Plano de gerenciamento de configuração

Projeto Jabiraca Web

Responsável: Guilherme Setim

Version 1.0

Última Atualização: 22/06/2023

Histórico de Alterações

Data	Versão	Descrição	Responsável
22/06/2023	1.0	Versão inicial do projeto	Guilherme Setim / André Luis Grein

Aprovadores

Nome	Função
Guilherme Setim	Engenheiro de configuração de software
André Luis Grein	Gerente/Líder de Projeto

Índice

1. Introdução

Este documento descreve planejamento para a gerência de configuração do projeto **Jabiraca Web** incluindo a identificação dos itens de configuração, estrutura adotada para o repositório dos itens, padrões de nomenclatura, ferramentas de apoio e outras informações relacionadas.

1.1 Público Alvo

Desenvolvedores e engenheiros de configuração.

1.2 Definições, Acrônimos e Abreviações.

Termo	Definição
Baseline	Marco no desenvolvimento do projeto, composta de um conjunto de artefatos aprovado, estáveis e consistentes entre si.
Branches	Caminho alternativo para o desenvolvimento em paralelo, criado através de rótulos aplicado à determinada versão de um artefato.
CR	Do inglês, Change Request, solicitação de mudanças.
IC	Item de Configuração, ou seja, qualquer artefato do projeto que será submetido à gerência de configuração.
Label, Tag, Rótulo	Marca usada para identificar facilmente uma versão específica de um artefato ou baseline.
CCB	Do inglês, Software Configuration Control Board ou Change Control Board, grupo responsável por autorizar modificações nos itens de configuração e estabelecimento de baselines.
Merge	Processo de combinar linhas de desenvolvimento separadas em um único conjunto de alterações em um projeto de software. Durante o merge, as alterações feitas em cada ramificação são comparadas e combinadas de forma a preservar o trabalho de todos os envolvidos
SCM, CM	Do inglês, Software Configuration Management.

2. Organização

2.1 Papéis e Responsabilidades

A seguir, são descritos os papéis e responsáveis relacionados às atividades de gerência da

configuração e mudanças.

Papel	Nome	Responsabilidades
Engenheiro de configuração de software	Guilherme Setim	● Definir e manter o plano de gerência de configuração e mudanças do projeto. ● Configurar ambiente para as atividades de gerência de configuração e mudanças; ● Realizar auditorias de configuração de software; ● Estabelecer baselines de acordo com critérios planejados. ● Efetuar merges de acordo com critérios planejados. ● Criar branches sempre que necessário dentro dos critérios previstos.
Gerente/Líder de Projeto	André Luis Grein	● Analisar impacto das solicitações de mudanças com relação aos compromissos assumidos e à arquitetura do software.
Desenvolvedor	André Luis Grein	● Segue as políticas de gerência de configuração estabelecidas no plano. ● Abre solicitações de mudança quando necessário.

2.2 Ferramentas, Ambientes e Infra-estrutura.

Esta seção descreve a infra-estrutura que será utilizada para a realização das atividades de gerência de configuração e mudanças.

Ferramenta	Versão	Propósito	Link/Acesso
TeamCity	2023.05	Automação da geração de Build	https://www.jetbrains.com/pt-br/teamcity/
Git	2.34.1	Controle de Versões	https://git-scm.com/
Ubuntu	22.04 LTS	Sistema Operacional	https://ubuntu.com/

2.3 Repositório

Informações Gerais do Repositório	
Tipo do Repositório	Git
Servidor do Repositório	GitHub

Diretório *home*	src/
Acesso ao Repositório	https://github.com/guilhermesetim/jabiraca-web
Estrutura do Repositório	

- //<diretório raiz/ Documentos/RequisitosAnaliseProjeto/...
    - Documento de Requisitos / estórias de usuários
    - Documento de Arquitetura
    - Diagramas de análise (ex. casos de uso)
    - Diagramas de projeto (ex. diagrama de classes, projeto de BD etc)
- //<diretório raiz/ Documentos/Planejamento/...
    - Plano Gerencia de Configuração;
    - Plano de Qualidade;
    - Plano de Projeto (cronograma geral, riscos);
- //<diretório raiz/ Documentos/Teste/...
    - Plano de Testes
    - Projeto de Testes
    - Resultados de teste
- //<diretório raiz/ Código/AmbienteDesenvolvimento...
    - Código fonte
- //<diretório raiz/ Código/AmbienteTeste
    - Código fonte

3. Políticas de Gerência de Configuração

Esta seção estabelece as políticas e diretrizes que devem ser seguidas pelo projeto em relação à gerência de configuração e controle de mudanças. A gerência de configuração é fundamental para garantir a estabilidade, rastreabilidade e integridade do software durante todo o ciclo de vida do projeto. Por meio da definição e aplicação de políticas consistentes, buscamos manter um ambiente controlado e organizado, garantindo que todas as mudanças e modificações sejam adequadamente registradas, avaliadas e implementadas.

- **Permissão de Acesso ao Repositório**

Guilherme Setim:

Nome de usuário: guilhermesetim
Senha: ********
Permissões de acesso: possui acesso completo ao repositório, incluindo a capacidade de fazer alterações, adicionar e excluir arquivos.

André Luis Grein:

Nome de usuário: AndreLuisGrein
Senha: ********
Permissões de acesso: tem acesso de leitura e gravação a determinadas pastas e arquivos específicos do repositório, relacionados às suas responsabilidades no projeto. No entanto, ele não possui permissão para excluir ou modificar arquivos fora de sua área de atuação.

Clientes:

Permissões de acesso: tem acesso somente leitura ao repositório. Podem visualizar e baixar arquivos, mas não podem fazer alterações ou adicionar novos arquivos.

- **Administração das Ferramentas de Controle de Versão e Mudanças**

  Guilherme Setim

- **Estabelecimento de Baselines**

  Guilherme Setim

- **Merges**

  O responsável por realizar uma Merge é Guilherme Setim.
  Para realização de uma merge no projeto a branch deve possui necessariamente:

  1. Código estável: Antes de realizar uma merge, o código no branch de desenvolvimento ou na branch feature deve estar estável e sem erros conhecidos.

  2. Revisão de código: Antes de efetuar a merge, é recomendado que o código seja revisado por um ou mais membros da equipe.

  3. Testes: Após a revisão do código, é importante realizar testes para verificar se as alterações realizadas não afetaram negativamente o funcionamento do sistema.

  4. Resolução de conflitos: Caso ocorram conflitos durante a merge, eles devem ser resolvidos de forma adequada.

- **Branches**

  Todos os desenvolvedores podem criar uma nova branch.

As branch’s devem ser criadas da seguinte forma:

➔ Propósito: Ao criar uma nova branch, pode ser uma nova funcionalidade, uma correção de bug, uma melhoria ou qualquer outro objetivo específico.

➔ Nome descritivo: A nova branch deve ter um nome descritivo que indique seu propósito. Utilize uma convenção de nomenclatura que seja compreensível e consistente entre a equipe.

➔ Base adequada: Ao criar uma nova branch, é importante realizar corretamente a branch de base. A branch principal de desenvolvimento é a **"main"**.

➔ Trabalho individual: Cada membro da equipe deve criar sua própria branch para trabalhar em suas tarefas.

- **Geração de Builds**

A política de geração de builds tem como objetivo garantir a qualidade e a consistência das versões do software entregues. Para isso, é necessário seguir as seguintes diretrizes:

➔ Automatização do processo: A geração de builds deve ser automatizada, utilizando TeamCity, como sistemas de integração contínua ou ferramentas de build.

➔ Versionamento adequado: Cada build gerado deve ter um número de versão único e significativo, para indicar claramente o nível de compatibilidade e as mudanças introduzidas em cada versão.

➔ Ambientes de build separados: Ambientes de build separados para diferentes tipos de builds, como desenvolvimento, teste e produção.

➔ Testes automatizados: Antes de gerar um build, é fundamental executar testes automatizados para garantir a integridade e a qualidade do software. Os testes devem abranger casos de uso essenciais, incluindo testes de unidade, testes de integração e testes de aceitação.

➔ Controle de dependências: Verificação e controle das dependências do software, garantindo que todas as bibliotecas e recursos necessários estejam corretamente incluídos no build. Utilizar o gerenciador de dependências Composer.

4. Identificação de Configuração

*4.1.1 **Nomenclatura***

4.1.1.1 Labels/Tags

4.1.1.2 *Nomenclatura Baseline e Releases*

Regra

<PROJETO>-<SIGLA>-<EXTRA><XX.YY>_<NN> **Exemplo: JW - BL - T - 1.5_3**

Onde,

<PROJETO> é o identificador do projeto;

<SIGLA> é a sigla utilizada (SPRINTXX, RELEASE, BASELINE)

<EXTRA> informação utilizada para identificar melhor a sigla. Exemplo: (TESTE)

<XX.YY> Esquema de Versionamento, XX para mudanças maiores (aprovada pelo cliente), YY para mudanças menores (aprovadas pelo moderador da inspeção ou pelo responsável pelo documento).

<NN> Build

4.1.1.3 Branches

Regra:

<PROJETO>-BRANC-<VERSÃO > **Exemplo: JW-API-1.2**

Onde,

<PROJETO> é o identificador do projeto;

<XX.YY> Esquema de Versionamento, XX para mudanças maiores (aprovada pelo cliente), YY para mudanças menores (aprovadas pelo moderador da inspeção ou pelo responsável pelo documento).

4.1.1.4 *Documentos*

Regra:

<PROJETO>- NOME DO DOC>-<TIPO>-<EXTRA>.<EXT> **Exemplo: JW-REQUISITOS-DOCUMENTAÇÃO.pdf**

Onde,

<PROJETO> é o identificador do projeto;

<NOME DO DOC> é o nome do documento;

<TIPO> é o identificador do tipo de documento;

<EXTRA> é um campo para informação extra

<EXT> é a extensão do documento.

4.1.1.5 Código

4.2 Configuração e Controle de Mudanças

4.2.1 Processo de Solicitação de mudança e Aprovação

Todos os desenvolvedores podem solicitar mudança. Entretanto, a aprovação é de responsabilidade de Guilherme Setim.

4.2.2 Estados (Ciclo de Vida) de uma Solicitação de Mudança

1. **NOVO**: Quando a CR é aberta. Qualquer pessoa pode submeter uma CR;
2. **EM DESENVOLVIMENTO:** A CR foi designada para alguém e está em desenvolvimento;
3. **RESOLVIDO:** A implementação da CR está concluída pelo responsável;
4. **VERIFICADO:** A CR foi verificada pelas partes interessadas;
5. **CONCLUIDA:** A CR está fechada.

4.2.3 Auditorias

O objetivo das auditorias de configuração é garantir que a configuração do software esteja alinhada com as políticas e diretrizes estabelecidas. Segue abaixo o planejamento:

Auditoria	Responsável	Data	Prazo	Template
Auditoria Inicial	Gerente/Líder de Projeto	28/06/2023	2 dias	Template de Auditoria de Configuração

Auditoria de Conformidade Mensal	Engenheiro de configuração de software	Última semana de cada mês	7 dias	Template de Auditoria de Configuração
Auditoria de Configuração após Alterações Significativas	Engenheiro de configuração de software	Após cada alteração significativa no software	Dentro de 1 semana após a alteração	Template de Auditoria de Configuração
Auditoria de Configuração de Entrega	Gerente/Líder de Projeto	Antes de cada entrega de versão	Dentro de 3 dias após a entrega	Template de Auditoria de Configuração
Auditoria de Encerramento do Projeto	Gerente/Líder de Projeto	21/07/2023	1 Dia	Template de Auditoria de Configuração