COMPUTERWORLD

EXECUTIVE BRIEFING

GUIA EXECUTIVO PARA DECISÕES ESTRATÉGICAS

AUTOMAÇÃO DE TESTES DE SOFTWARE

A automação de testes de software nasce como uma necessidade das empresas desenvolvedoras de aumentar a qualidade de suas soluções, reduzindo problemas comuns de desempenho, de falhas e de entraves com aumento da escalabilidade, além de trabalhar para reduzir ao máximo as brechas de segurança. Neste Executive Briefing preparado pelos editores do COMPUTERWORLD, o profissional vai entender a importância de se adotar uma política de testes e, mais do que isso, perceber como a aposta em ferramentas automatizadas garante melhores resultados.

CAPÍTULO 1

Histórico dos Testes

É possível justificar o nascimento da cultura de que testes não são necessários para o desenvolvimento de software pela a falta de recursos financeiros que acompanhou as primeiras companhias do setor. Muitas companhias, que nasceram de projetos mirabolantes de jovens recém-saídos da faculdade, tinham muita dificuldade para dispensar seus poucos profissionais que estavam codificando pesadamente para analisar os programas em busca de erros.

Contudo, com o amadurecimento do mercado, a situação mudou sensivelmente. Inúmeras foram as campanhas por mais qualidade de software, com diversas auditorias e consultorias crescendo cuidando exclusivamente desse tema. Na prática, os clientes estavam insatisfeitos com as diversas falhas no código do software – que vão de problemas em desempenho, passando pelos resultados gerados pela solução estarem errados, entraves com a solução recebendo mais usuários do que esperado, além das infinitas brechas de segurança. Isso demandou uma completa modificação na postura dos desenvolvedores de software e a busca por testes como uma maneira de aumentar a qualidade na solução final.

O problema é que ter pessoas testando centenas de milhares linhas de códigos, assim como no processo da codificação, existe uma enorme possibilidade de acontecerem erros. No final de uma extensa análise humana, é difícil garantir se o software está melhor e os erros que deveriam ser corrigidos foram de fato corrigidos ou se a suposta correção piorou o software com as intervenções do programador-corretor.

A resposta encontrada, neste momento, foi apostar em computadores para realizar os testes. Os parâmetros e a profundidade do exame são definidos pelos humanos, mas o trabalho "braçal" é feito pela máquina. Com isso, os riscos de novos erros são reduzidos substancialmente e os programadores têm tempo e possibilidade de pensar novas maneiras para melhoras os softwares, em vez de ficarem presos em um trabalho que muitos viam como retrabalho. A automação de testes de software que acabou tornando-se, também, um nicho muito interessante para conglomerados de tecnologia que produzem essas ferramentas.

ESCOPO DOS TESTES DE SOFTWARE

Os testes automáticos devem abordar funcionalidade, escalabilidade e desempenho, devendo ser realizados em três momentos: durante a criação; durante seu desenvolvimento e após a venda.

Tipos de ferramentas comuns de testes durante o processo de criação e desenvolvimento de um software

- Ferramentas de teste de componentes ou unidades (focadas em desempenho)
- Ferramentas de teste funcional (focadas em qualidade)
- Ferramentas para teste de stress (focada em desempenho e escalabilidade)

Tipos de ferramentas comuns de testes depois que o software foi lançado:

- Ferramentas de monitoramento e de gestão de desempenho
- Ferramentas de gestão de testes (auditoria dos outros testes realizados)

Mas como adicionar mais uma etapa detalhada e criteriosa, com diversos tipos de testes sendo realizados dentro de uma agenda apertada que tem no time-to-market o grande chamariz?

Ao serem realizados quando não havia produção propriamente dita, sendo popular a realização de rotinas de testes durante a noite, os testes acabam encontrando um espaço na concorrida agenda de produção de software, tornando-se mais uma fase necessária – sem gerar problemas com tempo ou entraves para o lançamento da solução.

Uma boa metáfora para comparar os testes em software é a indústria naval. Imagine um exemplo de uma lancha. É preciso que o futuro barco tenha todos os seus insumos testados para checar a capacidade de sobreviver durante muito tempo com o contato continuado com a água, seja de mares e oceanos; ou de rios e lagos. Logo depois de produzido, é preciso garantir que ele não afunde, realizando testes preliminares e verificando a correção de todos os cálculos que basearam a sua construção. Após a venda da lancha, o orgulhoso dono precisa também fazer as suas verificações recorrentes – uma rotina de testes – para garantir que a lancha não vai afundar quando tiver levando toda a sua família em um passeio.

Em resumo, os testes representam uma parte essencial da lancha antes da sua fabricação, durante a sua fabricação e mesmo depois de vendida. A relação com software é a mesma. Com a vantagem que os programas de computador podem ter boa parte de seus testes sendo feita de maneira automática, com menor interação de pessoas – e conseqüência menor chance de erro. Assim, com os computadores realizando testes repetitivos, é possível garantir com mais assertividade que a aplicação está fazendo apropriadamente o que ela foi criada para fazer.

Testes para garantir também segurança

Para garantir a segurança dos ambientes de informação, os muitos profissionais imaginaram que a ação necessária seria batalhar contra as pragas como vírus, worms e cavalos de tróia. Conforme o tempo foi passando, e os sistemas operacionais evoluindo, o que se viu é uma nova realidade que não tinha sido contemplada anteriormente – as brechas no código dos softwares.

Além dos testes de desempenho e integridade, outros testes que são imprescindíveis são os de segurança. Maior preocupação com segurança no processo de desenvolvimento garante um software com menos buracos – menor quantidade de falhas que podem ser exploradas por criminosos digitais interessados em dados críticos ou informações confidenciais.

Estimativas apontam que 6 a cada 10 softwares comerciais na internet estão rodando com falhas de segurança. Ao serem instalados nas máquinas, seja no computador de um consumidor final ou na estação de trabalho de uma grande corporação, esses programas representam uma porta de entrada para os criminosos digitais, porta de tão fácil acesso que pode destruir toda a estrutura de segurança criada. É fácil imaginar que uma política de testes automatizados, com análise das fragilidades de segurança, geraria soluções mais confiáveis.

Quando se analisa as aplicações web, esse problema é ainda maior. A OWASP (Projeto aberto de segurança em aplicações web, da sigla em inglês) definiu as 10 maiores vulnerabilidades encontradas nesse tipo de aplicação no ano de 2007. E todas são bem preocupantes.

A situação no Brasil é preocupante. Dados do Cert.br (Centro de Estudos, Resposta e Tratamento a Incidentes) apontam que em 2007 foram reportados mais de 160 mil ataques, divididos em worms (com 48%), fraudes (28%), scan de portas vulneráveis nas redes (21%), ataque de negação de serviço (1%) e ataque contra servidores web (1%). Boa parte desses ataques perpetrados durante o ano se apóiam nas vulnerabilidades par serem bem sucedidos. Uma política de testes em segurança poderia reduzir esses números sensivelmente.

Testes de segurança em ambientes virtuais

Uma das vantagens da virtualização é criar ambientes virtuais para fazer testes de segurança com simulação de situações drásticas sem riscos para a rotina produtiva. Essa capacidade acaba com a resistência das empresas em apontar vulnerabilidades no código de suas aplicações que estão correndo – principalmente por temerem que isso vai derrubar os programas que rodam baseados nestas aplicações ou corromper os dados que estão sendo manipulados.

Algumas empresas estão apostando em ofertas baseadas em virtualização para realizar testes de software sem riscos para a aplicação, diminuindo o impacto da análise de vulnerabilidade por usar cópias exatas da aplicação em ambiente virtual, não usando os dados ou a operação real.

A possibilidade de um ambiente virtual para testes sem impactos para a rotina coorporativa tam-

AS 10 MAIORES VULNERABILIDADES EM APLICAÇÕES WEB

1. Cross Site Scripting XSS (permite ao criminoso executar scripts no navegador da vítima)
2. Falhas de Injeção (permite manipular informações, o mais famoso é o SQL Injection)
3. Execução Maliciosa de Arquivos
4. Referência Insegura Direta a Objetos (permite o acesso a objetos confidenciais na base de dados)
5. Cross Site Request Forgery CSRF (força o navegador da vítima a executar ação maliciosa)
6. Vazamento de Informações e Tratamento de Erros Inapropriados
7. Falha de Autenticação e Gerenciamento de Sessão
8. Armazenamento Criptográfico Inseguro
9. Comunicações inseguras
10. Falha de Restrição de Acesso à URL

bém facilita a implementação da tão sonhada "política de testes contínuos". A política promete que os programadores estarão à frente das falhas e vulnerabilidades antes que elas sejam exploradas.

Especialistas de mercado apontam que a tendência, lançada pela companhia Cenzic, faz muito sentido e que tudo indica que outras empresas vão procurar lançar solução semelhante de testes de segurança em ambiente virtual. "É uma estratégia inovadora e as empresas rivais vão ter de correr atrás do setor", definiu Nick Selby, analista do 451 Group. O analista completa: "Esta tecnologia pode ser uma maneira criativa de quebrar a barreira que existe entre teste de aplicações no time de TI e o de segurança. Com isso, o pessoal de segurança não precisa mais de autorização de TI para rodar os testes, sem preocupação de afetar os negócios".

Além do teste: o desafio da Usabilidade

Várias organizações não compreendem completamente o porquê o software que foi desenvolvido por ela não tem o resultado esperado, ou o porquê os usuários cometem erros inesperados. De acordo com David Crow, conselheiro de usabilidade de empresas como a Microsoft Canada e Jay Goldman, presidente da companhia Radiant Core, o risco de não acordar a necessidade de realização de testes de usabilidade é fracassar na solução final.

Esse é outro ponto importante na criação de software. Além de testar automaticamente e repetidas vezes a aplicação para garantir que ela vai funcionar, é preciso entender como os usuários vão usar o software, garantindo que ele traga a melhor experiência possível para o usuário. Resources: The Usability Professional Association (http://upassoc.org), Sigchi.org and OpenUsability.org. Also, A Pattern Language, by Christopher Alexander.

PASSO 1: Admita que existe um problema. "É impossível criar usabilidade para si mesmo", diz Goldman. Eles defendem a utilização de personas, personagens ficcionais que são criados para representar diferentes tipos de usuários de diversas localidades que podem usar o site ou o produto. Táticas de usabilidade de guerrilha e entrevistas informais com usuários e equipe técnica podem ajudar. "Conheça seu usuário", resume Crow.

PASSO 2: Acredite na diversidade. Crow mostrou slides em três eventos com públicos diferentes. Inevitavelmente, as respostas foram diferentes sobre preferência. "Esse tipo de comportamento é visto no desenvolvimento de aplicações o tempo todo. Você precisa descobrir para qual público você vai criar a sua aplicação", afirma Crow.

PASSO 3: Defina um critério para reconhecer bom design. Goldman cita Steve Jobs para explicar. "Design não é apenas como parece e como usuário sente o produto, mas também como funciona", repete.

PASSO 4: Crie, sem medo, um inventário de seus fracassos em experiência do usuário. As experiências passadas são importantes para garantir soluções mais adequadas no futuro.

PASSO 5: Admita para outro pessoa que existe um problema. Mais do que conseguir feedback, falar como um igual para os usuários pode ajudar a explicar por que a aplicação não funciona. "É preciso muita coragem para dar esse passo [conversar com usuários regularmente], mas depois de fazer isso publicamente, já está no caminho certo", defende Goldman.

PASSO 6: Esteja pronto para tomar uma atitude e retirar os defeitos apontados pelos usuários. Crow usou o Office 2007 da Microsoft como um objeto de estudo. Conforme a empresa passou a adicionar novas funcionalidades no Word, por exemplo, passou a usar 31 barras de ferramentas minimizadas, em comparação com 12 do Word 2003. "Das 10 maiores funcionalidades usadas, cinco delas já estavam no Office por mais de uma versão", define Crow. A última versão da suíte, ao contrário, usa o chamado "ribbon" com uma combinação básica de barras de ferramentas para auxiliar a encontrar o que se procura.

PASSO 7: Peça auxílio. "[o auxílio] Está disponível", garante Crow. Mesmo grandes corporações com supostamente vastos recursos, como a fundação de código aberto Mozilla, optou por colocar rascunhos da nova versão do navegador Firefox em seu site. Esse tipo de comportamento pode assustar as empresas preocupadas em rivais tendo acesso aos seus planos, mas Goldman e Crow afirmam que os resultados podem ser recompensadores.

PASSO 8: Faça uma lista de todos os usuários atingidos por falta de usabilidade da sua aplicação e tente fazer a vida deles melhor. Goldman desenha uma escala que vai de funcional para honesta para usável para conveniente para prazeroso para significativo. "A maior parte das aplicações falha pouco antes de atingir o status conveniente. É bem difícil de saltar", afirmou Crow.

PASSO 9: Faça as correções imediatamente. Infelizmente, ouvir os usuários pode trazer problemas algumas vezes. Se você não conseguir melhorar o que foi apontado, prepare-se para o pior. "Se você brigar com os usuários, eles nunca mais voltam e contam tudo para seus amigos", afirma Goldman.

PASSO 10: Continue a fazer um inventário. Testes de usabilidade não é um evento único, mas um ciclo que envolve observação, análise e design.

PASSO 11: Acredite: sem o usuário, nada importa. O foco principal da usabilidade é garantir que o usuário fiel sinta-se satisfeito com a aplicação que usa; e que o novo usuário não se sinta perdido quando for experimentar o software.

PASSO 12: Compartilhe informações. A comunidade de software possui inúmeros locais como o a livraria de interface para usuário do Yahoo ou o Tango.freedesktop.org para compartilhar as lições aprendidas. Conversar é crucial para avançar nessa indústria.

Definição da própria livraria de testes

Em seu blog dedicado a empresas start-ups, o empreendedor Dharmesh Shah (fundador da Pyramid Digital Solutions) publicou um post interessante sobre a visão de negócios quando o tema é automação de testes de software. O especialista propõe a construção individual dos testes conforme a aplicação em questão, fugindo do modelo de grandes suítes com testes automatizados comercializadas pelos grandes fornecedores.

Em cinco passos, o programador dá dicas e aponta mudanças na maneira de pensar e codificar software para melhorar o resultado final. Confira alguns conselhos do especialista e defina a sua própria opinião sobre o tema.

1- CONSTRUA UM SOFTWARE MELHOR.

Ao construir uma livraria de testes automatizados, defende o programador, o resultado é invariavelmente a venda de um software melhor. Dessa maneira, o fornecedor tem – através da livraria de testes automatizados – garantia de que seu software vai sobreviver sem problemas a uma quantidade pré-definida de tarefas dentro de maneiras pré-estabelecidas no ambiente de desenvolvimento.

2- TESTE CONTINUAMENTE

Depois do investimento inicial para a criação da livraria de testes automatizados, quase não há custo para que eles rodem. Desta forma, acrescente, não há razão para que eles não sejam repetidos com freqüência. “Na minha primeira start-up, nós chegamos a ter mais de 20 mil scripts de testes que rodavam por diversas horas”, escreveu.

3- FICA MAIS BARATO CONSERTAR FALHAS EM SOFTWARE COM TESTES.

A maioria dos softwares tem bugs, garante o especialista. “De uma perspectiva de negócios, as questões são: quais bugs você conhece, quando você os encontra e quanto custa para consertá-los?”, escreve. Ele garante que o quanto custa consertar está intimamente relacionado com o quando as falhas são descobertas.

4- LIBERDADE PARA MUDAR:

De acordo com o especialista, o crescimento contínuo dos sistemas de software torna cada vez mais difícil modificar algum pedaço de código sem consequentemente quebrar a aplicação. Ele defende que certas aplicações, especialmente aquelas que possuem uma base de linhas de códigos estabelecidas há muito tempo, tem certas encruzilhadas que todos os participantes dentro do time de desenvolvimento vão fazer de tudo o que podem para ficar longe desses pedaços com medo de quebrar toda a aplicação. “Com uma grande bateria de testes automatizados, você tem uma rede de segurança que permite aos programadores fazerem coisas legais como mudar funcionalidades, refazer o código, ou pensar novas funções sem medo de comprometer o programa”, defende.

5- FAÇA SEUS CLIENTES MAIS FELIZES

O empreendedor aconselha reuniões regulares com os principais clientes, além de encontros anuais em que vários clientes encontrem-se e troquem experiências, como maneira de estreitar o relacionamento. Ao comentar a sua própria experiência, ele afirma que uma métrica que dava maior conforto para os clientes está no quão longa era a rotina de testes automáticos no software. “Esse conforto se traduzia em uma probabilidade maior de que os clientes instalariam as novas versões do software assim que ela ficasse disponível. Como nosso foco era em grandes corporações, se conseguíssemos cerca de 20% dos clientes migrando para a nova versão, estaríamos em vantagem”, escreveu. Além de garantir a maior satisfação dos clientes, ressalta o especialista, a empresa que adota a automação de testes de software acaba reduzindo também seus custos de suporte técnico, também aumentando a sua taxa de retenção.