

Documento técnico de negócios

Guia de sobrevivência para testar aplicativos modernos



Por que modernizar os aplicativos?

De acordo com uma pesquisa da Forrester, a agilidade, a inovação e economia de custos são os principais fatores que estimulam as iniciativas de modernização dos aplicativos.¹ Em tempos de expectativas imediatas e nesse mundo em constante mudança, a organização capaz de se adaptar rapidamente contribuirá para o sucesso dos negócios. A inovação é a chave para acelerar o crescimento e se diferenciar perante à concorrência. No caso dos aplicativos de negócios, isso significa proporcionar aos clientes os recursos que eles estão acostumados a usar em suas interações em mídias sociais: mais controle sobre como o conteúdo é apresentado, mais interação com o aplicativo e com outras pessoas, e mais experiências integradas, como vídeo. E tudo isso precisa ser alcançado rapidamente e de forma a promover a otimização, a fim de reduzir os custos operacionais e aprimorar os processos operacionais.

Para atender a essas necessidades comerciais, a empresa ágil está adotando novas tecnologias e arquiteturas, como a computação em nuvem, Web 2.0, aplicativos compostos e mobilidade, assim como novas metodologias como o Agile e outros processos de desenvolvimento iterativos. Essas novas e importantes iniciativas de TI podem afetar comercialmente as empresas, mas, do mesmo modo, também podem tornar os testes funcionais mais desafiadores para a organização de controle de qualidade (QA).

Resumo executivo

Neste documento, você descobrirá como as equipes de testes funcionais podem enfrentar os desafios oriundos dessas novas iniciativas de TI.

Os processos de desenvolvimento iterativos, como o Agile, trazem à tona a necessidade de realizar testes com prontidão e rapidez e baseados em menos documentação disponível sobre o aplicativo. Esses processos iterativos geraram uma necessidade crescente de executar testes exploratórios, em que o projeto e a execução do teste são realizados ao mesmo tempo.

As novas tecnologias dos aplicativos, como as arquiteturas de aplicativos compostos, a computação em nuvem e as plataformas móveis criam seus próprios desafios. Esses desafios incluem trabalhar com novos kits de ferramentas, realizar testes em várias plataformas e dispositivos, na nuvem e nas instalações da empresa, e testar aplicativos complexos com várias camadas pertencentes ou não à interface gráfica de usuário (GUI) que podem estar interconectadas com camadas de integração e intermediários.

Para atender a esses desafios, as equipes de QA devem mudar a maneira como os testes funcionais são realizados. As melhores práticas incluem iniciar o processo que envolve os testes o mais rápido possível, mesmo antes de um aplicativo estar pronto para ser testado, voltando-se, assim, para um quadro geral a fim de descobrir o que precisará ser testado e investindo antecipadamente tempo para projetar uma estratégia de teste eficiente e pronta para mudanças. Considere usar uma estrutura

com componentes de testes reutilizáveis para tornar a criação e a manutenção de testes mais eficientes. Implemente métodos modernos de testes manuais para eliminar as tarefas repetitivas e sujeitas a erros provenientes dos testes manuais e promova uma comunicação aperfeiçoada. Procure novas oportunidades de automatizar os testes. E para obter testes mais completos dos aplicativos compostos, teste a camada pertencente ou não à GUI e todo o processo empregando, sempre que possível, a automação.

As iniciativas de TI causam dores de cabeça para as equipes de teste

O influxo contínuo de tecnologias e processos gera mais complexidade e exige ciclos menores de testes. Uma equipe de teste despreparada pode ser um obstáculo para a agilidade dos negócios e para um lançamento mais rápido no mercado. Veja algumas razões pelas quais os testes funcionais podem ser particularmente difíceis para essas iniciativas de TI.

Agile: As organizações de TI estão adotando o Agile, os processos híbridos e outros processos de desenvolvimento de aplicativos iterativos para acelerar o lançamento no mercado e reduzir os custos das mudanças. Esses processos fornecem um mecanismo para incorporar feedbacks e minimizar o tempo gasto em fases longas de planejamento. Os métodos de testes funcionais precisam ser alterados de maneira apropriada para que não eliminem os benefícios do Agile e dos métodos similares ao Agile. Os testes precisam ser realizados antecipadamente no ciclo de vida, muitas vezes em paralelo com o desenvolvimento. Em muitos casos, tanto os testadores quanto os desenvolvedores participam dos testes de maneira conjunta para manter uma interação que obedece o cronograma. Os testadores precisam realizar seu trabalho mesmo que tenham à disposição uma quantidade mínima de documentação sobre o aplicativo. Além disso, eles ainda precisam transmitir uma descrição abrangente sobre as falhas, a fim de promover uma resolução rápida. Frequentemente, eles também precisam testar apenas uma parte parcial dos aplicativos e dos recursos em vez da versão final completa.

Esses desafios geram uma necessidade crescente de implementar práticas como os testes exploratórios. Após contatarem os principais envolvidos para entender os objetivos de negócios de um aplicativo e seus casos de uso, os testadores devem empregar seu conhecimento e habilidades, incluindo a criatividade e o pensamento crítico, para explorar rapidamente o aplicativo à procura de falhas. O maior benefício desse tipo de teste é a economia de tempo gerada, tanto na execução dos testes quanto na redução das iniciativas voltadas à documentação e projeto do teste.

Aplicativos compostos A organização de QA deve desenvolver habilidades e recursos para testar aplicativos modernos e compostos com arquiteturas complexas, incluindo GUIs ricas, camadas de lógica de negócios não pertencentes à GUI, serviços compartilhados (frequentemente aqueles baseados em SOA) e integrações de aplicativos fornecidos como pacotes. É necessário não apenas validar os serviços e componentes pertencentes ou não à GUI, como também visualizar, validar e informar sobre um cenário de teste integrado do processo de negócios que cruza as várias camadas de um aplicativo composto. No entanto, quando se trata de testar componentes não pertencentes à GUI, é mais fácil falar do que fazer. Testá-los através da GUI do aplicativo significa esperar muito no ciclo de vida do aplicativo para, enfim, descobrir bugs em um momento em que sua resolução é mais cara. Realizar os testes através da GUI pode concentrar sua ação apenas em um subconjunto da funcionalidade das camadas de lógica, deixando as falhas à mercê de serem descobertas em outro local do aplicativo ou quando outro aplicativo o executar. E se você assumir o desafio de testar componentes e serviços não pertencentes à GUI, você poderá acabar por descobrir que suas ferramentas de testes atuais não suportam as tecnologias necessárias. Além disso, as interfaces e os comportamentos esperados para os componentes não pertencentes à GUI não são muito bem compreendidos ou documentados em comparação aos componentes baseados na GUI.

Por exemplo, como você testa uma transação bancária que inicia um depósito, executa uma chamada segundo um banco de dados e envia uma mensagem de texto ao cliente ao mesmo tempo que confirma ao usuário que o depósito foi realizado com êxito? Muitos desses processos não revelarão sua funcionalidade através da GUI e devem ser validados de outra forma. E se o serviço que envia uma mensagem de texto ao cliente também for usado em uma transação que envia um aviso de saque a descoberto, haverá nesse caso falhas adicionais que passaram despercebidas?

Com a ocorrência de tantos componentes usados por vários aplicativos, pode ficar difícil para a equipe de testes entender as dependências entre os próprios componentes e os aplicativos que os utilizam, principalmente quando algum dos serviços estiver indisponível ou for "caixas pretas" fornecidas por terceiros.

Em razão de todas essas permutações de casos para teste necessárias para testar integralmente um aplicativo composto complexo, muitas organizações empregarão a automação de testes; todavia, há neste campo desafios inerentes. O trabalho necessário para automatizar os testes normalmente é subestimado. A automação nada mais é que o desenvolvimento do software e, por isso, precisa ser administrada por um corpo de funcionários qualificado no campo de projeto, validação e manutenção. Além disso, encontrar um conjunto de ferramentas para acomodar todos ou a maioria dos cenários de testes necessários pode ser difícil. A utilização de ferramentas especializadas para tecnologias ou aplicativos diferentes pode gerar custos maiores com ferramentas e treinamento, além de limitar a habilidade de padronizações na organização. Ademais, temos o problema dos scripts de teste mudando de proprietários ao longo do tempo, sem a geração de uma documentação adequada. Quase sempre esses testes são executados durante anos sem saber por que são importantes ou a quais requisitos eles atendem.

Veja os quadros Nuvem e Web 2.0 para obter mais detalhes.

Móvel: Os aplicativos corporativos referentes às interfaces móveis devem ser testados de modo tão completo como no caso de suas contrapartes provenientes do PC. Os usuários, cujas expectativas por qualidade foram definidas de acordo com aquelas encontradas nos aplicativos dos smartphones pessoais, não têm paciência com aplicativos corporativos móveis que não funcionam de forma adequada. Dessa forma, o teste que visa assegurar que esses aplicativos funcionam corretamente é essencial para a produtividade e fidelidade do cliente. Os aplicativos móveis, no que se referem aos testes, compartilham todas as complicações encontradas nos aplicativos compostos, no entanto, eles não imitam simplesmente o ambiente de desktop. Eles possuem seus próprios requisitos para a interface do usuário, fluxos de processos de negócios e dependências de infraestrutura. E eles são particularmente desafiadores para a organização de QA em razão do grande número de dispositivos possíveis que precisam de suporte. De acordo com a Forrester, "Seus clientes possuem vários dispositivos diferentes que contam com ambientes de desenvolvimento diferentes...As várias versões do mesmo SO, os diferentes tamanhos de tela e um alto número de dispositivos significam que os custos com a portabilidade, manutenção e promoção de aplicativos permanecerão altos."² Os testes manuais podem ser extremamente trabalhosos em smartphones, PDAs, tablets e equipamentos especializados, como dispositivos portáteis de impressão e escaneamento de agentes de locadoras de veículos. A situação é ainda pior quando se considera as permutações exponenciais de dados, cenários de utilização, condições da rede móvel e provedores. Embora a implementação possa parecer desafiadora, essa é uma área perfeita para as soluções de automatização de testes.

Aplicativos da nuvem

As equipes de testes funcionais normalmente precisam testar aplicativos compostos que incluem alguns serviços que se baseiam na nuvem. Para testar integralmente o aplicativo composto, incluindo a camada da nuvem, a equipe de teste deve ser capaz de acessar e gerenciar um projeto de teste em várias infraestruturas.

Além disso, a equipe de teste pode encontrar dificuldades em acessar determinados serviços baseados em nuvem para a realização de testes. Por exemplo, um aplicativo composto para o pedido de um par de sapatos on-line envolverá serviços baseados em nuvem, tal como o processamento de cartão de crédito e os serviços de envio. Levando em conta que os cartões de crédito e o envio correspondem a serviços de terceiros que provavelmente já estão em produção, a realização de um novo teste pode custar caro ou não ser possível. Isso pode gerar atrasos no teste do aplicativo composto.

Por outro lado, o acesso à nuvem traz ótimas oportunidades para a equipe de teste visto que ele proporciona uma capacidade sob demanda e aparentemente ilimitada aos aplicativos de teste. Isso é particularmente útil para o teste de desempenho.

Como modernizar os testes para uma empresa ágil

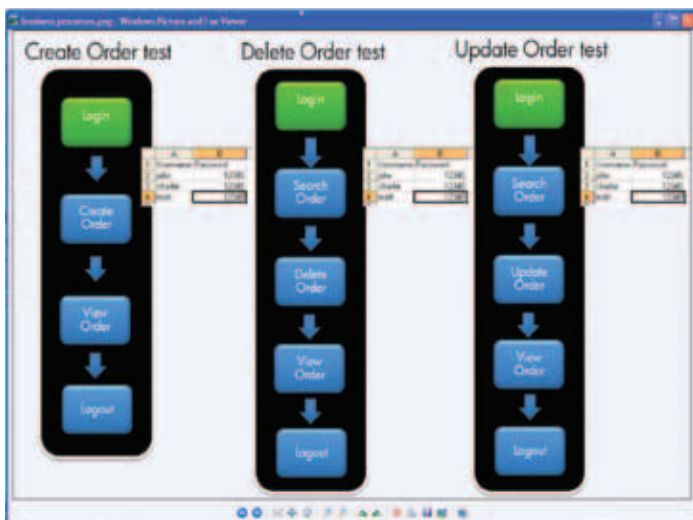
As organizações de QA podem atender a essas questões e se tornar parceiras plenas na implementação dos aplicativos modernos atuais, agindo como uma empresa verdadeiramente ativa. As organizações de QA podem se preparar e se modernizar ao expandirem seus conjuntos de habilidades e se envolverem mais com os aspectos comerciais daquilo que estão sendo solicitadas para testar. Elas devem realizar parcerias com as equipes de analistas de desenvolvimento e negócios ao compartilharem métricas e resultados. Elas devem se envolver com as iniciativas de gerenciamento da qualidade que resultam em fornecimento mais previsível dos aplicativos e, conseqüentemente, de melhor satisfação do cliente e menos falhas. Elas podem alcançar tudo isso ao empregarem processos e práticas que permitam a realização antecipada de testes no ciclo de vida, aumentando, assim, a automação sempre que possível e adquirindo maior confiança ao executar testes sem a necessidade de documentos detalhados sobre o projeto do teste. Em geral, as organizações de QA com visão de futuro veem suas práticas de teste de modo holístico, investem no projeto do teste e como resultado: testam mais, de maneira mais rápida e mais breve.

Eis algumas etapas específicas que a equipe de testes funcionais pode seguir para promover a modernização:

1. Comece com o projeto: Embora essa etapa seja normalmente negligenciada, o benefício de dedicar um determinado tempo para projetar cuidadosamente a estratégia de testes é percebido ao tornar os testes mais eficientes e fáceis de manter. Recomendamos o uso de uma estrutura—através da criação de um processo de negócios e da construção de uma biblioteca de componentes de testes reutilizáveis que podem ser montados em fluxos.

Figura 1

Reduza o tempo de criação e manutenção de testes ao validar os processos de negócios com componentes de testes reutilizáveis.



Por exemplo, vamos considerar que existam três processos de negócios que precisam ser validados: criação do pedido, remoção do pedido e atualização do pedido. Uma vez que os processos de negócios são modelados, fica fácil identificar a duplicação; todos três usam a mesma função de login (ver Figura 1). Com uma abordagem modular, o componente de teste para a função de login precisa apenas ser criado uma vez, não três vezes, eliminando a duplicação do esforço. Se uma mudança for feita na função de login, só é necessário mudar uma vez o componente de teste associado, e todos os testes que usam esse componente serão atualizados. Esse método de "teste por composição" pode reduzir a criação dos testes, e o esforço e tempo de manutenção dos testes. E essa maior eficiência é obtida sempre que os testes são automatizados ou realizados manualmente.

O software HP Business Process Testing (HP BPT) fornece uma estrutura para a criação de uma biblioteca de componentes de testes reutilizáveis que pode ser montada em fluxos. Ele permite que as equipes de teste aumentem a qualidade do aplicativo ao diminuir o tempo necessário para a criação e manutenção dos testes. Além de reduzir o custo da qualidade em razão da mudança rápida e regular.

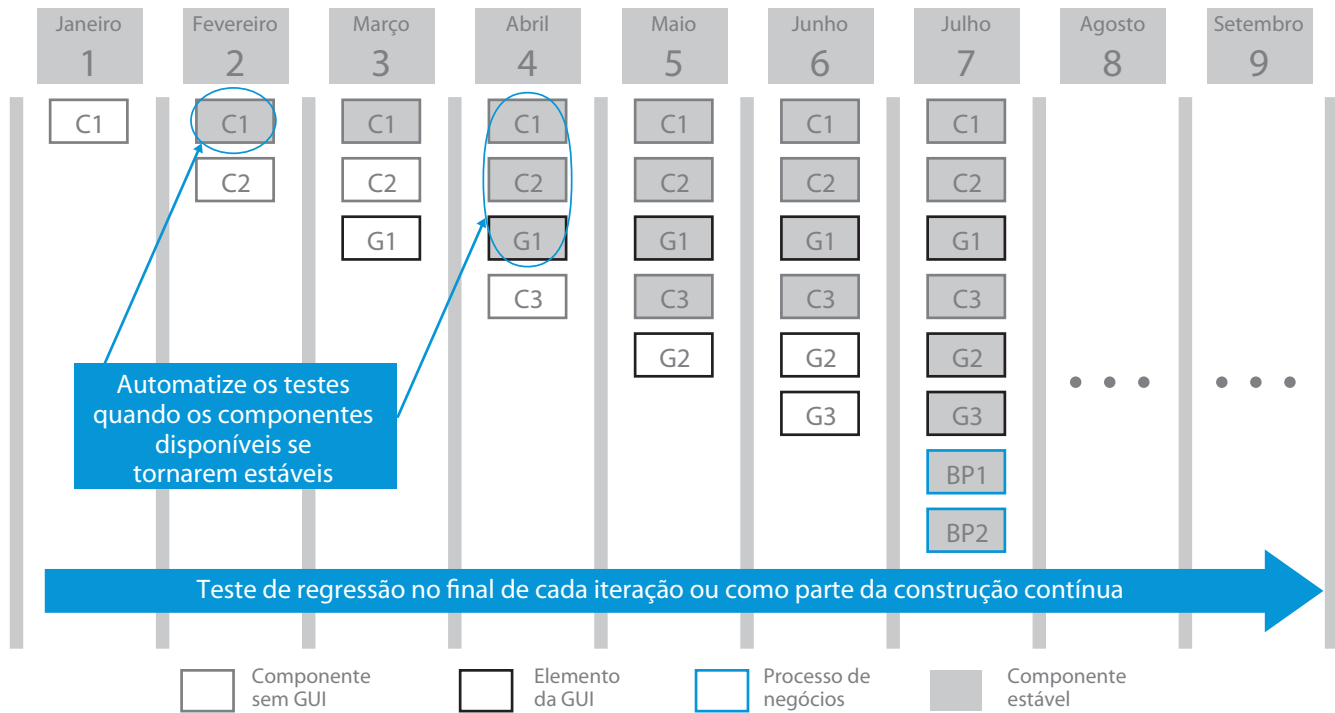
Outra vantagem de dedicar um tempo antecipado para projetar uma estratégia de teste com o HP BPT é devido à habilidade de determinar rapidamente quais componentes testar manualmente e quais automatizar. Na maioria dos casos, os componentes serão em primeiro plano testados manualmente e, em seguida, passarão a ser testados de maneira automatizada sempre que isso for o mais apropriado. O HP BPT oferece o bônus adicional de autogerar documentação de planos de teste e criar vínculos detalhados com os requisitos de software do Centro de Qualidade da HP. Os testes bem documentados e bem gerenciados continuam a ser úteis ao longo das iterações futuras de testes, mesmo que os testes mudem de proprietários.

Aplicativos Web 2.0

As tecnologias que envolvem o Aplicativo Rico de Internet (RIA) e o Web 2.0 são conhecidas por fornecer uma GUI rica que traz seu próprio conjunto de desafios quando se trata de teste. As ferramentas de testes funcionais devem ser capazes de testar um conjunto diverso e cada vez mais expandido de kits de ferramentas, como aqueles que aproveitam o Microsoft® Silverlight™, e a ampla gama de ofertas Ajax, incluindo o Dojo Toolkit, Yahoo! User Interface (YUI), ASP.NET AJAX e o Google® Web Toolkit (GWT), para citar alguns. Embora eles não alterem grande parte da infraestrutura de web subjacente, cada kit de ferramenta possui controles de web diferentes com comportamentos exclusivos, que podem não ser automaticamente reconhecidos pelas ferramentas de testes funcionais. Combine a situação com a natureza dinâmica dos dados e dos dados compartilhados entre aplicativos e você terá um ambiente de teste complexo.

Figura 2

É útil testar manualmente a iteração atual e automatizar os testes da iteração anterior.



2. Aumente e melhore a automação: Embora a automação nunca substituirá completamente os testes manuais, encontrar áreas para automatizar ajudará a organização de QA a realizar mais testes e de maneira mais rápida. Ao possibilitar que as equipes de QA testem mais configurações e iterações com recursos limitados, a automação ajuda a promover uma melhor cobertura do teste, o que pode reduzir o risco de falhas na produção.

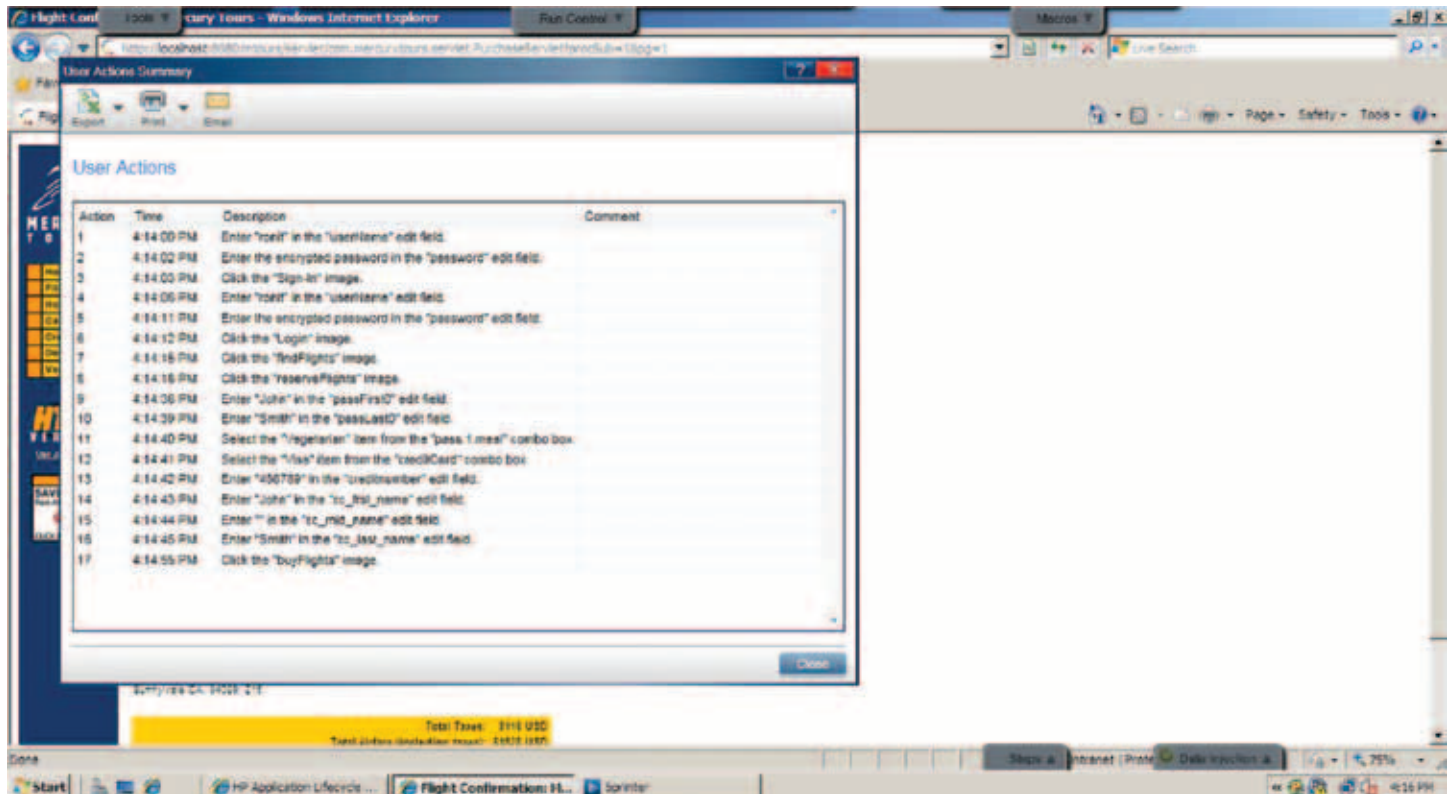
Para tirar o máximo proveito da automação, é importante aplicá-la de maneira prudente. Veja as nossas recomendações sobre o que e quando automatizar:

- Testes de regressão – Os testes de regressão são um conjunto de testes "básicos" utilizado junto com testes manuais, mas de fácil automação. Eles se concentram menos em encontrar bugs em um novo lançamento e mais em se certificar de que as alterações mais recentes não danificaram nada que estava funcionando previamente. Utilize a automação para criar conjuntos de testes de regressão repetíveis e de rápida execução que podem ser executados 24 horas por dia. Incorporar esses testes de regressão automatizados no processo de construção pode habilitar uma prática de "construção contínua" que encontra rapidamente os problemas e mantém os desenvolvedores sempre trabalhando em base estável e conhecida.

- A iteração anterior – Em métodos iterativos, como o Agile, pode ser difícil automatizar e recuperar resultados em uma iteração ou sprint. Frequentemente não há tempo suficiente, e o código pode ainda estar em fluxo. Mas, de maneira semelhante aos testes de regressão, é útil testar manualmente a iteração atual e automatizar os testes da iteração anterior (ver Figura 2). Isso ajuda as equipes de TI a realizar testes com mais prontidão e rapidez – duas características necessárias para um processo Agile.
- Aplicativos e componentes mais estáveis – É importante analisar o retorno do investimento (ROI) da automação. Os benefícios da automação com o tempo deve compensar os custos com o desenvolvimento e manutenção dos scripts de automação. Embora algumas ferramentas, como o software HP Unified Functional Testing, tenham recursos integrados para fortalecer os testes contra as mudanças e os testadores de guias através da realização de atualizações dos scripts em tempo real, um aplicativo que estiver em um estado de fluxo pode necessitar de muitas mudanças nos scripts de automação para tornar a automação eficiente em termos de custo.
- Aplicativos móveis – Embora quase sempre testados com métodos manuais hoje em dia, é útil automatizar o teste das permutações de dispositivos, sistemas operacionais, cenários de utilização e condições de rede necessárias para os aplicativos móveis.

Figura 3

○ HP Sprinter fornece um registro automático de todas as ações do usuário em formato de texto e com uma linha de tempo detalhada.



○ HP Unified Functional Testing (HP UFT) suporta todos os principais aplicativos e ambientes de softwares para os testes automatizados. Em parceria com empresas de ponta e inovadoras, a HP está estendendo seus recursos de testes funcionais líderes do setor para os aplicativos móveis e infraestrutura em nuvem em todos os sistemas operacionais e plataformas comumente usados. O HP UFT inclui o conhecido software de automação HP QuickTest Professional, junto com o software HP Service Test e HP Service Test Management para testes de serviços avançados e recursos de gerenciamento.

Para os aplicativos de Web 2.0, o HP UFT oferece suporte para uma gama crescente de kits de ferramentas que habilitam as tecnologias Web 2.0, incluindo o Microsoft Silverlight e os kits de ferramentas Ajax mais comumente usados—Google Web Toolkit (GWT), Yahoo! User Interface (YUI), ASP.NET AJAX e o Dojo Toolkit. O Web Add-in Extensibility permite que você desenvolva suporte para testar controles de web padrão e de terceiros que não são suportados inicialmente pelo Web Add-in. E o Extensibility Accelerator é um ambiente de desenvolvimento integrado (IDE) que facilita e acelera o projeto, desenvolvimento e implantação do suporte para controles de web padrão e de terceiros que não são suportados de início.

3. Modernize os testes manuais: Enquanto as organizações estão tentando cada vez mais tirar proveito dos testes automatizados, os testes manuais ainda representam 70% dos testes funcionais. No entanto, os testadores podem adotar métodos de testes manuais modernos para simplificar e minimizar suas tarefas mais tediosas e sujeitas a erros. Em vez de inserir manualmente dados

de planilhas, tomar nota durante execuções de testes e introduzir os resultados no conjunto de testes, além de executar os testes continuamente para ambientes e configurações diferentes, os testadores devem procurar uma solução que ajude a simplificar essas etapas.

○ HP Sprinter, a nova solução de testes manuais da HP, se concentra em simplificar e acelerar os testes manuais, além de aumentar a colaboração da equipe. O HP Sprinter registra automaticamente os casos de testes diretamente a partir do HP Quality Center e os apresenta na tela. Para minimizar os processos suscetíveis a erros, o HP Sprinter insere automaticamente os dados a partir de uma planilha externa para o aplicativo sob teste.

○ HP Sprinter automaticamente registra os casos de testes do software HP Quality Center ou HP Application Lifecycle Management (HP ALM) e os apresenta na tela de "aplicativo sob teste". A partir daqui, o usuário pode navegar para visualizar as etapas do teste, abrir uma falha, adicionar comentários e registrar resultados reais.

○ HP Sprinter automaticamente registra todas as ações do usuário de acordo com três formatos: uma descrição textual de todas as ações do usuário (ver Figura 3), uma captura de tela de todas as ações do usuário e uma gravação em vídeo da execução completa do teste. Os testadores podem vincular essa descrição detalhada a uma nova falha com o simples clique de um botão. Isso melhora drasticamente a comunicação das falhas para os desenvolvedores e promove uma resolução rápida.

Para os aplicativos que são em primeiro lugar testados manualmente, incluindo os aplicativos móveis, o recurso de "teste de espelhamento" do software HP Sprinter ajuda a clonar um teste diversas vezes para diferentes ambientes e configurações. Isso permite que o testador execute um teste apenas uma vez, enquanto o HP Sprinter o replica em até quatro máquinas ao mesmo tempo. Essa capacidade demonstrou aumentar a produtividade geral do teste manual em 50–70%.³

Com o HP Sprinter, o HP BPT e o HP UFT, os resultados dos testes e toda a documentação relevante são automaticamente salvos no HP ALM ou HP Quality Center, proporcionando aos testadores e aos outros envolvidos o rastreamento completo dos resultados dos testes, a fim de testar casos, requisitos e falhas em um único sistema de registro.

4. Serviços de teste e componentes não pertencentes à GUI:

Para os aplicativos compostos, em vez de contar apenas com testes da GUI para validar a lógica subjacente, recomendamos testar diretamente as camadas de serviço (também chamadas de camada não hierarquizada). Esses serviços estão normalmente disponíveis para teste antes da GUI estar pronta, o que possibilita à QA começar a realizar testes e automações antecipadamente no ciclo de vida do aplicativo. Isso gera grandes benefícios, esteja a equipe usando ou não os processos Agile. Fica mais simples localizar o local da falha ao lidar com componentes isolados e mais fácil corrigir os bugs quando eles são encontrados mais perto do momento da inclusão.

Cada vez mais a lógica de negócio de um aplicativo reside na camada não hierarquizada, que é frequentemente aproveitada por vários aplicativos. Uma falha que passa despercebida nesses serviços pode ter ramificações difundidas e imprevisíveis. Testar esses serviços independentemente reduz riscos e fortalece os blocos de construção dos seus aplicativos compostos.

À medida que mais componentes são disponibilizados, recomendamos testar aplicativos compostos de várias camadas ao automatizar uma transação completa através de componentes pertencentes ou não à GUI. O HP UFT possibilita que as equipes de QA testem a camada da GUI, a camada não hierarquizada e o processo completo. Isso ajuda tanto a aumentar a qualidade dos aplicativos compostos ao testar mais funcionalidades em várias camadas de aplicativos, quanto a encontrar falhas que, de outro modo, passariam despercebidas ao executar testes apenas através da GUI.

Para os aplicativos em nuvem, é necessário testar os componentes que podem residir na nuvem ou nas instalações da empresa. O HP SOA Systinet pode ser usado como um repositório de metadados para entender as dependências dos aplicativos e serviços reais e replicados, incluindo os serviços simulados. Utilizar a combinação do software HP SOA Systinet e HP UFT permite às equipes de QA testar os processos completos dos aplicativos em nuvem.

Além disso, podem ocorrer atrasos caso um componente não esteja disponível para teste em um aplicativo composto ou em um cenário de teste de processos de negócios que abrange várias etapas. Isso pode acontecer se um componente compartilhado for um aplicativo ou um serviço crítico aos negócios que já esteja em produção ou for um serviço de nuvem cujo teste adicional gere muito custo. A virtualização de serviços é uma abordagem que pode ser usada para evitar o atraso das iniciativas de teste. Com a virtualização de serviços, o componente do serviço ou do aplicativo no back-end do teste é emulado; além disso, é criado um serviço virtual que imita o comportamento e a resposta do serviço verdadeiro. O serviço virtual pode ser usado repetidamente para a realização de testes sem gerar taxas de serviços em nuvem ou atrapalhar o ambiente de produção.

A jornada para a equipe de testes funcionais

Para que as equipes de QA atendam às necessidades emergentes do mercado, incluindo a adoção de computação em nuvem, Web 2.0, aplicativos móveis e compostos, deve haver uma transformação no modo como os testes funcionais são feitos. As equipes de testes funcionais bem-sucedidas iniciam o processo de teste antecipadamente no ciclo de vida do aplicativo. Mesmo antes de um aplicativo estar pronto para ser testado, elas se voltam holisticamente ao problema do teste e dedicam algum tempo para projetar sua estratégia. A estratégia inclui o uso de uma estrutura e a criação de componentes de testes reutilizáveis. Elas se envolvem com os negócios e incorporam os testes antecipadamente no ciclo de vida, realizando testes exploratórios através do uso de métodos modernos de testes manuais, incluindo a geração automática de documentação e relatórios. Além disso, essas equipes começam a testar a camada não hierarquizada antes mesmo de uma GUI ficar disponível. As equipes de QA com visão de futuro procuram oportunidades de automatizar os testes quando e onde for apropriado. No que se refere às áreas onde os testes continuam a ser realizados manualmente, essas equipes aceleram o processo através da automação de tarefas tediosas e repetitivas, a fim de reduzir o tempo necessário para concluir o teste. E, por fim, elas validam completamente os aplicativos compostos ao testar a camada da GUI, a camada não hierarquizada e o processo completo.

A HP oferece soluções abrangentes e integradas de testes funcionais para auxiliar em cada etapa do caminho.

Para saber mais sobre as soluções de testes funcionais da HP, faça o download das versões de teste gratuitas dessas soluções HP:

HP Unified Functional Testing

HP Sprinter

Notas

¹ Base: 206 tomadores de decisão de TI com insights a respeito das atividades de modernização orçadas. Fonte: Clearing Your Path to Modern Applications and Business Agility (Abrindo o caminho rumo aos aplicativos modernos e a agilidade nos negócios), Forrester Research, 6 de abril de 2010. A Forrester Consulting thought leadership paper commissioned by HP (Um documento sobre liderança em ideias da Forrester Consulting solicitado pela HP).

² Forrester, 2011 Mobile Trends, Thomas Husson and Julie A. Ask, 24 de janeiro de 2011

³ Os depoimentos dos clientes estão disponíveis em techvalidate.com/portals/hp-sprinter

Conecte-se

hp.com/go/getconnected

Veja as tendências de tecnologia, os alertas e as soluções da HP



Compartilhar com colegas

© Copyright 2011-2012 Hewlett-Packard Development Company, L.P. As informações contidas neste documento estão sujeitas a alterações sem aviso. As únicas garantias para produtos e serviços HP são as estabelecidas nas declarações de garantia expressa que acompanham tais produtos e serviços. Nada aqui contido deve ser interpretado como constituindo uma garantia adicional. A HP não se responsabiliza por omissões, erros técnicos ou erros editoriais contidos neste documento.

Microsoft e Silverlight são marcas comerciais registradas nos EUA.
Google é uma marca comercial da Google Inc.

4AA3-5237PTL, criado em junho de 2011; atualizado em abril de 2012, Rev. 1

