**Projetos ágeis com SCRUM**

\*Introdução ao SCRUM - Conceitos básicos

-Desafios do desenvolvimento de software

● Para atender os objetivos de negócio, o software vem pra ajudar a empresa a atingir esses objetivos;

● Processo de desenvolvimento: concepção, analise/design, desenvolvimento, testes e implantação);

● Apenas 20% do software é utilizado

-Gestão de projetos trad x ágil

● Tradicional (waterfall): só permite que o projeto avance quando uma fase está completa;

● Ágil: software construído por partes (incremental) e cada parte executa-se em um ciclo (interativo).



● Em projetos trad. Vc corre o risco de descobrir que estava errado depois de meses. Com o SCRUM, vc descobre que estava errado em no máx 30 dias.

● Com ágil, o **escopo é definido ao longo do projeto.**

-O que é ser ágil?

● Rapidez (mudança) e desembaraço;

● Fazer coisas complexas de forma simples;

● Equipe comprometida com os objetivos;

● Maior valor para o cliente.

-SCRUM

● um dos frameworks de gerenciamento de projetos ágeis;

● projetos usando equipes pqns e multidisciplinares pra produzir melhores resultados;

● conversar + escrever – ;

● demonstrar o software aos usuários e obter feedbacks constantes;

● requisitos mudam ao longo do tempo;

● aprender progressivamente com o uso do software

-Razões pra adotar o SCRUM:

● desenvolvido e entregue em partes menores (2 a 4 semanas) com constante feedback;

● melhor gerenciamento de riscos (redução de incertezas);

● comprometimento, motivação e transparência da equipe (daily meeting);

● maior valor pro negocio (priorização do backlog);

● usuários envolvidos durante o ciclo;

● aplicação das lições aprendidas (melhoria continua).

-Time SCRUM:

● equipes que se auto-organizam;

● as tarefas são do time todo e todos são responsáveis;

● forte comprometimento com os resultados.

-Pilares do SCUM:

● adaptação;

● transparência;

● inspeção.

\*Papeis e responsabilidades de cada um do time

-Product Owner (PO)

● dono do negócio ou representante da área de negócios;

● uma pessoa só;

● entende a demanda e extrai o maior valor possível;

● define as funcionalidades do software e as prioriza de acordo com o valor do negócio;

● garante que o time de desenvolvimento entenda os itens do backlog no nível necessário;

● prioriza o backlog.

-Scrum Master (SM)

● garante o uso correto do SCRUM;

● NÃO É gerente de projetos;

● age como facilitador, ajudando por ex o PO a explicar algo pro time;

● ajuda o PO no planejamento e estimativas do backlog;

● ajuda a equipe a remover impedimentos;

● treina o time em autogerenciamento e interdisciplinaridade.

-Time de desenvolvimento (DEV)

● possui habilidades suficientes pra desenvolver, testar, criar e desenhar, ou seja, tudo que for necessário pra entregar o software funcionando.

\*Cerimônias do SCRUM

-Time box: tempo máximo pra fazer uma cerimônia ou Sprint.

-O que é Sprint?

● corrida, arrancada, principal evento do SCUM, dura no máximo 30 dias corridos;

● composta por: planejamento, reuniões diárias (daily meeting), revisão (review) e retrospectiva;

● planejamento: todos participam (PO, DEV E SM), 8h de reunião (4h do PO mostrando ‘o que fazer?’ e 4h do time DEV discutindo ‘como fazer?’), time explica pro PO o que realmente vão conseguir entregar ao fim daquela sprint;

● reuniões diárias: 15 min, todos participam, time DEV responde ‘o que fez no dia anterior? O que esta programado pra hj? Tem algum impedimento?’, reunião é feita de pé;

● revisão da sprint: time DEV apresenta pro PO o trabalho feito, dura 4h, como é feita no ultimo dia não da mais pra modificar no software caso o PO não goste, aí só na próxima sprint;

● retrospectiva: reunião da equipe para lições aprendidas, transparência para que evitem erros nas próximas sprints, dura 3h.

\*Diferença prática entre gestão tradicional x gestão ágil

O ágil responde muito mais rápido aos clientes, mudando conforme necessidade da empresa e dos consumidores da mesma.

1. Papeis e responsabilidades – Product Owner

-Quem é o Product Owner (PO)?

● O profissional que tem a visão do que será desenvolvido, as necessidades a serem atendidas, o publico que sera atingido pelos serviços e os objetivos a serem alcançados.

● Tbm é responsável por definir a ordem em que as atividades serão resolvidas, planejar os ajustes caso os objetivos não estejam sendo entregados de acordo com o combinado nas Sprints e, em ultimo caso, cancelar a Sprint quando as atividades não puderem ser entregues e replanejar a Sprint.

\*PO e Scrum Master (SM) não são chefes, os dois devem ser uma dupla sem divisão hierárquica.

\*Refining e Planning

-No refining, o PO apresenta pro time as histórias que deverão ser apresentadas no planning. O time recebe previamente o que será desenvolvido pra tirar duvidas e o PO tem tempo pra ir atras do que não souber responder pra ter tudo respondido no planning.

-o planning é dividido em 2 etapas:

● 1ª parte: o PO pega os itens mais refinados e priorizados pro backlog e apresentar pro time. É muito importante que seja feita a leitura completa de cada história pra tirar todas as dúvidas do time, aí que o time analisa e ve o que entra e o que não entra na sprint;

● 2ª parte: o PO não deve participar, o time pega cada história e escreve quais atividades devem ser desenvolvidas pra que a história seja considerada entregue. O time pode cortar e diminuir as histórias a serem entregues.

-Com as histórias mapeadas e as atividades descritas, é definido o sprint backlog, com essa info, o PO deve validar qual item é mais prioritário e que será considerado o objetivo principal da sprint, ou seja, se essa atividade der errado, o resto da sprint não fará sentido.

Obs: **Merge** é junção dos códigos fontes de cada sprint.

\*Release Planning: lançamento de software ou de sua nova versão. Cada vez que um produto desse é criado ou modificado, o fabricante e seus devs decidem sobre como distribuir o novo produto ou a modificação às pessoas que o utilizam. Existem dois tipos de release planning:

● Release Planning de Múltiplas Squads: vários times de dev agrupados fazendo coisas distintas (que podem ou não ter correlação entre as atividades) e que ao final da sprint, devem ser agrupadas em um único release pra serem implementadas em produção. Apesar de times e atividades diferentes, devem obedecer às mesmas regras técnicas, para que uma squad não impacte em outra squad;

● Release Planning de Projeto: o PO possui uma demanda muito grande, sendo necessário quebrar as histórias em varias sprints. O PO deve compreender a real dimensão da demanda e de cada historia, e deve quebrar as historias o máx que puder. EX: o PO deve entender que não é possível desenvolver uma nova plataforma de autenticação de usuário em uma única sprint e tbm entender que uma historia com titulo de ‘novo autenticador’ não é suficiente pro time compreender a dimensão do proj e quebrar corretamente as atividades.

Obs: **PO não é gerente de projeto!**

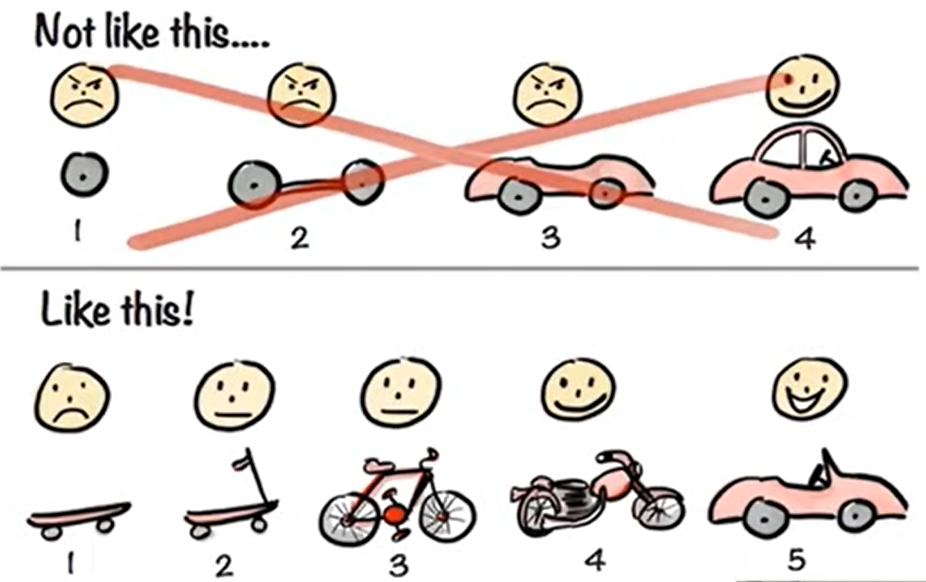
1. Analisando escopo e definindo prioridades

●Analisar o escopo é muito complexo, pois normalmente tem varias pessoas envolvidas, de vários departamentos diferentes e com objetivos diferentes. A **definição do escopo é uma das partes mais criticas do projeto**, pois é quando será definido o que será desenvolvido e é do artefato dessa fase que serão extraídos os benefícios a serem atingidos. Uma definição de escopo ruim pode gerar beneficios errados ou menos assertivos.

●Uma forma de **definir o escopo corretamente é invertendo a ordem e entendendo qual o objetivo ou valor que vc quer atingir**, antes mesmo de definir o escopo, assim vc terá claro onde quer chegar e terá mais possibilidades de caminhos a seguir pra atingir o objetivo.

●O método mais indicado pra definir o escopo é entender qual o ganho primeiro.

Ex: na 1ª imagem, vemos que o veículo foi entregue apenas no final (da sprint), fazendo com que o cliente ficasse sem um meio de locomoção até o final do projeto. Na 2ª imagem, vemos que, apesar de não ser o objetivo final do cliente, ele recebeu ao longo do tempo meios de locomoção que o ajudaram enquanto seu carro -o objetivo final do projeto- não estava pronto. Nesse caso, o cliente recebe o produto mais perto do imaginao possível, pois ele foi dando seus feedbacks durante o projeto fazendo com que o carro saísse exatamente da forma que ele queria.



● Por isso é importante validar cada entrega e ver com o cliente se o produto está dentro da expectativa, para assim, realizar as adaptações necessárias pra atender cada vez mais os objetivos do cliente.

\*Product Backlog

-Deve conter apenas o conteúdo de um produto. Cada produto tem seu próprio backlog e normalmente esse backlog nunca acaba. À medida que novas sprints são entregues e inspeções são realizadas, novos épicos são gerados pra manter os produtos em evolução.

-Composto por épicos e estórias:

● Épico: incremento sem muito detalhamento (serve pra definir as atividades de maneira básica, sem investir muito tempo em algo que não será desenvolvido agora ou que nunca será desenvolvido), ajuda a te direcionar aos caminhos que deve seguir. Quando um épico é priorizado, ele é detalhado, e a partir daí, um épico pode gerar N estórias.;

● Estória: detalhamento dos épicos (um épico normalmente se divide em várias estórias, onde ficam descritos o que deve acontecer e suas regras de negócio), deve ser descrita com o max de detalhes possivel.

-PO deve encontrar sinergia entre outros projetos contidos no backlog, por exemplo, pode ser que tenha um projeto com prioridade baixa, mas que tenha sinergia com um proj de prioridade super elevada, qnd o PO combina os dois projetos, eles se somam e a empresa acaba tendo um ganho inesperado. Essa é a importância do PO não só conhecer, mas dominar todo o backlog.

\*Escrevendo uma estória

● Nome da estória;

● Descrição da estória (eu, como, quero, quando);

● Regras do negócio (separar regras de front-end e back-end);

● Tela (link ou imagem das telas a serem desenvolvidas);

● KPI (quais objetivos/valores a estória precisa atingir);

● Tagueamento (como a estória será “tagueada” pra poder mensurar os KPI);

● Critérios de aceite (qual o passo-a-passo de todos os caminhos possíveis a estória deve cumprir pra que seja aceita).

\*Gestão de riscos em projetos ágeis

● Riscos positivos: muito ignorado nos proj, um dos fatores de maiores ganhos no desenvolvimento de sistemas. É comum que o que esteja sendo feito por um time pode ser aproveitado por outro time, mas se não houver acompanhamento, o PO pode perder essa oportunidade. É importante serem mapeados pra que outros itens no backlog sejam priorizados;

● Riscos negativos: itens que podem afetar o prazo, custo ou escopo de um proj de forma que pode inviabiliza-lo.

1. Papel do PO na transformação digital

\*Transformação digital:

● Processo no qual as empresas usam tecnologia pra melhorar o desempenho, aumentar o alcance e garantir resultados melhores. É uma mudança estrutural nas organizações, dando um papel essencial pra tecnologia. É mostrar para as pessoas como usar a tecnologia já disponível há um tempo da melhor forma possível, extraindo o melhor da tecn pra otimizar nosso trabalho no dia a dia. É sobre como podemos diminuir processos e atritos e potencializar resultados. Está nos objetivos das maiores empresas, mas muitas não entendem o que é de fato a transformação digital;

● **O papel do PO nessa mudança é fundamental**, pois representa a mudança na forma como os proj são tratados, a maneira como conhecimento é inseminado, entende a necessidade do cliente. Se antes vc precisava de um analista de proj dentro de cada depart pra que este acompanhasse e cobrasse o gerente de projetos e o departamento era o único dono do conhecimento que ali era gerado, com a mudança de mindset, teremos cada vez menos pessoas fazendo mais coisas e pra isso, precisaremos de tecnologia nos suportando pra aguentar uma maior carga de trabalho.

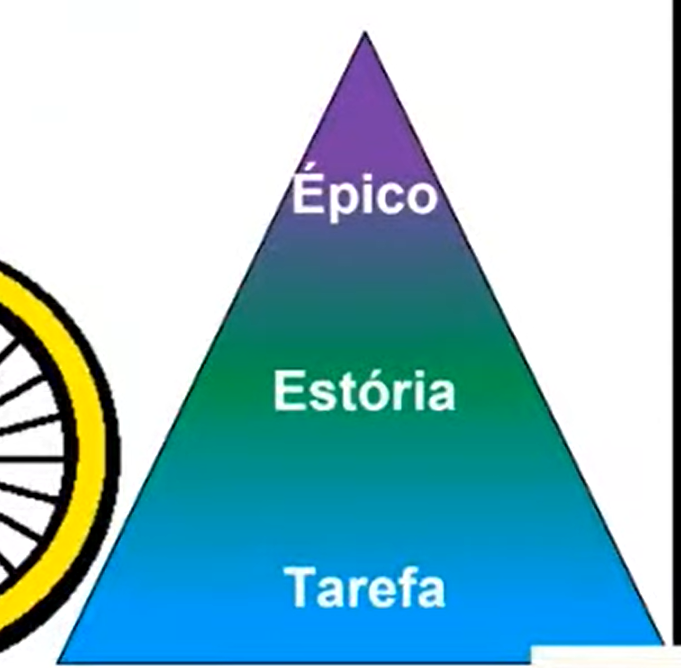
\*Conceitos e planejamento de tarefas

1. Estórias X Tarefas

● Imagine que vc quer montar uma bicicleta (objetivo do projeto). O objetivo do projeto estará pronto quando a bicicleta estiver pronta, logo, os épicos são ‘aro, garfo, banco, rodas, sistema de freios e sistema de tração’. **O épico é o objetivo macro a ser atingido e, pra isso, deve ser dividido em objetivos menores**, logo, o épico ‘garfo’ consiste nas estórias ‘guidão, amortecedores e garfo’, ou seja, o guidão é uma estória do épico garfo. O ‘pedal’ é uma estória do épico ‘sistema de tração’ e a ‘manopla de freio’ é uma estória do ‘sistema de freios’.

● As estórias por si só não dão a dimensão real do que deve ser feito pra que o trabalho seja concluído, é aí que entram as **tarefas**: pra estória ‘garfo’, é preciso realizar a tarefa ‘montar o guidão, montar o garfo, montar o garfo no quadro, montar as rodas no garfo e montar o sist. de freio nas rodas’ (considerando que as peças já estão prontas, se fossem ser produzidas, teríamos épicos de ‘usinagem’ e outros relacionados à produção das peças.

● Resumindo: **uma estória é um conjunto de tarefas e um épico é um conjunto de estórias.** Começamos pensando nos épicos dos projetos e quando esse proj for priorizado, nós quebramos as estórias e nesse momento, se for necessário, quebramos os épicos em mais épicos. Quando as estórias são priorizadas nas sprints, elas são quebradas em tarefas e neste momento tb podemos quebrar uma estória em mais estórias.

****

**●** Uma estória é uma tarefa descrita em nível de negócio. Uma tarefa é um conjunto de atividades que o time de dev desempenha pra entregar uma estória.

1. Critérios de aceite, estimativa e planejamento de tarefas

\*Critério de aceite:

● É uma lista de critérios que precisam ser alcançados pra que a User Story atenda os requisitos do usuário e seja aceita pelo PO. Os critérios tem objetivos de definir limites pras US e ajudar o PO a detalhar em alto nível o que é necessário pra entregar valor ao cliente. Conclusão: **o critério de aceite é o objetivo que aquela estória tem que concluir.** Se uma estória não tem critério de aceite, muito provavelmente o PO não entendeu a demanda e apenas escreveu a estória pra cumprir tabela.

● Como PO a primeira coisa descobrir de uma demanda é o objetivo dela, onde se quer chegar, qual objetivo queremos atingir? Muitas demandas não são claras nem pros demandantes

\*Estimativa e Planejamento (planning poker):

● Método mais usado pra se fazer **estimativas do tamanho das tarefas** em times ágeis (mas não é a única ferramenta disponível);

● Cada um do time recebe um deck com cartas contendo os números da sequência de Fibonacci, quando todos estão prontos, as estórias são lidas e as tarefas são criadas. Após isso, o time vota jogando uma das cartas falando qual o tamanho daquela atividade (**levando em conta a complexidade da tarefa, o trabalho manual em si e o tempo que será gasto**);

● Uma tarefa muito simples de se fazer pode receber voto 1, uma mais difícil pode receber 13. Quando uma tarefa recebe 20 pontos, o time entende que ela é muito difícil e deve ser quebrada em 2 ou mais estórias ou tarefas;

● Os membros do time que derem a maior e a menor nota devem justificar sua pontuação, pois eles podem ter pensado em algum detalhe que os outros membros do time não consideraram. Esse é o maior ganho que temos, pois o time fica mais próximo e um ajuda o outro.

1. Relacionamento com clientes/stakeholders

● O **stakeholder é uma pessoa ou grupo que legitima as ações de uma organização e que tem um papel direto ou indireto na gestão e resultados dessa mesma organização**. Desta forma, um stakeholder pode ser afetado positivamente ou negativamente, dependendo de suas politicas e formas de atuação. Alguns exemplos de stakeholders são: funcionários, gestores, gerentes, proprietários, fornecedores, concorrentes, ONGs, clientes, o Estado, credores, sindicatos e diversas outras pessoas ou empresas que estejam relacionadas a uma determinada ação ou projeto;

● Cliente interno ou stakeholder é a pessoa (ou grupo de pessoas) interessada no projeto em questão. Não necessariamente é o patrocinador de um projeto, o patrocinador pode ser o diretor de um departamento e o stakeholder pode ser um analista ou gerente deste mesmo departamento. **O PO deve saber conversar com todas essas pessoas**, daí a necessidade de saber fazer um bom jogo de cintura.

● Muitas pessoas abrem projetos que não trazem grandes benefícios pra empresa, mas sim grande ganho político pro stakeholder (muitos se aproveitam de demandas de grande relevância pra encaixar demandas de baixa relevância no meio, como por exemplo, trocar o excel da empresa por um sistema que muitas vezes tem um custo mais alto).

● Alguns casos podem não ser viáveis pra empresa, mas trazem benefícios, como melhora na imagem da cia como uma empresa de alta percepção de tecnologia, um ótimo lugar pra se trabalhar, etc... No fim, esses itens podem ser mensurados, mas de outra perspectiva, o que torna complexo.

\*Rotinas de um time ágil

1. Daily meeting e retrospectiva

\*Daily meeting:

● reunião diária, sempre no mesmo horário, durando 15min;

● todo o **time dev deve participar** (SM e PO não são obrigatórios);

● saber o que cada um de dev está fazendo e um ajuda o outro;

● qualquer assunto pode ser tratado, desde que agregue valor ao projeto.



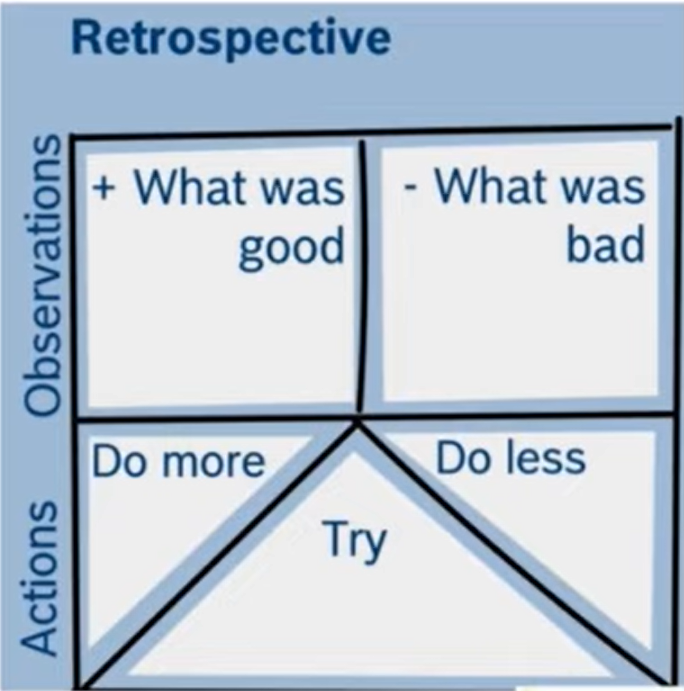
\*Retrospectiva:

● acontece apenas 1x na sprint (**após a conclusão da mesma**);

● quem participa é o **time dev e o SM**;

● a presença do PO é importante, mas não obrigatória;

● discutido os pontos bons e ruins da sprint, o que deve melhorar ou não se deve fazer e o que podemos tentar fazer na proxima sprint



Obs: **Daily é focada nas tarefas do dia-a-dia, a retrospectiva é focada na sprint como um todo.**

Obs²: **Um time maduro realiza essas duas cerimonias por conta própria, sem precisar que o PO ou o SM fiquem chamando.**

1. Refinamento e review

\*Refinamento

● cerimonia não oficial do SCRUM;

● muito utilizada pra que o time, junto com o PO, discuta como será a prox sprint, quais serão os entregáveis e adiantar possíveis duvidas;

● aumentar o entendimento da demanda por parte do time e melhorar a qualidade da entrega e aumentar o retorno esperado pro projeto;

● time dev faz varias perguntas pro PO pra entender do que se trata a demanda, por isso é importante que a estória esteja bem escrita nesse momento pra que fique menos duvidas possíveis;

● facilitar a planning (**acontece um passo antes da planning, no final da sprint anterior)** e torna-la mais efetiva e assertiva;

● **a presença de todo o time é obrigatória.**

\*Review

● todo o time deve estar presente;

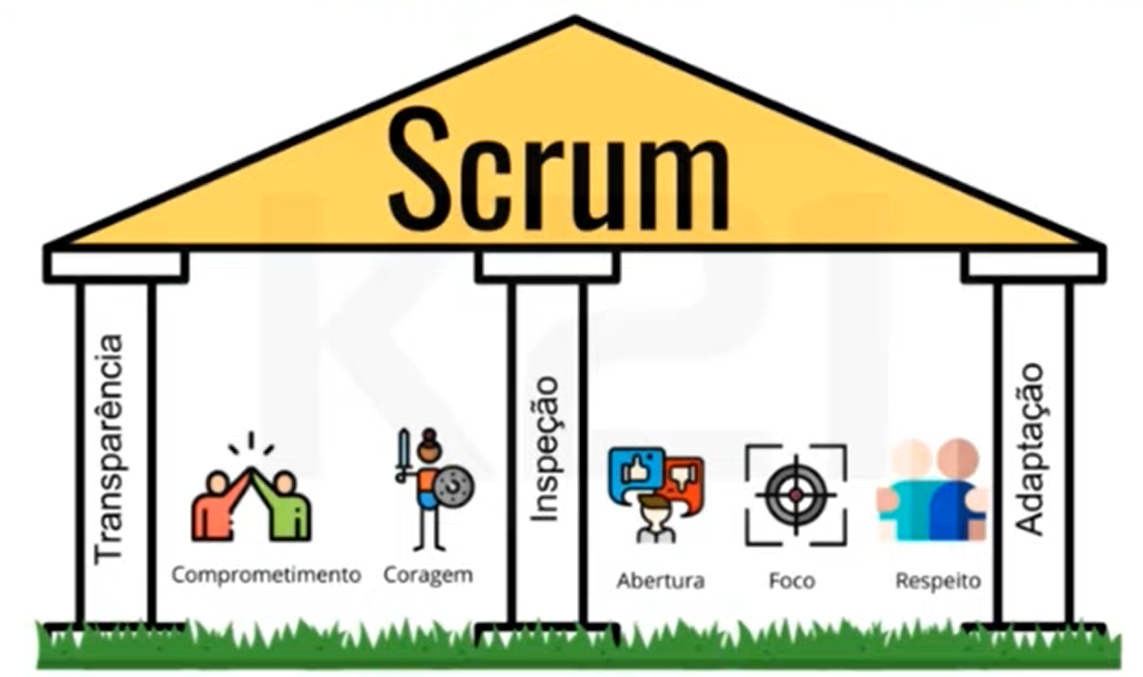
● time dev apresenta e mostram o trabalho que cada um desenvolveu na sprint anterior;

● tiram duvidas técnicas que o publico presente tenha (duvidas de negócio são tiradas pelo PO);

● nessa cerimonia são convidados todos os interessados na entrega, sejam stakeholders, demandantes diretos, lideres técnicos e outros gestores;

● a ideia principal é **ver o que está sendo entregue**, validar se está de acordo com o solicitado inicialmente, se houve mudança no meio do fluxo, bater com o que foi definido no refinamento e na planning e, o mais importante, **se agrega valor ao negócio**.

1. Maturidade da equipe



● um time não será maduro se a empresa não for tbm, não adianta o time ser ágil e a empresa ser waterfall;

● o amadurecimento do time está diretamente ligado ao ambiente em que ele está situado;

● um time que possui **transparência** sobre o que acontece ao seu redor, as demandas que estão por vir e pra saber se a empresa está indo bem ou não, tem muito mais capacidade de desenvolver suas atividades;

● um time que pode retornar para entregas anteriores, verificar o resultado alcançado e cruzar com os resultados esperados, consegue entender melhor o comportamento do seu cliente final e assim, se **adaptar** pra novas entregas e faze-las de maneira cada vez melhor e mais assertiva;

● um time maduro é o que de fato consegue vivenciar os pilares do SCRUM e não fica dependente do SM pra resolver seus problemas, um time que tem iniciativa pra resolver seus problemas;

● um time maduro auxilia o PO a escrever as estórias e o ajuda a quebrar as estórias de forma a encaixa-las perfeitamente na sprint;

● um time maduro não é puxado pelo PO ou pelo SM, e sim, ele que puxa o PO e o SM acaba não sendo mais necessário no dia-a-dia tornando-se um consultor pontual;

● **um time maduro é o time que anda sozinho.**