

Questão 3

Alunos: Matheus Carlos Lima e Silva
Emanuelle Passos Martins

202004728
202004710

A. No contexto de gerenciamento de projetos de software, explique o processo de análise de riscos.

A análise de riscos é um processo importante tanto para a viabilização de um projeto quanto para a manutenção e longevidade dele. Ela envolve a identificação, avaliação, priorização e mitigação de riscos, selecionando os que poderiam impactar mais abruptamente o projeto e traçando medidas de minimização. Podemos citar em etapas:

- **Identificação dos Riscos:** Reconhecimento de potenciais problemas que possam afetar o projeto. Isso não seria apenas uma análise técnica mas também de fatores externos, como, por exemplo, o financeiro. A exemplo, o custo de escalabilidade do sistema pode ser demasiadamente alto com as estratégias que até então estavam definidas.
- **Avaliação dos Riscos:** Reunião de informações de impacto e ocorrência provável de cada risco. Essa reunião de informação passa tanto por uma análise qualitativa, envolvendo a natureza do risco e probabilidade de ocorrência no sistema, quanto quantitativa por estatísticas de dados históricos ou estimativas. A análise qualitativa é uma excelente forma de métrica para fatores técnicos, enquanto que a quantitativa traz números melhores para a avaliação dos riscos por fatores externos.
- **Priorização dos Riscos:** Classificação dos riscos de acordo com a avaliação deles, mantendo em mente o impacto no projeto e a ocorrência. Em geral, os riscos de grande impacto e ocorrência são focados.
- **Mitigação dos Riscos:** Desenvolvimento de planos de ação para reduzir impactos e prováveis ocorrências, ou até mesmo eliminação do risco. Além disso, também se pode delegar a responsabilidade dos riscos a terceiros, por seguros ou contratos, ou até mesmo aceitá-los e agir apenas em caso de eventual ocorrência.
- **Monitoramentos dos Riscos:** Acompanhamento dos riscos ao longo do projeto, ajustando estratégias de mitigação ao longo do curso.

B. Uma empresa de desenvolvimento de software de médio porte precisa desenvolver um software de vendas de pacotes de viagens para uma

companhia de turismo. Explique como XP e Scrum podem ser combinados por esta empresa no desenvolvimento de software.

Essa combinação pode ocorrer pela aplicação do SCRUM no planejamento e gerenciamento do desenvolvimento, de forma a garantir de forma regular a entrega incremental de partes funcionais do software, e pela aplicação do XP para o desenvolvimento e práticas técnicas, assegurando, através de testes contínuos e programação em par, tanto a adequação aos requisitos, quanto a qualidade do código.

De forma geral podemos ter as seguintes práticas em cada um deles:

- **Planejamento e Gerenciamento com Scrum:**
 - Scrum Master: Responsável por facilitar as atividades do projeto, removendo obstáculos e garantindo que a equipe siga os processos do Scrum.
 - Sprints: Ciclos de desenvolvimento curtos (algumas semanas no geral) onde um incremento do software é desenvolvido, testado e lançado.
 - Product Backlog: Lista de requisitos priorizados que são selecionados para desenvolvimento durante as sprints.
 - Daily Scrum Meetings: Reuniões diárias para garantir que o trabalho está sendo gerenciado de acordo com o plano e para ajustar as ações conforme necessário.

- **Desenvolvimento e Práticas Técnicas com XP:**
 - Histórias de Usuário: Descrição de requisitos como histórias que representam cenários de uso. No caso da venda de pacotes de viagem, o usuário poderia ser um vendedor que poderia estar pedindo um painel de controle de vendas para monitorar o desempenho e ajustar suas estratégias. Normalmente as histórias de usuário são claras e essenciais para a entrega de valor ao usuário final.
 - Test-Driven Development: Desenvolvimento prévio dos testes, garantindo que os requisitos sejam atendidos desde o início.
 - Programação em Par: Trabalho conjunto de dois desenvolvedores em um mesmo código, de forma a melhorar a qualidade e compartilhar conhecimento.
 - Refatoração Contínua: Constantes melhorias no código, de forma a manter sua qualidade e flexibilidade (em termos de adequação a diferentes cenários, como formato de entradas).
 - Integração Contínua: Integração e teste frequente do código para detecção rápida de problemas.