**SCRUM**

**Reuniões:**

* ***Daily Scrum***: Todos os dias durante o Sprint é realizada a reunião diária do Scrum, nela todos os envolvidos no projeto ficam sabendo como o projeto está. Tem duração média de 15 minutos e começa sempre no mesmo horário e local, além disso, somente os envolvidos diretos nas atividades podem falar e responder as perguntas que tem como foco saber o que foi feito, o que será feito e se há algo impedindo o desenvolvimento das atividades, apesar dessa restrição os envolvidos indiretamente também podem participar como espectadores;
* **Sprint Planning Meeting**: Nessa reunião, conhecida como planejamento de Sprint, são selecionadas as atividades que serão desenvolvidas bem como uma avaliação para saber se elas podem ser desenvolvidas dentro do tempo do Sprint. Essa reunião é dividida em duas etapas sendo a primeira com o Product Owner onde são priorizados os itens da Product Backlog e uma segunda etapa com o time a fim de se criar um plano de trabalho que resulta no Sprint Backlog;
* **Sprint Review**: Essa reunião revisa todo o trabalho que foi desenvolvido, caso seja finalizado com sucesso uma demonstração é agendada com os stakeholders, caso contrário não há apresentação;
* **Sprint Retrospective**: Essa reunião é uma espécie de lições aprendidas onde são revistos e reavaliados processos de desenvolvimento utilizados durante o Sprint;

**Backlog da Sprint** é um conjunto de itens do Backlog do Produto selecionados para a Sprint, juntamente com o plano para entregar o incremento do produto e atingir o objetivo da Sprint.**O Backlog da Sprint é a previsão do Time de Desenvolvimento sobre qual funcionalidade estará no próximo incremento e sobre o trabalho necessário para entregar essa funcionalidade em um incremento “Pronto”**.  
**O Backlog da Sprint torna visível todo o trabalho que o Time de Desenvolvimento identifica como necessário para atingir o objetivo da Sprint.**  
O Backlog da Sprint é um plano com detalhes suficientes que as mudanças no progresso sejam entendidas durante a Reunião Diária. O Time de Desenvolvimento modifica o Backlog da Sprint ao longo de toda a Sprint, e o Backlog da Sprint vai surgindo durante a Sprint. Este surgimento ocorre quando o Time de Desenvolvimento trabalha segundo o plano e aprende mais sobre o trabalho necessário para alcançar o objetivo da Sprint. Sempre que um novo trabalho é necessário, o Time de Desenvolvimento adiciona este ao Backlog da Sprint. Conforme o trabalho é realizado ou completado, a estimativa do trabalho restante é atualizada. Quando elementos do plano são considerados desnecessários, eles são removidos. Somente o Time de Desenvolvimento pode alterar o Backlog da Sprint durante a Sprint. **O Backlog da Sprint é altamente visível, uma imagem em tempo real do trabalho que o Time de Desenvolvimento planeja completar durante a Sprint, e pertence exclusivamente ao Time de Desenvolvimento.**

**Duração dos Eventos/Cerimônias do Scrum:**

* Sprint Planning (Planejamento da Sprint) - **8h**
* **Sprint Review (Revisão da Sprint) -** **4h**
* Sprint Retrospective (Retrospectiva da Sprint) - **3h**
* Daily Meeting (Reunião diária) - **15min**

**Product Backlog - o** Backlog do Produto é uma lista ordenada de tudo que deve ser necessário no produto, e**é uma origem única dos requisitos** para qualquer mudança a ser feita no produto. **O Product Owner é responsável pelo Backlog do Produto, incluindo seu conteúdo, disponibilidade e ordenação. O que é?** É um dos **artefato**s do Scrum (segundo o scrum guide, existem 3, product backlog, sprint backlog e incremento), uma lista ordenada de tudo o que será necessário para construção do produto, sendo que o Product Ownner é o responsável por ele, incluindo o conteúdo, ordenação e disponibilidade.

**Características:**

* O product backlog **nunca estará completo**, ele **evolui** conforme o produto e o ambiente. É **dinâmico**;
* Existirá enquanto o produto existir; é um **artefato vivo**;
* Lista todas as características, funções, tecnologias, melhorias e correções que constituem o produto a ser entregue;
* Tem um processo continuo de **refinamento** (adição de detalhes, estimativas...) realizado pelo PO e o time de desenvolvimento;
* É composto de **itens** que possuem atributos de descrição, ordem, estimativa e valor;
* Os itens mais refinados (detalhados) ficam no **topo** da lista.
* Quando o time de desenvolvimento termina o refinamento de um item, diz-se que ele está "pronto" e este item é considerado como "preparado" para entrar no planejamento da sprint.

**Cancelamento de uma Sprint** - apenas o Dono do Produto – Product Owner tem autoridade para cancelar um Sprint. Um Sprint pode ser cancelado se o objetivo se tornar obsoleto. Isso pode ocorrer se a organização mudar a diretiva estratégica ou se as condições de mercado ou tecnologia mudarem. Quando um Sprint é cancelado, todos os itens do Backlog do Produto que estiverem prontos devem ser revisados. Se uma parte do trabalho estiver potencialmente entregável, o Dono do Produto geralmente o aceita. Todo o trabalho incompleto do Backlog do Produto é re-estimado e colocado novamente no Backlog Produto.

O cancelamento de Sprint consome recursos, uma vez que todos os envolvidos devem reagrupar-se em outra reunião de [Planejamento de Sprint (+)](http://jkolb.com.br/?p=9280)para iniciar um novo [Sprint (+).](http://jkolb.com.br/?p=9246)  O incremento é a soma de todos os itens do Backlog do Produto completados durante a Sprint e tudo das Sprints anteriores. Ao final da Sprint, um novo incremento deve estar “Pronto”, ou seja, estar na condição utilizável e atender a definição de “Pronto” do Time Scrum, independente do Product Owner decidir por liberá-lo realmente ou não.

***Planning Poker*** - usada no Scrum pra estimar o tamanho de um software. *No Planning Poker cada integrante tem a sua disposição um baralho de 13 cartas, numeradas em uma sequência similar a encontrada nos números de Fibonacci. As cartas contém os tamanhos de****0, ½, 1, 2, 3, 5, 8, 13, 20, 40 e 100****que serão atribuídos a um cartão, havendo ainda uma carta com o símbolo de interrogação no qual representa que o estimador não está apto a estimar e outra carta com a imagem de uma xícara de café no qual representa a sugestão de uma pausa. (*<https://www.devmedia.com.br/scrum-e-planning-poker-analise-de-estimativa-de-software/31019>).

O Scrum emprega uma abordagem **iterativa e incremental** para aperfeiçoar a **previsibilidade e o controle de riscos**. Scrum invoca transparência. Decisões para otimizar o valor e o controle de riscos são feitos com base na percepção existente do estado dos artefatos. Na medida em que a transparência é plena, estas decisões tem uma base sólida. Na medida em que os artefatos não são completamente transparentes, estas decisões podem ser falhas, valores podem diminuir e riscos podem aumentar.

* **Backlog do Produto**: apenas Product Owner pode alterar.
* **Backlog da Sprint**: apenas Time de Desenvolvimento pode alterar.

(lembrando que Time de Desenvolvimento não é a mesma coisa que Time Scrum).

**As 3 fases do Scrum:** Pré-planejamento, Desenvolvimento e Pós-planejamento.

* **Pré-planejamento (Pre-game phase)**: os requisitos são descritos em um documento chamado backlog. Posteriormente eles são priorizados e são feitas estimativas de esforço para o desenvolvimento de cada requisito. O planejamento inclui também, entre outras atividades, a definição da equipe de desenvolvimento, as ferramentas a serem usadas, os possíveis riscos do projeto e as necessidades de treinamento. Finalmente é proposta uma arquitetura de desenvolvimento. Eventuais alterações nos requisitos descritos no backlog são identificadas, assim como seus possíveis riscos.

* **Desenvolvimento (game phase):** as muitas variáveis técnicas e do ambiente identificadas previamente são observadas e controladas durante o desenvolvimento. Ao invés de considerar essas variáveis apenas no início do projeto, como no caso das metodologias tradicionais, no SCRUM o controle é feito continuamente, o que aumenta a flexibilidade para acompanhar as mudanças. Nesta fase o software é desenvolvido em ciclos (sprints) em que novas funcionalidades são adicionadas. Cada um desses ciclos é desenvolvido de forma tradicional, ou seja, primeiramente faz-se a análise, em seguida o projeto, implementação e testes. Cada um desses ciclos é planejado para durar de uma semana a um mês.
* **Pós-planejamento (post-game phase):** após a fase de desenvolvimento são feitas reuniões para analisar o progresso do projeto e demonstrar o software atual para os clientes.**Nesta fase são feitas as etapas de integração, testes finais e documentação.**

**Definição de “Pronto” (Definition of Done - DoD)**- Quando o item do Backlog do Produto ou um incremento é descrito como “Pronto”, todos devem entender o que o “Pronto” significa.

Embora, isso varie significativamente de um extremo ao outro para cada Time Scrum, os integrantes devem ter um entendimento compartilhado do que significa o trabalho estar completo, assegurando a transparência.

Esta é a “Definição de Pronto” para o Time Scrum e é usado para assegurar quando o trabalho esta completado no incremento do produto.

A mesma definição orienta o Time de Desenvolvimento no conhecimento de quantos itens do Backlog do Produto podem ser selecionados durante a Reunião de Planejamento da Sprint.

O propósito de cada Sprint é entregar incrementos de funcionalidades potencialmente utilizáveis que aderem à definição atual de “Pronto” do Time Scrum.

O PO usa as regras SMART para verificar os objetivos dos stakeholders, são elas:

* Specific (Específico): todos terão o mesmo entendimento a respeito dos objetivos.
* Measurable (Mensurável): podemos determinar objetivamente se os objetivos foram alcançados.
* Achievable (Realizável): os clientes concordam sobre o que são os objetivos.
* Realistic (Realista): devemos ser capazes de realizar os objetivos do projeto com os recursos disponíveis.
* Timed (Cronometrado): prazos definidos para a realização dos objetivos são viáveis.

**Gráfico Burndown:** torna visível a evolução diária do trabalho da equipe de desenvolvimento. No gráfico o eixo x (horizontal) representa o tamanho da Sprint em dias e o eixo y (vertical) a quantidade de tarefas. É uma representação gráfica do trabalho a ser feito versus tempo. Mostra o desenvolvimento planejado e o desenvolvimento ideal.

O **Scrum Team** é composto por: Product Owner (PO), Scrum Master (SM), Development  
Team (DT). **ST = PO+SM+DT.** Atenção é possível que haja sobreposição de papéis, o **SM**  
pode ser do **DT** e o **PO** pode ser do **DT**, mas um **SM** jamais pode ser o **PO!**

**XP – Extreming Programming**

Em **XP**, um cliente ou usuário é parte do time de XP e é responsável na tomada de decisões sobre requisitos. XP também encoraja todos da equipe a usar os cartões **CRC (classe-responsabilidade-desenvolvedor)** que serve como um mecanismo bastante eficaz para pensar sobre o software em um contexto **OO** (orientado a objetos). Esses cartões permitem identificarmos e organizarmos as classes OO para o incremento sendo desenvolvido. Os cartões CRC são o único artefato de projeto produzido como parte do projeto XP. Fazem parte das principais práticas do XP:

* Planejamento Incremental
* Pequenas Releases - entregas pequenas
* Projeto Simples
* Desenvolvimento Test-First - primeiro os testes
* Refatoração
* Programação em pares
* Propriedade coletiva
* Integração contínua
* **Ritmo sustentável - 40h semanais**
* Uso de metáforas
* Cliente on-site - cliente sempre disponível
* Reuniões em pé
* Jogo do planejamento
* Time coeso

Planejamento incremental e Histórias do Usuário são práticas XP.

* **Planejamento Incremental (Planning Game):** requisitos são registrados como histórias dos usuários e priorizados para serem incluídos em uma determinada iteração.
* **Histórias de Usuário:**descrevem requisitos de forma ágil. São textuais e escritas em cartões. Segue o padrão: "Como um aluno, quero registrar meus livros para que..."
* **Práticas XP:** Metáfora, **Histórias de Usuário**, Projeto Simples, Refatoração, Programação em Pares, Propriedade coletiva do código, Padrão de codificação, Ritmo sustentável, Cliente sempre presente, Reuniões em pé, TDD, Integração contínua, **Planejamento incremental** e Time coeso.

**Plano de Iteração ou Iteration Planning Game:** Cada iteração é de 1 a 3 semanas. Histórias de usuários são escolhidas para essa iteração pelo cliente no plano de liberação, na ordem de maior valor para o cliente primeiro. Testes de aceitação com falha a serem corrigidos também são selecionados.

As histórias do usuário e os testes com falha são divididos nas tarefas de programação que os darão suporte. Enquanto as histórias do usuário estão no idioma do cliente, as tarefas estão no idioma do desenvolvedor (CRC). Esses cartões de tarefas serão o plano detalhado da iteração.

**Os desenvolvedores se inscrevem para executar as tarefas e depois estimam quanto tempo suas próprias tarefas levarão para serem concluídas.** É importante que o desenvolvedor que aceita uma tarefa também seja quem calcula quanto tempo levará para concluir. As pessoas não são intercambiáveis ​​e a pessoa que fará a tarefa deve estimar quanto tempo levará.

Cada tarefa deve ser estimada em 1, 2 ou 3 (adicione 1/2, se necessário) dias de programação ideais. Os dias ideais de programação são quanto tempo você levaria para concluir a tarefa se não houvesse distrações. Tarefas com duração inferior a 1 dia podem ser agrupadas. Tarefas com duração superior a 3 dias devem ser divididas mais adiante.

Fonte: http://www.extremeprogramming.org/rules/iterationplanning.html

Cuidado para não confundir **iteration planning** com **planning poker**. Ele pode ser usado para estimar os cartões, mas não é citado explicitamente nas regras. É apenas uma das práticas citadas.