de teste, do teste de componentes individuais ao gerenciamento de teste. O resultado é o desenvolvimento de uma abordagem de software de teste homogênea. A principal vantagem é que todos os membros da equipe podem trabalhar em um único ambiente padronizado, o que facilita o compartilhamento de bibliotecas e de módulos de código entre a equipe. O treinamento para essa abordagem também é bem simplificado, já que a equipe trabalha em um único ambiente.

processos das máquinas criados com base no modo como os humanos comunicam naturalmente suas ideias — através da linguagem.

Após o sucesso do FORTRAN, mais linguagens, como C, Pascal, ATLAS e PAWS, foram desenvolvidas, cada uma apresentando novas estruturas e modelos de computação. Junto com cada nova linguagem

No entanto, padronizar uma única linguagem também tem suas desvantagens. Essa padronização pode limitar novas contratações a uma determinada qualificação ou forçar novos funcionários a aprender novas ferramentas. Esse tópico foi uma das tendências abordadas no e-book de 2014. Quando os alunos se graduam, muitas vezes preferem ter experiência em uma ou mais linguagens específicas. E quando novos gerentes assumem o cargo, eles geralmente optam por implementar uma linguagem de sua escolha. Isso pode ser dispendioso e muitas vezes exige migração do código, revalidação do código base e treinamento na nova linguagem.

Os melhores gerentes de teste precisam buscar uma abordagem mais nova para o desenvolvimento de sistemas de teste que cria um sistema heterogêneo com diversas linguagens. Com essa abordagem, a equipe pode usar diversas linguagens, cada uma com suas vantagens, para criar sistemas de teste mais poderosos. Como todas as linguagens são projetadas para aplicações específicas, usar cada uma de suas vantagens economizaria tempo e dinheiro.

Embora benéfica, essa abordagem pode apresentar um novo desafio para o desenvolvimento de sistemas de teste: linguagens diferentes precisam agora se comunicar e trabalhar juntas para formar um único sistema. Para solucionar essa questão, todos os engenheiros de teste precisam entender o ambiente no qual se especializam, além do ambiente de todas as demais linguagens utilizadas.

A solução de software

Os departamentos de teste estão agora migrando para um software de gerenciamento de teste pronto para

uso que opera como um Rosetta Stone integrando as diferentes linguagens. Além de oferecer aos usuários um ambiente comum onde eles podem trabalhar com qualquer tipo de código de teste, o software executa tarefas executivas, como sequenciamento e chamada de cada teste, tratamento do registro de dados e geração de relatórios. Desse modo, cada engenheiro pode se concentrar em escrever o melhor teste para cada componente do DUT sem precisar se preocupar em como se comunicar com outras partes do código. Uma vez que os engenheiros podem usar os ambientes mais cômodos para eles, a empresa pode se dedicar em contratar engenheiros especializados em determinadas aplicações, mesmo que eles não tenham conhecimento prévio ou treinamento na linguagem que a empresa precisa.

Além disso, os gerentes de teste podem aproveitar todo o potencial de um projeto heterogêneo enquanto evitam os novos desafios que esse projeto apresenta. Isso inclui o uso do software de gerenciamento de teste para um processo de desenvolvimento mais modular, que produz um sistema mais fácil de manter e atualizar porque cada componente pode ser atualizado individualmente sem afetar o resto do sistema.

Por fim, os executivos de teste normalmente incluem o suporte de fornecedores que atualizam e corrigem o software continuamente, o que reduz ainda mais o custo de manutenção e aumenta a sustentabilidade desses sistemas. Unindo essas vantagens aos benefícios de um projeto de sistema de teste heterogêneo é como os melhores gerentes de teste estão criando o futuro do teste automatizado.

ABORDAGENS HETEROGÊNEAS DE DESENVOLVIMENTO DE TESTE

