

Unidade de Ensino Menor

- Vantagens

Garante feedback rápido e constante, oferece mais flexibilidade em relação às mudanças de necessidades, reduz a codificação e possibilita o trabalho com modelos, tempo de entrega mais rápido e custo mínimo de desenvolvimento.

Maior satisfação do cliente com a lacuna mínima entre as expectativas do cliente e as entradas do desenvolvedor, é também um ótimo modelo para projetos com foco no orçamento e entrega rápida, apresentando consistentemente peças de trabalho do produto para as partes interessadas.

- Desvantagens

O RAD é menos escalável e não se encaixa bem em grandes projetos porque grandes equipes não conseguem acomodar mudanças tão rapidamente em um projeto.

Requer um maior comprometimento dos desenvolvedores e clientes para comunicar continuamente sobre os processos de feedback e avaliação. Este modelo é principalmente focado na interface do usuário e é usado principalmente para prototipagem para incorporar e avaliar mudanças com base no feedback do cliente

- Método Recomendado

Neste exemplo de negócio o melhor método seria o modelo RAD (Rapid Application Development), pois uma vez que a fase de planejamento é reduzida, a equipe consegue desenvolver mais em menos tempo, o projeto pode ser dividido em tarefas menores e mais gerenciáveis, o progresso é facilmente mensurado garantindo ciclos de desenvolvimento curtos e como o RAD enfatiza o uso de protótipos também é possível visualizar, testar e modificar o sistema mais rapidamente.

Unidade de Ensino Maior

- Vantagens

O Scrum pode ajudar as equipes a concluir as entregas do projeto de forma rápida e eficiente, garante o uso eficaz de tempo e dinheiro, grandes projetos são divididos em sprints facilmente gerenciáveis, os desenvolvimentos são co-dificados e testados durante a revisão do sprint.

Funciona bem para projetos de desenvolvimento rápidos, a equipe obtém visibilidade clara por meio de reuniões, ser ágil, adota feedback de clientes e partes interessadas, sprints curtos permitem mudanças com base no feedback com muito mais facilidade, o esforço individual de cada membro da equipe é visível durante as reuniões diárias de scrum.

- Desvantagens

Demora de entrega, a complexidade e robustez do código, não seria possível tanta flexibilidade e recepção de feedbacks vindas de usuários, mudar algo após a forma final do projeto seria muito mais complicado de ser feita sem consequências.

Scrum muitas vezes leva ao aumento do escopo, devido à falta de uma data final definida, As chances de fracasso do projeto são altas se os indivíduos não forem muito comprometidos ou cooperativos, Adotar a estrutura Scrum em grandes equipes é um desafio.

A estrutura pode ser bem-sucedida apenas com membros experientes da equipe, as reuniões diárias às vezes frustram os membros da equipe, se algum membro da equipe sair no meio de um projeto, isso pode ter um grande impacto negativo no projeto, a qualidade é difícil de implementar até que a equipe passe por um processo de teste agressivo.

- Método Recomendado

Neste exemplo o método mais recomendado seria o scrum pois, A técnica ágil, que inclui o framework Scrum, envolve gerenciar um projeto dividindo-o em várias partes e passando pelo ciclo de planejamento, execução e avaliação em cada etapa. Os ciclos do Scrum são conhecidos como sprints e normalmente duram entre duas e quatro semanas. Após cada corrida.

Todo o corpo de trabalho é revisto. Os participantes desempenham três papéis diferentes no processo: o proprietário do produto, que representa a organização, o scrum master, que atua como gerente de projeto e um grupo de indivíduos que executam as tarefas.

O Scrum é particularmente útil na indústria de desenvolvimento de software, pois ajuda uma equipe a se concentrar em um objetivo menor antes de integrá-lo ao objetivo principal do projeto. No entanto, suas qualidades podem ser usadas para uma ampla gama de projetos em vários setores, particularmente aqueles que exigem respostas rápidas a solicitações urgentes e uma equipe de desenvolvimento de projetos flexível.

- Conclusão

Antes de adotar um modelo de desenvolvimento de software é importante atentar para alguns pontos como:

1. As partes interessadas estarão dispostas a seguir a abordagem escolhida?
2. Qual é o tempo necessário para aplicar este modelo?
3. Sua equipe de desenvolvedores estará e terá capacidade para entregar nesse tempo estabelecido?
4. Você tem um baixo risco técnico?
5. Sua equipe tem as ferramentas e experiências necessárias para suportar tal tecnologia?

Aluno: José Carlos Gandra de Brito

Matricula: 202108695734

