**Scrum: Resumo**

*Autoria: Julia Soares Pereira*

**Introdução**

Ao longo dos anos, conforme a informática se tornava cada vez mais presente no mercado de trabalho, mais softwares precisavam ser desenvolvidos. Foi a partir dessa demanda que as estratégias de desenvolvimento de software começaram a surgir. Elas são criadas com o objetivo de organizar tanto o projeto de software quanto o time de desenvolvimento, visando a uma qualidade maior do produto final, maior eficácia e menores tempo gasto e custos de produção. Elas também são usadas para desviar de problemas como caos pela mudança constante de requisitos, estimativas de tempo, custo e qualidade do produto totalmente irreais e desenvolvedores que não sabem exatamente como está o andamento do projeto.

Junto das metodologias de desenvolvimento de software, algumas técnicas foram criadas com o foco em terminar os projetos de software rapidamente e de forma eficaz, mas sem deixar a qualidade a desejar. Este tipo de técnica foi categorizado com o nome de Técnica de Desenvolvimento Ágil, e o **SCRUM** é uma dessas técnicas. Ele lida totalmente com as pessoas e como elas vão desenvolver o projeto, sem muito foco na parte técnica (as tecnologias utilizadas no processo).

**SCRUM: O que é?**

O **SCRUM** é um modelo de desenvolvimento ágil de software que fornece métodos para definir o planejamento, os principais papéis das pessoas e a forma de trabalho do time. A ideia do SCRUM é justamente definir papéis bem específicos, para que não haja dúvidas sobre o que cada um fará para o time seguir em frente no desenvolvimento do software.

**Quais os pilares do Scrum?**

O Scrum tem seus pilares, que são baseados no controle empírico de processos e criam um ambiente propício para o desenvolvimento de projetos ágeis. São eles:

* **Inspeção:** Durante o desenvolvimento do produto, inspeções são feitas de tempos em tempos. Seu intuito não é somente avaliar o produto sendo construído, mas também a maneira como ele está sendo construído, ou seja, o processo de trabalho.
* **Transparência:** Para que as inspeções sejam feitas de forma eficaz, precisa-se de transparência. Todas as informações necessárias para a construção do produto devem estar disponíveis para todas as pessoas envolvidas.
* **Adaptação:** A transparência torna as inspeções possíveis, e, consequentemente, a adaptação, que garante às pessoas do grupo melhores alinhamento e comunicação entre si, o que aumenta a confiança da equipe.

O pensamento enxuto **(*lean thinking*)**, que é uma metodologia de gestão que prioriza agregar valor para o cliente e reduzir desperdícios, também é considerado um grande influenciador do Scrum. De fato, um Time Scrum busca a todo momento reduzir desperdícios de todos os tipos e agregar o maior valor possível ao produto sendo construído.

**Quais os valores do Scrum?**

Compreender os valores do Scrum é importante, pois eles ditam a maneira como as decisões são tomadas, melhoram a compreensão em momentos de ambiguidade e explicam por que o time faz o que ele faz. Quando esses valores são incorporados pelo Time Scrum, os pilares transparência, inspeção e adaptação ganham vida.

Os valores do Scrum são:

* **Foco:** O time precisa fazer o que for preciso para manter o foco durante toda a Sprint e concentrar-se na entrega a ser feita, evitando reuniões desnecessárias, mensagens ou qualquer interrupção que não seja relevante para entregar o trabalho na Sprint corrente.
* **Respeito:** Respeito entre os membros de um time Scrum é fundamental para o sucesso de um projeto. Muitas vezes, um grupo trabalhando com base no SCRUM alega ser ágil, mas integrantes da equipe acabam entrando em conflito entre si por algum motivo, perdendo o respeito uns pelos outros. Um time Scrum é um grupo com objetivos em comum, cujos membros devem manter a confiança uns nos outros e se respeitarem, visto que eles, na maior parte do tempo, dependerão uns dos outros para realizar suas tarefas. Assim, quando alguém do time se compromete com uma tarefa, o time confia, apoia e se coloca à disposição para fornecer suporte. Da mesma forma, o respeito permite que seja possível expor e resolver quaisquer dificuldades e obstáculos de forma objetiva e transparente.
* **Comprometimento:** A cada Sprint, por exemplo, o time Scrum define de forma clara e transparente o que se compromete a entregar no Sprint, e esse comprometimento jamais deve ser superficial. Ele é compartilhado dentro do time e para a empresa. Uma vez comprometido com as entregas, é obrigação do time Scrum fazer tudo o que estiver a seu alcance para fazer acontecer.
* **Coragem:** Apesar de todos possuírem seus medos, ao trabalhar com Scrum, a coragem torna-se essencial. Há muitos profissionais excelentes que simplesmente paralisam diante do medo. Nem todos os membros de um time conseguem lidar com o medo sem congelar. Por isso, o apoio e o auxílio mútuo são necessários entre os membros da equipe. Eles devem ajudar uns aos outros, aliviando o medo e fazendo com que o time todo saiba lidar melhor com suas dificuldades. A coragem também é necessária para que as discussões sobre dificuldades ou notícias ruins sejam possíveis e ocorram de forma aberta, respeitosa e produtiva. Problemas não podem ficar debaixo do tapete, senão a situação piorará em algum momento. Quando times não têm coragem para fazer o que eles acham que é o certo, provavelmente a coisa certa não será feita.
* **Abertura:** Em um time Scrum, deve existir abertura para que qualquer integrante possa colocar verbalmente quaisquer pontos relevantes para o desempenho do time e do projeto. A equipe procura sempre estar aberta para novas ideias, percepções e diferentes formas de pensar, itens que servem de combustível para a criatividade e aprendizagem e ajudam a construir times de alto desempenho.

**Como o SCRUM funciona:**

1. O produto precisa ser definido: quais são seus requisitos? O que realmente o cliente quer? O responsável por esta tarefa é quem chamamos de Proprietário do Produto (Product Owner ou P.O em inglês);
2. O P.O define quais são as funcionalidades do produto que mais importam, criando assim o Backlog do Produto.
3. Com as prioridades definidas, uma pessoa é definida para ser o ScrumMaster, uma espécie de coordenador do projeto. O ScrumMaster, junto ao Proprietário do Produto e ao Time de desenvolvimento, define as Sprints.
4. Cada Sprint possui uma parte de todo o Product Backlog, e devem ser trabalhadas de acordo com as prioridades definidas no Product Backlog. As Sprints devem ser preparadas de forma que durem de 2 a 4 semanas e que, no final de cada período, garantam ao time um produto apresentável para o cliente.
5. As Sprints vão sendo feitas até o Product Backlog acabar e o Proprietário do Produto definir que o projeto está pronto. Mas isso não quer dizer que novas funcionalidades não podem ser incluídas: basta ir adicionando ao Product Backlog e realizando mais Sprints.

O SCRUM tem três partes principais em seu modelo: **Papéis (Roles)**, **Cerimônias (Cerimonies)** e **Artefatos (Artifacts)**. Cada um destes três pilares contém outros subitens. Todas essas três partes principais são utilizadas no que chamamos de ciclo de desenvolvimento, ou seja, a Sprint. Cada Sprint possui suas fases e utiliza desses papéis, cerimônias e artefatos para alcançar seu objetivo final.

**1. Papéis do SCRUM (Roles)**

O primeiro passo para o desenvolvimento por SCRUM é definir quem vai fazer o quê. Há três entidades principais: o Proprietário do Produto (Product Owner), o ScrumMaster e o Time.

**1.1. Proprietário do Produto (Product Owner)**

O Proprietário do Produto representa os interesses do cliente. Ele tem que ser a interface entre o cliente e o time de desenvolvedores, ou seja, estar sempre em contato com o cliente e saber exatamente o que o projeto tem que ser.

Ele tem as seguintes responsabilidades:

* Definir as características e conteúdo do produto;
* Decidir sobre a data de término;
* Ser responsável pela rentabilidade do produto;
* Priorizar as funções de acordo com o valor de mercado e com o cliente (e outras possíveis variáveis);
* Ajustar recursos e priorizar tarefas a cada Sprint, como necessário;
* Aceitar ou rejeitar o resultado do trabalho;
* Definir o Product Backlog.

**1.2. ScrumMaster**

O ScrumMaster é, durante o trabalho, é quem mais fica em contato com o Proprietário do Produto. Podemos chamá-lo também de Coordenador do Projeto devido à importância de suas responsabilidades. Ele facilita a comunicação entre o time e faz o projeto andar de verdade.

Ele deve:

* Assegurar que a equipe de desenvolvimento funcione plenamente e seja produtiva;
* Ajudar na cooperação entre todas as funções e papéis do time;
* Remover barreiras (como problemas pessoais ou conflitos entre os integrantes do time de desenvolvimento);
* Proteger a equipe de interferências externas;
* Assegurar-se de que a metodologia está sendo seguida, incluindo chamadas para reuniões diárias (Daily Scrum Meetings), revisões de atividade (Sprint Reviews) e reuniões de planejamento das atividades (Sprint Planning).
* Saber quais atividades foram concluídas, iniciadas e descobertas e qualquer estimativa que possa ter mudado, para poder atualizar sempre o Burndown Chart, que mostra quanto trabalho falta para um Sprint acabar, dia por dia. Ele também tem que sempre olhar cuidadosamente para o número de tarefas em aberto. Elas devem ser minimizadas o máximo possível para garantir um trabalho sempre limpo e eficiente.
* Avaliar as dependências superficiais e bloqueios que causam prejuízos ao andamento do Projeto. Estes itens devem receber prioridade e um plano de solução deve ser implementado de acordo com a ordem de prioridade destes impedimentos. Alguns podem ser resolvidos pelo time, outros podem ser resolvidos por meio de vários times e outros podem precisar de envolvimento da gerência, pois podem ser problemas relacionados a atitudes da empresa. Como exemplo desse tipo de impedimento externo, há as questões judiciais.

**1.3. A Equipe de Desenvolvimento (Scrum Team ou Dev Team)**

A equipe de desenvolvimento é formada por aqueles que, efetivamente, trabalharão para que o software funcione. Pode haver uma ou mais equipes de desenvolvimento, dependendo da complexidade do software.

Algumas características das equipes de desenvolvimento:

* São pequenas e multidisciplinares, com, em média, 7 participantes;
* Devem fazer de tudo, dentro dos limites das diretrizes do projeto, para atingir a meta de cada Sprint;
* São autogerenciáveis (decidem internamente quem faz o que e como);
* Trabalham para atingir todos os resultados definidos pelo Proprietário do Produto.

**2. Cerimônias SCRUM (Cerimonies)**

São eventos realizados entre os integrantes do time que têm como proposta principal **ajudar o Scrum Master e o Product Owner a entender, acompanhar e auxiliar todos a alcançarem seus objetivos.** As cerimônias do Scrum giram em torno da Sprint e em todas elas há um input (entrada de informações), um processamento (entendimento, discussão e tomada de decisão) e um output (saída de um plano).

**2.1. Reunião de Planejamento de Sprint (Sprint Planning)**

É o primeiro evento numa Sprint do Scrum, e seu objetivo é determinar o que pode ser entregue na Sprint que se inicia, e como o trabalho será feito. Os participantes são todos os membros do time. O primeiro passo para esse evento é realizado pelo P.O quando ele desenvolve um plano para o projeto de software. Nesse plano, o P.O trabalha interagindo com o cliente para definir todas as funcionalidades desejadas no produto. A partir desta lista, ele também deve definir a ordem de prioridade dos itens do plano de acordo com várias variáveis: valor de mercado, importância para o cliente, entre outras. A lista resultante desse processo é o “Product Backlog”.

Na Reunião de Planejamento de Sprint, o ScrumMaster trabalha junto com o P.O e a Equipe de Desenvolvimento para definir qual carga de tempo cada tarefa do Product Backlog terá. Isso ajuda o P.O a definir prazos reais para o projeto e o habilita a poder verificar como está o andamento do projeto durante todo o período de desenvolvimento. Esta carga de tempo e esforço é definida de acordo com o tamanho do(s) time(s), horas disponíveis e o nível de produtividade do time.

Quando as prioridades e os prazos são definidos, o ScrumMaster começa a trabalhar juntamente com a Equipe de Desenvolvimento para quebrar as tarefas divididas entre os integrantes da equipe de desenvolvimento de acordo com suas especialidades grandes em tarefas menores. Esta reunião de planejamento geralmente dura até 4 horas e é ela que define o Sprint Backlog, um dos artefatos SCRUM.

Uma vez definido o Sprint Backlog, com a listagem de todas as tarefas, seus responsáveis e seus prazos, o processo de desenvolvimento começa.

**2.2. Reuniões Diárias SCRUM**

São reuniões nas quais o ScrumMaster busca saber como está o projeto por meio da interação com o Scrum Team. É realizada uma reunião por dia durante a Sprint, e ela tem, aproximadamente, 15 minutos de duração. Durante a reunião, cada um dos membros responde as seguintes três perguntas:

1. O que fiz ontem?
2. O que farei hoje?
3. Quais impedimentos e dificuldades apareceram e aparecerão no caminho?

O Scrum Master tem um papel muito importante nestas reuniões: ele terá que identificar todos os problemas ou novas tarefas que surgirem e planejar como estes serão resolvidos. Além do mais, ele terá, assim, uma visão completa do projeto, o que lhe garantirá a capacidade de gerenciar todas as sub-tarefas de cada Sprint e preencher o Burndown Chart.

**2.3. Reunião de Revisão do Sprint (Sprint Review)**

É um evento realizado no fim do período da Sprint. Nessa reunião, unem-se o Scrum Team, o Scrum Master e o Product Owner (e convidados, caso este ache adequado, como o cliente ou acionistas do projeto). Esta reunião dura aproximadamente 4 horas.

Na primeira parte da reunião, o resultado do Sprint é apresentado para o P.O, ou seja, tudo que foi desenvolvido durante o ciclo de desenvolvimento. As funcionalidades são revisadas e o valor do produto é definido. Depois de revisar todo este resultado, o P.O, então, define quais itens do Product Backlog foram completados na Sprint e discute com os membros da equipe e o cliente quais serão as novas prioridades. Este é o primeiro passo para criar uma nova Sprint, caso seja necessário, pois dessa primeira parte da reunião, um novo Product Backlog é gerado.

Na segunda parte da reunião, o Scrum Master e o Scrum Team se reúnem e o Master revisa os altos e baixos do ciclo de desenvolvimento até o momento. O time conversa para saber o que foi bom (e pode melhorar ainda mais) e o que foi ruim durante o processo. Esta parte é, basicamente, um aprimoramento interno.

Depois que a reunião é finalizada, a Sprint Retrospective acontece, permitindo que uma nova Sprint comece. O ciclo se repete até que o produto seja finalizado e o Product Backlog esteja limpo de pendências.

**2.4. Reunião de Retrospectiva da Sprint (Sprint Retrospective)**

Enquanto o objetivo do Sprint Review é verificar necessidades de adaptações no produto, o Sprint Retrospective é o evento realizado para verificar necessidades de adaptações no processo de trabalho.

Essa reunião é realizada logo após a reunião de revisão (após um pequeno intervalo de alguns minutos geralmente). Nela, a equipe apresenta os pontos que funcionaram na Sprint e os pontos nos quais houve problemas ou coisas que não surtiram efeito. Esse conteúdo é analisado pela equipe e monta-se um plano de melhorias para ser executado durante a próxima Sprint.

É muito comum encontrar times que escolhem adotar a técnica Scrum mas, ao primeiro sinal de dificuldade, os integrantes eliminam a reunião de retrospectiva do seu processo, como se ela fosse perda de tempo. Apesar de não serem fáceis e demandarem tempo, comprometimento, energia e coragem do time, as Sprint Retrospectives são a única oportunidade que o time tem para avaliar como eles estão trabalhando juntos e como podem melhorar, processo fundamental para tornar as próximas sprints mais produtivas e o produto final mais agradável.

Além dos fatores descritos acima, a retrospectiva é uma oportunidade para que os membros do time repensem seu comportamento e mindset diante do projeto. É um momento em que a confiança e a transparência devem prevalecer para que os pequenos problemas sejam resolvidos rapidamente e não virem uma grande dor de cabeça no futuro.

**3. Artefatos SCRUM (Artifacts)**

São as ferramentas básicas para se trabalhar com o modelo Scrum. Estes artefatos servem como guias e indicadores durante o processo.

**3.1. Backlog do Produto (Product Backlog)**

Dentro do Scrum, o Product Backlog é o ponto de partida do qual todas as práticas e ferramentas da metodologia partem e para o qual todas retornam.

É a lista de funcionalidades que devem fazer parte do produto preparada pelo Proprietário do Produto em conjunto com o cliente. A equipe SCRUM também contribui para sua organização, estimando o custo do desenvolvimento das funcionalidades. O Product Backlog deve incluir todas as funções visíveis ao cliente, mas também os requisitos técnicos para poder desenvolver o produto.

**3.1.1. História do Usuário (User Story)**

É um formato sucinto de escrita dos requisitos necessários para a construção de um produto que deve ser compreensível para os clientes e consumidores. Em outras palavras, é a forma de representar as tarefas do Product Backlog. Cada User Story (história do usuário) representa um único item do Product Backlog.

Modelo: Como um **<ator>**, quero/preciso de/gostaria de **<ação>** para que **<objetivo>**.

* **Ator:** É o proprietário da história, quem irá utilizar a funcionalidade ou requisito e espera que seu desejo seja atendido.
* **Ação:** É o que o ator quer que seja feito dentro do produto.
* **Objetivo:** É o que o ator espera que vá acontecer após a ação ser executada (também vista como uma justificativa.

Exemplos:

“Como gerente, quero conseguir entender o progresso dos meus colegas, para que eu possa ter relatórios melhores dos nossos acertos e falhas.”

“Como cliente, quero poder fazer compras online, para que eu possa economizar tempo e aproveitar a comodidade de receber os produtos em casa.”

**3.1.2. Meta do Produto**

O compromisso do Product Backlog é a Meta do Produto. Ela define um estado futuro para o produto. Trata-se do objetivo de longo prazo que o Time Scrum planeja-se para alcançar, um olhar para o produto como um todo.

**3.2. Backlog da Sprint (Sprint Backlog)**

Assim que a equipe de Scrum escolher e definir a lista de requisitos e a prioridade de seus itens do Product Backlog, cada um desses itens agora será dividido em partes menores para o Sprint Backlog. Ou seja, uma lista especificando os passos necessários para implementar uma funcionalidade do sistema.

Logo após o Sprint Backlog estar concluído, o total de horas trabalhadas é comparado com o total previsto anteriormente no Product Backlog. Caso haja uma diferença significativa, a equipe SCRUM deve negociar novamente com o cliente o número correto de horas, para que o Sprint seja realizado com maior probabilidade de sucesso.

**3.2.1. Meta da Sprint**

O compromisso do Sprint Backlog é a Meta da Sprint. Criada pelos Desenvolvedores e pelo P.O durante o Sprint Planning, ela refere-se ao conjunto de metas do Product Backlog selecionados para a Sprint do momento e a um plano para entregar o Incremento. Seu prazo é condicionado à duração da sprint. Em outras palavras, a meta da sprint esclarece o seu propósito durante a criação de um programa.

**3.3. Incremento do Produto (Product Increment)**

O Incremento do Produto representa a soma de todos os itens completos em uma Sprint. Ou seja, é a junção das funcionalidades "prontas" do produto.

**3.4. Gráfico de burndown (Burndown Chart)**

Durante as Sprints, a principal ferramenta de medição de desempenho é o Burndown Chart, que é uma das características mais especiais do SCRUM e que o torna um grande diferencial, no sentido positivo. Trata-se de um gráfico que é administrado pelo ScrumMaster e objetiva analisar se o grupo está adiantado, atrasado ou no prazo quanto ao desenvolvimento do software. Ele mostra a quantidade de trabalho cumulativo restante de um Sprint, dia por dia e, nele, a altura indica a quantidade de tarefas do Sprint Backlog não completadas, e o comprimento são os dias. Com isso, é possível visualizar facilmente se um trabalho está progredindo quando as colunas do gráfico vão caindo em sua altura. Se um ciclo de desenvolvimento, ou Sprint, tem a duração de 30 dias, haverão 30 colunas juntas. As colunas têm que diminuir ao longo do comprimento e antes ou até o dia n°30 não poderá haver colunas. Isso indicará que todas as tarefas foram completadas e a Sprint foi um sucesso.

As reuniões diárias da Sprint garantem ao Scrum Master a capacidade de, dia a dia, ajustar o Burndown Chart, pois cada reunião é uma atualização do progresso do Dev Team. O mesmo gráfico pode ser mostrado para o Proprietário do Produto, para que ele possa verificar se tudo está correndo bem. Por essas razões, o Burndown Chart é um dos principais recursos de medição do processo de desenvolvimento e um diferencial para a metodologia SCRUM.

**3.5. Definição de Pronto (Definition of Done – DoD)**

Em um grupo Scrum, tarefas são consideradas “prontas” quando elas condizem com a DoD estabelecida pelo time. A Definition of Done (DoD) é um acordo entre o grupo que define os requisitos aos quais cada item do backlog que é produzido no Sprint Planning deve atender para ser dado como pronto.

Embora isso varie significativamente para cada time Scrum, os membros do time 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 usada para assegurar quando o trabalho está completo no incremento do produto.

O DoD pode ser aplicado no sprint planning, além de poder ser estabelecido por backlog, por sprint, por equipe, por produto ou projeto etc, de acordo com o melhor cenário para o desenvolvimento do software.

Exemplos:

• Por Produto: para o Produto A tem uma DoD e para o produto B tem outra DoD.

• Por Item de Backlog: para a User Story X tem uma DoD e para a User Story Y tem outra DoD.

• Por Sprint: para o Sprint 1 tem uma DoD e para o Sprint 2 outra DoD.

Garantir que o uso da DoD está funcionando é responsabilidade de todos.

• O PO (Product Owner) tem a responsabilidade de aceitar que apenas itens Done sejam dados como concluídos pelo Dev Team.

• O Dev Team tem a autonomia de somente aceitar itens Ready mas tem também a responsabilidade de entregá-los como itens Done.

• O SM (Scrum Master) tem a responsabilidade de garantir que o Scrum Team saiba do que se trata e está aplicando da forma correta.

Todo o time tem a responsabilidade de garantir que haja uma DoD com a qual todos estejam de acordo, que ela esteja sendo usada de fato e que, a cada Sprint, ela seja revisitada para ajustes necessários.

**3.6. Definição de Preparado (Definition of Ready - DoR)**

No entanto, é necessário ter atenção para não confundir os conceitos de “Definition of Ready” e “Definition of Done”. Enquanto a DoD refere-se a tarefas que podem ser consideradas realizadas, a DoR trata-se de uma lista de requisitos que determinada tarefa precisa cumprir para que esteja apta a entrar no Backlog da Sprint. Ou seja, o termo refere-se a tarefas que podem entrar na lista de afazeres da sprint.

**3.7. Planning Poker**

Também chamado de Scrum Poker, é uma estratégia usada em projetos ágeis que busca priorizar as tarefas e fazer estimativas do esforço que é exigido para executá-las. Cada membro da equipe de desenvolvimento recebe um conjunto de cartas com os valores de uma certa sequência, que irá determinar, ao final do jogo, uma estimativa para as fases do Product Backlog.

A ferramenta utiliza, por padrão, o conceito de sequência de Fibonacci por meio de um baralho, mas o time também pode criar sua prórpia sequência baseada na Fibonacci para usar o Planning Poker. Em Fibonacci, o terceiro número da sequência sempre é a soma dos dois primeiros números. Por exemplo: 1,2,3,5,8,13,21,34 e assim por diante. Cada usuário pode ter seu próprio baralho impresso ou usar apps específicos de Planning Poker. Algumas cartas podem conter símbolos:

* Xícara de café: Significa que algum dos participantes gostaria de fazer uma pausa (algumas reuniões de planejamento demoram várias horas);
* Interrogação (?): Para eventuais dúvidas dos participantes;
* 0.5 ou ⅕: Remete à uma pontuação de uma user story muito pequena (um pouco acima de 0).

O P.O lê uma user story por rodada e os desenvolvedores avaliam sua complexidade e aguardam o momento certo para apresentar suas pontuações, dúvidas ou até mesmo solicitar uma pausa.

**4. Técnicas de Priorização Scrum:**

Priorizar as atividades e tarefas em um projeto de desenvolvimento de software é uma parte fundamental do Scrum. Por meio das técnicas de priorização adequadas, é possível aumentar a eficiência do time e garantir a entrega de valor de forma mais eficaz.

**4.1. Priorização Baseada em Valor**

Nessa técnica de priorização, o valor que as atividades trazem ao cliente ou ao produto é o determinante de sua importância. O time e o Product Owner trabalham juntos para identificar e classificar as funcionalidades e tarefas com base em sua importância estratégica. Dessa forma, é possível direcionar os esforços do time para as atividades que trarão o maior benefício para o cliente ou usuário final.

**4.2. Priorização Baseada em Esforço**

Outra técnica de priorização Scrum é a baseada em esforço. Nesse caso, as atividades são analisadas levando em consideração o tempo, recursos e complexidade necessários para sua execução. O time realiza estimativas e pontuações para cada atividade, levando em conta fatores como a complexidade do trabalho, a experiência e habilidades dos membros do time. A partir dessas avaliações, é possível priorizar as atividades com base na capacidade do time de entregá-las dentro do prazo desejado.

**4.3. Priorização Baseada em Riscos**

A priorização baseada em riscos é uma técnica que visa identificar e priorizar as atividades que podem representar maiores riscos para o projeto. Nessa abordagem, o time analisa os possíveis obstáculos, desafios e incertezas associadas a cada atividade. As atividades que apresentam maior risco são priorizadas, permitindo que o time possa lidar com os desafios mais críticos desde cedo no projeto. Dessa forma, é possível minimizar possíveis impactos negativos e garantir uma execução mais suave do trabalho.

**4.4. Priorização Baseada em Dependências**

A técnica de priorização baseada em dependências é utilizada quando existem atividades que possuem pré-requisitos ou dependem de outras atividades para sua execução. Nesse caso, o time analisa as relações de dependência entre as atividades e prioriza aquelas que são dependentes de outras. Garantir que as atividades pré-requisito sejam concluídas antes das atividades dependentes é essencial para evitar bloqueios ou atrasos no projeto. Essa técnica permite que o time trabalhe de forma mais eficiente, evitando que atividades importantes fiquem paradas por falta de dependências resolvidas.

**5. Vantagens do Scrum**

**5.1. Maior flexibilidade**

Ao invés de seguir um plano rígido, as equipes que escolhem o método Scrum têm a capacidade de se adaptar e responder às mudanças durante o processo de desenvolvimento. Isso permite que os integrantes do time lidem com requisitos em constante evolução e necessidades do cliente com mais facilidade.

**5.2. Entrega de valor incremental**

O Scrum é baseado em ciclos de desenvolvimento chamados de Sprints, nos quais pequenas partes do produto são desenvolvidas e entregues ao longo do tempo. Isso possibilita que o cliente tenha acesso a certas funcionalidades logo em seus estágios iniciais, proporcionando feedback valioso e permitindo ajustes conforme necessário. Ou seja, a entrega de valor é feita de forma incremental, o que pode resultar em maior satisfação do cliente.

**5.3. Colaboração e comunicação eficazes**

O método Scrum é baseado na cultura de colaboração e comunicação entre os membros da equipe. As reuniões diárias, por exemplo, são um dos vários métodos que o Scrum proporciona para facilitar a identificação de problemas e a tomada de decisões rápida e eficaz.

**5.4. Melhoria contínua**

Grupos que trabalham utilizando Scrum buscam melhorar continuamente. Por meio das retrospectivas realizadas ao final de cada Sprint, a equipe analisa o que funcionou bem e o que precisa ser aprimorado. Essa abordagem permite que a equipe aprenda com seus erros e busque um processo de desenvolvimento cada vez mais eficiente.

**5.5. Visibilidade e transparência**

O Scrum, que possui transparência como um de seus pilares, possibilita uma visão clara do progresso do projeto. Todos os membros da equipe são capazes de acompanhar as tarefas em andamento, o que já foi concluído e o que está pendente. Isso facilita o monitoramento do projeto para todos os envolvidos.

**6. Desvantagens do Scrum**

**6.1. Dificuldade na definição de prazos:**

De fato, a abordagem flexível do método Scrum é uma de suas vantagens, mas, às vezes, isso pode dificultar a definição de prazos reais para a conclusão de um projeto. Como as necessidades e os requisitos podem mudar ao longo do tempo, é desafiador estimar com precisão quando o projeto estará concluído. Isso pode ser um problema especialmente para clientes que têm expectativas claras de prazos.

**6.2. Requer uma equipe altamente autogerenciável:**

O Scrum pressupõe que a equipe seja autogerenciável e capaz de tomar decisões de forma independente. Entretanto, isso pode ser um desafio para um time que não possua experiência ou maturidade suficiente para assumir essa responsabilidade. A falta de autogerenciamento pode levar a atrasos e problemas de qualidade.

**6.3. Necessidade de colaboração intensa:**

O Scrum exige intensa colaboração entre os integrantes da equipe. Embora isso possa ser benéfico em muitos aspectos, pode ser difícil para equipes com membros que moram longe ou que têm diferentes horários de trabalho. A falta de colaboração eficaz pode prejudicar a produtividade.

**6.4. Risco de sobrecarga de trabalho:**

O Scrum é baseado em ciclos curtos de desenvolvimento, e isso pode resultar em uma pressão adicional sobre os membros da equipe. A necessidade de entregar tarefas valiosas em prazos curtos pode sobrecarregar e, possivelmente, esgotar o time. É essencial garantir um equilíbrio adequado entre a produtividade e o bem-estar dos membros da equipe.

**6.5. Requer um comprometimento total da equipe:**

O Scrum funciona melhor quando todos os membros da equipe estão totalmente comprometidos com os objetivos do projeto. Isso significa que cada membro deve estar disponível e dedicado ao projeto durante toda a duração do Sprint. Se houver falta de comprometimento por parte de algum membro, isso pode prejudicar o progresso e a eficácia da equipe.

**Referências Bibliográficas:**

1 CISNEIROS, Hugo. **Modelo de Desenvolvimento Ágil SCRUM**. Devin, 2009. Disponível em: <https://www.devin.com.br/modelo-scrum/>. Acesso em: 27 mar. 2024. (bom para exemplos práticos)

2 RIBAS, Thomaz. Scrum: **O que É, Como Funciona e Exemplos Práticos [GUIA]**. Thomaz Ribas, s.d. Disponível em: <https://thomazribas.com/agile/scrum>. Acesso em: 27 mar. 2024. (bom para exemplos práticos)

3 VENTURELLI, Darlan. **Os artefatos do Scrum e seus Compromissos**. Gran Cursos Online, 2024. Disponível em: <https://blog.grancursosonline.com.br/os-artefatos-do-scrum/>. Acesso em: 28 mar. 2024.

4 HARRIS, Chandler. **Artefatos do Scrum ágil**. Atlassian, s.d. Disponível em: <https://www.atlassian.com/br/agile/scrum/artifacts>. Acesso em: 28 mar. 2024.

5 **SCRUM: A Metodologia Ágil Explicada de forma Definitiva**. Mind Master, s.d. Disponível em: <https://mindmaster.com.br/scrum/>. Acesso em: 28 mar. 2024.

6 VENTURA, Plínio. **Definition of Done (DoD) –** **A importância do item Pronto no Scrum**. Até o Momento, 2020. Disponível em: <https://www.ateomomento.com.br/definition-of-done-dod-a-importancia-do-item-done-no-scrum/>. Acesso em: 27 mar. 2024.

7 **TÉCNICAS De Priorização Scrum: Aumente A Eficiência Do Seu Time**. Awari, 2023. Disponível em: <https://awari.com.br/tecnicas-de-priorizacao-scrum-aumente-a-eficiencia-do-seu-time/>. Acesso em: 28 mar. 2024.

8 CUNHA, Wallace. **DOR e DOD: entenda a diferença desses conceitos**. Lumis, s.d. Disponível em: <https://www.lumis.com.br/a-lumis/blog/dor-e-dod.htm>. Acesso em: 28 mar. 2024.

9 **O QUE é e como funciona o Planning Poker?** Luby, 2021. Disponível em: <https://luby.com.br/scrum/o-que-e-e-como-funciona-o-planning-poker/>. Acesso em: 28 mar. 2024.

10 FILHO, Avelino. **Product Backlog: Épico, História de Usuário e Tarefas.** K21, 2020. Disponível em: <https://k21.global/br/blog/product-backlog-epico-historia-tarefas#historia-de-usuario-user-story-us>. Acesso em: 29 mar. 2024.