Análise de Viabilidade - Suricato

1. Introdução

Este documento descreve a análise de viabilidade do projeto Suricato.

2. Definição dos problemas

I. **Baixa produtividade no home office**
Diversas empresas não conseguiram sustentar esta forma de trabalho, após o ápice da pandemia, sendo alguns dos motivos, a falta de preparo e produtividade dos funcionários e cargos de liderança. Empresas reportaram uma queda em sua produtividade. Como iniciativa de tentar aumentar a eficiência, somado à falta de capacitação para a novo modelo de trabalho, ocorreu aumento de reuniões desnecessárias com os líderes de equipe que causaram um efeito inverso, contribuindo para diminuição do desempenho dos funcionários, como dito na pesquisa publicada no *Becker Friedman Institute*.

II. **Falta de eficiência na coordenação das equipes**
O principal problema que o software visa solucionar é a falta de eficiência na gestão de projetos de desenvolvimento de software. Isso inclui a dificuldade em coordenar equipes de desenvolvimento, rastrear o progresso dos projetos, garantir a comunicação eficaz e manter um alto padrão de qualidade de código.

2.2 Quais pessoas são afetadas?

Este problema afeta principalmente profissionais e equipes envolvidas no desenvolvimento de software, incluindo desenvolvedores, designers, testadores e gerentes de projeto. Além disso, as empresas de desenvolvimento de software como um todo são afetadas, pois problemas de eficiência podem resultar em atrasos nos projetos, baixa qualidade do software e custos adicionais.

2.2 Qual o motivo desses problemas?

I. **Complexidade dos projetos:** Os projetos de desenvolvimento de software estão se tornando mais complexos devido à diversidade de tecnologias e às crescentes expectativas dos clientes.
II. **Equipes distribuídas:** Muitas equipes de desenvolvimento estão distribuídas geograficamente, o que pode tornar a comunicação e a coordenação mais desafiadora.

III. **Crescimento de código aberto:** O aumento da colaboração em projetos de código aberto significa que as equipes frequentemente dependem de ferramentas externas e precisam integrar seus projetos com outras comunidades de desenvolvedores.
IV. **Necessidade de qualidade:** A qualidade do software é fundamental, e problemas de qualidade podem resultar em retrabalho e custos adicionais.
V. **Aprendizado contínuo:** A indústria de desenvolvimento de software evolui rapidamente, e as equipes precisam de maneiras eficazes de aprender e incorporar novos conhecimentos e melhores práticas.
VI. **Coordenação de tarefas:** À medida que as equipes crescem, coordenar as tarefas e responsabilidades individuais se torna um desafio.
VII. **Reuniões desnecessárias**: Muitas vezes, por falta de preparo na tentativa de aumentar a eficiência dos colaboradores, os líderes exageram na quantidade de reuniões que pode ter o efeito inverso e dificultar o trabalho.

O software proposto visa abordar esses problemas, oferecendo uma solução abrangente que melhora a gestão de projetos de desenvolvimento de software, promove a colaboração eficaz e fornece assistência contínua para manter a qualidade e a eficiência em um ambiente em constante mudança.

3. Análise de Viabilidade do Projeto

Considerar a viabilidade deste software com base nos seguintes aspectos:

1. **Viabilidade técnica:**
    - **Integração com outras ferramentas:** A capacidade de integrar-se a ferramentas populares como Github e Jira é um ponto forte, pois não compete diretamente com essas ferramentas, mas sim complementa suas funcionalidades.
    - **Tecnologia disponível:** A tecnologia necessária para desenvolver o software já existe, e as técnicas de IA estão amplamente disponíveis para uso.
2. **Viabilidade de mercado:**
    - **Demanda crescente:** O mercado de desenvolvimento de software está em constante crescimento, com uma demanda crescente por ferramentas que melhorem a eficiência e a qualidade.
    - **Concorrência:** Embora haja concorrência, a proposta de valor única deste software, com foco na integração e assistência baseada em IA, pode atrair um nicho específico de usuários.
3. **Viabilidade financeira:**

- **Modelos de negócios sustentável:** Um modelo de negócio sólido, com assinaturas mensais ou anuais, pode garantir a sustentabilidade financeira do software.
- **Custos de desenvolvimento:** Os custos iniciais de desenvolvimento e manutenção devem ser estimados com precisão.

4. **Viabilidade operacional:**
    - **Facilidade de uso:** O software deve ser intuitivo e de fácil adoção, permitindo que as equipes o utilizem sem grandes curvas de aprendizado.
    - **Suporte e manutenção:** Deve haver planos de suporte e manutenção para garantir que o software permaneça eficaz e atualizado.
5. **Viabilidade legal e ética:**
    - **Conformidade com regulamentos:** É importante garantir que o software esteja em conformidade com regulamentos de proteção de dados e questões éticas relacionadas ao uso da IA.
6. **Viabilidade de desenvolvimento:**
    - **Equipes de desenvolvimento:** É fundamental ter uma equipe competente para o desenvolvimento, manutenção e melhorias contínuas do software.
    - **Recursos e prazos:** Os recursos necessários, incluindo tempo e habilidades técnicas, devem ser avaliados em relação aos prazos de desenvolvimento.
7. **Viabilidade Estratégica:**
    - **Diferenciação no Mercado:** O software deve oferecer recursos que o diferenciam da concorrência e o tornem uma escolha estratégica para as equipes de desenvolvimento.

4. Conclusão

Com base nessa análise de viabilidade, o software é viável, desde que seja desenvolvido e comercializado de forma estratégica, considerando as necessidades específicas do mercado de desenvolvimento de software e as capacidades únicas que ele oferece. Além disso, a integração com ferramentas existentes, como GitHub e Jira, pode aumentar sua atratividade para empresas que já as utilizam. No entanto, é fundamental realizar pesquisas de mercado, obter feedback de potenciais usuários