

Plano de Gerenciamento de Configuração

Introdução

Este documento possui o propósito de guiar as alterações futuras do software em questão.

Escopo

O projeto foi feito para o plano de gerenciamento da configuração da versão 1.2 do software “Keygle”, criado para a disciplina de Web 1. O software funciona como uma ferramenta de busca de chaves de aplicações em sites confiáveis e que apresentam preços muitas vezes mais acessíveis que nos sites oficiais.

Foi definido modelo Scrum de processo para a manutenção do software pela sua praticidade e agilidade no versionamento de projetos de alta mutação, além da garantia de que o projeto está de acordo com a necessidade do cliente.

Termos usados

Gerenciamento, Scrum, evolução, processo, desenvolvimento.

SCM Management

Organização

Projeto será dividido em cliente, *scrum master* e *scrum team* (desenvolvedores).

Desenvolvedores são divididos em duas equipes: front-end e *back-end*.

Responsabilidades

Cliente: terá a responsabilidade de fornecer as informações de forma clara e de participar da validação dos protótipos criados no final de cada sprint.

Scrum master: fará a comunicação com cliente, será responsável por toda a documentação criada (documento de requisitos, de artefatos e demais documentos de controle), e participará no desenvolvimento quando e se possível.

Scrum team: responsável pela parte de desenvolvimento de código fonte (*front-end* realizando scripts e estilização, *back-end* realizando APIs e banco de dados), implementação do produto (participantes: desenvolvedores, *scrum master*).

Autoridades

Apenas o *scrum master* terá autorização para mexer no Jira e no Trello. Desenvolvedores *front* e *back-end* um não podendo alterar o código do outro, qualquer necessidade de mudança em partes não autorizadas por um integrante da equipe será direcionada ao scrum master, por isso ocorrerá a divisão por equipes no git. Somente o Scrum Master tem a permissão de inserir um novo desenvolvedor (ou usuário quando produto não finalizado) no projeto.

Políticas aplicadas

Políticas de privacidade de dados públicos e pessoais.

Diretivas

Cada uma das ferramentas durante o processo devem possuir usabilidade restrita de acordo com a função (ex: *Scrum team* deve utilizar o Visual Studio Code para programar e o Git para o versionamento da aplicação, podendo apenas visualizar os itens do trello, enquanto que o *Scrum Master* pode visualizar e editar os itens do trello).

Procedimentos

- Recebimento de requisição de mudança do cliente pelo scrum master;
- análise de viabilidade juntamente com a equipe;
- início do processo Scrum de desenvolvimento;
- Ao término do desenvolvimento do *product backlog* ocorre validação do protótipo pelo cliente;
- Implementação do produto e o término do processo Scrum.

SCM Activities

Identificação de configuração

Todos os desenvolvedores devem utilizar o git para versionamento, controle de código, build, teste e deploy automatizados, lista de tarefas e funções no Trello.

O *Scrum master* discutirá com a equipe para verificar a aplicabilidade das mudanças, seus valores, margens de erro, duração de desenvolvimento e implementação.

Controle de configuração

Toda submissão feita no github terá necessariamente o número da versão do código, em qual parte do desenvolvimento ele se encontra (nome definido no Trello) e descrição das alterações feitas. O controle do versionamento será feito no Github e no Trello, preocupações com o deploy e build serão desnecessárias devido a automatização realizada no GitHub e também ao teste automatizado com o uso da tecnologia Selenium, apenas o Scrum master terá a disponibilidade de utilizar o software Jira para monitorar o processo de desenvolvimento do software em escala geral.

SCM Schedules

Para o projeto, foi definido a seguinte programação: *Sprint backlog* decidido em 15 minutos, sprint 3 dias máximo, reunião diária de 5 minutos, reunião de sprint review duração de 10 minutos.

Coordenação com outros projetos ativos

Não precisaremos da coordenação com outros projetos pois as alterações feitas serão básicas e não necessitarão de muitas mudanças. Caso necessário, utilizaremos o modelo definido.

Sprint backlog decidido em 1 dia numa reunião com duração de no máximo 3 horas, sprint com duração de 2 semanas no máximo, reunião diária de 20 minutos, reunião semanal de sprint review duração de 2 horas.

SCM Resources

Ferramentas

Para o desenvolvimento deste projeto será utilizado o editor de códigos Visual Studio Code, gerenciador de processos Jira (junto com Trello), Github para repositório de código fonte, build e deploy automatizados, Selenium para teste automatizado e banco de dados Firebase.

Recursos Físicos

Serão necessários computadores pessoais para o desenvolvimento e gerenciamento do projeto e acesso a internet para envio do código desenvolvido.

Recursos Humanos

A equipe será formada por designers, programadores *back-end*, programadores *front-end*, *scrum master* e *product owner*. Pode ser necessário a contratação de serviços de desenvolvimento de software de pessoas ou empresas que dominam tecnologia desconhecida pela equipe.