

Impacto da Adoção de DevOps na Qualidade do Software

Ana Luiza Rezende¹, Gustavo C. Gabrich¹, Luidi C. Silva¹

¹Departamento de Engenharia de Software
Pontifícia Universidade Católica de Minas Gerais (PUC-Minas)
Caixa Postal 1.686 – 30.535-901 – Belo Horizonte – MG – Brazil

ana.rezende.1452875@sga.pucminas.br

gcgabrich@sga.pucminas.br, luidi.cadete@sga.pucminas.br

Abstract. *This article deals with the area of Software Engineering focused on DevOps and its influence on software quality. The problem addressed is whether the adoption of DevOps results in a significant improvement in software quality, measured by details such as bug reduction, increased user satisfaction, and compliance with requirements. Furthermore, it seeks to understand how the integration between development and operations teams, promoted by DevOps, reduces communication bottlenecks and delays in software delivery, and whether process automation contributes to more efficient and reliable continuous delivery. The relevance of solving this problem is to understand whether the DevOps approach really contributes to improving software quality, helping organizations decide on the implementation of these practices. The general objective is to evaluate ways of adopting DevOps and their impact on software quality. Specifically, the work investigates the influence of DevOps practices on software quality, examines the integration between development and operations teams, and identifies automated processes and their impact on the efficiency and reliability of continuous software delivery.*

Resumo. *Este artigo trata da área de Engenharia de Software focada em DevOps e sua influência na qualidade do software. O problema abordado é se a adoção de DevOps resulta em uma melhoria significativa na qualidade do software, medida por métricas como redução de bugs, aumento da satisfação do usuário e conformidade com os requisitos. Além disso, busca entender como a integração entre as equipes de desenvolvimento e operações, promovida pelo DevOps, reduz os gargalos de comunicação e os atrasos na entrega de software, e se a automação de processos contribui para uma entrega contínua mais eficiente e confiável. A relevância de resolver este problema está em compreender se a abordagem DevOps realmente contribui para a melhoria da qualidade do software, auxiliando organizações a decidirem sobre a implementação dessas práticas. O objetivo geral é avaliar formas de adoção de DevOps e seu impacto na qualidade do software. Especificamente, o trabalho investiga a influência das práticas DevOps na qualidade do software, examina a integração entre equipes de desenvolvimento e operações e identifica os processos automatizados e seu impacto na eficiência e confiabilidade da entrega contínua de software.*

1. Introdução

A Engenharia de Software é uma área ampla e multifacetada que abrange diversas abordagens e metodologias para melhorar a eficiência e a qualidade dos processos de desenvolvimento de software. Uma dessas abordagens, que tem ganhado destaque significativo nos últimos anos, é o DevOps. Este trabalho trata especificamente do DevOps e sua influência na qualidade do software, investigando como sua adoção pode impactar diversas métricas de desempenho e satisfação.

O problema central abordado neste estudo está relacionado à eficácia do DevOps na melhoria da qualidade do software. As questões de pesquisa que guiam este trabalho são: a adoção de DevOps resulta em uma melhoria significativa na qualidade do software, medida por métricas como redução de bugs, aumento da satisfação do usuário e conformidade com os requisitos? Em que medida a integração entre as equipes de desenvolvimento e operações, promovida pelo DevOps, reduz os gargalos de comunicação e os atrasos na entrega de software? A automação de processos no DevOps contribui para uma entrega contínua mais eficiente e confiável de software de alta qualidade? Estas perguntas são cruciais para entender o real impacto do DevOps em ambientes de desenvolvimento de software.

A relevância de resolver este problema está fundamentada na necessidade de avaliação concreta do impacto da adoção de práticas DevOps. Compreender se o DevOps realmente contribui para a melhoria da qualidade do software pode auxiliar organizações a decidirem sobre a implementação dessas práticas. A redução de bugs, o aumento da satisfação do usuário e a melhoria na conformidade com os requisitos são benefícios potenciais que, se comprovados, podem justificar o investimento em DevOps. Além disso, a identificação de benefícios tangíveis pode facilitar a adoção de DevOps por organizações relutantes ou desconhecedoras das vantagens dessa abordagem.

O objetivo geral deste trabalho é avaliar as formas de adoção de DevOps por equipes de desenvolvimento de software e seu impacto na qualidade do software. Esta avaliação será realizada por meio de uma análise detalhada de diferentes métricas e percepções relacionadas à qualidade do software, sendo explicadas de maneira mais abrangente ao decorrer do trabalho.

Para atingir este objetivo geral, três objetivos específicos foram delineados: (1) Investigar se a adoção de práticas DevOps influencia significativamente a qualidade do software, através da avaliação de métricas como a redução de bugs, a percepção dos usuários em relação à qualidade do software e a conformidade do software com os requisitos antes e após a implementação do DevOps; (2) Examinar como a integração entre as equipes de desenvolvimento e operações, promovida pelo DevOps, afeta os gargalos de comunicação, identificando os principais gargalos antes da implementação e avaliando a percepção das equipes sobre a comunicação e colaboração após a implementação do DevOps; (3) Identificar os processos que foram automatizados como parte da implementação do DevOps e analisar o impacto dessa automação na eficiência e confiabilidade da entrega contínua de software de alta qualidade.

Este estudo se propõe a fornecer uma compreensão aprofundada dos benefícios e desafios associados à adoção de DevOps. Através de uma análise rigorosa e sistemática, espera-se contribuir com evidências empíricas sobre a eficácia do DevOps, oferecendo

insights valiosos para pesquisadores e profissionais da área de Engenharia de Software. A investigação detalhada dos objetivos específicos permitirá uma visão abrangente dos aspectos críticos envolvidos, desde a qualidade do software até a eficiência dos processos e a integração entre equipes.

2. Fundamentação Teórica

No cerne deste trabalho reside o conceito de DevOps, uma filosofia de desenvolvimento de software que revolucionou a maneira como as empresas constroem, lançam e gerenciam seus produtos digitais. Em sua essência, o DevOps propõe a integração e colaboração contínuas entre as equipes de desenvolvimento e operações, eliminando os silos tradicionais que frequentemente impedem a agilidade e a qualidade.

2.1. Práticas Fundamentais DevOps

Essa união de forças é viabilizada por um conjunto de práticas fundamentais, cada uma desempenhando um papel crucial na orquestração de um ciclo de desenvolvimento de software otimizado. Dentre essas práticas podemos citar a integração contínua (A automação da integração de código frequente no repositório central, permitindo a detecção precoce de erros e a rápida correção de problemas.), a entrega contínua (A automatização do processo de entrega de software para produção, garantindo lançamentos frequentes e confiáveis de novas funcionalidades e correções.), o monitoramento e observabilidade (A coleta e análise contínuas de dados de desempenho e logs do sistema, fornecendo insights valiosos para identificar e resolver problemas rapidamente.) a infraestrutura como código (O tratamento da infraestrutura como código, permitindo provisionamento, configuração e gerenciamento automatizados, garantindo ambientes consistentes e escaláveis.), a automação de testes (A criação e execução automatizadas de testes, otimizando o processo de validação de software e liberando tempo para atividades de maior valor.) e por fim a cultura colaborativa (O fomento de uma cultura de colaboração, comunicação aberta e compartilhamento de conhecimento entre as equipes, quebrando barreiras e promovendo a resolução conjunta de problemas.).

2.1.1. Primeira obra literária

O conceito/teoria principal associado a este trabalho é DevOps e suas práticas. A sua definição neste trabalho é "uma combinação de práticas e ferramentas de engenharia de software que visa aumentar a capacidade de uma organização de entregar aplicações e serviços em alta velocidade, melhorando e evoluindo produtos em um ritmo mais rápido do que as organizações que utilizam processos tradicionais de desenvolvimento de software e gestão de infraestrutura, conforme as práticas citadas anteriormente e definido na obra literária científica "The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations" pelos autores Gene Kim, Jez Humble e Forrest North (2016).

2.1.2. Segunda obra literária

No contexto deste trabalho, a qualidade de software se manifesta em três pilares fundamentais: a redução de bugs, que seria a minimização de falhas e defeitos no software,

garantindo uma experiência fluida e livre de frustrações para os usuários, a satisfação do usuário, sendo a priorização das necessidades e expectativas dos usuários, garantindo que o software atenda aos seus objetivos e proporcione valor real e a conformidade com requisitos, sendo esse o cumprimento rigoroso de todos os requisitos técnicos, funcionais e não funcionais definidos para o software, garantindo sua confiabilidade e robustez.

Pode-se ver muito desses três pilares no conceito/teoria secundário associado a este trabalho, no caso a qualidade do software. A sua definição neste trabalho é "a medida em que um software atende aos requisitos funcionais e não funcionais, tendo uma ligação direta com os pilares descritos" conforme definido em outra obra científica literária "Software Quality: Concepts and Practice" pelo autor Daniel Galin (2018).

3. Trabalhos Relacionados

O trabalho mais relacionado é "DevOps-RAF: An assessment framework to measure DevOps readiness in software organizations" (<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9650363>) , publicado em 2021 pelos autores Lilianny Marrero e Hernán Astudillo. Este estudo se concentra em desenvolver uma estrutura de avaliação para medir a prontidão do DevOps em organizações de software. A importância deste trabalho reside em fornecer uma ferramenta que permite às empresas avaliar seu nível de preparação para implementar práticas DevOps, identificando áreas de melhoria e destacando os benefícios potenciais de uma adoção mais completa dessa abordagem.

O segundo trabalho mais relevante é "Expanding DevOps Principles and Best Practices Based on Practical View" (<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9994216>), publicado em 2022 pelos autores Muntaha Alawneh e Imad M. Abbadi. Este estudo aborda a expansão dos princípios e das melhores práticas do DevOps com base em uma perspectiva prática. Ao focar em como os princípios do DevOps podem ser melhorados e aplicados de forma eficaz, este trabalho oferece insights valiosos para profissionais que buscam otimizar suas práticas de desenvolvimento e operações, garantindo uma integração mais harmoniosa entre essas duas áreas cruciais.

O terceiro trabalho mais relacionado é "A Study of Adoption and Effects of DevOps Practices" (<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10033313>), publicado em 2022 pelos autores Tyron Offerman, Robert Blinde, Christoph Johann Stettina e Joost Visser. Este estudo investiga a adoção e os efeitos das práticas DevOps em várias organizações. Através da análise de dados empíricos, os autores exploram como a implementação de práticas DevOps impacta a eficiência operacional, a qualidade do software e a satisfação da equipe. Este trabalho é fundamental para entender os desafios e os benefícios reais enfrentados por empresas que adotam essa metodologia.

Esses três estudos fornecem uma visão abrangente das práticas e dos desafios associados à implementação do DevOps. O primeiro estudo oferece uma ferramenta para avaliação da prontidão, essencial para qualquer organização que deseja iniciar sua jornada DevOps. O segundo estudo se aprofunda na aplicação prática dos princípios DevOps, propondo melhorias baseadas em experiências reais. O terceiro estudo complementa os dois anteriores ao apresentar evidências empíricas sobre os efeitos da adoção das práticas DevOps.

Em conjunto, esses trabalhos destacam a importância de uma abordagem estruturada e bem informada para a adoção do DevOps, mostrando que a implementação bem-sucedida não apenas melhora a eficiência e a qualidade do desenvolvimento de software, mas também contribui para um ambiente de trabalho mais colaborativo e satisfatório. Eles também sublinham a necessidade contínua de pesquisa e desenvolvimento de ferramentas e práticas que suportem a evolução contínua da integração de desenvolvimento e operações.

4. Metodologia

Este estudo será conduzido utilizando uma abordagem quantitativa para avaliar o impacto da adoção de práticas DevOps na qualidade do software. A pesquisa será realizada por meio de uma coleta de dados sistemática e análise estatística das informações obtidas, visando fornecer evidências concretas sobre os efeitos do DevOps em diferentes métricas de desempenho e satisfação.

5. Materiais e Métodos

Este estudo adota uma abordagem quantitativa e descritiva. A pesquisa é quantitativa porque envolve a coleta e análise de dados numéricos relacionados à qualidade do software e à eficiência das práticas DevOps. É descritiva porque busca descrever como a adoção de práticas DevOps influencia na qualidade do software e na eficiência das equipes de desenvolvimento e operações, podendo ser tanto de maneira positiva quanto negativa. Para os Repositórios que Usam DevOps, há presença de arquivos de configuração para CI/CD (e.g., `.travis.yml`, `Jenkinsfile`), uso de scripts de automação de deploy, menções a práticas DevOps na documentação e presença de badges de CI/CD. Já nos Repositórios que Não Usam DevOps, há ausência de arquivos de configuração de CI/CD, indicações de deploys manuais na documentação ou histórico de commits, uso de métodos tradicionais de comunicação de problemas e pouca ou nenhuma evidência de automação de processos.

Os materiais utilizados neste trabalho são os dados de commits, bugs e releases serão coletados utilizando a API do GitHub (<https://docs.github.com/en/rest?apiVersion=2022-11-28>), que fornece informações detalhadas sobre o histórico de desenvolvimento e manutenção dos projetos. Além disso, informações sobre a satisfação do usuário serão obtidas através de análises de reviews e ratings disponíveis nas plataformas de distribuição de software relevantes, como no GitHub (para projetos open source).

Os métodos empregados nesta análise incluirão várias etapas. Os métodos de análise incluirão várias etapas. Primeiramente, os dados serão coletados utilizando a API do GitHub para obter informações sobre commits, bugs e releases de 10 projetos selecionados, sendo 5 com a adoção de práticas DevOps e 5 sem essa adoção, além de avaliações e feedbacks de usuários das plataformas de distribuição de software. Em seguida, as métricas serão calculadas. Para a redução de bugs, será contado o número de bugs reportados e corrigidos em períodos de tempo definidos. A satisfação do usuário será analisada e quantificada com base nas avaliações e feedbacks dos usuários. A conformidade com os requisitos será avaliada a partir da documentação e relatórios de conformidade dos projetos. A comunicação entre equipes será analisada por meio de logs de comunicação e reuniões, e os atrasos na entrega serão calculados como o tempo médio

entre commits e releases. A eficiência da entrega contínua será medida pela frequência e sucesso das entregas contínuas, e a confiabilidade será avaliada contando o número de falhas ou incidentes pós-deployment. As análises estatísticas incluirão o uso de gráficos para medir a relação entre a adoção de DevOps e as métricas de qualidade do software entre os projetos que adotaram DevOps e os que não adotaram. Análises de regressão serão empregadas para entender o impacto específico de cada prática DevOps (integração de equipes e automação de processos) nas métricas de qualidade do software.

As métricas utilizadas para avaliar a qualidade do software e a eficiência das práticas DevOps são divididas em três categorias principais: qualidade do software, integração entre equipes e automação de processos. Para a qualidade do software, as métricas incluem a redução de bugs (número de bugs reportados e corrigidos ao longo do tempo), a satisfação do usuário (avaliações e feedbacks dos usuários) e a conformidade com os requisitos (avaliação do atendimento aos requisitos funcionais e não funcionais do software). Para a integração entre equipes, as métricas incluem a comunicação (número e frequência de interações entre as equipes de desenvolvimento) e os atrasos na entrega (tempo médio entre commits e releases). Para a automação de processos, as métricas incluem a eficiência da entrega contínua (frequência e sucesso das entregas contínuas) e a confiabilidade (falhas ou incidentes pós-deployment). Com relação à identificação dos gargalos, coletamos os dados na seleção de repositórios, como citado anteriormente, e sobre a identificação usaremos os critérios de frequência e volume de commits: Analisar a frequência de commits para identificar padrões de colaboração e possíveis gargalos e por fim comentários em pull requests e issues: Avaliar a quantidade e natureza dos comentários para detectar problemas de comunicação.

A forma de amostragem envolverá a seleção de projetos que adotaram práticas DevOps e projetos que não adotaram, para fins de comparação. A seleção incluirá projetos de diferentes domínios e tamanhos para garantir a representatividade. A análise comparativa entre esses dois grupos permitirá avaliar o impacto específico das práticas DevOps nas métricas de qualidade do software.

5.1. Análise Comparativa

Os projetos selecionados para o estudo serão comparados com base em várias métricas de qualidade e eficiência:

1. Automação com Práticas DevOps:

- **Projetos com DevOps:** Os projetos que adotaram práticas DevOps mostrarão maior automação em seus processos de CI/CD, com menos necessidade de intervenção manual e maior frequência de deploys bem-sucedidos. Isso resulta em uma menor probabilidade de erros humanos durante os processos de deployment e rollback.
- **Projetos sem DevOps:** Projetos sem DevOps tendem a ter processos de deployment manuais, maior incidência de erros humanos e menor frequência de deploys. A falta de automação pode resultar em tempos de deploy mais longos e maior dificuldade em manter a consistência entre os ambientes de desenvolvimento, teste e produção.

2. Comunicação entre Equipes:

- **Projetos com DevOps:** A comunicação entre as equipes de desenvolvimento e operações é facilitada por ferramentas de colaboração e integração contínua, resultando em menos gargalos e maior eficiência na resolução de problemas. Logs de comunicação, frequências de reuniões e plataformas de mensagens instantâneas serão analisados para entender como a colaboração evoluiu após a implementação do DevOps.
- **Projetos sem DevOps:** A falta de integração e automação pode resultar em comunicação fragmentada e atrasos na resolução de problemas, devido à dependência de métodos tradicionais de comunicação. Isso pode ser evidenciado por uma maior quantidade de discussões em threads de emails e tickets de suporte que não são rapidamente resolvidos.

3. Velocidade de Entrega e Cumprimento dos Requisitos:

- **Projetos com DevOps:** Tendem a apresentar ciclos de desenvolvimento mais curtos e uma entrega contínua mais eficiente, cumprindo os requisitos de maneira mais rápida e confiável. A frequência de releases e o tempo médio para entrega de novas funcionalidades serão utilizados como indicadores.
- **Projetos sem DevOps:** Apresentam ciclos de desenvolvimento mais longos e maior probabilidade de não cumprimento dos prazos e requisitos devido à falta de automação e integração eficaz. Comparações serão feitas com base no número de funcionalidades entregues por unidade de tempo e no tempo médio entre releases.

5.1.1. Comparação de Resultados

Os resultados das análises quantitativas e qualitativas serão comparados para avaliar as diferenças entre os projetos que utilizam práticas DevOps e aqueles que não utilizam. As principais métricas de comparação incluirão:

1. Qualidade do Software:

- **Projetos com DevOps:** Espera-se observar uma redução significativa no número de bugs e falhas pós-deployment, bem como uma maior conformidade com os requisitos funcionais e não funcionais.
- **Projetos sem DevOps:** Podem apresentar maior número de bugs e falhas, além de dificuldades em cumprir todos os requisitos estabelecidos.

2. Eficiência e Velocidade de Entrega:

- **Projetos com DevOps:** Devem mostrar ciclos de entrega mais rápidos e uma maior frequência de releases bem-sucedidos.
- **Projetos sem DevOps:** Tendem a ter ciclos de entrega mais longos e menos frequentes, com maiores intervalos entre releases.

3. Satisfação do Usuário:

- **Projetos com DevOps:** A satisfação do usuário tende a ser maior devido à melhoria na qualidade do software e na experiência geral de uso, além do tempo de entrega.
- **Projetos sem DevOps:** A satisfação do usuário tende a ser menor, refletindo problemas de bugs, performance e usabilidade.

5.2. Etapas de Execução

As etapas de execução do trabalho incluem:

1. Coletar os dados dos projetos no GitHub e avaliações dos usuários nas plataformas de distribuição de software;
2. Calcular as métricas de qualidade do software, integração entre equipes e automação de processos;
3. Alisar a relação entre a adoção de DevOps e as métricas de qualidade do software utilizando gráficos, testes de comparação de amostras independentes e análises de regressão;
4. Interpretar os resultados para responder às questões de pesquisa e atingir os objetivos gerais e específicos do estudo.

6. Refêrencias

[Kim et al. 2016], [Galin 2018], [Offerman et al. 2022], [Alawneh and Abbadi 2022], [Marrero and Astudillo 2021], [Jenkins Community 2023], [Vijay 2023], [Community 2023], [Cid 2023], [Chan 2023], [Nguyen 2023], [Lissner 2023], [Wildrider09 2023], [Ngugi 2023], [Org 2023].

References

- Alawneh, M. and Abbadi, I. M. (2022). Expanding devops principles and best practices based on practical view. *IEEE*. Accessed: 2023-06-16.
- Chan, P.-F. (2023). Duframework. GitHub repository. Accessed: 2023-06-16.
- Cid, A. A. (2023). poppler_mirror. GitHub repository. Accessed: 2023-06-16.
- Community, T. C. (2023). travis-build. GitHub repository. Accessed: 2023-06-16.
- Galin, D. (2018). *Software Quality: Concepts and Practice*. Wiley, 1st edition.
- Jenkins Community (2023). jenkinsfile-runner. GitHub repository. Accessed: 2023-06-16.
- Kim, G., Humble, J., and North, F. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. IT Revolution Press, 1st edition.
- Lissner, S. (2023). meteor-deploy. GitHub repository. Accessed: 2023-06-16.
- Marrero, L. and Astudillo, H. (2021). Devops-raf: An assessment framework to measure devops readiness in software organizations. *IEEE*. Accessed: 2023-06-16.
- Ngugi, A. (2023). Ict-assets-management-system-iams. GitHub repository. Accessed: 2023-06-16.
- Nguyen, K. (2023). Jupyterhub-on-kubernetes. GitHub repository. Accessed: 2023-06-16.
- Offerman, T., Blinde, R., Stettina, C. J., and Visser, J. (2022). A study of adoption and effects of devops practices. *IEEE*. Accessed: 2023-06-16.
- Org, E. (2023). web-deploy. GitHub repository. Accessed: 2023-06-16.
- Vijay, D. (2023). jenkinsfiles. GitHub repository. Accessed: 2023-06-16.
- Wildrider09 (2023). Devops_calculator. GitHub repository. Accessed: 2023-06-16.