Atividade - GQS

Exemplos de aplicações dos conteúdos de base que serão estudados na UC Gestão e Qualidade de Software – GQS.

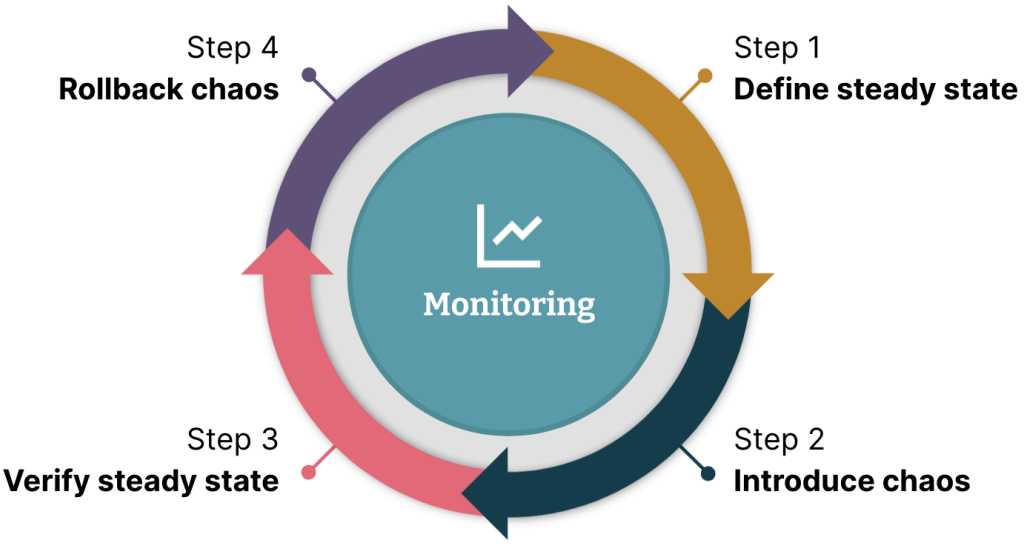
Eduardo Filipe Silva S. Santos - 82426451

- **GQS:**

  Mapa Conceitual:

Autor: Professor Robson Calvetti

1. Chaos Engineering (Teste de disponibilidade)



A engenharia do caos é uma metodologia onde se realizam experimentos para identificar falhas a fim compreender a causa raiz, ajudar a evitá-las e manter o sistema resiliente. Usada principalmente em ambientes de software distribuídos de larga escala por times de Site Reliability Engineering (Engenharia de Confiabilidade do Site), experimentos introduzindo e/ou emulando eventos de que ocorrem em produção, como disco cheio ou um

ataque DDoS, para garantir que mesmo sob tais circunstâncias o sistema permanecerá estável, em seu comportamento normal.

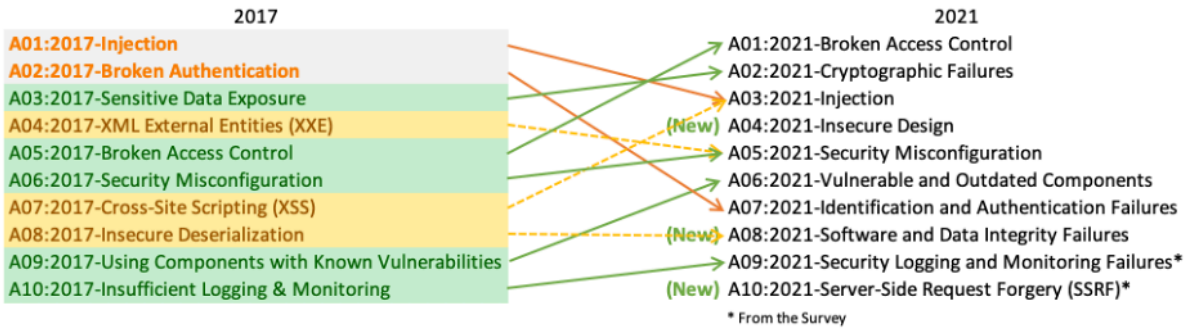
O chaos engineering é diferente dos testes de software pois se concentra em encontrar possíveis pontos de falha em ambiente de produção antes que eles causem problemas. Os testes, por outro lado, concentram-se em verificar se o sistema funciona conforme o esperado. Chaos engineering é proativo, enquanto o teste é reativo. Os experimentos podem tanto ser executados manualmente ou via automações, como o produto Gremlin de “Attack as a Service”.

O chaos engineering tem como objetivo garantir a qualidade da arquitetura e configuração da plataforma e infraestrutura das aplicações.

Fontes:
https://principlesofchaos.org/

https://www.resilience-engineering-association.org/resources/where-do-i-start/

2. OWASP Top 10 - SQL Injection (Testes de segurança)



A segurança de software é algo crucial para proteger informações sensíveis e garantir a integridade dos sistemas. Para conseguirmos atestar a segurança de um software , é necessário identificar e mitigar vulnerabilidades que possam ser exploradas por atacantes. Entre as principais referências globais para a segurança de aplicações está o OWASP Top 10, uma lista das vulnerabilidades mais críticas e recorrentes em sistemas web, que serve como um guia para desenvolvedores e equipes de segurança.

A OWASP Top 10 destaca vulnerabilidades amplamente conhecidas que, apesar disso, continuam a ser exploradas em aplicações modernas devido a falhas de implementação e configuração. Essas vulnerabilidades representam riscos significativos para a segurança da aplicação, podendo resultar em acesso não autorizado, roubo de dados, manipulação de informações e até na interrupção de serviços críticos.

Dentre as vulnerabilidades da OWASP Top 10, uma das mais perigosas e recorrentes são as vulnerabilidades de ‘Injection’, principalmente SQL Injection. Ela ocorre quando uma aplicação permite a inserção de comandos SQL maliciosos em consultas de banco de

dados, devido à falta de validação ou sanitização adequada de entradas fornecidas pelo usuário. Podendo permitir que atacantes obtenham dados sensíveis, modifiquem ou excluam dados, executem comandos no servidor e até mesmo escalar privilégios.

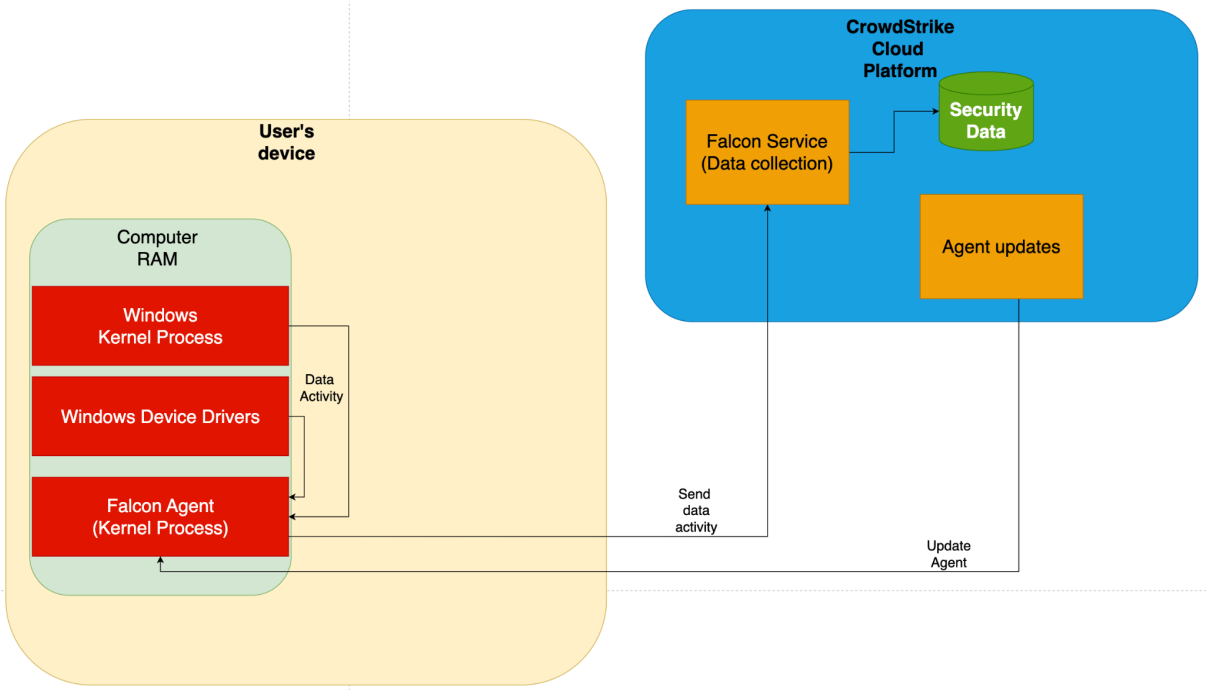
Serviços expostos a internet com vulnerabilidades de SQL Injection podem, ao serem atacados, causar consequências catastróficas, principalmente em ambientes corporativos onde o vazamento de dados ou a interrupção dos sistemas pode acarretar em prejuízos financeiros, danos à reputação da empresa e implicações legais.

3. Microsoft - Crowdstrike Crash (Falta de teste de softwares)

No dia 19 de Junho de 2024, uma atualização defeituosa do CrowdStrike Falcon causou o travamento de sistemas Windows, ocasionando o aparecimento da tela azul, afetando muitas organizações e serviços em todo o mundo e deixando off-line empresas inteiras e centenas de milhares de dispositivos.

A causa desse problema foi um componente defeituoso em uma atualização do CrowdStrike Falcon, uma solução desenvolvida pela empresa CrowdStrike para a proteção de endpoints contra ameaças de malwares, que provocou interrupções generalizadas nos sistemas (exceptions), travando os sistemas Windows com a “tela azul da morte” (BSOD).

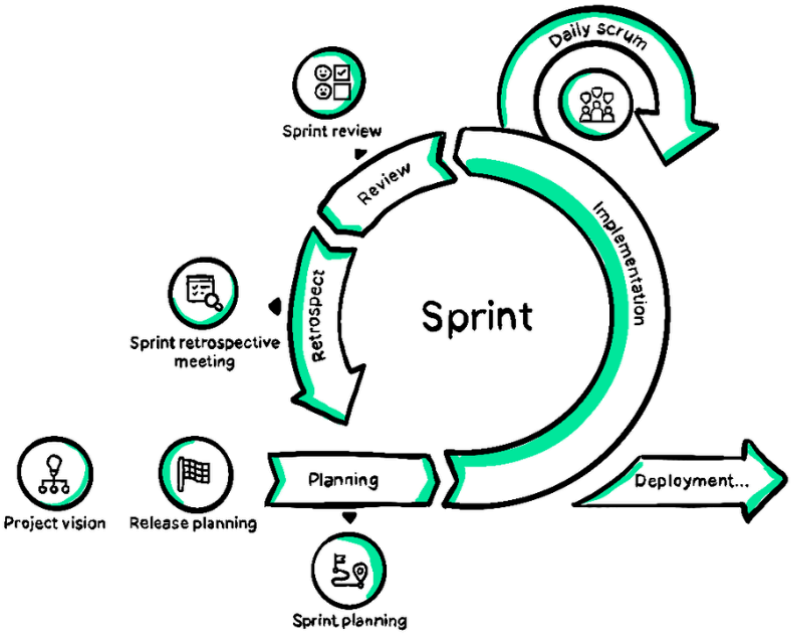
Somente dispositivos rodando o Microsoft Windows com o agente do Falcon foram afetados, devido o Falcon ser executado com privilégios elevados em modo kernel. Tal falha, com impacto global, poderia ter sido evitada com algumas medidas de teste e gestão da qualidade no deploy de novas versões do agente.



Fontes:
https://www.bleepingcomputer.com/news/security/crowdstrike-update-crashes-windows-systems-causes-outages-worldwide/

https://engineeringatscale.substack.com/p/blue-screens-to-blackouts-the-story

4. Uso de SCRUM no projeto Sentinel do FBI (Melhoria no processo de desenvolvimento)



Em 2003, o FBI decidiu digitalizar os arquivos de caso das suas investigações. Isso permitiria que os investigadores pudessem comparar rapidamente os casos e descobrir conexões entre eles. O projeto para automatização dos processos de comparação foi chamado de Sentinel.

Em março de 2006, o FBI iniciou o desenvolvimento do Sentinel, com um orçamento inicial de US$451 milhões para o desenvolvimento e implementação do Sentinel até dezembro de 2009. De acordo com o plano original do FBI, o Sentinel deveria ser desenvolvido em quatro fases. O FBI contratou a empresa Lockheed Martin para o projeto, que desenvolveria o projeto usando a metodologia tradicional de desenvolvimento de software, chamada Cascata.

Em agosto de 2010, o FBI gastou US$405 milhões do orçamento de US$451 milhões do Sentinel, entregando apenas algumas funcionalidades para apenas duas das quatro fases do projeto. Devido ao excesso de custo e ao cronograma apertado, o FBI cancelou o Sentinel em julho de 2010, deixando-o incompleto.

O FBI recrutou um novo diretor de tecnologia (CTO), que decidiu mudar a abordagem do projeto do Sentinel. Alegando que simplificaria os processos de tomada de decisão e permitiria ao FBI entregar o Sentinel dentro do orçamento.A equipe do Sentinel foi reduzida de 400 para 45 pessoas, 15 das quais eram programadoras. A nova abordagem tinha o objetivo de fornecer novas funcionalidades do Sentinel a cada 30 dias em vez de uma única grande entrega ao fim de cada ciclo.

Em novembro de 2011, após um ano de recomeço do projeto com o Scrum, todas as fases do Sentinel foram concluídas. O software foi implementado em um grupo piloto de escritórios do FBI. Os demais escritórios tiveram implementação até junho de 2012. O FBI concluiu o Sentinel por US$30 milhões em 12 meses, uma economia de mais de 90%.

Fontes:
https://www.scrumalliance.org/about-scrum

https://www.amazon.com.br/SCRUM-fazer-dobro-trabalho-metade/dp/8543107164/ref=as_li_ss_tl

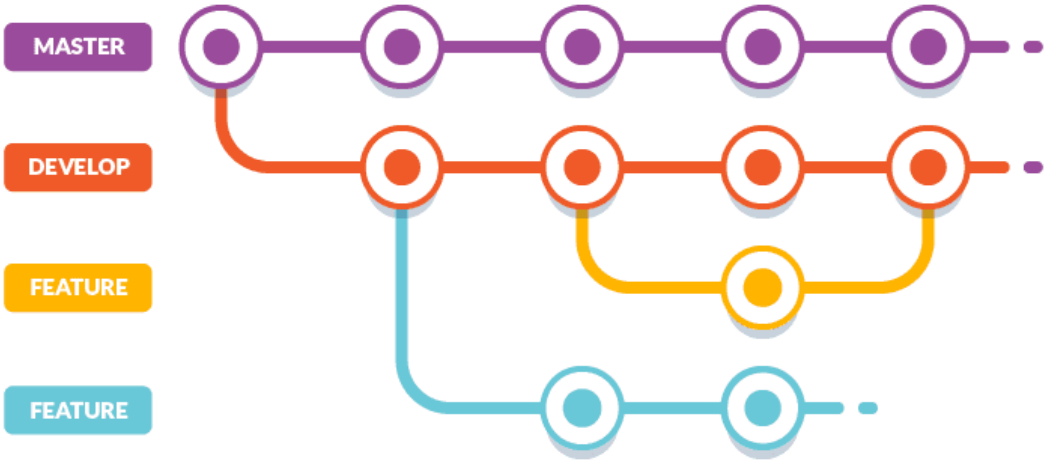
5. Linus Torvald - GIT (Gestão de configuração)

Após desenvolver as primeiras versões do Linux, Linus Torvald estava enfrentando algumas dificuldades em gerenciar as versões do código. Sendo um projeto Open Source, conforme o projeto ganhou popularidade, mais desenvolvedores ficaram interessados em interagir e dar manutenção no código.

Durante muito tempo (1991-2002) o código do Linux, as mudanças eram compartilhadas como correções e arquivos via email e mídias físicas. Em 2002, o projeto começou usar uma DVCS proprietária chamada BitKeeper.

Porém, em 2005, a relação entre a comunidade do Linux e a empresa que desenvolveu BitKeeper foi quebrada e a ferramenta passou a ser paga. Isto alertou a comunidade que desenvolvia o Linux, especialmente Linus Torvalds, que começou a desenvolver a sua própria ferramenta de gerenciamento de versões baseada em lições aprendidas ao usar o BitKeeper. Algumas metas do novo sistema era os seguintes:

Em 2005 nasce o Git, que evoluiu e amadureceu para ser fácil de usar. Hoje sendo amplamente utilizado em diversos projetos e sendo a base de produtos de gerenciamento de versão como o GitHub e o GitLab.



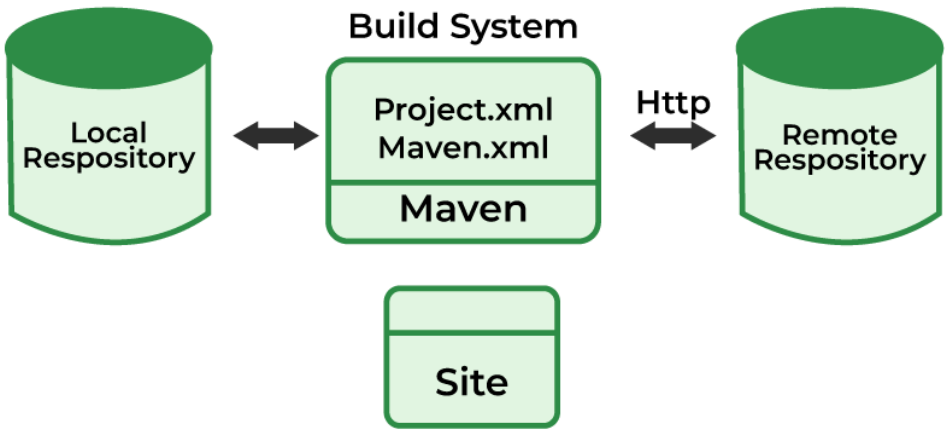
Fonte:
https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F

6. Apache Maven (Gestão de configuração)

O Apache Maven é uma ferramenta de gerenciamento de build de projetos de software, focada em automação e compreensão. Com base no conceito de um Modelo de Objeto de Projeto (POM), o Maven pode gerenciar a construção, os relatórios e a documentação de um projeto a partir de uma peça central de informação.

O Maven simplifica e padroniza o processo de construção de um projeto. Ele cuida de tarefas como compilação, distribuição, documentação, colaboração de equipe e outras tarefas de gerenciamento de projetos.

O Maven gerencia as dependências e verifica a compatibilidade entre diferentes versões de uma biblioteca, reduzindo o risco de conflitos de versão. Ele também suporta a integração com ferramentas de controle de versão como Git e SVN, facilitando o gerenciamento do código fonte.



Fontes:
https://maven.apache.org/what-is-maven.html