

## **METODOLOGIA ÁGIL – SCRUM**

### **SCRUM**

Scrum é um framework Ágil, simples e leve, utilizado para a gestão do desenvolvimento de produtos complexos imersos em ambientes complexos. Ele utiliza uma abordagem iterativa e incremental para entregar valor com frequência e, assim, reduzir os riscos do projeto.

Scrum é uma ferramenta que pode trazer diversos benefícios em comparação a outras formas de se conduzir projetos, mas somente se bem utilizada. Scrum pode permitir reduzir os riscos de insucesso, entregar valor mais rápido e desde cedo, e lidar com as inevitáveis mudanças de escopo, transformando-as em vantagem competitiva. Seu uso pode também aumentar a qualidade do produto entregue e melhorar a produtividade das equipes.

Ao aprender Scrum, você passará por termos como facilitação, trabalho em equipe, auto-organização, metas de negócios, motivação, relacionamento com os clientes, entre tantos outros. Scrum utiliza-se de poucos conceitos novos, e essa é uma de suas grandes qualidades: junta práticas de mercado já conhecidas e consagradas de uma forma organizada e que funciona.

Mas por que Scrum é uma boa escolha? É importante lembrar que não existe uma solução única para todos os problemas. Scrum é um framework simples e pequeno e, assim, funciona bem em cada contexto se for utilizado em conjunto com outras técnicas e práticas a serem experimentadas e adaptadas.

Os benefícios do uso do Scrum incluem:

- Entregas frequentes de retorno ao investimento dos clientes;
- Redução dos riscos do projeto;
- Maior qualidade no produto gerado;
- Mudanças utilizadas como vantagem competitiva;
- Visibilidade do progresso do projeto;
- Redução do desperdício;
- Aumento da produtividade.

Scrum é Ágil

Scrum é ágil porque, assim como outros métodos, metodologias e frameworks, sua utilização deve seguir os princípios e valores do Manifesto para o Desenvolvimento Ágil de Software.

## **Manifesto para Desenvolvimento Ágil de Software**

Estamos descobrindo maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Por meio deste trabalho, passamos a valorizar:

- **Indivíduos e interações** mais que processos e ferramentas;
- **Software em funcionamento** mais que documentação abrangente;
- **Colaboração com o cliente** mais que negociação de contratos;
- **Responder a mudanças** mais que seguir um plano.

Ou seja, mesmo havendo valor aos itens à direita, valorizamos mais os itens à esquerda. Sendo assim, os valores do Scrum partem de dar prioridade a quem produz e depois ao produto.

## **Os Princípios Ágeis**

Os doze princípios Ágeis foram cunhados a partir do Manifesto por parte de seus autores e podem ser vistos em seguida, cada um com uma pequena explicação adicional:

### **1) Nossa maior prioridade é satisfazer o cliente por meio da entrega cedo e frequente de software com valor.**

O foco no desenvolvimento do produto está na satisfação dos clientes. Gera-se, desde cedo e frequentemente, retorno ao investimento dos clientes no projeto a partir da entrega de partes do produto que atendam às suas necessidades. Esse princípio se opõe à prática de se seguir um plano detalhado, sugerindo que a prioridade está em se adaptar e buscar, em cada momento, o que de fato trará valor aos clientes, para entregar-lhes o mais cedo e frequentemente possível.

### **2) Mudanças de requisitos são bem-vindas, mesmo em fases tardias do desenvolvimento. Os processos Ágeis utilizam a mudança em favor da vantagem competitiva para o cliente.**

Aceitar a mudança como natural no processo de desenvolvimento do produto, para melhor atender às necessidades dos clientes. Ao se trabalhar em ciclos curtos de feedback, permite-se aos clientes evoluírem o produto à medida que melhor entendem suas necessidades e adaptarem às mudanças de mercado,

tornando-se mais competitivos. Esse princípio se opõe a se tratar o processo de desenvolvimento do produto como previsível, cenário irreal no qual a necessidade de mudança poderia e deveria ser prevenida, já que ela seria considerada indesejada e custosa.

**3) Entregar software em funcionamento com frequência, desde a cada duas semanas até a cada dois meses, com uma preferência por prazos mais curtos.**

Entregar a seus clientes e usuários, com frequência, partes do produto prontas gera, a cada entrega, retorno ao investimento dos clientes e permite obter-se feedback sobre o que foi produzido. Assim, pode-se adaptar o produto às necessidades dos clientes incrementalmente, reduzindo os riscos do projeto. Esse princípio se opõe a realizar poucas ou, no limite, uma entrega de valor única, apenas ao final do projeto.

**4) As pessoas do negócio e os desenvolvedores devem trabalhar em conjunto diariamente ao longo do projeto.**

Pessoas de negócio e desenvolvedores do produto possuem o objetivo comum de garantir a geração de valor para os clientes e, para atingir esse objetivo, cooperam continuamente durante todo o projeto, interagindo com frequência. Esse princípio se opõe ao cenário de antagonismo comum em projetos de desenvolvimento de software, nos quais pessoas de negócios – que frequentemente incluem os próprios clientes do projeto – e desenvolvedores raramente se comunicam e, muitas vezes, estão em lados opostos.

**5) Construa projetos em torno de indivíduos motivados. Dê-lhes o ambiente e o suporte que precisam e confie neles para realizarem o trabalho.**

O produto é construído por pessoas. O ambiente, o suporte e a confiança necessários para realizar seu trabalho são fatores fundamentais para sua motivação. Esse princípio se opõe à crença de que o produto se constrói em torno das melhores ferramentas e processos, e não das pessoas, apoiando-se nos melhores instrumentos de monitoração e controle externos, por exemplo.

**6) O método mais eficiente e efetivo de se transmitir informação para e entre uma equipe de desenvolvimento é a conversa face a face.**

A melhor forma de comunicação entre membros do time que desenvolve o produto e entre esse time e o mundo externo é a comunicação face a face, que é direta, síncrona e enriquecida pela entonação de voz, olhar e linguagem corporal, entre outros fatores. Quando a comunicação presencial não é viável (em um projeto distribuído, por exemplo) é uma boa prática fazer o melhor uso possível da tecnologia disponível para se aproximar da comunicação face a face. Esse

princípio se opõe à utilização de documentos, e-mails, telefone, entre outros, como formas padrão de comunicação em um projeto.

#### **7) Software em funcionamento é a principal medida de progresso.**

O progresso do projeto ocorre à medida que partes do produto que signifiquem valor são entregues aos clientes do projeto. Esse princípio se opõe à prática de se gerar artefatos como protótipos e extensos documentos de planos e especificações e, assim, acreditar que se progrediu no projeto. Isso também se opõe à geração de quaisquer artefatos e partes do produto – inclusive documentação – que não gerem valor para os clientes do projeto.

#### **8) Os processos Ágeis promovem o desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter indefinidamente um ritmo constante.**

Busca-se promover um ritmo constante e sustentável para o trabalho do time que desenvolve o produto, o que se torna possível quando esse ritmo é apoiado por toda a cadeia, incluindo usuários e patrocinadores. No entanto, ao se exigir do time um compromisso com mais trabalho do que ele é capaz de produzir, são muitas vezes adotadas as horas extras, o trabalho em fins de semana e a pressa exagerada para se cumprir o prazo de entrega, por exemplo. Essas práticas podem levar à insatisfação dos membros do time de desenvolvimento, a uma menor produtividade e a uma menor qualidade no produto gerado.

#### **9) A atenção contínua à excelência técnica e a um bom projeto aumentam a agilidade.**

O produto projetado com qualidade e produzido com excelência técnica permite que seja facilmente ser modificado e, assim, aceite a mudança como natural no processo de seu desenvolvimento. Assim, a alta qualidade no produto gerado é essencial para se manter a Agilidade. Esse princípio se opõe totalmente à crença de que, para se obter velocidade e flexibilidade no desenvolvimento do produto, a qualidade deveria ser sacrificada.

#### **10) Simplicidade - a rate de se maximizar a quantidade de trabalho não feito - é essencial.**

Evita-se o desperdício no desenvolvimento do produto ao não se realizar trabalho que não é necessário. Exemplos comuns de desperdícios incluem desenvolvimento de funcionalidades de que os clientes não precisam ou de soluções desnecessariamente complexas, planejamento com nível de detalhes maior do que se pode ter em um determinado momento e uso ou geração de artefatos desnecessários.

**11) As melhores arquiteturas, requisitos e projetos emergem de equipes que se auto-organizam.**

Equipes com maior autonomia são mais eficientes. Essas equipes auto-organizadas trabalham em direção a metas acordadas, mas têm a liberdade de decidir qual a melhor forma de realizar esse trabalho e, assim, são responsáveis e responsabilizadas por seus resultados. Dessa forma, geram um melhor produto.

**12) Em intervalos de tempo regulares, a equipe reflete sobre como se tornar mais efetiva e então refina e ajusta seu comportamento de acordo.**

Para se tornar cada vez mais efetiva, a equipe regularmente inspeciona suas formas de trabalho, identifica pontos de melhoria e se adapta de acordo, promovendo a melhoria incremental contínua. É a inspeção e adaptação que o time realiza em seus processos de trabalho.

## **Os Valores do Scrum**

Além dos princípios Ágeis, Scrum também tem o seu próprio conjunto de valores a serem seguidos. Eles complementam as regras do Scrum descritas em seus papéis, artefatos e eventos. Assim, os valores do Scrum representam os pilares para todo o trabalho realizado pelas pessoas que trabalham no projeto, ou seja, o time que desenvolve o produto e a pessoa de negócios que o define, chamada no Scrum de Product Owner.

Os cinco valores do Scrum são: foco, coragem, franqueza, compromisso e respeito.

- Foco: os times mais produtivos trabalham em apenas um projeto de cada vez, evitando a multitarefa;
- Coragem: as pessoas que trabalham no projeto têm coragem para aceitar a mudança como parte natural do processo de desenvolvimento do produto;
- Franqueza: a franqueza ou transparência é necessária para que se possa realizar a inspeção e adaptação. Assim, o time busca melhorar sua forma de trabalhar a partir do feedback frequente de seus membros, o que cria visibilidade sobre os problemas e estimula a busca por soluções;
- Compromisso: o time determina como seu trabalho será realizado, monitora seu progresso e realiza as correções de rumo que achar necessárias. Assim, é responsável e responsabilizado pelos seus resultados;

- Respeito: os membros do time trabalham juntos, compartilhando responsabilidades, e assim ajudam-se uns aos outros em seu trabalho. Todos que trabalham no projeto respeitam as opiniões uns dos outros, ouvem e buscam entender os diferentes pontos de vista.

## **SPRINT**

Scrum é iterativo. O produto é desenvolvido em ciclos ou iterações sucessivas. Em cada um desses ciclos é gerado um incremento no produto, que se soma e modifica o que já se tem pronto do produto até o momento. No Scrum, os ciclos são chamados de Sprints. Eles têm tamanho fixo e acontecem um depois do outro, sem intervalos entre eles, ou seja, tudo em um projeto que utiliza Scrum acontece dentro dos Sprints. Cada Sprint é composto por:

- reunião de Sprint Planning, onde se realiza o planejamento do Sprint;
- trabalho de desenvolvimento do produto, que inclui tudo que é necessário para se entregar uma parte do produto pronta ao final do Sprint;
- reunião de Daily Scrum, contida no dia a dia de desenvolvimento, que é uma reunião diária para gerar visibilidade e planejar o próximo dia de desenvolvimento;
- reunião de Sprint Review, onde os resultados do trabalho realizado no Sprint são apresentados a clientes e demais partes interessadas, que oferecem seu feedback;
- reunião de Sprint Retrospective, onde o time avalia o que foi bem e o que precisa ser melhorado na sua forma de trabalho;
- quaisquer reuniões ou sessões programadas adicionais que se fizerem necessárias, como por exemplo a reunião de Release Planning, as sessões de Refinamento do Product Backlog etc.

A figura representa a iteração do Scrum, indicando a sucessão de elementos presentes em todos os ciclos (ou Sprints) e a sucessão de ciclos. A reunião de Daily Scrum e quaisquer reuniões adicionais acontecem no dia a dia de desenvolvimento.

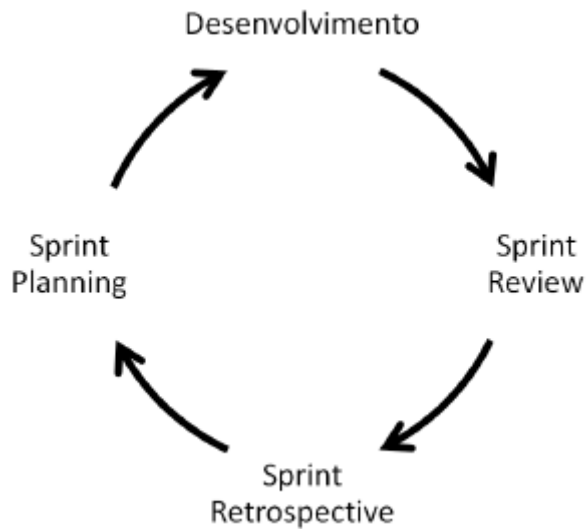


Imagem retirada do Livro: Scrum – Gestão Ágil para Projetos de Sucesso

O projeto para o desenvolvimento de um produto é contratado ou simplesmente se inicia para suprir algum objetivo ou necessidade de negócios, seja de um cliente específico, de um grupo de clientes ou visando uma oportunidade de mercado. A **Visão do Produto** é uma boa forma de traduzir esse objetivo a ser alcançado.

### Analogia do Bolo

Vamos supor que dois times de confeiteiros irão assar um bolo com diversas camadas, cada um utilizando um método diferente

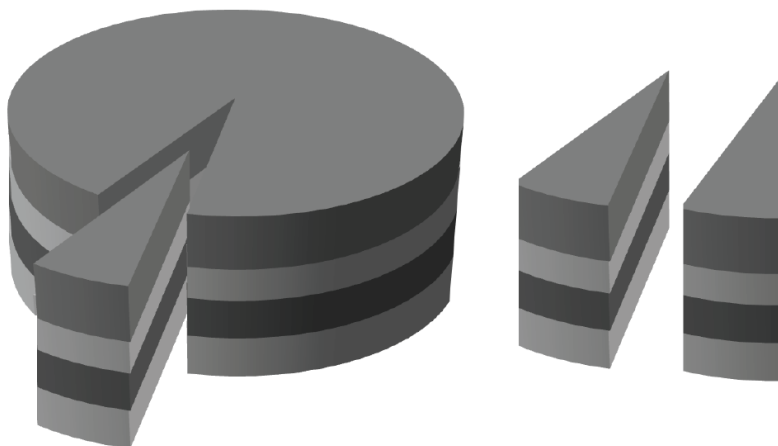


Imagem retirada do Livro: Scrum – Gestão Ágil para Projetos de Sucesso

O primeiro time irá usar um método tradicional, em que se produz o bolo camada por camada. Assim, seus membros misturam os ingredientes da primeira camada do bolo e então a assam. Em seguida, misturam os ingredientes da segunda camada, assam-na e colocam sobre a primeira. Fazem o mesmo com a terceira, quarta e quinta camadas, colocando cada uma sobre a anterior. Somente ao final uma fatia pode ser cortada do bolo para ser consumida.

O segundo time tem um objetivo ousado. Deve-se permitir que o bolo seja consumido o mais rápido possível, e não apenas no final. Para atingir esse objetivo, o time utiliza uma forma de trabalho bem diferente: um método iterativo e incremental. Eles produzem o bolo fatia por fatia, sendo que cada uma já possui todas as camadas necessárias. Assim, o bolo já pode ser consumido desde a sua primeira fatia. Mesmo que cada uma possa ser bem fina.

É claro que nenhum confeitiro irá utilizar o método do segundo time para produzir um bolo, mas o trabalho de desenvolvimento do produto com Scrum se assemelha muito com essa forma de se trabalhar. Diferentemente do trabalho com métodos tradicionais, o time de desenvolvimento não trabalha em camadas do produto. A parte do produto, gerada pelo time em cada ciclo de desenvolvimento, é uma fatia do produto de ponta a ponta, pronta para ser utilizada. Essa entrega de fatias funcionando do produto gera desde cedo retorno ao investimento dos clientes e possibilita o feedback, reduzindo os riscos do projeto.

## **PAPÉIS**

### **Product Owner**

O Product Owner é o responsável por definir, comunicar e manter a Visão do Produto relativamente constante ao longo do projeto. O Product Owner é único. Ele trabalha com os clientes do projeto e com quaisquer outras partes interessadas que possam contribuir para o entendimento e definição da Visão do Produto. O grupo de partes interessadas do projeto também inclui os próprios usuários do produto, que receberão ao longo do desenvolvimento partes prontas do produto para serem utilizadas.

### **Time de Desenvolvimento**

O Time de Desenvolvimento realiza o trabalho de desenvolvimento do produto. Ele é multidisciplinar, o que significa que possui, em seus membros, todo o conhecimento necessário para realizar esse trabalho. O Time de Desenvolvimento é também auto-organizado, ou seja, ele próprio define como irá



realizar o trabalho e gerenciar seu progresso em direção a metas de negócios acordadas com o Product Owner.

## **Scrum Master**

O Scrum Master é o responsável por garantir que os impedimentos que o Time de Desenvolvimento encontra em seu trabalho sejam removidos, atuando quando necessário como um agente de mudança na organização. Esses impedimentos geram o risco de não se atingirem os objetivos. O Scrum Master está presente e age como um facilitador em todas as reuniões do Scrum, facilita o trabalho do dia a dia do Time de Desenvolvimento e facilita as interações entre o Time de Desenvolvimento e o Product Owner. Ele também ensina Scrum ao Time e a se auto-organizarem. O Scrum Master é tão neutro quanto possível e possui soft skills, ou seja, competências comportamentais e pessoais, para realizar seu trabalho.

## **ARTEFATOS / EVENTOS**

### **Product Backlog**

Antes do início do desenvolvimento, o Product Owner inicia, a partir da Visão do Produto, a criação de uma lista ordenada, incompleta e dinâmica de itens que representam o que ele acredita que será produzido ao longo do projeto. Essa lista é chamada de Product Backlog.

Os itens do alto do Product Backlog são os mais importantes naquele momento e, por essa razão, possuem mais detalhes para serem desenvolvidos primeiro. Os itens a seguir têm gradativamente menos detalhes.

O Product Backlog inicial pode ser longo, contendo desde itens pequenos e bem detalhados até itens grandes e vagos. Mas ele também pode conter apenas uma quantidade de itens necessária para se iniciar o desenvolvimento. O Product Backlog evoluirá ao longo de todo o projeto e será frequentemente modificado com a adição, subtração, reordenamento e modificação de seus itens.

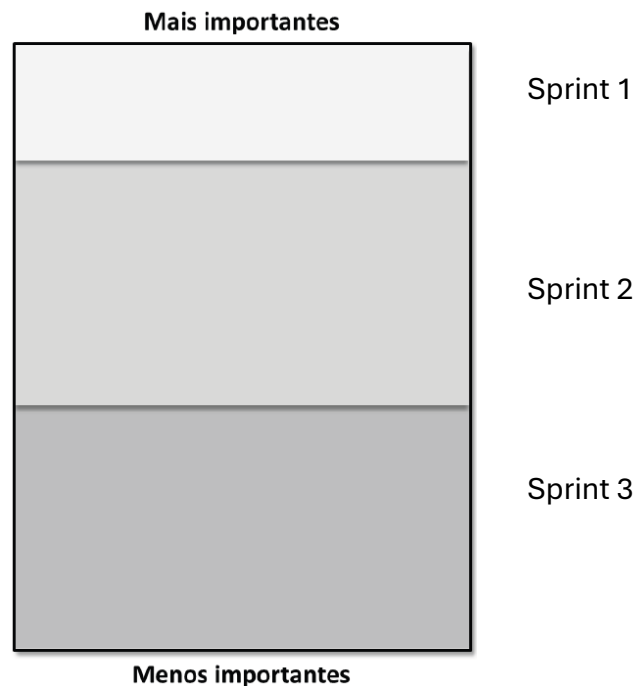


Imagem retirada do Livro: Scrum – Gestão Ágil para Projetos de Sucesso

## **Sprint Planning**

O Sprint Planning é a reunião onde se inicia o Sprint, na qual se planeja o trabalho a ser realizado no próprio Sprint.

Nessa reunião, Time de Desenvolvimento e Product Owner negociam, a partir dos itens do alto do Product Backlog, o que será desenvolvido. Ou seja, facilitados pelo ScrumMaster, eles selecionam um conjunto de itens do alto do Product Backlog que julgam ser capazes de desenvolver na duração do Sprint, o que é apenas uma previsão, e estabelecem um objetivo ou meta de negócios a ser alcançada com o desenvolvimento desses itens, chamada de Meta do Sprint. O Time de Desenvolvimento, então, se compromete com atingir essa Meta do Sprint. Os membros do Time de Desenvolvimento definem, durante a reunião de Sprint Planning, como o trabalho será realizado.

É importante que os itens do alto do Product Backlog estejam preparados para que a reunião de Sprint Planning seja eficiente e produtiva. Itens que chegam à reunião sem detalhes suficientes, por exemplo, podem colocar todo o Sprint em risco.

## Sprint Backlog

Além de, juntamente com o Product Owner, selecionar os itens e definir uma meta, o Time de Desenvolvimento também cria um plano de como o que foi selecionado será desenvolvido. Esse plano é geralmente expresso por tarefas a serem realizadas durante o Sprint. O conjunto de itens selecionados e seu respectivo plano é chamado de Sprint Backlog e é geralmente representado na forma de um Quadro de Tarefas.

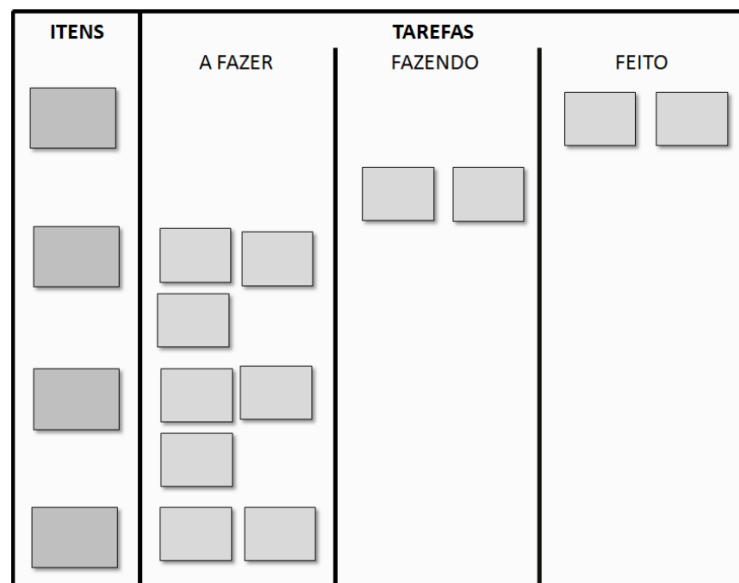


Imagem retirada do Livro: Scrum – Gestão Ágil para Projetos de Sucesso

## Daily Scrum

A Daily Scrum é uma reunião que acontece em cada dia de trabalho de desenvolvimento, onde os membros do Time de Desenvolvimento se encontram por no máximo quinze minutos, preferencialmente no mesmo horário e local com o objetivo de garantir a visibilidade de seu trabalho entre eles e planejar, informalmente, o próximo dia de trabalho. O ScrumMaster pode, se necessário, facilitar essa reunião, mas sua presença não é obrigatória.

Em uma reunião de Daily Scrum é esperado que cada membro do Time de Desenvolvimento informe o que já foi feito, o que ainda falta fazer e se há algum tipo de impedimento que esteja atrapalhando o desenvolvimento da atividade.

## **Incremento do Produto**

O Incremento do Produto é apresentado ao final normal do Sprint, ele é gerado, a partir dos itens do Sprint Backlog pelo Time de Desenvolvimento, e deve ser entregável, que representa um valor visível para os clientes do projeto. Ser entregável significa que nenhum trabalho adicional é necessário para que esse Incremento do Produto possa ser entregue aos clientes do projeto. Entregar ou não o Incremento ao final do Sprint, no entanto, é uma decisão de negócios e, mesmo que já seja possível, cabe ao Product Owner decidir quando fazê-lo.

## **Sprint Review**

A reunião de Sprint Review, é onde os resultados do trabalho realizado no Sprint são apresentados a clientes e demais partes interessadas, que oferecem seu feedback.

O Product Owner convida para a reunião de Sprint Review os clientes do projeto e demais pessoas relevantes que possam prover feedback sobre o que foi produzido durante o Sprint. Nessa reunião, o Time de Desenvolvimento e o Product Owner apresentam e demonstram para os presentes tudo o que foi produzido no Sprint.

## **Sprint Retrospective**

Após a reunião de Sprint Review, Time de Desenvolvimento e Product Owner realizam a reunião de Sprint Retrospective, facilitados pelo ScrumMaster. Nela, ambos identificam o que foi bem no Sprint corrente, e que por essa razão pode ser mantido no próximo Sprint, e o que pode melhorar, buscando formas práticas e traçando planos de ação para fazê-lo. O objetivo desse processo de melhoria contínua é tornar o Time de Scrum cada vez mais efetivo. É onde o time avalia o que foi bem e o que precisa ser melhorado na sua forma de trabalho.

Uma vez terminada a reunião de Sprint Retrospective, está encerrado o Sprint. Um novo Sprint se inicia imediatamente após o término da anterior.

A reunião de Sprint Retrospective não deve ser utilizada para se identificarem ações de melhoria no produto, trabalho que acontece na reunião de Sprint Review.

O principal foco da reunião de Sprint Retrospective é a identificação do que precisa ser melhorado nos processos de trabalho do Time de Scrum. Por essa

razão, é necessário que falhas ocorridas durante o Sprint que se encerra sejam identificadas e exploradas com franqueza. Assim, embora não se devam apontar erros individuais nessa reunião, o trabalho de inspeção envolve a capacidade dos participantes de expor, em algum grau, suas próprias limitações e pontos fracos e, principalmente, as do Time de Scrum como um todo.

### Definição de Preparado (DoR)

A Definição de Preparado é um artefato utilizado para garantir que os itens a serem considerados na reunião de Sprint Planning estejam preparados segundo um critério bem definido. É um acordo formal entre Product Owner e Time de Desenvolvimento sobre o estado em que um item do Backlog deve estar para estar qualificado para discussão na reunião de Sprint Planning. Ele deve ser detalhado, ter critérios de aceitação claros e estar planejado seu tempo de entrega. Caso um dos itens de alta prioridade não chegue à reunião de Sprint Planning preparado de acordo com a Definição de Preparado, o item será rejeitado pelo Time de Desenvolvimento.

### Definição de Pronto (DoD)

A Definição de Pronto é um acordo entre o Product Owner e o Time de Desenvolvimento que estabelece os critérios mínimos para que um item do backlog ou um incremento do produto seja considerado "pronto". Isso garante que tudo o que foi desenvolvido durante o Sprint está em condições de ser entregue, com qualidade e funcionalidade adequadas. Sem uma DoD clara, corre-se o risco de entregar trabalho incompleto ou com defeitos.

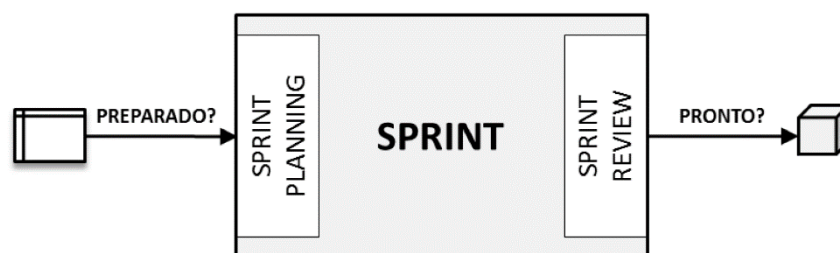


Imagem retirada do Livro: Scrum – Gestão Ágil para Projetos de Sucesso

### Pronto?

Um desenvolvedor de software entrega uma funcionalidade que considera pronta. Ao começar a experimentar o que lhe foi entregue, o

usuário dessa funcionalidade encontra um erro. Ele então pergunta ao desenvolvedor:

- Você testou isso?

Ao que o desenvolvedor responde:

- Não...

- Mas você não disse que está pronto?

- Sim, está pronto. Só falta testar. – afirma o desenvolvedor.

- Se falta testar, não está pronto!

Situações como essa são comuns em projetos onde não há uma compreensão conjunta do que “estar pronto” significa.

## User Story

User Story é uma forma concisa e simples de descrever uma funcionalidade ou necessidade de negócio sob a perspectiva do usuário final. Ela segue geralmente o formato: "Como [usuário], eu quero [funcionalidade] para [benefício]".

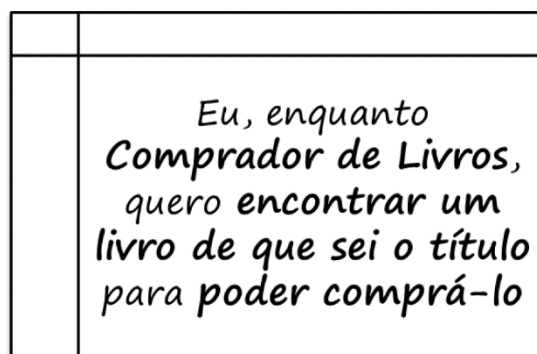


Imagem retirada do Livro: Scrum – Gestão Ágil para Projetos de Sucesso

A User Story é apenas uma promessa de uma conversa, um lembrete de que mais detalhes serão necessários, e não deve ser considerada suficiente para a realização do trabalho.

## Story Point

Story Point é uma unidade de medida relativa usada pelo Time de Desenvolvimento para estimar o esforço necessário para implementar um item do

backlog. Diferente de horas ou dias, story points avaliam a complexidade, o risco e o volume de trabalho de forma comparativa entre os itens. Essa abordagem facilita o planejamento e a previsão de entregas.

### **Para que serve estimar?**

Uma estratégia muito comum para se obter um Product Backlog planejável é aquela em que o Time de Desenvolvimento estima cada um de seus itens individualmente e então somam-se essas estimativas até o suficiente para que se possa planejar o próximo Sprint ou a próxima Release. Ou seja, estimar pode ajudar a planejar.

As Story Points costumam ter um valor não linear (1, 2, 3, 5, 8, 13, 20, 40 e 100), parecidos com a sequência de Fibonacci, que ajudam a medir relativamente o esforço, complexidade e risco de uma tarefa, onde os números maiores (20, 40 e 100) representam tarefas extremamente complexas ou incertas.

### **Velocidade**

Velocidade é uma forma de medir a quantidade de trabalho que o Time de Desenvolvimento consegue realizar dentro de uma Sprint, sendo utilizados Story Points ou alguma unidade de medida para estimar o tempo necessário para realizar uma determinada tarefa.

Conhecer a Velocidade do Time de Desenvolvimento pode ajudar a tornar o Product Backlog planejável.

### **Impedimento**

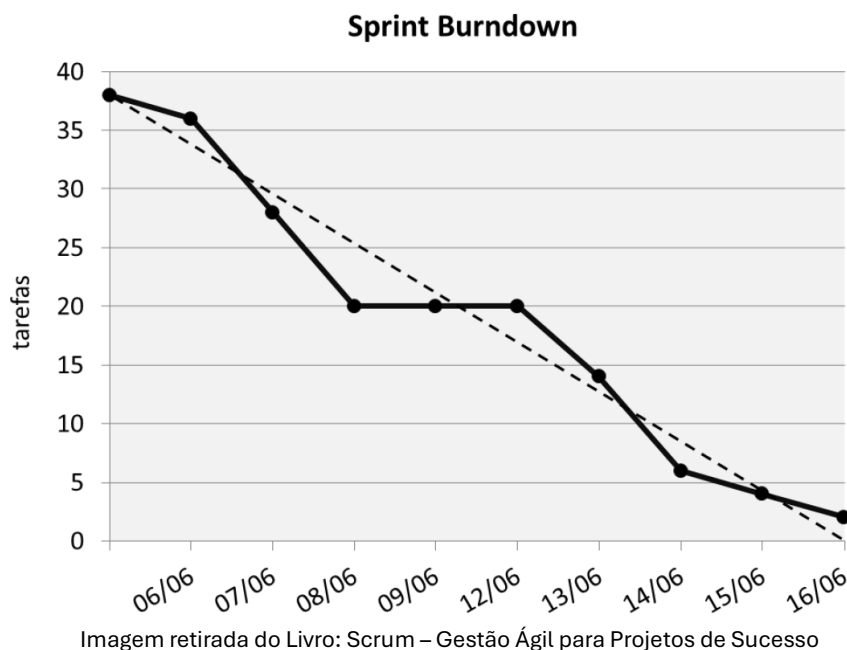
Um impedimento é qualquer obstáculo ou barreira que dificulta ou paralisa o trabalho do Time de Desenvolvimento, ameaçando a meta do Sprint. Informa-se o impedimento ao ScrumMaster assim que ele é identificado, de forma que possa ser tratado o mais rapidamente possível. O objetivo de se informar, durante a reunião de Daily Scrum, quais impedimentos surgiram desde a última reunião é apenas o de dar visibilidade a todos os membros do Time de Desenvolvimento sobre o que atrapalhou ou ainda atrapalha o trabalho.

Como por exemplo, um imprevisto que desorganize o tempo reservado para determinada tarefa pode afetar o andamento das atividades da equipe, atrasando o processo e o alcance do objetivo.

## Burndown

O Burndown é um gráfico mantido e utilizado pelo Time de Desenvolvimento para monitorar seu progresso no desenvolvimento em direção ao final de um Sprint. Ele mostra o trabalho restante estimado para um Sprint em cada dia de trabalho para o desenvolvimento do produto no Sprint. Embora não seja parte integrante do framework Scrum, o Gráfico de Sprint Burndown é a maneira mais rápida e prática de se visualizar o andamento do Sprint.

Há também uma linha ideal, opcional, que serve como guia visual para o comportamento do Gráfico de Burndown ao longo do Sprint. A figura mostra um exemplo de gráfico Burndown ao final de um Sprint bem sucedido.





## Burnup

O Burnup, assim como o Burndown, é um gráfico mantido pelo Product Owner e utilizado tanto por ele quanto pelo Time de Desenvolvimento para monitorar o progresso no desenvolvimento em direção a uma entrega. O Gráfico de Burnup também não é parte do framework Scrum, mas pode ser útil para acompanhar o andamento da entrega do produto.

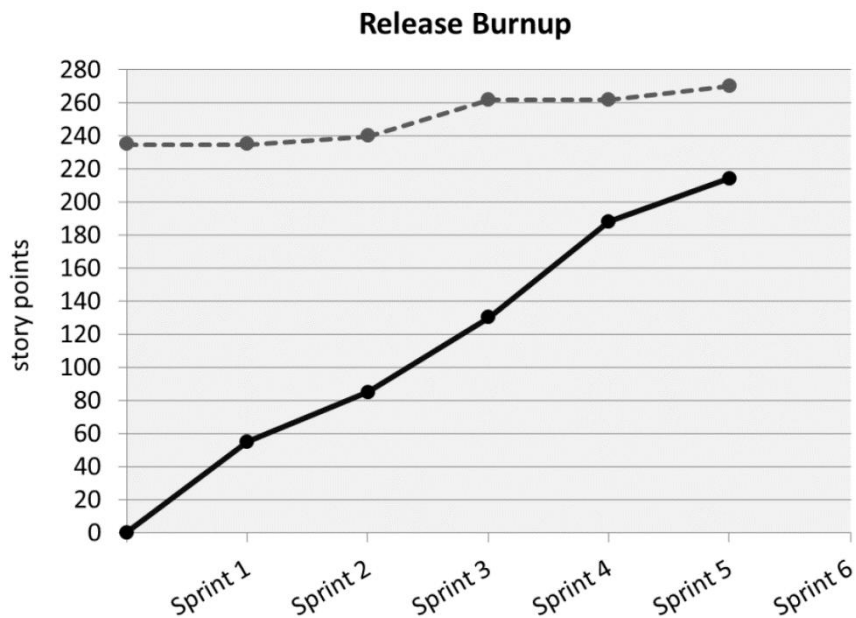


Imagem retirada do Livro: Scrum – Gestão Ágil para Projetos de Sucesso