

## HOMEPAGE:

# Metodologias Ágeis:

[Descrição Breve]

Metodologias ágeis são eficazes para gestão de projetos, otimizando processos, organizando demandas e entregas, ou seja, tem foco em aceleração de finalização de projetos.

As 5 metodologias ágeis mais utilizadas são:

- Kanban
- Lean
- Scrum
- Extreme Programming (XP)
- Test Driven Development (TDD)

Cada uma dessas têm sua forma de trabalho e de aplicação, a escolha varia conforme a necessidade e preferência do órgão aplicador.

**Kanban:** O método Kanban veio à tona como melhoria nos processos de gerenciamento de projetos em 2019, apesar de ter sido aplicado a primeira vez em 1947. No mesmo ano de sua primeira aplicação, surgiu seu nome: “Kanban”, japonês para “Sinalização”; o nome remete diretamente ao funcionamento dessa metodologia ágil, que consiste em 5 princípios: a) Definir o Trabalho em Andamento (WIP);

- b) Visualizar o fluxo de trabalho;
- c) Mensurar e organizar esse fluxo;
- d) Deixar claro as políticas do processo de criação;
- e) Usar modelos para reconhecer as oportunidades de melhorias.

Apesar de parecerem princípios rasos ou básicos, cada um deles têm suas especificidades:

a) O WIP é o núcleo de toda a metodologia Kanban, é o que define o máximo de tarefas em cada etapa do quadro Kanban. Usualmente identificado pelo gerente do projeto para evitar períodos estagnados.

b) Visualização do fluxo de trabalho é tão importante quanto o WIP, pois é o processo de trazer à tona os mecanismos, interações, filas de espera e demandas, que são elementos cruciais para se verificar antes da aplicação no projeto.

c) Mensurar e organizar o fluxo de trabalho é crucial para que não haja pausas no progresso geral, além de ser utilizado como um quadro para melhorias. O foco é continuar seguindo em frente, apesar de não remover todos os empecilhos.

d) Definir claramente as políticas de criação é de extrema importância para que todos se sintam encorajados e dispostos no time, costuma a trazer bons ritmos de trabalho

por considerar diversas alternativas para prosseguir com o que foi requerido, sem restrição por regras rígidas.

e) Usar modelos para melhorias é extremamente comum, visto que o Kanban adota um método quantitativo para melhorias inovadoras. Essa etapa geralmente é a que controla os desperdícios e controla os fluxos a partir da Teoria das Restrições, que aborda sobre não alcançar mais do que suas metas por um número mínimo e seletivo de restrições.

Apesar de todas essas especificações, o mais visto e utilizado do método Kanban ainda não foi aprofundado: seus quadros. Os quadros de Kanban se dividem em, primordialmente, 3 colunas: To Do (Para fazer), Doing (Fazendo) e Done (Feito), nessas colunas são colocadas as tarefas e seus respectivos estágios, há de exemplo lavar louça - quando a pia possui louças não lavadas, ela está no estágio de To Do, pois será necessário lavar essas louças em algum momento, quando estão sendo lavadas, mesmo que em períodos diferentes e pausados, elas vão para o Doing; assim que as louças estão limpas e a pia vazia, considera-se a tarefa como Done.

Fontes: <http://paulorodrigues.pro.br/ojs/ijmp/index.php/ijmp/article/view/1482>

**Lean:** A metodologia Lean é amplamente utilizada no âmbito de desenvolvimento de Software, sendo muitas vezes usada junta a outras metodologias ágeis. Lean é visto tal qual uma filosofia, potente e benéfico.

O pensamento Lean surgiu no Japão, nos anos 40, o foco é maximizar o valor e diminuir os desperdícios na produção, mas nunca houve um modo específico de se categorizar esse método, mas é possível dividir o Lean, no geral, por alguns princípios:

- a) Valor (Value);
- b) Fluxo de Valor (Value Stream);
- c) Fluxo (Flow);
- d) Empurrar (Pull);
- e) Perfeição (Perfection).

Esses princípios possuem suas especificações, sendo elas:

- a) O Valor é entender pela visão do cliente, o que for feito organizadamente terá valor maior ao cliente. Mas, se algo é de pouco valor, acaba por virar desperdício e deve ser retirado.
- b) O Fluxo de Valor é uma sequência de ações que leva o produto desde a ordem do cliente até o cuidado do cliente, garantindo que cada passo dessas atividades produza valor ao produto.
- c) O Fluxo é necessário para que o Fluxo de Valor não fique descontínuo nem desorganizado, um “Fluxo” correto significa permitir uma entrega sem turbulências, suave.
- d) O Empurrar implica na produção de produtos apenas quando necessário ou

requisitado pelo cliente.

e) A Perfeição é o anseio por sempre ser melhor, obter melhorias até chegar em zero defeitos.

O Lean também utiliza desses princípios para ser combinado com as outras metodologias ágeis, tal qual o Scrum ou o Kanban; no Scrum é mais útil para identificar desperdícios para descarte, já com o Kanban, foca em otimização de fluxo para a entrega de prazos. O ponto é que Lean cria uma melhora contínua, uma abordagem robusta e eficaz.

Fontes: <https://ieeexplore.ieee.org/abstract/document/6759188>  
[https://link.springer.com/chapter/10.1007/978-3-642-16416-3\\_4](https://link.springer.com/chapter/10.1007/978-3-642-16416-3_4)

**Extreme Programming (XP):** Extreme programming (Programação ao extremo) não foca apenas em ser uma metodologia ágil, mas também em ser uma metodologia para times, possuindo foco específico para trabalhos em equipe. É um método associado à performance, do time e do projeto, em XP, colaboração é a chave para uma performance bem-sucedida.

Há três práticas de equipe se destacam nessa metodologia:

- a) Padrões de código coletivos: todos seguem um estilo comum, garantindo compreensão compartilhada do software.
- b) Propriedade coletiva do código: qualquer membro pode modificar qualquer parte do sistema, o que distribui responsabilidade e fortalece a colaboração.
- c) Integração contínua: pequenos incrementos são testados e validados constantemente, assegurando qualidade e rapidez no desenvolvimento.

Aplicando-as, é observado um estímulo de responsabilidade compartilhada, redução de falhas e transformação para uma equipe mais ágil na resposta a mudanças. O XP entende que os melhores resultados em arquitetura, requisitos e design emergem de equipes organizadas do modo mais benéfico para elas (auto-organizado), que aprendem e evoluem a cada interação com o cliente e entre si.

Fontes: <https://www.sciencedirect.com/science/article/pii/S0950584912002091>

**Test Driven Development (TDD):** Test Driven Development (Desenvolvimento Orientado a Testes) é utilizado esporadicamente por décadas, com ele, casos testes são incrementados de antemão, anterior ao código da produção ser implementado. O TDD permite o uso do XP e, ao usar esse método, se observam aumentos de até 40% na produtividade da equipe. O modo de funcionamento é: primeiro um código é feito, após isso, observa-se seus erros e implementa o

necessário para o funcionamento correto desse código, em seguida é reestruturado até ficar o mais otimizado e funcional. Utilizando outras metodologias, pode vir a ser um dos métodos ágeis mais úteis no desenvolvimento de software e trabalho em equipe.

Fontes: <https://ieeexplore.ieee.org/abstract/document/1251029>

## **SCRUM:**

[Descrição Breve]

Scrum é uma das metodologias ágeis mais utilizadas nos tempos atuais. Ela foi criada em 1995 por Ken Schwaber e Jeff Sutherland, visando uma estrutura básica, mas funcional, para trabalho em equipe, com divisões de trabalho, priorização e tempo limite antes de cada entrega, que são divididas em partes chamadas de “Sprint”. Apesar de não haver uma hierarquia ou cargos superiores, existem dois cargos que possuem mais responsabilidades, sendo esses os Scrum Master (SM) e Project Owner (PO), enquanto o SM cuida das seções sociais, o PO tende a focar na parte técnica, ambos costumam ter mais contato com o cliente do que os Team Developers (TD).

### **SPRINT:**

Uma sprint é a reunião das pessoas envolvidas no projeto, aspirando promover um desenvolvimento mais focalizado do projeto. Sprints normalmente têm duração de uma a quatro semanas e são apresentações dos resultados até o dado momento.

### **FUNÇÕES:**

Scrum Master (SM):

#### **1. FUNÇÃO DO CARGO**

O Scrum Master é responsável por estabelecer toda a ideia desta metodologia ágil, ele sabe como funciona a prática e a teoria da aplicação, tanto dentro da equipe como na organização.

No entanto, um Scrum Master é muito mais do que isso. O papel do Scrum Master tem muitas camadas e facetas. Ao mesmo tempo em que conscientiza sobre o Scrum e permite maior agilidade, os Scrum Masters também precisam de habilidades interpessoais, necessárias para orientar e orientar os membros do Time Scrum e outros na organização

No dia a dia, ele seria o “eliminador de ruídos”, alguém com a capacidade de resolver conflitos e amenizar problemas de forma eficaz. Se há algum impedimento no trajeto

do projeto, ele que vai ser a primeira pessoa a tomar a atitude de entender o que está dando errado.

O pilar filosófico que sustenta o papel do Scrum Master é o conceito de liderança servidora. Um líder servidor prioriza as necessidades dos membros da equipe e dos clientes que eles servem, com o objetivo de alcançar resultados alinhados aos valores e objetivos da organização. Esta abordagem representa uma inversão do modelo de gestão tradicional de "comando e controle". Em vez de ditar ordens e gerenciar tarefas, o Scrum Master lidera "por trás", capacitando a equipe, facilitando suas interações e atuando como um defensor de suas necessidades.

Além disto tudo, ele serve a equipe atuando como um coach, orientando os membros com sua autogestão e multifuncionalidade e ajuda na questão da produtividade eficaz da equipe, ajudando-os a se alinharem nos eixos das demandas e objetivos do projeto.

Junto com isso, ele também precisa fazer uma grande parceria com o Product Owner (PO) que é vital para o sucesso do produto. Eles vão tentar achar a forma mais eficaz de compreender a necessidade de itens de backlog claros e concisos.

Em relação aos Developers, ele os ajuda de diversas formas, pode ser por treinamento de alguma habilidade, prover os recursos necessários para o avanço do projeto através das Soft Skills e Hards Skills

## 1.1. DESENVOLVIMENTO DE SKILLS

Soft e Hard Skills são conceitos importantes para poder desenvolver-se na vida pessoal e na vida empresarial. Respectivamente sendo habilidades interpessoais e técnicas. Elas não necessariamente surgem do nada, elas são desenvolvidas e podem ser estimuladas através do exercício delas.

### SOFT SKILLS DE UM SCRUM MASTER:

- **Comunicação Eficaz:** A habilidade de se comunicar de forma clara, concisa e adaptada a diferentes públicos é muito necessária, para poder esclarecer tudo que está acontecendo no projeto.
- **Empatia e Inteligência Emocional:** Compreender e se conectar com os sentimentos e perspectivas dos membros da equipe é essencial para poder entender a situação de forma humanizada e orgânica
- **Resolução de Conflitos:** Conflitos são naturais em qualquer equipe, mas como SM você tem o papel de amenizar qualquer faísca de problema que surgir.
- **Liderança Servidora:** O Scrum Master não é um gerente tradicional, no Scrum não há hierarquias de poder, apenas de organização. Ser um servidor significa

estar pronto para adaptar-se em qualquer situação que pode dificultar o processo.

- Coaching e Mentoria: Atuar como um coach, ajudando a equipe a desenvolver suas habilidades, a se auto-organizar e a internalizar os princípios ágeis é uma responsabilidade central. Isso envolve orientar, mas não ditar, as soluções.

Se você é um novo Scrum Master, o conhecimento técnico é importante. Porém o essencial é desenvolver o lado emocional por ser um gestor que auxilia sua equipe em crescer nessa metodologia ágil.

#### HARD SKILLS DE UM SCRUM MASTER:

- Conhecimento Profundo do Framework Scrum e dos Princípios Ágeis: É imprescindível que o Scrum Master domine o Guia do Scrum, seus papéis, eventos, artefatos e regras. Além disso, deve ter uma compreensão sólida dos valores e princípios do Manifesto Ágil.
- Gerenciamento de Riscos: A capacidade de identificar, analisar e planejar a mitigação de riscos que possam impactar o andamento do projeto e o alcance das metas da Sprint.
- Domínio de Ferramentas de Gerenciamento Ágil: Proficiência em ferramentas como Jira, Trello, Azure DevOps ou similares é crucial para gerenciar o backlog do produto, acompanhar o progresso das Sprints e gerar métricas que ajudem a equipe a inspecionar e adaptar seu trabalho.

Não se preocupe em possuir todos estes requerimentos da noite pro dia, você desenvolve praticando-as no seu projeto, e está tudo bem errar ali e aqui, tudo faz parte do aprendizado final. Scrum também é sobre tomar riscos e decisões complexas.

#### 1.2. DESAFIOS COMUNS NA JORNADA DO SCRUM MASTER

Como dito anteriormente, haverá muitos riscos no meio processo que precisarão ser tomados de forma crítica e empírica. As vezes pode ser um pouco frustrante ou comprometedor realizar alguma tarefa, mas lembre-se que isso no fim irá gerar os frutos das suas decisões. Se suas escolhas levaram a maior qualidade do produto, você está caminhando certo! As vezes o problema é sobre a motivação da equipe em relação ao projeto, ou a falta de conexão entre os membros, e há várias formas de poder resolver isto.

Basta você pode analisar a situação de forma "Macro", ou seja, de forma global e entender o motivo do distanciamento. Será que ninguém se conhece? Ou a relação é meio tímida? Ou eles no passado já se conheciam, mas houve algo entre eles? As possibilidades são infinitas, e as soluções também! O importante é ser honesto e falar individualmente com os integrantes se há algo impedindo-os de poder prosseguir com

a comunicação, as vezes é uma questão de tempo, ou de adaptação até chegar no ideal que é um ecossistema saudável de conversações e debates do projeto.

Outro problema bem comum é sobre querer adentrar na metodologia em si, sempre é importante ressaltar as diferenças dos outros processos e do Scrum, o que torna o Scrum mais ágil que os outros? O sistema cascata (ou waterfall) pode soar ultrapassado as vezes, por depender de outro funcionário para realizar o seu próximo passo, deixando os outros integrantes sem objetivo. O Scrum é totalmente o contrário disto, ele proporciona que cada um possa ser seu próprio gerente e gestor, como pode ser visto nesta imagem: (imagem1)

#### PERGUNTAS DA ATIVIDADE:

1. O conceito de "Liderança Servidora", fundamental para um Scrum Master, significa principalmente:

- a) Ter controle total sobre as tarefas e prazos do projeto para garantir a entrega.
- b) Servir aos diretores da empresa, garantindo que seus relatórios de status sejam a prioridade.
- c) Priorizar as necessidades da equipe, removendo impedimentos e capacitando-os para que atinjam seu potencial máximo.
- d) Atuar como o líder absoluto.
- e) Suprir as necessidades da equipe a partir da empatia.

2. Quando o texto descreve o Scrum Master como um "eliminador de ruídos", a melhor interpretação é que sua função é:

- a) Impedir os problemas de crescerem e atrapalharem o desenvolvimento do projeto.
- b) Saber quando há um problema e avisar para todos da equipe.
- c) Comunicar um problema presente para todos ajudarem na remoção dele.
- d) Filtrar a dificuldade e resolver ela sem o apoio da equipe
- e) Tratar o problema como algo absoluto e eliminar ele imediatamente.

3. Se dois ou mais desenvolvedores entram em um conflito técnico que está paralisando uma tarefa da Sprint, a atitude mais adequada do Scrum Master é:

- a) Usar sua autoridade para escolher a melhor solução técnica e instruir a equipe a segui-la.
- b) Facilitar uma conversa entre os dois, fazendo perguntas que os ajudem a chegar a uma solução, garantindo que a discussão permaneça construtiva.
- c) Levar o problema para alguém da equipe deles para que ele decida quem está certo.
- d) Pedir para eles deixarem o conflito de lado e trabalharem em outras tarefas para não atrasar a Sprint.
- e) Locomover eles para outra tarefa temporária.

4. A habilidade de "Comunicação Eficaz" para um Scrum Master é mais crucial para:

- a) Preparar apresentações de status detalhadas e complexas para a alta gestão.
- b) Convencer a equipe a trabalhar mais rápido para bater as metas estabelecidas.
- c) Traduzir os requisitos de negócio em tarefas técnicas para os desenvolvedores.
- d) Garantir que haja transparência sobre o progresso, facilitando a comunicação com o P.O e a equipe.
- e)

5. Como funciona a relação de um P.O com o S.M?

- a) Os dois trabalham paralelamente para entregar os resultados após o processo.
- b) Realizam a construção do backlog juntos, com o P.O. realizando a parte do produto, e o S.M ajudando de forma auxiliar.
- c) O P.O apenas monta o backlog e o S.M a computa em algum local.
- d) Ambos possuem funções semelhantes, mas o P.O ele foca apenas no produto.
- e) Nenhuma das questões está correta.

1-C

2-A

3-B

4-D



## Project Owner (PO):

### 1. FUNÇÃO DO CARGO

O Product Owner (P.O.) é o dono do produto, sendo o único responsável por maximizar o valor do produto resultante do trabalho da equipe de desenvolvimento. A forma como isso é feito pode variar amplamente entre organizações, equipes Scrum e indivíduos, mas a responsabilidade central permanece a mesma: garantir que a equipe esteja construindo a coisa certa, na hora certa.

O P.O. é a ponte entre o mundo dos negócios (clientes e mercado) e a equipe de desenvolvimento. Ele é a "voz do cliente" dentro do time, garantindo que as necessidades e desejos dos usuários sejam compreendidos e traduzidos em funcionalidades concretas.

No dia a dia, ele é o "guardião da visão", alguém com a capacidade de definir o que é mais importante a ser feito e de comunicar o "porquê" por trás de cada funcionalidade. Se há dúvidas sobre o que uma funcionalidade deve fazer ou qual item é mais prioritário, ele é a pessoa com a autoridade final para decidir.

O pilar que sustenta o papel do Product Owner é o conceito de gerenciamento de valor. O P.O. não é um gerente de projetos que controla o cronograma; ele é um empreendedor do produto que busca o maior retorno sobre o investimento (ROI). Em vez de focar em "entregar tudo o que foi pedido", o P.O. foca em "entregar o que gera mais impacto para o negócio e para o cliente", mesmo que isso signifique dizer "não" para muitas outras ideias.

Junto com isso, ele precisa de uma parceria fortíssima com os Developers. Enquanto o P.O. é responsável pelo "o quê" e pelo "porquê", os Developers são responsáveis pelo "como". Essa colaboração é diária e essencial para criar um produto de alta qualidade.

Em relação ao Scrum Master, eles são parceiros estratégicos. O S.M. ajuda o P.O. a ser mais eficaz, ensinando técnicas de gerenciamento de backlog, facilitação de reuniões com stakeholders e garantindo que a comunicação com a equipe flua sem ruídos.

#### 1.1. DESENVOLVIMENTO DE SKILLS

Assim como o Scrum Master, um Product Owner de sucesso precisa de um conjunto equilibrado de habilidades interpessoais e técnicas para navegar em um ambiente complexo e dinâmico.

## SOFT SKILLS DE UM PRODUCT OWNER:

**Comunicação e Negociação:** Talvez a mais crítica das habilidades. O P.O. precisa articular a visão do produto para a equipe, negociar prioridades com múltiplos stakeholders (que muitas vezes têm interesses conflitantes) e comunicar o progresso de forma clara.

**Visão de Negócio e Pensamento Estratégico:** Um bom P.O. não apenas entende o que os clientes pedem, mas também entende o mercado, a concorrência e os objetivos da empresa, alinhando o produto a essa visão maior.

**Tomada de Decisão:** O P.O. precisa tomar decisões difíceis constantemente, muitas vezes com informações incompletas. A habilidade de dizer "não" de forma construtiva é essencial para manter o foco e evitar que o produto se torne uma coleção de funcionalidades sem coesão.

**Empatia com o Cliente:** Para construir um produto que as pessoas amem, é preciso entender profundamente suas dores, necessidades e desejos. O P.O. deve ser o maior defensor do usuário final.

**Storytelling (Contar Histórias):** A capacidade de criar uma narrativa convincente sobre o produto e suas funcionalidades. Isso ajuda a motivar a equipe e a alinhar os stakeholders, explicando o valor de cada entrega de forma humana e impactante.

## HARD SKILLS DE UM PRODUCT OWNER:

**Gerenciamento de Product Backlog:** Esta é a principal ferramenta do P.O. Envolve criar, refinar e priorizar os Itens do Backlog do Produto (PBIs), escrevendo-os de forma clara (ex: usando o formato de Histórias de Usuário) para que todos entendam o que precisa ser feito.

**Conhecimento do Domínio e do Mercado:** É imprescindível que o P.O. entenda profundamente o produto, seu mercado, seus usuários e a tecnologia envolvida. Ele não precisa ser um expert técnico, mas deve entender o suficiente para ter discussões produtivas com a equipe.

**Técnicas de Priorização:** Dominar frameworks para priorização (Ex: MoSCoW, Kano Model, Value vs. Effort) é crucial para garantir que o tempo valioso da equipe de desenvolvimento seja gasto nos itens de maior impacto.

**Análise de Dados e Métricas de Produto:** Um P.O. moderno não se baseia apenas em

intuição. Ele deve ser capaz de definir e analisar métricas de sucesso do produto (Ex: taxa de conversão, engajamento do usuário, satisfação do cliente - CSAT) para tomar decisões informadas.

## 1.2. DESAFIOS COMUNS NA JORNADA DO PRODUCT OWNER

A jornada de um Product Owner é cheia de desafios que exigem equilíbrio e resiliência. O sucesso não vem de ter todas as respostas, mas de saber como navegar na incerteza e guiar o produto na direção certa, aprendendo e adaptando ao longo do caminho.

Um dos maiores desafios é gerenciar a pressão dos stakeholders. Muitas vezes, cada departamento ou cliente acredita que sua necessidade é a mais urgente. O P.O. pode se sentir como se estivesse no meio de um cabo de guerra. A solução não é tentar agradar a todos, o que é impossível. A chave é a transparência. Um Product Backlog bem priorizado e visível para todos é a melhor ferramenta. Ele deixa claro o que está sendo feito, o que virá a seguir e, mais importante, o que não está sendo feito agora. A conversa muda de "por que o meu item não está pronto?" para "vamos discutir onde este item se encaixa na estratégia geral do produto?".

Outro desafio comum é escrever bons Itens de Backlog. Um item mal escrito pode gerar retrabalho, frustração e desperdício. Itens que são muito vagos ("Melhorar a tela de login") ou muito prescritivos ("Colocar um botão azul de 12px na coordenada X, Y") são problemáticos. A solução está na colaboração. O P.O. deve trabalhar junto com os Developers no Refinamento do Backlog (Backlog Refinement). Nessa cerimônia, eles discutem os itens, tiram dúvidas, estimam o esforço e garantem que todos tenham um entendimento compartilhado antes mesmo de a Sprint começar. Um bom item de backlog é um convite para uma conversa.

Por fim, o desafio de dizer "não". Muitos P.O.s, especialmente no início, têm dificuldade em negar pedidos, com medo de frustrar stakeholders importantes. Mas um "sim" para tudo significa um produto sem foco e uma equipe sobrecarregada. A melhor forma de dizer "não" é explicando o custo de oportunidade. "Sim, podemos fazer a sua ideia, mas isso significa que teremos que adiar aquela outra funcionalidade que impacta um número maior de clientes. Dado nosso objetivo atual, ainda faz sentido priorizar a sua?". Isso transforma uma negativa em uma decisão de negócios estratégica e colaborativa.

## ATIVIDADE DO PO

**1. A principal responsabilidade de um Product Owner (P.O.), que guia todas as suas ações, é:** a) Gerenciar o cronograma do projeto e garantir que tudo seja entregue na data prometida. b) Escrever o código das funcionalidades mais importantes do

produto. c) Maximizar o valor do produto, garantindo que a equipe construa as funcionalidades de maior impacto para o negócio e para o cliente. d) Organizar as reuniões do Scrum e garantir que a equipe siga as regras da metodologia. e) Atender a todos os pedidos dos stakeholders para mantê-los satisfeitos.

**2. Como funciona a relação de um Product Owner com a equipe de Developers? a)**

O P.O. atua como um chefe, distribuindo tarefas diárias e cobrando o progresso individual de cada um. b) É uma relação de colaboração: o P.O. define "o quê" e "porquê" (o problema a ser resolvido e seu valor) e os Developers definem "como" (a solução técnica). c) Os Developers trabalham de forma isolada, e o P.O. apenas valida o trabalho no final da Sprint. d) O P.O. é responsável por escrever todo o detalhamento técnico das tarefas para os Developers. e) Eles não possuem uma relação direta, pois o Scrum Master faz a mediação.

**3. Um dos desafios mais comuns do P.O. é lidar com múltiplos stakeholders que querem suas solicitações priorizadas. A melhor ferramenta para gerenciar isso é:**

a) Dizer "sim" para todos e pedir para a equipe trabalhar mais rápido. b) Um cronograma detalhado (gráfico de Gantt) mostrando exatamente quando cada item será entregue. c) Fazer reuniões individuais e secretas com cada stakeholder. d) Manter um Product Backlog transparente e bem priorizado, que sirva como a única fonte da verdade sobre o que será feito. e) Ignorar os stakeholders e focar apenas no que a equipe de desenvolvimento acha importante.

**4. Qual das seguintes opções é considerada uma HARD SKILL essencial para um Product Owner? a)**

Liderança Servidora. b) Gerenciamento de Product Backlog e conhecimento de técnicas de priorização. c) Resolução de conflitos internos da equipe. d) Coaching e Mentoria sobre a metodologia ágil. e) Empatia e inteligência emocional.

**5. Por que a habilidade de "dizer não" é tão crucial para o sucesso de um Product Owner? a)**

Para mostrar autoridade e controle sobre a equipe de desenvolvimento. b) Para evitar trabalho e reduzir os custos do projeto o máximo possível. c) Porque o foco é essencial. Dizer "não" para ideias de baixo valor permite dizer "sim" para as funcionalidades que realmente importam e movem o produto na direção estratégica

correta. d) Para garantir que apenas as tarefas mais fáceis e rápidas sejam selecionadas para a Sprint. e) Nenhuma das alternativas está correta.

## GABARITO

1. C
2. B
3. D
4. B
5. C

### Scrum Team (ST):

#### 1. Função do cargo:

O Scrum Team exerce apenas os papéis que o Scrum estabelece, nada mais além do que a metodologia já explica, diferenciando-se do Scrum Master e do Product Owner, que possuem papéis mais específicos e além de apenas seguir o método.

Não é incomum que os integrantes, no cargo de ST, sejam vistos como menos importantes. O fato é que todas as funções desempenhadas no método Scrum são dependentes uma das outras, assim como não haveria Scrum Team sem a guia de um Scrum Master, não haveria SM sem as atualizações dadas pelo Product Owner, além disso, ambos dependem da produção e do desenvolvimento estabelecido pelos membros do ST, assim formando uma coexistência necessária.

Os que desempenham essa função necessitam sempre estar alerta sobre suas próximas tarefas, mas nunca sendo omitidos da partilha dessas: não há cargos maiores ou menores, apenas diferentes, então os integrantes do Scrum Team também possuem a liberdade de opinar na organização e divisão de atividades. Com isso em mente, sabe-se que as tarefas na área de desenvolvimento serão, majoritariamente, direcionadas ao ST, pois eles possuem um tempo e foco maior para essa seção, tendo em vista que o SM e PO terão de cuidar da parte sociável e administrativa também de todo o negócio, além de repassarem e reforçarem os passos que devem ser tomados para o produto final.

Sempre atento, dividindo suas tarefas para entregar no prazo correto e congruente com as explicações e observações dos outros cargos, o Scrum Team deve também ser

âncora de auxílio para seus parceiros de equipe; é factual que alguns membros terão mais facilidades em certos assuntos do que outros, cada um terá sua área de expertise, na ciência disso, aqueles que finalizaram suas tarefas com maior prioridades, devem estar abertos para auxiliar os com maior dificuldade, mas apenas se for em momentos oportunos e de conhecimento do que terminou sua tarefa, caso contrário, duas mentes confusas podem acabar desacelerando ainda mais um processo. Mas vale lembrar que talvez o desacelerar de um processo pode vir acompanhado de uma aquisição de competência e conhecimento, que vale tanto quanto uma tarefa adiantada (mas não vale o preço de uma tarefa atrasada).

Na equipe em que se utiliza a metodologia ágil Scrum, será decidido entre eles quem ocupa cada cargo, então é de se esperar que os PO e SM sejam aqueles com maiores habilidades para exercer tais cargos, significando que devem servir de instrutores para os ST em momentos de “aperto”, facilitando para o aprendizado de habilidades necessárias.

## 1.1 Skills Necessárias:

Algumas habilidades são indispensáveis para quando se é um membro do Scrum Team, isto é, a falta de algumas habilidades ou a falta de interesse para as obter pode causar uma gradual desaceleração em todo o processo, podendo danificar os prazos de Sprints ou os estabelecidos pela equipe. Essas skills se dividem em dois ramos: Soft Skills (Habilidades sociais) e Hard Skills (Habilidades técnicas):

### SOFT SKILLS DO SCRUM TEAM:

**Atenção NÃO Seletiva:** Para um membro do SM, é indispensável que se preze para ter atenção e foco nos ensinamentos, instruções e atividades que o PO e SM irão comunicar, não há como haver uma boa comunicação se ambos os lados não colaborarem.

**Diálogo Saudável:** Eventualmente, pode vir a ocorrer momentos em que alguns membros estão fazendo mais que outros, em pontos como esses a calma e respeito devem ser mantidos, não há método Scrum sem o trabalho em equipe, comunicar aos seus parceiros de equipe de forma respeitosa e com profissionalismo é prioridade sempre na comunicação.

**Conhecimento da Metodologia:** Mesmo que o Scrum Team não seja igual o PO ou SM, que direcionam o projeto, eles também devem saber o funcionamento do método,

visto que isso facilita para toda a equipe se mobilizar e performar dentro das expectativas e dos prazos.

## HARD SKILLS DO SCRUM TEAM:

**Skills de Desenvolvimento Conjuntas:** cada membro possui competências e técnicas diferentes (ex.: programação, testes, design, integração), utilizar suas competências em desenvolvimento juntos permite que o time entregue um incremento completo

**Estimativa:** habilidade de prever a quantidade de trabalho que pode ser concluída no ciclo conforme o nível do time, skill necessária para o facilitamento do Planning Poker.

**Criação e Execução de Tarefas Técnicas:** capacidade de transformar User Stories em tarefas concretas, focando sempre no entendimento da tarefa para outros membros, assim, entregando software funcional dentro do sprint. Quem cria o Backlog são os PO, porém, é de incontestável ajuda se o ST puder enriquecer as User Stories.

Apesar de necessárias, não são habilidades pré-requisitos para entrar numa equipe Scrum, mas sim habilidades que terão de ser desenvolvidas, com vontade e foco para atingir máximas capacidades. Durante as Sprints, todos trabalhando no projeto irão inconscientemente desenvolver grande parte dessas Skills.

## 1.2 Complicações e Desafios:

Muitas das complicações já foram abordadas anteriormente neste capítulo, retomando, seriam: atenção aos pedidos do cliente repassados pelo PO e SM; desenvolver habilidades para o melhor desempenho de toda a equipe; colocar-se como importante e necessário, não menos que PO e SM; aprender a comunicação saudável, enfrentando os empecilhos que virão a ocorrer por descuidos ou desatenção de alguém da equipe.

Todas essas complicações devem ser contornadas de alguma maneira, sempre buscando o caminho menos hostil e o mais eficaz. Deve-se observar o que está acarretando nessas complicações e qual a forma mais adequada de se resolver o problema pela raiz.

### **PERGUNTAS PARA ATIVIDADE (ST):**

1. Em uma equipe dividida pela metodologia Scrum, qual a ordem mais adequada para se distribuir os cargos de forma hierárquica?

- a) Scrum Master > Product Owner > Scrum Team
- b) Scrum Team > Scrum Master > Product Owner
- c) Product Owner > Scrum Master > Scrum Team
- d) Product Owner = Scrum Master > Scrum Team
- e) Scrum Team = Scrum Master = Product Owner

(1 – e)

2. Para um membro do ST, quais Skills seriam mais apropriadas e de maior auxílio para o time?

- a) Liderança e organização.
- b) Proatividade e boa comunicação.
- c) Entendimento do método e do Backlog.
- d) Resolução de conflitos e estimativas de trabalho.
- e) Conhecimento do trabalho em equipe e coaching.

(2 – b)

3. Qual a função primordial do Scrum Team numa equipe de desenvolvimento?

- a) Seguir à risca as tarefas designadas pelos outros cargos, Scrum Master e Product Owner.
- b) Atentar-se apenas aos detalhes técnicos para que a Sprint seja entregue no prazo correto.
- c) Cumprir o trabalho que lhe foi dado e aguardar as outras partes ociosamente.
- d) Concluir sua parte de desenvolvimento, auxiliando ou sendo auxiliado por outros cargos.
- e) Voltar a atenção para a liderança na ausência do Scrum Master ou do Product Owner.

(3 – d)

---



## TESTE:

Caso queira descobrir qual das categorias mais te adequa, basta realizar este QUIZ!  
[Quiz para identificação: Mais parecido com SM, PO ou TD?]

## CONTATO:

Quer entrar em contato conosco?

[Nome / email / mensagem --> Teria que linkar direto do site com forms html]

Aqui estão algumas alternativas!

Scrum Master:

Project Owner:

[email institucional dos SM e PO (ou de todos)]

## SOBRE NÓS:

Bem-vindo ao FatecOS! Somos uma equipe de desenvolvedores de software direcionados pela metodologia ágil Scrum, com o objetivo de produzir um site focado em lecionar o método Scrum. Nossa equipe trabalha em conjunto, seguindo os princípios ensinados, para garantir a entrega de um produto de alta qualidade.

Nosso objetivo é criar um site que seja uma referência para quem deseja aprender ou aprimorar suas habilidades em Scrum. Queremos fornecer uma plataforma que seja útil, intuitiva e fácil de usar, refletindo os valores da Fatec e da metodologia.

Trabalhamos em sprints, utilizando o GitHub para gerenciar nossos códigos e documentações. Cada sprint é cuidadosamente planejada para garantir que estamos entregando um produto de valor aos nossos usuários. Nossa equipe é composta por desenvolvedores talentosos que estão comprometidos a entregar um resultado além das expectativas.

Nossa equipe é formada por universitários inspirados, divididos em:

- Desenvolvedores: responsáveis por escrever, testar e corrigir o código que constitui o site.
- Product Owner: define as prioridades e fornece orientações sobre o que cada sprint abrangerá.
- Scrum Master: facilita as reuniões e melhora a eficiência de cada sprint.

Estamos comprometidos em entregar um site que atenda às necessidades dos nossos usuários. Trabalhamos com dedicação para garantir que o FatecOS seja uma ferramenta valiosa para a comunidade.

---

## # Informações Teóricas

### MANIFESTO ÁGIL:

O manifesto foi criado por um grupo de 17 desenvolvedores de software que se autodenominaram "The Agile Alliance". O objetivo era discutir o futuro do desenvolvimento de software. Eles estavam frustrados com os processos tradicionais da época (como o modelo "Cascata"), que eram pesados, burocráticos e lentos. Cada um deles já praticava ou havia criado suas próprias metodologias "leves" (lightweight methods). O Manifesto foi o documento que consolidou os valores que todos eles tinham em comum.

Este manifesto é a raiz de todas as outras metodologias ágeis conhecidas hoje em dia. Algumas até existiam antes dele, mas foram se adaptando com os novos valores propostos pelo documento. Ele deu ao Scrum um apoio enorme, validando os princípios que o Scrum já utilizava, mostrando que não era apenas uma ideia isolada, mas parte de um movimento maior com valores compartilhados por outros grandes pensadores da área.

Fonte: <https://agilemanifesto.org/>

<https://agilealliance.org/>

### # COMO FUNCIONA O PROCESSO

Nosso site oferece um treinamento online sobre metodologias ágeis, com um foco principal no framework Scrum. A estrutura do treinamento foi projetada para ser intuitiva, guiando o usuário através de um aprendizado progressivo a partir de capítulos.

Os cursos são dos 3 cargos Scrum, respectivamente: Scrum Master, Product Owner e Development Team. Cada curso possui 3 capítulos com cada um deles intercalando com atividades e perguntas para reforçar o seu aprendizado.

Ao fim do curso, você fará o teste final para ver seu conhecimento ao longo do percurso total. O resultado será avaliado pelo o avaliador para poder atribuir o valor em seu sistema.

