

Proposta de Desenvolvimento de Software: Sistema de Gerenciamento de Tarefas para Equipes Remotas

1. Escolha do Modelo de Processo

Para o desenvolvimento deste software, o Modelo Incremental foi o processo de desenvolvimento de software adotado. Este modelo é uma representação simplificada das atividades que constituem o processo de software e os papéis das pessoas envolvidas na engenharia. A filosofia central do Modelo Incremental é entregar o software em versões sequenciais, adicionando novas funcionalidades a cada incremento.

As principais características deste modelo incluem a divisão do projeto em partes menores, com a entrega de um produto funcional em cada fase. Ao invés de entregar um produto final completo e acabado, o desenvolvimento e a entrega ocorrem de modo incremental, ou seja, as funcionalidades vão sendo disponibilizadas aos poucos.

A escolha do Modelo Incremental é justificada por sua flexibilidade e capacidade de adaptação a mudanças. Diferente do Modelo Cascata, que é um processo sequencial e sistemático, o Incremental não exige que todos os requisitos sejam definidos antecipadamente. Ele permite que o feedback do cliente seja incorporado a cada entrega, garantindo que o produto final atenda às expectativas. Enquanto o Cascata se concentra em ter todas as etapas concluídas antes de passar para a próxima, o Incremental é ideal para este projeto, pois a natureza dinâmica de equipes remotas pode levar a requisitos que evoluem com o tempo.

2. Descrição do Projeto

O software proposto é um Sistema de Gerenciamento de Tarefas para Equipes Remotas, que será desenvolvido como um aplicativo web e mobile. A intenção é oferecer uma ferramenta completa para organização e acompanhamento de tarefas para equipes que trabalham à distância.

Requisitos Funcionais:

- **Incremento 1:**
 - Criação e gerenciamento de contas de usuário.
 - Login e logout de usuários.
 - Criação, edição e exclusão de tarefas.
 - Definição de status para tarefas (por fazer, em andamento, concluída).
- **Incremento 2:**
 - Funcionalidade de atribuição de tarefas a outros membros da equipe.
 - Ferramenta de comentários em tarefas para comunicação.
 - Painel individual para visualização das tarefas atribuídas a cada usuário.

- **Incremento 3:**

- Implementação de um painel de visualização de progresso no formato Kanban.
- Filtros avançados de busca por status, prazo ou responsável.
- Geração de relatórios básicos de produtividade.

Requisitos Não Funcionais:

- **Segurança:** O sistema deve ser protegido para que usuários maliciosos não consigam acessar ou danificar o sistema.
- **Usabilidade:** O sistema deve ser compreensível e de fácil uso.
- **Desempenho:** As telas devem carregar em um tempo aceitável para o usuário.
- **Confiabilidade:** O sistema deve funcionar conforme a especificação do cliente.

As expectativas do cliente incluem uma ferramenta intuitiva, segura e que simplifique a comunicação e o acompanhamento das tarefas. Os principais desafios são garantir que a arquitetura do sistema permita a escalabilidade de forma eficiente e que a usabilidade seja priorizada em todas as fases do desenvolvimento.

3. Aplicação do Modelo Incremental

O desenvolvimento do software será estruturado em três incrementos, cada um entregando um conjunto de funcionalidades prontas para uso e teste.

- **Fase 1 (Incremento 1):** Foco na implementação das funcionalidades básicas do sistema. Ao final desta fase, o cliente receberá um MVP (Produto Mínimo Viável) que permite o gerenciamento inicial de tarefas. O feedback sobre a interface de cadastro, login e a criação de tarefas será coletado.
- **Fase 2 (Incremento 2):** Com base no feedback da fase 1, serão implementadas as funcionalidades de colaboração. A equipe adicionará a atribuição de tarefas e o sistema de comentários, permitindo que o cliente comece a usar o software de forma colaborativa. O feedback desta fase será crucial para validar se as funcionalidades de equipe atendem às necessidades.
- **Fase 3 (Incremento 3):** Esta fase se concentrará em funcionalidades de visualização e análise de dados. Serão implementados o painel Kanban e os relatórios. O feedback final do cliente garantirá que as ferramentas de análise e visualização estejam de acordo com as expectativas e que o software final esteja pronto para uso em larga escala.

4. Análise Crítica

A utilização do Modelo Incremental neste projeto apresenta diversas vantagens. A principal delas é a capacidade de

reduzir riscos, pois o desenvolvimento e a validação ocorrem em etapas, permitindo a correção de falhas e a adaptação a novas necessidades de forma contínua. Além disso, a entrega progressiva do software aumenta o envolvimento do cliente no projeto, pois ele pode testar e dar feedback sobre um produto funcional em cada incremento.

Uma desvantagem e um risco potencial é a necessidade de um planejamento arquitetural robusto desde o início para garantir que as novas funcionalidades se integrem sem problemas. Sem um bom planejamento, a adição de novos recursos pode se tornar complexa e custosa.

Em comparação com o Modelo Cascata, o Modelo Incremental é mais adequado porque a natureza do projeto, focado em trabalho remoto, exige flexibilidade. O Cascata, com seu fluxo de trabalho rígido, não permitiria que a equipe se adaptasse facilmente a mudanças nos requisitos, o que poderia levar a um produto final que não se alinhava com as necessidades do cliente.

Para adaptar o modelo, seria interessante integrar práticas ágeis, como reuniões diárias e automação de testes, para aumentar a eficiência da equipe. Isso daria ao modelo incremental a agilidade de um processo ágil, mas mantendo a estrutura e o planejamento do desenvolvimento por fases.

5. Conclusão

O Modelo Incremental foi a escolha ideal para o desenvolvimento do Sistema de Gerenciamento de Tarefas para Equipes Remotas. Sua abordagem de entregas progressivas e a incorporação contínua de feedback do cliente garantem que o software final seja de alta qualidade, confiável e alinhado com as necessidades do usuário.

O modelo contribui diretamente para a viabilidade do software, pois o cliente pode testar as funcionalidades em cada fase, o que minimiza os riscos de falhas funcionais e de aceitação do produto. Ao final do desenvolvimento, espera-se que o sistema melhore a produtividade e a organização das equipes, tornando o trabalho remoto mais eficiente e colaborativo.