

Scrumops: a sinergia entre Scrum e devops na entrega contínua de valor

Scrumops: a sinergia entre Scrum e devops na entrega contínua de valor

ARK: 44123/multi.v8i13.1574

Henderson Matsuura Sanches

 <https://orcid.org/0000-0003-2354-3393>

 <http://lattes.cnpq.br/8112131443508423>

UniProcessus – Centro Universitário Processus, DF, Brasil

E-mail: xhenderson.sanches@uniprocessus.edu.br

Josyane Lannes Florenzano de Souza

 <https://lattes.cnpq.br/0278395131866960>

UniProcessus – Centro Universitário Processus, DF, Brasil

E-mail: josyane.souza@uniprocessus.edu.br

Resumo:

Este artigo explora a integração entre o *framework* Scrum e as práticas DevOps, resultando no modelo conhecido como ScrumOps. Analisa-se a evolução das metodologias ágeis, o surgimento do DevOps como um complemento essencial e a forma como a sinergia entre ambos otimiza a entrega contínua de valor no desenvolvimento de software. Serão abordados os pilares do Scrum, a cultura CAMS do DevOps e os benefícios dessa união para a redução de silos organizacionais, melhoria da qualidade e segurança, e o encurtamento dos ciclos de feedback.

Palavras-chave: ScrumOps; Scrum; DevOps; Engenharia de Software; Agilidade.

Abstract:

This article explores the integration between the Scrum framework and DevOps practices, resulting in the model known as ScrumOps. It analyzes the evolution of agile methodologies, the emergence of DevOps as an essential complement, and how the synergy between both optimizes the continuous delivery of value in software development. The pillars of Scrum, the CAMS culture of DevOps, and the benefits of this union for reducing organizational silos, improving quality and security, and shortening feedback loops will be addressed.

Keywords: ScrumOps; Scrum; DevOps; Software Engineering; Agility.

1. Introdução

A evolução do desenvolvimento de *software* tem sido marcada por uma busca incessante por maior eficiência, qualidade e capacidade de resposta às demandas do mercado. Nesse contexto, as metodologias ágeis emergiram como um paradigma transformador, priorizando a colaboração, a adaptabilidade e a entrega iterativa de valor. Dentre elas, o Scrum se destacou como um *framework* amplamente adotado para a gestão

de projetos complexos, enfatizando a transparência, a inspeção e a adaptação em ciclos curtos de trabalho, conhecidos como *Sprints*.

Paralelamente à ascensão do Agile, o cenário da tecnologia da informação testemunhou o surgimento do DevOps, uma cultura e um conjunto de práticas que visam unificar o desenvolvimento de software (Dev) e as operações de TI (Ops). O DevOps busca quebrar os silos tradicionais entre essas equipes, promovendo a comunicação, a colaboração e a automação de processos, desde a concepção até a entrega e operação de sistemas. O objetivo primordial é acelerar o ciclo de vida do desenvolvimento, garantindo entregas mais rápidas, confiáveis e seguras.

A convergência natural entre a agilidade proporcionada pelo Scrum e a eficiência operacional do DevOps deu origem ao conceito de ScrumOps. Este termo, popularizado por iniciativas como a parceria entre Scrum.org e DevOps Institute em 2017 (WALTERS, 2017), representa a aplicação dos princípios do Scrum em um ambiente DevOps, buscando maximizar a entrega contínua de valor. Análises comparativas recentes reforçam que, enquanto o Scrum foca na gestão iterativa e no desenvolvimento, o DevOps provê a infraestrutura e a cultura de automações necessárias para que essa iteração não encontre gargalos na entrega (SANCHES, 2025). O ScrumOps não é uma nova metodologia, mas sim uma abordagem que integra as melhores práticas de ambos os mundos para criar um fluxo de trabalho coeso e otimizado, onde as equipes de desenvolvimento e operações trabalham em conjunto para alcançar objetivos comuns.

2. Fundamentação Teórica

Para compreender o ScrumOps, é fundamental revisitar os conceitos e princípios que regem o Scrum e o DevOps individualmente, e então analisar como suas filosofias se entrelaçam para formar uma abordagem mais robusta.

2.1. Scrum

O Scrum é um *framework* leve que ajuda pessoas, equipes e organizações a gerar valor por meio de soluções adaptativas para problemas complexos. Seus três pilares empíricos **transparência**, **inspeção** e **adaptação** são a base para a tomada de decisões e a melhoria contínua. A transparência garante que todos os aspectos do processo sejam visíveis para aqueles que são responsáveis pelo resultado. A inspeção permite que os artefatos e o progresso em direção a um objetivo sejam verificados frequentemente para detectar variações indesejáveis. A adaptação, por sua vez, é a capacidade de ajustar o processo ou o material quando os resultados da inspeção indicam que um desvio inaceitável está ocorrendo (SCHWABER, 2020).

Na Figura 1 é apresentado diagrama do *framework* Scrum, ilustrando papéis, artefatos e eventos.

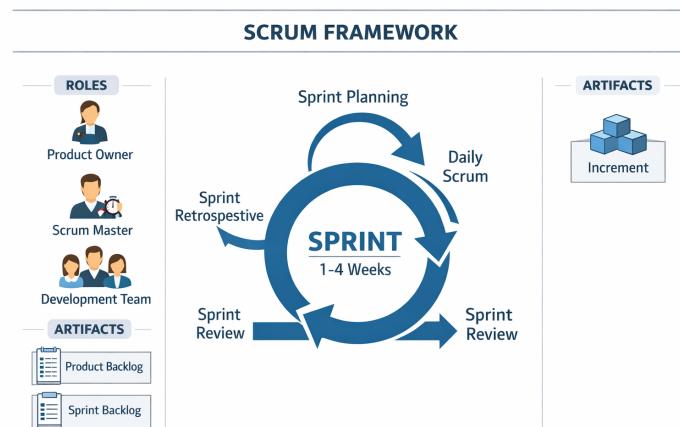


Figura 1: Diagrama do *Framework* Scrum, Ilustrando Papéis, Artefatos e Eventos.

Os papéis no Scrum são bem definidos e colaborativos:

- **Product Owner (PO):** Responsável por maximizar o valor do produto resultante do trabalho da Equipe de Desenvolvimento. Gerencia o Product Backlog, garantindo que ele seja claro, visível e compreendido por todos.
- **Scrum Master (SM):** Responsável por garantir que o Scrum seja compreendido e implementado. Atua como um líder-servidor para a Equipe Scrum, removendo impedimentos e facilitando os eventos Scrum.
- **Time de Desenvolvimento:** Composto por profissionais que realizam o trabalho de entregar um incremento de produto potencialmente utilizável a cada Sprint. São auto-organizáveis e multifuncionais.

2.2. DevOps

DevOps é mais do que um conjunto de ferramentas ou processos; é uma **cultura** que busca integrar as equipes de desenvolvimento e operações, historicamente separadas, para melhorar a colaboração e a comunicação. O objetivo é automatizar e otimizar o fluxo de trabalho, desde o desenvolvimento até a implantação e operação, resultando em entregas mais rápidas, confiáveis e de alta qualidade. A cultura DevOps é frequentemente caracterizada pelo acrônimo CAMS (KIM, 2016).

Na Figura 2 é apresentado o ciclo de vida DevOps, destacando as fases de desenvolvimento e operações.

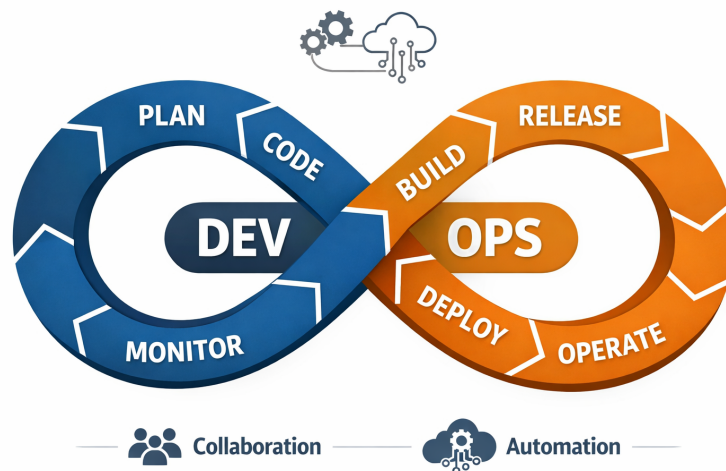


Figura 2: Ciclo de Vida DevOps.

- **Cultura (Culture):** Foco na colaboração, comunicação e responsabilidade compartilhada entre as equipes.
- **Automação (Automation):** Uso extensivo de ferramentas e scripts para automatizar tarefas repetitivas, como integração contínua, entrega contínua e testes.
- **Medição (Measurement):** Coleta e análise de métricas para monitorar o desempenho, identificar gargalos e impulsionar a melhoria contínua.
- **Compartilhamento (Sharing):** Disseminação de conhecimento, feedback e melhores práticas entre as equipes.

2.3. A Complementaridade entre Scrum e DevOps

A aparente dicotomia entre Scrum (focado na agilidade do desenvolvimento) e DevOps (focado na agilidade da entrega e operação) é, na verdade, uma relação de profunda complementaridade. O Scrum, com seus ciclos iterativos e foco na entrega de incrementos de valor, cria um fluxo constante de software que precisa ser integrado, testado e implantado de forma eficiente. É aqui que o DevOps entra, fornecendo as práticas e a cultura necessárias para que essas entregas sejam contínuas e sem atritos (WEST, 2017).

Enquanto o Scrum otimiza o *quê* e o *como* o *software* é construído em pequenas iterações, o DevOps otimiza o *como* esse *software* é entregue e operado de forma contínua. A união dessas abordagens permite que as equipes não apenas desenvolvam *software* de forma ágil, mas também o entreguem e o mantenham em produção com a mesma agilidade e qualidade, fechando o ciclo de *feedback* e garantindo que o valor seja continuamente entregue aos usuários finais. Na Tabela 1 é observado uma análise comparativa do Scrum e DevOps.

Tabela 1: Análise Comparativa entre Scrum e DevOps.

Característica	Scrum	DevOps
Foco Principal	Gestão de projetos e desenvolvimento ágil de produtos.	Cultura e práticas para integração e automação da entrega de software.
Natureza	Framework de processo prescritivo.	Filosofia cultural e conjunto de práticas flexíveis.
Unidade de Trabalho	Sprint (iteração de 1 a 4 semanas).	Fluxo contínuo (CI/CD).
Papéis Principais	Product Owner, Scrum Master, Time de Desenvolvimento.	Não há papéis formais; promove a colaboração entre Dev e Ops.
Artefatos	Product Backlog, Sprint Backlog, Incremento.	Pipelines de automação, Infraestrutura como Código (IaC).
Métricas Chave	Velocidade da equipe, Burndown Chart.	Frequência de implantação, Tempo de recuperação (MTTR), Taxa de falha.

3. ScrumOps: Integração e Práticas

O termo **ScrumOps** surgiu da necessidade de formalizar a união entre o *framework* Scrum e as práticas DevOps, reconhecendo que a agilidade no desenvolvimento de *software* não pode ser plenamente alcançada sem a agilidade na sua entrega e operação. A parceria entre a Scrum.org e o DevOps Institute em 2017 foi um marco importante para a popularização e o entendimento dessa abordagem, que visa transformar as organizações de TI em entidades de entrega de produtos do século XXI (WATERS, 2017). Na Figura 3 é visto integração entre scrum e devops, formando o scrumops, a Tabela 2 são apresentados os benefícios do ScrumOps.

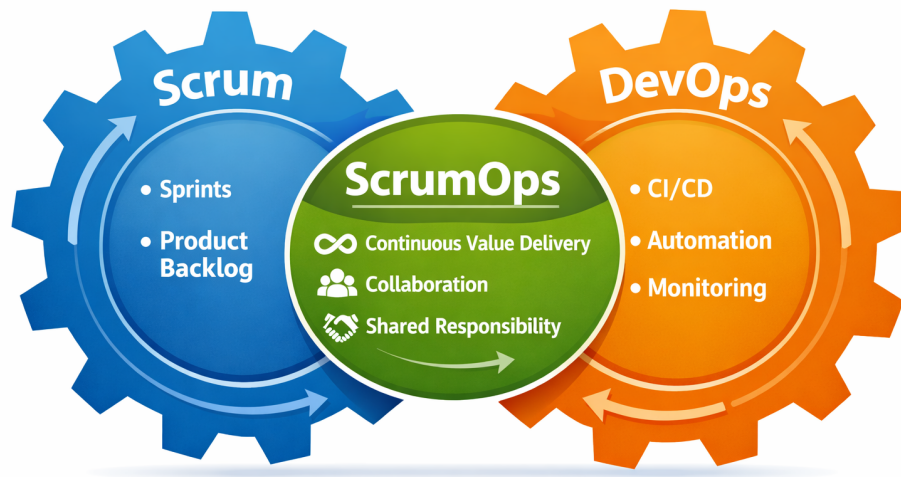


Figura 3: Integração entre Scrum e DevOps, formando o ScrumOps.

Tabela 2: Características e Benefícios do ScrumOps.

Característica	Descrição	Benefício Chave
Integração de Equipes	Desenvolvimento (Dev) e Operações (Ops) trabalham em conjunto.	Redução de silos organizacionais e melhoria da comunicação.
Entrega Contínua de Valor	Foco na entrega rápida e frequente de software funcional.	Maior satisfação do cliente e adaptação ágil às mudanças do mercado.
Definition of Done (DoD) Expandida	Inclui práticas de CI/CD, automação, segurança e monitoramento.	Aumento da qualidade, segurança e confiabilidade do software.
Automação	Uso extensivo de ferramentas para automatizar builds, testes e implantações.	Feedback mais rápido, redução de erros manuais e aceleração do ciclo de desenvolvimento.
Cultura de Colaboração	Compartilhamento de responsabilidades e conhecimento entre Dev e Ops.	Melhoria contínua, inovação e resolução proativa de problemas.

Feedback Loops Curtos	Obtenção e incorporação rápida de feedback em todas as etapas.	Otimização contínua do produto e alinhamento com as necessidades do usuário.
DevSecOps	Integração de práticas de segurança desde o início do ciclo de desenvolvimento.	Redução de vulnerabilidades e conformidade com políticas de segurança.

3.1. Origem e Conceito de "*Continuous Value Delivery*"

Conforme destacado por Dave West, CEO da Scrum.org, e Jayne Groll, CEO do DevOps Institute, o ScrumOps não deve ser visto como uma nova metodologia, mas sim como uma abordagem que integra o Scrum e o DevOps para maximizar a **Entrega Contínua de Valor (*Continuous Value Delivery*)** (WATERS, 2017). A ideia central é que as equipes de desenvolvimento e operações trabalhem de forma colaborativa, eliminando as barreiras tradicionais e garantindo que o software não apenas seja construído rapidamente, mas também seja entregue de forma segura, funcional e com qualidade, de maneira contínua. Isso significa que o valor para o cliente é o foco principal, desde a concepção até a operação em produção.

3.2. Ajustando a *Definition of Done (DoD)* para Práticas DevOps

No Scrum, a ***Definition of Done (DoD)*** é um compromisso crucial que garante a qualidade e a completude de um incremento de produto. Tradicionalmente, a DoD pode focar em aspectos como testes unitários, testes de integração e revisão de código. No contexto do ScrumOps, a DoD é expandida para incluir práticas e critérios de qualidade do DevOps. Isso pode envolver:

- **Testes automatizados** em todos os níveis (unitário, integração, sistema, aceitação).
- **Integração Contínua (CI) e Entrega Contínua (CD)**, garantindo que o código seja continuamente integrado, testado e preparado para implantação.
- **Segurança (Security)**: Inclusão de verificações de segurança automatizadas (DevSecOps) e conformidade com políticas de segurança.
- **Monitoramento e Observabilidade**: Implementação de ferramentas de monitoramento e logs para garantir a visibilidade do sistema em produção.
- **Infraestrutura como Código (IaC)**: Definição e gerenciamento da infraestrutura por meio de código, garantindo consistência e reprodutibilidade.

Ao incorporar esses elementos na DoD, as equipes garantem que cada incremento de *software* não seja apenas funcional, mas também robusto, seguro e pronto para ser operado em um ambiente de produção, alinhando as expectativas de desenvolvimento e operações desde o início.

3.3. Automação como Facilitadora do Empirismo

A automação é um pilar fundamental do DevOps e, conseqüentemente, do ScrumOps. Ela não apenas acelera os processos, mas também atua como um facilitador do empirismo, um dos pilares do Scrum. Ao automatizar tarefas repetitivas, como *builds*, testes e implantações, as equipes podem obter *feedback* mais rápido e frequente sobre o estado do software e do sistema. Esse *feedback* rápido permite uma **inspeção** contínua e uma **adaptação** ágil, permitindo que as equipes identifiquem e corrijam problemas mais

cedo no ciclo de desenvolvimento. A automação também aumenta a **transparência**, pois o estado dos processos e do *software* se torna mais visível para todos os envolvidos, promovendo uma cultura de melhoria contínua e aprendizado (WEST, 2017).

4. Benefícios e Impactos no Desenvolvimento de *Software*

A adoção do ScrumOps traz uma série de benefícios e impactos positivos no ciclo de vida do desenvolvimento de *software*, transformando a maneira como as equipes operam e entregam valor. A sinergia entre Scrum e DevOps resulta em um ambiente mais colaborativo, eficiente e responsivo às necessidades do negócio.

4.1. Redução de Silos Organizacionais

Um dos maiores desafios em organizações tradicionais é a existência de silos entre as equipes de desenvolvimento e operações. Essa separação pode levar a gargalos, comunicação ineficiente e conflitos de prioridades. O ScrumOps, ao promover uma cultura de colaboração e responsabilidade compartilhada, atua diretamente na **redução desses silos**. As equipes passam a trabalhar em conjunto desde o planejamento até a operação, compartilhando conhecimentos, ferramentas e objetivos. Isso resulta em um fluxo de trabalho mais integrado e contínuo, onde todos os membros da equipe se sentem parte de um objetivo comum: a entrega de valor ao cliente (GARCIA, 2024).

4.2. Melhoria na Qualidade e Segurança (DevSecOps)

A integração das práticas DevOps na *Definition of Done* do Scrum, conforme discutido anteriormente, eleva significativamente a qualidade e a segurança do *software*. A automação de testes, a integração contínua e a entrega contínua garantem que os defeitos sejam identificados e corrigidos precocemente, reduzindo o custo e o esforço de retrabalho. Além disso, a incorporação de práticas de segurança desde as fases iniciais do desenvolvimento (DevSecOps) assegura que as vulnerabilidades sejam tratadas proativamente, resultando em sistemas mais robustos e seguros. A visibilidade proporcionada pelo monitoramento contínuo em produção permite uma resposta rápida a quaisquer problemas que possam surgir, minimizando o impacto nos usuários (MOTA, 2024).

4.3. Feedback Loops Mais Curtos

O ScrumOps acelera o ciclo de *feedback* em todas as etapas do desenvolvimento e operação. Com entregas menores e mais frequentes, as equipes podem obter *feedback* dos usuários e das partes interessadas de forma mais rápida. Esse *feedback* é então incorporado nos próximos *Sprints*, permitindo que o produto evolua de maneira mais alinhada às necessidades do mercado. A automação e o monitoramento contínuo também fornecem *feedback* técnico imediato sobre o desempenho e a estabilidade do sistema, permitindo ajustes e otimizações em tempo real. A capacidade de inspecionar e adaptar rapidamente com base em *feedback* contínuo é um dos pilares do empirismo do Scrum, potencializado pelas práticas DevOps (RODRIGUEZ, et al, 2017). Além disso, a introdução da Inteligência Artificial (IA) nesse ecossistema tem demonstrado potencial para otimizar ainda mais esses ciclos, automatizando a análise de grandes volumes de dados de monitoramento e auxiliando na tomada de decisões preditivas durante as *Sprints* (SANCHES, 2025).

5 Considerações Finais

O ScrumOps representa uma evolução natural e necessária no cenário do desenvolvimento de *software*, consolidando a sinergia entre a agilidade do Scrum e a eficiência operacional do DevOps. Longe de ser uma nova metodologia, é uma abordagem integrada que potencializa a entrega contínua de valor, permitindo que as organizações respondam de forma mais rápida e eficaz às demandas de um mercado em constante mudança. A união desses dois pilares não apenas otimiza os processos técnicos, mas, fundamentalmente, promove uma transformação cultural que quebra barreiras e fomenta a colaboração entre equipes.

É imperativo reconhecer que o sucesso do ScrumOps reside menos na adoção de ferramentas específicas e mais na construção de uma cultura organizacional que valorize a comunicação, a responsabilidade compartilhada, a automação inteligente e a busca incessante por *feedback* e melhoria contínua. Ao abraçar o ScrumOps, as empresas não apenas aceleram a entrega de *software*, mas também constroem equipes mais engajadas, produtos de maior qualidade e um ambiente de trabalho mais produtivo e inovador. A jornada para o ScrumOps é uma jornada de aprendizado contínuo e adaptação, mas os benefícios em termos de agilidade, qualidade e satisfação do cliente justificam plenamente o investimento.

Referências

- GARCIA P. S. et. al, **Current DevOps Teaching Techniques: A Systematic Literature Review, 2024**. Disponível em:
<https://sol.sbc.org.br/index.php/sbes/article/download/30379/30185/>. Acesso em: 12 fev. 2026.
- KIM, G.; DEBOIS, P.; WILLIS, J. **The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations**. IT Revolution, 2016.
- MOTA L. L. et.al. **Exploring the Role-play to Teach DevOps, 2024**. Disponível em:
<https://sol.sbc.org.br/index.php/sbes/article/download/30384/30190/>. Acesso em: 12 fev. 2026.
- RODRIGUEZ, P; et. al, **DevOps adoption in Junior Enterprise: an experience report of a Brazilian case, 2017**. Disponível em:
<https://sol.sbc.org.br/index.php/eres/article/download/10085/9966/>. Acesso em: 12 fev. 2026.
- SANCHES, H. M., **Scrum e DevOps: Uma análise comparativa e a sinergia para a entrega contínua de valor, 2025**. Disponível em:
<https://atenaeditora.com.br/catalogo/post/scrum-e-devops-uma-analise-comparativa-e-a-sinergia-para-a-entrega-continua-de-valor>. Acesso em: 12 fev. 2026.
- SANCHES, H. M., **A sinergia entre as metodologias Scrum, DevOps e Inteligência Artificial (IA) no desenvolvimento de software moderno, 2025**. Disponível em:
<https://atenaeditora.com.br/catalogo/post/a-sinergia-entre-as-metodologias-scrum-devops-e-inteligencia-artificial-ia-no-desenvolvimento-de-software-moderno>. Acesso em: 12 fev. 2026.
- SCHWABER, K.; SUTHERLAND, J. **O Guia do Scrum: Um Guia Definitivo para o Scrum: As Regras do Jogo**. Scrum.org, 2020. Disponível em:
<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Portuguese-Brazilian.pdf>. Acesso em: 12 fev. 2026.
- WATERS, J. K. Scrum + DevOps = ScrumOps. **ADT Magazine**, 11 maio 2017. Disponível em:
<https://adtmag.com/articles/2017/05/11/scrumops.aspx>. Acesso em: 12 fev. 2026.

WEST, D. ScrumOps: Making of a 21st Century Product Delivery Organization. In: **Scrum Germany 2017**. YouTube, 17 nov. 2017. Disponível em:
https://www.youtube.com/watch?v=ZmyOnz_d5GU. Acesso em: 12 fev. 2026.