

Matéria Adiantada dia 20/10

Semana 29 - Aula 01

Tópico Principal da Aula: Papéis e Responsabilidades em um Projeto Ágil (Visão Geral)

Subtítulo/Tema Específico: Os Papéis Essenciais (Product Owner, Scrum Master e Time de Desenvolvimento)

Código da aula: [SIS]ANO1C3B4S29A1

Objetivos da Aula:

- Compreender os papéis e responsabilidades fundamentais em um projeto ágil.
- Reconhecer a importância da liderança servidora do Scrum Master e da colaboração do Time de Desenvolvimento.
- Entender o papel estratégico do Product Owner na priorização e gestão do Backlog do Produto.

Recursos Adicionais (Sugestão):

- Caderno para anotações;
- Acesso ao laboratório de informática e/ou internet.

Exposição do Conteúdo:

Referência do Slide: Introdução e o Papel do Product Owner (PO)

- **Definição:** O **Product Owner (PO)**, ou **Dono do Produto**, é o principal responsável por maximizar o valor do produto resultante do trabalho do Time de Desenvolvimento. Ele é o único responsável por gerenciar o **Backlog do Produto**, que é uma lista ordenada de tudo que é necessário no produto.
- **Aprofundamento/Complemento (se necessário):** O PO é o representante do cliente e dos *stakeholders* (interessados). Ele não apenas lista os itens, mas também é o decisor final sobre o conteúdo, ordem e escopo do produto. Um bom PO deve ter conhecimento de negócio, visão de mercado e autoridade para tomar decisões rápidas, garantindo que o time esteja sempre trabalhando no item de maior valor.
- **Exemplo Prático:** Em um projeto de desenvolvimento de um aplicativo de banco, o PO é quem decide se a funcionalidade de "pagamento por QR Code" deve ser desenvolvida antes ou depois da funcionalidade de "emissão de extrato em PDF", com base no valor de negócio e nas necessidades dos usuários.

Referência do Slide: O Papel do Scrum Master (SM)

- **Definição:** O **Scrum Master (SM)** é um líder servidor para o Time Scrum. Sua principal função é garantir que o Scrum seja compreendido e aplicado, além de facilitar o processo e remover impedimentos que possam atrapalhar a produtividade do Time de Desenvolvimento.
- **Aprofundamento/Complemento (se necessário):** O SM atua como um *coach* para o time, ajudando-os a se auto-organizarem e a serem multifuncionais. Ele também

ajuda o PO a refinar o Backlog e a organização a entender e adotar a agilidade. O SM não é um gerente de projeto tradicional nem um chefe.

- **Exemplo Prático:** Se um desenvolvedor do time está bloqueado porque precisa de uma licença de software nova ou porque um servidor de testes parou de funcionar, o Scrum Master intervém para resolver esses problemas, permitindo que o desenvolvedor mantenha o foco no trabalho da Sprint.

Referência do Slide: Time de Desenvolvimento e Patrocinador

- **Definição:** O Time de Desenvolvimento é o grupo de profissionais que entrega um incremento "Pronto" e potencialmente liberável em cada Sprint. A equipe deve ser auto-organizada e multifuncional. O Patrocinador (Sponsor) é o indivíduo ou grupo que fornece recursos e suporte ao projeto, garantindo que ele esteja alinhado com os objetivos estratégicos da organização.
- **Aprofundamento/Complemento (se necessário):** O time de desenvolvimento é cross-funcional, significando que ele possui todas as habilidades necessárias para transformar itens do Backlog em software funcionando, sem depender de pessoas externas. No Scrum, não há títulos de "desenvolvedor sênior" ou "arquiteto", todos são apenas "Desenvolvedores".
- **Exemplo Prático:** O Patrocinador é o executivo que aprova o orçamento e o prazo inicial do projeto. O Time de Desenvolvimento, ao final de uma Sprint de duas semanas, entrega um conjunto de funcionalidades prontas (como o "login de usuário" e a "criação de perfil") que podem ser mostradas e testadas.
- **Links de Vídeo (Papéis Ágeis / Scrum):**
 - [Papéis do Scrum - Scrum e Métodos Ágeis - YouTube](#)
 - [Curso de Scrum: 2.Papéis - YouTube](#)

Semana 29 - Aula 02

Tópico Principal da Aula: Papéis e Responsabilidades em XP (eXtreme Programming)

Subtítulo/Tema Específico: A Equipe XP: Foco na Colaboração e nas Práticas Técnicas

Código da aula: [SIS]ANO1C3B4S29A2

Objetivos da Aula:

- Compreender os papéis e responsabilidades específicos da metodologia eXtreme Programming (XP)ANO1C3B4S29A2.pdf - snippetFromFront].
- Reconhecer o papel crucial do Cliente como parte integrante da equipe de desenvolvimento.
- Entender como os papéis do Coach e do Tracker garantem a aplicação correta e o progresso da metodologia.

Recursos Adicionais (Sugestão):

- Caderno para anotações;
- Acesso ao laboratório de informática e/ou internet.

Exposição do Conteúdo:

Referência do Slide: O Cliente e a Parceria (Customer)

- **Definição:** Em XP, o **Cliente (Customer)** é um membro da equipe responsável por escrever as histórias de usuário (*User Stories*), definir prioridades e determinar os critérios de aceitação. A presença constante do Cliente ou de um *Cliente Proxy* é fundamental para sanar dúvidas e garantir que o produto construído atenda às necessidades do negócio.
- **Aprofundamento/Complemento (se necessário):** O XP defende a comunicação **face a face** e o *Feedback* constante. A inclusão do Cliente na equipe minimiza especulações e mal-entendidos, agilizando o ciclo de desenvolvimento. O Cliente também auxilia na realização dos testes de aceitação.
- **Exemplo Prático:** O Cliente se senta com os Programadores para revisar uma *User Story* que descreve como o usuário deve se cadastrar. Ele esclarece de imediato se a validação do CPF deve ocorrer antes ou depois do envio do formulário, garantindo que não haja trabalho desperdiçado.

Referência do Slide: Programador, Tester e Tracker

- **Definição:** Os **Programadores** ocupam o principal papel, sendo responsáveis por estimar as tarefas, realizar a **Programação em Par** e aplicar as práticas técnicas do XP (como TDD e Integração Contínua). O **Tester** ajuda o Cliente a definir e executar os testes de aceitação. O **Tracker** é responsável por medir o progresso do projeto, monitorar as métricas e reportar o *status* da equipe.
- **Aprofundamento/Complemento (se necessário):** O XP valoriza a **Propriedade Coletiva do Código**, onde todos os programadores são responsáveis por todo o código, incentivando a revisão e a qualidade. O Tracker, ou Rastreador, provê informações sobre o andamento e identifica pontos que precisam ser melhorados, muitas vezes elaborando **Radiadores de Informação** (como gráficos e painéis de progresso).
- **Exemplo Prático:** Dois programadores trabalham juntos em um mesmo computador (*Pair Programming*) para desenvolver uma funcionalidade. O Tracker, ao perceber que o tempo estimado para as tarefas está consistentemente incorreto, levanta essa questão na reunião de retrospectiva para que a equipe possa melhorar suas estimativas.

Referência do Slide: Coach e o Foco no Processo

- **Definição:** O **Coach (Treinador)** é geralmente o programador mais experiente. Ele se preocupa com a execução técnica e a evolução do processo, garantindo que a equipe utilize as práticas de XP de forma correta e consistente (como Simplicidade e *Refactoring*). Ele atua como um mentor e facilitador.
- **Aprofundamento/Complemento (se necessário):** O Coach não toma decisões técnicas, mas garante que o time chegue às melhores decisões por meio das práticas ágeis. Ele deve ser um bom comunicador e ter profundo conhecimento técnico. O XP possui outros papéis complementares, como o *Interaction Designer*, o *Business Analyst* e o *Redator Técnico* (que foca na documentação).

- **Exemplo Prático:** O Coach observa que a equipe está criando soluções complexas demais (*over-engineering*). Ele intervém, questionando: "Qual é a coisa mais simples que funciona?", reforçando o valor da **Simplicidade** em XP e guiando o time para uma solução mais direta.
- **Links de Vídeo (eXtreme Programming - XP):**
 - [Extreme Programming — Metodologia XP: o que é? como funciona? - YouTube](#)
 - [O Que É eXtreme Programming - YouTube](#)

Semana 29 - Aula 03

Tópico Principal da Aula: Papéis e Responsabilidades no Scrum (Aprofundamento)

Subtítulo/Tema Específico: O Product Owner e o Development Team em Detalhes

Código da aula: [SIS]ANO1C3B4S29A3

Objetivos da Aula:

- Compreender os papéis e responsabilidades específicos dentro do *framework* ScrumANO1C3B4S29A3.pdf - snippetFromFront].
- Detalhar as funções do Product Owner na gestão da visão do produto e do Backlog.
- Analisar a auto-organização e a multifuncionalidade do Development Team.

Recursos Adicionais (Sugestão):

- Caderno para anotações;
- Acesso ao laboratório de informática e/ou internet.

Exposição do Conteúdo:

Referência do Slide: O Product Owner (PO): Visão e Priorização

- **Definição:** O PO é o responsável por toda e qualquer decisão sobre o **Backlog do Produto**, atuando como a única voz do cliente para o Time de Desenvolvimento. Suas responsabilidades incluem expressar claramente os itens do Backlog, ordená-los para maximizar o valor e garantir que o Backlog seja transparente e visível para todos.
- **Aprofundamento/Complemento (se necessário):** O PO deve estar envolvido nas atividades de **Refinamento do Backlog**, que consistem em adicionar detalhes, estimativas e ordenação aos itens para que fiquem "prontos" para a próxima Sprint. Seus artefatos principais são o Backlog do Produto e a **Meta do Produto (Product Goal)**.
- **Exemplo Prático:** O PO realiza uma reunião de refinamento com o time para decompor uma *feature* grande (Épico) em tarefas menores (*User Stories*). Por exemplo, a *feature* "Nova Área de Perfil" é dividida em "Login de Usuário", "Edição de Dados Pessoais" e "Visualização de Histórico de Compras", e o PO define a ordem de desenvolvimento.

Referência do Slide: O Time de Desenvolvimento (Development Team)

- **Definição:** O Time de Desenvolvimento é composto por profissionais que transformam os itens do Backlog em um incremento de produto funcional. Eles são auto-organizados, decidindo como realizar o trabalho e gerenciar a si mesmos para entregar o resultado da Sprint [ANO1C3B4S29A3.pdf - snippetFromBack].
- **Aprofundamento/Complemento (se necessário):** No Scrum, o time é responsável pela criação do **Plano da Sprint** e pela adesão à **Definição de Pronto (*Definition of Done* - DoD)**. O tamanho ideal da equipe de desenvolvimento é de 3 a 9 membros para otimizar a comunicação e a produtividade.
- **Exemplo Prático:** Durante a **Reunião de Planejamento da Sprint**, o Time de Desenvolvimento seleciona os itens do Backlog que pode completar. Eles, e somente eles, definem o volume de trabalho e como o farão, por exemplo, decidindo usar *Pair Programming* ou dividir as tarefas de *backend* e *frontend* entre os membros.

Referência do Slide: O Scrum Master (Revisão)

- **Definição:** Embora já apresentado, é crucial reforçar que o Scrum Master complementa o time, garantindo a adesão ao Scrum e removendo impedimentos. Ele treina, orienta e faz o *coaching* do Time Scrum.
- **Aprofundamento/Complemento (se necessário):** As responsabilidades do SM incluem facilitar os Eventos Scrum (Planejamento, Daily, Revisão e Retrospectiva) e proteger o time de interferências externas. Seu sucesso é medido pela melhoria contínua da eficácia do Time Scrum.
- **Exemplo Prático:** Na *Daily Scrum* (Reunião Diária), se o SM percebe que o time está focando em discutir soluções técnicas em vez de alinhar o progresso para a Meta da Sprint, ele intercede para redirecionar a conversa e reforçar o propósito do evento.
- **Links de Vídeo (Scrum: Foco nos Papéis):**
 - [O Que Faz Um Scrum Master? Dia a Dia Completo em Uma Sprint - YouTube](#)
 - [Esse é o papel de um PO! - YouTube](#)