FACULDADE DE TECNOLOGIA DE SÃO PAULO

MATHEUS CARDOSO DE SOUZA

Engenharia de Confiabilidade de Sites: Aplicabilidade do SRE nas empresas de tecnologia

SÃO PAULO
2024

FACULDADE DE TECNOLOGIA DE SÃO PAULO
MATHEUS CARDOSO DE SOUZA

Engenharia de Confiabilidade de Sites: Aplicabilidade do SRE nas empresas de tecnologia

Trabalho submetido como exigência parcial
para a obtenção do Grau de Tecnólogo em
Análise e Desenvolvimento de Sistemas
Orientador: Professora Mestre Vânia Franciscon Vieira

SÃO PAULO
2024

Primeiramente ao Senhor Deus, agradeço por Sua divina proteção e pelas bênçãos derramadas ao longo da minha vida. A força e a fé depositadas em mim foram fundamentais para superar os desafios encontrados nesta etapa de aprendizado e crescimento profissional.

Aos meus pais, Maria e José, pilares fundamentais da minha existência, expresso meu mais profundo apreço pelo amor incondicional, apoio e incentivos constantes que me proporcionaram durante toda a minha vida. Sou grato por tudo que fazem por mim.

Por último, mas não menos importante, a todos aqueles que me ajudaram durante esta jornada acadêmica, companheiros e confidentes em todos os momentos, estendo meus sinceros agradecimentos pela amizade inestimável e pelo apoio mútuo que tanto fortaleceram minha caminhada acadêmica.

RESUMO

O Site Reliability Engineering (SRE) é uma abordagem que integra a engenharia de software e a operação de sistemas com o intuito de garantir a entrega de serviços de software mais disponíveis, confiáveis, escaláveis e eficientes. O SRE surgiu da necessidade de atender às crescentes demandas de confiabilidade e disponibilidade de softwares, que são cada vez mais complexos e distribuídos, visando uma abordagem mais eficaz e robusta. A adoção essa cultura por empresas de tecnologia e de outros setores tem aumentado nos últimos anos, à medida que se torna cada vez mais importante garantir a disponibilidade e a confiabilidade de serviços críticos, a partir disto iremos abordar o contexto dos princípios e práticas do SRE e como as empresas que optam por implementá-lo devem se preparar para isto.

Palavras-chave: SRE, Site Reliability Engineering, DevOps, Serviços de T.I. e Operações de T.I.

ABSTRACT

The Site Reliability Engineering (SRE) is an approach that integrates software engineering and systems operation to ensure the delivery of more available, reliable, scalable and efficient software services. SRE arose from the need to meet the growing demands for reliability and availability of software, which is increasingly complex and distributed, aiming for a more effective and robust approach. The adoption of this culture by technology companies and other sectors has increased in recent years, as it becomes increasingly important to ensure the availability and reliability of critical services. From this we will address the context of SRE principles and practices and how companies that choose to implement it should prepare for it.

Keywords: SRE, Site Reliability Engineering, DevOps, I.T. Services, and I.T. Operations.

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1. INTRODUÇÃO

As transformações tecnológicas afetam a maioria das organizações do mundo, se fazendo necessário acompanhar e evoluir as suas ferramentas juntamente com o mercado de T.I. para poder proporcionar serviços mais robustos, escaláveis e ágeis para seus usuários. Devido a essas transformações, surgiu a necessidade de um enorme investimento em ferramentas de automação, implantação, integração contínua e monitoração, o que acabou resultando na criação da disciplina do SRE, que visa fazer entregas mais frequentes, com um padrão mais elevado de qualidade e resiliência, tornando os serviços de T.I. mais disponíveis e confiáveis. A disciplina do SRE vem para ajudar as empresas a mudarem de patamar, com isto quero dizer que a aplicação dos princípios e práticas do SRE somadas com uma gestão de qualidade poderão criar uma cultura mais estruturada e um modelo padronizado de trabalho.

2. O QUE É O SRE?

O termo SRE, abreviação de *Site Reliability Engineering* (Engenharia de Confiabilidade de Sites), foi cunhado por Benjamin Treynor Sloss, até então vice-presidente de engenharia do Google. O termo surgiu enquanto Treynor gerenciava uma equipe de engenheiros de software, que tinha como principal objetivo construir sistemas com maior escalabilidade e alta confiabilidade online nas operações de T.I., a partir disto, Treynor decidiu incorporar os aspectos da engenharia de software à resolução de problemas de operações de tecnologia da informação.

Depois dessa experiência pioneira, o SRE se consolidou como uma disciplina que integra conhecimentos de engenharia de software e os aplica à gestão de sistemas em produção. De acordo com a Red Hat (2024)

> A abordagem de SRE ajuda as equipes a encontrarem um equilíbrio entre lançar novas funcionalidades e assegurar que elas sejam confiáveis para os usuários. [...] Dessa forma, a SRE ajuda a melhorar a confiabilidade do sistema hoje e à medida que cresce ao longo do tempo.

Sendo assim, podemos assumir que o SRE, visa garantir a confiabilidade, escalabilidade de softwares, assegurando a disponibilidade contínua de serviços

online para os usuários, o que acabou representando em uma mudança paradigmática na área de operações de T.I., abrindo caminho para uma abordagem mais proativa e focada na engenharia.

2.1 Como atua uma equipe SRE?

Uma equipe de SRE inicia seu trabalho avaliando os serviços de negócio para determinar o nível de confiabilidade real necessário, essa avaliação leva em consideração diversos fatores, como a natureza do serviço, o impacto de falhas nos usuários e os custos associados à indisponibilidade.

Com base nesta análise, a equipe define a estratégia operacional mais adequada para o serviço, essa estratégia consiste em um conjunto de medidas que visam garantir a confiabilidade desejada e ao mesmo tempo otimizar a velocidade de implantação de software. O objetivo final é fornece novos recursos aos usuários com maior rapidez, sem comprometer a estabilidade dos sistemas, entretanto há algumas exceções que devem ser discutidas, abordaremos elas nos próximos tópicos.

2.1.1 O equilíbrio entre Confiabilidade e Velocidade

Encontrar o equilíbrio ideal entre confiabilidade e velocidade é um desafio central para o SRE, tendo em vista que as equipes devem buscar soluções que minimizem o tempo de inatividade dos sistemas, sem comprometer a agilidade da entrega de novos recursos, porém uma análise cuidadosa dos riscos e dos benefícios é necessária em cada tomada de decisão.

2.1.2 Transparência e Compartilhamento de Conhecimentos

O Google tem se mostrado uma empresa referência na implementação do SRE, compartilhando abertamente suas experiências com a comunidade global de SRE, tanto as bem-sucedidas quanto as desafiadoras. Essa transparência contribui para a difusão da metodologia e para o aprimoramento das práticas do SRE em diversas organizações.

3. PAPÉIS E RESPONSABILIDADES DE UM SRE

A confiabilidade emerge como uma característica fundamental em sistemas de T.I., estabelecendo a base para o desenvolvimento e para a operação eficientes. Ben Treynor, defende que a busca pela confiabilidade deve nortear a construção e o gerenciamento de sistemas, permitindo a otimização de recursos e a criação de valor para as empresas sistemas, além da busca incessável no aprimoramento do design de sistemas para torná-los mais confiáveis, escaláveis e eficientes, direcionando esforços para a criação de novas funcionalidades e produtos, impulsionando o crescimento e a inovação. Nesse contexto, surge a necessidade de profissionais qualificados para assumirem o papel de SRE, que combina habilidades de administração de sistemas com conhecimentos em desenvolvimento de código e automação. Para descrevermos as atribuições e competências um profissional de SRE, podemos incorporar as definições da IBM e da Red Hat.

Um SRE ideal é um profissional com experiência em desenvolvimento de software e em operações de T.I. (IBM, 201-), com responsabilidades de implantar sistemas, monitorar eventos, gerir mudanças, responder às emergências e gerir a capacidade dos serviços em produção (Red Hat, 2024).

Ao aplicar a Engenharia de Confiabilidade de Sites nos concentramos em oito princípios de operação de um serviço, estes princípios são a base de todo bom engenheiro de SRE, logo uma equipe competente deve ter em mente a melhor maneira de aplicar tais princípios às suas operações:

- **Desempenho:** é a capacidade de comportamento e rendimento do serviço realizado.
- **Disponibilidade:** quanto um serviço está disponível para ser utilizado.
- **Eficiência:** é a capacidade de realizar operações com menos recursos e tempo.
- **Gerenciamento de Capacidade dos Serviços:** é o dimensionamento planejado, crescimento projetado, ou seja, a capacidade de tratar os serviços com elasticidade de forma manual e automática.
- **Gerenciamento de Mudanças:** são as estratégias de implantação e controle de qualidade.
- **Latência:** quanto tempo o serviço atende a uma solicitação.

- **Monitoramento:** são visões de monitoração que apoiam no diagnóstico, rastreabilidade, métricas etc.
- **Resposta a Emergências:** são procedimentos a serem realizados em caso de eventuais emergências.

4. COMO A METODOLOGIA SRE SE APLICA NAS EMPRESAS?

Em empresas de grande porte, como Amazon, Facebook, X (o antigo Twitter) e outras gigantes do mercado de tecnologia, o modelo SRE é adotado, porém na maior parte dos casos ele é adaptado conforme as necessidades da empresa. Essas adaptações são o que podem melhorar ou piorar a metodologia a depender das alterações realizadas.

A implementação da Engenharia de Confiabilidade de Sites em uma organização pode ser um processo desafiador, para promover uma transformação eficiente na sua implementação é de extrema importância que haja uma mudança cultural significativa, além da mudança de mentalidade, da colaboração, do patrocínio da hierarquia e do apoio da liderança.

As operações de T.I. possuem atividades muito complexas, e o seu desafio transcende a execução eficiente de sistemas, há também uma questão, que ainda busca soluções definitivas, a gestão eficaz das equipes. A recente revolução na organização do trabalho em T.I. busca solucionar este conjunto de problemas de gestão através de duas metodologias distintas, o DevOps e o SRE. Apesar de serem frequentemente tratadas como entidades separadas, estas duas metodologias representam soluções convergentes para os desafios das operações de Tecnologia da Informação e apesar das distinções em suas origens e ênfases, ambas as abordagens compartilham princípios e valores fundamentais, como foco no cliente, colaboração, automação, aprendizado contínuo e experimentação.

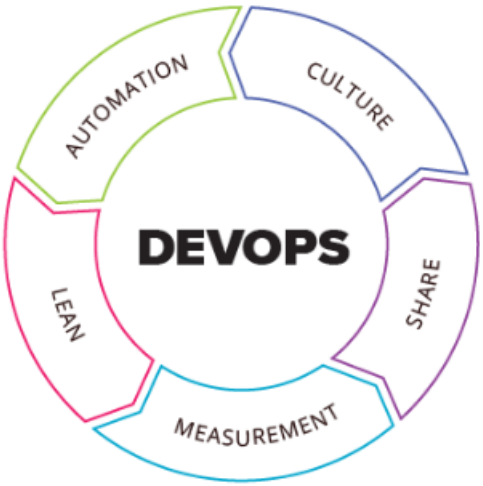
4.1 Imersão no DevOps

Kim *et al.* (2018) nos propõe uma reflexão sobre o *DevOps*, “imagine um mundo onde donos de produtos, Desenvolvimento, Q.A., Operações de T.I. e InfoSec trabalhem juntos, não apenas para ajudar uns aos outros, mas também para garantir o sucesso da organização como um todo”, a partir desta reflexão podemos inferir que

o DevOps é um conjunto abrangente de práticas, diretrizes e cultura, idealizado para eliminar as barreiras entre as áreas de desenvolvimento de software, operações, redes e segurança da informação dentro de uma organização.

O modelo *CALMS*, proposto por John Willis, Damon Edwards e Jez Humble, apresenta as principais diretrizes para a implementação do DevOps, direcionadas tanto para empresas em processo de transição para essa estrutura, quanto para aquelas que buscam aprimorar suas implementações já existentes.

Ilustração 1 *- Modelo CALMS*



Fonte: Oleksandr Mykhalchuk

CALMS é um acrônimo que representa os cinco pilares fundamentais do DevOps:

- ***Culture*****:** Deve-se fomentar a colaboração e o compartilhamento de conhecimentos entre as equipes de desenvolvimento e operações para promover um ambiente de trabalho sinérgico e interdisciplinar.
- ***Automation*****:** Deve-se automatizar as tarefas repetitivas e manuais, a fim de disponibilizar mais tempo para as equipes se concentrem em atividades de maior valor estratégico, como inovação e otimização de processos.
- ***Lean*****:** Deve-se adotar metodologias de desenvolvimento ágil e fluxo contínuo, para possibilitar entregas frequentes e confiáveis de software.
- ***Measurement*****:** Deve-se estabelecer indicadores-chave de performance (KPIs) relevantes para monitorar o desempenho do

processo DevOps, identificando áreas de melhoria e otimizando continuamente a entrega de valor.

- ***Share*:** Deve-se promover a cultura de compartilhamento de conhecimentos, práticas e aprendizados entre as equipes, criando um ambiente propício para a inovação colaborativa e resolução conjunta de problemas.

A implementação da filosofia DevOps exige um compromisso cultural por parte da organização, criando um ambiente propício à colaboração, à experimentação e ao aprendizado contínuo. Essa mudança cultural é fundamental para o sucesso da abordagem DevOps, permitindo que as equipes trabalhem de forma mais integrada e eficiente, impulsionando a entrega de valor para o cliente e o crescimento da organização como um todo.

A filosofia DevOps é uma abordagem inovadora para o desenvolvimento e entrega de software, criando-se assim um ciclo virtuoso de melhoria contínua através da colaboração entre equipes de desenvolvimento e operações, desta maneira é possível promover um alto acoplamento entre todos os cinco pilares do DevOps, porém há ideias que podem ser discutidas separadamente.

Um dos pilares fundamentais da filosofia DevOps e a primeira ideia é a *eliminação de silos organizacionais*, por muitas das vezes, são eles que impedem a comunicação eficaz e a colaboração entre as equipes. Para superar esse obstáculo, a adoção de um hábito de compartilhamento transparente e constante de informações se torna crucial, pois, somente através da comunicação aberta e da troca de conhecimentos, as equipes podem trabalhar de forma coesa e assim alcançar resultados mais eficazes.

A filosofia DevOps reconhece que acidentes e falhas são inevitáveis em qualquer processo, por isso, a segunda ideia é *apreender com os erros*. Ao invés de atribuir a culpa a indivíduos específicos, essa abordagem enfatiza a importância de identificar as falhas sistêmicas que contribuem para os erros. Através da análise crítica dos erros e da implementação de salvaguardas adequadas, as equipes podem aprender com suas experiências e evitar que os mesmos problemas se repitam no futuro.

Na terceira ideia, a filosofia DevOps propõe a *mudança gradual e sustentável*, ou seja, dividir as entregas em subcomponentes menores e implementá-los em uma pipeline estável de mudanças de baixo risco. Essa abordagem incremental permite

que as equipes testem e avaliem as mudanças de forma contínua, minimizando o impacto negativo de falhas e otimizando o processo de adaptação.

A quarta ideia é *promover ferramentas e automação*, pois, sabemos que o ferramental desempenha um papel importante na filosofia DevOps, especialmente no contexto da automação de tarefas repetitivas e manuais. Ao automatizar processos, as equipes liberam tempo para se concentrarem em atividades de maior valor estratégico, como a inovação e a resolução de problemas complexos. No entanto, a filosofia DevOps reconhece que o sucesso na adoção de ferramentas depende fundamentalmente da cultura organizacional e do compromisso com a colaboração e o aprendizado.

A quinta e última ideia é a *medição como base para a tomada de decisões*, sendo um elemento essencial da filosofia DevOps, essa ideia permite que as equipes avaliem o impacto das mudanças implementadas e identifiquem áreas que necessitam de novos aprimoramentos. Através da coleta e análise de dados, as equipes podem tomar decisões mais estratégicas e assertivas, direcionando seus esforços para as áreas que realmente geram um impacto positivo.

4.2 Imersão no SRE

Na seção anterior, exploramos o DevOps como um conjunto abrangente de princípios que promovem a colaboração entre as equipes de desenvolvimento e operações durante todo o ciclo de vida de um produto. Nesta seção, aprofundaremos nossa análise no *Site Reliability Engineering*, uma função de trabalho que complementa o DevOps, ao aplicar princípios de engenharia de software para resolver problemas operacionais e garantir a confiabilidade, a escalabilidade e a disponibilidade dos sistemas. O SRE se baseia em seis princípios concretos que guiam suas práticas e decisões:

1. **Operações são um Problema de Software:** O SRE reconhece que as operações de sistemas complexos podem ser abordadas como problemas de software, utilizando métodos e ferramentas de engenharia de software para solucionar falhas, otimizar o desempenho e garantir a confiabilidade dos sistemas.
2. **Gerenciar por Objetivos de Nível De Serviço (SLOs):** Ao invés de focar em métricas de infraestrutura isoladas, o SRE define objetivos de

nível de serviço (SLOs) que traduzem as expectativas de desempenho e disponibilidade do sistema para a experiência do usuário. Esses SLOs servem como base para a tomada de decisões e a medição do sucesso das iniciativas SRE.

3. **Minimizar o trabalho Pesado (Toil):** O SRE busca automatizar tarefas repetitivas e manuais, liberando tempo para que os profissionais se concentrem em atividades de maior valor estratégico, como a otimização da infraestrutura, a criação de ferramentas e a resolução de problemas complexos.
4. **Automatizar o Trabalho Antecipadamente:** O SRE assume uma postura proativa na automação, buscando identificar e automatizar tarefas antes que elas se tornem um problema. Essa abordagem preventiva permite que as equipes se concentrem em atividades mais estratégicas e reduzem o tempo de resposta a incidentes.
5. **Mover-se Rapidamente Reduzindo o Custo do Fracasso:** O SRE reconhece que falhas e incidentes são inevitáveis em sistemas complexos. O foco central reside na redução do tempo médio de reparo (MTTR) e na implementação de mecanismos de recuperação rápida para minimizar o impacto negativo das falhas.
6. **Compartilhar a Propriedade com os Desenvolvedores:** O SRE propõe a quebra de silos entre as equipes de desenvolvimento e operações, promovendo a colaboração e a responsabilidade compartilhada pela confiabilidade e a disponibilidade dos sistemas. Essa colaboração permite que as equipes trabalhem de forma mais integrada e eficiente, otimizando o processo de desenvolvimento e entrega de software.

O DevOps e o SRE são conceitos frequentemente associados, mas com distinções importantes e a escolha entre um ou o outro depende das necessidades específicas da organização e do contexto do negócio. O DevOps oferece uma abordagem mais abrangente para o desenvolvimento e entrega de software, enquanto o SRE se concentra na otimização da operação e da confiabilidade dos sistemas.

4.3 A Importância das Ferramentas

A padronização das ferramentas é um elemento crucial para a efetividade da metodologia SRE, pois, com a utilização de um conjunto unificado de ferramentas pelas equipes, garante diversos benefícios, como a consistência, a precisão, a redução da complexidade e o aumento da eficiência, porém ainda não há uma boa maneira de gerenciar um serviço que tem uma ferramenta para os SREs e outra ferramenta distinta para os desenvolvedores, pois, se comportam de maneiras diferentes em diversos aspectos e quanto mais divergência houver, menos a empresa se beneficiará com cada esforço para melhorar as ferramentas individualmente.

4.4 Prontidão Operacional

No cenário atual, a experiência do cliente é o ponto principal para o sucesso das organizações, eles exigem serviços mais rápidos, entregas mais ágeis de novos produtos e recursos, além de soluções confiáveis, com isto, qualquer falha na infraestrutura pode gerar impactos negativos na experiência do cliente, levando à frustração e à perda de oportunidades de negócio. O tempo excessivo para identificar a causa raiz do problema e restaurar o serviço agrava essa situação, exigindo medidas proativas para garantir a prontidão operacional, portanto, as empresas precisam se concentrar em fortalecer sua infraestrutura e garantir a prontidão operacional para atender a essas demandas. Diante desta situação uma avaliação abrangente da prontidão operacional é uma ferramenta essencial para que as empresas identifiquem os pontos fracos e implementem melhorias contínuas.

Uma avaliação eficaz de prontidão operacional, deve responder a três perguntas fundamentais para nortear as ações de aprimoramento, “*O que precisa mudar?”*, *“Quão significativa é a mudança?”* e “*Quais são os benefícios esperados?”*. As respostas a essas perguntas fornecem *insights* valiosos para identificar as lacunas nos processos que não atendem aos requisitos mínimos. Com base nessas informações, as organizações podem revisar, analisar, melhorar e monitorar seus planos de ação além de implementar soluções mais adequadas e medir o impacto das mudanças.

Através das avaliações de prontidão operacional e da realização de uma revisão mais aprofundada, as organizações podem identificar as áreas que mais

precisam de atenção e implementar as mudanças necessárias para garantir uma operação mais eficiente, confiável e resiliente. Isso contribui para a redução das falhas de aplicativos, a diminuição da frustração dos clientes e a otimização dos recursos, gerando diversos benefícios para a organização como um todo.

5. O QUE É NÍVEL DE SERVIÇO?

De acordo com Beyer *et al.* (2016), “é impossível gerenciar um serviço corretamente, e muito menos bem, sem compreender quais comportamentos realmente importam para esse serviço e como medir e avaliar esses comportamentos”. Para suprir o déficit de medição e avaliação dos comportamentos dos serviços de software, foi estabelecido o Nível de Serviço como métrica.

O Nível de Serviço é um indicador da qualidade dos serviços prestados, que abrange a eficiência dos serviços além do produto em si, estabelecendo parâmetros de desempenho e disponibilidade que devem ser cumpridos pela organização. O SRE, por sua vez, surge como uma metodologia eficaz para o, monitoramento e otimização do nível de serviço no pós-implantação, garantindo a confiabilidade e disponibilidade dos serviços, impulsionando a inovação e minimizando impactos negativos para o consumidor.

5.1 Objetivo de Nível de Serviço (SLO)

Quando o Google propôs os termos do SRE, o intuito era definir uma meta numérica precisa para a disponibilidade do sistema, desde então convencionou-se chamar esta meta de “objetivo de nível de serviço” ou SLO, que especificam um nível alvo para a confiabilidade de um serviço. Caso o resultado fique acima do nível alvo é esperado que os usuários fiquem satisfeitos com a utilização do serviço, mas caso o resultado fique abaixo do nível alvo, muito provavelmente os usuários reclamem ou parem de utilizar o serviço oferecido. O principal objetivo é garantir que o serviço atenda às expectativas dos usuários e atinja o nível de confiabilidade mínimo esperado pela organização.

Deve-se adotar uma abordagem criteriosa e equilibrada ao estabelecer o SLO de um serviço, definindo um nível mínimo e aceitável de confiabilidade para ele e esse nível deve ser determinado com base nas necessidades dos usuários, nos custos de

operação e nos recursos disponíveis. Com o SLO é possível julgar se o recurso precisa ser mais ou menos confiável, nos casos em que há uma disponibilidade excessiva, a Atlassian (201-) adverte que “é sempre melhor prometer menos e entregar mais”, pois, é possível gerar uma expectativa indesejada para os serviços, por isto, é importante evitar a busca por níveis excessivos de confiabilidade se não houver pretensão para tal, pois pode gerar custos desnecessários e desviar recursos de outras áreas importantes.

5.2 Acordo de Nível de Serviço (SLA)

O SLA estabelece um acordo formal entre o provedor de serviço e o cliente, definindo os parâmetros de qualidade esperados para o serviço em um determinado período, esses parâmetros podem incluir, a disponibilidade, o desempenho, a segurança e o suporte. Caso estes parâmetros não sejam atendidos haverá algum tipo de penalidade a ser paga, que é definida em consenso entre ambas as partes em um acordo formal, garantindo aos clientes a ciência do que podem esperar dos serviços contratados e permite ao provedor do serviço gerenciar expectativas e estabelecer um relacionamento de confiança com seus clientes.

5.3 Indicador de Nível de Serviço (SLI)

O SLI tem a função de indicador do comportamento de um serviço, medindo a frequência de sondagens bem-sucedidas em um sistema. O cálculo do SLI é dado pela razão de dois números, resultando em uma porcentagem que varia de 0% (indica que o serviço apresenta falhas significativas) até 100% (indica que o serviço está operando normalmente, sem falhas detectadas). É importante ressaltar que a interpretação do SLI precisa ser contextualizada de acordo com o tipo de serviço e o SLA estabelecidos, por exemplo, um SLI de 99% para um serviço de e-mail pode ser considerado aceitável, enquanto para um serviço de transações financeiras, o mesmo valor pode ser inaceitável.

Ilustração 2 *- Fórmula do SLI*

Fonte: Jesus Climent

A disponibilidade de um serviço, medida pela fração de tempo em que ele está utilizável, é um SLI crucial para os SREs, pois, ele representa um indicador quantitativo que mede o nível de serviço fornecido aos usuários, geralmente calculado como a fração de requisições atendidas com sucesso. Embora alcançar 100% de disponibilidade seja um objetivo teórico, o mercado de tecnologia almeja valores próximos a esta marca, expressos em números nove após a primeira casa decimal, por exemplo, 99,9% ou 99,99% de disponibilidade.

Ao avaliar se um serviço está atendendo ao seu SLO em um determinado período, o SLI da disponibilidade é a métrica chave para esta avaliação, pois, se o SLI ficar abaixo do SLO especificado, então o serviço não está atendendo às expectativas e medidas devem ser tomadas para aumentar sua disponibilidade, a fim de garantir uma melhor experiência para os usuários do serviço, reduzir custos com indisponibilidades e aumentar a confiabilidade do serviço.

6. ORÇAMENTO DE ERRO

O Orçamento de Erro, também conhecido como, *Error Budget*, é a tolerância máxima de tempo de indisponibilidade para um serviço em um período específico sem consequências contratuais (Atlassian, 201-). Essa métrica serve como guia para a tomada de decisões estratégicas, permitindo que equipes equilibrem a entrega de novos recursos com a manutenção da qualidade e confiabilidade do serviço.

Além disso, o *Error Budget* vai além de definir um limite de indisponibilidade para um serviço, funcionando como um verdadeiro sistema de alerta antecipado, sinalizando quando necessário tomar medidas corretivas para garantir a confiabilidade do serviço. Ao implementá-lo como parte de uma cultura de SRE, as empresas podem discutir e tomar decisões estratégicas sobre a priorização do trabalho, otimizando o tempo e os recursos disponíveis, além de alcançarem um equilíbrio entre inovação e confiabilidade, impulsionando o sucesso do negócio.

6.1 Políticas de Orçamento de Erro

A Blameless (2020) afirma que não é suficiente saber qual o seu *Error Budget*, é necessário saber o que fazer em caso de extrapolar esse orçamento. Por isso, as Políticas de Orçamento de Erro se fazem necessárias, pois, complementam o Orçamento de Erro, transformando-o em um guia prático para a ação, indo além de definir limites de indisponibilidade, estabelecendo um conjunto detalhado e estruturado de ações a serem tomadas quando esses limites forem atingidos, determinando onde a equipe de SRE ou de Operações deve direcionar os desenvolvedores para corrigir problemas de confiabilidade, garantindo que o serviço seja restaurado o mais rápido possível.

7. ELIMINANDO O TOIL

O *Toil* é um termo utilizado para descrever tarefas manuais e repetitivas que consomem tempo e esforço das equipes de engenharia, tornando-se um obstáculo significativo para a produtividade e a inovação. Tarefas como a configuração de infraestrutura, resolução de problemas manuais, gerenciamento de logs e monitoramento, embora essenciais, são tediosas e tendem a desmotivar as equipes.

Os SREs assuem um papel fundamental no processo de automatização dessas tarefas, pois, através da sua expertise, em infraestrutura, monitoramento, automação e resolução de problemas, são aptos para liderar a implementação de ferramentas e práticas que eliminem o Toil, permitindo que as equipes envolvidas no desenvolvimento do serviço se concentrarem no que realmente importa, a criação de soluções tecnológicas inovadoras e eficientes.

7.1 Características do Toil

Segundo Beyer *et al.* (2016), o Toil “é o tipo de trabalho vinculado à execução de um serviço de produção que tende a ser manual, repetitivo, automatizável, tático, desprovido de valor duradouro e que se dimensiona linearmente à medida que o serviço cresce”. Entretanto, essa definição captura apenas a essência dos tipos de Toil, na prática, ele se manifesta de forma ampla e diversificada, sendo assim, nem todas as tarefas que se encaixam na descrição original apresentam todos os atributos

mencionados. A chave para identificar o Toil em uma tarefa reside em reconhecer a presença de pelo menos um dos atributos citados por Beyer *et al.* (2018):

- **Manual:** tarefas que exigem interação humana direta e repetitiva, sem automação possível.
- **Repetitivo:** tarefas que seguem um padrão fixo, sem variação significativa ou necessidade de criatividade.
- **Automatizável:** tarefas com o potencial para serem automatizadas, mas que ainda são realizadas manualmente por falta de implementação ou conhecimento técnico.
- **Não Tático:** tarefas com foco em ações imediatas e sem impacto estratégico a longo prazo.
- **Desprovido de valor duradouro:** tarefas que não geram benefícios a longo para a empresa ou para o usuário final.
- **Dimensionamento linear:** tarefas que aumentam em proporção direta ao crescimento do serviço, demandando mais tempo e recursos humanos sem otimização.

A presença de apenas um destes atributos citados, já é caracterizado como Toil, mesmo que não apresente os demais, isso ocorre porque, mesmo que a tarefa não seja totalmente manual ou repetitiva, por exemplo, a falta de automação ou a ausência de valor duradouro já se torna um impeditivo para a produtividade e a inovação.

7.2 O que não é Toil

O Toil não se define simplesmente por ser um trabalho desagradável ou por se diferenciar de tarefas rotineiras, mas sim por possuir um caráter improdutivo e sem valor duradouro. Por vezes há uma confusão entre os profissionais sobre o que se caracteriza ou não como Toil. Tarefas comuns com propósitos definidos e com um valor tangível, não são consideradas Toil, mesmo que essas tarefas sejam repetitivas e complexas. Elas não se enquadram, pois, elas contribuem para o funcionamento do dia a dia da empresa e muitas vezes são essenciais para o seu progresso.

8. OBSERVABILIDADE, TELEMETRIA E MONITORAMENTO

A base da Observabilidade, da Telemetria e do Monitoramento, parte do princípio que diz, que se deve aprender tudo o que acontece dentro dos sistemas, muito além das linhas código. Essa premissa permite desvendar os mistérios por traz de todo o negócio, com o intuito de prevenir interrupções e localizar eventuais falhas de forma ágil e eficiente, garantindo assim um nível de confiabilidade mais elevado.

No mundo corporativo, a construção de softwares resilientes e com alto nível de disponibilidade sempre foi um objetivo a ser alcançado, com a motivação de garantir a continuidade das operações e a satisfação dos clientes. No entanto, alcançar essa meta se tornou um desafio ainda maior após as migrações de arquiteturas monolíticas para arquiteturas *cloud-native*.

Em arquiteturas monolíticas, a simplicidade do design proporcionava visibilidade completa das aplicações, isso só era possível, porque o monitoramento por meio de métricas e o registro de logs auxiliavam na identificação e na resolução de problemas, entretanto esse tipo de arquitetura apresentava limitações em ambientes de nuvem, pois, as arquiteturas do tipo *cloud-native* se caracterizam por serem feitas componentes menores, de curta duração e efêmeros, tornando os sistemas mais dinâmicos e granuláveis em relação a arquitetura monolítica, inviabilizando o uso somente de técnicas tradicionais de monitoramento e se fazendo necessário a utilização do monitoramento de ponta a ponta, para garantir uma melhor visibilidade dos serviços e uma rápida resolução de problemas.

O monitoramento de ponta a ponta é uma abordagem abrangente para monitorar o desempenho e a saúde de um sistema como um todo, desde a infraestrutura até a experiência do usuário final. Esse tipo de monitoramento vai além da análise dos tradicionais logs, ele coleta métricas, logs e rastreia as requisições e os eventos acionados, organizando esses dados em um sistema centralizado (como o *Grafana* e o *Datadog*), onde podem ser analisados para identificar problemas e tendências.

8.1 Observabilidade

A IBM (201-), diz que a Observabilidade “é a medida em que se pode compreender o estado interno ou condição de um sistema complexo baseando-se apenas no conhecimento de suas saídas externas”. Em outras palavras, a observabilidade nos permite “ler as entre linhas” das saídas do sistema, desvendando seus comportamentos e características intrínsecas.

Em termos formais, um sistema é considerado observável se o seu estado atual puder ser estimado usando apenas informações de saída, ou seja, a partir dos dados identificados que foram coletados. Isso significa que, através da análise aprofundada das saídas do sistema, podemos inferir seu estado interno, incluindo variáveis e comportamentos não diretamente observáveis.

Com as migrações para ambientes *cloud-native*, impulsionadas por provedores como, Amazon Web Services (AWS), Microsoft Azure e o Google Cloud Platform (GCP), ficou muito mais fácil para as empresas desenvolverem, implantarem e gerenciarem serviços e recursos para seus sistemas mais distintos, sejam eles microsserviços, *serverless* e até mesmo *containers*, pois, promovem agilidade, escalabilidade e flexibilidade. No entanto, essa arquitetura distribuída e descentralizada também introduziu desafios, especialmente no rastreamento de eventos, sendo necessário milhares de processos em execução para que ocorra de maneira eficaz.

A observabilidade permite que as empresas monitorem suas arquiteturas distribuídas de maneira eficaz, os ajudando a encontrar e conectar os efeitos de uma cadeia detalhada de eventos e assim rastreá-los até a causa raiz, oferecendo uma visibilidade de toda a arquitetura.

8.1.1 Ferramentas de Observabilidade

Kidd (2023) nos fala que “os produtos de observabilidade são projetados para ajudar desenvolvedores, equipes de T.I. e outras partes interessadas a monitorar e gerenciar sistemas, aplicativos e infraestrutura complexos.”, por isso, é necessário a utilização de ferramentas apropriadas para coletar dados de telemetria minimamente adequados. Há dois tipos de ferramentas disponíveis no mercado atualmente, as *open source* (ferramentas de código aberto) ou as soluções comerciais pagas.

A escolha entre uma ou outra é dependente das necessidades específicas de cada empresa, por exemplo, para empresas com conhecimento técnico e orçamento limitado, as ferramentas *open source* podem ser a melhor escolha. Entretanto, para empresas que buscam facilidade de uso, suporte dedicado e recursos avançados, as soluções comerciais pagas podem ser a escolha mais adequada.

8.2 Telemetria

Tebaldi (2019) define a telemetria como “[...] uma tecnologia que permite a medição remota e a comunicação de informações entre sistemas, através de dispositivos de comunicação sem fio, como ondas de rádio ou sinais de satélite”. Entretanto, essa definição é muito abrangente, por isso, na área de tecnologia aumentamos a precisão desta definição dizendo que, é a medição dos dados de diversas fontes para um local centralizado, onde os dados serão monitorados e analisados com o intuito de aprimorar as aplicações, mas principalmente atender as necessidades dos clientes.

Antes das migrações para o ambiente *cloud-native* a coleta de telemetria era predominantemente feita por meio de profissionais que implementavam logs informativos em cada peça monolítica. Embora essa abordagem ainda seja eficaz em alguns casos, o cenário de coleta de telemetria se transformou drasticamente ao decorrer do tempo, tendo um volume de dados infinitamente maior a ser analisado, devido a proliferação de *endpoints* e a adoção de sistemas e *frameworks open source* que disponibilizam naturalmente o rastreamento eventos, logs e outras métricas. Por isso, centralizar todos esses dados em um único lugar facilita tanto no gerenciamento e prevenção de incidentes, pois as ferramentas de centralização, como *Datadog*, podem aplicar filtros nos dados, observando padrões de comportamento, anomalias e correlações entre os dados, que provavelmente um humano não conseguiria achar.

8.3 Monitoramento

O monitoramento de sistemas, se refere à utilização de soluções ou ferramentas que permitem às equipes observarem e determinarem o estado de saúde de seus sistemas. Essa prática proativa oferece uma visão abrangente da situação, indo além da mera identificação de falhas, abrangendo a antecipação de problemas e

a garantia da disponibilidade e do desempenho ideal dos sistemas. De acordo com Ravichandran, Taylor e Waterhouse (2016)

> Monitorar é um verbo, algo que executamos em nossos aplicativos e sistemas para determinar seu estado de testes básicos de condicionamento físico e se eles estão ativos ou inativos. Há verificações de integridade e desempenho mais proativas. Monitoramos os aplicativos para detectar problemas e anomalias.

Tecnologias de Gerenciamento de Desempenho de Aplicativos (APM), como o *New Relic* e o *Dynatrace*, oferecem uma visão profunda da experiência do usuário. Desta forma, ao invés de depender de agentes externos, essas ferramentas permitem a instalação de bibliotecas no código-fonte que capturam detalhes internos da linguagem e das requisições, fornecendo dados valiosos para identificar gargalos e otimizar o desempenho. Através da agregação e correlação, podemos desvendar padrões e tendências, obtendo inferências valiosas sobre o sistema em questão.

8.3.1 Tipos de Monitoramento

De forma sucinta, há dois tipos de monitoramento, o primeiro é o monitoramento de caixa branca (também conhecido como monitoramento intrusivo), que oferece uma visão profunda do funcionamento interno do sistema. O segundo que é o monitoramento de caixa preta (também conhecido como monitoramento não intrusivo), que foca no comportamento externo do sistema, sem a necessidade de acesso ao seu código interno.

8.4 Monitoramento Versus Observabilidade

O Monitoramento e a Observabilidade, embora conceitos distintos, se complementam e a realidade é que não se pode ter um sem o outro, ou seja, você pode monitorar sem observar, mas isso fará com que o entendimento e a resolução de possíveis erros, não serão tão eficientes e rápidos (Tang, 2021), por isso, ambos são elementos essenciais para garantir a saúde e o bom funcionamento de sistemas.

O Monitoramento consiste em observar o desempenho de um sistema ao longo do tempo, utilizando ferramentas que coletam e analisam dados relevantes. Essa

prática permite identificar padrões, tendências e anomalias, possibilitando uma resposta rápida e eficaz em caso de falhas ou erros.

A Observabilidade vai além da coleta de dados, focando na interpretação e análise profunda do comportamento do sistema. Através da análise de dados e *insights* gerados pelo monitoramento, a observabilidade oferece uma visão holística do sistema, incluindo sua integridade, desempenho e comportamento.

8.5 Golden Signals

No mundo dos sistemas de software, especialmente em ambientes DevOps e SRE, os 4 Sinais de Ouro são indicadores para a observação da saúde e do desempenho de um sistema. São métricas que fornecem uma visão holística do funcionamento do sistema, permitindo a rápida identificação de problemas e a tomada de decisões estratégicas para otimização e resolução de falhas. Sendo eles:

1. **Latência:** é o tempo que leva para um evento ocorrer ou para uma informação ser transmitida de um ponto para outro.
2. **Tráfego:** se refere à quantidade de dados que fluem pelo sistema em um determinado período.
3. **Erros:** é referente a taxa de falha das solicitações.
4. **Saturação:** é a capacidade do sistema e aos recursos que dispõem.

Ao monitorar e analisar esses sinais de forma eficaz, os SREs podem identificar problemas rapidamente, otimizando seus sistemas, melhorando a experiência do usuário e impulsionando o crescimento do negócio.

9. ANTIFRAGILIDADE E POSTMORTEM

As crescentes demandas dos clientes por serviços online que sejam, rápidos, seguros e altamente disponíveis, exigem que as organizações priorizem a resiliência e a confiabilidade de suas aplicações. Embora seja praticamente impossível evitar todos os incidentes, é de extrema importância aprender com eles, para que as organizações possam identificar pontos fracos, implementar medidas preventivas e aprimorar seus processos de recuperação.

O medo do fracasso é um sentimento natural que permeia a maioria das organizações e muitas vezes leva à criação de uma cultura do medo, onde a

experimentação e a assunção de riscos são desencorajadas. Paradoxalmente, essa mesma cultura do medo pode impulsionar as organizações a adotarem o modelo de implantação conhecido como *Big Bang*.

O *Big Bang* envolve a implementação abrangente de um novo sistema ou processo de uma só vez, geralmente sem testes extensivos ou pilotos. Essa abordagem, embora aparentemente eficiente, apresenta um grau de risco e incerteza muito elevados.

9.1 O que é a Antifragilidade?

De acordo com Taleb (2020)

> Algumas coisas se beneficiam de choques; eles prosperam e crescem quando expostos à volatilidade, aleatoriedade, desordem e fatores estressantes e amam a aventura, o risco e a incerteza. No entanto, apesar da onipresença do fenômeno, não há palavra para o exato oposto de frágil. Vamos chamá-lo de antifrágil. A antifragilidade está além da resiliência ou robustez. O resiliente resiste a choques e permanece o mesmo; o antifrágil fica melhor.

Nesse contexto, práticas como *Site Reliability Engineering*, DevOps e a Antifragilidade se convergem em uma jornada em busca da previsibilidade e da capacidade de lidar com a desordem de forma estratégica, buscando entender por que as coisas quebram, como consertá-las e como evitar que quebrem novamente.

Em seu livro, Taleb (2020) propõe uma visão revolucionária sobre a natureza dos sistemas, introduzindo uma classificação tripartite que desafia as concepções tradicionais de robustez e resiliência.

Na base da pirâmide residem os *sistemas frágeis* que tendem ao caos e à volatilidade, eles são caracterizados por sua incapacidade de lidar com eventos inesperados e distúrbios, esses sistemas se deterioram e falham diante de choques, mesmo que de pequena magnitude. Exemplos clássicos incluem, castelos de cartas, sistemas financeiros instáveis e empresas inflexíveis que não conseguem se adaptar às mudanças do mercado.

O segundo nível é ocupado pelos *sistemas robustos* que são mais resistentes que seus pares frágeis mas ainda assim são incapazes de prosperar, esses sistemas podem suportar choques e distúrbios até certo ponto, retornando ao seu estado original após a perturbação. No entanto, eles não se beneficiam do caos e da

volatilidade, permanecendo estáveis e inalterados, sem capacidade de aprender e se fortalecer com as adversidades. Exemplos incluem, máquinas projetadas para operar em ambientes específicos, burocracias rígidas e sistemas imunológicos que não se adaptam a novas doenças.

No ápice da classificação se encontram os *sistemas antifrágeis,* que ao contrário dos sistemas frágeis que se deterioram e dos sistemas robustos que apenas resistem, os sistemas antifrágeis se beneficiam do caos, da volatilidade e do estresse, tornando-se mais fortes e robustos após serem expostos a eles. Exemplos incluem, o sistema imunológico humano que se fortalece ao combater patógenos, empresas inovadoras que aprendem com os erros e sociedades que se adaptam às mudanças.

O DevOps reconhece que a falha é inerente a qualquer sistema complexo e ao invés de tentar eliminá-la completamente, defende a aceitação da falha como um elemento fundamental do processo de aprendizado e crescimento. Ao invés de focar em evitar falhas a todo custo, deve-se concentrar em métricas como o Tempo Médio para Detectar Falhas Incidentes (MTTD), o Tempo Médio de Recuperação de Componentes (MTTR) e o Tempo Médio de Recuperação de Serviço (MTTRS), pois, elas fornecem informações valiosas sobre a rapidez com que os sistemas são detectados, reparados e restaurados, permitindo identificar áreas para melhorias.

Transformar a cultura organizacional de um medo do fracasso para um investimento no aprendizado, exige um compromisso genuíno da liderança e uma mudança abrangente no processo de entrega.

Através da experimentação contínua, da assunção calculada de riscos e da valorização do aprendizado com os erros, as organizações podem cultivar um ambiente propício à inovação, ao crescimento e ao sucesso. Ao criar um ambiente de alta confiança, onde os colaboradores se sintam seguros para assumir riscos, aprender com os erros e colaborar abertamente, fazendo com que as equipes de SRE possam alcançar resultados excepcionais.

A antifragilidade tem a ver com a compreensão da desordem e como usá-la a seu favor para ser mais resiliente, ao executar sistemas distribuídos em grande escala, compreender a desordem potencial que pode ocorrer, ajuda a projetar e a tornar esses sistemas muito mais robustos e resilientes. Para testar e compreender a desordem dos sistemas, as equipes de SRE utilizando de técnicas e ferramentas para testarem os seus ambientes e aplicações a fim de simular erros, estes serão abordados nos tópicos subsequentes.

9.1.1 Combate a Incêndios

As simulações de incêndio, embasadas nos conceitos de *Business Continuity Planning* (BCP) e *Disaster Recovery* (DR), se concentram em examinar o que acontece quando algo dá errado, testando os aspectos funcionais e não funcionais.

Essas simulações são ferramentas valiosas para testar a resiliência das organizações e aprimorar seus planos de resposta a emergências. Ao investir em simulações realistas e seguras, as organizações podem minimizar perdas, aprimorar seus planos de contingência e garantir a continuidade de seus negócios.

9.1.2 Chaos Engineering

Segundo Rosenthal e Jones (2020) a Engenharia do Caos, ou *Chaos Engineering*, “é a disciplina de experimentação em um sistema distribuído, a fim de construir confiança na capacidade do sistema de resistir a condições turbulentas na produção”. Criada pela Netflix, essa prática formalizada de engenharia, visa fortalecer a resiliência de sistemas de software e produção através da experimentação controlada e estratégica, se fundamentando na premissa de que, ao expor um sistema a falhas e eventos disruptivos de forma controlada, podemos identificar pontos fracos, aprimorar a capacidade de resposta a incidentes e construir sistemas mais robustos e adaptáveis.

A Netflix, pioneira no movimento da Engenharia do Caos, oferece um conjunto de ferramentas valiosas para testar a confiabilidade, segurança e resiliência da infraestrutura em ambientes de produção na AWS chamado de *Netflix Simian Army* (Bradley, 2014). Este nome é dado, devido as ferramentas proporcionadas pela empresa, como, o *Chaos Monkey* (simula a interrupção de uma instância), o *Chaos Gorilla* (simula uma interrupção de toda uma zona de disponibilidade) e o *Chaos Kong* (simula uma interrupção em toda uma região da AWS).

Ilustração 3 *- Netflix Simian Army*

Fonte: DevOps.com

A antifragilidade é uma ferramenta poderosa para a construção de sistemas distribuídos em grande escala mais robustos, adaptáveis e preparados para enfrentar os desafios do mundo digital em constante mudança. Ao compreender a desordem com a ajuda dessas ferramentas e utilizá-las a seu favor, as organizações podem fortalecer sua infraestrutura, garantir a continuidade das operações e alcançar o sucesso em um ambiente cada vez mais incerto e dinâmico.

9.2 O que é Postmortem?

Sabemos que as falhas em sistemas são inevitáveis, com isso, a questão que fica é "*como lidar com elas de forma eficaz?*". Sabemos que os fracassos são experiências estressantes e até mesmo assustadoras, com consequências que podem ser graves mas esse medo do fracasso gera um ambiente frágil, onde a experimentação e o aprendizado com erros são reprimidos, desta forma, ao invés de temer o fracasso, podemos transformá-lo em uma oportunidade de aprendizado e crescimento.

Sabemos que a falha é inevitável, por isso, é necessário prevenir a recorrência dos incidentes, reduzindo o impacto ou ao menos encurtar o tempo de resolução das falhas (Ruqayya, 2024), a questão mais importante para o princípio de *Postmortem* não é identificar culpados, mas sim aprender com os erros e implementar medidas para evitar que se repitam no futuro. No contexto do SRE, essa abordagem é concretizada através do conceito de "Postmortem sem culpa", um princípio fundamental da cultura SRE que é definido como, uma análise sistemática de um incidente ou falha, realizada sem a atribuição de culpa a qualquer indivíduo ou equipe.

O objetivo principal é identificar as causas raízes do problema, entendendo os fatores que contribuíram para a falha e implementando ações preventivas para evitar que situações semelhantes se repitam no futuro.

Este princípio pressupõe que todos os envolvidos em um incidente tiveram boas intenções e fizeram a coisa certa com as informações que possuíam, caso o incidente ocorra em um ambiente onde a cultura de culpa predomina, as pessoas tendem a esconder erros e problemas por medo de punição ou represálias. Essa atitude impede a identificação de falhas e a implementação de soluções eficazes, perpetuando problemas e comprometendo a confiabilidade dos sistemas.

10. APLICANDO O SRE NAS ORGANIZAÇÕES

10.1 Porque as organizações aplicam o SRE?

Para dar mais profundidade ao assunto abordado neste tópico, cabe ressaltar a citação de Gallupo (1995, p. 62, *apud* Gonçalves, 1998, p. 17)

> As organizações não podem impedir o mundo de mudar. O melhor que elas podem fazer é se adaptar. As mais espertas mudam antes de serem obrigadas a fazê-lo. Aquelas de sorte conseguem dar um jeito quando a pressão inevitável chega. As outras são as perdedoras e acabam virando história.

O cenário corporativo atual se caracteriza por um ritmo acelerado de mudanças, exigindo das empresas uma postura adaptável e proativa para garantir sua sobrevivência e prosperidade. Essa dinâmica, assemelha-se ao funcionamento de um organismo vivo, necessitando de constante movimento e renovação para se manter saudável e competitivo, por isso, as empresas reconhecem que a adoção da cultura SER, é a peça-chave que faltava para impulsionar a otimização do desempenho e da confiabilidade de seus sistemas.

A adoção do SRE pelas empresas é impulsionada por diversos fatores, entre os quais se destaca a busca pela otimização do custo total de propriedade (TCO) de seus softwares. Estimativas apontam que entre 40% e 90% dos custos totais de um software incidem na fase pós-produção, exigindo investimentos significativos em manutenções corretivas e preventivas. É nesse contexto, que a figura do Engenheiro de Confiabilidade de Sites se torna essencial para o sucesso das empresas, pois, com

a combinação de habilidades de desenvolvimento de software com a expertise em operações, o SRE assume um papel multifacetado que transcende a mera codificação de aplicações, abrangendo um espectro amplo de responsabilidades críticas para o sucesso organizacional.

10.2 Estrutura Organizacional

A crescente dependência das novas tecnologias impacta profundamente a sociedade, redefinindo relações e modelos de negócios, com isso surge a questão *“como preparar as organizações para essa jornada de mudança e adaptação?”*.

Para alcançar esse objetivo, é preciso estruturar processos e direcionar estratégias que facilitem a transição para o mundo digital, sendo necessário um conhecimento profundo da organização. Através de uma análise profunda, as empresas podem identificar seus pontos fortes e fracos, mapear oportunidades de crescimento e traçar um plano de ação estratégico para a transformação digital, entretanto, quando nos referimos ao setor de infraestrutura e direcionamos esta função diretamente para os SREs, a realização de uma análise de maturidade organizacional é a opção de ferramenta mais viável para identificar as oportunidades significativas de crescimento.

10.2.1 Maturidade Organizacional

Para descobrir o nível de aderência de uma empresa as boas práticas e princípios da cultura SRE, é preciso traçar um diagnóstico organizacional e para nortear essa investigação, o SRE propõe um sistema de escala de maturidade organizacional, onde para alcançar um resultado positivo no diagnóstico organizacional, é necessário realizar uma avaliação dos participantes, a fim de mapear suas áreas de atuação, habilidades técnicas (*hard skills*) e interpessoais (*soft skills*). Essa avaliação permite identificar os pontos fortes e fracos de cada indivíduo, bem como seu nível de maturidade profissional, contribuindo para a construção de um retrato fiel das competências presentes na equipe como um todo.

Ao mapear as áreas de atuação, as habilidades e a maturidade profissional dos participantes da avaliação, as empresas podem obter *insights* para aprimorar seus

processos, desenvolver seus colaboradores e desta forma, podendo definir seu nível de maturidade organizacional de maneira mais precisa.

A maturidade organizacional profissional é considerada baixa, quando a organização ainda não adotou, ou adotou poucos princípios e boas práticas da cultura SRE em seus processos. De maneira oposta, a maturidade organizacional profissional é considerada alta, quando a organização tiver pelo menos uma equipe SRE bem estabelecida, com os princípios e boas práticas do SRE sendo amplamente implementados naturalmente e sempre fazendo parte da cultura da empresa para que se mantenham em alto nível de aderência.

10.2.2 Familiaridade Organizacional

A familiaridade organizacional exerce um papel muito importante na jornada de aprendizado e desenvolvimento de um Engenheiro de Confiabilidade de Site. Essa familiaridade se traduz na abertura ou resistência que o profissional demonstra em relação a novos conhecimentos e na importância que ele atribui a diferentes conteúdos, por exemplo, os engenheiros mais experientes possuem uma vasta experiência de trabalho, seja na empresa atual ou em outras organizações. Eles dominam os princípios, práticas e cultura do SRE, demonstrando alto nível de familiaridade e interesse em aprofundar seus conhecimentos. Diferentemente dos novos graduados, pois, eles têm pouca ou nenhuma experiência prática com SRE, além dos conceitos da área não serem frequentemente ensinados nas faculdades. Eles demonstram grande entusiasmo em aprender e se desenvolver na área, mas ainda precisam de um direcionamento mais específico. Para auxiliar os novos SREs a se familiarizarem organizacionalmente, as empresas buscam promover diversos tipos de treinamentos, que podem variar a depender, do tamanho da organização, a velocidade com que a empresa se expande e a quantidade de recursos que a empresa precisa gastar para treinar os novos engenheiros.

10.3 Habilidades e treinamentos de SREs

Para aplicar o SRE de forma concisa nas organizações é importante falarmos a respeito de suas habilidades necessárias, que devem se estender além do conhecimento técnico fundamental. Dominar a infraestrutura e as técnicas de

resolução de problemas durante o plantão é importante, mas para um profissional ser excepcional, é necessário que ele explore áreas periféricas que complementam a expertise técnica.

Manter as habilidades de resolução de problemas afiadas é a chave para o sucesso de um SRE e para auxiliar nesse aspecto, é recomendado que esses profissionais participem de plantões regulares dos serviços, para fornecer uma familiaridade com as operações e responder prontamente a incidentes. No entanto, encontrar o equilíbrio ideal entre tempo de plantão e tempo livre, é essencial para evitar o esgotamento e a perda da expertise.

Para que essas pessoas operem com excelência, especialmente quando recém-integradas à organização, torna-se fundamental investir em treinamentos eficazes que transmitam as habilidades críticas para o sucesso. Felizmente, existe uma gama diversificada de técnicas de treinamento que podem ser utilizadas para desenvolver as habilidades críticas das equipes SRE. A escolha da técnica ideal depende de diversos fatores, como o nível de conhecimento da equipe, o tempo e os recursos disponíveis.

10.3.1 Sink or Swim

Ilustração 4 *- Sink or Swim*

Fonte: Karen Colligan

O método *Sink or Swim*, em português "Afundar ou Nadar", se posiciona na extremidade de baixo esforço do espectro das técnicas de treinamento. Bailey (2020) diz que essa abordagem, coloca a responsabilidade principal do aprendizado no aluno, exigindo que ele descubra as informações e soluções por conta própria, com pouca ou nenhuma orientação formal.

É importante usar este método com cautela e considerar cuidadosamente as necessidades e habilidades individuais dos alunos, fornecendo um ambiente de apoio com recursos e mentores disponíveis para auxiliar os alunos que precisarem de ajuda.

10.3.2 Autoexplicação

Ilustração 5 *- Autoexplicação*

Fonte: o Autor

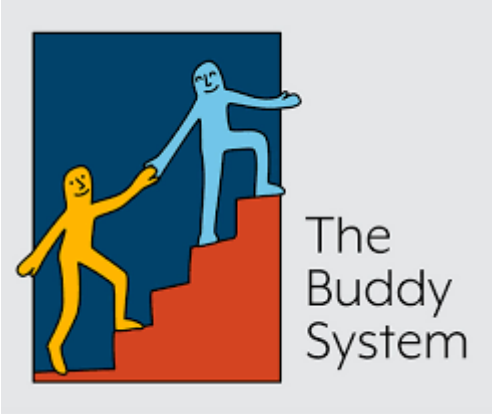
A técnica de Autoexplicação é apenas um degrau acima do *Sink or Swim* no espectro de técnicas de treinamento para profissionais. Este treinamento, como o próprio nome sugere, consiste em explicar algo para si mesmo, geralmente a partir de materiais de aprendizado depois faz-se anotações e explica o conteúdo para si mesmo (Santos, 2023).

O fornecimento de material de apoio e vídeos para auxiliar no aprendizado dos profissionais pode ser uma ferramenta valiosa para as organizações. Entretanto, há o risco de o material ficar desatualizado ou depreciado, isso pode ocorrer caso não haja curadoria regular.

O material de apoio e vídeos podem ser uma ferramenta valiosa para auxiliar no aprendizado dos profissionais, desde que sejam utilizados de forma estratégica e com cuidado para evitar os riscos associados à desatualização e à falta de curadoria. Ao combinar o material online com o treinamento presencial e oportunidades de interação, as organizações podem criar um ambiente de aprendizado eficaz e engajador para seus colaboradores.

10.3.3 Buddy System

Ilustração 6 - *The Buddy System*



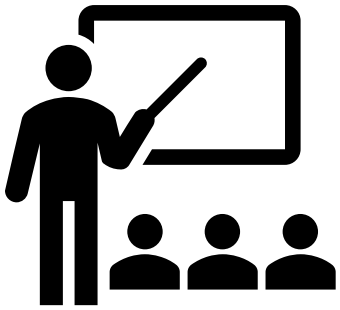
Fonte: Macalester College

A técnica conhecida como Sistema de Camaradagem é o nível acima da Autoexplicação no espectro de técnicas de treinamento para profissionais. Este treinamento fornece um amigo, colega ou aliado para um funcionário, a fim de manter os funcionários em boas condições durante tarefas desafiadoras, com o objetivo de fomentar a camaradagem em momentos de vitórias, derrotas e reveses, aproximando o time cada vez mais (Santos, 2022).

O treinamento de profissionais, especialmente para novos membros da equipe, pode ser significativamente aprimorado através da implementação de estratégias que combinem orientação individualizada, materiais de autoaprendizado de alta qualidade e um ambiente de camaradagem colaborativa. Esses elementos auxiliam os novos profissionais a terem confiança nos materiais e nos treinamentos, além de promover a confiança entre todos os integrantes do grupo.

10.3.4 Programas de Treinamento

Ilustração 7 *- Programas de Treinamento*



Fonte: o Autor

Para organizações de grande porte ou em rápido crescimento, investir em um programa de treinamento é viável para o desenvolvimento de seus profissionais. Esse tipo de treinamento auxilia os funcionários a receberem os conhecimentos e habilidades específicas, geralmente inerentes ao seu cargo atual, ou a uma futura promoção (Kolinski, 2021), sendo um treinamento importante para empresas que estão migrando para uma cultura nova, pois a mudança deve ser gradual, algo inviável em treinamentos como a Autoexplicação.

Adotar um programa de treinamento permite às empresas moldarem profissionais com mais facilidade, pois, colocando-os no mesmo círculo de convivência, faz com que a chance dos colaboradores se sentirem sozinhos ao enfrentar divergências culturais durante o processo seja mínima, além de lutar contra a síndrome do impostor e reforçar o laço de confiança entre aqueles profissionais que já atuam em conjunto.

11. CONCLUSÃO

A adoção da cultura do *Site Reliability Engineering* em uma organização não é uma tarefa trivial e como acontece com diversas decisões estratégicas, a resposta para a pergunta “*Como implementar o SRE?*” é complexa e depende de diversos fatores específicos de cada empresa.

Antes de embarcar na jornada de adoção do SRE, é fundamental que a organização estabeleça uma compreensão clara e alinhada do que o SRE significa e como ele se integrará à cultura e aos objetivos da empresa. Essa clareza é essencial para garantir o "*buy-in*" da empresa, ou seja, a adesão e o apoio de todos os *stakeholders* envolvidos, desde a alta liderança até a equipe de desenvolvimento. O SRE não se resume a um conjunto de práticas pré-definidas, ele representa uma filosofia que deve permear toda a organização, impactando a forma como os sistemas são projetados, desenvolvidos, operados e monitorados. É crucial que a empresa defina o que o SRE significa em seu contexto específico, considerando suas necessidades, desafios e cultura.

O ponto de partida para uma implementação bem-sucedida do SRE é a definição clara e precisa do que essa metodologia significa para a empresa em questão. Isso envolve, a compreensão profunda das especificidades do negócio, incluindo os objetivos de confiabilidade, disponibilidade, escalabilidade e eficiência, além do apoio da liderança da empresa, para que seja adotada de forma ampla, não apenas pelas equipes de SRE.

Com a visão clara do que o SRE significa para a empresa, chega o momento de traçar um plano de implementação detalhado, garantindo que essa jornada seja sólida e eficaz, levando em consideração três aspectos.

O primeiro é a *Estrutura Organizacional*, uma vez que a efetividade do SRE depende em grande parte da estrutura organizacional que o sustenta. Para garantir uma abordagem holística à confiabilidade, a equipe de SRE deve estar integrada às equipes de Engenharia e Operações, promovendo colaboração e sinergia entre os processos de desenvolvimento, operação e monitoramento de sistemas.

O segundo aspecto são os *Processos e Ferramentas*, pois, o SRE exige uma infraestrutura robusta de processos e ferramentas para garantir a eficiência e a efetividade das atividades da equipe. A implementação de ferramentas adequadas e

a definição de processos claros são essenciais para o sucesso do SRE, permitindo que a equipe monitore, gerencie incidentes e automatize tarefas de forma eficaz.

O terceiro e último aspecto é a *Capacitação*, tendo em vista que as equipes de SRE precisam de um conjunto abrangente de habilidades para garantir o sucesso em suas funções. Através de um programa de treinamento robusto, os membros da equipe podem desenvolver as competências essenciais em Engenharia de Software, Operação de Sistemas e Análise de Dados.

Assim, podemos concluir que a implementação do SRE não é um evento único, mas sim um processo em constante evolução e para garantir o sucesso a longo prazo, é preciso monitorar e avaliar continuamente as práticas e fazer ajustes conforme necessário. Essa abordagem proativa garante que o SRE permaneça relevante, eficaz e adaptado às necessidades dinâmicas de cada organização.

REFERÊNCIAS

ATLASSIAN. **O que é orçamento de erros — e por que ele é importante?**. [S. l.]. Disponível em: https://www.atlassian.com/br/incident-management/kpis/error-budget. Acesso em: 28 abr. 2024.

ATLASSIAN. **SLA vs. SLO vs. SLI: qual a diferença?**. [*S. l.*]. Disponível em: https://www.atlassian.com/br/incident-management/kpis/sla-vs-slo-vs-sli. Acesso em: 19 abr. 2024.

BAILEY, Ian. **Sink or Swim Method of Learning**. [*S. l.*], 26 jun. 2020. Disponível em: https://www.linkedin.com/pulse/sink-swim-method-learning-ian-bailey. Acesso em: 16 maio 2024.

BEYER, Betsy *et al*. **Site Reliability Engineering**: How Google Runs Production Systems. 1. ed. [*S. l.*]: O'Reilly Media, 2016. 550 p. ISBN 978-1491929124.

BEYER, Betsy *et al*. **The Site Reliability Workbook**: Practical Ways to Implement SRE. 1. ed. [*S. l.*]: O'Reilly Media, 2018. 506 p. ISBN 978-1492029502

BLAMELESS. **Determining Error Budgets and Policies that Work for Your Team**. [*S. l.*], 2 set. 2020. Disponível em: https://www.blameless.com/blog/determining-error-budgets-and-policies. Acesso em: 28 abr. 2024.

BRADLEY, Tony. **Netflix, the Simian Army, and the culture of freedom and responsibility**. [*S. l.*], 27 mar. 2014. Disponível em: https://devops.com/netflix-the-simian-army-and-the-culture-of-freedom-and-responsibility. Acesso em: 6 maio 2024.

CLIMENT, Jesus. **How maintenance windows affect your error budget — SRE tips**. [*S. l.*], 22 jun. 2020. Disponível em: https://cloud.google.com/blog/products/management-tools/sre-error-budgets-and-maintenance-windows. Acesso em: 20 abr. 2024.

COLLIGAN, Karen. **Sink or Swim is Not Effective Leadership Development**. [*S. l.*]. Disponível em: https://www.peoplethink.biz/sink-or-swim-is-not-effective-leadership-development/. Acesso em: 18 maio 2024.

GONÇALVES, José Ernesto Lima. A Necessidade de Reinventar as Empresas. **RAE - Revista de Administração de Empresas**, São Paulo, v. 38, n. 2, p. 6-17, abr./jun. 1998. Disponível em: https://www.scielo.br/j/rae/a/SNkw4mmTWVywRMfFmkPW3TH/?format=pdf. Acesso em: 09 maio 2024.

IBM. **O que é a engenharia de confiabilidade de sites (SRE)?**. [*S. l.*]. Disponível em: https://www.ibm.com/br-pt/topics/site-reliability-engineering. Acesso em: 25 mar. 2024.

IBM. **O que é observabilidade e por que é importante?**. [*S. l.*], 21 fev. 2022. Disponível em: https://www.ibm.com/br-pt/resources/automate/observability-basics. Acesso em: 28 abr. 2024.

KIDD, Chrissy. **Monitoring vs Observability vs Telemetry**: What's The Difference?. [*S. l.*], 1 mar. 2023. Disponível em: https://www.splunk.com/en_us/blog/learn/observability-vs-monitoring-vs-telemetry.html. Acesso em: 3 maio 2024.

KIM, Gene *et al*. **Manual de DevOps**: Como Obter Agilidade, Confiabilidade e Segurança em Organizações Tecnológicas. 1. ed. [*S. l.*]: Alta Books, 2018. 464 p. ISBN 978-8550802695.

KOLINSKI, Helga. **Como criar um programa de treinamento de funcionários bem-sucedido**. [*S. l.*], 5 set. 2023. Disponível em: https://www.ispringpro.com.br/blog/como-elaborar-um-treinamento. Acesso em: 18 maio 2024.

MACALESTER COLLEGE. **Buddy Up**. [*S. l.*], 21 out. 2020. Disponível em: https://www.macalester.edu/news/2020/10/buddy-up/. Acesso em: 18 maio 2024.

MYKHALCHUK, Oleksandr. **Assess DevOps Structure Through CALMS**. [*S. l.*], 27 mar. 2019. Disponível em: https://www.softserveinc.com/en-us/blog/assess-devops-structure-through-calms. Acesso em: 16 abr. 2024.

RAVICHANDRAN, Aruna; TAYLOR, Kieran; WATERHOUSE, Peter. **DevOps for Digital Leaders**: Reignite Business with a Modern DevOps-Enabled Software Factory. 1. ed. [*S. l.*]: Apress, 2016. 188 p. ISBN 978-1484218419.

RED HAT. **O que é SRE (engenharia de confiabilidade de sites)?**. [*S. l.*], 18 jan. 2024. Disponível em: https://www.redhat.com/pt-br/topics/devops/what-is-sre. Acesso em: 25 mar. 2024.

ROSENTHAL, Casey; JONES, Nora. **Chaos Engineering**: System Resiliency in Practice. 1. ed. [*S. l.*]: O'Reilly Media, 2020. 305 p. ISBN 978-1492043867.

RUQAYYA, Noor-ul-Anam. **What are Blameless Retrospectives? How Do You Run Them?**. [*S. l.*], 29 mar. 2024. Disponível em: https://www.blameless.com/blog/what-are-blameless-postmortems-do-they-work-how. Acesso em: 7 maio 2024.

SANTOS, Adriano da Silva. **Camaradagem nas empresas como indicador**. [*S. l.*], 19 out. 2022. Disponível em: https://rhpravoce.com.br/colab/camaradagem-nas-empresas-como-indicador/. Acesso em: 18 maio 2024.

SANTOS, Tatiana. **Autoexplicação**: como aplicar nos estudos para concurso?. [*S. l.*], 6 jun. 2023. Disponível em: https://blog.grancursosonline.com.br/autoexplicacao-nos-estudos-para-concurso/. Acesso em: 17 maio 2024.

TALEB, Nassim Nicholas. **Antifrágil**: Coisas que se beneficiam com o caos. 1. ed. [*S. l.*]: Objetiva, 2020. 616 p. ISBN 978-8547001087.

TANG, JJ. **Observability vs Monitoring**: What‘s The Difference?. [*S. l.*], 27 out. 2021. Disponível em: https://devops.com/observability-vs-monitoring-whats-the-difference/. Acesso em: 3 maio 2024.

TEBALDI, Pedro César. **Telemetria**: o que é e como funciona?. [*S. l.*], 5 dez. 2019. Disponível em: https://www.opservices.com.br/telemetria/. Acesso em: 30 abr. 2024.