**Plano de Gerência de**

**Configuração de Software**

Sistema de Controle de Agendas

Cliente: IFMG – Instituto Federal de Minas Gerais – Campus Sabara

**Real Life**

**Versão: 0.01** | 20/06/2016

Responsáveis: Ubiratan Hilario e Lais Guimarães

**Histórico de Alterações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| **20/06/2016** | **0.01** | **Criação Inicial** | **Ubiratan Hilario** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Lista de Aprovadores**

|  |  |
| --- | --- |
| **Nome** | **Cargo** |
| **Lais Guimarães** | **Gerente de Projeto** |

**Conteúdo**

***1. Introdução***

**1.1Propósito**

**1.2Escopo**

**1.3Convenções, termos e abreviações**

**1.4Referências**

***2. Organização***

**2.1 Identificação de Documentos**

**2.1 Versão dos Documentos**

**2.2 Localização de Artefatos**

**2.3 Baselines do Projeto**

***3. Controle de Configuração***

**3.1 Procedimentos de Mudança**

**3.1.2 Ciclo de Vida das solicitações de mudança**

***4. Auditoria de Configuração***

***5. Plano de Contingência***

***6. Ferramentas***

**Introdução**

O Plano de Gerenciamento de Configuração estabelece e mantem a integridade de códigos-fonte e demais produtos do REALLIFE, permitindo o acompanhamento destes itens durante todo o ciclo de vida do projeto, e preservando o histórico de evolução do sistema. Auxiliando a gerenciar o estado dos itens de configuração dos sistemas, controlar as mudanças em itens de configuração e rastrear modificações nos itens de configuração ao longo do tempo.

* 1. **Propósito**

Este documento descreve a organização, nomenclatura e regras para o versionamento do projeto REALLIFE.

* 1. **Escopo**

Este documento descreve toda a infraestrutura utilizada durante o desenvolvimento do projeto REALLIFE.

* 1. **Convenções, termos e abreviações**

Esta seção explica o conceito de alguns termos importantes que serão mencionados no decorrer deste documento. Estes termos são descritos na tabela a seguir, estando apresentados por ordem alfabética.

|  |  |
| --- | --- |
| ***Termo*** | ***Descrição*** |
| **Baseline** | Conjunto de artefatos que recebe uma aprovação de estabilidade. Um *baseline* é usado como uma base no desenvolvimento das próximas fases dos artefatos e tem suas modificações controladas por um processo. |
| **CR** | Solicitação de Mudança (*Change Request*) |
| **CVS** | Sistema de Controle de Versão (*Control Version System*) |
| **SCMP** | Plano de Gerência de Configuração de Software  (*Software Configuration Management Plan*) |

1. **Organização**
   1. **Identificação de Documentos**

Todos dos itens de configuração, com exceção do código fonte, devem ser identificados baseados na nomenclatura descrita a seguir:

<PROJETO>\_<ID\_ARTEFATO>\_<DATA>

em que:

<PROJETO> é o nome do projeto;

<ID\_ARTEFATO> é a identificação do artefato;

<DATA> é a data de criação do artefato, ou seja, a data da primeira versão do artefato em questão. Para descrição da data o formato AAAAMMDD deve ser utilizado.

Por exemplo: REALLIFE\_SCM\_20070901

É importante salientar que todas as letras dos nomes que compõem os documentos devem estar em caixa alta.

A apresentação dos artefatos que podem ser gerados no processo e suas respectivas identificações.

|  |  |
| --- | --- |
| **Artefato** | **Identificação** |
| Casos de Uso | UC |
| Documento de Arquitetura | ARC |
| Documento de Requisitos | REQ |
| Plano de Gerência de Configuração | SCM |
| Plano de Teste de Software | STP |
| Status Report | ST |
| Relatório de Testes | TSR |
| Story Board (documento de telas) | STB |
|  |  |

* 1. **Versão dos Documentos**

O padrão de versionamento dos artefatos (exceto código e relatórios) deve ter um número de versão segundo o padrão descrito a seguir:

XX.YY

onde:

X é um número decimal que representa uma versão final do artefato;

YY é um número hexadecimal que representa um *draft* da versão X do artefato.

O número de versão dos artefatos muda de acordo com as regras descritas:

* A primeira versão do artefato deve ser 0.01;
* A cada modificação no artefato, o valor YY deve ser incrementado;

Após cada aprovação do artefato, a versão X deve ser incrementada de uma unidade e o valor YY retorna para 00, sendo assim gerada uma nova versão oficial;

Para que a versão de um artefato seja modificada é necessária a aprovação do moderador. É considerada uma aprovação do documento a aprovação do documento após uma revisão e aprovação por pelo menos dois membros do core group responsável pelo artefato respectivo.

* 1. **Localização de Artefatos**

A organização hierárquica do local onde serão armazenados os produtos do projeto, incluindo uma breve descrição do conteúdo de cada item da estrutura (pasta).

* 1. **Baseline do Projeto**

**As baselines geradas para o projeto:**

|  |  |  |
| --- | --- | --- |
| ***Baseline*** | ***Descrição*** | ***Padrão*** |
| **Requisitos** | Marcado assim que for concluída análise de requisitos da iteração. | RealLife\_REQ\_<interaçao> |
| **Análise e Projeto** | Marcado quando forem concluídos a análise e o projeto de cada iteração | RealLife\_ARCH\_<interaçao> |
| **Build** | Criada a cada build para o software | RealLife\_BUILD\_<build> |
| **Release** | Criado a cada release do software | RealLife\_RELEASE\_<versão> |
| **Documentos** | Criado após a aprovação de um documento. | RealLife\_NOME |
|  |  |  |
|  |  |  |

<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.

1. **Controle de Configuração**
   1. **Procedimentos de Mudança**

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.

* + 1. **Criando Solicitação de Mudanças**

As solicitações de mudanças devem ser criadas através da ferramenta de *issues* disponível no site do projeto https://github.com/LaisIFMG/Real-Life/. Para tanto o usuário deve estar logado como administrador. O preenchimento dos campos deve ser feito seguindo as orientações descritas a seguir. Devem ser aprovadas pelo *core team* responsável pelo artefato em questão.

|  |  |
| --- | --- |
| Campo | Valor |
| Summary | Breve descrição do problema |
| Description | Descrição detalhada do problema |
| Status | Open Statuses:  New- A issue ainda não teve sua revisão inicial  Accepted – problema reproduzido/precisa reconhecimento.  Started – o trabalho nesta issue começou  Closed Statuses:  Fixed – Desenvolvedor fez as mudanças pedidas, QA deve verificar.  Verified – QA verificou que o reparo foi trabalhado  Invalid – relatório não válido  Duplicate – este relatório duplica uma issue existente  WonFix – foi decidido não examinaresta issue |
| 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 issue, o sistema operacional.  Type-Defect – Relato de um defeito no artefato a ser mudado.  Type-Enhancement – Requerimento para melhoria no artefato.  Type-Task – os artigos trabalhados não requerem mudança no código ou nos docs.  Type-Pach – Correção do código fonte para revisão  Type-Other – Algum outro tipo de issue  Priority-Hign – Prioridade alta para a resolução do evento especificado.  Priority-Medium – Prioridade média para a resolução do evento especificado.  Priority-Low – Prioridade baixa para a resolução do evento especificado.  OpSys-All – Todos os Sistemas  Operacionais  OpSys-Windows – Sistema Operacional Windows  OpSys-Linux – Sistema Operacional Linux  OpSys-OSX – Sistemas Operacional OSX  Milestone-Release1.0 -  Component-UI – issue relaciona-se ao programa UI  Component-Logic – issue relaciona-se a lógica da aplicação.  Component-Persistence – issue relaciona-se ao componente de armazenamento de dados.  Component-Scripts – roteiro de utilização e instalação  Componet-Docs – issue relacionada a documentação do usuário final.  Security – segurança de risco dos usuários  Performance – issue de desenpenho  Usability – efeitos da usabilidade do programa  Maintainability – impedir mudanças futuras. |
|  |  |

**3.1.2 Ciclo de vida das solicitações de mudança**

Os possíveis estados, as possíveis soluções e o ciclo de vida de uma solicitação de mudanças. A seguir são descreve os estados possíveis para uma solicitação de mudanças.

1. **Auditoria de Configuração**

As auditorias de configuração devem ser realizadas para cada ciclo do processo de desenvolvimento de forma a garantir que os processos de gerência de configuração estão sendo aplicados corretamente. Os artefatos gerados devem armazenados no repositório do projeto e devem ser acompanhados pelos Gerentes do Projeto.

1. **Plano de Contingência**

Uma vez por semana será feito um backup da versão mais recente dos artefatos que se encontram no CVS na máquina de dois membros da fábrica participantes do projeto.

1. **Ferramentas**

Segue abaixo uma lista das ferramentas utilizadas no processo de desenvolvimento do projeto RealLife.

|  |  |
| --- | --- |
| **Ferramenta** | **Descrição** |
| **https://github.com/LaisIFMG/Real-Life** | Ambiente para promoção de código livre, interação com desenvolvedores e gerenciamento de projetos. |
|  |  |