

PLANO DE GESTÃO DE CONFIGURAÇÃO DE SOFTWARE (SOFTWARE CONFIGURATION MANAGEMENT - SCM)

1.0 Introdução (Dieison)

A fim de organizar as atividades de controle de configuração de software, garantir a qualidade dos produtos implementados, manter a integridade dos objetos e monitorar as versões do sistema, foi desenvolvido este plano de gestão de configuração de software.

1.1 Escopo e objetivo das atividades de SCM (Dieison)

Para garantir a integridade do projeto, assim como a qualidade do sistema a ser desenvolvido, este documento descreve a infraestrutura detalhada a ser usada. Permitindo um gerenciamento e controle do projeto mais precisos, a fim de garantir a qualidade durante as modificações e adequações que possam surgir durante o desenvolvimento.

1.2 Papel organizacional da SCM (Dieison)

Devido ao tamanho reduzido do projeto, não haverá um grupo específico de SCM, o Gerenciamento de Configuração será feito pelos próprios participantes, que devem ter a responsabilidade de solicitar, reportar e registrar qualquer mudança realizada ou novo SCI criado, de modo que todos possam trabalhar sem ocorrer conflitos entre versões. Todavia, o documentador do projeto deve controlar o andamento de SCM através dos relatórios gerados, de modo a evitar inconsistências.

2.0 Tarefas de SCM

Esta seção detalha todas as tarefas importantes de SCM.

2.1 Identificação (Ricardo)

Será utilizado um padrão de nomenclatura para documentos, visando uma boa prática que facilitará a identificação como um todo. O padrão será formado pelas seguintes características:

- Os arquivos serão nomeados de acordo com o seguinte *template*:

NOMEDOPROJETO_ARTEFATO_DATA

exemplo:

- BBTEC_QA_25052022

Na nomenclatura acima, fica explícito que o documento se refere à área de qualidade do projeto, pois tem a sigla QA, e foi criado no dia 25/05/2022.

Observações:

- O tamanho total do nome de identificação não deve ultrapassar 30 caracteres.
- O nome do arquivo deve ser escrito em letras maiúsculas
- Os arquivos de código seguirão um padrão diferente, que consiste em descrever a funcionalidade de seu conteúdo.

Tabela de Documentos e siglas para serem usadas nas nomenclaturas.

Artefato	SIGLA
Plano de Gerência de Projeto de Software	PS
Plano de Gerência de Configuração de Software	CS
Plano de Garantia da Qualidade de Software	QA
Plano de Testes do Software	PTS
Pauta e Ata de Reunião	REU
Relatórios (status, métricas, auditorias, resultado de testes)	REL
Arquitetura	ARQ
Requisitos	REQ
Processo	PROC
Testes	TST
Modelo de Classes	MDL
Modelo de Entidade-Relacionamento	MER
Caso de Uso	UC
Planilha de Estimativas	EST

Tabela 1. Artefatos

2.2 Controle de Configuração (Ricardo)

Qualquer mudança que seja necessária, em qualquer documento, deve ser pré-aprovada pelo gerente ou pelo analista do projeto.

Após qualquer mudança em documentos, as mesmas devem ser revisadas pelo analista do projeto, que dirá se estão adequadas ou não, ou se poderia ser feito de uma maneira mais eficiente. Isso deve ocorrer antes de fundir as alterações à ramificação principal do repositório do projeto.

Após qualquer mudança em documentos de código, devem ser realizados testes para garantir que não há nenhum efeito indesejado imprevisto.

As mudanças devem ser descritas em um documento de relatório. Para mudanças em código, o conteúdo do relatório deve estar presente, também, na descrição do pedido de mesclagem (*Pull Request* ou *Merge Request*) à ramificação de nível superior do repositório do projeto.

A tabela a seguir demonstra os campos que devem ser apresentados na solicitação de mudança.

Campo	Valor
Componente	BBTECH
Versão	A versão do projeto
Plataforma	Todas
OS	Todos
Prioridade	Prioridade da mudança requisitada. P1 - Alta prioridade; P2 - Média prioridade; P3 - Baixa prioridade. Em dúvida, deixar o valor padrão (P3) selecionado
Tipo de Problema	Escolha o tipo: REFACTOR: melhoria do artefato/feature já existente FEATURE: criação de novo artefato/feature TASK: uma tarefa PATCH: selecione esse tipo para contribuições externas na forma de patches
Responsável	Nome do responsável (requerente) da mudança
CC	Lista de destinatários de cópias de e-mails relativos a essa mudança.
URL	URL para o card correspondente à mudança requisitada
Descrição	Descrição detalhada do problema. Para arquivos MD (MARKDOWN), utilizar o seguinte template: # Objetivo descrição do objetivo ## Visão Geral visão geral do que foi alterado ### Alterações - alteração 1 - detalhe alteração 1 - alteração 2 - alteração 3

Tabela 2. Formulário de Mudanças

- A. Os itens a seguir demonstram os passos para realização de uma mudança em objetos referenciais:
- B. O problema é reconhecido
- C. O problema é avaliado
- D. Relatório contendo sugestões de mudança
- E. Reunião com analistas e desenvolvedores para discussão de baixo-nível acerca das sugestões de mudança

- F. Relatório de mudanças definidas
- G. O pedido será aceito, e nesse caso colocado em fila, ou será negado, e nesse caso o processo se encerra neste item
- H. A modificação é feita e submetida para análise
- I. A modificação é revisada pelo responsável
- J. Testes de garantia de qualidade são realizados
- K. A modificação é mesclada à ramificação de nível superior do repositório do projeto, para a criação ou complementação de uma nova versão
- L. Testes finais são realizados na versão finalizada
- M. Nova versão é distribuída

2.3 Controle de Versão (Adriano)

O controle de versão deve ser feito através da nomenclatura descrita abaixo, com o gerenciamento feito através de uma ferramenta de CVS.

O versionamento do sistema deve seguir o seguinte esquema:

<XX>.<YY>.<ZZ>

onde:

<XX> = Major Release Number. Versão do software a ser entregue ao cliente com modificações substanciais nos requisitos/funcionalidades do sistema. Inicia-se este campo com o valor "01" e só deve ser incrementado depois do primeiro release final.

<YY> = Feature Release Number. Número a ser incrementado quando um release for produzido com uma nova funcionalidade incluída no sistema. Inicia-se este campo com o valor "00" e volta a este valor quando <XX> é alterado. Este campo só deve ser incrementado depois do primeiro release final.

<ZZ> = Defect Repair Number. Número a ser incrementado quando um novo release for produzido para correção de um release <XX>.<YY>. Inicia-se este campo com o valor "00" e volta a este valor quando <XX> ou <YY> é alterado. Este campo só deve ser incrementado depois do primeiro release final.

O versionamento da documentação do projeto deve seguir o seguinte esquema:

<XX>.<YY>

onde:

<XX> é o número que representa a versão do ultimo baseline;

<YY> representa a versão draft do documento <XX>.

As mudanças de versão são baseadas nas seguintes regras:

- A primeira versão de um documento deve ser 00.01;
- A cada modificação do documento o <YY> deve ser incrementado;
- Após a aprovação de cada documento, o campo <XX> deve ser incrementado em uma unidade e o campo <YY> deve voltar a 00, gerando uma nova versão estável e oficial.

2.4 Configuration status accounting (CSA) (Dieison)

A finalidade CSA será manter as informações de configuração do produto para garantir que as configurações atuais e históricas do produto e as informações de configuração do produto possam ser determinadas com precisão durante todo o ciclo de vida do produto. Por isso é de suma importância que todas as informações essenciais do CSA sejam relacionadas com precisão à versão/revisão exata aplicável do produto e mantidas em repositório disponível para todos os envolvidos, para acompanhar as modificações realizadas.

2.5 Auditorias and Revisões (Pedro)

Como todo o projeto será acompanhado por ferramentas de controle, para visualização e gerência por parte da equipe, nestas também serão destinados etapas do projeto somente para Auditorias, onde o documentador irá atualizar os relatórios e a documentação(que estará em apenas um local, centralizada para visualização de qualquer colaborador do projeto). Porém antes mesmo destas etapas de Auditoria ou até mesmo durante elas, revisores irão observar atualizações ou qualquer modificação significativa ao projeto, para eventuais correções técnicas e/ou procedimentos necessários em falta para só então o processo do documentador ser efetuado.

Após todos os procedimentos serem revisados de acordo com o SCM e o processo de auditoria executado corretamente, então tudo é registrado em relatórios e documentações e armazenadas no repositório do projeto.

2.6 Armazenamento(Adriano)

Descreve-se abaixo a forma de armazenamento dos SCIs relacionados ao projeto BEBETECH. Para um melhor entendimento é usada uma demonstração gráfica em forma de árvore facilitando a visualização da estrutura de diretórios de armazenamento. E através de uma tabela, os diretórios são explicados mais especificamente.

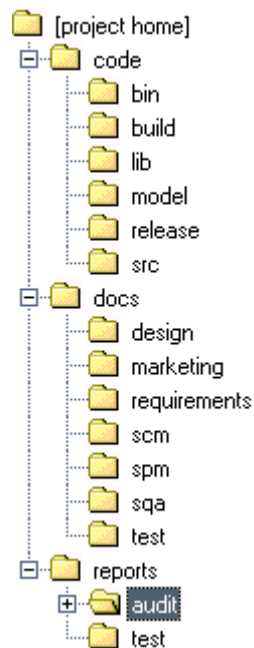


Figura1. Estrutura de diretórios do CVS

Diretório	Descrição
code	Contém diretórios para o código fonte, bibliotecas utilizadas e scripts de compilação.
bin	Arquivos binários e executáveis
build	Scripts de compilação (build.xml).
lib	Bibliotecas externas utilizadas pelo projeto.
model	Modelos UML
release	Versões de release do projeto
src	O código fonte está localizado nesta pasta ou em sub-pastas.
docs	A documentação para o projeto está nas sub-pastas de docs
design	Artefatos de design, como modelos UML e descrições da arquitetura do sistema.
marketing	Documentos de marketing como a proposta comercial, resposta ao RFP, e SLA.
requirements	Documentos de requisito.
scm	Documentos de gerência de configuração de software.
spm	Documentos de planejamento.
sqa	Documentos de garantia de qualidade de software.
test	Documentos de teste para o projeto, incluindo documentos de procedimentos de testes, planos de teste, etc.
reports	Relatórios do projeto
audit	Relatórios de auditoria do projeto
test	Relatórios de teste do projeto

Tabela 3. Detalhamento dos diretórios

4.0 Recursos para SCM

Ferramenta	Descrição
DotProject	Gerenciamento de Projetos
Subversion	Sistema de controle de versão
NetBeans	IDE para desenvolvimento da aplicação
PHP	Linguagem de desenvolvimento utilizada
Apache	Servidor Web
MySQL	SGBD a ser utilizado
JUDE	Geração de diagramas
GoogleGroups	Comunicações adicionais da equipe

5.0 Ferramentas, Técnicas e Métodos SCM (Adriano)

Ferramentas, Técnicas e métodos especializados a serem empregados pelo grupo de SCM são descritas aqui

6.0 Apêndices (Adriano)

Informação suplementar é fornecida aqui

7.0 Repositório

<https://github.com/FredThiesen/EngSoft-CTRL-ALT>