**Parte 1 - Introdução a Gestão de Projetos e ao SCRUM**

Como toda empresa, seja ela pequena ou grande, cada empresa tem um objetivo, um objetivo de negócio. Para entender esse objetivo o software vem para facilitar a empresa no atingimento desses objetivos. Desses objetivos de negócios nascem os **Requisitos**.

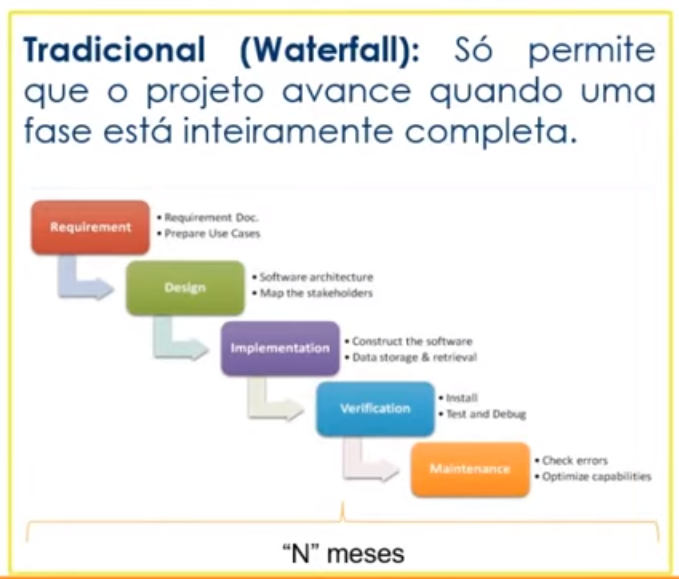
O gerenciamento de projeto tradicional, dentro do **Processo de Desenvolvimento**, temos:

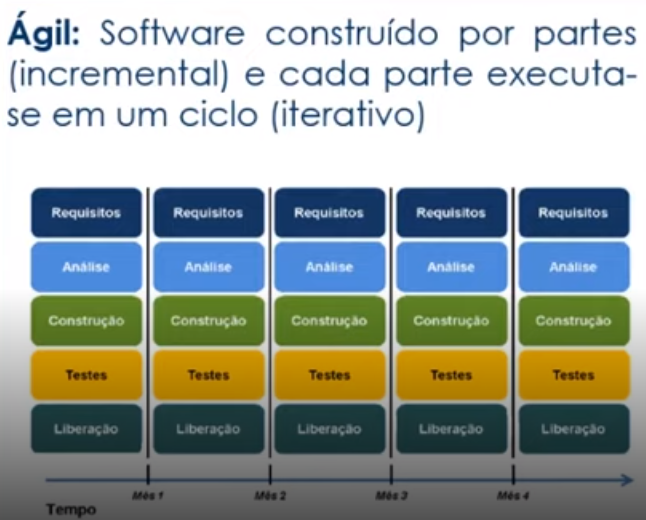
* Concepção;
* Análise & Design;
* Desenvolvimento;
* Testes;
* Implantação.

Aí nasce o software que vai ajudar a empresa a atender um objetivo de negócio.

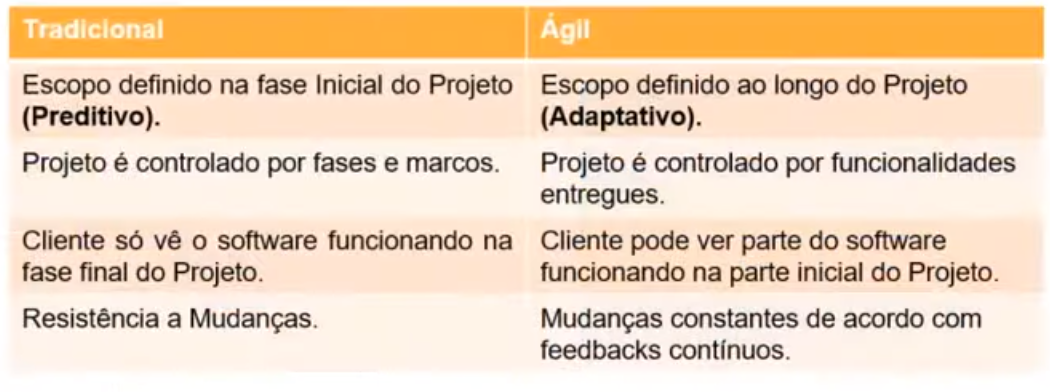
Na realidade, quando é desenvolvido um software para um cliente, tem um estudo dizendo que apenas 20% é utilizado pela empresa/usuário. Ou seja, quando um software é entregue pro cliente, apenas 20% do software é utilizado como um todo, tendo um desperdício de tempo, de custo, de colaboradores e o benefício é muito pouco.

No modelo tradicional (Waterfall/Cascata) só é permitido que o projeto avance quando uma fase está inteiramente completa (acaba um e começa o outro). Pensando nesse contexto, demanda muito tempo por durar muitos e muitos meses, além de uma entrega com muitas falhas.



Já na gestão de projetos ágil, o software é construído por partes (incremental) e é construído em partes que duram no máximo em um mês e tem um ciclo interativo.

**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.

Ágil é diferente de ser rápido, mas está no sentido de:

* Rapidez (mudança) e desembaraço;
* Fazer coisas complexas de forma simples;
* Equipe comprometida com os objetivos;
* **Maior valor para o cliente.**

Em resumo: Ter capacidade de responder rapidamente às mudanças.

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

Os pilares do SCRUM são **transparência**, **adaptação** e **inspeção**. O que isso significa:

* Conversar mais e escrever menos;
* Demonstrar o software constantemente aos usuários e obter feedbacks constantes;
* Requisitos mudam ao longo do tempo; (Adaptação)
* Aprender progressivamente com o uso do software. (Inspeção)

Na inspeção tem a reunião diária para ver o que cada um tá fazendo para cumprir o objetivo.

**Razões para adotar o SCRUM**

* Desenvolvido e entregue em parte menores (2 a 4 semanas), com constantes feedbacks dos usuários;
* Melhor gerenciamento de riscos; (Redução de incertezas)
* 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.

As Startups costumam utilizar o framework ágil, pois costumam trabalhar muito com **MVP (Minimum Viable Product)**. MVP é o mínimo viável do produto que pode ser entregue para o cliente para usar o software. Com os feedbacks que são recebidos, vai evoluindo o software.

**Parte 2 - Papéis e Responsabilidades do Time Scrum**

**Product Owner (PO):**

* Representante da área de Negócios;
* PO não é um Comitê, ou seja, 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.

O PO que explica pro time de desenvolvimento o que é preciso fazer para aquela funcionalidade e ele que representa a Área de Negócios.

**Scrum Master (SM):**

* Garante o uso correto do SCRUM;
* Scrum Master não é Gerente de Projetos;
* Age como facilitador, faz com o que o PO consiga explicar para o time;
* 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. O time de desenvolvimento precisa ser multidisciplinar, ou seja, as tarefas são divididas entre o time, de acordo com as habilidades de cada membro.

**Parte 3 - Cerimônias do SCRUM**

**Time Box:** Tempo máximo definido para fazer uma cerimônia ou fazer todo o desenvolvimento em uma Sprint (duração de 15min).

**Sprint:** Corrida, arrancada. É o principal evento do SCRUM.

* Tem duração máxima de 30 dias corridos (ou menos). O time que determina esse tempo.

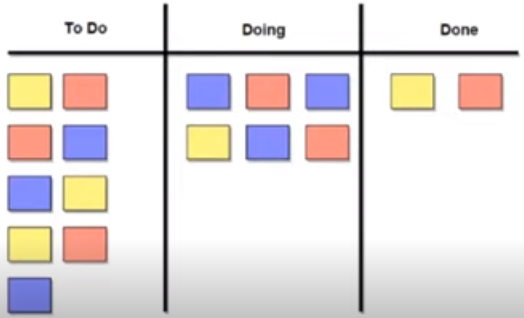
**Composição de uma Sprint:**

* Planejamento da Sprint;
* Reuniões Diárias (Daily Meeting);
* Revisão da Sprint (Review);
* Retrospectiva da Sprint.

No **Planejamento da Sprint** participa o PO, SM e o time de desenvolvimento. Nesse planejamento, num time-box de 30 dias, a reunião de planejamento dura 8h. Nas 4h iniciais é definido o que fazer e o PO explica pro time o que ele deseja para aquela sprint, o que o time precisa entregar naquela sprint. Nas últimas 4h o time de desenvolvimento quebra as habilidades e entende tecnicamente o que precisa ser feito, e definem o que precisa ser feito para entregar o que o PO pediu.

Nesse planejamento, o time pode utilizar várias ferramentas e técnicas, uma delas é o Planning Poker, onde cada um define uma estimativa de cada tarefa, e ai é verificado se tudo o que o PO queria, realmente vai dar pra entregar na sprint, e devolve pro PO o que pode realmente ser entregue.

**Reuniões Diárias (Daily Meeting):** Diariamente existem as reuniões do time, onde participam o time de desenvolvimento, o PO e o Scrum Master. O time de desenvolvimento responde três perguntas: O que foi feito no dia anterior? O que está programado para o dia? Se tem algum impedimento? Essa reunião precisa ser de pé, no mesmo local e em um mesmo horário definido pelo time (conhecida como Stand Up Meet).



O próprio time as tarefas que precisam fazer, as que estão sendo feitas e as realizadas.

**Revisão da Sprint (Review):** No último dia da sprint ocorre uma revisão, onde o time de desenvolvimento apresenta pro PO tudo que foi desenvolvido, passo a passo, pra quando colocar em produção, não seja uma surpresa para o PO. Todo o feedback falado precisa ser priorizado, pelo PO, para outra sprint, pois não dá mais tempo de mexer no software.

**Retrospectiva da Sprint:** Ocorre no último dia da sprint, todos da equipe (PO, SM e DEV) fazem uma reunião no qual todos precisam ser transparentes para apontar os erros da equipe, seja no desenvolvimento, seja na review, no planejamento da sprint entre outros, para que o time fique comprometido, melhore e não repita os erros.

**GESTÃO DE PROJETOS ÁGEIS COM SCRUM**

**Parte 1 - Papéis e Responsabilidades do 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, é o profissional que visualiza o valor que será agregado para empresa e pro cliente que vai utilizar o produto construído. O PO define a ordem que as atividades serão desenvolvidas, ele também é responsável por validar se esses itens, que estão sendo entregues nas sprints, estão agregando o valor esperado, caso não esteja, ele deve planejar os ajustes necessários nas próximas sprints. O PO deve, antes de tudo, 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 depois ver se o produto faz sentido ou não.

É importante dizer que uma planning eficiente começa com a execução de uma outra cerimônia, que não é uma oficial do framework SCRUM, mas que ajuda e muito aumentar a qualidade das plannings, ela se chama **Refining**. Na **Refining** o PO apresenta previamente pro time as histórias que deverão ser apresentadas na planning. O objetivo é o time já saber previamente o que será desenvolvido e fazer questionamentos para o PO e outros envolvidos.

A Planning precisa ser dividida em duas etapas, na **primeira etapa**, o PO irá pegar os itens mais refinados e priorizados para o Backlog e apresentar pro time. Neste momento é muito importante o PO efetuar a leitura completa das estórias pro time, para que nenhuma dúvida fique em aberto e evitar os achismos por parte do time de desenvolvimento.

Na **segunda etapa** da Planning é sugerido que o PO não participe para não exercer influência sobre as atividades a serem desenvolvidas. Nessa segunda parte, o time irá pegar cada estória e irá escrever quais atividades deverão ser desenvolvidas para que sejam consideradas entregues. Por ser uma parte mais técnica, é importante que o PO se abstenha para que o time possa colocar suas próprias opiniões e até mesmo ter algumas discussões que 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.

Com as estórias mapeadas e as atividades descritas, é definido o **Sprint Backlog**, com essa informação na mão, o PO deve avaliar quais dos itens é considerado mais prioritário e que deve ser considerado o objetivo principal da sprint.

O **Release Planning**  é a liberação ou lançamento de **software** (em inglês: **release**) é o lançamento de nova versão oficial de 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 o utilizam. Existem dois tipos de Release Planning:

* **Release Planning de Múltiplas Squads:** consiste em vários times de desenvolvimento agrupadas, fazendo várias coisas distintas, que podem ou não ter correlação entre as atividades mas que ao final da sprint devem ser agrupadas em uma única Release para serem implementadas em produção. Esse é um tipo de Release Planning 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 a produção, sem que a entrega de um squad impacte na entrega de outro squad (importante mas não tem ação direta do PO).
* **Release Planning de Projeto:** O PO possui uma demanda muito grande e será necessário quebrar a entrega em várias estórias e, consequentemente, em vários sprints. O PO deve compreender a real dimensão da demanda e a real dimensão de cada estória e deve quebrar as histórias o máximo que puder.

O papel do PO, no planejamento da Release consiste em definir as maiores entregas de valor pro cliente, priorizar as maiores entregas de valor para serem desenvolvidas primeiro, organizar as sprints para que seja possível captar o retorno do cliente o mais breve possível, com o retorno do cliente ajustar as estórias para que estas entreguem cada vez mais valor e organizando quando as release serão realizadas . Outro ponto importante é entender quando a entrega, em si, traz um ganho real para o cliente ou se a entrega será apenas estética ou funcional.

**Parte 2 - Analisando o Escopo e Definindo Prioridades**

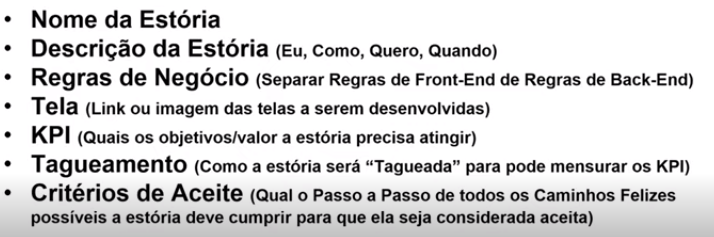
A definição do escopo é um dos pontos mais críticos do processo de criação de projetos , pois é nesse momento que será definido o que será desenvolvido e é do artefato desta fase que serão extraídos benefícios a serem atingidos. Portanto, se a definição do escopo for ruim, poderá ser atingido benefícios errados ou menos assertivos.

É importante inspecionar cada entrega, validar com o cliente 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:** é composto por artefatos chamados de **Épicos** e **Estórias:**

* **Épicos:** incremento sem muito detalhamento, ajuda a te direcionar dos caminhos que deve seguir.
* **Estória:** Detalhamento dos épicos, um épico normalmente se divide em várias estórias, onde ficam descritos o que deve acontecer e suas regras de negócio.

Modelo para escrever uma estória (não padrão):



A **Gestão de Riscos** em projetos ágeis é tão importante quanto em qualquer outro tipo de projeto. A gestão de riscos positivos é tão importante quanto os riscos negativos, pois eles também podem afetar o prazo, custo e escopo do projeto de maneira positiva e às vezes inviabilizando o projeto de maneira positiva, trazendo o resultado esperado sem necessariamente desprender do desenvolvimento planejado.

**Parte 3 - 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.

O **PO** será apenas o orquestrador desta comunicação, será responsável por juntar todas as pequenas partes e montar o planejamento do produto, porém esse planejamento poderá ser alterado a cada iteração/sprint. O **Mindset Ágil** prega que o PO deve ter autonomia para o desempenho de suas funções.

Ser ágil não é agregar valor para o cliente o mais rápido possível, é errar rápido e corrigir rápido, é estar um passo à frente.

**Transformação digital:** É trabalhar junto com o cliente, ser centralizado com no cliente e ter uma mente focada no objetivo de atender o cliente da melhor forma possível.

**Conceitos e Atividades Essenciais Para o Sucesso de um Projeto Ágil**

**Parte 1 - Estória x Tarefas**

O projeto final (objetivo) é um conjunto dos **épicos**, e este é o objetivo macro que deve ser atingido e, para ser atingido, esse objetivo deve ser quebrado em objetivos menores, esses objetivos menores são as **estórias**. As estórias por si só não dão dimensão do trabalho que deve ser feito para que o trabalho seja concluído, aí entram as **tarefas**.

**Resumindo:** Estórias são conjuntos de tarefas e um épico é um conjunto de estórias.

A **estória** deve ser descrita em nível de negócio e serve para que o time de desenvolvimento entenda o objetivo daquela atividade, 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.

**Parte 2 - Critérios de Aceite, Estimativa e Planejamento de Tarefas**

**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 P.O. a detalhar em alto nível o que é necessário para entregar valor ao cliente.

O P.O. precisa saber qual o objetivo a demanda precisa cumprir para que a demanda seja considerada aceita e concluída, ou seja, é o objetivo que aquela estória deve concluir.

**Estimativa e Planejamento:** o método mais utilizado para fazer estimativa é o *Planning Poker.* Esse método é de fato um jogo de cartas onde cada membro do time de desenvolvimento recebe um deck contendo todos os números da sequência de fibonacci. 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 em se realizar aquela tarefa, o trabalho manual em si e o tempo. Existe tarefas em que a complexidade de execução é muito baixa, porém demandam muito trabalho manual dos desenvolvedores. Os membros do time que deram notas altas devem justificar o motivo da sua pontuação, pois podem ter visualizado/pensado em alguma coisa que outros membros do time não consideraram e esse é o um dos maiores ganhos que temos nesse framework.

Obs.: Existe o método **Planning Tshirt Size**.

**Planejamento:** por ser uma parte mais técnica, é importante que o P.O. se abstenha para que o time possa colocar suas próprias opiniões e ter umas discussões que não teriam na frente do P.O. e durante a segunda parte da planning o time pode fazer um outro corte e diminuir ainda mais as estórias a serem entregues, visto que ao quebrar as estórias podem validar que são mais complexas do que imaginavam.

**Parte 3 - Relacionamento com Clientes/Stakeholders**

O **Stakeholder** 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 podem 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.

**Rotinas de Um Time Ágil**

**Parte 1 - Daily e Retrospectiva**

A **Daily Stand-Up Meeting** com já sabemos, deve ter um tempo total de 15 min, ocorrer no mesmo horário e lugar, com a presença de todo o time de desenvolvimento. A presença do SM e PO não é obrigatória. A *Daily* é para o time de desenvolvimento saber o que cada um está fazendo, e se alguem tiver com problema, alguem pode ajudar de alguma forma. Normalmente é cobrado as três perguntas mágicas: O que foi feito ontem? O que será feito hoje? Possui algum impedimento?

Ao contrário da *Daily* a **Retrospectiva** acontece apenas uma vez na sprint, sempre após a conclusão da sprint. Nessa cerimônia a presença do PO é muito importante, mas não obrigatória. É discutido principalmente os pontos que foram bons e ruins na sprint, o que deve-se melhorar ou não se deve fazer e o que pode tentar ser feito já na próxima sprint.

**Parte 2 - Refinamento**

O **Refinamento** é uma cerimônia não oficial da SCRUM, porém muito utilizada para que o time, juntamente com o PO, discuta como será a próxima sprint, adiantar os entregáveis e adiantar possíveis dúvidas. O refinamento tem a finalidade de aumentar o entendimento da demanda por parte do time, melhorar a qualidade da entrega e aumentar o retorno esperado para o projeto. É de extrema importância que a estória já esteja escrita para que fique menos dúvidas possíveis, com a ideia de facilitar a Planning (Todo o time SCRUM deve participar).

Na **Review**, quem apresenta é o time de desenvolvimento, neste momento, eles tem a oportunidade de mostrar o que cada um desenvolveu na sprint anterior e podem tirar qualquer dúvida técnica que o público presente tenha em relação a solução. Dúvidas de negócio continuam sendo tiradas pelo PO, que deve estar presente na cerimônia, assim como o SM. Todos os interessados na entrega são convidados (stakeholders).

**Objetivo:** Verificar se a demanda agrega valor ao negócio.

**Parte 3 - Maturidade da Equipe**

Várias ferramentas na internet podem auxiliar na definição da maturidade. O amadurecimento do time está diretamente ligado ao ambiente em que está situado. Um time maduro, é um time que anda sozinho.