Semana 26 - Aula 1

Tópico Principal da Aula: Metodologias Ágeis Mais Utilizadas na Indústria de DS

Subtítulo/Tema Específico: Extreme Programming (XP) e a Transformação Ágil

Código da aula: [SIS]ANO1C3B4S26A1ANO1C3B4S26A1.pdf]

Objetivos da Aula:

- Compreender a metodologia ágil **XP (Extreme Programming).
- Reconhecer a mudança de paradigma do desenvolvimento tradicional para o ágil, iniciado pelo **Manifesto Ágil os Adicionais (Sugestão, pode ser adaptado):**
- Caderno para anotações;
- Acesso ao laboratório de informática e/ou internet.

Exposição do Conteúdo:

Referência do Slide: Introdução: O Manifesto Ágil e o Desafio às Crenças Tradicionais

- **Definição:** **O Manifesto Ágil** surgiu como uma reação aos métodos tradicionais, como o Cascata (Waterfall), que se mostravam lentos, inflexíveis e ineficientes diante de requisitos. O princípio central do Manifesto é **"Responder à mudança mais do que seguir um plano".
- **Aprofundamento/Complemento (se necessário):** No modelo Cascata, a fase de planejamento deve ser concluída antes de iniciar a execução (codificação). O Manifesto Ágil inverte essa prioridade, focando em entregas menores e frequentes de software funcional, permitindo que a equipe incorpore o *feedback* do cliente ao longo de todo o projeto.
- **Exemplo Prático:** Uma empresa usava o modelo Cascata e o cliente só via o produto final após 10 meses. Com a adoção do Ágil, o cliente recebe uma nova versão a cada 2 ou 3 semanas, podendo fornecer *feedback* imediato e direcionar o desenvolvimento.
- **Link de Vídeo:**
    - Agile vs. Waterfall - Uma análise de diferenças (YouTube)

Referência do Slide: Construindo o Conceito: Extreme Programming (XP)

- **Definição:** O **Extreme Programming (XP)** é uma metodologia ágil criada por **Kent Beck** que se destaca por aplicar "ao extremo" boas práticas de engenharia de software. Seu objetivo é a entrega de *software* de **alta qualidade**, adaptável e que satisfaça o cliente. Os valores do XP são: **Comunicação, Simplicidade, Feedback, Coragem e Respeito**.
- **Aprofundamento/Complemento (se necessário):** O XP é ideal para projetos com requisitos que mudam constantemente e que exigem um alto grau de qualidade técnica, sendo a metodologia ágil mais focada em práticas de programação.
- **Exemplo Prático:** Uma *startup* desenvolvendo um produto inovador onde o mercado e os requisitos mudam a cada semana. O XP se encaixa perfeitamente,

pois suas práticas técnicas permitem que o código seja adaptado rapidamente sem perder a qualidade e a capacidade de manutenção.

- **Link de Vídeo:**
  - Extreme Programming — Metodologia XP: o que é? como funciona? (YouTube)

Referência do Slide: Práticas-Chave do XP

- **Definição:** O XP se apoia em um conjunto de práticas técnicas e de gestão para alcançar seus objetivos:
  - **Desenvolvimento Orientado a Testes (TDD):** Escrever primeiro o teste automatizado (que falha) antes de escrever o código de produção.
  - **Programação em Pares (Pair Programming):** Dois desenvolvedores trabalham juntos em um único computador.
  - **Integração Contínua (CI):** Integrar e testar o código várias vezes ao dia.
  - **Refatoração Contínua:** Melhorar o *design* interno do código, ajustando-o para ser mais limpo e eficiente, sem alterar seu comportamento externo.
  - **Cliente no Local:** O representante do cliente trabalha junto com o time de desenvolvimento, fornecendo *feedback* imediato e esclarecendo dúvidas.
- **Aprofundamento/Complemento (se necessário):** A **Programação em Pares** é uma forma poderosa de **revisão de código** e **transferência de conhecimento**, garantindo que o conhecimento sobre o sistema não fique centralizado em uma única pessoa.
- **Exemplo Prático:** Durante a **Programação em Pares**, um desenvolvedor mais experiente ensina uma técnica de *design* para o colega novato, enquanto ambos revisam o código em tempo real. A **Refatoração Contínua** permite que o time mantenha o **Design Simples**, ajustando a arquitetura apenas quando necessário e não de forma antecipada.
- **Link de Vídeo:**
  - XP Programming (COMECE POR AQUI) // Dicionário do Programador (YouTube)

Semana 26 - Aula 2

Tópico Principal da Aula: Metodologias Ágeis Mais Utilizadas na Indústria de DS

Subtítulo/Tema Específico: Introdução e Princípios de Scrum e Kanban

Código da aula: [SIS]ANO1C3B4S26A2ANO1C3B4S26A2.pdf]

Objetivos da Aula:

- Aprender sobre as metodologias ágeis **Scrum** e **Kanban**.
- Entender os princípios do **Empirismo** (Transparência, Inspeção e Adaptação) que guiam as metodologias ágeis.

Recursos Adicionais (Sugestão, pode ser adaptado):

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

Exposição do Conteúdo:

Referência do Slide: Ponto de Partida e Pilares da Agilidade

- **Definição:** Scrum e Kanban são as metodologias ágeis mais utilizadas na indústria. Ambas são baseadas no **Empirismo**, que postula que o conhecimento vem da experiência e da tomada de decisões com base no que é observado. O Empirismo é sustentado por três pilares:
  - **Transparência:** O processo e o trabalho devem ser visíveis para todos.
  - **Inspeção:** Os artefatos e o progresso devem ser inspecionados regularmente.
  - **Adaptação:** Desvios indesejados devem ser corrigidos o mais rápido possível.
- **Aprofundamento/Complemento (se necessário):** Os pilares do Empirismo garantem que a equipe possa aprender rapidamente com a execução e ajustar o curso do projeto em vez de seguir um plano rígido que pode estar desatualizado.
- **Exemplo Prático:** A **Retrospectiva** no Scrum é um momento de **Inspeção** e **Adaptação**, onde a equipe analisa o que deu certo/errado no ciclo anterior e cria um plano de melhoria para o próximo ciclo.
- **Link de Vídeo:**
  - Fundamentos do Scrum e Kanban: O que são e como funcionam (YouTube)

Referência do Slide: Construindo o Conceito: Scrum (Framework)

- **Definição:** O **Scrum** é um **framework** que fornece um conjunto de regras, papéis (Product Owner, Scrum Master, Time de Desenvolvimento), artefatos e eventos (**cerimônias**) para gerenciar o desenvolvimento de produtos complexos. Sua principal característica é o **Sprint**, um ciclo de duração fixa (geralmente 1 a 4 semanas) no qual um incremento de produto funcional é criado.
- **Aprofundamento/Complemento (se necessário):** Os eventos do Scrum (Planejamento, Daily Scrum, Revisão e Retrospectiva) são *time-boxed* (têm duração máxima), garantindo a disciplina e a frequência de Inspeção e Adaptação. A *Daily Scrum* é a reunião mais curta (15 minutos) e visa inspecionar o progresso da Sprint.
- **Exemplo Prático:** A equipe realiza uma **Revisão da Sprint** ao final do ciclo para demonstrar o incremento concluído ao *Product Owner* e *stakeholders*. Esse é o momento formal para coletar *feedback* (Inspeção) e decidir o que priorizar na próxima Sprint (Adaptação).
- **Link de Vídeo:** https://youtu.be/3aCww_1RnL0?si=RL7sSdTmhP2-pJHb
-
- **Link de Vídeo:**
  - SCRUM em 5 minutos - O método do Google e da NASA para fazer mais em menos tempo (YouTube)

Referência do Slide: Construindo o Conceito: Kanban (Método)

- **Definição:** O **Kanban** é um **método** que tem como foco principal o gerenciamento e a melhoria do **fluxo de trabalho contínuo**. Suas práticas-chave são:
    - **Visualizar o Fluxo:** Usar um quadro (*Kanban Board*) para mapear as etapas do trabalho.
    - **Limitar o Trabalho em Progresso (WIP):** Definir um número máximo de tarefas em andamento.
    - **Gerenciar o Fluxo:** Medir o tempo de ciclo e otimizar o processo.
    - **Explicitar Políticas:** Definir regras claras de como o trabalho flui entre as colunas.
    - **Melhoria Colaborativa:** Usar modelos e o método científico para evoluir.
- **Aprofundamento/Complemento (se necessário):** O **limite de WIP** é a prática mais transformadora do Kanban, pois impede que a equipe inicie mais trabalho do que consegue finalizar, reduzindo a sobrecarga e o tempo que as tarefas levam para serem concluídas (Lead Time).
- **Exemplo Prático:** O time de suporte usa um Kanban. A coluna "Em Resolução" tem um WIP de 3. Um técnico só pode puxar um novo *ticket* para essa coluna se o número de *tickets* nela for inferior a 3. Isso garante que o foco seja na rápida resolução dos 3 *tickets* atuais.
- **Link de Vídeo:**
    - O que é Kanban e como ele funciona (Explicação Simples) (YouTube)

Semana 26 - Aula 3

Tópico Principal da Aula: Metodologias Ágeis Mais Utilizadas na Indústria de DS

Subtítulo/Tema Específico: Comparativo: Scrum vs. Kanban

Código da aula: [SIS]ANO1C3B4S26A3ANO1C3B4S26A3.pdf]

Objetivos da Aula:

- Compreender as diferenças fundamentais entre **Scrum** e **Kanban**.
- Saber em quais contextos cada metodologia se aplica melhor.

Recursos Adicionais (Sugestão, pode ser adaptado):

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

Exposição do Conteúdo:

Referência do Slide: Diferenças Principais e Aplicações

- Definição: A distinção central está na natureza: Scrum é um framework (prescritivo), e Kanban é um método (evolutivo).
  | Característica | Scrum | Kanban |

Ciclo de Tempo	Sprints de duração fixa (Time-boxed)ANO1C3B4S26A3.pdf]	Fluxo Contínuo, sem intervalos fixos
Papéis	Prescritivos: Product Owner, Scrum Master e Time de Dev	Não Prescritivos: Adapta-se à estrutura existente
Mudança de Escopo	Desencorajada dentro do SprintANO1C3B4S26A3.pdf]	Aceita a qualquer momento
Métrica de Foco	Velocidade (Velocity)	Lead Time (Tempo de entrega)
- **Aprofundamento/Complemento (se necessário):** Scrum é mais adequado para o **desenvolvimento de produtos** onde a previsibilidade das entregas em blocos de tempo é crucial. Kanban é excelente para a **manutenção, suporte ou processos de fluxo contínuo** onde as prioridades mudam com alta frequência e a meta é reduzir o tempo de espera.
- **Exemplo Prático:** Uma equipe de desenvolvimento de *games* utiliza o Scrum para planejar e desenvolver novos recursos a cada mês (Sprints). Já a equipe de infraestrutura, que lida com incidentes e manutenções emergenciais, usa o Kanban para gerenciar o fluxo de requisições que chegam de forma imprevisível.
- **Link de Vídeo:**
  - Scrum vs Kanban: diferenças explicadas de forma simples (YouTube)

Referência do Slide: Proto-Kanban e Quadros

- **Definição:** O **Proto-Kanban** é uma forma inicial de implementação, onde se utiliza apenas a **visualização do fluxo** em um quadro (físico ou eletrônico) sem aplicar, no início, os limites de WIP (Work in Progress. Isso permite uma adoção mais suave do método, buscando mudanças evolutivas.
- **Aprofundamento/Complemento (se necessário):** A transição para o ágil pode ser difícil. O Kanban, por exigir menos mudanças de papéis e rituais, e permitir começar com o Proto-Kanban, é frequentemente usado por equipes que precisam de uma melhoria gradual e incremental do processo.
- **Exemplo Prático:** Um time usa um quadro Kanban eletrônico (Trello, Jira) para visualizar as colunas "A Fazer", "Em Análise", "Em Dev" e "Concluído". O quadro é o elemento central que, ao se tornar transparente, permite a identificação rápida de *gargalos* (colunas com muitas tarefas).
- **Link de Vídeo:**
  - O melhor método de Gestão ágil: Scrum ou Kanban? (YouTube)