**O SCRUM E A SPRINT**

**Um pouco de história**

Um pouco de história, atualmente quando fazemos menção a Scrum, é comum o associarmos aos métodos ágeis, processo utilizado mundialmente! O conceito, entretanto, é mais antigo.

O termo "Scrum" foi cunhado por Takeuchi e Nonaka em alusão à formação do Rugby, em que todo o time está junto buscando a posse da bola. Se alguém quebra a formação, todo o time perde. Da mesma forma, quando alguém consegue a posse da bola, o time inteiro é responsável. Ou seja, os indivíduos são menos importantes do que o time em si.

Para Scrum, o conceito de time é extremamente importante! O product owner, o scrum master e a equipe de desenvolvimento estão todos englobados no time. Todos são igualmente responsáveis pelo sucesso ou fracasso de um projeto ou melhoria na vida do usuário.

A metáfora que utiliza o Rugby ainda se relaciona ao conceito de sprint, que é quando o time rouba a bola e o jogador deve correr rapidamente com ela. A partir disso mais uma Sprint é iniciada e novos pontos são acumulados. É um processo iterativo e incrementativo.

Vale notar que o Scrum de 1995 não é mesmo de hoje: com o passar dos anos e a grande adoção do processo, ele deixou de ser um método e passou a ser um framework, um conjunto de ideias.

Nesse curso a filosofia envolta no Scrum é abordada, e também são fornecidas sugestões de como implementar o framework em sua empresa.

**Time-boxes e Sprints**

Conforme mencionado no vídeo anterior, a Sprint é um dos time-boxes, e este conceito é extremamente importante para Scrum. A principal diferenciação entre o Scrum e outros processos ágeis é seu forte embasamento com time-boxes: caixas de tempo com capacidade definida, rígida e não negociável. Uma vez que a Sprint ou qualquer reunião define uma certa duração de tempo, ela passa a ser fixa. Já que esse é o principal conceito, começaremos este curso pelos time-boxes! A Sprint é o tempo que temos para agregar valor e marcar pontos. Adicionamos valor ao software conforme o que o usuário acredita que é necessário. O tempo máximo para isso é de até um mês, período definido no "Scrum Guide", leitura básica sobre o tema, publicada por Ken Schwaber e Jeff Sutherland.

**Duração da Sprint**

Conforme comentado, a Sprint limita-se ao máximo de um mês de duração. Esse período, entretanto, pode compreender 1 mês, 1 semana, 3 dias, e assim por diante. A decisão de demarcar um intervalo é de extrema importância, pois ela será a base de todos os demais time-boxes do Scrum.

Para delimitar o tamanho da Sprint vários fatores devem ser levados em consideração. Vamos refletir sobre o início de um projeto. Como qualquer começo, nesse momento não temos conhecimento sobre o destino final dele, pois nem o cliente sabe muito bem o que deseja. Nesse momento tendemos às Sprints com durações de tempo mais curtas.

Ao fim de cada Sprint é realizada uma Review Meeting, que é uma reunião na qual são mostrados os produtos

elaborados durante esse período. A partir dela recebemos feedbacks, e quanto mais retornos tivermos, melhor será para o cliente e para nós, pois evita a execução de trabalhos desnecessários.

Em determinados períodos do desenvolvimento de um projeto pode ser difícil devolver algum resultado devido à grande porção de trabalho envolvida. Por exemplo, na primeira vez que uma equipe criou uma nota fiscal eletrônica, demorou mais que o tempo previsto.

Há momentos em que a granularidade não pode ser quebrada, pois não adianta entregar meia nota fiscal, é preciso entregá-la inteira. Nesses casos, seremos obrigados a aumentar o tamanho da Sprint.

Outra variável que influencia na sua definição é o conhecimento de tecnologia. Se o time conhece pouco da tecnologia em questão, talvez valha a pena ter uma Sprint maior, considerando que será necessário que o time estude para produzir valor. O mesmo pode ocorrer com uma equipe composta de muitos indivíduos inexperientes.

Outra variável são as pessoas mais jovens, que tendem a estabelecer Sprints mais curtas. São pessoas normalmente acostumadas a uma vida mais dinâmica e objetivos mais próximos. Nesse caso, Sprints muito longas podem, inclusive, acarretar em problemas de desmotivação e distração. Outra questão é a "Síndrome do estudante", que implica em pessoas que agem somente em cima da hora, geralmente envolvendo indivíduos de determinada faixa etária.

Por outro lado, quando mais curta for a Sprint, maior é o \*stress \* dos envolvidos. Existem aqueles que trabalham bem sob pressão, mas outros não lidam de maneira positiva com a situação.

A duração ideal da Sprint depende muito da equipe e das pessoas envolvidas. Como dito anteriormente, a duração da Sprint atual não pode ser alterada. No entanto, ao final de um ciclo, ela pode ser revista.

**REVIEW MEETING**

**A review meeting**

No último vídeo abordamos o time-box base do Scrum, a Sprint.

A Sprint contém diversas etapas: no início há uma reunião para decidir o que fazer, trata-se do planning (planejamento). A maior parte é ocupada pelo desenvolvimento em si, que é o momento em que se coloca a mão na massa, parando todos os dias para fazer o Daily Scrum. E, por fim, ocorre a Review Meeting (reunião de revisão), revisão do que foi feito e a Retrospective (retrospectiva), parte dedicada à melhoria contínua.

O primeiro time-box que estudaremos é a Review Meeting, que na verdade é uma das partes finais da Sprint. Esse é o momento no qual o cliente e o time de desenvolvimento se reúnem para mostrar os incrementos feitos na Sprint. A Review Meeting também é uma time-box, ou seja, uma caixa de tempo rígida que não pode ultrapassar 2,5% do tempo total da Sprint.

Portanto, para uma semana de Sprint, teremos 1 hora dedicada ao consumo do tempo do cliente e da equipe para realização da review. Essa reunião é muito importante, porque paramos de "esconder" o time do cliente e vice versa. Isto é, as pessoas adquirem um rosto, tornando o trabalho mais fácil, pois são pessoas

reais que estão envolvidas no projeto.

A palavra "cliente" é recomendada pelo "Scrum Guide", porém vale trazer para junto das reuniões o "usuário". A ideia é envolver um ou ambos: cliente e/ou usuário.

Junto ao time de desenvolvimento, eles auxiliam a entender melhor o lado um do outro. As relações entre o time de desenvolvimento com o cliente e vice versa podem

não ser tão boas, já que existem alguns estereótipos, como o cliente ser alguém que apenas reclama ou o time de desenvolvimento não querer trabalhar.

Portanto, a Review Meeting é uma ótima oportunidade para unir o cliente, interessado na questão financeira, e o usuário, que relaciona-se diretamente com o produto,

junto com a equipe de desenvolvimento.

Esse é o framework, é até onde o Scrum vai!

Como executar a reunião fica a critério de quem organiza, adiante veremos sugestões de como realizá-la!

**Como fazer a review meeting**

Saindo um pouco da parte de framework, vamos dar algumas sugestões de como elaborar a Review Meeting.

Ela se inicia com o time mostrando os itens prontos, desenvolvidos durante a Sprint. A definição de pronto será abordada de maneira mais profunda futuramente.

Existem diferentes formas de se mostrar resultados, uma delas é abrir um arquivo feito no Power Point, e a equipe aponta o que foi realizado. Ao fazê-lo, no entanto,

estamos perdendo uma oportunidade incrível de descobrir como o usuário utilizaria o produto. Portanto, uma maneira de apresentar os resultados é fazer um teste de usuário,

que é pedir ao cliente ou alguém o mais próximo possível de usuário, para testar o produto.

O que pode acontecer ao pedirmos para o usuário testar uma funcionalidade específica é ela não funcionar conforme esperado. Não é raro o desenvolvedor acompanhar o usuário e

ficar se perguntando como ele pode proceder de maneira equivocada, e o mesmo ocorre na visão do usuário, que fica pensando na maneira estranha como o produto foi elaborado.

Uma maneira de realizar testes é pedir para o usuário testar uma possibilidade, sem falar como ela funciona. A visão do cliente traz uma nova maneira de compreender a aplicação.

O teste de usuário é útil para descobrir de maneira eficaz o que e como o usuário pensa.

Quando o time vivencia a oportunidade de verificar o produto sendo manuseado pelo usuário, a elaboração do produto pode se tornar mais simples no futuro. Isto é, o time se preocupará

em refletir se o produto funcionará conforme imaginado ou até começará a entender o raciocínio do usuário quando ele utiliza o sistema. Desta forma, fica ainda mais fácil acertar nas próximas elaborações.

Isto permite o surgimento de um leque de possibilidades: por exemplo, o usuário pode achar tudo maravilhoso, ou sua opinião pode ser o extremo oposto. Diante de respostas negativas, deve-se manter a calma. O Product Owner fará anotações, criará uma história de usuário ou qualquer outra maneira que esteja sendo utilizada para documentação. Essa história volta para o Product Backlog (lista priorizada do que devemos fazer no projeto).

Se o bug impedir o restante do projeto de funcionar, sua prioridade será direcionada como alta, mas se for um com menor importância, ele pode entrar no backlog de acordo com a prioridade que melhor lhe couber.

No processo de testes, o usuário também pode sugerir ideias, as quais devem ser anotadas e somadas ao backlog. Muitas vezes os desenvolvedores podem ficar temerosos em relação a novas ideias,

mas no Scrum não há o que temer. Se for uma ideia deveras importante, será preciso o P.O. (Product Owner) negociar com o cliente. Isto porque, para que uma grande mudança ocorra, será preciso retirar alguma outra atividade.

O Scrum troca de escopo com muita facilidade e frequência, e afirma que as variáveis de software não se fixam: custo, escopo, qualidade e tempo. Para Métodos Ágeis, a qualidade não pode ser sacrificada, e essa é uma regra que deveria ser levada bastante a sério. Mudar tempo e custo é bastante difícil, mas mudar escopo é mais simples.

Para finalizar a reunião nos despedimos do cliente e avaliamos se a Sprint foi bem ou mal sucedida, se bateu a meta no planning ou não.

Resumindo: na Review Meeting avaliamos os itens prontos e, se surgem novas ideias ou sugestões, criamos um novo cartão e colocamos no Product Backlog, na prioridade que lhe convém.

Voltaremos a abordar as metas quando falarmos sobre Planning Meeting.

**Definição de pronto**

No último vídeo abordamos a Review Meeting, e sugestões sobre como realizá-la. Um dos pontos mencionados foi que só poderíamos levar itens prontos.

Mas o quê exatamente significa isto?

É importante que o conceito de pronto seja claro e que haja consenso entre os integrantes do time inteiro. Portanto, vamos pensar no seu critério: ele envolve uma sequência de passos que vamos cumprindo até o item ser considerado "feito" .

Por exemplo, pensando em um exercício da faculdade, o "pronto" equivale àquilo que foi desenvolvido e testado manualmente. No entanto, trabalharemos com a ideia do pronto um pouco mais aprofundada, e colocaremos nela critérios.

Um deles é que o produto está pronto quando foi minimamente desenvolvido e testado automaticamente.

Vamos um pouco além? Adicionaremos como critério de pronto o produto homologado pelo cliente. Mas talvez seu time também sinta a necessidade de garantir certa performance ou que cada item desenvolvido seja documentado.

Isso também pode ser parte do critério de "pronto".

Para se obter um critério completo acrescentamos as seguintes ideias:

Testes de aceitação automatizados;

Desenvolvimento utilizando testes de unidade e integração automatizados;

Code Review para garantir que o código está dentro do padrão que o time aceita;

Homologação, que é pedir para a pessoa que comentou sobre o incremento de tarefas fazer a verificação. No nosso caso, isso é muito fácil e acessível. Também poderíamos pedir para o P.O. aprovar em um ambiente de homologação.

Portanto, o conceito de "pronto" varia dependendo das necessidades da equipe, que pode considerar que pronto é inclusive o produto posto em circulação. Ou seja, na Review Meeting são levados itens com que os usuários já estão em contato.

Lembrando que mesmo o produto já estando em uso, a Review Meeting é essencial, para colocar usuário e/ou cliente em contato com a equipe.

É preciso atenção, pois o critério "pronto" também deve atender a um requisito mínimo. Por exemplo, se o critério de pronto for muito grande e complexo, a Sprint nunca será pequena. Porém, com disponibilidade e flexibilidade,

o critério pode ser incrementado.

O "Scrum Guide" afirma que "o critério de pronto cresce conforme o tempo passa". Mas não necessariamente! Às vezes, quando o time amadurece, é possível retirar alguns critérios, pois determinados comportamentos já estão interiorizados

no time e não é mais necessário defini-los.

Lembrando que manter as diretrizes que compõem o conceito de pronto em formato visual é essencial. Uma ideia é utilizar o quadro do time:



mostrando a imagem do quadro do time que contém divisão em linhas e colunas Independente de ser físico ou visual, este recurso auxilia a deixar tudo mais visível. As colunas são as partes que compõem o critério, e usar essa estratégia facilita a compreensão de qual parte estamos, pois podemos mover o conteúdo

indicando que ele está realmente pronto.

Retomando: "pronto" é uma sequência de passos pré-estabelecidos pelo time, que devem ser feitos para certificar de que tudo estará funcionando corretamente. O "pronto" serve para garantir que o produto elaborado está finalizado e, ao mesmo tempo,

é um conceito resumido o suficiente para caber na Sprint, do time-box definido.

**Review Meeting**

Mesmo com o feedback que o PO dá ao longo da Sprint, tudo o que foi produzido pode ainda não ser bem o que o usuário final quer. Pensando nisso e na construção de um relacionamento pessoal entre time e cliente ou usuário, temos um momento na Sprint em

que ambas as partes participam: a Review Meeting.

Na reunião de Review, mostra-se para o cliente e, se possível, o usuário homologa tudo o que ficou pronto na Sprint. Além de cliente e usuários convidados, todos os membros do time participam dessa reunião: desenvolvedores, Scrum Master e Product Owner.

A mecânica é geralmente bem simples. O usuário final experimenta cada uma das funcionalidades que o time terminou na Sprint e dá feedback. Esse feedback é variado, afinal o cliente pode dar ok, encontrar bugs e até mesmo pensar em melhorias ou novas funcionalidades.

O PO nesse momento, toma notas desses itens e os adiciona ao Product Backlog na prioridade que lhes couber. Mesmo um bug pode não ser tão importante a ponto de entrar no topo das prioridades: pode ser que valha a pena adiar a correção de um bug não-impeditivo para colocar

uma feature que agregue mais valor no lugar.

Ao fim da reunião, definimos se o Sprint foi bem-sucedido ou mal-sucedido, de acordo com a meta definida na Planning Meeting.

Definição de Pronto

Um detalhe importante do Scrum nesse momento é definir o que levar para o cliente validar. O time deve levar apenas os itens prontos para o usuário validar. Mas a questão que surge é: o que é "pronto"?

A Definição de Pronto é a sequência de passos que um item a ser desenvolvido tem que passar, desde o zero até estar Pronto para mostrar para o cliente. A Definição de Pronto varia de projeto para projeto,

de time para time, de momento em momento do projeto. Exemplos de Definições de Pronto são:

Desenvolvido + Testado Manualmente

Análise + Desenvolvimento + Em homologação

Teste de aceitação + desenvolvimento + code review + aprovação

Quanto mais completa a Definição, mais trabalho existe para ter itens prontos na Review, mas também implica em menos chances de bugs e falhas de entendimento chegarem à Review Meeting. Quanto menos completa,

mais itens serão levados à Review, mas maior a chance de queda na qualidade.

Apenas lembre-se: seja lá qual for a Definição de Pronto do seu time, ela precisa estar clara para todos os membros do time, a todo momento!

**RETROSPECTIVA**

Terminada a Review Meeting nos despedimos do cliente e partimos para o próximo processo. Claro que o início de uma nova etapa não é tão imediato, o seguimento pode ocorrer no dia seguinte ou após a reunião de revisão.

Mas é importante destacar que entre a reunião e a retrospectiva, não existe trabalho de desenvolvimento.

O que é retrospectiva?

É o último time-box da Sprint. O primeiro é o planning, o segundo o desenvolvimento, o terceiro a Review e o último é Retrospective. Após finalizarmos a última etapa emendamos uma nova Sprint e as próximas iterações do processo.

A reunião de retrospectiva é a mais importante, pois é por meio dela que é possível exprimir o que afinal é agilidade. Esta reunião fornece a possibilidade de melhoria contínua em que pode-se "lavar roupa suja" para nos reinventarmos

para uma próxima Sprint. O problema não é errar, é ficar enroscado sempre no mesmo problema, no mesmo erro! Quando essa situação ocorre é um desperdício de tempo e esforço.

Versões mais novas do "Scrum Guide" afirmam que a cada duas semanas de Sprint deve ser feita 1h30 de retrospectiva. A versão anterior do Scrum determina que a retrospectiva deve equivaler a 5% da Sprint, ou seja, igual a 2h a cada semana

de Sprint. Particularmente, o investimento nas melhoras é algo que deve ser feito com certa constância, portanto, a versão anterior que traz a diretriz de 5% do tempo é mais adequada.

É preciso lembrar que retrospectiva não deixa de ser um time-box, portanto seu tempo de duração é rígido. Então, é bom fazer uso consciente desse tempo.

Retrospectiva é uma reunião na qual o time conversa, e dela participam:

desenvolvedores

Scrum Master

Product Owner

A não ser que exista uma razão muito forte para violar essa formação, são essas as pessoas que a compõem.

Apesar de às vezes o time ter receio da presença do Scrum Master na reunião, o ideal é que a equipe seja capaz de falar francamente com todos.

Para começar a retrospectiva, iniciamos citando o Prime Directive, inclusive vários facilitadores da comunidade ágil mundial iniciam assim. Prime Directive é a diretiva primária, que resume-se a um parágrafo com o seguinte conteúdo: não importa o que descobriremos nessa reunião, consideraremos que as pessoas agiram dessa forma devido aos conhecimentos que possuíam na época, tempo e recursos disponíveis. Considerando esses aspectos, as pessoas fizeram seu melhor, e agora devemos seguir adiante.

Algumas empresas criam adesivos de parede com os dizeres da Prime Directive, justamente para que o time visualize com facilidade a mensagem. Retrospectiva é um momento de verificarmos as possibilidades para melhorar na próxima iteração, em um mesmo projeto e com o mesmo time e contexto. Todos devem saber que é um momento de melhoria contínua, não um momento de apontar culpados.

Essa é a parte do framework, e no próximo vídeo daremos uma sugestão de como realizar a retrospectiva.

Para acessar dicas de interessantes formas de realizar uma reunião de revisão,

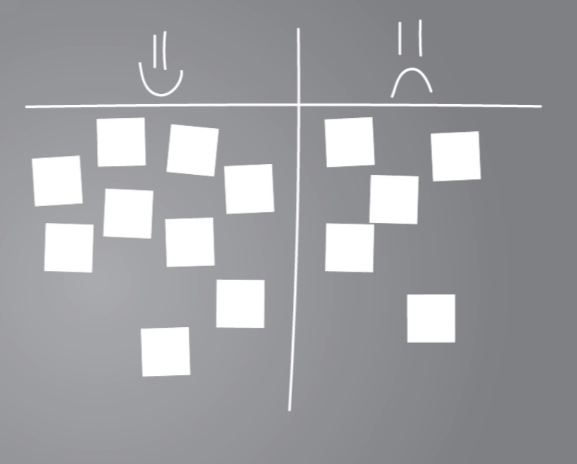
com dinâmicas associadas às equipes ou projetos gerando mais engajamento, https://www.dbccompany.com.br/8-dicas-para-realizar-retrospectivas/

**Ações e Sugestões**

Já sabemos o que é retrospectiva, portanto, vamos observar algumas sugestões de como fazer esse processo:

A retrospectiva é o momento de melhoria contínua; é quando levantamos pontos positivos e negativos.

Uma forma muito comum de fazê-lo é deixar espaço para que todos falem ou façam perguntas. Porém, se há alguém tímido, ele provavelmente não se sentirá à vontade para participar. Uma estratégia é entregar canetas e post-its para todos e pedir para que anotem pontos positivos e negativos: um ponto por post-it. O facilitador da reunião pode passar e recolher esses papéis, ele mesmo colocando-os na lousa. Dependendo da maturidade do time, e se os integrantes não tiverem medo de falar, ele mesmo pode pegar o papel e colocar na lousa.



Os post-its podem ser agrupados caso os assuntos sejam parecidos. Esses grupos de post-its criam o que chamamos de clusters.

Por onde começar?

Pessoalmente acredito que começando com resultados positivos tudo tende a se manter igual. Isto é, se está bom, continua bom, e se está ruim, continua ruim. Então, vamos iniciar abordando os aspectos negativos (mas a escolha de por onde começar depende de vocês!)

Ao observarmos os post-its, podemos verificar que os pontos que juntam aglomerações deles indicam problemas que necessitam ser analisados, ou de repente um aspecto positivo possível de ser discutido mais a fundo. Assim, as aglomerações indicam tópicos.

Supondo que iniciamos pelos pontos negativos, com vários post-its indicando que o cliente sumiu, enquanto a equipe precisava tirar diversas dúvidas com ele. Como agir diante dessa situação? O desdobramento poderia ser um muro de lamentações por parte da equipe, ou alguns integrantes poderiam partir para o famoso wishful thinking, pensando de maneira positiva para que a situação não se repita e... Nem preciso continuar para saber que essa é uma má ideia, e um caminho inadequado!

A partir da retrospectiva devem sair ações! Tanto dos pontos negativos quanto dos positivos.

No caso dos aspectos negativos levantados, qual deve ser a ação para que o problema diminua ou até desapareça no futuro?

No caso do cliente ter sumido, o que poderia ser feito?

Uma ação possível é o scrum master conversar com o cliente, explicando que suas ações acarretam em grandes impactos no projeto como um todo, portanto, o fato dele sumir no meio de uma iteração tem consequências. Ou então decidimos que o P.O. se informará melhor, mas isso pode causar mais trabalho para ele, que já está atolado de afazeres.

A solução seria ele coletar especificações mais detalhadas para que não precise mais falar com o cliente. Ou utilizamos o "chato da vez", a pessoa que ficará em contato direto com os clientes insistindo para que ele fale conosco. Ou podemos pedir para o cliente acessar diretamente o usuário, que quer muito que a funcionalidade seja desenvolvida.

Algo que já optamos fazer e até certo ponto arriscamos, e que acabou dando certo é: se o cliente não tem tempo para nos atender é porque a funcionalidade talvez não seja tão importante. Portanto, estabelecemos como regra de que, diante dessa situação, não iríamos desenvolvê-la.

As pessoas normalmente acostumam-se com qualquer restrição. Algumas são aceitas e outras não, claro. Mas dependendo da restrição, o público, cliente e usuários aprendem a lidar.

Voltando a falar de ações, independente daquilo que a equipe pensou depois da retrospectiva, é preciso deixar isso visível na área de trabalho do time (seja em ambiente virtual ou físico).

Na próxima retrospectiva, antes mesmo de levantar pontos positivos ou negativos, vamos verificar o que foi estabelecido como ação. Elas foram feitas? Deixaram de ser feitas? Melhorou? Eram necessárias? Ou seja, buscaremos aprender com as ações passadas.

**Retrospective Meeting**

O último timebox dentro de uma Sprint é o que chamamos de reunião de Retrospectiva. Em toda Sprint, problemas acontecem e sempre temos como crescer e melhorar. Para isso, contudo, é necessário parar e pensar em o que foi legal, para mantermos, e no que não foi bom, para mudarmos.

A ideia de uma Retrospectiva é pôr em prática o conceito de melhoria contínua. Nessa reunião, o time todo (PO, Scrum Master e Desenvolvedores) se foca em descobrir como melhorar ainda mais o time, o processo e o projeto já na próxima Sprint.

Há diversas mecânicas para essa reunião. Para saber mais sobre elas, você sempre pode olhar referências como o blog Fun Retrospectives ou a Retrospectives Wiki. Aqui, vamos mostrar um modelo básico e bastante utilizado.

Nesse modelo, levantaremos pontos positivos e negativos da Sprint anterior, para aprendermos com ela. Costumamos fazer isso em silêncio, para que cada um dê a própria opinião, sem enviesar os outros.

Para os pontos negativos, buscamos entender o que aconteceu para causá-los, claro, mas mais importante do que isso, definimos ações de melhoria para que tais problemas sejam evitados (ou reduzidos) no futuro.

Similarmente, analisamos os pontos positivos e, se o time sentir necessidade, definimos também lembretes para os pontos que ainda não viraram rotina.

É comum também que durante a retrospectiva, discutam-se também os itens da retrospectiva anterior, com o objetivo de validar se os problemas se repetiram e se as ações anteriores deram o efeito esperado.

Ao final da reunião, o time possui uma lista de ações a fazer na próxima Sprint. Essa lista deve ficar visível durante o andamento da próxima Sprint, para que o time se relembre delas.

**Ações**

É importante também notar que a lista de itens a fazer que sai de uma retrospectiva contém apenas ações, isto é, atividades que membros do time vão efetivamente fazer para obter algum resultado.

Note que o time não pode decidir que o cliente vá mudar seu comportamento ou que outra área da empresa vá passar a ajudá-los. Essas não são ações, são desejos de que algo magicamente vá mudar. No mundo da agilidade, isso é frequentemente chamado de wishful thinking.

Ações, por outro lado, envolvem os membros do time. Se o problema em questão é que o cliente desaparece e não conseguimos tirar dúvidas com eles:

desejo: o cliente vai entender a importância de estar presente

uma ação: o Scrum Master vai explicar as perdas e ganhos de uma maior participação do cliente

outra ação: em toda história daqui para a frente, haverá também o contato de quem pode sanar dúvidas dos desenvolvedores desse item em particular.

O resultado de uma retrospectiva é uma lista, preferencialmente curta, de ações que serão tomadas durante o próximo Sprint para melhorar ainda mais o time e o andamento do projeto.

**A Daily Scrum**

No vídeo passado abordamos as Retrospectivas, reuniões focadas nas melhorias contínuas do time. Quando falamos de agilidade, estamos falando de melhoria constante, mas isso não acontecerá se fizermos essa conversa apenas uma vez a cada Sprint.

Mesmo durante o período de desenvolvimento é preciso focar em entregar valor da melhor forma possível, sem morrer de stress.

O Scrum fornece outra possibilidade de cerimônia feita diariamente, a Daily Scrum (ou "Scrum diário"). Antigamente essa reunião era conhecida como Daily Meeting ("reunião diária"), mas o nome foi alterado justamente para afastar a ideia de uma reunião ou algo mais formal.

A Daily Scrum é rápida, com duração de no máximo 15 minutos, realizada no próprio ambiente de trabalho do time. Dessa forma, cada integrante compartilha o que fez, o que fará, e quais problemas enfrentou. Esse último aspecto serve para levantar os desafios enfrentados e superados, e os que ainda necessitam de resoluções.

A Daily Scrum é um momento no qual cada indivíduo do time precisa pensar e responder essas três perguntas: o que fez? O que fará? Quais problemas enfrentou? É parte do processo responder todo dia essas mesmas questões.

Segundo o "Scrum Guide", quem participa da reunião são os desenvolvedores. Também está implícito que o Scrum Master deve participar, e vários agilistas acreditam que, além deles, deveria ser considerada a presença do Product Owner, que é o representante do cliente dentro do time.

Diante disso, se a equipe está com algum problema que faça com que gastemos mais tempo que o esperado, é bom que o product owner saiba disso. Ele é responsável pelos negócios, portanto, é a pessoa certa para notar um problema, inverter prioridades e entregar o máximo de valor possível ao cliente.

Além disso, o P.O. faz parte do time, e essa é uma forma de criar confiança entre os desenvolvedores e a pessoa de negócios. Todos estão juntos pelo mesmo motivo! Portanto, o P.O. deve participar dessas conversas para ficar a par do processo de desenvolvimento do projeto, tanto para aspectos positivos quanto negativos.

Assim, o P.O. pode conversar com o time de maneira a rearranjar as atividades que estão sendo desenvolvidas para que tudo fique em ordem em casos em que ocorra adiantamentos ou atrasos.

Portanto, a Daily Scrum é: Uma reunião diária; com duração de até 15 minutos; feita em pé para cansar caso ultrapasse os 15 minutos; três perguntas principais devem ser respondidas: O que fez? O que fará? Quais problemas enfrentou?

toda a equipe participa: desenvolvedores, Scrum Master e Product Owner.

Agora, quando exatamente esta reunião acontece? Esse é assunto do próximo vídeo!

**Quando fazer a Daily Scrum**

No último vídeo abordamos a definição de Daily Scrum, e agora vamos descrever algumas regras necessárias para facilitar a organização desse tipo de reunião diária.

É importante fixar que ela deve ser realizada sempre no mesmo horário e lugar. Claro que fixar isto em empresas que possuem regras de horários mais flexíveis pode ser complicado, portanto, daremos algumas dicas, saindo do prisma do framework.

É essencial que todos os integrantes do time estejam disponíveis para a reunião, dessa forma, marcá-la no início do expediente implica em possíveis ausências, pois algumas pessoas ainda não chegaram no trabalho.

Diante disso, é possível pensar que é mais plausível escolher o primeiro horário em que todos da equipe estejam na empresa. O problema de fazer isso é que a pessoa que acabou de chegar ainda está se situando, entrando no ritmo do trabalho. O mesmo acontece quando se opta pelo último horário. Para quem está encerrando o expediente, fica um pouco complicado, pois a pessoa está na iminência de ir embora.

Uma dica é evitar momentos extremos do dia - início e final do horário de trabalho - e utilizar intervalos existentes na rotina. Por exemplo: o time para e almoça ao meio dia? Então o Daily Scrum pode ser realizado às 11:45, ou, se metade do time volta do almoço e metade sai depois, o intervalo entre os almoços pode ser um momento bacana.

Outra opção é escolher outra pausa, algum momento utilizado para descontração do time. Por exemplo, durante muito tempo na Alura o Daily era aplicado antes da pausa da tarde, assim, quando as pessoas começavam a ficar com fome, já sabiam que esse era o momento do Daily. O interessante de se fazer isso é que mesmo a reunião acabando, o time ainda conversa sobre isso no momento do lanche.

A dica é fazer a reunião em momentos de pausa ou próximos a eles. Evitando que caia no início ou final do dia, e mantendo sempre o mesmo horário e lugar pra que não seja preciso pensar nessa logística todos os dias.

**Daily Scrum**

No capítulo de Retrospectiva você aprendeu sobre essa importante ferramenta para promover melhoria contínua, mas esperar o fim da Sprint para resolver probleminhas do dia-a-dia seria um grande desperdício de tempo.

Para resolver obstáculos do dia-a-dia e para todos saberem como está o andamento das tarefas e histórias nessa Sprint, o Scrum tem ainda mais uma cerimônia. Todos os dias, no mesmo horário e no mesmo local, a equipe se reúne para responder 3 perguntas simples:

O que fiz desde o último Daily Scrum?

O que pretendo fazer até o próximo?

Quais problemas me atrapalharam?

O timebox para essa reunião é de apenas 15 minutos por dia e, como qualquer outro timebox, deve ser respeitado. Para isso, é importante lembrar que essa é uma reunião expositiva: o objetivo aqui não é resolver o problema, mas sim apontá-lo. Se alguém souber como ajudar, apenas indica que sabe e após o fim do Daily os interessados se reúnem para conversar.

Há uma divergência de opiniões sobre quem deve estar presente nessa reunião. O Scrum Guide aponta que os desenvolvedores respondem as perguntas e o Scrum Master apenas assiste a reunião, eximindo a participação do P.O.

Na prática, vemos uma real vantagem em ter o P.O. presente nessa reunião, já que ele precisa saber do andamento da Sprint, entender os problemas que estão desacelerando desenvolvedores e ainda ajudar a alterar o plano de ação, se o time estiver atrasado. Assim, recomendamos fortemente que o P.O. esteja, sim, presente nessa reunião.

Com o Daily Scrum, o time evita com que problemas durem muitos dias e que desenvolvedores façam trabalhos repetidos - tanto trabalhando sobre uma mesma feature, quanto sofrendo para resolver um problema que outro sabe solucionar.

**A reunião de planejamento**

Neste vídeo será abordada a etapa que ainda não vimos: a Planning Meeting. Esse é o momento em que se inicia a Sprint.

A Planning Meeting limita-se a 5% da Sprint, e é um processo de planejamento. O time se reúne com o Scrum Master, que trabalha mais a parte da facilitação, os desenvolvedores, para pensarem na parte mais técnica de como realizar certas tarefas. Nisso, o Product Owner é responsável por priorizar um grupo de itens que agregue o máximo de valor possível.

O P.O. chega na reunião com o Product Backlog, lista ordenada ou priorizada dos itens a serem feitos durante o projeto. A ideia é sair da reunião com a lista priorizada dos itens e sub-itens técnicos que o time está comprometido a fazer durante a Sprint em questão. E também com uma meta, uma frase que exprima o valor a ser entregue ao cliente.

Resumindo, a Plannig Meeting é uma reunião de planejamento que reúne a equipe inteira. Entramos nela com uma lista de todos os afazeres e saímos com outra, de tarefas específicas a serem concluídas. Lembrando que o Planning equivale a um tempo de 5% da Sprint, se o time-box é de 2 semanas, passaremos 4 horas planejando. Para isto, o P.O. deve ter passado um tempo considerável pegando o topo do Backlog, os itens mais importantes, e refinando-os.

Esta parte de refinamento do topo do Backlog é um dos trabalhos do P.O. durante todo o desenvolvimento. Ele chega à reunião pronto, com o topo do Backlog refinado, itens menores que também façam sentido para os clientes e pessoal técnico.

Podemos consumir cerca de 10% do tempo do time, pelo menos no início, para perguntar o que faz sentido e o que não faz. Isso é feito individualmente, para que na Planning Meeting o P.O. traga tudo pronto.

Isto é o que o framework nos fornece, e no próximo vídeo falaremos sobre o método para fazê-lo.

**Mecânica da reunião**

No vídeo passado falamos sobre o framework Scrum na parte do Planning Meeting. Estamos falando sobre a estrutura, a moldura. O conteúdo ou como vamos proceder decidimos depois.

Neste vídeo vamos falar sobre como implementar o Planning Meeting, em que o P.O. chega com o Product Backlog, a lista de coisas priorizadas, e inicia a reunião apresentando o item de maior prioridade para a visão de negócios. Ele deve lidar apenas com a questão do cliente, sem "dar pitaco" em relação à parte técnica.

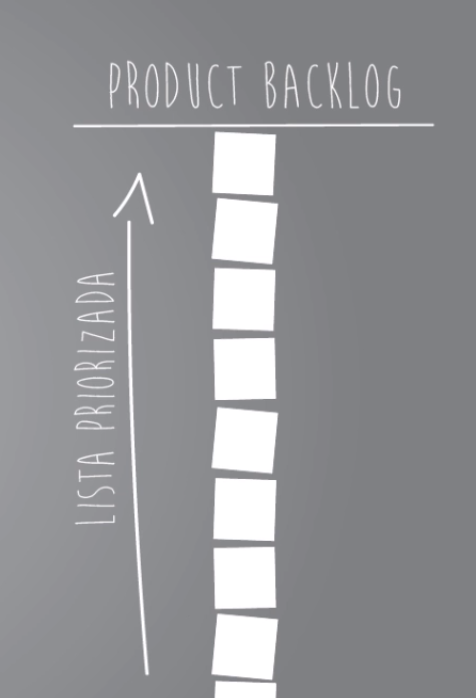
Os desenvolvedores vão discutir sobre o item, pensar na parte técnica, talvez quebrar o item de negócios em tarefas menores e mais técnicas e, a partir disso, estimar quanto de esforço deve ser empregado nesse tipo de tarefa.

Ou seja, não chegamos a discutir a nível de código. A ideia dessa discussão é compreender a complexidade embutida no processo e fornecer uma estimativa do que fazer. Assim, seguimos por cada item da lista de prioridades até passarmos um pouco do quanto o time consegue de fato fazer. Passando disto, vamos para a próxima parte da reunião, de negociação daquilo que realmente cabe na Sprint.

Por exemplo, há um item na lista que é muito grande, então iremos deixá-lo para a próxima Sprint. Isso não acarreta em nenhum problema, pois puxaremos algum outro item para cima.

Claro que isso é decidido em equipe, e se o P.O. achar que não é tranquilo deixar esse item maior de lado, é porque provavelmente não é uma boa ideia.

A ideia é ponderar, não podemos deixar muito tempo de trabalho vago e tampouco ficar com a corda no pescoço, com prazos muito estreitos.

O Scrum também faz parte da agilidade, sendo o objetivo trabalhar em ritmo constante indefinidamente. Isto faz parte dos princípios ágeis, e para isso é preciso ter uma noção do quanto conseguimos entregar sem nos sobrecarregarmos, e sem deixar espaços não preenchidos.

A ideia é que essa análise seja negociável, pois os desenvolvedores podem visualizar algo que pode dar problema no futuro, uma vez que a dívida técnica é muito grande. Assim, em vez de pegar uma tarefa nova a ideia é investir o tempo na solução da dívida técnica. Inevitavelmente esta falha técnica precisará de atenção, e isto pode custar mais caro se continuarmos adiando a situação.

A negociação pode ser feita conforme a demanda dos participantes. O essencial é pensar no Sprint Backlog, lista priorizada do que o time deve fazer na Sprint. Lembrando que não é nesse momento que decidimos quem vai fazer o quê: isso cabe ao escopo do Daily.

O que nós obteremos ao final desse processo é o que o time irá entregar ao final da Sprint.

Uma vez que o grupo está fechado é preciso se perguntar: qual é o principal valor de negócio que conseguimos tirar do time? Assim, a equipe decide, por exemplo, que três dos itens devem agregar muito valor de negócio à Sprint, portanto, a meta que vamos traçar pode ser feita a partir deles!

Falaremos mais sobre metas no próximo vídeo!

**Planning Meeting**

No Scrum, toda Sprint começa com uma reunião para entendermos os itens a serem feitos, planejarmos o que cabe no tempo disponível e definirmos a meta para o time nessa Sprint. O time todo (todos os desenvolvedores, Product Owner e Scrum Master) participa dessa reunião.

O timebox dessa reunião é de 5% do tempo total da Sprint, isto é, temos 4 horas para planejar o que será feito em uma Sprint de duas semanas. E, como qualquer outro timebox, ele deve ser estritamente respeitado.

Em preparação para essa reunião, o P.O. já deve ter olhado as histórias mais prioritárias do projeto, confirmando o entendimento delas com o cliente, melhorando sua clareza, quebrando grandes funcionalidades em partes menores que já agreguem valor para o cliente, etc. Esse processo é conhecido como grooming, ou refinamento, do topo do Product Backlog.

É importante que o P.O. já tenha feito esse processo antes do Planning começar, para que o timebox seja suficiente para essa reunião!

**Mecânica da reunião**

Existe uma certa divergência dentro da comunidade ágil sobre como essa reunião deve acontecer, mas isso é menos importante já que o Scrum é um framework, isto é, foca em o que fazer em vez de como fazê-lo. A forma mostrada aqui é uma das várias utilizadas mundialmente.

A reunião começa com o P.O. explicando o item de maior prioridade da visão de negócios, isto é, da perspectiva do usuário/cliente da aplicação. Então, os desenvolvedores tiram dúvidas de entendimento com o P.O., quebram a funcionalidade em sub-itens técnicos e discutem em alto nível sobre as dificuldades de implementação desse item, apenas o bastante para conseguirem atribuir a ele uma estimativa de esforço.

Esse processo se repete com o próximo item mais prioritário ainda não estimado, o próximo, o próximo, etc... mas só até passar um pouco do que o time historicamente consegue fazer em uma Sprint!

Então, os desenvolvedores e o P.O. negociam quais desses itens mais prioritários realmente serão feitos na próxima Sprint e, assim, definem o Sprint Backlog: a lista priorizada das histórias (itens de negócio) e tarefas (sub-itens técnicos) que o time planeja fazer nessa Sprint.

Uma vez que o Sprint Backlog foi decidido, o time olha para as histórias selecionadas e cria uma meta: uma frase que exprime o maior valor que esse Sprint vai trazer para o usuário! Vale qualquer frase que mostre para os desenvolvedores o que está em jogo e como um Sprint bem sucedido vai impactar positivamente na vida dos usuários da aplicação.

Resumindo, como o nome já indica, na Planning Meeting o time faz o planejamento do que será feito na iteração. Antes de começar a reunião, o PO traz o Product Backlog já priorizado e com histórias mais prioritárias já refinadas e, ao final do Planning, o time tem em mãos a lista de histórias a serem feitas, as tarefas para cada uma delas e uma meta motivante.