**Metodologias Ágeis:**

* Kamban
* Scrum
* XP
* Lean

**Kanban:**

**Esboço da Metodologia Kanban para o Projeto TechFit**

A metodologia Kanban é um sistema de fluxo de trabalho visual que foca em otimizar o processo e limitar a quantidade de trabalho em andamento. Para o projeto **TechFit**, o Kanban ajudaria a equipe a visualizar todas as tarefas, identificar gargalos e garantir que as funcionalidades sejam entregues de forma contínua e eficiente.

**1. Quadro Kanban (O Coração da Metodologia)**

O elemento central do Kanban é o quadro, que pode ser físico (um quadro branco) ou digital (Trello, Jira). Ele terá colunas que representam o fluxo de trabalho do projeto TechFit.

* **Para Fazer (To Do / Backlog):** Todas as funcionalidades e tarefas que precisam ser desenvolvidas. As tarefas seriam representadas por cartões.
  + [Cartão] Desenvolver Tela Inicial
  + [Cartão] Criar Telas de Cadastro e Login
  + [Cartão] Implementar Lógica de Agendamento
  + [Cartão] Integrar com Sistema de Biometria
  + [Cartão] Configurar Chat de Suporte
  + [Cartão] Criar Sistema de Notificação de Aulas
* **Em Andamento (Doing / In Progress):** Tarefas que a equipe está trabalhando no momento. Esta coluna teria um limite de trabalho em andamento (WIP - Work In Progress) para evitar sobrecarregar a equipe. Por exemplo, "WIP = 3", significando que no máximo 3 cartões podem estar nesta coluna ao mesmo tempo.
* **Em Teste (Testing / Review):** Tarefas que foram desenvolvidas e estão prontas para serem testadas.
* **Pronto (Done):** Tarefas concluídas e prontas para serem implementadas.

**2. Fluxo de Trabalho e Regras**

O fluxo no Kanban é **puxado**. Isso significa que a equipe só "puxa" um novo cartão para a coluna "Em Andamento" quando há espaço (ou seja, quando o WIP não foi atingido) e quando eles terminam a tarefa anterior.

1. Um desenvolvedor finaliza a tela de login.
2. Ele move o cartão para a coluna **"Em Teste"**.
3. Um novo espaço se abre na coluna **"Em Andamento"**.
4. O desenvolvedor "puxa" a próxima tarefa de alta prioridade (por exemplo, "Agendamento de Avaliação Física") da coluna **"Para Fazer"** para a coluna **"Em Andamento"**.

Essa abordagem garante que as tarefas sejam finalizadas antes de começar novas, o que melhora a eficiência e a qualidade.

**3. Foco na Melhoria Contínua**

O Kanban não tem eventos formais como as reuniões diárias do Scrum, mas incentiva a comunicação e a análise do fluxo de trabalho. A equipe pode se reunir informalmente para discutir o progresso e o que pode ser melhorado.

* **Métricas Visuais:** O quadro por si só já fornece uma visão clara do andamento. A equipe pode ver rapidamente onde estão os gargalos (pilhas de cartões em uma única coluna) e trabalhar para resolvê-los. Por exemplo, se muitos cartões se acumulam na coluna "Em Teste", a equipe pode decidir priorizar a automação de testes ou adicionar mais pessoas a essa etapa.
* **Tempo de Ciclo (Cycle Time):** Uma métrica importante no Kanban é o tempo que um cartão leva para ir de "Em Andamento" até "Pronto". O objetivo é reduzir este tempo, tornando o processo mais rápido e previsível.

O Kanban para o projeto TechFit seria ideal para uma equipe que precisa de flexibilidade para adicionar ou mudar a prioridade das tarefas a qualquer momento, sem a rigidez das Sprints. Ele se concentra na entrega contínua e na otimização do fluxo de trabalho, garantindo que o valor seja entregue aos clientes de forma constante.

**KANBAN EXEMPLO:**

Tela de celular com publicação numa rede social

O conteúdo gerado por IA pode estar incorreto.

**SCRUM:**

**Esboço da Metodologia Scrum para o Projeto TechFit**

A metodologia Scrum é um framework ágil, ideal para projetos como o **TechFit**, porque permite que a equipe se adapte rapidamente às mudanças e entregue valor de forma contínua. Aqui está um esboço de como ela poderia ser aplicada.

**Papéis Scrum**

* **Product Owner (PO):** A pessoa responsável por definir a visão do produto. Para o TechFit, o PO seria o dono da academia ou um gerente de produto que entende as necessidades dos clientes. Ele(a) seria responsável por gerenciar o **Product Backlog** e priorizar as funcionalidades.
* **Scrum Master (SM):** O(A) facilitador(a) que garante que a equipe siga as práticas Scrum, remove impedimentos e protege o time de interrupções externas.
* **Time de Desenvolvimento:** A equipe multifuncional (programadores, designers, etc.) que constrói o sistema. Para o TechFit, isso incluiria desenvolvedores front-end para as telas e back-end para as lógicas de agendamento e biometria, além de designers de UX/UI.

**Eventos Scrum**

* **Sprint:** Um período de tempo fixo (geralmente de 1 a 4 semanas) durante o qual o time trabalha em um conjunto de funcionalidades. No TechFit, uma Sprint poderia durar 2 semanas.
* **Sprint Planning:** No início de cada Sprint, a equipe se reúne para planejar o que será desenvolvido. O time selecionaria itens do Product Backlog para criar o **Sprint Backlog**. Por exemplo, na primeira Sprint, o time poderia focar nas telas de login e cadastro.
* **Daily Scrum (Reunião Diária):** Uma reunião rápida de 15 minutos, todos os dias, para sincronizar o progresso. Cada membro do time responde a três perguntas: O que eu fiz ontem? O que eu farei hoje? Há algum impedimento me atrapalhando?
* **Sprint Review:** Ao final da Sprint, a equipe demonstra as funcionalidades concluídas para o Product Owner e outras partes interessadas. Por exemplo, o time poderia apresentar a tela inicial com as fotos das filiais.
* **Sprint Retrospective:** Após a Sprint Review, a equipe se reúne para discutir o que funcionou bem, o que pode ser melhorado e o que deve ser ajustado para a próxima Sprint.

**Artefatos Scrum**

* **Product Backlog:** Uma lista de todas as funcionalidades, melhorias e correções para o projeto **TechFit**, priorizadas pelo Product Owner.
  + **Exemplos de itens do Product Backlog:**
    - Tela de login e cadastro com planos.
    - Agendamento de avaliação física.
    - Sistema de controle de acesso com biometria.
    - Chat de suporte.
    - Notificação para professores em caso de ausência do aluno.
* **Sprint Backlog:** O subconjunto de itens do Product Backlog que o time seleciona para trabalhar em uma Sprint específica.
* **Incremento:** O resultado do trabalho concluído em uma Sprint. É a versão atualizada e funcional do sistema TechFit.

Esse ciclo se repete até que o sistema **TechFit** esteja completamente desenvolvido e todas as funcionalidades tenham sido entregues. O Scrum garante que o projeto evolua de forma transparente e que a equipe possa se adaptar a qualquer mudança ou novo requisito que surja no caminho.

**SCRUM EXEMPLO:**

Interface gráfica do usuário, Site

O conteúdo gerado por IA pode estar incorreto.

**LEAN:**

**Esboço Usando a Metodologia Lean para o Projeto TechFit**

A metodologia **Lean** se concentra na eliminação de desperdícios, na entrega de valor ao cliente e na melhoria contínua. Para o projeto TechFit, isso significa construir apenas o que é realmente necessário e validar cada etapa com os usuários.

**1. Definir Valor e Fluxo de Valor**

O primeiro passo é entender o que o cliente (o aluno da academia) realmente valoriza. Para o TechFit, o valor principal é a **facilidade de acesso e agendamento**. O fluxo de valor pode ser mapeado da seguinte forma:

* **Necessidade do Cliente:** O aluno precisa se cadastrar, ver horários e agendar uma aula ou avaliação.
* **Fluxo:** Tela inicial → Cadastro/Login → Agendamento de aulas/avaliações → Acesso à academia.

A partir desse mapeamento, identificamos possíveis desperdícios (atrasos, funcionalidades complexas, retrabalho) que devem ser eliminados.

**2. Mapeamento de Fluxo de Valor (Value Stream Mapping)**

A equipe do TechFit criaria um quadro (Kanban) para visualizar o fluxo de trabalho e identificar gargalos. O quadro teria colunas como:

* **Backlog (To Do):** Lista de funcionalidades a serem desenvolvidas.
  + Tela de Login e Cadastro
  + Agendamento de Avaliação Física
  + Agendamento de Aulas
  + Integração com Catraca
  + Notificação para Professor
  + Chat de Suporte
* **Em Desenvolvimento (Doing):** O que a equipe está trabalhando no momento.
* **Em Teste (Testing):** Funcionalidades prontas para serem validadas.
* **Pronto (Done):** Funcionalidades concluídas e prontas para o lançamento.

O objetivo é manter o fluxo contínuo e limitar o trabalho em andamento (WIP - Work In Progress) para evitar a sobrecarga da equipe.

**3. Eliminar Desperdícios (Mudas)**

A filosofia Lean se baseia em eliminar oito tipos de desperdícios. No contexto do TechFit, isso se traduziria em:

* **Desperdício de Funcionalidade:** Em vez de construir todo o sistema de uma vez, a equipe começa com um **Produto Mínimo Viável (MVP)**. O MVP inicial do TechFit poderia ser apenas as telas de login/cadastro e a funcionalidade de agendamento de avaliação física, pois isso é a primeira coisa que um novo aluno precisa fazer. A integração com biometria ou o chat de suporte podem vir depois.
* **Desperdício de Espera:** Em vez de esperar meses para lançar tudo, a equipe lança pequenas partes do sistema (como o MVP) e coleta feedback dos primeiros usuários. Isso evita que o time gaste tempo construindo algo que os clientes não precisam.
* **Desperdício de Movimento:** A equipe deve ter todas as informações e ferramentas necessárias para trabalhar, eliminando a necessidade de buscar por detalhes ou aprovações constantes. A comunicação deve ser direta e eficiente.

**4. Construir, Medir e Aprender (Build-Measure-Learn)**

O ciclo Lean é um processo contínuo de aprendizado. Para o TechFit, isso funcionaria assim:

1. **Construir (Build):** A equipe desenvolve o MVP (Login, Cadastro e Agendamento de Avaliação).
2. **Medir (Measure):** Eles lançam o MVP para um grupo pequeno de usuários e coletam dados. As métricas podem incluir:
   * Tempo que o usuário leva para completar o cadastro.
   * Taxa de sucesso do agendamento.
   * Pontos em que os usuários desistem do processo.
3. **Aprender (Learn):** A equipe analisa os dados. Eles podem descobrir que o formulário de cadastro é muito longo e causa desistência. Com base nesse aprendizado, eles decidem refatorar a tela de cadastro para torná-la mais simples na próxima iteração.

Esse ciclo se repete para cada nova funcionalidade (agendamento de aulas em grupo, notificação para professores, etc.), garantindo que o produto evolua com base em evidências reais e no que os clientes realmente usam. A metodologia Lean permite que o projeto TechFit se adapte rapidamente, entregando valor de forma eficiente e contínua.

**XP:**

METODOLOGIA XP

Trabalhar em duplas (Pair Programming)

Pair Programming é quando duas pessoas trabalham juntas no mesmo código, uma escreve (Driver) e a outra revisa e dá ideias (Navigator).  
Depois elas trocam de função. Isso ajuda a evitar erros e aumenta a qualidade do software.

Exemplo:

# Driver escreve

def soma (a, b):

return a + b

# Navigator revisa

print (soma(3, 5)) # Verifica se está correto

**Testar antes de jogar (TDD – Test Driven Development)**

Antes de escrever o código do jogo, vocês criam um teste:

"O personagem tem que pular 2 blocos de altura".

Só depois escrevem o código até o teste passar.  
Isso garante que tudo funciona direitinho.

[Escrever Teste] → [Executar Teste] → [falha] → [Escrever Código] → [Executar Teste] → [passou?] → [Refatorar] → volta para [Executar Teste]

**Salvar e juntar tudo sempre (Continuous Integration)**

Cada vez que alguém termina um pedacinho do código, coloca no repositório (tipo GitHub).

O sistema roda os testes sozinho e avisa se quebrou algo.  
Assim, o jogo nunca “quebra de vez”, só melhora aos pouquinhos.

**Mostrar para o jogador logo cedo (Small Releases)**

Em vez de esperar meses para entregar o jogo inteiro, vocês lançam versões pequenas.

Primeira semana → personagem só anda.

Segunda semana → personagem pula.

Terceira semana → aparece um inimigo.

O jogador (cliente) testa e dá opinião logo, evitando perder tempo em algo errado.

**Sempre simples e arrumado (Refatoração)**

Se o código ficou bagunçado, vocês o param e deixam limpo antes de seguir.  
Assim, dá para entender e melhorar o jogo no futuro sem dor de cabeça.

**Jogo do Planejamento (Planning Game)**

**O que é:** Reunião semanal da equipe com o cliente para definir o que será feito na próxima semana.

**Como funciona:**

O cliente apresenta as funcionalidades que deseja.

A equipe decide prioridades e estima o tempo necessário.

Define-se o que vai entrar na próxima iteração (semana).

**Benefício:** Garante que a equipe trabalha no que é mais importante, evitando desperdício e retrabalho.

**Metáfora**

Usar **histórias ou comparações simples** para explicar como o sistema funciona.

Desenvolvedores e clientes usam a mesma linguagem simples.

Ex.: “O jogo funciona como um castelo de LEGO: cada bloco é uma funcionalidade”.

**Benefício:** Cria entendimento comum entre todos, facilitando comunicação e decisões.

**Testes de Aceitação**

Testes feitos **junto com o cliente** para verificar se a funcionalidade atende ao que ele espera.

Cliente descreve o que espera.

Equipe implementa.

Todos testam e validam juntos.

**Benefício:** Confirma que o software realmente atende às necessidades do cliente antes de liberar a funcionalidade.

**Ritmo Sustentável**

Manter um ritmo de trabalho constante, que a equipe consegue manter indefinidamente.

**Como funciona:**

Evitar horas extras contínuas.

Planejar entregas que caibam no tempo da equipe.

**Benefício:** Evita cansaço e estresse, mantém produtividade alta a longo prazo.

**Padronização do Código**

Todos seguem **as mesmas regras de codificação e estilo**.

Nomes de variáveis padronizados.

Formatação consistente.

Regras de organização do código combinadas pelo time.

**Benefício:** Facilita leitura, manutenção e colaboração; qualquer dev consegue mexer em qualquer parte do sistema sem confusão.

**As Diferenças entre Metodologias Ágeis: Kanban, Scrum, XP e Lean**

As metodologias ágeis são frameworks que ajudam equipes a entregar valor ao cliente de forma rápida e eficiente, adaptando-se a mudanças. Embora compartilhem a filosofia ágil, **Kanban, Scrum, XP e Lean** se diferenciam em seus focos e práticas.

**Kanban**

O Kanban é um sistema de **fluxo visual** focado em otimizar o processo de trabalho. Seu objetivo principal é a **melhoria contínua** e a estabilidade do fluxo de entrega.

* **Foco:** Otimização do fluxo de trabalho.
* **Papéis:** Não possui papéis formais definidos (como Product Owner ou Scrum Master), embora equipes geralmente tenham gerentes de projeto ou líderes de fluxo.
* **Eventos:** Não tem reuniões fixas. A comunicação e a sincronização são contínuas, geralmente com reuniões rápidas e informais.
* **Artefatos:** O principal artefato é o **quadro Kanban**, que visualiza as tarefas e seu status (Para Fazer, Em Andamento, Concluído).
* **Diferencial:** **Limitação do Trabalho em Andamento (WIP - Work In Progress)**. Isso impede que a equipe se sobrecarregue, forçando-a a terminar uma tarefa antes de começar outra. É ideal para equipes que precisam de flexibilidade e prioridades que mudam com frequência.

**Scrum**

O Scrum é um **framework de gestão de projetos** que organiza o trabalho em ciclos curtos e fixos, chamados de **Sprints**. Ele foca em entregar incrementos de produto funcional ao final de cada ciclo.

* **Foco:** Entregar valor em ciclos de tempo fixos.
* **Papéis:** Possui papéis bem definidos: **Product Owner**, **Scrum Master** e o **Time de Desenvolvimento**.
* **Eventos:** Tem uma rotina de eventos obrigatórios: **Sprint Planning**, **Daily Scrum**, **Sprint Review** e **Sprint Retrospective**.
* **Artefatos:** Os principais são o **Product Backlog** (lista de todas as funcionalidades) e o **Sprint Backlog** (funcionalidades da Sprint atual).
* **Diferencial:** Estrutura **rígida e cíclica** (as Sprints). A equipe se compromete com um conjunto de funcionalidades no início da Sprint, e as prioridades não devem mudar até o fim do ciclo.

**XP (Extreme Programming)**

O XP é uma metodologia ágil que se concentra nas **práticas de engenharia de software** para garantir a alta qualidade do código e a capacidade de resposta a mudanças.

* **Foco:** Qualidade técnica e adaptabilidade.
* **Papéis:** Tem papéis como o **Programador**, o **Cliente no Local** (que fica com a equipe) e o **Coach**.
* **Eventos:** Usa iterações curtas (geralmente 1 a 2 semanas) e reuniões informais.
* **Práticas:** Suas práticas são o diferencial: **Programação em Par**, **Desenvolvimento Orientado a Testes (TDD)**, **Integração Contínua** e **Refatoração**.
* **Diferencial:** É muito mais **prescritivo** em como o código deve ser escrito e testado. Muitas equipes combinam a estrutura de gestão do **Scrum** com as práticas de engenharia do **XP** para obter o melhor dos dois mundos.

**Lean**

A metodologia Lean, que teve origem no sistema de produção da Toyota, é uma filosofia que visa **maximizar o valor e minimizar o desperdício**. No desenvolvimento de software, isso significa construir apenas o que é realmente necessário e entregar de forma rápida.

* **Foco:** Eliminação de desperdícios.
* **Princípios:** Baseia-se em princípios como "eliminar desperdícios", "entregar rápido", "aprender continuamente" e "respeitar as pessoas".
* **Práticas:** Não tem práticas fixas como o Scrum ou o XP. Usa ferramentas como o **mapeamento de fluxo de valor** e o ciclo **Construir-Medir-Aprender** para validar ideias e evitar o desenvolvimento de funcionalidades que ninguém usará.
* **Diferencial:** É mais uma **filosofia** do que um conjunto de regras. Kanban, por exemplo, é uma técnica que pode ser usada para implementar a filosofia Lean. O foco é sempre em entregar o **Produto Mínimo Viável (MVP)** e evoluir com base no feedback real.

**Resumo das Diferenças**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Característica | Kanban | Scrum | XP | Lean |
| Foco Principal | Fluxo de Trabalho | Gestão de Projetos | Qualidade de Código | Eliminação de Desperdícios |
| Ciclos | Contínuo, sem ciclos fixos | Sprints (ciclos fixos) | Iterações curtas | Contínuo (Construir-Medir-Aprender) |
| Papéis | Não tem papéis formais | Product Owner, Scrum Master, Time | Programador, Cliente no Local | Não tem papéis formais |
| Ponto Forte | Flexibilidade e controle de fluxo | Rotina previsível e organizada | Qualidade técnica do produto | Eficiência e foco no valor |

Cada metodologia tem suas vantagens e pode ser mais adequada para diferentes tipos de projetos e equipes.