



# METODOLOGIAS ÁGEIS

AULA 3



Profª Mariana Lucia Kalluf Dakkache Leal



## CONVERSA INICIAL

### Projetos ágeis

No mundo acelerado do desenvolvimento de *software*, é primordial adotar abordagens ágeis para gerenciar projetos com eficiência e alcançar resultados de alta qualidade. O método Scrum tem se destacado como uma metodologia popular e eficaz para equipes que buscam flexibilidade, transparência e entrega contínua de valor. Já tivemos uma introdução ao Scrum. Agora, aprofundaremos nosso conhecimento sobre o assunto, desde sua contextualização até a coleta de requisitos, as fases da *sprint*, o *backlog* de produtos e as *user stories*, além da análise das reuniões essenciais e dos papéis cruciais desempenhados.

### TEMA 1 – MÉTODO SCRUM: CONTEXTUALIZAÇÃO

Para Sutherland e Sutherland (2019, p. 17), “Scrum é uma metodologia ágil para gerenciar projetos complexos, em que não se conhece todas as etapas ou necessidades. Ela se baseia em valores, princípios e práticas que estimulam a colaboração, a criatividade e a adaptação às mudanças.”



Crédito: EamesBot/Shutterstock.



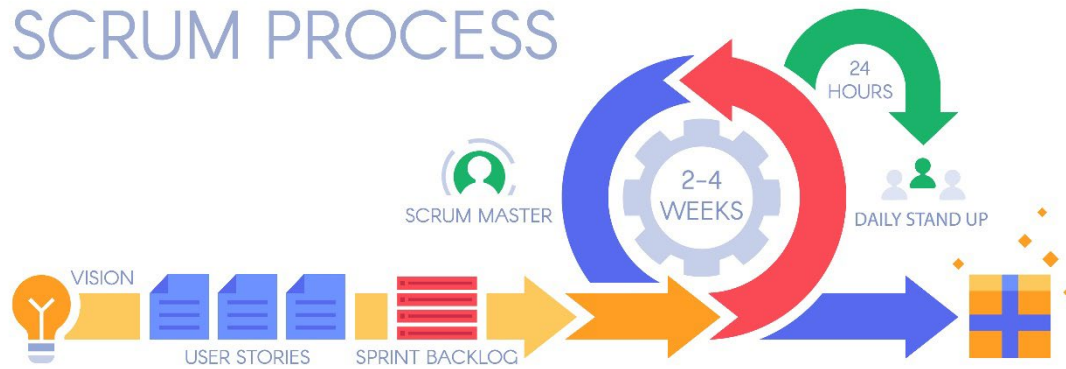
Já de acordo com Pacheco e Prikladnicki (2011, p. 33), “Scrum é uma metodologia de desenvolvimento de software ágil que tem como principal objetivo o desenvolvimento de um produto de software de forma iterativa e incremental, em ciclos curtos de tempo denominados *sprints*”. Como vimos anteriormente, o Scrum é um *framework* ágil amplamente utilizado no desenvolvimento de *software* e em outros projetos complexos e que surgiu no início dos anos 1990, quando Jeff Sutherland e Ken Schwaber (2014) criaram essa abordagem para melhorar a eficiência e a qualidade dos projetos. Seus princípios são transparência, inspeção e adaptação. Nessa metodologia, o projeto é dividido em iterações, chamadas de *sprints*, que possuem uma duração fixa, geralmente de uma a quatro semanas. Durante uma *sprint*, a equipe de desenvolvimento se concentra em um conjunto específico de itens de trabalho, que são selecionados do *backlog* do produto.

O *product owner* é responsável por gerenciar e priorizar, em sua relação com o cliente, a lista de requisitos do produto. Essa lista de requisitos é chamada de *product backlog* (*backlog* do produto). No início de cada *sprint*, o *product owner* e a equipe de desenvolvimento realizam a reunião de planejamento da *sprint* (*sprint planning*), em que definem os objetivos dessa *sprint* e os itens de trabalho. Em seguida, a equipe de desenvolvimento estima o esforço e a quantidade de itens que farão parte da *sprint*, levando em consideração sua capacidade e velocidade. Durante a *sprint*, são realizadas reuniões diárias (*daily Scrum* ou *stand-up meetings*), de curta duração, para que a equipe de desenvolvimento compartilhe suas entregas, o que está fazendo e se tem algum impedimento para realizar suas tarefas.

O *Scrum master* atua como facilitador, removendo obstáculos e ajudando a equipe a seguir as práticas ágeis. Ao término da *sprint*, a equipe de desenvolvimento apresenta suas entregas para validação do *product owner*, na reunião de revisão da *sprint*, e em seguida é realizada uma retrospectiva da *sprint*, na qual a equipe reflete sobre o processo e identifica melhorias para as próximas *sprints*.



# SCRUM PROCESS



Crédito: Tartila/Shutterstock.

O Scrum permite uma abordagem iterativa e incremental, possibilitando a entrega contínua de valor ao cliente. Ele oferece transparência sobre o progresso do projeto, permitindo uma maior visibilidade e controle sobre o trabalho realizado. Além disso, o Scrum promove a colaboração, a comunicação eficaz e a adaptabilidade, propiciando que a equipe se ajuste rapidamente a mudanças de requisitos e necessidades do cliente.

## TEMA 2 – COLETA DE REQUISITOS DO MÉTODO SCRUM

A coleta de requisitos é uma etapa fundamental para o método Scrum, pois garante que o produto atenda às necessidades dos *stakeholders* (clientes, usuários finais, especialistas do domínio e outros envolvidos no projeto) e que a equipe tenha total entendimento sobre o que precisa ser desenvolvido. Embora o Scrum seja uma metodologia ágil que valorize a adaptabilidade e a flexibilidade, a coleta de requisitos ainda desempenha um papel importante na definição do escopo do projeto. Essa etapa normalmente se inicia com a definição dos *stakeholders* e a compreensão de suas necessidades e expectativas em relação ao produto.

Uma prática comum no Scrum é a criação do *product backlog*. O gerenciamento, priorização e atualização do *product backlog* é realizado pelo *product owner*, que utiliza o *feedback* dos *stakeholders* e as mudanças nas necessidades do produto para executar essas ações.

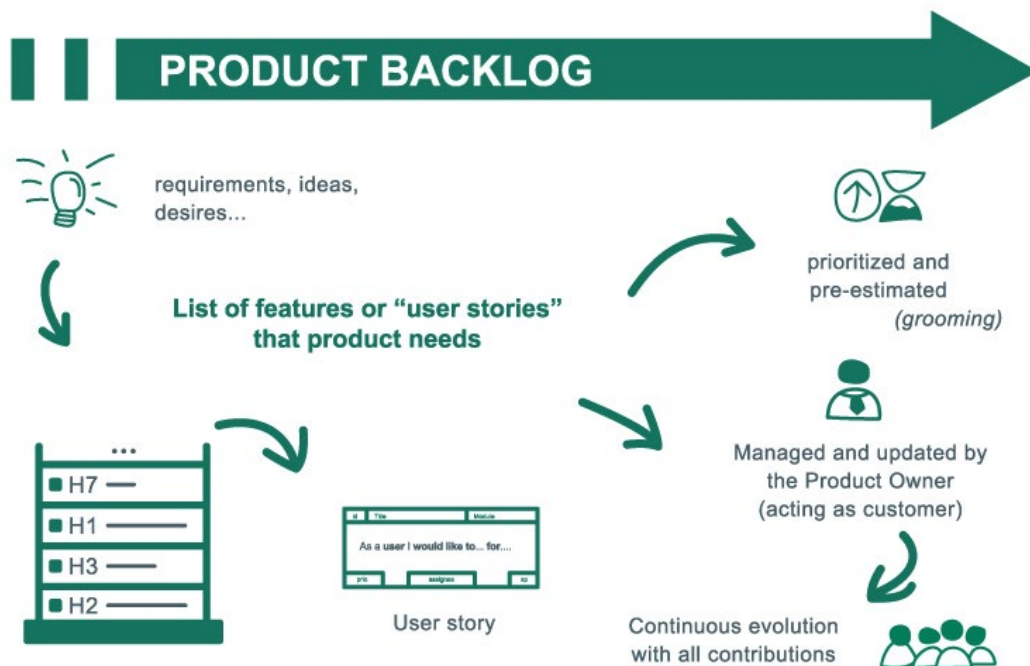


Crédito: Tartila/Shutterstock.

A coleta de requisitos no Scrum é um processo contínuo, iterativo e adaptativo. Durante a *sprint planning*, o *product owner* e a equipe de desenvolvimento revisam os itens do *product backlog* e selecionam as *user stories* que serão abordadas na próxima *sprint*. Durante a *sprint*, podem surgir dúvidas ou a necessidade de complementar ou alterar algum requisito do projeto e, para esclarecer esses pontos, é preciso que a equipe de desenvolvimento trabalhe junto ao *product owner*, de maneira colaborativa.

## 2.1 Product backlog

O *product backlog*, elemento-chave da metodologia Scrum, é a lista priorizada de requisitos, funcionalidades e melhorias desejadas para o produto em desenvolvimento. Essa lista é gerenciada pelo *product owner* e serve como um guia para a equipe de desenvolvimento, durante o projeto.



Crédito: AmaiaL/Shutterstock.

Nessa lista, os itens são elencados de acordo com sua priorização, sendo que os mais importantes ficam no topo. A classificação dos itens é realizada de acordo com as necessidades e expectativas dos *stakeholders*. O *product owner* deve entender as demandas recebidas dos *stakeholders* e garantir que os itens mais importantes sejam resolvidos primeiro. O *product backlog* é composto por itens chamados de *user stories* (*histórias de usuário*), que são breves descrições de funcionalidades ou requisitos, do ponto de vista do usuário final. As *user stories* devem ser objetivas e sucintas e fornecer um contexto compreensível para a equipe de desenvolvimento. No decorrer do projeto, o *product backlog* é regularmente revisado e atualizado, de acordo com o *feedback* recebido dos *stakeholders* e das mudanças efetivadas no produto. Com essa análise, é possível que *user stories* sejam excluídas, incluídas ou alteradas.

Na reunião de planejamento da *sprint*, a equipe de desenvolvimento revisa o *product backlog* e seleciona as *user stories* que serão abordadas no próximo ciclo. Essas *user stories* são, então, detalhadas em tarefas mais específicas, na *sprint backlog*.

## 2.2 Técnicas de coleta de requisitos no método Scrum

As técnicas de coleta de requisitos no Scrum são essenciais para se obter uma compreensão clara e detalhada das necessidades dos *stakeholders*. Essas





técnicas visam facilitar a colaboração, a comunicação e a obtenção de *feedback* contínuo ao longo do processo de desenvolvimento. A seguir, apresentaremos as principais técnicas de coleta de requisitos utilizadas na metodologia Scrum:

- **Entrevistas com *stakeholders*:** realizam-se entrevistas individuais ou em grupo com os *stakeholders*, para entender suas necessidades, expectativas e requisitos específicos ao projeto. Com base nessas entrevistas, é possível identificar requisitos funcionais e não funcionais, além de sanar eventuais dúvidas e obter informações adicionais.
- **Workshops colaborativos:** sessões interativas, com a participação de *stakeholders*, membros da equipe Scrum e especialistas relevantes. Nesses *workshops*, a coleta de requisitos pode ocorrer por meio de *brainstorming*, jogos, dinâmicas de grupo e técnicas de visualização.
- **Observação do usuário:** observa-se o uso real do produto ou do sistema pelos usuários finais. Com essa técnica, a equipe compreende melhor os fluxos de trabalho, necessidades e dificuldades dos usuários, bem como identifica oportunidades de melhoria.
- **Prototipação:** criam-se protótipos rápidos e iterativos, que auxiliam na validação e refinamento dos requisitos. Essa técnica permite que o *stakeholder* visualize e interaja com as funcionalidades do produto antecipadamente e, com isso, seja possível realizar os ajustes e aprimoramentos devidos nos requisitos.
- **Feedback contínuo:** nas cerimônias do Scrum (reuniões diárias, revisões de *sprint* e retrospectivas), os *stakeholders* podem fornecer *feedback* sobre as entregas realizadas, sugerir ajustes e apresentar novas necessidades ou mudanças de requisitos.

Vale ressaltar que as técnicas de coleta de requisitos no Scrum são adaptáveis e podem variar de acordo com as necessidades e particularidades de cada projeto. A escolha das técnicas mais adequadas dependerá do contexto, dos *stakeholders* envolvidos e da natureza do produto em desenvolvimento. A questão principal é garantir uma comunicação eficaz e uma compreensão clara das necessidades existentes, para se guiar o desenvolvimento do produto de forma ágil e satisfatória para todas as partes interessadas.



## TEMA 3 – ESTRUTURA DO SCRUM

### 3.1 Sprint

Uma *sprint*, no Scrum, é um período fixo e curto, durante o qual a equipe de desenvolvimento trabalha para entregar um incremento do produto funcional e de valor. É um ciclo de trabalho definido, no qual são realizadas atividades de planejamento, execução, revisão e retrospectiva do projeto.

O Scrum muda a forma como você pensa no tempo. Depois de participar de alguns *sprints* e reuniões diárias, você para de enxergar o tempo como uma seta apontando para o futuro e passa a vê-lo como algo fundamentalmente cíclico. Cada *sprint* é uma oportunidade de criar algo totalmente novo; cada dia é chance de melhorar. O Scrum promove uma visão de mundo holística. Quem se engaja nele passa a valorizar cada momento como um ciclo de respiração de vida em constante movimento. (Sutherland, 2016, p. 84)

A *sprint* é uma unidade fundamental no Scrum, proporcionando um ritmo consistente e previsível para o desenvolvimento do produto. Durante o ciclo da *sprint*, a equipe se compromete a atingir as metas estabelecidas, trabalhando de forma colaborativa e auto-organizada para cumprir os objetivos propostos.



Crédito: Buravleva Stock/Shutterstock.

A *sprint* tem sua duração definida como um período fixo, que deve ser consistente ao longo do projeto. A escolha da duração adequada é importante para manter o foco da equipe, promover a entrega contínua de valor e facilitar a previsibilidade do trabalho realizado. Geralmente, uma *sprint* dura de uma a





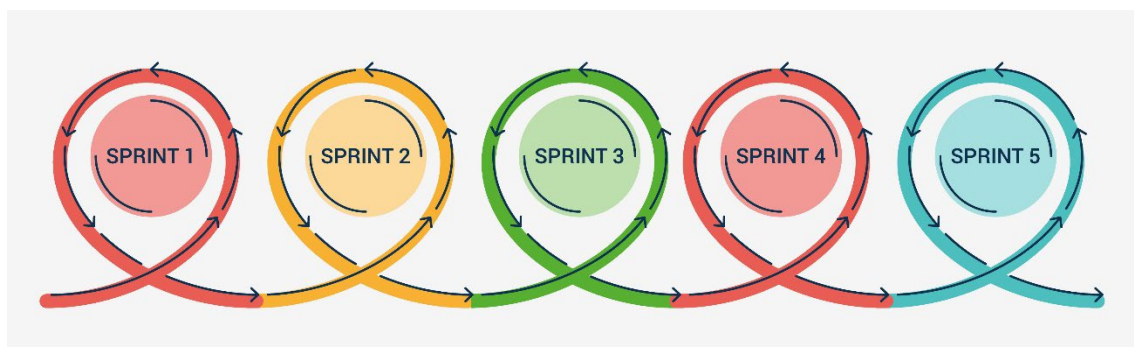
quatro semanas, sendo que o período mais comum é de duas semanas para cada ciclo. No entanto, a equipe Scrum tem autonomia para determinar a duração da *sprint* com base nas necessidades do projeto, na complexidade das tarefas, na capacidade da equipe e em outros fatores relevantes. É importante que a duração da *sprint* seja a mesma durante o projeto, para facilitar a previsibilidade e permitir a comparação dos resultados entre as *sprints*. Alterar a duração da *sprint* frequentemente pode causar interrupções no fluxo de trabalho e na capacidade de entrega de resultados da equipe.

### 3.1.1 Elementos

Existem três elementos-chave associados a cada *sprint*:

1. **Objetivo da *sprint*:** descrição concisa do que a equipe pretende alcançar durante a *sprint*. O objetivo da *sprint* fornece direção e orientação para a equipe, ajudando-a a se concentrar nas entregas prioritárias do projeto e no valor a ser entregue ao final do ciclo.
2. **Product backlog:** lista dos itens priorizados que descrevem as funcionalidades, requisitos e tarefas a serem entregues. O *backlog* do produto pode ser ajustado e atualizado ao longo da *sprint*, mas sempre focando no objetivo definido.
3. **Incremento do produto:** soma de todos os itens do *backlog* do produto concluídos durante a *sprint*, juntamente com o trabalho realizado nas *sprints* anteriores. Ao término da *sprint*, a equipe deve entregar um incremento do produto funcional e de valor que esteja em um estado utilizável e pronto para ser entregue aos *stakeholders*.

### 3.1.1 Funcionamento da *sprint*



Crédito: KPDMedia/Shutterstock.



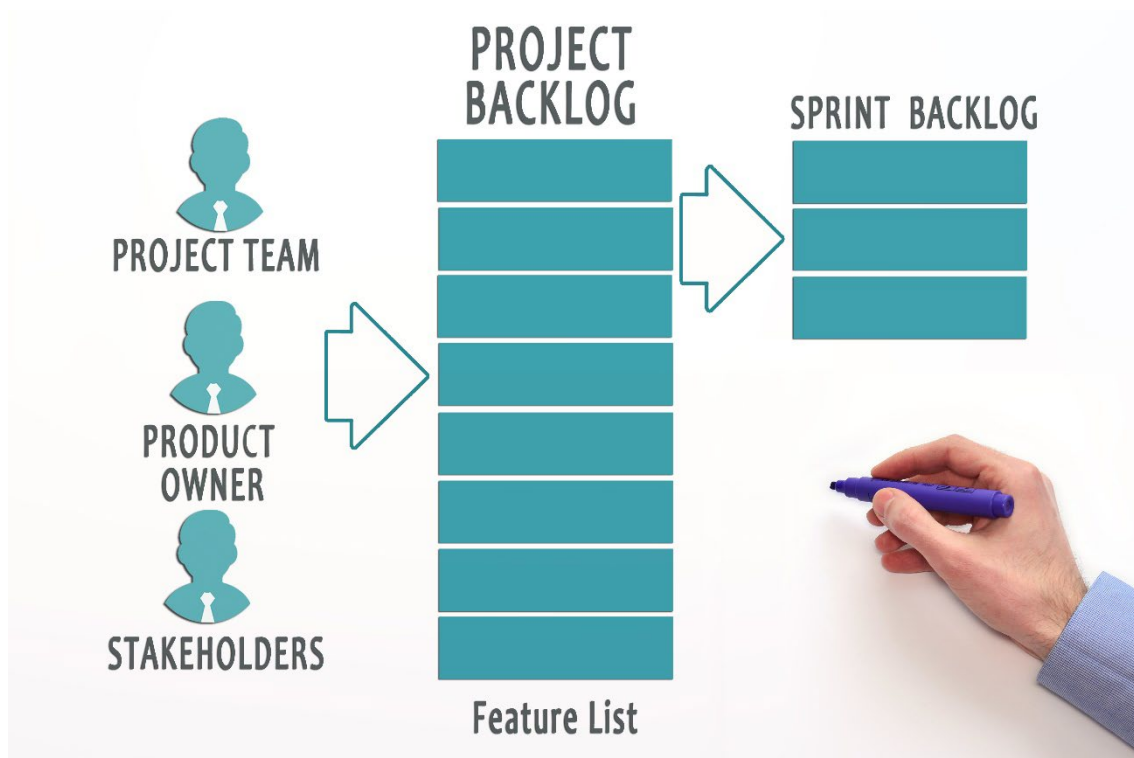
O funcionamento da *sprint* pode ser explicado em etapas, que consistem em:

- a. **Planejamento da *sprint*:** a equipe realiza uma reunião de planejamento da *sprint*, na qual são definidos o objetivo da *sprint* e os itens do *backlog* do produto que serão trabalhados. Os itens definidos são detalhados em *user stories*.
- b. **Execução da *sprint*:** a equipe trabalha de forma colaborativa e auto-organizada para desenvolver os incrementos do produto, realizando reuniões diárias curtas (*daily Scrum*), para o melhor alinhamento do projeto.
- c. **Refinamento e adaptação do *backlog* do produto:** no decorrer da *sprint*, o *product owner* e a equipe do projeto podem identificar a necessidade de ajuste do *backlog* do produto para que seja entregue o máximo valor possível, no término da *sprint*.
- d. **Revisão da *sprint*:** ao final da *sprint*, a equipe realiza uma reunião de revisão da *sprint* para apresentar os itens desenvolvidos e, com isso, o *stakeholder* tirar suas dúvidas e dar o seu *feedback* sobre o incremento entregue.
- e. **Retrospectiva da *sprint*:** na sequência da reunião de revisão, a equipe faz uma reflexão sobre os acontecimentos da *sprint* e identifica possíveis pontos de melhoria e que práticas devem ser mantidas.
- f. **Início de uma nova *sprint*:** após o encerramento de uma *sprint*, inicia-se uma nova, repetindo-se o ciclo de planejamento, execução, revisão e retrospectiva.

### 3.2 Sprint *backlog*

O *sprint backlog* (*backlog* da *sprint*) é uma lista de itens de trabalho que a equipe seleciona do *backlog* do produto para serem desenvolvidos durante uma *sprint* específica. A equipe decide quais itens podem ser concluídos dentro do período da *sprint* e os adiciona ao *sprint backlog*.

O objetivo do *backlog* da *sprint* é tornar visível o trabalho necessário para que a equipe de Desenvolvimento atinja a meta da *sprint*. Para isso, os membros dessa equipe (e somente eles) podem adicionar novas tarefas, caso descubram, no decorrer da *sprint*, que mais trabalho será necessário. Da mesma forma, também podem remover tarefas, caso se mostrem desnecessárias. No entanto, é importante



Crédito: Astel Design/Shutterstock.

### 3.2.1 Características

Algumas características importantes do *sprint backlog* são:

- a. **Detalhamento das tarefas:** os itens do *sprint backlog* são detalhados em tarefas específicas e atribuídos ao participante da equipe responsável pela execução.
- b. **Estimativas de esforço:** normalmente, as tarefas de *sprint backlog* possuem uma estimativa de esforço para desenvolvimento definida, para ajudar a equipe a planejar e acompanhar seu progresso durante a *sprint*.
- c. **Priorização:** os itens do *sprint backlog* são organizados em ordem de prioridade, com base no valor que cada item agrega ao produto. Isso permite que a equipe se concentre nos itens mais importantes e entregue valor, ao final da *sprint*.
- d. **Atualização contínua:** no decorrer da *sprint*, o *sprint backlog* pode ser atualizado à medida que a equipe adquire mais informações, enfrenta obstáculos ou identifica mudanças necessárias. Essa atualização



contínua permite que a equipe se adapte e ajuste seu plano para atingir o objetivo da *sprint*.



Crédito: Julslst/Shutterstock.

Na reunião de revisão da *sprint*, o *sprint backlog* é revisado e as tarefas não concluídas retornam ao *backlog* do produto para serem reavaliadas e priorizadas em *sprints* posteriores.

### 3.3 User story

A *user story* (história do usuário) é uma ferramenta essencial para capturar requisitos de forma clara e concisa e possibilita que a equipe de desenvolvimento entenda as necessidades dos usuários e consiga propor soluções efetivas, no projeto. Na *user story*, temos a descrição simplificada de uma funcionalidade ou requisito do sistema, escrita na perspectiva do usuário. Sua estrutura inclui o ator, a ação desejada e o objetivo final. Por exemplo, uma *user story* poderia ser escrita da seguinte maneira: *Como um usuário, eu quero fazer login no sistema, para acessar minhas informações pessoais.*



Como [Quem?]  
eu quero [O quê?]  
para [Por quê?]

A habilidade de especificar os requisitos com maior eficiência entre os membros da equipe é, sem dúvida nenhuma, uma das maiores vantagens das *user stories*, que são escritas utilizando-se uma linguagem acessível e sem termos técnicos, tornando-se compreensíveis para desenvolvedores, *designers*, testadores e demais *stakeholders*. Com isso, as equipes podem direcionar sua atenção às necessidades do usuário, sem demandar tempo para a especificação técnica, no início do levantamento de requisitos. Essa abordagem evita a criação de soluções excessivamente complicadas e direciona o desenvolvimento para soluções mais simples e eficientes.

Outro aspecto importante das *user stories* é a sua flexibilidade. Elas são pequenas e podem ser facilmente priorizadas e reorganizadas em um *backlog*, permitindo que a equipe responda rapidamente às mudanças de requisitos e prioridades. Isso é particularmente valioso em projetos ágeis, nos quais a adaptabilidade é fundamental. No entanto, é importante ressaltar que as *user stories* devem ser complementadas por conversas e discussões entre os membros da equipe. Elas não devem ser consideradas como uma especificação completa e detalhada por si só, mas como uma ferramenta para iniciar conversas e colaboração entre os envolvidos no projeto.

#### TEMA 4 – MEETINGS

No Scrum, existem algumas reuniões-chave, que são realizadas durante cada *sprint* do projeto. Essas reuniões são projetadas para promover a colaboração, a comunicação e a transparência entre os membros da equipe Scrum e os *stakeholders* envolvidos.



As principais reuniões do Scrum são:

- *sprint planning*;
- *daily Scrum*;
- *sprint review*;
- *sprint retrospective*.

Além dessas reuniões principais, outras reuniões podem ocorrer durante o ciclo do projeto, dependendo das necessidades da equipe e do contexto específico. Por exemplo, reuniões de refinamento do *backlog* (*backlog refinement*) podem ser realizadas para discutir e priorizar as *user stories* futuras, e reuniões de revisão de produto (*product review*) podem ser realizadas para compartilhar atualizações, com os *stakeholders*, sobre o progresso geral do produto. É importante lembrar que as reuniões, no Scrum, são projetadas para serem rápidas, focadas e eficientes, e nelas a participação de todos os membros da equipe é fundamental para o sucesso do processo.

#### 4.1 *Sprint planning*

O *sprint planning* ocorre no início de cada *sprint*, sendo uma oportunidade para a equipe Scrum e o *product owner* colaborarem na definição do objetivo da *sprint* e no planejamento das atividades que serão realizadas. Essa reunião é essencial para estabelecer um entendimento compartilhado sobre as metas e expectativas da equipe para a *sprint*.

Durante o *sprint planning*, a equipe Scrum revisa os itens do *product backlog*. Em conjunto com o *product owner*, a equipe seleciona as *user stories* que serão trabalhadas durante a *sprint*. O *product owner* esclarece os detalhes dessas *user stories*, fornecendo os critérios de aceitação e respondendo às perguntas dos membros da equipe. Com as *user stories* selecionadas, a equipe Scrum estima o esforço necessário para completar cada uma delas. Essa estimativa pode ser feita por meio do *planning poker*, que explicaremos a seguir. Uma vez que as *user stories* estejam estimadas, a equipe Scrum se reúne para definir as tarefas necessárias para implementar cada uma delas. Essas tarefas são registradas em um quadro *kanban* ou em qualquer outra ferramenta de gestão escolhida pela equipe. Essa fase de planejamento detalhado permite que a equipe tenha uma visão clara do trabalho a ser realizado e possa se organizar adequadamente para isso.





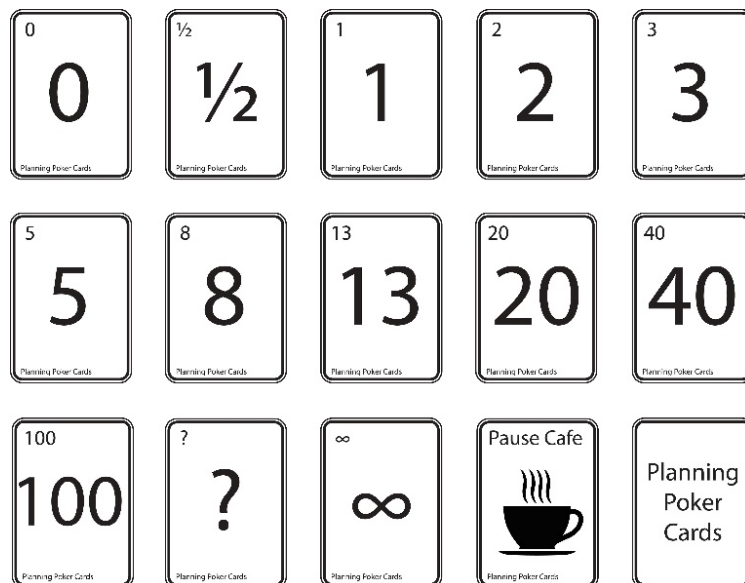
O resultado do *sprint planning* é o plano detalhado da *sprint*, que inclui o objetivo da *sprint*, as *user stories* selecionadas, as estimativas de esforço e as tarefas definidas. Esse plano serve como guia para a equipe, durante a *sprint*, fornecendo uma estrutura clara para a execução das atividades. A importância do *sprint planning* está na sua capacidade de alinhar a equipe Scrum e o *product owner* em relação aos objetivos e prioridades da *sprint*. Ele permite que a equipe compreenda o que deve ser realizado e como será feito, promovendo a transparência, a colaboração e o engajamento de todos os membros.

#### 4.1.1 Planning poker

O *planning poker* facilita a estimativa colaborativa e precisa do esforço necessário para concluir as *user stories* ou tarefas, em uma abordagem ágil. Ao usar o *planning poker*, a equipe Scrum é capaz de obter uma compreensão compartilhada da complexidade e do tamanho relativo do trabalho. Por meio da discussão e da atribuição de pontos de estimativa por cada membro da equipe, o *planning poker* promove a transparência, a participação igualitária e a convergência em estimativas mais confiáveis. Esse processo ajuda a equipe a planejar o trabalho de forma mais precisa e assim tomar boas decisões sobre a capacidade de entrega em cada *sprint*.

A ideia é simples, cada pessoa tem um baralho de cartas com estes interessantíssimos números de Fibonacci: 1, 2, 3, 5, 8, 13 etc. Escolhe-se uma tarefa por vez para ser avaliada. Então, cada integrante do grupo separa a carta que considera corresponder à quantidade de esforço exigida por aquela tarefa e a coloca virada para baixo na mesa. Em seguida, todo mundo vira sua carta ao mesmo tempo. Se as opiniões de todos estiverem a uma distância de até duas cartas umas das outras (digamos, há 5, dois, 8 e um 13), a equipe apenas soma esses números, tira a média (neste caso 6,6) e passa para o próximo item. Lembre-se de que estamos falando de estimativas, não de cronogramas rígidos. Além disso, trata-se de previsões para pequenas partes do projeto.

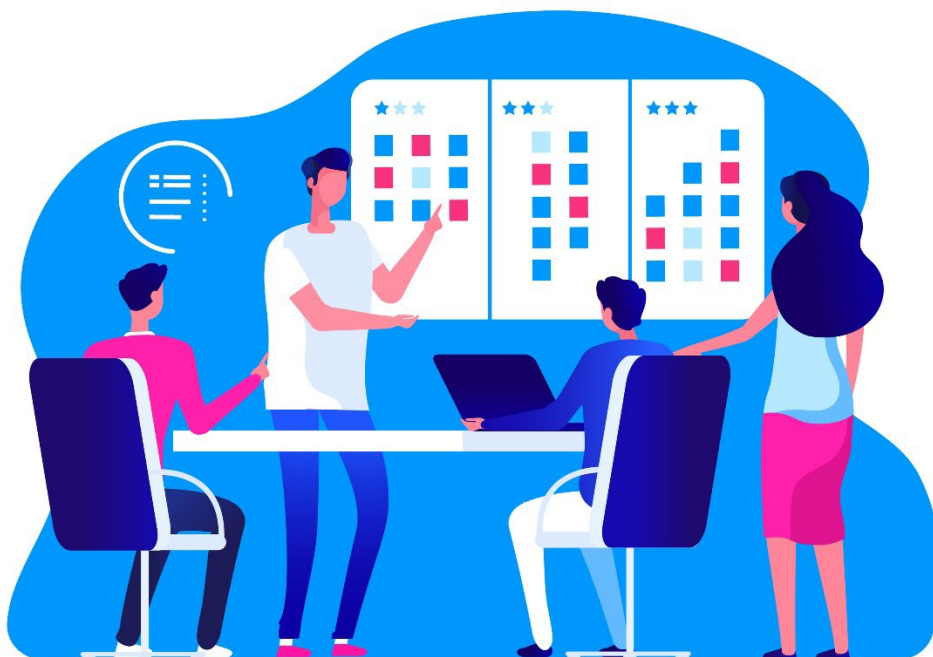
Se as cartas escolhidas estiverem a uma distância de mais de três números presentes na sequência, quem selecionou a mais alta e a mais baixa explica seu raciocínio. Então, todo mundo faz outra rodada. Caso contrário, se calcula a média das estimativas, que ficarão próximas do resultado encontrado pelas estatísticas da Rand Corporation. (Sutherland, 2016, p. 129)



Crédito: Gstraub/Shutterstock.

## 4.2 Daily Scrum

O *daily Scrum* é uma reunião curta e diária, com duração máxima de 15 minutos, realizada no mesmo horário e local todos os dias, preferencialmente no início do dia de trabalho. Seu principal objetivo é manter a equipe Scrum atualizada sobre o andamento do projeto, seus possíveis impedimentos e, com isso, se planejar as tarefas do próximo período.



Crédito: Net Vector/Shutterstock.



Durante o *daily Scrum*, cada membro da equipe responde a três perguntas básicas:

1. *O que fiz desde o último daily Scrum?*
2. *O que planejo fazer até o próximo daily Scrum?*
3. *Existem obstáculos ou impedimentos que estão me atrapalhando?*

Como a reunião tem duração máxima de 15 minutos, as discussões ou soluções mais complexas devem ser tratadas em uma outra reunião mais específica, pois, nessa reunião, cada colaborador deve ter por objetivo apenas manter os integrantes da equipe atualizados e compartilhar com eles o progresso das atividades planejadas, permitindo que eles identifiquem possíveis conflitos, sobreposições ou dependências, facilitando a resolução proativa de problemas. Além disso, o *daily Scrum* também ajuda a manter o foco no objetivo da *sprint* e a acompanhar o cumprimento das metas estabelecidas, promovendo a responsabilidade individual e coletiva, incentivando os membros da equipe a assumirem compromissos e a se apoiarem mutuamente para todos alcançarem os objetivos definidos.

Ao término do *daily Scrum*, a equipe deve ter uma visão clara do progresso do projeto, dos próximos passos e dos desafios a serem superados. Essa reunião diária ajuda a manter a equipe alinhada, a melhorar a comunicação, a incentivar a colaboração e a maximizar a eficiência do trabalho.

### 4.3 *Sprint review*

A *sprint review* (revisão da *sprint*) ocorre no final de cada *sprint* e é uma oportunidade para a equipe Scrum compartilhar com os *stakeholders* o trabalho concluído durante a *sprint*. Essa reunião é uma ocasião para a equipe demonstrar as funcionalidades implementadas e obter *feedback* sobre o que foi entregue. Durante a *sprint review*, a equipe apresenta as *user stories* ou funcionalidades concluídas durante a *sprint*, o que é uma oportunidade para os *stakeholders* verem o progresso alcançado, validar se o trabalho atende às suas expectativas e fornecer *feedback* sobre a funcionalidade implementada. O *product owner* também está presente para explicar o contexto e o objetivo de cada funcionalidade.



Crédito: Vector Stock Pro/Shutterstock.

Outro objetivo da *sprint review* é avaliar se as metas estabelecidas no início da *sprint* foram alcançadas. A equipe Scrum e o *product owner* têm a oportunidade de verificar se o trabalho concluído atendeu aos critérios de aceitação definidos e se está alinhado com o objetivo da *sprint*. Isso permite que a equipe aprenda com o processo e faça ajustes para melhorar a entrega do projeto, no futuro.

A *sprint review* não é apenas uma apresentação passiva do trabalho concluído, mas também uma oportunidade para a equipe Scrum e os *stakeholders* colaborarem na definição das próximas prioridades. Com base no *feedback* recebido durante a revisão, a equipe pode atualizar o *product backlog* e priorizar as novas funcionalidades de acordo com as necessidades e demandas dos *stakeholders*. Vale ressaltar que, na *sprint review*, não devem ser discutidos detalhes técnicos ou a solução dos problemas. Caso surjam questões complexas no decorrer da reunião, elas podem ser registradas como requisitos para futuras iterações ou tratadas em discussões posteriores.

#### 4.4 *Sprint retrospective*

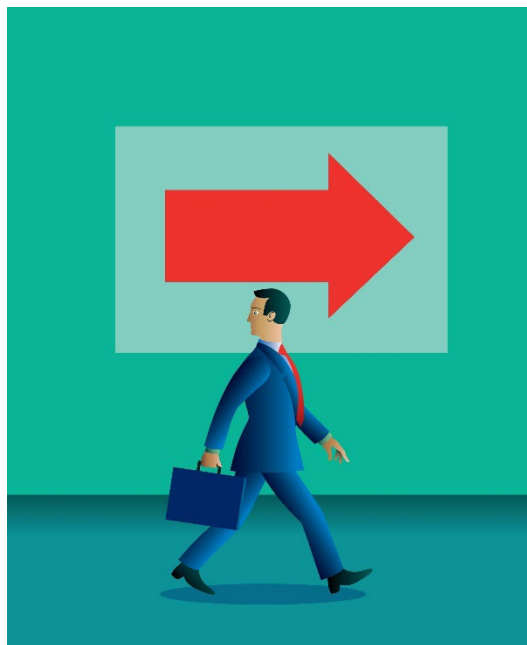
A *sprint retrospective* (retrospectiva da *sprint*) ocorre no final de cada *sprint* e é dedicada à reflexão sobre o trabalho realizado. É uma oportunidade



para a equipe Scrum analisar a *sprint* finalizada, em conjunto, e identificar formas de melhorar o processo, o trabalho em equipe e o próprio produto. Nessa reunião, a equipe Scrum revisa a *sprint* passada, analisando o que funcionou bem e o que poderia ter sido feito de maneira diferente. Esse é um momento de honestidade e abertura, no qual os integrantes da equipe são encorajados a expressar suas opiniões, compartilhar suas percepções e contribuir para a discussão.

A retrospectiva geralmente segue uma estrutura de três etapas:

1. **O que deu certo?** – destacam-se os aspectos positivos da *sprint*, reconhecendo os sucessos alcançados e identificando as práticas e processos que funcionaram bem.
2. **O que não deu certo?** – examinam-se os desafios e obstáculos encontrados durante a *sprint*. Essa é uma oportunidade para discutir problemas ou dificuldades que surgiram, identificar suas causas subjacentes e entender como eles podem ser evitados ou superados, no futuro.
3. **O que podemos melhorar?** – identificam-se ações específicas para melhorar o desempenho do produto. Essas ações podem incluir ajustes no processo, mudanças na comunicação ou colaboração ou implementação de práticas que ajudem a equipe a alcançar resultados mais efetivos, na próxima *sprint*.



Crédito: Stan Eales/Shutterstock.



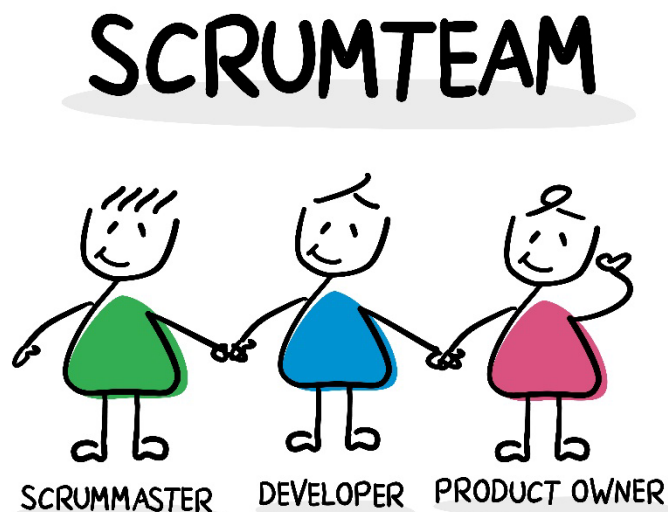
A *sprint retrospective* é uma reunião voltada para o aprendizado contínuo e a melhoria, permitindo que a equipe Scrum analise suas próprias práticas, experimente abordagens diferentes e evolua, com base em suas experiências. Ao realizar retrospectivas regularmente, a equipe Scrum desenvolve uma mentalidade de melhoria contínua, buscando maneiras de aprimorar seu trabalho e maximizar o valor entregue ao cliente. É importante ressaltar que a *sprint retrospective* deve ser um ambiente seguro e de confiança, no qual todos os membros da equipe possam expressar suas opiniões e contribuir para a discussão. A transparência e a honestidade são fundamentais para garantir que os problemas sejam identificados e abordados de forma construtiva, sem culpas ou julgamentos.

## TEMA 5 – PAPÉIS E RESPONSABILIDADES

No Scrum, existem três papéis principais a serem desempenhados:

1. *Scrum master*;
2. *product owner*;
3. equipe de desenvolvimento.

Cada um desses papéis possui responsabilidades específicas, que contribuem para o sucesso do projeto.



Crédito: Gumbart/Shutterstock.





## 5.1 Scrum master

O *Scrum master* “É um membro da equipe que, durante uma *sprint*, garante o bom andamento do projeto, assegurando que seus ritos sejam cumpridos, seus artefatos sejam usados de maneira correta e qualquer obstáculo ao trabalho da equipe seja removido.” (Brod, 2015, p. 48). Ele é responsável por garantir que o Scrum seja compreendido e implementado corretamente pela equipe de desenvolvimento. Suas principais responsabilidades incluem:

- Facilitar as reuniões do Scrum, como o *sprint planning*, o *daily Scrum*, a *sprint review* e a *sprint retrospective*.
- Remover obstáculos que possam impedir o progresso da equipe.
- Garantir que os princípios e práticas do Scrum sejam seguidos.
- Promover a colaboração e a auto-organização da equipe.
- Proteger a equipe de interferências externas e manter um ambiente de trabalho focado e produtivo.

## 5.2 Product owner

O dono do produto é o responsável por gerenciar o *backlog* do produto, garantir o retorno sobre o investimento, definir a visão do produto, gerenciar a entrada de novos requisitos e definir sua ordem, gerenciar o plano de releases, gerenciar o orçamento e riscos do produto ou projeto e aceitar ou rejeitar o que será entregue ao final de cada iteração. Ou seja, ele é o responsável por gerenciar o produto de forma a assegurar o valor do trabalho executado pela equipe de desenvolvimento. O sucesso do produto está relacionado à sua capacidade de compreender as necessidades do negócio e do mercado, de forma que o *backlog* do produto reflita a importância dos seus itens, e de transmitir essas informações ao restante da equipe. (Prikladnicki et al., 2014, p. 27)

O *product owner* é responsável por representar os interesses dos *stakeholders* e definir as necessidades e prioridades do produto. Suas principais responsabilidades incluem:

- Elaborar e manter o *product backlog*, que é a lista de requisitos e funcionalidades do produto.
- Priorizar os itens do *product backlog* com base no valor e nas necessidades dos *stakeholders*.
- Participar das reuniões de planejamento da *sprint* e fornecer orientações claras sobre as funcionalidades a serem implementadas.



- Estabelecer critérios de aceitação para os itens do *product backlog*, garantindo que sejam entregues com qualidade e atendam às expectativas dos *stakeholders*.

### 5.3 Equipe de desenvolvimento

O time de desenvolvimento é envolvido e tem a responsabilidade de entender e extrair os detalhes essenciais para determinar os tamanhos das histórias do *backlog* da versão de entrega, compreendendo suas importâncias e se possível nesse momento determinando a velocidade e prevendo o esforço indispensável para completar o trabalho da versão da entrega. (Cruz, 2015, p. 101)

A equipe de desenvolvimento é responsável por transformar os itens do *product backlog* em incrementos de produtos funcionais. Suas principais responsabilidades incluem:

- Estimar o esforço necessário para implementar os itens do *product backlog*.
- Participar ativamente das reuniões do Scrum, como o *sprint planning*, o *daily Scrum*, a *sprint review* e a *sprint retrospective*.
- Trabalhar de forma colaborativa para desenvolver as funcionalidades do produto, de acordo com as orientações do *product owner*.
- Auto-organizar-se para determinar como o trabalho será realizado a fim de atingir os objetivos da *sprint*.
- Manter um alto padrão de qualidade ao desenvolver o produto e cumprir os critérios de aceitação estabelecidos.

É importante destacar que todos os membros da equipe são responsáveis pelo sucesso do projeto e devem colaborar ativamente para alcançar os objetivos estabelecidos. O *Scrum master*, o *product owner* e a equipe de desenvolvimento trabalham em conjunto, cada um desempenhando seu papel específico, para garantir a entrega de um produto de valor e qualidade aos *stakeholders*.

### FINALIZANDO

Nesta etapa, exploramos o método Scrum, uma abordagem ágil para a gestão de projetos. Começamos entendendo a contextualização do Scrum e como ele se destaca como uma metodologia eficaz no desenvolvimento de *software*. Em seguida, discutimos a importância da coleta de requisitos no Scrum, ressaltando a necessidade de colaboração entre a equipe e os



*stakeholders* para entender as necessidades do produto. Aprofundamos nas fases do Scrum, com destaque para a *sprint*, uma iteração curta e focada, em que a equipe seleciona histórias do *product backlog*, uma lista priorizada de requisitos, para implementar. As histórias representam as funcionalidades ou melhorias desejadas pelo cliente ou usuário final. Falamos também sobre as reuniões essenciais no Scrum, como o *sprint planning*, que é quando a equipe define metas e seleciona as histórias a serem desenvolvidas; e o *daily Scrum*, uma reunião diária, de curta duração, para atualização e sincronização da equipe. Por fim, abordamos os papéis cruciais no Scrum: o *Scrum master*, responsável por garantir a correta implementação do Scrum e facilitar as atividades da equipe; e o *product owner*, responsável por representar os interesses dos *stakeholders*, definir os requisitos e priorizar o *product backlog*.

Esses conceitos fundamentais fornecem uma base sólida para a compreensão do Scrum e do seu potencial para entregar valor de forma incremental e adaptativa.



## REFERÊNCIAS

CRUZ, F. **Scrum e Ágil em projetos**: guia completo. Rio de Janeiro: Brasport, 2015.

PACHECO, R.; PRIKLADNICKI, R. **Métodos ágeis para desenvolvimento de software**: Scrum. Porto Alegre: Bookman, 2011.

PRIKLADNICKI, R. et al. **Métodos ágeis para desenvolvimento de software**. Porto Alegre: Bookman, 2014.

SUTHERLAND, J. **Scrum**: a arte de fazer o dobro do trabalho na metade do tempo. 2. ed. São Paulo: Leya, 2016.

SUTHERLAND, J.; SCHWABER, K. **Guia do Scrum**: um guia definitivo para o Scrum – as regras do jogo. [S.l.]: Scrum.Org and ScrumInc, 2014. Disponível em: <<https://Scrumguides.org/docs/Scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>. Acesso em: 23 jun. 2023.

SUTHERLAND, J.; SUTHERLAND, J. J. **Scrum**: a arte de fazer o dobro do trabalho na metade do tempo. Tradução de Nina Lua. São Paulo: Sextante, 2019.