VERIFICAÇÃO DE SEGURANÇA E DESEMPENHO

FAÇA DAR CERTO

Promova melhorias na verificação de segurança e desempenho focando nos oito principais fatores que influenciam as práticas relacionadas.

O QUE CONTÉM ESSE RELATÓRIO?

Esse relatório reúne os resultados de um estudo na área de Verificação de Segurança e Desempenho, conduzido em parceria entre a COPPE/UFRJ no Brasil e o SINTEF Digital, na Noruega.

O estudo observou organizações de desenvolvimento de software no Brasil para identificar e caracterizar as práticas de Verificação de Segurança e Desempenho utilizadas por essas organizações. Foi possível identificar oito fatores de sucesso relacionados à verificação de segurança e desempenho.

Assim, este relatório apresenta os oito principais fatores de sucesso relacionados à verificação de segurança e desempenho. Além disso, indica ações para promover esses fatores.

Introdução

A popularização e o uso em massa de sistemas de software trazem benefícios para a vida moderna. Entretanto, a disponibilidade desses sistemas também traz consigo preocupações, principalmente com algumas dimensões de qualidade como segurança e desempenho. Organizações de desenvolvimento de software geralmente incluem atividades de garantia de qualidade ao longo do ciclo de vida do software com a finalidade de avaliar sua qualidade e prevenir falhas após sua entrega. A verificação de software, incluindo testes e revisões, engloba ações que possuem o objetivo de analisar o software procurando por defeitos.

Segurança é relevante! Atualmente os sistemas de software são responsáveis pela manipulação e armazenamento de informação crítica e sensível. Por exemplo, sistemas de software são responsáveis por manipular dados pessoais, dados estratégicos de organizações e controlar transações financeiras. Como resultado, existe um crescente interesse em ter acesso a essas informações para obter vantagens indevidas.

Desempenho é relevante devido às limitações dos recursos computacionais! Um tempo de resposta demasiadamente longo pode fazer com que usuários migrem para o software concorrente, um atraso no envio de transações financeiras pode representar perda de capital e o consumo excessivo de energia pode inviabilizar a execução de um sistema em um hardware alimentado por bateria ou aumentar os custos com energia de *data-centers*.

As atividades de verificação de segurança e desempenho têm por finalidade exercitar o software em busca de defeitos relacionados a essas perspectivas de qualidade. Práticas individuais de verificação, ou a combinação de práticas, trazem benefícios. Porém, impõem múltiplos desafios à verificação de segurança e desempenho.

Apesar da existência de técnicas de verificação de segurança e desempenho, sistemas de software continuam apresentando falhas relacionadas a essas propriedades. Falhas de desempenho são as mais frequentes em alguns domínios (ex., telecomunicações). Relatórios sobre ataques a sistemas são frequentemente divulgados. Algumas razões são (1) a ineficiência das práticas de verificação de segurança e desempenho, (2) as organizações de desenvolvimento de software não adotarem práticas adequadas e/ou (3) a falta de divulgação de práticas cientificamente comprovadas, revelando uma desconexão entre os resultados acadêmicos e a demanda da indústria de software. Adicionalmente, scripts automatizados para ataques, a abundância de informação sobre procedimentos de invasão e a disponibilidade dos sistemas de software através da internet facilitam as tentativas de ataques.

Esse relatório apresenta os resultados de um estudo executado no contexto de organizações de desenvolvimento de software Brasileiras com o objetivo de compreender as práticas de verificação de segurança e desempenho utilizadas em seus projetos. Assim, é possível identificar necessidades reais das organizações, permitindo aos pesquisadores direcionar suas pesquisas para suprir essas necessidades, fornecendo respostas úteis às organizações.

ESTE RELATÓRIO DESTACA **OITO FATORES ESSENCIAIS** PARA UMA VERIFICAÇÃO DE SEGURANÇA E DESEMPENHO BEM SUCEDIDA.

1 PROMOVER A CONSCIÊNCIA DE SEGURANÇA E DESEMPENHO NA ORGANIZAÇÃO

\+ 2 MANTER UMA EQUIPE MULTIDISCIPLINAR

\+ 3 PRODUZIR REQUISITOS CLAROS E NÃO AMBÍGUOS

\+ 4 POSSUIR FERRAMENTAS DE APOIO ADEQUADAS

\+ 5 CONFIGURAR UM AMBIENTE ADEQUADO

\+ 6 UTILIZAR UMA METODOLOGIA SISTEMÁTICA

\+ 7 PLANEJAR AS ATIVIDADES DE VERIFICAÇÃO DE SEGURANÇA E DESEMPENHO

\+ 8 ESTIMULAR PRÁTICAS DE REUTILIZAÇÃO

=

Verificação de Segurança e Desempenho Bem-Sucedida!

1 CONSCIÊNCIA DE SEGURANÇA E DESEMPENHO NA ORGANIZAÇÃO

A verificação de segurança e desempenho não deve ser responsabilidade de um único departamento. A percepção global da importância de segurança e desempenho dos sistemas de softwares desenvolvidos pela organização afeta as atividades de verificação. Assim, as atividades de segurança e desempenho dependem do apoio de todos os *stakeholders.*

Ações

- ✓ Promover treinamento
- ✓ Informar ao cliente o estado real da segurança e desempenho do sistema
- ✓ Manter programadores bem informados sobre segurança e desempenho

A alta gerência deve apoiar financeiramente as atividades de verificação de segurança e desempenho. Por exemplo, apoiando a aquisição de ferramentas e incluindo os custos de segurança e desempenho no planejamento do projeto

A equipe de desenvolvimento deve considerar a verificação de segurança e desempenho como um benefício, entendendo que ao reportar uma falha a equipe de verificação não está agindo contra o projeto. Além disso, a equipe de desenvolvimento tem conhecimento aprofundado no domínio e na arquitetura do software desenvolvido. Assim, eles podem auxiliar a tomada de decisões sobre o que deve ser verificado, a priorização e a identificação das dependências dos cenários de verificação.

O Cliente deve compreender que a verificação de segurança e desempenho não é um desperdício de recursos. Eles devem ser informados sobre a real situação da segurança e desempenho do software que estão operando. Além disso, os clientes devem entender que as atividades de verificação de segurança e desempenho não garantem a construção de um sistema totalmente seguro ou um sistema totalmente sem problemas de desempenho.

2 EQUIPE MULTIDISCIPLINAR

A verificação de segurança e desempenho não deve ser desempenhada de forma isolada por uma única equipe. As atividades de verificação requerem interação entre equipes diferentes bem como diferentes habilidades.

Ações

- ✓ Estimular interação entre membros de diferentes equipes
- ✓ Formar uma equipe com diferentes habilidades
- ✓ Disseminar a visão de que a equipe de verificação não é inimiga, mas aliada

Especialistas em verificação de segurança e desempenho são responsáveis por fornecer conhecimento relacionado à segurança e desempenho como, por exemplo, políticas de segurança da informação, padrões de desenvolvimento seguro e com performance, certificação digital e criptografia. Adicionalmente, a equipe de verificação de segurança e desempenho deve possuir conhecimento técnico necessário para compreender as tecnologias empregadas no desenvolvimento de software (ex.: banco de dados, *web server* e linguagens de programação) e conhecimento técnico para utilização das ferramentas de verificação de segurança e desempenho.

Equipe de infraestrutura deve apoiar os especialistas durante as atividades de verificação de segurança e desempenho. Por exemplo, pode ser necessário liberar acesso para que um IP (*internet protocol*) específico acesse o servidor, realizar alguns ajustes no *kernel* do sistema operacional ou reiniciar o servidor após uma falha catastrófica causada pelos testes.

Equipe de banco de dados pode ser útil para realizar a restauração do banco de dados após uma bateria de testes.

Especialistas em legislação apoia o entendimento dos impactos legais causados pela falta de segurança e desempenho. Além disso, podem auxiliar no cálculo do risco legal.

3 REQUISITOS CLAROS E NÃO AMBÍGUOS

Os requisitos são o oráculo para a verificação de segurança e desempenho. Portanto, a falta de requisitos impede a equipe de avaliar se os resultados da verificação estão corretos. Além disso, requisitos imprecisos sobrecarregam outras equipes (ex.: analistas, arquitetos e desenvolvedores), pois a equipe de verificação precisa contatá-los frequentemente. Há um conjunto de questões trazidas pela falta de requisitos inadequados.

Ações

- ✓ Envolver a equipe de verificação na fase de requisitos
- ✓ Estimular a equipe de verificação a avaliar a testabilidade dos requisitos
- ✓ Utilizar técnicas para lidar com requisitos de segurança e desempenho

A **falta de um oráculo** faz com que a verificação seja executada para identificar o atual comportamento do software. Por exemplo, se não existe um requisito de desempenho, as atividades de verificação não são executadas para avaliar se o software atende a esse requisito, mas para identificar qual é a capacidade do software em termos de desempenho.

A falta de requisitos de segurança e desempenho pode ser arriscada, pois **a equipe de verificação determina os requisitos** por sua própria experiência, o que pode não refletir as expectativas dos clientes. Além disso, a equipe de verificação pode ter dificuldades em determinar os requisitos de segurança e desempenho.

A interação entre os colaboradores é essencialmente importante no desenvolvimento de software. Entretanto, a comunicação desnecessária é uma perda de recursos, implicando em perda de eficiência. Por que interromper o trabalho de um colaborador se é possível obter a mesma informação consultando um documento? Portanto, a falta de requisitos ou requisitos imprecisos causam **interação desnecessária entre *stakeholders***, sobrecarregando outras equipes (ex.: analistas, arquitetos e desenvolvedores), pois a equipe de verificação precisa contatá-los continuamente.

4 FERRAMENTAS DE APOIO ADEQUADAS

A utilização de ferramentas adequadas é essencial para as atividades de verificação de segurança e desempenho, pois tendem a diminuir o esforço de atividades manuais. Ferramentas livre são aconselháveis, pois o processo de aquisição é mais rápido por envolver apenas a equipe técnica. No caso da adoção de ferramenta proprietária, é necessário a permissão dos gerentes e o preço pode ser um fator de impedimento.

Ações

- ✓ Permitir que esquipe técnica sugira e adote ferramentas de apoio
- ✓ Apoiar a utilização de ferramentas livres
- ✓ Utilizar ferramentas compatíveis com o nível de conhecimento da equipe de verificação

É essencial **considerar a capacidade da equipe** ao escolher as ferramentas, pois a equipe deve ter capacidade técnica necessária para explorar as vantagens da ferramenta selecionada.

Deve-se atentar ao **número excessivo de falsos positivos** produzido pelas ferramentas. Nesse caso, os resultados da execução da ferramenta podem ser ignorados devido ao grande esforço necessário para analisar os relatórios de falhas gerados.

Os relatórios gerados pelas ferramentas não devem ser entregues diretamente aos clientes ou aos desenvolvedores. Os **relatórios devem ser previamente analisados e processados pela equipe de verificação**. Assim, um relatório consistente pode ser entregue aos interessados.

Nos relatórios, é essencial **evidenciar as atividades de verificação que não revelaram incidentes**. Essa prática é psicologicamente positiva para o cliente ou desenvolvedor, pois podem tomar conhecimento que o sistema desenvolvido opera corretamente em alguns cenários.

5 AMBIENTE ADEQUADO

Um ambiente adequado é essencial para a verificação de segurança e desempenho. Este ambiente deve englobar tanto a configuração da infraestrutura responsável pela operação (parâmetros dos servidores de aplicação e banco de dados) quanto a configuração do próprio sistema (dados armazenados no início da verificação).

Um **ambiente isolado** deve ser prioridade, pois outras atividades podem influenciar os resultados da verificação.

A **utilização do ambiente de homologação não é recomendada**, pois causa influência bidirecional: (1) testes de performance podem prejudicar a homologação do sistema pelos usuários porque a simulação de muitos usuários acessando um sistema causa sobrecarga no hardware; (2) se o sistema está em execução para a homologação, os testes de desempenho apresentam resultados aleatórios.

A **utilização do ambiente de produção não é recomendada**, pois é difícil prever com confiança o comportamento dos usuários reais, prejudicando os resultados da verificação.

Ações

- ✓ Utilizar tecnologias de virtualização para simular o ambiente de execução
- ✓ Utilizar tecnologias de virtualização para configurar agentes de testes
- ✓ Agendar as atividades de verificação caso não seja possível instanciar um ambiente específico de forma que a verificação nunca seja executada em paralelo com outras atividades
- ✓ Manter a equipe de verificação bem informado sobre as tecnologias utilizadas
- ✓ Executar cada caso de teste mais de uma vez em diferentes momentos para mitigar influências externas

A rede influencia principalmente os testes de desempenho. Se uma máquina utilizada para os testes de desempenho utiliza a rede comum da organização, as requisições e respostas podem ser retardadas devido à sobrecarga nos nós da rede (roteadores, *sweets*) que encaminham os dados para o servidor no qual o software está hospedado.

As tecnologias utilizadas para construir o sistema podem influenciar nos resultados das atividades de verificação. Por exemplo, a utilização da tecnologia de *cache* para recuperar informações do banco de dados podem levar à resultados imprecisos de tempo de resposta.

Outra questão relacionada ao ambiente de verificação é a **diferença entre a configuração de hardware utilizada para a verificação e a utilizada em produção**. Em alguns casos, o hardware utilizado no ambiente de produção é mais potente que o utilizado no ambiente de verificação, podendo gerar uma falsa impressão sobre o desempenho do sistema.

6 METODOLOGIA SISTEMÁTICA

Uma metodologia apoia a equipe de verificação a saber o que deve ser feito em cada fase do processo de verificação. Caso uma metodologia adequada esteja disponível, as práticas de verificação de segurança e desempenho podem ser desempenhadas de forma sistemática; portanto, tornando-se mais eficientes. Além disso, existem alguns atributos e elementos que uma boa metodologia deve contemplar.

Ações

- ✓ Utilizar uma metodologia apropriada e adaptá-la ao contexto da organização

A **metodologia deve ser adaptável às tecnologias** utilizadas no desenvolvimento do sistema de software. Por exemplo, é inútil executar análise de vulnerabilidade web em um sistema embarcado ou a verificação de banco de dados em sistema que não armazenam nenhuma informação.

A **metodologia deve permitir a adoção incremental das práticas propostas** de forma que as equipes possam se adaptar a essas novas práticas.

A **metodologia deve permitir evolução**, pois as necessidades de segurança e desempenho dos sistemas evoluem ao longo do tempo. Em relação à segurança, a evolução é obrigatória, pois novas técnicas de invasão são constantemente criadas.

A **identificação dos ativos e análise de risco** são elementos essenciais à uma metodologia. Além disso, uma metodologia deve deixar claro que a verificação de segurança e desempenho deve ocorrer após a correção dos defeitos revelados pela verificação funcional. Caso contrário, a verificação de segurança e desempenho pode identificar falhas relacionadas aos requisitos funcionais, contrariando sua real finalidade.

7 PLANEJAR AS ATIVIDADES DE VERIFICAÇÃO DE SEGURANÇA E DESEMPENHO

Planejamento é essencial para definir o que deve ser realizado, conhecer os recursos disponíveis (ex.: recursos humanos, tempo, ferramentas) e então estabelecer a forma como as atividades devem ser desempenhadas. Entretanto, geralmente a verificação de segurança e desempenho não é adequadamente planejada, levando à necessidade de redefinição na prioridade das atividades de verificação e, consequentemente, a redução de sua cobertura.

Ações

- ✓ Criar um plano para apoiar as atividades e antecipar risco, esforço e custos relacionados às atividades de verificação de segurança e desempenho

Geralmente, os gerentes têm a **percepção que verificação de segurança e desempenho requer esforços e custos adicionais**. Então eles negligenciam essas atividades, excluindo-as do planejamento de verificação.

Os stakeholders (gerentes e clientes) têm a **percepção que as atividades de verificação alteraram o prazo de entrega e o custo de produção dos sistemas de software**, não considerando os benefícios dessas atividades.

8 PRÁTICAS DE REUTILIZAÇÃO

A reutilização de conhecimento e artefatos, como os casos de teste funcionais, traz benefícios à verificação de segurança e desempenho. Além disso, a verificação funcional também se beneficia da prática de reutilização.

Ações

- ✓ Reutilização de casos de teste funcionais, pois representam cenários reais
- ✓ Reutilizar casos de testes de sistemas similares, adaptando parâmetros
- ✓ Reutilizar conhecimento adquirido na verificação de sistemas similares como base de definição de requisitos
- ✓ Conhecer defeitos comuns (ex.: vulnerabilidades) e utilizar testes previamente definidos para identificar as falhas causadas por esses defeitos

Reutilização de conhecimento de sistemas anteriores e dos casos de teste funcionais **aumenta a agilidade e reduzem os custos** da verificação de segurança e desempenho, pois é possível se apropriar de decisões passadas (ex.: escolha de técnicas e critérios) e os casos de testes não precisam ser construídos a partir do zero.

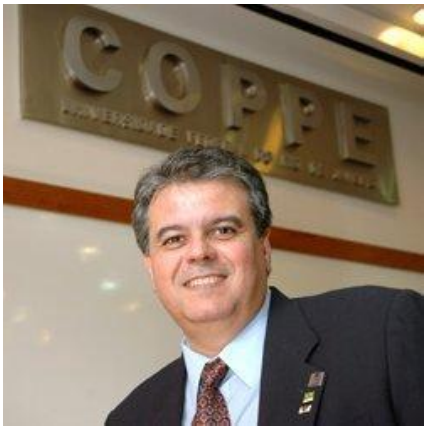
A reutilização dos casos de testes funcionais **aumenta a cobertura da verificação e melhora a taxa de detecção de falhas**, pois representam cenários reais de utilização do software.

Reutilizar casos de testes funcionais também beneficia a própria verificação funcional. Há **aumento na qualidade dos testes funcionais**, pois o esforço economizado com os testes não-funcionais pode ser utilizado para aprimorar os testes funcionais e há **aumento na divulgação dos testes funcionais** devido sua reforçada importância no processo de desenvolvimento.

Autores

Victor Vidigal Ribeiro é mestre e candidato ao doutorado em Sistemas de Engenharia e Computação pela Universidade Federal do Rio de Janeiro (COPPE/UFRJ). Seus interesses de pesquisa incluem a aplicação de métodos experimentais no estudo de verificação de sistemas de software (testes e revisões), principalmente com foco em requisitos não funcionais de sistemas contemporâneos. Detalhes disponíveis em http://www.cos.ufrj.br/~vidigal.

Daniela Soares Cruzes é pesquisadora sênior do SINTEF. Foi professora adjunta da Universidade Norueguesa de Ciência e Tecnologia (NTNU). Trabalhou como pesquisadora na Universidade de Maryland e no Fraunhofer Center for Experimental Software Engineering, em Maryland. Dra. Daniela Cruzes é gerente de projeto no projeto SoS-Agile (Ciência da Segurança para o Desenvolvimento de Software Ágil), financiado pelo Conselho de Pesquisa da Noruega. Seus interesses são desenvolvimento ágil de software, DevOps, segurança de software, engenharia de software global, processos de teste de software, métodos de pesquisa empírica, desenvolvimento de teorias e síntese de estudos de engenharia de software.



Guilherme Horta Travassos recebeu o título de Doutor em Engenharia de Sistemas e Computação pela COPPE/UFRJ, com pós-doutorado em Engenharia de Software Experimental na UMCP/USA. Atualmente é professor titular da COPPE/UFRJ e pesquisador do CNPq. Ele é o coordenador do Programa de Engenharia de Sistemas e Computação e lidera o Grupo de Engenharia de Software Experimental da COPPE/UFRJ. Membro do ISERN, SBC e ACM. Além disso, participa do conselho editorial da Elsevier - IST (editor associado), World Scientific-IJSEKE, SBC - JSERD e e-Informatica. Detalhes disponíveis em http://www.cos.ufrj.br/~ght.