
Minecrafters

Minecrafters Project Plan

Versão 1.0

Histórico da Revisão

Data	Versão	Descrição	Autor
20/06/2023	1.0	Preenchimento do Documento	Arthur Nogueira

1. Introdução

O Plano de Desenvolvimento de Software descreve as atividades, processos e recursos necessários para o desenvolvimento bem-sucedido de um plugin de Minecraft. Este plano estabelece uma estrutura para garantir que o desenvolvimento seja realizado de forma eficiente, cumprindo requisitos e prazos estabelecidos.

2. Objetivos do Desenvolvimento:

Definir claramente os objetivos do desenvolvimento do plugin, incluindo funcionalidades, recursos e metas específicas a serem alcançadas.

Estabelecer critérios de sucesso que serão usados para avaliar a conclusão e a qualidade do plugin.

3. Equipe de Desenvolvimento:

Identificar os membros da equipe de desenvolvimento, suas responsabilidades e suas habilidades técnicas relevantes.

Assegurar que a equipe tenha o conhecimento e as ferramentas adequadas para realizar o desenvolvimento do plugin.

4. Metodologia de Desenvolvimento:

Escolher uma metodologia de desenvolvimento adequada, como Scrum, Kanban ou Waterfall, levando em consideração as necessidades do projeto e da equipe.

Definir os principais marcos e entregáveis do projeto, bem como os prazos para cada fase do desenvolvimento.

5. Requisitos e Especificações:

Documentar os requisitos e especificações do plugin de forma clara e detalhada.

Identificar as funcionalidades-chave, os fluxos de trabalho e as interfaces com o usuário.

Estabelecer critérios de aceitação para validar a conformidade do plugin com os requisitos.

6. Arquitetura e Design:

Projetar a arquitetura do plugin, identificando os componentes principais, suas interações e dependências.

Criar diagramas de classe, diagramas de sequência ou outros artefatos de design necessários para auxiliar na implementação.

7. Implementação e Testes:

Implementar o plugin de acordo com as especificações e design estabelecidos.

Realizar testes unitários, de integração e de sistema para garantir a qualidade e a conformidade com os requisitos.

Utilizar ferramentas de controle de versão e gerenciamento de código-fonte para facilitar o trabalho colaborativo e o rastreamento das alterações.

8. Documentação:

Elaborar documentação abrangente, incluindo guias de instalação, configuração e uso do plugin.

Criar documentação interna para facilitar a manutenção futura e fornecer orientações para a equipe.

9. Controle de Qualidade:

Estabelecer métricas e critérios para avaliar a qualidade do código, a cobertura de testes e a conformidade com as melhores práticas de desenvolvimento.

Realizar revisões de código e inspeções regulares para identificar e corrigir problemas e garantir a qualidade do software.

10. Implantação e Suporte:

Preparar o plugin para implantação, incluindo a criação de pacotes de distribuição e instruções de instalação.

Fornecer suporte pós-lançamento, respondendo a dúvidas dos usuários e solucionando eventuais problemas.