Plano de gerência de configuração de software

Histórico de revisões do modelo

Versão (XX.YY)	Data (DD/MMM/YYYY)	Autor	Descrição	Localização
1.0.0	09/ABR/2013	Pedro	Draft inicial	

Aprovadores

Flávio Luciano	Desenvolvedor
Pedro Alexandre	Desenvolvedor

Faculdade Nova Roma

Projeto Dado Virtual

Plano de Gerência de Configuração de Software

Versão: Versão 1.0.0
Data:09/04/2013
Identificador do documento:SCMP

Histórico de revisões

Versão (XX.YY)	Data (DD/MMM/YYYY)	Autor	Descrição	Localização

Índice

1. Introdução

1.1. Propósito

Este documento apresenta os processos necessários para gerenciar as mudanças no projeto, provendo assim uma infra-estrutura necessária para a evolução do projeto.

1.2. Público Alvo

Os interessando diretamente nesse documento são todas as pessoas envolvidas no projeto, em especial, aqueles envolvidos nas atividades de engenharia do projeto (principalmente programação).

1.3. Escopo

Neste documento esta detalhada toda a infra-estrutura que será utilizada durante o projeto.

2. Atividades

Nesta seção serão apresentadas algumas atividades a serem seguidas pelos colaboradores do Projeto Dado Virtual.

2.1. Nomenclatura e identificação

2.1.1. Identificador de Documento (Doc ID)

Todos os itens de configuração do projeto, com exceção do código fonte, devem possuir um identificador de documento (doc_id). Os doc_ids devem seguir o seguinte esquema de nomenclatura:

<nome_projeto>_<tipo_artefato>_<data_da_criação>[_<descrição>]

Onde:

<nome_projeto> Nome do projeto em letras maiúsculas;

<tipo_artefato> tipo do artefato, conforme apresentado na Tabela 2 -1;

<data_da_criação> momento da criação do arquivo no formato AAAAMMDD;

<descrição> breve descrição do arquivo, pode ser o título do documento. Esta descrição só pode conter caracteres maiúsculos e não acentuados (A-Z), números (0-9), hífen (-) e sublinhado (_).

Notas:

- Idealmente não deve ultrapassar 50 caracteres.
- Este campo é opcional, caso seja omitido, a identificação do arquivo termina no último dígito da data da criação.
- Para artefatos únicos de um projeto, como os planos, a descrição pode ser omitida, no caso de artefatos passíveis de ter várias instancias, como relatórios, por exemplo, então é necessário o preenchimento deste campo.

Artefato	**tipo _artefato**
Plano de Gerência de Projeto de Software	SPMP
Plano de Gerência de Configuração de Software	SCMP
Plano de Garantia da Qualidade de Software	SQAP
Plano de Testes do Software	STP
Pauta e Ata de Reunião	MEET
Relatórios (status, métricas, auditorias, resultado de testes)	RPT
Documentos de Arquitetura	ARCH
Artefatos Comerciais	MKT
Documentos de Requisitos	REQ
Documentos de Processo	PRC
Documentos de Testes	TST

Tabela 2-1 Tipos de artefatos

Arquivos de código fonte são identificados pelo seu nome completo, ou seja, o nome do pacote, seguido por ponto (.) e o nome da classe.

2.1.2. Número de versão

<defina padrão para versionamento dos artefatos ou aceite a sugestão definida nessa seção>

Todos os artefatos, excluindo código e relatórios, devem possuir um numero de versão seguindo o padrão a seguir:

XX.YY

Onde:

XX Número da versão do documento

YY Número de rascunho (*draft*) do documento

A evolução do número de versão segue as seguintes regras:

- A primeira versão do documento deve ser 00.01;
- A cada modificação do documento, o valor YY deve ser incrementado;
- Após cada aprovação do documento, sua versão XX deve ser incrementada em um, e o valor YY volta a 00, formando uma "versão cheia" ou oficial;

É considerada uma aprovação do documento: a) Aprovação do documento após uma revisão; b) Aprovação por pelo menos dois (2) membros do core group responsável pelo artefato respectivo.

Artefatos que são código-fonte ou relatórios de vem possuir o padrão a seguir:

XX.YY.ZZ

Onde:

XX Número da versão Major do item de configuração (Modificações que são incompatíveis com as anteriormente definidas);

YY Número da versão Minor do item de configuração (Modificações que acrescentam funcionalidades novas substanciais).

ZZ Número do versionamento de revisão ou correção de bugs.

A evolução do número de versão segue as seguintes regras:

- A primeira versão do documento deve ser 1.0.0;
- A cada modificação do documento, o valor ZZ deve ser incrementado;
- Após cada inclusão de nova funcionalidade, sua versão YY deve ser incrementada em um, e o valor ZZ volta a 00, formando uma "versão cheia" ou oficial;

- Após cada modificação de Framework, Banco de Dados ou Padrões de interface gráfica o valor XX deve ser incrementado em um e o valor YY volta a 00,e o valor ZZ volta a 00, formando uma "versão cheia" ou oficial;

2.1.3. Local de armazenamento

O repositório especificado para armazenar os artefatos de projeto Sistema de Gerenciamento de Configuração de Software deverá ser o GITHub.

2.1.4. *Baselines* de projeto

Os *baselines* de projeto são descritos na Tabela 2 -2.

Baseline	Descrição	Formato
1.0.0	PROTÓTIPO FUNCIONAL	

Tabela 2-2 Baselines de projeto

2.1.5. Política de uso de *branches*

2.1.5.1. *Branch* principal (*main*)

É o *branch* principal do projeto. Apenas versões *baseline* devem ser integradas a este *branch*, isso garante que este branch conterá um código testado e funcional ou documentos revisados e aprovados.

2.1.5.2. *Branch* de integração

Os *branches* de integração são utilizados para integrar um conjunto de mudanças antes de integrá-las ao *branch* principal. Um conjunto extensivo de testes deve ser realizado antes desta integração, assim, o *branch* principal conterá apenas código já testado, e documentos revisados.

O *label* de um branch de integração segue o seguinte padrão:

<nome_projeto>_int_<build>

Onde:

<nome_projeto> Nome do projeto em letras maiúsculas;

<build> Número da *build*.

2.1.5.3. *Branch* de desenvolvimento

É neste *branch* onde todos os desenvolvimentos devem ser feitos. Ele deve ser sempre criado a partir de um *baseline* de release (*branch* principal).

Antes de integrar o código de um *branch* de desenvolvimento no *branch* de integração, o engenheiro deve garantir que o código esteja atualizado com o ultimo *baseline* e então testar novamente a modificação para garantir que continua funcionando.

<nome_projeto>_dev_<issue_id>_<descrição>

Onde:

<nome_projeto>	Nome do projeto em letras maiúsculas;
<issue_id>	Identificador da *issue* sendo resolvida neste *branch*;
<descrição>	Breve descrição sobre as mudanças feitas.

2.1.6. Política de uso de branches para documentos

Documentos não utilizaram *branches*. Antes de criar uma nova versão é preciso fazer um *check out* no documento e um *lock* (travamento), para impedir o trabalho paralelo no documentos.

2.2. Relatórios de configuração

A cada release deve ser gerado um documento de *release notes*, contendo quais são as modificações adicionadas àquela versão (incluindo o número do bug), além de informação útil para os usuários.