Projetos Ágeis com SCRUM

# Introdução ao SCRUM

## Introdução a Gestão de Projetos e ao Scrum

### Objetivos da Aula

* Conceitos Básico
  + Quais são os desafios na gestão de projetos para desenvolvimento;
  + Qual a diferença entre o projeto tradicional e gestão de projetos ágeis.
* Papéis e responsabilidades de cada um do time
  + O que é um Scrum Master e o que ele faz;
  + O que é o PO e o que ele faz;
  + O que é o time DEV e o que eles fazem.
* Cerimônias do Scrum
  + O que uma Daily;
  + O que é uma Sprint Planning;
  + O que é uma Sprint Review;
  + O que é uma Retrospectiva;
  + O que é um timebox;

### Conceitos Básicos

#### Desafios do Desenvolvimento de Software

Como toda empresa, seja ela pequena ou grande, startup, cada empresa tem um objetivo, um objetivo de negócio, e para atender esse objetivo o software vem para facilitar a empresa atingir o objetivo.

Desses objetivos de negócio nascem os requisitos. Falando um pouco sobre gerenciamento de projeto tradicional, no processo de desenvolvimento temos:

* Concepção
* Análise & Design
* Desenvolvimento
* Testes
* Implantação

E aí nasce um software que vai ajudar a empresa a atender algum objetivo de negócio.

Quando uma empresa ou uma equipe desenvolve um software para um cliente, tem um estudo dizendo que quando desenvolvemos esse software apenas 20% são utilizadas pela empresa, pelo usuário. 7% sempre de alguma particularidade do software e frequentemente 13%, e 80% nunca é utilizada ou raramente ou às vezes. Ou seja, quando entregamos um software para uma empresa ou para um cliente interno apenas 20% daquele software é utilizado como um todo, logo temos muito desperdício, seja ele de tempo, de custo ou de colaboradores que fazem parte desse projeto e que o benefício é muito pouco se for pensar nos 20%.

(...) o Princípio de Pareto também se aplica ao desenvolvimento de software, onde 20% das funcionalidades costumam gerar 80% ou mais do benefício esperado.

#### Gestão de Projetos Tradicional x Ágil

Em projetos tradicionais (Waterfall), você corre o risco de descobrir que estava errado depois de meses. Com o SCRUM, você descobre que estava errado em no máximo 30 dias.

##### Tradicional (Waterfall):

Só permite que o projeto avance quando uma fase está inteiramente completa.

Diagrama

Descrição gerada automaticamente

Pensando nesse contexto todo, um gerenciamento de projetos tradicional (Waterfall) demanda muito tempo, leva muito meses, pois quando a galera está discutindo sobre requisitos e design nem todos sabem o que realmente quer e ai ficam em uma discussão sem fim geralmente, e quando essa discussão finaliza e chega na implementação com muitas incertezas e quando vem a homologação, vem toda aquela dor de cabeça de homologar, porque o cliente entende que o que foi feito não é o que ele queria, o cliente quer mudar, o mercado mudou, enfim, tem várias discussões e o projeto só na parte burocrática demora muito tempo.

* Escopo definido na fase Inicial do Projeto (Preditivo)
* Projeto é controlado por fases e marcos.
* Cliente só vê o software funcionando na fase final do Projeto
* Resistência a mudanças.

##### Ágil

Software construído por partes (incremental) e cada parte executa-se em um ciclo (iterativo)

Diagrama

Descrição gerada automaticamente

É construído em parte no máximo em um mês, e com o ciclo iterativo você acaba de construir uma parte entrega e vai para outra parte e assim por diante.

* Escopo definido ao longo do Projeto (Adaptativo).
* Projeto é controlado por funcionalidades entregues.
* Cliente pode ver parte do software funcionando na parte inicial do Projeto.
* Mudanças constantes de acordo com feedbacks contínuos.

#### Gestão de Projetos Ágil

##### O que é ser Ágil?

Ágil é diferente de rápido. O Ágil nesse sentido não quer dizer ser rápido no intuito de colocar vários desenvolvedores para ser mais rápida a entrega.

* Rapidez (mudança) e desembaraço;
  + Se eu tiver que mudar algumas coisas, eu vou mudar, vou adaptar o software no próximo ciclo da gestão do projeto ágil.
* Fazer coisas complexas de forma simples;
  + Priorizar algumas coisas para ficar mais simples a forma de desenvolvimento, então eu não vou colocar um negócio extremamente complexo para entregar em um mês, eu vou fatiar para que eu possa entregar de uma forma mais simples.
* Equipe comprometida com os objetivos;
  + Como eu entrego muito mais rápido e vejo basicamente o software funcionando em menos tempo, eu vou ter uma equipe comprometida porque se for identificado um problema logo, será resolvido logo.
* Maior valor para o cliente.
  + O meu foco é o cliente não é o projeto então eu vou fazer com que eu entregue mais rápido para o cliente usufruir mais rapidamente.

Ter capacidade de responder rapidamente as mudanças.

#### SCRUM

* SCRUM é um dos frameworks de gerenciamento de projetos ágeis.
* Projetos usando equipes pequenas e multidisciplinares produzem os melhores resultados.

##### Os Pilares do SCRUM

Os três pilares do SCRUM é a transparência, a adaptação e a inspeção, que quer dizer:

* Conversar mais e escrever menos;
  + Não quer dizer que eu não vou escrever nada, quer dizer que eu vou escrever o suficiente para que a gente possa conversar e ter uma comunicação com a equipe como um todo. (Transparência)
* Demonstrar o software constantemente aos usuários e obter feedbacks constantes;
  + Fazer com que o usuário veja rapidamente o software funcionando, ele dê feedbacks e a gente melhore o sistema nas próximas etapas, ou seja, o cliente está também totalmente abraçado contigo. (Transparência)
* Requisitos mudam ao longo do tempo;
  + Adaptar o software rapidamente para gerar valor para o cliente. (Adaptação)
* Aprender progressivamente com o uso do software.
  + Como a equipe está toda junta, temos a reunião diária, que é para ver o que cada um está fazendo para atender o objetivo daquela entrega. (Inspeção)

##### Razões para adotar o SCRUM:

* Desenvolvido e entregue em partes menores (2 a 4 semanas), com constante feedback dos usuários;
* Melhor gerenciamento de riscos; (Redução de incertezas);
  + Quanto mais eu foco em um escopo e no backlog de curta duração, eu reduzo a incerteza, porque eu sei e o cliente sabe do que ele está falando, porque quanto mais você descreve o teu backlog com curta duração mais você consegue listar os requisitos e menos você tem os riscos.
* Comprometimento, motivação e transparência da equipe. (Daily Meeting);
* Maior valor para o negócio; (Priorização do backlog)
* Usuários envolvidos durante todo o ciclo;
* Aplicação das Lições Aprendidas. (Melhoria contínua).

##### Características do time SCRUM:

* Equipes capazes de se auto-organizarem;
* As tarefas são do time e todos são responsáveis;
* Forte comprometimento com os resultados.

### Exercício Final

1. Qual característica de um projeto ágil?
2. ( ) Escopo definido na fase inicial do Projeto
3. ( **x** ) Escopo definido ao longo do Projeto
4. ( ) Resistente à mudanças
5. ( ) Aumento de pessoas para que o projeto seja rápido
6. ( ) Cliente paga mais caro para que o projeto seja mais rápido.
7. Quais são os pilares do SCRUM?
8. ( ) Desenvolvedores e Gerente
9. ( ) Processos e Mudanças
10. ( **x** ) Adaptação, Transparência e Inspeção
11. ( ) Comunicação e Status Report
12. ( ) Riscos, Escopo e Custos.

## Papéis e Responsabilidade de cada um do Time

### Product Owner (PO)

* Representante da área de Negócios;
* PO não é um Comitê;
  + Não é um grupo de pessoas.
* Define as funcionalidades do software (Product Backlog);
* Prioriza as funcionalidades de acordo com o valor do negócio;
* Garante que o time de desenvolvimento entenda os itens do Backlog no nível necessário.

### Scrum Master (SM)

* Garantir o uso correto do SCRUM;
* Scrum Master não é Gerente de Projetos;
* Age como facilitador;
* Auxilia o Product Owner no planejamento e estimativas do backlog;
* Auxilia a equipe a remover impedimentos;
* Treina o time em autogerenciamento e interdisciplinaridade.

### Time de Desenvolvimento (DEV)

* Possui habilidades suficientes para desenvolver, testar, criar e desenhar, ou seja, tudo que for necessário para entregar o software funcionando.

### Exercício Final

1. Quem é o responsável por priorizar o Backlog?
2. ( ) Scrum Master
3. ( ) Time DEV
4. ( ) Área de Negócios
5. ( **x** ) Product Owner
6. Quem é o responsável por garantir que o time utilize da melhor forma o SCRUM?
7. ( **x** ) Scrum Master
8. ( ) Time DEV
9. ( ) Área de Negócios
10. ( ) Product Owner
11. Quem é o responsável por desenhar, construir e entregar o software funcionando?
12. ( ) Scrum Master
13. ( **x** ) Time DEV
14. ( ) Área de Negócios
15. ( ) Product Owner

## Cerimônias do SCRUM

### Time Box

Tempo máximo para fazer uma cerimônia ou Sprint. Em muitos casos uma reunião que foi marcada para durar 1 hora, dura de 1 hora e meia a 2 horas. O Time Box serve para doutrinar o pessoal a utilizar um tempo máximo para fazer uma cerimônia.

### Sprint

Sprint quer dizer corrida ou arrancada. É o principal evento do SCRUM. Uma Sprint tem a duração de no máximo 30 dias corridos ou menos. O tempo da Sprint quem vai definir é o próprio time (SM, PO e DEV’s).

#### Composição de uma Sprint

##### Planejamento da Sprint

O PO, o Time de DEV e o SM participa desse planejamento. Nesse planejamento da Sprint, um time box de uma Sprint de 30 dias, essa reunião de planejamento leva 8h, nas primeiras 4h o objetivo e definir o que fazer, o PO vai explicar para o time o que ele deseja naquela Sprint, quais são as funcionalidades, o que precisa fazer, o que precisa ser entregue naquela Sprint. O Time de DEV vai tirar todas as dúvidas necessárias para entender o que o PO quer nessa Sprint. E nas outras 4h o objetivo é definir como fazer, normalmente quem faz é o time de desenvolvimento, onde o time de DEV quebra as atividades, entende tecnicamente o que precisa ser feito e o próprio time acaba estimando as atividades e como fazer, como fazer para entregar aquilo que o PO deseja.

Nesse planejamento o time pode utilizar várias ferramentas e técnicas, uma delas é a planning poker, que ela utiliza baralho mesmo, e cada um opina uma estimativa de cada tarefa, e o time entra em consenso para definir uma estimativa, um esforço de cada tarefa. Depois que o time estima todas as tarefas, ele vai verificar se tudo o que o PO queria realmente vai dar para entregar n a Sprint, então o time devolve para o PO o que ele realmente vai conseguir entregar dentro do prazo estipulado.

##### Reuniões Diárias (Daily Meeting)

Reuniões do time onde participa o time de DEV, PO e o SM. O Time de DEV responde três perguntas, o que fez no dia anterior, o que está programado no dia e se tem algum impedimento. Essa reunião é feita em pé, no mesmo horário que o time define e no mesmo local. Essa reunião é conhecida também como standup meeting. Essa reunião é realizada em pé, pois trata-se de uma reunião rápida, e existe um estudo que diz que as reuniões sentadas demoram mais pois as pessoas se acomodam. O ideal é que a reunião duro no máximo 15 minutos.

É muito utilizado também o quadro Kamban, onde o próprio time de desenvolvimento vai atualizando o status das atividades para, a fazer, fazendo e realizadas. Isso dá uma visibilidades e uma transparência, o time consegue fazer uma inspeção se tudo está ocorrendo dentro do previsto.

##### Revisão da Sprint (Review)

Geralmente no último dia da Sprint ocorre uma revisão onde o time DEV apresenta para o PO o trabalho feito. O time vai apresentar tudo que foi feito passo a passo e para quando colocado em produção não seja uma surpresa para o PO.

##### Retrospectiva da Sprint

Reunião da equipe para lições aprendidas. É uma reunião em que todos precisam ser muito transparentes e apontar os erros que a equipe teve seja no desenvolvimento, seja na review, seja no planejamento da Sprint, ou algo que não foi perguntado para o PO. Para que na próxima Sprint não ocorra mais nenhum problema.

## Gestão de Projetos Tradicional x Ágil

E na prática qual a diferença entre gestão de projetos tradicional e gestão de projetos ágil?

Imagine que o Nubank, se fosse na Gestão de Projetos Tradicional, no planejamento do projeto, imaginasse o tanto que ela queria oferecer para os clientes. Por exemplo, eu quero oferecer um cartão de crédito, então eu vou criar as funcionalidades de cartão de crédito, algum programa de benefícios, alguma conta difital para fazer a abertura de conta, para conseguir ver o extrato, fazer algo de empréstimo, ou seja, se ela fosse planejar tudo que ela imaginaria e talvez, nem todas essas funcionalidades ela iria colocar ou na verdade ela iria implantar um monte de mais coisas que talvez hoje não existiria mais. E na Gestão de Projetos Ágil, ela vai colocando o mínimo do produto viável, o produto com o mínimo de esforço, de desenvolvimento, de usabilidade, que aí ela vai melhorando o seu produto baseado no seu feedback. Então, o Nubank, por exemplo, lançou no início da sua história o cartão de crédito, foi evoluindo essa funcionalidade, ficando famosa, agradando o cliente, tendo feedback do cliente, ela foi evoluindo em outras funcionalidades. Então assim, essa é a diferença de Gestão de Projetos Tradicional x Gestão de Projetos Ágil.

# Fundamentos de um Projeto Ágil

## Papéis e Responsabilidades – Product Owner

### Quem é o Product Owner?

O Product Owner (PO) representa o profissional que tem a visão do que será desenvolvido, as necessidades a serem atendidas, o público que vai utilizar os serviços e os objetivos a serem alcançados.

Mais do que isso, ele é o profissional que visa o valor que será agregado para a empresa e para o cliente que vai utilizar o produto que será construído. O PO é quem define a ordem que as atividades serão desenvolvidas, é responsável também por validar se esses itens estão sendo entregues nas Sprints, se estão agregando o valor esperado, caso não esteja, ele tem que planejar outro ajuste necessário nas próximas Sprints a fim de manter o objetivo inicial proposto.

Na pior das hipóteses o PO també é responsável por cancelar um Sprint, quando as atividades planejadas não puderem mais ser entregues ou o PO entender que o valor esperado não vai ser mais atingido, nesse caso, cabe ao PO remanejar a Sprint mantendo os dias para a conclusão dela, com alguma entrega de valor possível. Se nenhuma entrega de valor executável for possível, o PO pode priorizar um estudo a fim de agilizar o desenvolvimento da próxima Sprint. Esse estudo é conhecido como Spyke.

* Representante da área de Negócios;
* PO não é um Comitê;
* Define as funcionalidades do Software (Product Backlog);
* Prioriza as funcionalidades de acordo com o valor do negócio;
* Garante que o time de desenvolvimento entenda os itens do Backlog no nível necessário.

Mas como dito anteriormente o principal papel do PO é trazer o máximo de valor possível para o produto. O PO deve garantir que todo esforço empregado no desenvolvimento daquele produto traga o maior retorno possível, o retorno poder financeiro, poder facilitar ao máximo a execução de alguma atividade, pode ser permitir que o cliente conclua um processo crítico de maneira mais simples possível.

O PO deve sempre inicialmente entender qual o objetivo daquele produto, o que ele quer alcançar, quem vai utilizar, como vai utilizar, em qual circunstância aquele produto será utilizado e só após entender tudo isso o PO deve validar antes de tudo se o produto faz sentido ou não.

Dentro da Sprint o PO tem participação obrigatória nas cerimônias de planning e de review, a participação do PO nas dailys não devem ser obrigatórias, porém, o PO deve se fazer presente tempo suficiente para que toda as dúvidas de negócio do time sejam sanadas em tempo para não causar nenhum atraso no desenvolvimento devido à falta de informação.

O PO nunca deve julgar qualquer dúvida, toda dúvida é válida, se houve alguma dúvida é porque não estava claro na estória escrita pelo PO, o PO dev e sempre fazer as dúvidas do time um aprendizado para melhorar cada vez mais as suas estórias, o PO não deve aguardar até a review para validar as entregas, na verdade é importante que o PO acompanhe o desenvolvimento e participe das dailys sempre que possível mesmo não sendo obrigatório, pois é importante o PO ter a visão se a Sprint está de pé ou está fadada ao fracasso para antever suas ações e até priorizar o backlog de maneira mais assertiva, afinal é fundamental o PO entender qual o desempenho do time a cada Sprint e assim aumentando a sinergia e melhorando a qualidade das entregas a cada ciclo.

### O que o PO faz na Planning e no Refining?

Primeiramente é importante dizer que uma planning eficiente começa com a execução de uma outra cerimônia, que não é uma cerimônia oficial do framework SCRUM, mas que ajuda e muito a aumentar a qualidade das plannings, essa cerimônia se chama Refining.

No Refining o PO apresenta previamente para o time as estórias que deverão ser apresentadas na planning, deverão porque até que a planning efetivamente aconteça qualquer mudança necessária é tolerada, o objetivo da refining é o time já saber previamente o que será desenvolvido e poder fazer questionamentos para o PO e para os demais envolvidos, nessa cerimônia é permitida a participação dos stakeholders, caso o PO ou qualquer outro envolvido não consiga responder a alguma dúvida do time, o PO terá tempo suficiente para correr atrás da resposta a tempo da planning o importante é que na planning ninguém saia com nenhuma dúvida e que todas as regras e negócios estejam muito claras.

A planning precisa ser dividida em duas etapas, primeira parto o PO irá pegar os itens mais refinados e priorizados do Product Backlog e irá apresentar para o time, neste momento é muito importante que o PO efetue a leitura completa de cada estória com o time para que mais uma vez nenhuma dúvida fique em aberto e para evitar os achismos por parte do time de desenvolvimento, o time não deve pressupor absolutamente nada, qualquer coisa que ele entenda de mais diferente, deve ser validado com o PO, assim que todas as estórias tenham sido apresentadas, lidas e aceitas pelo time, após a leitura que o time irá fazer a análise do que entra e o que não entra na Sprint, então só depois disso que a planning pode entrar na segunda etapa.

Na segunda etapa da planning é sugerido que o PO não participe, para não exercer uma influência sobre as atividades a serem desenvolvidas. O time irá pegar cada estória e irá escrever quais atividades deverão ser desenvolvidas para que a estória seja considerada entregue, por ser uma parte mais técnica é importante que o PO se abstenha para que o time possa colocar a suas próprias opiniões e até mesmo ter algumas discussões que normalmente não teriam na frente do PO, acredite isso é muito comum, principalmente em times pouco maduros, que ainda olham o PO como um cargo superior ao invés de um par. Durante a segunda etapa da planning o time pode fazer um outro corte e diminuir ainda mais as estórias a serem entregues, ao quebrar as estórias eles podem validar que ela é mais complexa do que imaginavam anteriormente, isso é aceitável, por isso a existência da quebra das atividades na planning e não após o início da Sprint.

Com as estórias mapeadas e as atividades descritas é definido o Sprint Backlog, com essa informação na mão, o PO deverá validar qual dos itens é considerado mais prioritário e que deve ser considerado o objetivo principal da Sprint. Isso significa que em teoria se essa atividade cair todo o resto da Sprint não vai fazer sentido e uma mudança deverá ser realizada no meio da Sprint para que o objetivo seja atingido ou no caso extremo a Sprint é cancelada já que o objetivo não ser alcançado.

### Release Planning

Liberação ou lançamento de software (em inglês: release) é o lançamento de nova versão oficial do produto de software. Cada vez que um produto de software é criado ou modificado o fabricante e seus desenvolvedores decidem sobre como distribuir ou o novo produto ou a modificação às pessoas que utilizam. O importante aqui é entender que não é obrigatório ter uma release no final de cada Sprint, você pode acumular uma, duas, três, dez Sprint’s para poder realizar uma release, porém, deve ser considerado que quanto mais Sprints você acumular antes de fazer uma release, maior a probabilidade de você enfrentar algum problema durante o merge (Merge é a junção dos códigos fontes de cada Sprint) e maior o tempo de teste necessário para garantir que todas as funcionalidades estejam funcionando corretamente após esse merge. Então não é um problema você acumular Sprint’s antes de fazer release, você só precisa tomar cuidado que quanto mais Sprint você acumular antes de uma release mais complexo essa release vai se tornar.

#### Tipos de Release

##### Release Planning de Múltiplas Squads

São vários times de desenvolvimento agrupado fazendo várias coisas distintas que podem ou não ter correlação com as atividades, mas que ao final da Sprint devem ser agrupadas em uma única release para serem implantadas em produção. Esse é um tipo de release técnica em que a complexidade consiste nas várias entregas distintas feitas por times distintos, mas que precisam obedecer às mesmas regras técnicas, para que possam subir juntas para produção sem que a entrega de uma squad impacte na entrega da outra squad. Está é uma parte técnica importante, mas que não tem ação direta do PO. O importante neste caso é o PO ficar atento aos prazos que o time deve cumprir da entrega do código e acompanhar a homologação após o merge para se certificar que suas entregas serão cumpridas ao final da release.

##### Release Planning de Projeto

Na release planning de projeto o PO possui uma demanda muito grande e que será necessário quebrar essa entrega em várias estórias e consequentemente quebrar em várias Sprints. Neste ponto é importante que o PO tenha certa senioridade, pois o PO deve compreender a real dimensão da demanda, deve compreender a real dimensão de cada estória e deve quebrar as estórias o máximo que puder. Por exemplo, o PO deve ter a consciência de que não é possível se desenvolver uma nova plataforma de autenticação do usuário em uma única Sprint e ele também deve ter discernimento para entender que uma estória com um título de novo autenticador não será o suficiente para o time compreender a dimensão e poder quebrar corretamente as atividades.

O PO também é responsável por gerenciar a expectativa dos stakeholders, então é importante o PO ter discernimento para já na reunião que ele recebe a demanda acalmar os ânimos dos stakeholders e informar que a demanda é grande e poderá levar várias sprints para ser entregue, nunca informar quantas Sprints, pois isso só será possível saber ao logo do projeto.

O PO deve quebrar as estórias e priorizá-las de acordo com a entrega de mais valor primeiro e se possível que já possa ir testando a solução com o cliente e ir captando retorno para ir adaptando a solução ao longo do desenvolvimento.

A release planning talvez seja um dos pontos mais críticos que o PO tenha que lidar. As empresas ainda estão acostumadas com os métodos tradicionais de gestão de projetos onde todo o planejamento é feito antes do início do desenvolvimento e neste ponto é que o PO enfrenta as maiores dificuldades já que ele nunca terá a visão completa do projeto, visto que esse irá se modificar ao longo do processo. O PO ainda é visto erroneamente como Gerente de Projetos do Scrum, porém, os papéis não são iguais e a parte mais parecida entre os dois é justamente o planejamento da release e isso acaba confundindo a maioria das pessoas.

O papel do PO no planejamento da release, consiste em definir as maiores entregas de valores para o cliente, priorizar as entregas de maior valor para serem desenvolvidas primeiro, organizar as Sprints para que seja possível captar o retorno do cliente mais breve possível, com o retorno dos clientes ir ajustando as estórias para que essas entreguem cada vez mais valor e organizando quando as releases serão realizadas para agregar cada vez mais valor.

Outro ponto importante no planejamento da release é quando a entrega em si trará um ganho real para o cliente ou se a entrega será apenas estética ou funcional, não alterando a dinâmica do usuário do sistema. Dependendo da entrega pode valer a penas esperar uma ou duas Sprints para entrar na release a fim de impactar menos possível o cliente, pois cada desenvolvimento de aplicativos, cada release lançada é necessária que o cliente efetue uma atualização e atualização em excesso pode acabar gerando impacto negativo para o cliente e ao invés de incentivá-lo você acaba afastando do seu produto.

### Exercício Final

1. Qual característica melhor descreve uma atividade do PO?
2. ( ) Recebe a demanda e passa para o time
3. ( ) Escreve a estória e valida com o demandante
4. ( **x** ) Entende a demanda e extrai o maior valor possível
5. ( ) Escreve a estória e deia o time avaliar os impactos
6. Qual o papel do PO dentro do TIME?
7. ( **x** ) Ajuda o time a entender a demanda e tirar dúvidas ao longo da Sprint
8. ( ) Coordena as pessoas do time e acompanha suas atividades
9. ( ) Remove os impedimentos que impactam o desenvolvimento da Sprint
10. ( ) Responsável por manter o time abastecido de café.

## Analisando Escopo e Definindo Prioridades

A definição do escopo do projeto é algo muito complexo pois normalmente temos muitas pessoas envolvidas de vários departamentos diferentes com objetivos diferentes. A definição do escopo é um dos pontos mais críticos do processo de gestão de projetos pois é neste momento que será definido o que será desenvolvido e é do artefato desta fase que serão extraídos os benefícios que serão atingidos. Portanto, se você fizer uma definição de escopo ruim você poderá atingir benefícios errados ou menos assertivos. Uma forma muito eficiente para você conseguir definir um escopo corretamente é justamente inverter a ordem começando a entender qual o objetivo ou valor que você quer atingir antes mesmo de definir como, assim você terá muito claro onde quer chegar e você terá muito mais possibilidades e caminhos a seguir para atingir o objetivo.

Por isso a importância de se inspencionar cada entrega, validar com os clientes se o produto está correto e dentro da expectativa e realizar as adaptações necessárias para atender cada vez mais os objetivos do cliente.

### Product BackLog

O Product Backlog é composto por artefatos chamados de épicos e sãos os incrementos necessários para que o produto seja considerado entregue. O Product BackLog deve conter apenas o conteúdo de um produto, cada produto tem o seu próprio backlog e normalmente esse backlog nunca se acaba, visto que conforme as Sprints vão sendo entregues, as inspeções vão sendo realizadas novos épicos vão sendo gerados para manter o produto sempre em evolução.

Como falamos o PO deve buscar o maior valor possível independente se for financeiro ou não, e o PO nunca deve negar um projeto logo de cara, ele deve sempre validar os possíveis ganhos, pois até mesmo os demandantes não enxergam todos os benefícios, uma vez que eles podem estar focados apenas em um tipo de ganho.

Um exemplo prático e simples de um tipo de validação que o PO deve fazer é, vamos supor que o desenvolvimento de uma Sprint custe em torno de 50mil reais para a empresa, chega para o PO uma demanda de desenvolvimento de um produto muito simples e ao analisar o PO verifica baseado nas suas experiências que deve ser possível realizar a construção em apenas uma Sprint, a certeza só vamos ter na planning quando o time de desenvolvimento irá verificar e estimar o tempo, mas quando o PO verifica qual o retorno esperado para o produto ele vê que o ganho máximo esperado do produto é de 35mil reais, pode parecer estranho, mas não é incomum este tipo de análise dentro das empresas, uma vez que os desenvolvedores são considerados um custo fixo dentro das empresas. Como PO é superimportante que este volte para os demandantes, os stakeholders, e informe que o custo de desenvolvimento será maior que o retorno financeiro esperado a fim de validar se existe mais algum outro retorno que não foi levantado anteriormente e que pode fazer com que o projeto continue fazendo sentido. Por exemplo, neste momento o PO pode descobrir que mais importante que retorno financeiro será o ganho de usabilidade, pois o sistema ficará mais fácil para o cliente utilizar e com isso ele terá menos dúvidas e acabará ligando menos no call center.

Outra atuação importante do PO é de encontrar sinergia entre outros projetos existentes no backlog, por exemplo, pode ser que tenho outro projeto com prioridade muito baixa mas que tenha sinergia com outro projeto que tem uma prioridade super elevada e quando o PO combina os dois projetos os ganhos se somam e a empresa acaba tendo um ganho maior que o esperado. Essa é a importância do PO não só conhecer mas também dominar todo o backlog.

Quando um épico é priorizado este deve ser detalhado e a partir dai um épico pode gerar N estórias.

#### Épicos

Os épicos não possuem muito detalhamento, eles servem para você definir as atividades de maneira marcos sem necessariamente investir muito tempo em algo que não será desenvolvido agora ou as vezes nunca será desenvolvido.

Incremento sem muito detalhamento, ajuda a te direcionar aos caminhos que deve seguir.

#### Estória

A estória deve ser escrita com o máximo de detalhes possível, normalmente eu defino uma estória duas Sprint antes de entrar em desenvolvimento, assim eu tenho tempo suficiente para detalhar as estórias com as regras de negócio mais atualizadas e também não corro o risco de não ter a estória finalizada na planning devido ao atraso e levantamento de alguma regra, acredite isso é muito mais comum do que se imagina.

Detalhamento dos épicos, um épico normalmente se divide em várias estórias, onde ficam descritos o que deve acontecer e suas regras de negócio.

#### Tasks

As tasks ou atividades são definidas e escritas pelo time de desenvolvimento e pode haver N tasks para cada estória.

### Escrevendo uma Estória

Particularmente você pode escrever uma estória utilizando o modelo abaixo:

* Nome da estória
* Descrição da estória (Eu, Como, Quero, Quando)
* Regras de Negócio (Separar Regras de Front-End de Regras de Back-End)
* Tela (Link ou Imagem das telas a serem desenvolvidas)
* KPI (Quais objetivos/valores a estória precisam atingir)
* Tagueamento (Como a estória será “Tagueada” para poder mensurar os KPI)
* Critérios de Aceite (Qual o passo a passo de todos os caminhos felizes possíveis a estória deve cumprir para que ela seja considerada aceita)

### Riscos Positivos

Gestão de riscos em projetos ágeis é tão importante quanto em qualquer outro tipo de projeto, normalmente nós olhamos apenas para os risco que podem impactar negativamente os nossos projetos.

#### Positivos

A gestão de riscos positivos é tão importante quanto visto que eles também podem afetar o prazo, custo e escopo do projeto mas de maneira positiva e as vezes inviabilizando o projeto de maneira positiva, ou seja, trazendo o resultado esperado sem necessariamente desprender do desenvolvimento planejado.

Todas as grandes empresas vivem situações onde existem projetos similares ou até mesmo conflitantes mas que são desenvolvidos mesmo assim, simplesmente porque uma área não tem conhecimento sobre o que a outra está fazendo ou devido a uma falha de gestão de projetos.

O risco positivo é muito comum por incrível que pareça, por exemplo, um projeto está em backlog a muito tempo e foi priorizado para iniciar na próxima Sprint, porém, em dois meses entrará em vigor uma nova legislação que afetará o sistema e esse deverá ser refeito, se não houver uma interação entre o PO e os Stakeholders após a priorização inicial esse tipo de risco pode não ser mapeado e algo que era ruim pois ficou muito tempo no backlog acaba se tornando algo positivo, pois a empresa terá a oportunidade de se adequar a nova legislação com mais eficiência.

Outro risco positivo muito comum é quando existe vários times de desenvolvimento trabalhando várias frentes diferentes, é comum o que está sendo feito em um time ser reaproveitado por outro time, porém, se não houver um acompanhamento sobre o que os outros times estão desenvolvendo o PO pode perder uma oportunidade para entregar valor sem que haja um desenvolvimento do próprio time, exemplo, um outro time está desenvolvendo um fluxo para que o cliente consiga solicitar o cancelamento do seu plano de telefone diretamente pelo site ou app sem que o cliente precise ligar no callcenter, e você está desenvolvendo um produto de auto atendimento em que uma das saídas é permitir que o cliente solicite o cancelamento do seu produto de telefone, se você não souber que o outro time está desenvolvendo uma solução para tratar esse caso você pode acabar refazendo uma solução que já está sendo desenvolvida ou pior você pode enviar o cliente para uma solução pior ainda, mais demorada e com uma usabilidade muito ruim.

Muito ignorado nos projetos, porém, um dos fatores de maiores ganhos no desenvolvimento de sistemas.

#### Negativos

Itens que podem afetar o prazo, custo ou escopo de um projeto de maneira que pode acabar inviabilizando-o

### Exercício Final

1. Qual método é mais indicado para definir o Escopo do Projeto?
2. ( ) Ler a solicitação junto com o Time de Desenvolvimento
3. ( ) Pedir para o Demandante escrever o que ele quer
4. ( **x** ) Entender qual o ganho antes de definir o escopo
5. (  ) Escrever o escopo baseado no seu conhecimento do produto.
6. O que compõe um Product Backlog?
7. ( ) Estória e Escopo
8. ( ) Escopo e Atividade
9. ( ) Épico e Atividade
10. ( **x** ) Épico e Estória
11. Qual a importância de se mapear Riscos Positivos?
12. ( ) Para o time poder descansar
13. ( **x** ) Poder priorizar outros itens no backlog
14. ( ) Cobrar a entrega dos outros times
15. ( ) Conhecer a legislação vigente

## Papel do PO na Transformação Digital

A Transformação Digital é um processo no qual as empresas fazem uso da tecnologia para melhorar o desempenho, aumentar o alcance e garantir resultados melhores. É uma mudança estrutural nas organizações dando um papel essencial para a tecnologia.

Não é segredo para ninguém que a transformação digital nos objetivos das maiores empresas do país, porém, nem todas ainda estão investindo nesta área simplesmente por não entenderem o que é de fato a transformação digital.

Existem diversos conceitos e teorias sobre o que é a transformação digital que podem ser facilmente encontrados na internet, porém, para quem já viveu a transformação digital e a viu crescer e cair eu posso lhe afirmar uma coisa, transformação digital é muito mais sobre pessoas do que tecnologia, a transformação digital é mostrar para as pessoas como usar a tecnologia que já está disponível a algum tempo da melhor maneira possível extraindo o melhor da tecnologia para otimizar o nosso trabalho no dia a dia, é sobre como podemos diminuir processos, diminuir atritos e principalmente potencializar resultados.

A transformação digital é muito mais uma mudança de mindset, alterar a forma como as pessoas agem e pensam da tecnologia, é sobre entender que a evolução é inevitável e sobre o que podemos fazer para nos beneficiar dela.

O papel do PO nesse contexto eu entendo como fundamental, pois o papel do PO representa a mudança na forma como os projetos são tratados, a maneira como o conhecimento é disseminado, se antes você precisava de uma analista de projetos dentro de cada departamento para que este acompanhasse e cobrasse o gerente de projetos e o departamento era o único dono do conhecimento do que ali era gerado, com a mudança de mindset teremos cada vez menos pessoas fazendo mais coisas e para isso precisaremos de tecnologia nos suportando para aguentar uma maior carga de trabalho o PO não é o novo GP e o GP não foi quebrado entre GP e SM, na verdade o papel do GP está sendo dividido em pequenas partes especialistas dentro dos departamentos, e essas partes especialistas precisam interagir entre si em prol de entregar um beneficio maior para a empresa ou para o cliente. O PO será apenas o orquestrador desta comunicação, o PO é responsável por juntar todas as penas partes e montar o planejamento do produto, porém, este planejamento poderá ser alterado a cada interação ou Sprint e essa é a complexidade que teremos que conviver de agora em diante. Por isso acredito que o PO se tornará cada vez mais um papel estratégico dentro das empresas, pois ele será o responsável direto pelo sucesso ou não do produto.

Lembrando que o mindset ágil também prega que o PO deve ter autonomia para o desempenho de suas funções o que ainda é um entrave nas grandes organizações ocasionando o surgimento de falsos otimizagens. Muitas empresas passaram a organizar os seus times em squads, implementaram o Kamban como metodologia de acompanhamento de tasks, criaram o falso backlog e assim dizem para o mercado que implementaram com sucesso a transformação digital, porém, continuam pensando e agindo como antes, continuam planejando 100% do projeto antes de testar com o cliente e continuam com a briga interna de poder.

Ser ágil não é usar postit, ser ágil é agregar valor para o cliente o mais rápido possível, ser ágil é errar rápido e corrigir rápido, ser ágil é estar um passo a frente e principalmente, projetos ágeis são muito mais planejados do que um projeto tradicional. Transformação digital é trabalhar junto com o cliente é ser centralizado no cliente e ter uma mente focada no objetivo de atender o cliente da melhor forma possível, por isso o PO tem um papel fundamental na transformação digital, o PO é quem entende a necessidade do cliente e transforma em produtos melhores.

# Conceitos e Atividades Essenciais para o Sucesso de um Projeto Ágil

## Aprenda sobre os conceitos e planejamento de tarefas

### Objetivos da Aula

* Estória vs Tarefas
* Critérios de Aceite, Estimativa e Planejamento de Tarefas
* Relacionamento com Clientes / Stakeholders

### Estória vs Tarefas

Primeiro é importante entender a diferença entre estória, épico e tarefa. Imagine que você quer construir uma bicicleta, a bicicleta em si é o objetivo do projeto, então quanto todos os épicos estiverem sido concluídos o projeto estará pronto, logo o projeto bicicleta é um conjunto dos épicos, quadro, garfo, selim, rodas, sistema de freio e sistema de tração, repare que não foi mencionado o guidão e nem os pedais, o épico é o objetivo macro que deve ser atingido e para ser atingido esse objetivo macro deve ser quebrado em objetivos menores, logo o épico garfo consiste nas estórias guidão e amortecedores, e o garfo que também é o nome da peça central. O guidão é uma estória do épico garfo assim como o pedal é uma estória do épico sistema de tração, e a manopla de freio é uma estória do sistema de freios. Porém, essas estórias por si só ainda não dão a dimensão do trabalho que deve ser feito para que o trabalho seja concluído aí é que entra as tarefas, para a realização da estória garfo, eu preciso realizar a tarefa montar o guidão, montar o garfo, montar o garfo no quadro, montar a roda no garfo e montar o sistema de freio na roda. Neste nosso exemplo estou considerando que as peças já estão prontas caso não estivessem eu poderia criar épicos relacionados a construção usinagem das peças e outros épicos relacionados a montagem. Verifique que o nosso exemplo tem tarefas dependentes de outras estórias, este é um dos grandes motivos de realizarmos este tipo de atividade antes de iniciar a construção, para podermos mapear todas as dependências internas e externas e assim conseguirmos priorizar estórias e atividades de maneira mais assertiva.

Ou seja, basicamente uma estória é um conjunto de tarefas e um épico é um conjunto de estórias. Nós começamos pensando nos épicos do projeto quando esse projeto foi priorizado nós quebramos as estórias e neste momento se acharmos necessário podemos quebrar os épicos em mais épicos, quando as estórias forem priorizadas nas Sprints, essas estórias são quebradas em tarefas e neste momento também podemos quebrar uma estória em mais estória se for entendido pelo time que é o melhor caminho. E assim ficamos com alguns épicos com algumas estórias vinculadas e algumas estórias com algumas tarefas vinculadas.

#### Exemplo de uma Estória

A estória deve ser descrita em nível de negócio e serve para que o time de desenvolvimento entenda o objetivo daquela atividade e para isso é importante que o time entenda não só o que deve ser feito, mas também para que será usado e como será usado, assim o time poderá desenvolver com mais assertividade aumentando a qualidade da entrega, consequentemente do produto e do retorno.

* Nome da estória
* Descrição da estória (Eu, Como, Quero, Quando)
* Regras de Negócio (Separar Regras de Front-End de Regras de Back-End)
* Tela (Link ou Imagem das telas a serem desenvolvidas)
* KPI (Quais objetivos/valores a estória precisam atingir)
* Tagueamento (Como a estória será “Tagueada” para poder mensurar os KPI)
* Critérios de Aceite (Qual o passo a passo de todos os caminhos felizes possíveis a estória deve cumprir para que ela seja considerada aceita)

#### Exercício Final

1. Qual característica melhor descreve uma estória?
2. ( ) Uma tarefa descrita em nível executivo.
3. ( ) Uma tarefa descrita em nível de desenvolvimento.
4. ( **x** ) Uma tarefa descrita em nível de negócio.
5. ( ) Uma tarefa descrita em nível de entrega.
6. O que é uma tarefa?
7. ( ) Um conjunto de estórias que o time de desenvolvimento deve desempenhar para entregar para o projeto.
8. ( **x** ) Um conjunto de atividades que o time de desenvolvimento deve desempenhar para entregar a estória
9. ( ) Um conjunto de épicos que o time de desenvolvimento deve desempenhar para entregar a estória
10. ( ) Um conjunto de atividades que o PO deve desempenhar para entregar para o time de desenvolvimento.

### Critérios de Aceite, Estimativa e Planejamento de Tarefas

#### O que é um Critério de Aceite?

É uma lista de critérios que precisam ser alcançados para que a User Story atenda os requisitos do usuário e seja aceita pelo Product Owner.

Os critérios de aceitação têm o objetivo de: definir limites para as user stories. Ajudar o PO a detalhar em alto nível o que é necessário para entregar valor ao cliente.

O PO precisa saber qual objetivo a demanda precisa cumprir para que ele considere aquela estória como aceita e concluída, ou seja, é o objetivo que aquela estória tem que concluir. Se uma estória não tem um critério de aceite muito provavelmente o PO não entendeu a demanda e apenas escreveu a estória para cumprir tabela.

Como PO a primeira coisa que eu tenho que descobrir de uma demanda é justamente qual o objetivo dela, onde se quer chegar, qual objetivo queremos atingir, pois uma vez com esse entendimento claro é muito mais fácil definir um caminho para chegar no objetivo. Por incrível que pareça muitas demandas não são muito claras para os demandantes nem eles entendem as vezes os objetivos que querem atingir, eles apenas querem resolver o problema, mas nem se quer sabem se é de fato um problema.

Demandantes também costumam ocultar os objetivos para que consigam fazer com que o projeto seja priorizado, pois quando o objetivo real é descoberto o projeto pode não fazer sentido, porém, aquele projeto era importante politicamente para o demandante.

#### Estimativa e Planejamento

O Planning Poker é em disparado o método mais usado para se fazer estimativas do tamanho das tarefas, porém, essa não é única ferramenta disponível. O planning poker é de fato um jogo de cartas, onde cada membro do time de desenvolvimento recebe um deck contendo todos os números da sequencia de fibonaci, quando todos estão prontos as estórias são lidas e as tarefas são criadas, após a criação de cada tarefa o time de desenvolvimento inteiro vota jogando uma das cartas falando qual o tamanho daquela atividade, deve se levar em conta a complexidade de se realizar aquela tarefa, o trabalho manual em si e o tempo, existem tarefas que a complexidade de se fazer é muito baixa, porém, exige muito trabalho manual por parte do desenvolvedor e isso acaba aumentando a complexidade e consequentemente o tempo de desenvolvimento e são essas as questões que os membros do time de desenvolvimento usam para estimar as tarefas.

Uma tarefa considerada muito simples a se fazer pode receber voto um, uma tarefa muito difícil pode receber treze, normalmente quando uma tarefa recebe vinte pontos os times entendem que a tarefa ou a estória está grande demais e deve se quabra-la em duas ou mais estórias ou tarefas. Os membros do time que derem a maior e a menor nota devem justificar o motivo de sua pontuação pois eles podem ter pensado em alguma coisa que os outros membros do time não consideraram e esse é o maior ganho que temos nesse framework. O time acaba ficando muito próximo e um acaba ajudando o outro, mesmo que sejam de área diferentes como front-end e back-end, android, ios, web, mobile, etc.

O Planning Tshirt Size também um é um método usado para se fazer estimativas do tamanho das tarefas, porém, é usado o tamanho das camisetas, P, M, G ou GG. Alguns times têm dificuldades de utilizar o Fibonaci por achar muito complicado, porém, como o tamanho de camisa fica muito mais simples para eles, devendo sempre levar em consideração os fatores de complexidade vs esforço vs tempo, e deve-se manter o processo de quem deu o menor e o maior tamanho justificar a sua pontuação.

#### Planejamento

Conforme dito anteriormente a planning precisa ser dividida em duas etapas, na primeira parte o PO irá pegar os itens mais refinados e priorizados do product backlog e irá apresentar para o time. Conforme visto a pouco, o PO junto com o time de DEV lê as estórias e tira as dúvidas, na segunda parte da planning o time quebra as tarefas e as estima. Na segunda parte da planning é sugerido que o PO não participe para não exercer uma influência sobre as atividades a ser desenvolvidas.

Na segunda parte o time irá pegar cada estória e irá escrever quais atividades deverão ser desenvolvidas para que a estória seja considerada entregue. Por ser uma parte mais técnica é importante que o PO se abstenha para que o time possa colocar a suas próprias opiniões e até mesmo ter algumas discussões que normalmente não teriam na frente do PO. Durante a segunda etapa da planning o time pode fazer um outro corte e diminuir ainda mais as estórias a serem entregues visto que ao quebrar as estórias eles podem validar que ela é mais complexa do que se imaginava anteriormente e isso é aceitável, por isso a existência da quebra das atividades na planning e não após o início da Sprint.

Com as estórias mapeadas e as atividades descritas é definido o Sprint Backlog, com essa informação na mão o PO deverá validar qual dos itens é considerável mais prioritário e que deve ser considerável o objetivo principal da Sprint, isso significa que em teoria se essa atividade cair, todo o resto da Sprint não vai fazer sentido e que alguma mudança deverá ser realizada no meio da Sprint para que o objetivo seja atingido, ou em casos extremos, a Sprint é cancelada já que o objetivo da Sprint não será mais alcançado.

#### Exercício Final

1. O que é um Critério de Aceite?
2. ( ) É uma lista de demandas que o PO deve atingir na entrega das tarefas
3. ( ) É uma lista de critérios para que o épico atenda aos requisitos do cliente.
4. ( **x** ) É uma lista de critérios para que a estória atenda aos requisitos do cliente.
5. ( ) É uma lista de cenários de teste.
6. O que é Planning Poker?
7. ( ) Uma atividade de integração do Time
8. ( ) Uma atividade de mensuração da capacidade do time
9. ( ) Uma atividade de mensuração da complexidade do Épico descrito pelo cliente
10. ( **x** ) Uma atividade de mensuração do esforço e complexidade das tarefas ou estórias.

### Relacionamento com Clientes/Stakeholders

O stakeholders é uma pessoa ou um 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 das suas políticas e forma de atuação.

Alguns exemplos de stakeholder de uma empresa poder ser os seus funcionários, gestores, gerentes, proprietários, fornecedores, concorrentes, ONGs, clientes, o Estado, credores, sindicatos e diversas outras pessoas ou empresas que estejam relacionadas com uma determinada ação ou projeto.

Cliente interno ou stakeholder é a pessoa ou grupo de pessoas interessados no projeto em questão, o stakeholder não necessariamente é o patrocinador do projeto, o patrocinador pode ser o diretor de um departamento, mas o stakeholder pode ser um analista ou um gerente deste mesmo departamento e o PO precisa saber conversar com todos esses públicos, daí a necessidade de o PO ter um excelente jogo de cintura na hora de tocar os seus projetos.

Muitas pessoas abrem projetos que não necessariamente trazem um grande beneficio para a empresa mas que trazem um grande ganho político para o stakeholder, como por exemplo, substituir uma planilha Excel chata por um sistema, acontece que em alguns casos o custo de produção de um sistema para a substituição da planilha Excel não se paga, ou seja, o retorno é muito pouco em detrimento do custo de implementação e por isso é muito comum um stakeholder se aproveitar de uma outra demanda de maior relevância para poder encaixar demandas de baixíssima relevância. Esse é um dos motivos que o PO deve sempre entender muito bem a demanda e os objetivos que devem ser alcançados para que não acabe focando esforço onde não é interessante para a empresa.

Alguns casos podem não ser viáveis economicamente para a empresa, porém, trazem outros benefícios, como melhora na percepção de imagem da companhia, uma empresa com percepção de alta tecnologia, uma empresa com percepção de ser um ótimo lugar para se trabalhar, no final todos esses itens até podem ser mensurados, mas de outras perspectivas o que torna muito complexo.

O relacionamento do PO com este publico deve ser feito sempre de maneiro mais amistosa possível, lidar com pessoas é uma arte e apenas poucas pessoas têm o dom de fazê-lo naturalmente.

A principal ferramenta do PO é o relacionamento, pois se você conseguir transitar entre várias áreas e vários departamentos você terá muito mais facilidade em orquestrar as soluções e ajudar o SM a remover os impedimentos.

## Rotinas de um time ágil

### Objetivos da Aula

* Daily e Retrospectiva
* Refinamento e Review
* Maturidade da Equipe

### Daily e Retrospectiva

A Daily deve ter um tempo total de 15 minutos, ocorrer sempre no mesmo horário e no mesmo lugar com a presença de todo o time de desenvolvimento a presença do SM e do PO não é obrigatória pois a Daily é para o time de desenvolvimento saber o que cada um está fazendo e se alguém estiver com algum problema o outro membro pode ajudar de alguma forma. É a cerimônia do SCRUM que mais agrega valor ao produto.

Na Daily normalmente é cobrado as três perguntas, o que eu fiz ontem, o que farei hoje, se possuo algum impedimento, mas a Daily não se limita somente a isso, desde que respeitado o prazo de 15 minutos se algum membro do time achar que deve falar alguma outra coisa ele pode falar, o SM irá verificar se é um assunto para se encerrar na própria Daily ou se será necessário marcar um novo horário para discutir somente aquele ponto. Porém, é permitido falar qualquer assunto na Daily desde que agregue valor ao projeto.

Ao contrário da Daily, como o próprio nome já diz é executado diariamente dentro da Sprint, a retrospectiva acontece apenas uma vez na Sprint, sempre após a conclusão da Sprint, nessa cerimônia a presença do PO é muito importante, porém, não é obrigatória. Em times menos maduros é até importante separar a retro em duas partes uma com e outra sem o PO.

Na retrospectiva é discutido principalmente os pontos que foram bons e ruins na Sprint, o que se deve melhorar e o que não se deve fazer e o que podemos fazer já na próxima Sprint.

Verifique que a essência da Daily e da Retrospectiva são as mesmas a diferença é que a Daily é focada nas tarefas do dia a dia e a Retro é foca na Sprint como um todo. Essas duas cerimônias quando bem executadas tem um poder transformador gigante dentro do time e um time maduro que entende a importância dessas cerimônias fazem elas acontecerem sozinhas sem que o SM ou o PO precise ficar chamando para que elas aconteçam.

#### Exercício Final

1. Quem tem participação obrigatória na Daily?
2. ( ) O Time de DEV e o SM
3. ( ) O Time de DEV e o PO
4. ( **x** ) O Time de DEV apenas
5. ( ) O Time de DEV, o SM e o PO
6. Quem tem participação obrigatória na Retro?
7. ( **x** ) O Time de DEV e o SM
8. ( ) O Time de DEV e o PO
9. ( ) O Time de DEV apenas
10. (  ) O Time de DEV, o SM e o PO

### Refinamento e Review

#### Refinamento

O refinamento é uma cerimônia não oficial do SCRUM, porém, muito utilizada para que o time junto com o PO discuta como será a próxima Sprint, quais serão os entregáveis e adiantar possíveis dúvidas que normalmente apareceriam apenas na planning.

O refinamento tem a finalidade de aumentar o entendimento da demanda por parte do time e melhorar a qualidade da entrega e aumentar o retorno esperado para o projeto. Neste momento normalmente o time de desenvolvimento começa a fazer várias perguntas para o PO a fim de entender do quem se trata a demanda, por isso é de suma importância que a estória já esteja muito bem escrita neste momento para que fique menos dúvidas possíveis ou que não se tenha dúvidas.

A ideia é facilitar a planning e torná-la muito mais efetiva e assertiva, nesta cerimônia a presença de todo o time SCRUM é obrigatória para que todos possam ouvir e entender a demanda e possam ajudar no desenho da melhor solução.

#### Review

Na review quem apresenta é o time de desenvolvimento, neste momento eles tem a oportunidade de mostrar o trabalho que cada um desenvolveu na Sprint anterior e podem tirar qualquer dúvida técnica que o público presente tenha em relação a solução. Dúvidas de negócios continuam sendo tiradas pelo PO que deve estar presente na cerimônia assim como o SM, nesta cerimônia são convidados todos os interessados na entrega, sejam eles os stakeholders, os demandantes diretos, os líderes técnicos e outros gestores. A ideia principal dessa cerimônia é de fato ver o que está sendo entregue, validar se está de acordo com o que foi solicitado inicialmente, validar se houve alguma mudança no meio do fluxo, bater com o que foi definido no refinamento e na planning e por fim e mais importante se agrega valor ao negócio.

#### Exercício Final

1. Em que momento acontece o Refinamento?
2. ( ) O refinamento acontece um passo antes da Daily
3. ( ) O refinamento acontece um passo depois da Retro
4. ( **x** ) O refinamento acontece um passo antes da Planning
5. ( ) O refinamento acontece um passo depois da Review
6. Qual o principal objetivo da Review?
7. ( ) Verificar se o time entregou o trabalho direito
8. ( ) Verificar se o PO entendeu a demanda
9. ( ) Verificar se a demanda foi priorizada na Sprint
10. ( **x** ) Verificar se a demanda agrega valor ao negócio.

### Maturidade da Equipe

Um time nunca vai ser maduro se a empresa não for madura, não adianta o time ser ágil e a empresa ser Waterfall, não adianta você desenvolver em Sprint mas ter que passar por um processo de gestão de mudança. O amadurecimento do time está diretamente ligado ao ambiente em que ele está situado. Por isso medimos a maturidade do time utilizando os pilares do SCRUM.

Um time que possui transparência sobre o que acontece ao seu redor, possui transparência sobre as demandas que estão por vir e possui transparência para saber se a sua empresa está indo bem ou não, tem muito mais capacidade e tranquilidade para poder desenvolver suas atividades, assim como um time que pode retornar para as suas entregas anteriores, verificar o resultado alcançado, cruzar com os resultados esperados, consegue entender melhor o comportamento do seu cliente ou usuário final e assim pode se adaptar para novas entregas e faze-las cada vez mais de maneira mais assertiva.

Um time maduro é um time que de fato consegue vivenciar os pilares do SCRUM, e consequentemente um time que não fica dependendo do SM para resolver os seus problemas, é um time que já conhece o caminho das pedras e tem iniciativa para resolver os seus problemas.

Um time maduro não só questiona o PO sobre suas estórias como também auxilia o PO a escrever as estórias de maneira mais clara e ajuda o PO a quebrar mais ou menos as estórias de forma a encaixá-la perfeitamente uma Sprint.

Um time maduro não é o PO nem a SM que puxa o time de DEV, é o time de DEV que puxa o PO e o SM acaba não sendo mais necessário no dia a dia e se torna um consultor pontual.

Basicamente um time maduro é o time que anda sozinho.

## Certifique seu Conhecimento