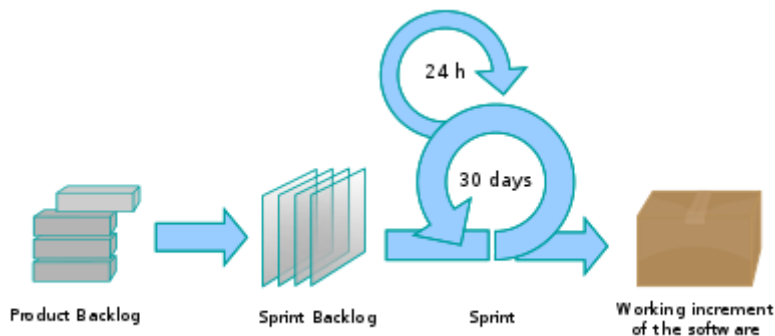


# Scrum (desenvolvimento de software)

Origem: Wikipédia, a enciclopédia livre.

O **Scrum** (pron. [skrʌm])<sup>[1]</sup> é um **framework**<sup>[2]</sup> de trabalho que emprega diversas ferramentas para o **desenvolvimento iterativo e incremental** utilizado no **gerenciamento de projetos** diversos e no **desenvolvimento de software ágil**

Scrum possui seu foco no gerenciamento e projeto da organização onde é difícil planejar à frente. Mecanismos do Controle de Processo Empírico, onde ciclos de feedback constituem o núcleo da técnica de gerenciamento que são usadas em oposição ao tradicional gerenciamento de comando e controle.<sup>[carece de fontes ?]</sup> É uma forma de planejar e gerenciar projetos trazendo a autoridade da tomada de decisão a níveis de propriedade de operação e certeza.<sup>[3]</sup>



O processo Scrum.

Apesar da palavra não ser um **acrônimo**, algumas empresas que implementam o processo a soletram com letras maiúsculas como SCRUM. Isto pode ser devido aos primeiros artigos de Ken Schwaber que capitalizava SCRUM no título.

Scrum não é um processo prescribente, ou seja, ele não descreve o que fazer em cada situação. Ele é usado para trabalhos complexos nos quais é impossível prever tudo o que irá ocorrer

Além disso, o Scrum é um conjunto de valores, princípios e práticas que fornecem a base para que a sua organização adicione suas práticas particulares de engenharia e gestão e que sejam relevantes para a realidade da sua empresa. O resultado será uma versão de Scrum que é exclusivamente sua.

Apesar de Scrum ter sido destinado para gerenciamento de projetos de software, ele pode ser utilizado em equipes de manutenção de software ou como uma abordagem geral de gerenciamento de projetos/programas.

As regras e definições do Scrum estão documentadas no Guia Scrum<sup>[2]</sup> (Scrum Guide, no Inglês). Este documento, por sua vez, é mantido por duas instituições, a Scrum.Org e a Scrum Alliance. Ambas oferecem os cursos preparatórios para certificações relacionadas ao Scrum.<sup>[4]</sup>

Atualmente, as técnicas de **DevOps** são utilizados por programadores no desenvolvimento de software em conjunto com técnicas de gestão e desenvolvimento ágil de software, como o Scrum.<sup>[5][6][7]</sup>

## Índice

- 1 **História**
- 2 **Características**
- 3 **Gerenciamento Ágil de Projetos com Scrum**
- 4 **Sprint**
- 5 **Papéis**
  - 5.1 Papéis principais

5.2	Papéis auxiliares
<b>6</b>	<b>Artefatos Oficiais do Scrum<sup>[2]</sup></b>
6.1	BACKLOG
6.1.1	Product Backlog
6.1.2	Sprint Backlog
<b>7</b>	<b>Artefatos Complementares ao Scrum</b>
7.1	Burndown Chart
<b>8</b>	<b>Eventos Scrum</b>
<b>9</b>	<b>Scrum simplificado</b>
<b>10</b>	<b>Scrum Solo</b>
<b>11</b>	<b>Referências</b>
<b>12</b>	<b>Ver também</b>
<b>13</b>	<b>Ligações externas</b>

## História

---

Inicialmente, o Scrum foi concebido como um estilo de gerenciamento de projetos em empresas de fabricação de automóveis e produtos de consumo, por Takeuchi e Nonaka no artigo "The New Product Development Game" (*Harvard Business Review*, Janeiro-Fevereiro 1986). Eles notaram que projetos usando equipes pequenas e multidisciplinares (*cross-functional*) produziram os melhores resultados, e associaram estas equipes altamente eficazes à formação Scrum do Rugby (utilizada para reinício do jogo em certos casos). Jeff Sutherland,<sup>[8]</sup> John Scumniotales e Jeff McKenna conceberam, documentaram e implementaram o Scrum, conforme descrito abaixo, na empresa Easel Corporation em 1993, incorporando os estilos de gerenciamento observados por Takeuchi e Nonaka. Em 1995, Ken Schwaber formalizou a definição de *Scrum* e ajudou a implantá-lo no desenvolvimento de softwares em todo o mundo.

Scrum junta conceitos de Lean, desenvolvimento iterativo e do estudo de Hiroataka Takeuchi e Ikujiro Nonaka

A função primária do Scrum é ser utilizado para o gerenciamento de projetos de desenvolvimento de software. Ele tem sido usado com sucesso para isso, assim como Extreme Programming e outras metodologias de desenvolvimento. Porém, teoricamente pode ser aplicado em qualquer contexto no qual um grupo de pessoas necessitem trabalhar juntas para atingir um objetivo comum, como iniciar uma escola pequena, projetos de pesquisa científica, ou até mesmo o planejamento de um casamento.

Mesmo que idealizado para ser utilizado em gestão de projetos de desenvolvimento de software ele também pode ser usado para a gerência de equipes de manutenção, ou como uma abordagem para gestão de programas *Scrum de Scrums*

## Características

---

- Clientes se tornam parte da equipe de desenvolvimento (os clientes devem estar genuinamente interessados na saída);
- Entregas frequentes e intermediárias de funcionalidades 100% desenvolvidas;
- Planos frequentes de mitigação de riscos desenvolvidos pela equipe;
- Discussões diárias de status com a equipe de desenvolvimento;
- A discussão diária na qual cada membro da equipe de desenvolvimento responde às seguintes perguntas:
  - O que fiz desde ontem em direção a meta?
  - O que estou planejando fazer até amanhã em direção?
  - Existe algo me impedindo de atingir meta?
- Transparência no planejamento e desenvolvimento;
- Reuniões frequentes com os *stakeholders* (partes interessadas no projeto) para monitorar o progresso;
- Problemas não são ignorados e ninguém é penalizado por reconhecer ou descrever qualquer problema não visto;

- Locais e horas de trabalho devem ser energizadas, no sentido de que "trabalhar horas extras" não necessariamente significa "produzir mais".

## Gerenciamento Ágil de Projetos com Scrum

---

Scrum não só reforçou o interesse em gerenciamento de projetos de software, mas também desafiou as ideias convencionais sobre essa gestão. Scrum é voltado para instituições de gerenciamento de projetos, onde é difícil planejar o futuro com mecanismos de controle de processos empíricos, como loops de feedback, onde constituem o elemento central do desenvolvimento do produto em comparação com a gestão de comando e controle tradicionais orientado. Ela representa uma abordagem radicalmente nova para o planejamento e gerenciamento de projetos de software, trazendo poder de decisão ao nível das propriedades operação e certezas. Scrum reduz defeitos e torna o processo de desenvolvimento mais eficiente, bem como reduzindo os custos de manutenção a longo prazo.

## Sprint

---

Uma sprint é a unidade básica de desenvolvimento em Scrum. Sprints tendem a durar entre uma semana e um mês, e são um esforço dentro de uma faixa de tempo (ou seja, restrito a uma duração específica) de comprimento constante. A adoção de ciclos relativamente curtos permite entregas de partes dos sistemas, gerando valor para os clientes e permitindo uma avaliação dinâmica do trabalho.

Cada sprint é precedido por uma reunião de planejamento (Sprint Planning), onde as tarefas para a sprint são identificadas e um compromisso estimado para o objetivo da sprint é definido e seguido por uma reunião de revisão ou de retrospectiva, onde o progresso é revisto e lições para os próximos sprints são identificadas.

Durante cada sprint, a equipe cria um incremento de produto potencialmente entregável (por exemplo, software funcional e testado). O conjunto de funcionalidades que entram em um sprint vêm do Product Backlog, que é um conjunto de prioridades de requisitos de alto nível definidos pelo **Product Owner**.

Quais itens do backlog que entram para a sprint são determinados durante a reunião de planejamento da sprint (Sprint Planning). Durante esta reunião, o **Product Owner** informa a equipe dos itens no backlog do produto que ele ou ela quer concluídos.

A equipe então determina quantos eles podem se comprometer a concluir durante o próxima sprint, e registram isso no backlog da sprint. Durante um sprint, ninguém está autorizado a alterar o backlog da sprint, o que significa que os requisitos são congelados para esse sprint.

O desenvolvimento de cada sprint deve terminar na "caixa de tempo" prevista. Se os requisitos não são completados por qualquer motivo, eles são deixados de fora e voltam para o backlog do produto. Depois que um sprint é completado, a equipe demonstra como usar o software.

O Scrum permite a criação de equipes auto-organizadas, encorajando a co-localização de todos os membros da equipe e a comunicação verbal entre todos os membros e disciplinas da equipe no projeto.

Um princípio chave do Scrum é o reconhecimento de que, durante um projeto, os clientes podem mudar de ideia sobre o que eles querem e precisam (muitas vezes chamados requisitos churn), e que os desafios imprevisíveis não podem ser facilmente tratados de uma maneira preditiva ou planejada tradicional. Como tal, o Scrum adota uma abordagem empírica, aceitando que o problema não pode ser totalmente entendido ou definido, focando na maximização da habilidade da equipe para entregar rapidamente e responder às necessidades emergentes.

Como outras metodologias de desenvolvimento ágil, o Scrum pode ser implementado através de uma ampla gama de ferramentas. Muitas empresas utilizam ferramentas de software universais, como planilhas para construir e manter artefatos como o backlog da sprint. Há também pacotes de software open-source e proprietários dedicados à gestão de produtos no âmbito do processo Scrum. Outras organizações implementam o Scrum sem o uso de quaisquer ferramentas de software, e mantêm seus artefatos na forma de cópias impressas, como papel, quadros e notas.

- Cada *sprint* é uma iteração que segue um ciclo PDCA e entrega incremento de software pronto.
- Um *backlog* é conjunto de requisitos, priorizado pelo Product Owner (responsável pelo ROI e por conhecer as necessidades do cliente);
- Há entrega de conjunto fixo de itens do *backlog* em série de interações curtas ou *sprints*;
- Breve reunião diária, ou daily scrum, em que cada participante fala sobre o progresso conseguido, o trabalho a ser realizado e/ou o que o impede de seguir avançando (também chamado de Standup Meeting ou Daily Meeting, já que os membros da equipe geralmente ficam em pé para não prolongar a reunião).
- Breve sessão de planejamento, na qual os itens do *backlog* para uma *sprint* (iteração) são definidos;
- Revisão da Sprint, é executada no final da Sprint, para inspecionar o incremento e adaptar o backlog do produto se necessário. As partes interessadas colaboram sobre o que foi feito na Sprint e com base nisso, e em qualquer mudança no backlog do produto durante a sprint, os participantes colaboram naquilo que pode ser feito para otimizar o valor do produto.
- Retrospectiva, é uma oportunidade para o Time Scrum inspecionar a si próprio e criar um plano para melhorias a serem aplicadas na próxima Sprint.

O Scrum é facilitado por um *Scrum Master*, que tem como função primária remover qualquer impedimento à habilidade de uma equipe de entregar o objetivo da *sprint*. O Scrum Master não é o líder da equipe (já que as equipes são auto-organizadas), mas atua como um mediador entre a equipe e qualquer influência desestabilizadora. Outra função extremamente importante de um Scrum Master é o de assegurar que a equipe esteja utilizando corretamente as práticas de Scrum, motivando-os e mantendo o foco na meta da Sprint.

## Papéis

Scrum é um esqueleto de processos que contém grupos de práticas e papéis pré-definidos. Os três papéis definidos no Guia Scrum<sup>[2]</sup> são:

1. o **Scrum Master**, que mantém os processos (normalmente no lugar de um gerente de projeto);
2. o **Dono do Produto** ou **Product Owner**, que representa os *stakeholders* e o negócio;
3. a **Equipe de desenvolvimento** ou **DevTeam**, um grupo multifuncional entre 3 a 9 pessoas e que fazem a análise, projeto, implementação, teste etc.

### Papéis principais

Os papéis principais em equipes Scrum são aqueles comprometidos com o projeto no processo do Scrum - são os que produzem o produto (objetivo do projeto).

#### Product Owner (dono do produto)

O Product Owner representa a voz do cliente e é responsável por garantir que a equipe agregue valor ao negócio. O Product Owner escreve centrado nos itens do cliente (histórias tipicamente do usuário), os prioriza e os adiciona para o product backlog. Equipes de Scrum devem ter um Product Owner, e, embora esse possa também ser um membro da equipe de desenvolvimento, recomenda-se que este papel não seja combinado com o de ScrumMaster.

#### Equipe de Desenvolvimento (*Development Team*)

A equipe de desenvolvimento é responsável pela entrega do produto. A equipe é tipicamente composta de 5-9 pessoas com habilidades multifuncionais que fazem o trabalho real (analisar, projetar, desenvolver, testar técnicas de comunicação, documentos, etc.) Recomenda-se que a equipe seja auto-organizada e auto-conduzida, mas que muitas vezes trabalhem com alguma forma de projeto ou gestão de equipe.

#### Scrum Master

Scrum é facilitado por um Scrum Master, que é responsável pela remoção de impedimentos à capacidade da equipe para entregar o objetivo da sprint / entregas. O Scrum Master não é o líder da equipe, mas age como um tampão entre a equipe e qualquer influência ou distração. O Scrum Master garante que o processo Scrum seja usado como pretendido. O Scrum Master é o responsável pela aplicação das regras. Uma parte fundamental do papel

do Scrum Master é proteger a equipe e mantê-la focada nas tarefas em mãos. O papel também tem sido referido como um líder-servo para reforçar essa dupla perspectiva.

## Papéis auxiliares

Os papéis auxiliares em equipes Scrum são aqueles com nenhum papel formal e envolvimento frequente no processo de Scrum, mas, ainda assim, devem ser levados em conta.

### Partes interessadas (clientes, fornecedores)

Estas são as pessoas que permitem o projeto e para quem o projeto vai produzir o acordado benefício, que justifica a sua produção. Eles só estão diretamente envolvidos no processo durante as revisões sprint.

### Gerentes (incluindo gerentes de projeto)

Pessoas que irão configurar o ambiente para desenvolvimento de produtos.

## Artefatos Oficiais do Scrum<sup>[2]</sup>

---

O Scrum Guide define os seguintes artefatos obrigatórios como necessários para o sucesso de um projeto que utilize Scrum:

### BACKLOG

Um *backlog* é uma lista de itens priorizados a serem desenvolvidos para um software. Este artefato é a principal fonte de informação para o Planejamento *desprint* (Sprint Planning). No decorrer da *sprint*, o Product Owner, o Scrum Master e a Equipe decidem no que a equipe irá trabalhar. O Product Owner mantém uma lista priorizada de itens de *backlog*, o *backlog* do produto, o que pode ser repriorizado durante o planejamento da *sprint*. A Equipe seleciona itens do topo do *backlog* do produto. Eles selecionam somente o quanto de trabalho eles podem executar para terminar. A Equipe então planeja a arquitetura e o design de como o *backlog* do produto pode ser implementado. Os itens do *backlog* do produto são então destrinchados em tarefas que se tornam *backlog* da *sprint*.

#### Product Backlog

O *Product Backlog* é mantido pelo *Product Owner* e é uma lista de requisitos que tipicamente vêm do cliente. O *Product Backlog* pode ser alterado a qualquer momento pelo *Product Owner* ou por decisão deste.

#### Sprint Backlog

O *Sprint Backlog* é uma lista de itens selecionados do *Product backlog* e contém tarefas concretas que serão realizadas durante a próxima *sprint* para implementar tais itens selecionados.

O *Sprint Backlog* é uma representação em tempo real do trabalho que o *Development Team* planeja concluir na *sprint* corrente, e ele pertence unicamente ao *Development Team*.

## Artefatos Complementares ao Scrum

---

Além dos artefatos oficiais estabelecidos no Scrum Guide,<sup>[2]</sup> existem artefatos complementares que podem ser utilizados para melhorar a qualidade das entregas e a eficiência do time.<sup>[9]</sup> Alguns deles são:

### Burndown Chart

O Burndown é um simples gráfico, com dois eixos X e Y, baseado nas atividades que não ultrapassem um dia de trabalho. O eixo Y indica o número de tarefas existentes na sprint e o eixo X os dias que representam o tamanho da sprint.

# Eventos Scrum

---

Um momento bom para as discussões diárias é depois do almoço. Durante a manhã pode ser complicado. <sup>[carece de fontes?]</sup> Estas discussões de status não demoram e uma forma eficiente de fazer estas reuniões seria ficar em pé e em frente a um quadro negro. Como as pessoas tendem a ficar cansadas depois do almoço, ter uma viva reunião em pé nessa hora permite que a equipe mantenha a sua energia alta. Como todos estiveram trabalhando durante a manhã, suas mentes estão focadas no trabalho e não em questões pessoais. Grandes usuários do processo são enfáticos na necessidade de os membros da equipe estarem em pé durante a reunião, para permitir maior agilidade e evitar perdas no foco. Recomenda-se inclusive evitar lugares onde as pessoas possam se apoiar. <sup>[carece de fontes?]</sup>

## Daily Scrum Meeting

Cada dia durante a sprint, uma reunião de status do projeto ocorre. Isso é chamado de "scrum diário", ou "de pé o dia". Esta reunião tem diretrizes específicas:

- A reunião começa precisamente no horário marcado.
- Todos são bem-vindos, mas apenas "pouco" podem falar.
- O encontro tem duração determinada (Time-Box) e dura no máximo 15 minutos.
- A reunião deve acontecer no mesmo local e mesma hora todos os dias
- Durante a reunião, cada membro da equipe responde a três perguntas:
  - O que você tem feito desde ontem em direção a meta?
  - O que você está planejando fazer hoje em direção a meta?
  - Você tem algum problema impedindo você de realizar seu objetivo em direção a meta?

É papel do Scrum Master para facilitar a resolução desses impedimentos. Normalmente, isso deve ocorrer fora do contexto do Daily Scrum para que a reunião possa durar menos de 15 minutos.

## Reunião de Planejamento da Sprint (*Sprint Planning Meeting*)

No início do ciclo de sprint (a cada 7-30 dias), um *Sprint Planning Meeting* é realizado.

- Selecione o trabalho que está a ser feito.
- Prepare o Sprint Backlog que detalha o tempo que levará para fazer esse trabalho, com toda a equipe.
- Identificar e comunicar o quanto o trabalho é susceptível de ser feito durante a sprint atual.
- Dividida em duas partes:
  - Parte 1 (Primeiras quatro horas): ~~com~~ Product Owner: diálogo para priorizar o Product Backlog.
  - Parte 2 (Próximas quatro horas): ~~com~~ apenas: hash de um plano para a Sprint, resultando na Sprint Backlog.

No final de um ciclo de sprint, são realizadas duas reuniões: a "Sprint Review" e do "Sprint Retrospective".

## Reunião de Revisão da Sprint (*Sprint Review*)

- Rever o trabalho que foi concluído e não concluído.
- Apresentar o trabalho realizado para os *stakeholders* (ou "a demo"). Um trabalho incompleto não pode ser demonstrado.

## Retrospectiva da Sprint (*Sprint Retrospective*)

- Todos os membros da equipe refletem sobre a sprint passada.
- Faça melhorias contínuas de processos.

Duas questões principais são feitas na retrospectiva da sprint: O que correu bem durante a corrida? O que poderia ser melhorado na próxima sprint?

# Scrum simplificado

---

Muitas organizações têm sido resistentes às metodologias introduzidas em baixos níveis da organização. Porém, a adaptabilidade do Scrum permite que ele seja introduzido de forma invisível (*stealth*), usando os três passos:

- Agende uma demonstração do software com seu cliente em um mês a partir de agora;
- Como equipe, tome um mês para deixar o software pronto para uma demo, com funcionalidades prontas, não simplesmente telas;
- Na demonstração, obtenha *feedback* e use-o para guiar o seu próximo mês de trabalho de desenvolvimento.

## Scrum Solo

---

Scrum é baseado em pequenas equipes. Ele permite a comunicação entre os membros da equipe. Entretanto, há uma grande quantidade de softwares desenvolvidos por programadores solos. Um software sendo desenvolvido por um só programador pode ainda se beneficiar de alguns princípios do Scrum, como: um *backlog* de produto, um *backlog* de *sprint*, um *sprint* e uma retrospectiva *desprint*. Scrum Solo é uma versão adaptada para uso de programadores solos.<sup>[10]</sup>

## Referências

---

1. «Scrum» (<http://dictionary.reference.com/browse/scrum?s=t>) (em inglês). [Dictionary.com](http://dictionary.reference.com). Consultado em 25 de fevereiro de 2013
2. «Home | Scrum Guides» (<http://www.scrumguides.org>) [www.scrumguides.org](http://www.scrumguides.org) (em inglês). Consultado em 1 de junho de 2017
3. Schwaber, Ken (fevereiro de 1, 2004). *Agile Project Management with Scrum* [S.l.]: Microsoft Press. ISBN 978-0-7356-1993-7
4. Site do Scrum.Org oferece o curso de treinamento oficial de Scrum Master (<https://www.scrum.org/>)
5. DevOps and Agile (Scrum Alliance) (<https://www.scrumalliance.org/community/articles/2014/april/devops-and-agile>)
6. Qual a diferença entre extreme programming, Scrum e DevOps? (<https://gaea.com.br/qual-a-diferenca-entre-extreme-programming-scrum-e-devops/>)
7. Entenda o que a união entre DevOps e Scrum pode gerar (<https://gaea.com.br/entenda-o-que-a-uniao-entre-devops-e-scrum-pode-gerar/>)
8. *Scrum - a arte de fazer o dobro de trabalho na metade do tempo* [S.l.: s.n.] 2014 | nome1= sem | sobrenome1= em Authors list (ajuda)
9. **Erro de citação: Código <ref> inválido; não foi fornecido texto para as refs de nome 1**
10. Pagotto, T.; J. A. (1 de junho de 2016). «Scrum solo: Software process for individual development» (<http://ieeexplore.ieee.org/document/7521555/>) *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)* 1–6. doi:10.1109/CISTI.2016.7521555 (<https://dx.doi.org/10.1109%2FCISTI.2016.7521555>)

## Ver também

---

- [Ken Schwaber](#)
- [John Scumniotales](#)
- [Jeff Sutherland](#)

## Ligações externas

---

- [Guia oficial do Scrum, em português \(PDF\)](#)
- [Scrum Alliance.org](#) organização de âmbito mundial que certifica e fomenta o desenvolvimento do Scrum no mundo
- [Comunidade Scrum no Brasil](#)
- [Comunidade Scrum em Portugal](#)
- [Uma introdução ao Scrum](#) (título em inglês: *A Reusable Scrum Presentation*)
- [Guia oficial do Scrum, em vários idiomas](#)
- [Referências online do livro Scrum - Guia Prático para Projetos Ágeis, de Cesar Brod](#)

---

**Esta página foi editada pela última vez à(s) 16h25min de 13 de novembro de 2017.**

Este texto é disponibilizado nos termos da licença [Creative Commons - Atribuição - Compartilha Igual 3.0 Não Adaptada \(CC BY-SA 3.0\)](#); pode estar sujeito a condições adicionais. Para mais detalhes, consulte [as condições de uso](#)