
Plano de Gerência de Configuração

Projeto UFRPE – Sistema Geral de Controle

Versão 1.0

Histórico de Alterações

Data	Versão	Descrição	Autor
18/06/13	1.0	Criação do plano de gerência de configuração	Aldo Romero
18/06/13	1.0	Revisão e ajustes	Matheus Xavier

Índice

1. Introdução	4
1.1 Público Alvo	4
1.2 Glossário	4
2. Organização	4
2.1 Ferramentas	4
2.2 Localização dos artefatos	5
2.3 Repositório	6
2.4 <i>Baselines</i> do Projeto	7
2.5 Política para o uso de <i>branches</i>	7
2.5.1 <i>Branch principal (main)</i>	7
2.5.2 <i>Branch de integração</i>	7
2.5.3 <i>Branch de desenvolvimento</i>	8
3. Identificação de Configuração	8
3.1 Nomenclatura	8
3.1.1 <i>Documentos</i>	8
3.1.2 <i>Código</i>	9
3.1.3 <i>Commit</i>	11
3.1.4 <i>Criação do Scrummy</i>	12
3.2 Configuração e Controle de Mudanças	12
3.2.1 <i>Processo de Solicitação de Mudança</i>	12
3.2.2 <i>Estados (Ciclo de Vida) de uma Solicitação de Mudança</i>	13
4. Auditoria de Configuração	14

1. Introdução

Este documento descreve planejamento para a gerência de configuração do projeto **UFRPE – Sistema Geral de Controle** 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

Stakeholders do projeto, especialmente os desenvolvedores e engenheiros de configuração.

1.2 Glossário

Esta seção explica o conceito de alguns termos importantes que serão mencionados no decorrer deste documento.

Tabela 1 – Glossário

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, <i>Change Request</i> , 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, <i>Software Configuration Control Board</i> ou <i>Change Control Board</i> , grupo responsável por autorizar modificações nos itens de configuração e estabelecimento de baselines.
SCM, CM	Do inglês, <i>Software Configuration Management</i> .

2. Organização

2.1 Ferramentas

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

Tabela 2 – Ferramentas, Ambientes e Infraestrutura

Ferramenta	Versão	Propósito
SVN	1.7	Controle de Versões
code.google.com.br	-	Ambiente para promoção de código livre, interação com desenvolvedores e gerenciamento de projetos
PHP	5.4	Linguagem Backend
HTML	5	Linguagem Frontend
CSS	3	Linguagem Frontend
MySQL	5.6	SGBD
Redmine	2.3	Ferramenta de Gerenciamento de Projeto
Apache	2.4	Servidor HTTP
SublimeText	2	Ferramenta de Desenvolvimento
NetBeans IDE	7.3	Ferramenta de Desenvolvimento
Eclipse	4.2	Ferramenta de Desenvolvimento
TortoiseSVN	1.7	Cliente SVN
GoogleGroups	-	Comunicação da equipe
Facebook	-	Comunicação da equipe
Hudson	3.0	Ferramenta de teste

2.2 Localização dos artefatos

Esta seção descreve a localização onde os artefatos estão armazenados.

Diretório	Descrição
<i>branches</i>	Contém todos os <i>branches</i> realizados no projeto, isto é, após a definição de que o projeto está pronto para ser liberado como uma versão, a pasta <i>trunk</i> é copiada para a pasta branch. Lá essa cópia receberá um nome de versão e esse branch não sofre mais alterações, apenas correções. Testes rigorosos devem ser efetuados com o branch que foi criado.
<i>tags</i>	Local que armazena a versão rigorosamente testada de um <i>branch</i> . Esta cópia testada do <i>branch</i> dá origem a um <i>release</i> ou a uma versão liberada para uso. É esse diretório <i>tags</i> que é empacotado e enviado para uso do cliente.
<i>trunk</i>	A pasta <i>trunk</i> contém todos os artefatos do projeto em desenvolvimento. Neste diretório, também estão todas as atualizações efetuadas no dia-a-dia. Este diretório é responsável por armazenar a estrutura principal do projeto, por meio dos diretórios <i>docs</i> e <i>src</i> .

2.3 Repositório

Tabela 3 - Repositório

Informações Gerais do Repositório	
Tipo do Repositório	SVN
Servidor do Repositório	GoogleCode
Diretório <i>home</i>	https://code.google.com/p/gp-darkbin/
Estrutura do Repositório	
<ul style="list-style-type: none"> • //doc/requisitos_analise/... <ul style="list-style-type: none"> ○ 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) • //doc/planejamento/... <ul style="list-style-type: none"> ○ Plano Gerencia de Configuração; ○ Plano de Qualidade; ○ Plano de Projeto (cronograma geral, riscos); • //doc/teste/... <ul style="list-style-type: none"> ○ Plano de Testes ○ Projeto de Testes 	

- Resultados de teste
- //src/...
 - Código fonte

2.4 **Baselines do Projeto**

Esta seção descreve quais *Baselines* são geradas para o projeto.

Tabela 4 – Baselines do Projeto

Baseline	Descrição	Padrão
Requisitos	Marcado assim que for concluída análise de requisitos da iteração.	SGC_REQ_ <ITERAÇÃO>
Análise de Projeto	Marcado quando forem concluídos a análise e o projeto de cada iteração.	SGC_ARCH_<ITERAÇÃO>
Build	Criada a cada <i>build</i> para o software.	SGC_BUILD_<BUILD>
Release	Criado a cada <i>release</i> do software.	SGC_RELEASE_<VERSÃO>
Documentos	Criado após a aprovação de um documento.	SGC_<NOME>

Onde,

<ITERAÇÃO> É o número da iteração, sendo utilizados para identificação dois dígitos começando em 01 e sendo incrementado de uma unidade a cada nova iteração;

<BUILD> É o número da *build*, sendo utilizados para a identificação três dígitos começando em 001 e sendo incrementado de uma unidade a cada nova *build*;

<VERSÃO> É o número da versão lançada.

2.5 **Política para o uso de *branches***

2.5.1 *Branch principal (main)*

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

2.5.2 *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:

SGC_int_<BUILD>

Onde,

<Build> É o número da *build*.

2.5.3 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 Gerente de Configuração deve garantir que o código esteja atualizado com o ultimo *baseline* e então testar novamente a modificação para garantir que continua funcionando.

SGC_dev_<ISSUE_ID>_<DESCRICAO>

Onde,

<ISSUE_ID> Identificador da *issue* sendo resolvida neste *branch*;

<DESCRICAO> Breve descrição sobre as mudanças feitas.

3. Identificação de Configuração

3.1 Nomenclatura

3.1.1 Documentos

Regra:

SGC_<ACRONIMO>_[<DESCRICAO>].<EXT>

Onde,

<ACRONIMO> É o acrônimo utilizado;

<DESCRICAO> É uma breve descrição do arquivo, **se necessário**. Esta descrição só pode conter caracteres maiúsculos e não acentuados (A-Z), números (0-9) e hífen (-);

<EXT> É a extensão do documento.

Tabela 5 - Acrônimos

Acrônimo	Descrição
SPMP	Plano de Gerência de Projeto de Software
SCMP	Plano de Gerência de Configuração de Software
SQAP	Plano de Garantia da Qualidade de Software
STP	Plano de Testes de Software
TRK	Relatórios (status, métricas, auditorias, resultado de testes)
ADT	Auditoria
UC	Documento de Casos de Uso
REQ	Documento de Requisitos
ARCH	Documento de Arquitetura
CD	Diagrama de Classes
ER	Diagrama Entidade Relacionamento
DF	Diagrama de Fluxo de Dados
SD	Diagrama de Sequência
ACT	Diagrama de Atividades
TC	Casos de Teste
DBM	Modelagem de Banco de Dados
BKI	Itens de <i>Backlog</i>
MEET	Atas de Reuniões
SPM	Planejamento de <i>Sprints</i>
TTC	Tabela de Horários dos Consultores

3.1.2 Código

3.1.2.1 Classes

Regra:

<NOME_CLASSE_CAMEL_CASE>

Onde,

<NOME_CLASSE_CAMEL_CASE> É o nome da classe iniciada com maiúscula. Não pode conter caracteres acentuados (A-Z), números (0-9) e hífen (-). Para palavras compostas, cada palavra é iniciada com maiúsculas unidas sem espaço.

3.1.2.2 Métodos

Regra:

<NOME_METODO>

Onde,

<NOME_METODO> É o nome do método iniciado com minúscula. Não pode conter caracteres acentuados (A-Z) e hífen (-). Para palavras compostas, a partir da segunda palavra, é iniciada com maiúsculas unidas sem espaço.

3.1.2.3 Parâmetros

Regra:

<NOME_PARAMETRO>

Onde,

<NOME_PARAMETRO> É o nome do parâmetro iniciado com minúscula. Não pode conter caracteres acentuados (A-Z) e hífen (-). Para palavras compostas, a partir da segunda palavra, é iniciada com maiúsculas unidas sem espaço.

3.1.2.4 Variáveis

Regra:

<NOME_VARIAVEL>

Onde,

<NOME_VARIAVEL> É o nome da variável iniciada com minúscula. Não pode conter caracteres acentuados (A-Z) e hífen (-). Para palavras compostas, a partir da segunda palavra, é iniciada com maiúsculas unidas sem espaço.

Observação: Para constantes, todas as letras são maiúsculas.

3.1.2.5 Comentários

Regra:

<DESCRICAO>

Onde,

<DESCRICAO> É o comentário com minúsculas e quebra linhas.

3.1.2.6 *Banco de Dados*

3.1.2.6.1 *Tabelas*

Regra:

<NOME_TABELA_CAMEL_CASE>

Onde,

<NOME_TABELA_CAMEL_CASE> É o nome da tabela iniciada com maiúscula. Não pode conter caracteres acentuados (A-Z), números (0-9) e hífen (-). Para palavras compostas, cada palavra é iniciada com maiúsculas unidas sem espaço.

3.1.2.6.2 *Colunas*

Regra:

<NOME_COLUNA>_<NOME_TABELA>

Onde,

<NOME_COLUNA> É o nome da coluna com minúsculas;

<NOME_TABELA> É o nome da tabela com minúsculas.

3.1.3 *Commit*

Regra:

SPRINT <NÚMERO DA SPRINT> - <TÍTULO> - <RESUMO>

Onde,

<NÚMERO DA SPRINT> É o número da *Sprint* atual;

<TÍTULO> É o título da atividade desenvolvida;

<RESUMO> É um resumo da descrição das alterações feitas.

3.1.4 Criação do Scrumy

Regra:

<code>www.scrumy.com/sw-2013.1-<NOME DO GRUPO>-sprint<NUMERO></code>
--

Onde,

<NOME DO GRUPO> É o nome do grupo responsáveis pelos requisitos descritos na página;

<NUMERO> É o número da *Sprint* atual.

3.2 Configuração e Controle de Mudanças

As mudanças nos itens de configuração do projeto devem estar sempre associadas a uma ou mais solicitações de mudanças. Dessa forma, para toda e qualquer modificação nos itens de configuração, um CR deve ser aberta conforme descrito.

3.2.1 Processo de Solicitação de Mudança

As solicitações de mudanças devem ser criadas através da ferramenta *Issues* disponível no site do projeto <https://code.google.com/p/gp-darkbin/>. Para tanto o usuário deverá estar logado no mail.google.com. O preenchimento dos campos deve ser feito seguindo as orientações descritas a seguir.

Tabela 6 – Campos e valores a serem preenchidos em uma CR

Campo	Valor
Summary	Breve descrição do problema
Description	Descrição detalhada do problema
Status	Status Aberto: New- A CR ainda não teve sua revisão inicial; Accepted – problema reproduzido/precisa reconhecimento; Started – o trabalho nesta CR começou. Status Fechado: Fixed – Desenvolvedor fez as mudanças pedidas, Gerente de Configuração deve verificar; Verified – Gerente de Configuração verificou que o reparo foi trabalhado; Invalid – relatório não válido; Duplicate – este relatório duplica uma CR existente; WonFix – foi decidido não examinar esta CR.
Owner	Responsável pela CR
Cc	Lista de destinatários de cópias de e-mails relativos a essa mudança
Labels	Permite que sejam selecionados o tipo e a prioridade da CR, o sistema operacional, etc. Type-Defect – Relato de um defeito no artefato a ser mudado;

	<p>Type-Enhancement – Requerimento para melhoria no artefato;</p> <p>Type-Task – os artigos trabalhados não requerem mudança no código ou nos documentos;</p> <p>Type-Pach – Correção do código fonte para revisão;</p> <p>Type-Other – Algum outro tipo de CR;</p> <p>Priority-Hign – Prioridade alta para a resolução do evento especificado;</p> <p>Priority-Medium – Prioridade média para a resolução do evento especificado;</p> <p>Priority-Low – Prioridade baixa para a resolução do evento especificado;</p> <p>OpSys-All – Todos os Sistemas Operacionais;</p> <p>OpSys-Windows – Sistema Operacional Windows;</p> <p>OpSys-Linux – Sistema Operacional Linux;</p> <p>OpSys-OSX – Sistemas Operacional OSX;</p> <p>Milestone-Release1.0 - ;</p> <p>Component-UI – CR relaciona-se ao programa UI;</p> <p>Component-Logic – CR relaciona-se a lógica da aplicação;</p> <p>Component-Persistence – CR relaciona-se ao componente de armazenamento de dados;</p> <p>Component-Scripts – roteiro de utilização e instalação;</p> <p>Componet-Docs – CR relacionada a documentação do usuário final;</p> <p>Security – segurança de risco dos usuários;</p> <p>Performance – CR de desempenho;</p> <p>Usability – efeitos da usabilidade do programa;</p> <p>Maintainability – impedir mudanças futuras.</p>
--	--

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

- **Aberta**

1. **New** - A CR ainda não teve sua revisão inicial;
2. **Accepted** – problema reproduzido/precisa reconhecimento;
3. **Started** – o trabalho nesta CR começou.

- **Fechada**

1. **Fixed** – Desenvolvedor fez as mudanças pedidas, Gerente de Configuração deve verificar;
2. **Verified** – Gerente de Configuração verificou que o reparo foi trabalhado;
3. **Invalid** – relatório não válido;
4. **Duplicate** – este relatório duplica uma CR existente;
5. **WonFix** – foi decidido não examinar esta CR.

4. Auditoria de Configuração

As auditorias de configuração devem ser feitas para cada ciclo do processo de desenvolvimento de forma a garantir que o processo de gerência de configuração vem sendo aplicado corretamente. Os artefatos gerados devem ser armazenados no repositório do projeto e devem ser acompanhados pelos Gerentes de Configuração.