ROTEIRO DE PESQUISA E APRESENTAÇÃO

XP - *EXTREME PROGRAMMING*

YURI, THIAGO, RAPHAEL, VITOR e PLINIO

12/08 – Relatório/Roteiro de pesquisa

19/08 – Checklist

02/09 – Apresentação

Tópicos de pesquisa:

* OQUE É
* APLICAÇÃO

Práticas XP:

* PRINCIPIOS DE ETICA, COMUNICAÇÃO E FEEDBACK (Mostrar aplicação em projetos reais).
* Planejamento de jogo (Como planejar iterações curtas e frequentes).
* Desenvolvimento Orientado a Teste (Importância dos testes automáticos.
* Demonstração prática de TDD).
* Programação em PAR (Benefícios e como implementar e exemplo interativo).
* Integração Continua (Como garantir a integração frequente do código)***.***
* Designer Simples (Focar em soluções simples e diretas).
* Refatoração (Melhorias contínuas no código sem alterar a funcionalidade).
* Propriedade Coletiva do código (Todos podem alterar qualquer parte do código.
* Ritmo Sustentável (Equilibrar a carga de trabalho para evitar burnout).
* Metáfora (Uso de metáforas para facilitar a compreensão do sistema.)
* Padronização de Código (Manter a consistência do código).
* Documentação Justa (Documentação necessária, sem excessos).

Aplicação e Casos de Sucesso:

* Estudos de caso (Apresentar exemplos de empresas que adotaram XP com sucesso).
* Discussão de Resultado (Impactos positivos observados com a implementação de XP).

Ferramenta de suporte ao XP:

* Ferramentas de TDD e Integração Continua
* Ambientes de desenvolvimento e colaboração (JIRA, Git, CI/CD pipelines etc.).

**Tecnologias para a apresentação: Uso de PowerPoint, Canvas e imagens do google.**

Pesquisa abaixo:

Diagrama

Descrição gerada automaticamente

**O Que É XP**

XP foi introduzido por Kent Beck em 1999 como uma resposta à necessidade de processos de desenvolvimento de software mais flexíveis e adaptáveis. A metodologia é baseada em princípios e valores que promovem a entrega frequente de software funcional, permitindo ajustes rápidos com base no feedback contínuo dos clientes.

**Aplicação**

XP é aplicado principalmente em ambientes onde os requisitos mudam rapidamente, como em startups ou projetos inovadores. A metodologia é ideal para projetos que exigem uma colaboração estreita entre desenvolvedores e clientes, permitindo que o software evolua à medida que as necessidades do cliente se tornam mais claras.

**Práticas XP**

**Princípios de Ética, Comunicação e Feedback**

Esses princípios são essenciais em XP para garantir que todos os membros da equipe estejam alinhados com os objetivos do projeto. Por exemplo, em projetos reais, a comunicação aberta e o feedback constante permitem que problemas sejam identificados e resolvidos rapidamente, evitando surpresas no final do projeto.

**Planejamento de Jogo**

XP utiliza iterações curtas, geralmente de 1 a 2 semanas, para planejar e entregar funcionalidades incrementais. Cada iteração é planejada com base nas prioridades do cliente, garantindo que as entregas mais importantes sejam feitas primeiro.

**Desenvolvimento Orientado a Teste (TDD)**

TDD é uma prática fundamental em XP, onde os testes são escritos antes do código. Isso garante que o código atenda exatamente aos requisitos e que erros sejam identificados cedo. Uma demonstração prática de TDD mostraria a criação de um teste para uma funcionalidade simples, seguido pelo desenvolvimento do código para passar no teste.

**Programação em Par**

Programação em par envolve dois desenvolvedores trabalhando juntos em uma única estação de trabalho. Isso aumenta a qualidade do código, pois ambos os desenvolvedores revisam o código em tempo real. Um exemplo interativo poderia ser a solução conjunta de um problema de codificação, onde cada membro da dupla contribui com ideias e revisa o trabalho do outro.

**Integração Contínua**

A integração contínua garante que o código novo seja frequentemente integrado ao repositório principal, permitindo a detecção precoce de conflitos e problemas. Ferramentas como Jenkins ou GitLab CI/CD são usadas para automatizar esse processo.

**Design Simples**

XP valoriza soluções simples que atendam aos requisitos atuais sem complicações desnecessárias. A simplicidade evita o acúmulo de dívida técnica e facilita a manutenção do código.

**Refatoração**

Refatoração é a melhoria contínua do código sem alterar sua funcionalidade. Essa prática permite que o código seja mantido limpo e eficiente, mesmo após várias iterações.

**Propriedade Coletiva do Código**

Todos os desenvolvedores têm a permissão e a responsabilidade de alterar qualquer parte do código. Isso promove um senso de responsabilidade compartilhada e evita gargalos quando apenas um membro da equipe pode trabalhar em determinada área do código.

**Ritmo Sustentável**

XP promove um ritmo de trabalho que pode ser mantido indefinidamente, evitando o burnout da equipe. Isso significa equilibrar as cargas de trabalho e evitar horas extras excessivas.

**Metáfora**

O uso de metáforas ajuda a equipe e os stakeholders a entenderem o sistema de maneira simples. Por exemplo, uma aplicação pode ser descrita como uma "fábrica de processamento", facilitando a compreensão de seu funcionamento.

**Padronização de Código**

Manter um estilo de codificação consistente é importante para que todos os membros da equipe possam facilmente entender e trabalhar no código de outros.

**Documentação Justa**

A documentação em XP é mantida ao mínimo necessário, evitando a criação de documentos extensos que rapidamente se tornam obsoletos. A ênfase está na documentação que realmente agrega valor ao projeto.

**Aplicação e Casos de Sucesso**

**Estudos de Caso**

Empresas como ThoughtWorks e Pivotal Labs adotaram XP com sucesso, alcançando alta qualidade de software e satisfação do cliente.

**Discussão de Resultados**

Com a implementação de XP, muitas equipes relatam uma redução nos defeitos de software, maior satisfação do cliente e melhoria na moral da equipe.

**Ferramentas de Suporte ao XP**

**Ferramentas de TDD e Integração Contínua**

Ferramentas como JUnit, NUnit, Jenkins, e Travis CI são amplamente utilizadas para suportar as práticas de TDD e integração contínua em XP.

**Ambientes de Desenvolvimento e Colaboração**

Plataformas como JIRA para gerenciamento de projetos, Git para controle de versão e pipelines de CI/CD são fundamentais para suportar a colaboração contínua e eficiente em projetos XP.