

# Processos de revisões

**Disciplina:** Qualidade de *Software*



## Conteúdos:

Processos de revisões.

## Habilidade(s):

- Aplicar medidas corretivas em sistemas;
- Como gerenciar projetos utilizando o Manifesto Ágil.

# Bloco 1

---

Conhecendo os principais valores e princípios do  
Manifesto Ágil.

# O que você faria?

Observe a situação abaixo e indique qual é a decisão mais ética e moral para lidar com o caso.

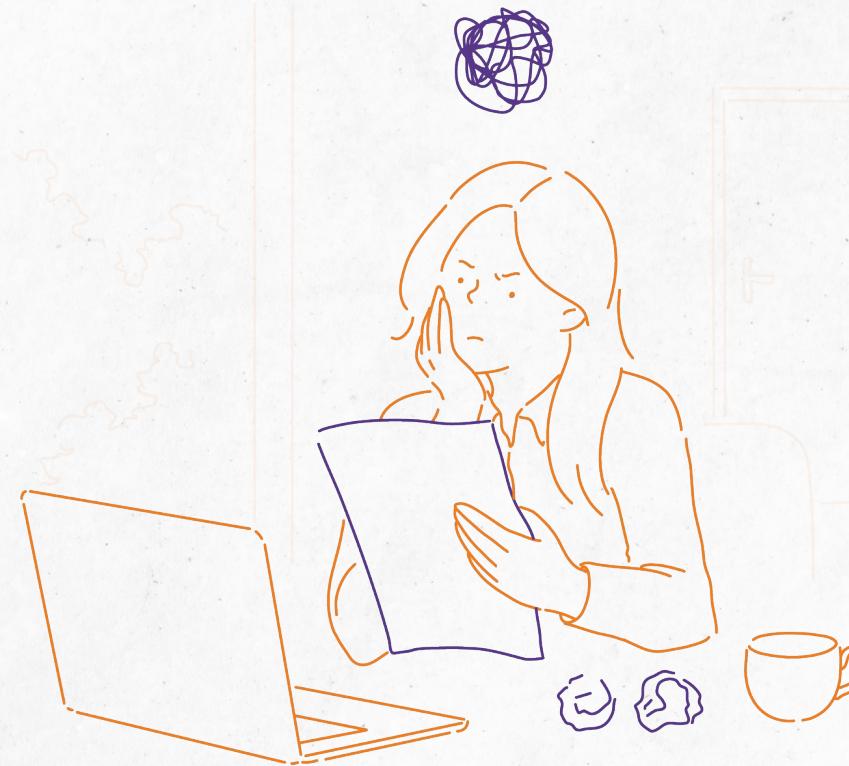
O que você faria se você descobrisse que um colega de equipe está compartilhando informações confidenciais da empresa com os concorrentes?



# O que você faria?

Observe a situação abaixo e indique qual é a decisão mais ética e moral para lidar com o caso.

O que você faria se percebesse que um colega de equipe está assumindo uma carga de trabalho desproporcional em comparação aos outros membros?



## O que você faria?

Observe a situação abaixo e indique qual é a decisão mais **ética** e **moral** para lidar com o caso.

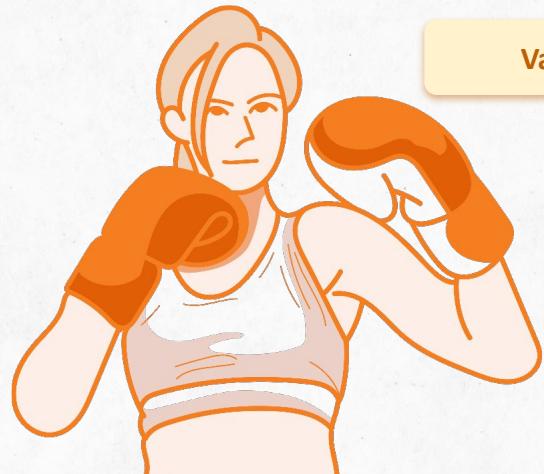
O que você faria se notasse que alguns membros estão excluindo informações vitais dos demais, visando obter vantagem?



# Valores e princípios

Diariamente, enfrentamos situações que demandam julgamentos morais e éticos. Quando isso acontece, precisamos avaliar o que consideramos **certo e errado**.

Geralmente, isso vai de encontro com os nossos **valores e princípios**. Você sabe qual é a diferença entre eles?



VS

Princípios



# Princípios e valores: gabarito

E aí, acertou?

## Valores

São crenças que guiam as nossas preferências e os nossos julgamentos sobre o que é certo e errado, bom e ruim.

Um exemplo é alguém que valoriza a “honestidade” como valor.

VS

## Princípios

São regras ou diretrizes gerais que derivam dos nossos valores e que funcionam como “códigos de conduta”.

Um exemplo disso é alguém que valoriza a “justiça”. Assim, o seu princípio é **não avaliar as pessoas injustamente**.

# Valores e princípios na criação de um *software*

Ter valores e princípios durante o desenvolvimento de *software* é crucial no processo da sua criação e também no seu resultado.

Por isso, em 2000, foi criado o Manifesto Ágil pelos líderes da comunidade do Extreme Programming (XP). Ele é baseado em:

**4 valores**

**12 princípios**

# Valores do Manifesto Ágil

Eles servem como base e, além disso, influenciam nos princípios e nas práticas utilizados pelas equipes ágeis para desenvolver produtos de maneira mais eficaz e adaptativa.

1

**Indivíduos e iterações mais que processos e ferramentas.** Prioriza o trabalho colaborativo e a comunicação eficaz entre as pessoas envolvidas no projeto em vez da produção de trabalho.

2

**Software em funcionamento mais que documentação abrangente.** Valoriza a entrega de *software* funcional, não a produção excessiva de documentação.

3

**Colaboração com o cliente mais que negociação de contratos.** Enfatiza a importância de envolver os clientes de forma contínua e adaptativa.

4

**Responder a mudanças mais que seguir um plano.** Reconhece que a flexibilidade é crucial para lidar com a incerteza e adaptar o projeto conforme as necessidades.

# Princípios do Manifesto Ágil

A partir desses princípios, é possível promover uma abordagem focada na entrega de valor ao cliente.

Nossa maior prioridade é satisfazer o cliente por meio da entrega contínua e adiantada;

*Software* funcionando é a medida primária de progresso;

Processos ágeis se adaptam às mudanças;

Processos ágeis promovem desenvolvimento sustentável;

Entregar frequentemente *softwares* funcionando, dando preferência à menor escala de tempo;

Atenção à excelência técnica e bom *design* aumentam a agilidade;

# Princípios do Manifesto Ágil

A partir desses princípios, é possível promover uma abordagem focada na entrega de valor ao cliente.

Pessoas de negócio e desenvolvedores devem trabalhar diariamente e em conjunto por todo o projeto;

Construir projetos em torno de indivíduos motivados;

O método mais eficiente e eficaz de transmitir informações é por meio da conversa face à face;

Simplicidade: a arte de maximizar a quantidade de trabalho não realizado é essencial;

As melhores arquiteturas, requisitos e *designs* emergem de times auto-organizáveis;

Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e, então, refina o seu comportamento.

# Bloco 2

---

Entendendo o funcionamento do *framework* Scrum.

## Vamos relembrar?

No bloco anterior, você conheceu os quatro principais valores do Manifesto Ágil. Analise as descrições abaixo e tente identificar a qual dos princípios cada uma se refere.

A) No meio do projeto, ocorre uma mudança significativa nas preferências do cliente, exigindo que a equipe reavalie a direção do desenvolvimento.



B) Em vez de criar uma documentação extensa para cada recurso do *software*, a equipe concentra os seus esforços em criar versões funcionais.



C) Durante uma reunião de planejamento, a equipe decide ajustar os requisitos do projeto para acomodar uma nova prioridade do cliente.



D) A equipe trabalha em estreita colaboração com os representantes do cliente, realizando reuniões regulares para obter *feedbacks*.



# Gabarito

E aí, acertou?

A) No meio do projeto, ocorre uma mudança significativa nas preferências do cliente, exigindo que a equipe reavalie a direção do desenvolvimento.

B) Em vez de criar uma documentação extensa para cada recurso do software, a equipe concentra os seus esforços em criar versões funcionais.

C) Durante uma reunião de planejamento, a equipe decide ajustar os requisitos do projeto para acomodar uma nova prioridade do cliente.

D) A equipe trabalha em estreita colaboração com os representantes do cliente, realizando reuniões regulares para obter *feedbacks*.

**Responder às mudanças**

**Software em funcionamento**

**Indivíduos e iterações**

**Colaboração com o cliente**

## Scrum *framework*

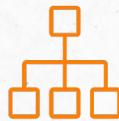
É um *framework* utilizado para gerenciar projetos complexos por meio de **processos dinâmicos, papéis e cerimônias**.

Ele foi criado na década de 1990 e foi inspirado por uma jogada de rúgbi, esporte no qual jogadores se reúnem para remover obstáculos e resolver problemas.



# Principais pilares do Scrum

São um conjunto de princípios fundamentais que sustentam a estrutura e a filosofia desse *framework*.



## Transparência

Todos possuem o conhecimento dos requisitos, dos processos e do andamento do projeto.



## Inspeção

Durante todo o tempo, o que está sendo feito é inspecionado. Isso acontece por meio de reuniões diárias ou no momento de revisão.



## Adaptação

O processo e o produto podem sofrer adaptações, mas precisam preservar os seus valores.

# Valores do Scrum

Assim como no Manifesto Ágil, o Scrum também possui os seus próprios valores.



**Foco:** todos estão focados em um único objetivo;



**Coragem:** é importante para enfrentar desafios;



**Comprometimento:** cada membro possui uma responsabilidade específica;



**Respeito:** é necessário que os membros do time respeitem uns aos outros;



**Abertura:** é essencial para garantir oportunidades.

# Princípios do Scrum

Completam os valores e são fundamentais para o bom funcionamento do *framework*.



**Controle dos processos empíricos:** as decisões são baseadas na observação e nos experimentos;



**Priorização baseada em valor:** a priorização feita considera valor, risco, incerteza e dependências;



**Auto-organização:** os times são auto-organizáveis e possuem a responsabilidade compartilhada;



**Time-boxing:** propõe a fixação de um determinado tempo para cada processo;



**Colaboração:** o trabalho colaborativo defende um processo de criação de valor compartilhado;



**Desenvolvimento iterativo:** permite que os times aprendam com o decorrer dos ciclos.

# Papéis fundamentais no Scrum

Sem essas funções, não é necessário realizar uma abordagem confiável e eficaz.



**Product owner (PO)**

Dono do produto e o responsável por decidir as suas funcionalidades.



**Scrum master (SM)**

Líder do processo. Ajuda a entender os valores e os princípios da cultura ágil.



**Time Scrum**

Time de desenvolvimento, composto por vários membros.

# Como é realizado o planejamento de *software*?

É baseado em cerimônias e artefatos.

## Cerimônias

***Product backlog:*** lista de funcionalidades priorizadas.

***Sprint backlog:*** lista de tarefas para cada *sprint*.

***Incremento:*** versão do aplicativo com novas funcionalidades a cada *sprint*.

## Artefatos

***Sprint planning:*** definir objetivos e tarefas.

***Daily Scrum:*** reuniões diárias para alinhamento.

***Sprint review:*** avaliação das funcionalidades entregues.

***Sprint retrospective:*** análise das práticas do Scrum.

# Como é realizado o planejamento de *software*



O *product owner* descreve como quer o produto.

A partir disso, são levantadas atividades que serão distribuídas em *sprints* (cada uma com duração de duas a quatro semanas).

## Planning tático

Antes de iniciar, o *product owner* realiza uma *sprint planning* para determinar qual o objetivo das *sprints*.

Durante a reunião, o *product owner*, o *Scrum master* e time Scrum entra em consenso sobre as *sprints*.

## Planning estratégico

Com a *sprint* definida, o time Scrum se reúne para discutir as atividades e negociar os prazos estabelecidos.

## Andamento do projeto

Aqui, devem ser realizadas as *daily Scrum*, reuniões diárias utilizadas para acompanhar o progresso do que está sendo construído. Três perguntas devem ser feitas:

O que eu fiz ontem que contribuiu para alcançar a meta do *sprint*?

O que eu farei hoje para contribuir com o alcance da meta do *sprint*?

Existe alguma circunstância que impeça a mim ou ao time de atingir a meta do *sprint*?



## O começo do fim

Ao se aproximar do fim das atividades, é importante usar o *Sprint review* com o intuito de verificar e adaptar o produto que está sendo desenvolvido.

Aqui, é possível observar se há a necessidade de alguma adaptação do produto.



# Finalização das atividades

Para finalizar, é realizado o *Sprint retrospective*, visando analisar se há a necessidade de melhoria no processo como um todo.

Alguns aspectos podem ser observados, como:

pontos positivos

pontos negativos



# Bloco 3

---

Vamos praticar tudo que aprendemos sobre o  
*framework Scrum!*

# Vamos realizar uma dinâmica?

Separem-se em grupos e analisem o caso a seguir.

Você faz parte de um time responsável por desenvolver um aplicativo de gerenciamento de tarefas para equipes de estudantes. O aplicativo deve permitir que os usuários criem, atribuam e acompanhem tarefas, além de facilitar a comunicação e a colaboração entre os membros da equipe. A interface também deve ser intuitiva e amigável.

Com base nas informações presentes na folha de ofício que cada grupo recebeu, construam um roteiro de projeto baseado em um modelo Scrum. Nele, deverá conter **prazo, cronograma, equipe de desenvolvimento, artefatos e cerimônias**.



# Bloco 4

---

Agora, chegou a hora de aprender um pouco mais  
sobre o Extreme Programming.

# Duas verdades e uma mentira

Duas pessoas estão dizendo a verdade e uma está mentindo. Identifique-as com base nas descrições.



Eu sou a Alice e sou a **product owner**.  
Sou a responsável por definir as  
necessidades do cliente.



Olá, eu sou a Carol e sou do **time Scrum**. Trabalhamos juntos para  
desenvolver o produto.



Meu nome é Roberto e eu sou o **Scrum master**. Minha maior função é definir  
os *sprints* do projeto.

# Gabarito

E aí, acertou?



Verdade!



Verdade!



Mentira! Na verdade, quem costuma definir os *sprints* é o *product owner*!

# Extreme Programming (XP)

É uma metodologia de desenvolvimento de *software* leve e não prescritiva. Foi criada em 1997.

Ela enfatiza a agilidade e a qualidade, levando as boas práticas da engenharia de *software* com bastante seriedade.



# Valores do Extreme Programming

Assim como o Scrum, o XP se baseia em valores e práticas.



**Comunicação:** preza pela comunicação constante;



**Simplicidade:** prioriza fazer tudo da forma mais simples possível;



**Feedback:** busca conseguir a devolutiva o mais rápido possível com o auxílio de testes automatizados;



**Coragem:** é necessário para dar e receber *feedbacks*;



**Respeito:** é o principal pilar da comunicação.

# Práticas do Extreme Programming

**Propriedade coletiva:** o código não tem dono, logo, todos podem modificá-lo.

**Padronização de código:** define regras de programação para manter a padronização.

**Versões pequenas:** libera versões funcionais para a aceitação do cliente.

**Planejando o jogo:** reuniões com o cliente para priorizar funcionalidades.

**Testes de aceitação:** o cliente e o time criam testes de aceitação para requisitos.

**Programação pareada:** desenvolvimento em duplas, compartilhando o conhecimento.

**Desenvolvimento orientado a testes (TDD):** cria testes antes do código.

**Integração contínua:** integra funcionalidades frequentemente.

**Design simples:** mantém o código simples para facilitar o entendimento.

**Refatoração:** melhora o código continuamente.

**Metáforas:** usa metáforas para facilitar a comunicação com o cliente.

**Semana de 40 horas:** trabalho com qualidade por 40 horas semanais.

# Bloco 5

---

Vamos praticar o que aprendemos sobre o  
Extreme Programming?

# Vamos aprender um pouco mais sobre valores?

Separem-se em cinco grupos.

Cada equipe receberá um dos temas a seguir:

Comunicação

*Feedback*

Respeito

Simplicidade

Coragem

A partir deles, utilizem cartolinhas e canetas coloridas para responder: “**o que essa prática tem a ver com desenvolvimento de software?**”.

- Cada grupo terá **vinte minutos** para realizar a atividade;
- As equipes terão **cinco minutos** para apresentar o que foi feito.

# Bloco 6

---

Hora de descontrair.

## Vamos sortear?

Nessa atividade, a dinâmica será um pouco diferente.

O(A) professor(a) deverá sortear **cinco alunos** aleatórios e que deverão formar uma equipe entre si. Os cinco alunos sorteados serão os **representantes** de cada time.

Chegou a hora de sortear!



Acesse o *site* para sortear os alunos de forma rápida e prática!



## Quem sou eu?

Cada equipe deve escolher um grupo e um tema da aula, escrever o tópico escolhido em um *post-it* e colar na testa do um representante de um dos times.

Os cinco alunos poderão fazer uma pergunta. As respostas deve ser apenas de “sim” ou “não”.

Caso a resposta seja “sim”, o representante terá direito a mais uma resposta. Caso seja “não”, ele deverá passar a sua vez.

A dinâmica não terá vencedor, mas terá classificação do primeiro ao quinto lugar. Sendo assim, caso uma pessoa acerte, o jogo deverá continuar até que sobre apenas a última equipe.



# Fechamento

Que bom!

Que pena...

Que tal?

# Referências Bibliográficas

PROZ EDUCAÇÃO. *Qualidade de software*. 2023.