Termo de Abertura de Projeto

Título: SuBox (Software de Avaliação para Pequenos Estabelecimentos)

1. Finalidade (Justificativa)

Essa ferramenta pretende facilitar a avaliação direta do desempenho interno de comércios através das opiniões dos consumidores, além de transparecer de forma clara quais são as expectativas que os clientes têm em relação à organização. Portanto, essa solução é recomendável para facilitar o acesso de comércios menores com um recurso que pode melhorar o ambiente comercial e proporciona uma visão mais ampla de quais melhorias podem ser elaboradas.

O principal foco são pequenos comércios físicos que não possuem recursos financeiros para este tipo de produto, desta forma, será implementado o processo de retorno de avaliações na comunicação entre cliente com o estabelecimento. O propósito será de ajudar na identificação rápida dos problemas internos dentro da empresa e com os clientes, para poderem aprimorar suas operações.

2. Descrição

O principal impeditivo que pequenos estabelecimentos enfrentam para utilizar recursos de retorno da avaliação, como o processo de feedback direto, é a questão financeira por ter um custo elevado. Por mais que haja métodos alternativos para esse tipo de gerenciamento de avaliação como: planilhas, papéis ou até mesmo caixas de sugestões, ainda existem riscos nesses casos. Como por exemplo:

- A planilha se torna um recurso fácil de acessar, porém se torna muito difícil de gerenciar e/ou armazenar informações, principalmente para pessoas sem o acesso da informação de como utilizar esse tipo de ferramenta. Normalmente, existem cursos para ter conhecimento de como utilizar as planilhas (exemplo:

Excel), mas isso demanda tempo e atrasa tanto o funcionário quanto a empresa de aprimorar financeiramente.

- A utilização de papéis é mais arriscada ainda, tendo em mente que é muito fácil perder esse tipo de documento rapidamente e muitas vezes a franquia de pequeno porte não vai ter disponibilidade de avaliar cada comentário individualmente para fazer uma análise geral.

- Utilização das caixas de sugestões seria mais inviável ainda, tendo em mente: o material utilizado para fazer a caixa, os lápis e canetas para anotar, os papéis que podem ser perdidos/alterados/rasurados e com o mesmo empecilho da utilização de papel, demanda muito esforço fazer a leitura de cada avaliação separadamente.

3. Objetivo

Criar um programa capaz de fazer uma análise geral da avaliação do cliente, através de questionários e alternativas, que será enviado diretamente para a empresa com uma entrega de relatório e coletânea de dados gerais, sendo mais simples e objetivo na hora de definir as melhorias que precisam ser feitas no comércio. A primeira implementação do software será armazenar as avaliações feitas pelos clientes através de um painel de instruções.

4. Critérios para o Sucesso (Benefícios Esperados)

- ***Adoção do Software:*** Medir o número de pequenos estabelecimentos que adotaram e estão usando o software regularmente. Uma meta poderia ser que 70% dos estabelecimentos-alvo adotem o software nos primeiros seis meses após o lançamento.

- ***Satisfação dos Usuários:*** Avaliar a satisfação dos usuários através de pesquisas de feedback. Um objetivo pode ser alcançar uma pontuação média de satisfação de 8/10 nos primeiros três meses.

- ***Melhoria na Eficiência:*** Verificar se o software ajuda os estabelecimentos a melhorar a eficiência das suas operações. Isso pode ser medido pelo tempo economizado nas avaliações ou pela redução de erros nas avaliações. Uma meta poderia ser reduzir o tempo de avaliação em 30%.

- ***Retenção de Clientes:*** Medir a taxa de retenção dos clientes que usam o software. Uma meta pode ser que 80% dos estabelecimentos continuem usando o software após o primeiro ano.

- ***Geração de Relatórios e Insights:*** Avaliar a capacidade do software em gerar relatórios úteis e insights para os estabelecimentos. Um critério de sucesso pode ser que 90% dos usuários considerem os relatórios gerados como úteis para a tomada de decisões.

- ***Lançar a Versão Beta:*** Lançar uma versão beta do software para um grupo seleto de estabelecimentos em três meses.

- ***Treinamento de Usuários:*** Realizar sessões de treinamento para 100% dos usuários do software nos primeiros dois meses após o lançamento.

- ***Atualizações e Melhorias:*** Lançar atualizações mensais para o software, incorporando feedback dos usuários e melhorias contínuas.

- ***Parcerias Estratégicas:*** Estabelecer parcerias com associações de pequenos estabelecimentos para promover o software e aumentar a adoção.

- ***Relatórios de Performance:*** Desenvolver e implementar um sistema de relatórios de performance que possa ser acessado pelos estabelecimentos a qualquer instante por um editor de textos.

5. Equipe

- Estevan Hernandes Silva de Oliveira;
- Lucas Vinicius de Oliveira;
- Melissa Batista Junqueira;
- Milena Souza Borges Silva;
- Soap Sophie Daisy (Caio Felipe) Lopes Maran.

6. Principais Entregas

Abaixo estão especificados os principais entregáveis do projeto, definidos por um escopo não mutável.

Documento que engloba o cronograma para cada fase do projeto, estimativa dos custos envolvidos, incluindo recursos humanos, plano de comunicação

6.1. Definição do escopo
6.2. Plano de atividades
6.3. Definição dos requisitos
6.4. Documentação do projeto
6.5. Cronograma definido
6.6. Planejamento de comunicação e recursos
6.7. Arquitetura do sistema
6.7.1 Interface de painel de comandos
6.7.2. Código fonte em linguagem C do software
6.8. Implementação do projeto, com usuários de teste
6.9. Realização dos testes com o usuário
6.10. Análise de melhorias a serem implementadas
6.11. Documento final

6.12. Implementação oficial do projeto

6.13. Apresentação do projeto

7. Critérios de Avaliação

Como o projeto é acadêmico e não tem intenções lucrativas, como o objetivo principal sendo a assistência com a comunidade local, o único critério de avaliação possível será a documentação completa.

Seguindo os seguintes parâmetros:

- Entrega do TAP (Termo de Abertura do Projeto) Final;
- EAP (Estrutura Analítica do Projeto);
- Cronograma de Gantt;
- Carta de Envolvimento;
- Dicionário EAP;
- Documentação e Requisitos Funcionais;
- Pitch de Apresentação;
- Fluxograma do Projeto;
- Código em C;
- Apresentação final com relatório do desenvolvimento;

8. Programação de Eventos

- **Planejamento das atividades:** Março a Julho de 2024, responsável: Melissa Batista Junqueira.
- **Desenvolvimento do Código em C:** Agosto de 2024, responsável: Estevan Hernandes Silva de Oliveira e Soap Sophie Daisy (Caio Felipe) Lopes Maran.
- **Testes e Validação do SuBox:** Agosto de 2024, responsável: Milena Souza Borges Silva.
- **Lançamento e treinamento da ferramenta:** Agosto de 2024, responsável: Lucas Vinicius de Oliveira e Envolvimento Externo.

9. Hipótese-chave

a. Tecnologias

O desenvolvimento será utilizando a IDE CodeBlocks, em conjunto com a linguagem de programação C. Para estruturação do projeto será projetado um fluxograma contendo as funcionalidades do software.

b. Análise de Riscos Tecnológicos

Sendo a linguagem C limitada apenas a aplicações de console, torna-se dificultoso o desenvolvimento de uma solução completa sem um armazenamento adequado, sem uma interação agradável ao usuário e de difícil implementação.

c. Critérios de Sucesso

O software possui um código identado de fácil manutenção, e sua execução possui interações simples onde apenas o teclado é utilizado. Como se trata de um protótipo, a empresa possui todos os requisitos necessários para o uso cotidiano.

10. Restrições

Entre as principais restrições para o desenvolvimento e implementação do projeto encontra-se:

- **Disponibilidade de Recursos:** Devido ser um projeto voluntário/institucional, os recursos são limitados e são utilizados apenas softwares gratuitos para o desenvolvimento.
- **Prazos:** O projeto deve ser produzido em tempo limitado equivalente a um semestre letivo, e entregue até agosto/2024.
- **Regulamentos:** Desenvolvimento de programa na área da tecnologia que possua impacto na comunidade local.
- **Restrições Críticas:** Os membros do projeto não possuem conhecimentos aprofundados sobre desenvolvimento de softwares Web.

11. Riscos

a. Riscos Técnicos

- **Falhas no Desenvolvimento de Software:** Problemas com bugs, incompatibilidade com diferentes sistemas operacionais ou plataformas.
- **Tecnologia Desatualizada:** Usar tecnologias que se tornam obsoletas rapidamente, o que pode afetar a manutenção e a escalabilidade do software.
- **Segurança:** Riscos de segurança cibernética, como vulnerabilidades que podem ser exploradas por hackers.

b. Riscos de Gestão

- **Estimativa de Tempo e Recursos:** Subestimação do tempo necessário para completar o projeto ou sobrecarga de recursos, o que pode levar a atrasos e aumento de custos.
- **Escopo do Projeto:** Mudanças frequentes no escopo podem levar a problemas de gerenciamento e atrasos.
- **Comunicação:** Falhas na comunicação entre membros da equipe, stakeholders e outros envolvidos no projeto.

c. Riscos Financeiros

- **Orçamento:** Despesas imprevistas ou custos que superam o orçamento previsto podem afetar a viabilidade do projeto.
- **Falta de Investimento:** Dificuldades para garantir financiamento suficiente para a conclusão e manutenção do software.

d. Riscos de Mercado

- **Aceitação do Mercado:** O software pode não atender às necessidades ou expectativas dos pequenos estabelecimentos, resultando em baixa adoção.
- **Concorrência:** Novos concorrentes ou mudanças nas ofertas de concorrentes podem impactar a posição do SuBox no mercado.

e. Riscos Operacionais

- **Treinamento e Suporte:** Falta de treinamento adequado para os usuários finais ou suporte técnico insuficiente pode impactar a experiência do usuário.
- **Infraestrutura:** Dependência de infraestruturas externas (como servidores e provedores de serviços) pode ser um ponto de falha.

f. Riscos Legais e Regulatórios

- **Conformidade:** Necessidade de conformidade com regulamentações e leis locais, como proteção de dados pessoais (por exemplo, GDPR).
- **Licenças e Direitos Autorais:** Problemas relacionados a licenciamento de software ou propriedade intelectual.

Estratégias de Mitigação

Para cada tipo de risco, você deve considerar estratégias de mitigação:

- **Riscos Técnicos:** Implementar testes rigorosos, manter atualizações regulares e realizar auditorias de segurança.
- **Riscos de Gestão:** Usar metodologias ágeis, como Scrum, para gerenciar mudanças e comunicação de forma eficaz.
- **Riscos Financeiros:** Realizar um planejamento financeiro detalhado e manter uma reserva de contingência.
- **Riscos de Mercado:** Conduzir pesquisas de mercado e envolver os stakeholders durante o desenvolvimento do produto.
- **Riscos Operacionais:** Oferecer treinamento completo e garantir suporte técnico contínuo.
- **Riscos Legais e Regulatórios:** Consultar um especialista jurídico para garantir conformidade com as leis e regulamentações.

12. Requisitos de Aprovação

Requisitos a serem cumpridos:

- Entregar e apresentar um fluxograma do programa para a Professora Zady Castaneda Salazar, a partir da matéria de Algoritmo e Programação;

- Entregar e apresentar código em C para os Professores: Fábio Feliciano e Ricardo Sovat.

Para cada professor, os requisitos se aplicam da seguinte forma:

- Deverá ser desenvolvida uma aplicação de console, na linguagem de programação C, utilizando o ambiente de desenvolvimento Code::Blocks.
- A aplicação deverá ter uma função bem definida, ou seja, resolver/solucionar algum problema/situação.

Será considerado o nível de complexidade da aplicação desenvolvida.

- O projeto deverá conter, no mínimo, uma implementação dos seguintes conceitos:
    - Estrutura condicional;
    - Estrutura de repetição;
    - Vetor e/ou Matriz;
    - Estruturas (Struct) e/ou Vetor de Estruturas;
    - Funções – Passagem de Parâmetros por Valor e Retorno de Valores (pelo menos um exemplo de função com passagem e retorno de valores).

      Caso seja implementada mais de uma função, não é obrigatório que cada função receba e retorne valores. Mas é obrigatório que esses conceitos estejam presentes na aplicação.

    - Código indentado;
    - Nomes de variáveis representativa;
    - Validação da entrada dos dados;

- Entregar todos os documentos exigidos para o Professor Carlos Belluzzo, referente à matéria de Gestão de Projetos pelo GitHub:
    - Raiz(./):
    - Proposta de Projeto;
    - Documentos Preliminares:
    - Termo de Abertura;
    - EAP;
    - Cronograma;
    - Plano de Trabalho;
    - Carta de Envolvimento;
    - Dicionário EAP;
    - Documento de Requisitos Funcionais;
    - Pitch de Apresentação;

Com isto, os requisitos para a aprovação do projeto se encerram,com prazo para o final do mês de Agosto/2024.