

ENGENHARIA DE SOFTWARE – ATIVIDADE TEÓRICA 3

3.1) Explique por que, para as empresas, a entrega rápida e implantação de novos sistemas é mais importante do que a funcionalidade detalhada desses sistemas?

3.2) Explique como os princípios básicos dos métodos ágeis levam ao desenvolvimento e implantação de softwares acelerados?

3.3) Quando você não recomendaria o uso de um método ágil para o desenvolvimento de um sistema de software?

3.4) Extreme Programming expressa os requisitos dos usuários como histórias, com cada uma escrita em um cartão. Debata sobre as vantagens e desvantagens dessa abordagem para a descrição de requisitos.

3.5) Explique por que o desenvolvimento "Test-First" ajuda o programador a desenvolver um melhor entendimento dos requisitos do sistema. Quais são as possíveis dificuldades com o desenvolvimento Teste-First?

RESPOSTAS

3.1) As empresas priorizam a entrega rápida de sistemas porque precisam corresponder rapidamente às mudanças de mercado e à concorrência. Lançar um sistema funcional básico permite iniciar operações, atrair usuários e obter feedbacks para futuras melhorias. A funcionalidade completa pode ser construída pelo modelo incremental, evitando atrasos que comprometam oportunidades de negócio.

3.2) Os métodos ágeis permitem desenvolvimento rápido por meio de curtos ciclos, entregas frequentes e feedback constantes dos usuários. A autonomia das equipes, o foco em pessoas e a aceitação natural de mudanças reduzem a burocracia e facilitam adaptações rápidas. Além disso, a simplicidade evita o desenvolvimento de funcionalidades desnecessárias, evitando gastos ou desperdícios.

3.3) Métodos ágeis não são recomendados em projetos que exigem alta segurança e validação rigorosa, em organizações com forte rigidez burocrática, em equipes com pouca comunicação ou em situações conflitantes entre stakeholders e organizações. Portanto, a técnica visa evitar possíveis transtornos que danificam e prejudicam a dinâmica exigida entre os métodos ágeis.

3.4) As histórias de usuários facilitam a comunicação, o planejamento e a priorização, pois são simples e claras. Entretanto, podem faltar detalhes importantes, levando a ambiguidades que são limitadas pela produtividade de um sistema. Em projetos complexos, a falta de documentação detalhada prejudica o controle, a rastreabilidade e a manutenção do sistema. Embora seja eficiente, o desenvolvimento do sistema pode ser limitado por exigir uma abordagem robusta.

3.5) O desenvolvimento “Test-First” auxilia o programador a compreender os requisitos, pois força a análise antes de codificá-lo, apresentando melhoria de qualidade e agilidade em testes de regressão. Porém, o conceito exige técnica e habilidades, dificultando o uso em sistemas complexos sem cobrir requisitos não funcionais, exigindo um esforço inicial que compensa a qualidade e estabilidade do produto a longo prazo.