
Modelo de Processo S.C.U.

Felipe Issamu de Melo Kamimura, Isabelle Ichikawa Yagi, Lucas Freitas Costa, Lucas Tarumoto, Nathiely Moraes Macedo, Raniel Carlos Bispo dos Santos

Universidade Tecnológica Federal do Paraná - Campus Cornélio Procópio

Este documento descreve a gerência de configuração do modelo de processo de desenvolvimento de Software S.C.U.

7 de dezembro de 2018

Lista de figuras

1	BPMN - Gerenciamento de Configuração	5
2	BPMN - Gerenciamento de Mudanças	6
3	BPMN - Gerenciamento de Versões	7
4	BPMN - Gerenciamento de Releases	7
5	Template de Formulário de Mudança	9
6	Template de Relatório de Mudança	10
7	Template Modelo de Solicitação de Mudança	11
8	Template de Relatório de Release	12

Sumário

1	Introdução	4
2	Requisitos	4
3	Novos Integrantes	4
4	Gerenciamento de Configurações	4
4.1	Gerenciamento de mudanças	5
4.1.1	Rastreamento de alterações	5
4.1.2	Controle de mudanças para itens de configuração	5
4.2	Gerenciamento de Versões	6
4.2.1	Codelines	6
4.2.2	Baselines	6
4.3	Integridade	7
4.4	Gerenciamento de releases	7
5	Referências bibliográficas	8
6	Anexos	8

1 Introdução

A parte de gerência de configuração é essencial em um processo de desenvolvimento de um projeto, utilizando métodos de controle de versões, controle de mudanças e auditoria das configurações é possível solidificar o projeto com integridade, completude nos requisitos e consistência do mesmo. Este processo utiliza a norma [1] para auxiliar na questão de mudanças e versionamento de projetos, para que tudo seja documentado e especificado da maneira correta com o intuito de simplificar o entendimento e o trabalho para equipes futuras, a execução de atividades foram feitas com base no Modelo S.C.U..

2 Requisitos

Os requisitos elicitados que atendem ao cenário proposto para solução seriam:

RF01 - O usuário deve ser capaz de cadastrar cidades

RF02 - o Sistema deve ser capaz de retornar todas as cidades

RF03 - Os dados cadastrados devem ser mantidos

RF04 - Qualquer alteração (remoção e edição) deve ser mantida atualizada

RF05 - O usuário deve ser capaz de remover cidades

RF06 - O usuário deve ser capaz de editar os atributos de cidades existentes

3 Novos Integrantes

Para novos integrantes (desenvolvedores) é necessário a utilização do GitHub para o desenvolvimento, nela será feita o controle de versionamento em repositórios na plataforma. Para isso os desenvolvedores devem fazer um fork do repositório, realizar as mudanças e requisitar um pull-request para o responsável pela coordenação da atividade ocorrente. Os processos de gerência de configurações serão explicados mais profundamente neste documento onde os novos desenvolvedores poderão ter acesso para o entendimento do mesmo. Por via das dúvidas sempre comunicar e repassar ao código para o gerente de projeto, para que este suba ao github caso aja algum conflito de configuração, como é possível ocorrer conflitos no desenvolvimento de de branches. Para isso é necessário possuir um computador com conexão à Internet, JDK Enviroment, Postgress SQL configurado e uma Ide como Netbeans/Eclipse. Para o projeto em questão o repositório a ser clonado é o <https://github.com/FelipeKamimura/ProdutoInova>.

4 Gerenciamento de Configurações

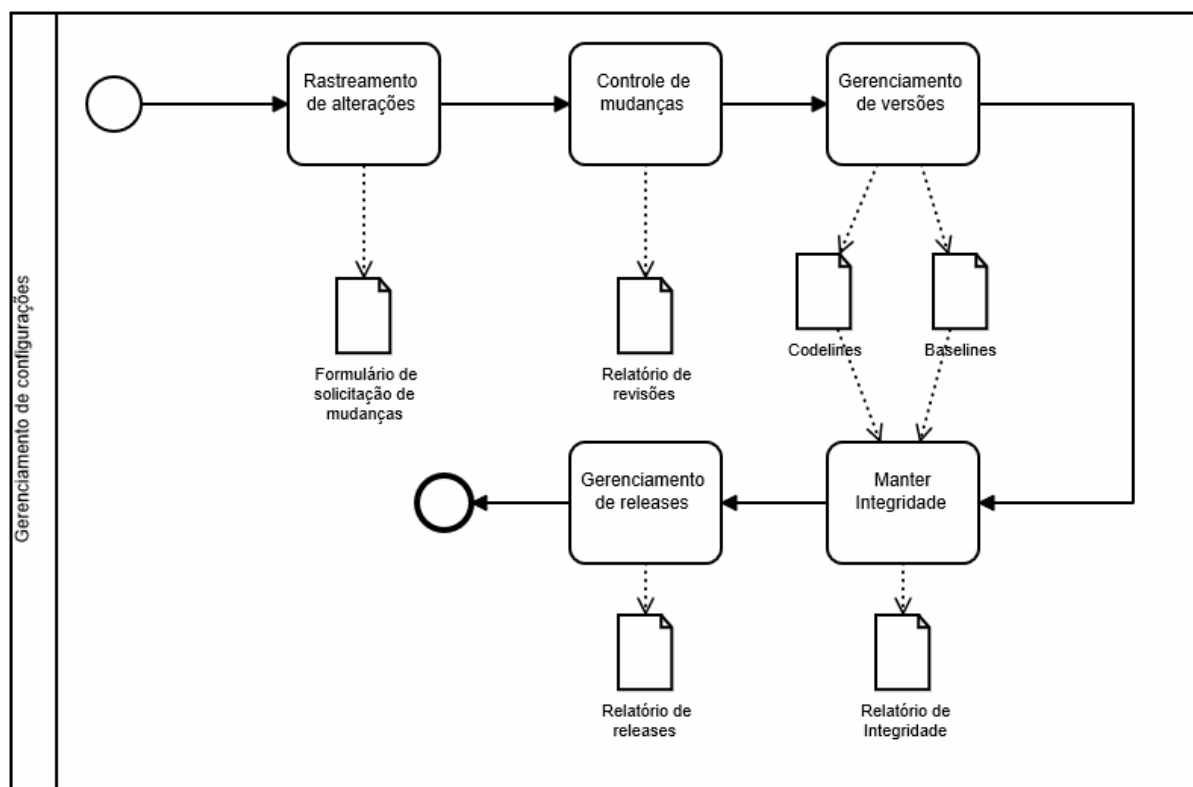


Figura 1: BPMN - Gerenciamento de Configuração

4.1 Gerenciamento de mudanças

O processo de mudança em muitos projetos é constante e determinam o sucesso ou fracasso do produto final esperado. As necessidades e requisitos organizacionais se alteram durante a vida útil de um sistema, bugs precisam ser reparados e os sistemas necessitam se adaptar às mudanças em seu ambiente[2]. Assim deve ser feito um levantamento de quais tipos de mudanças podem ocorrer no decorrer do desenvolvimento dos processos, se seriam relevantes e como teriam impacto no mesmo. Para isso é necessário o rastreamento dos pedidos de mudanças que serão guardados em um banco de dados que possuirá atribuições dos detalhes ditos anteriormente. Além do controle do mesmo para não haver inconsistências e erros que possam ocorrer. Para melhor especificação foi feito um diagrama BPMN mostrando o processo de gerenciamento de mudanças.

4.1.1 Rastreamento de alterações

As solicitações de alterações são analisadas para determinar o impacto que a alteração terá no produto de trabalho, nos produtos de trabalho relacionados, no cronograma e no orçamento. Assim serão categorizados e indexados para melhor estruturação e vai ser gravado em formulários que possuirão os dados relevantes das alterações.

4.1.2 Controle de mudanças para itens de configuração

Durante toda vida útil do item de configuração é feito controle das mudanças sobre o mesmo, portanto deve haver um arquivamento das baselines. Para começar, a revisão

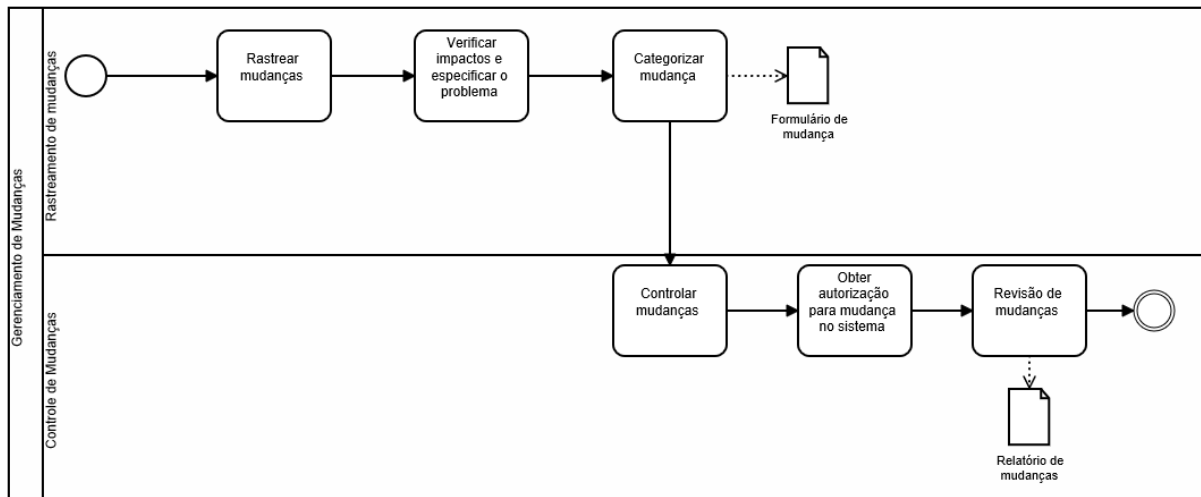


Figura 2: BPMN - Gerenciamento de Mudanças

histórica dos itens é indispensável para possuir uma base das mudanças, além disso revisões periódicas também são essenciais para não ocorrer efeito indesejados nos baselines (Serão gerados artefatos de relatórios para as revisões). Vão ser criados artefatos com as mudanças feitas nos itens e as razões do mesmo, no processo BPMN 2 será especificado mais a fundo como será feito este controle e será disponibilizado os templates.

4.2 Gerenciamento de Versões

O gerenciamento de versões (VM, do inglês version management) é o processo de acompanhamento de diferentes versões de componentes de software ou itens de configuração e os sistemas em que esses componentes são usados [2]. Para garantir que as mudanças realizadas nos artefatos por vários desenvolvedores não sofra interferência uma das outras, todas as versões dos componentes serão tratadas e documentadas como codelines e baselines. O diagrama pode ser visto na figura 3

Para o gerenciamento de versões foi utilizado o GitHub com 1 repositório <https://github.com/FelipeKamimura/ProdutoInova> e a organização das pastas ficou dividida da seguinte forma a pasta process para o documento do processo e a pasta projeto referente ao projeto.

4.2.1 Codelines

Codelines serão as sequências de versões dos códigos-fonte dos componentes que sofreram mudanças. Será utilizado o GitHub para fazer esse versionamento, podendo assim ver todas as versões anteriores e as diferenças entre uma e a outra.

4.2.2 Baselines

Codelines serão as sequências de versões dos códigos-fonte dos componentes que sofreram mudanças. Será utilizado o GitHub para fazer esse versionamento, podendo assim ver todas as versões anteriores e as diferenças entre uma e a outra.

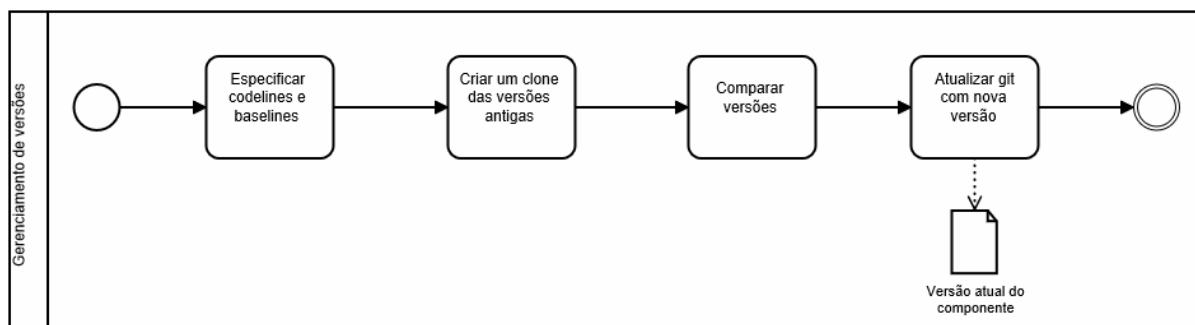


Figura 3: BPMN - Gerenciamento de Versões

4.3 Integridade

A integridade será garantida pelo gerenciamento de versões através do gitHub, pois nele há informações suficientes para verificar as alterações realizadas e verificar se uma mudança afetaria a integridade (no sentido de não corromper os requisitos)

4.4 Gerenciamento de releases

É normal um sistema grande ter diversos releases e a gerência de configurações tem o papel de mapear essas releases e saber que versão do sistema cada cliente tem, para que quando necessário essa release possa ser reconstruída. Para garantir a segurança, qualidade e disponibilidades dessa versão, que está sendo utilizada para cada cliente, é utilizado as baselines, codelines, assim como um relatório de releases para armazenar as bibliotecas utilizadas, componentes, assim como suas versões e as plataformas.(4)

No mais, os deploys aos clientes podem ser feitos de modo manual a partir do acesso ssh do servidor e instalado em cada um dos clientes, essa operação é viável por ainda ser um número reduzido de clientes. Para auxiliar pode ser utilizado o repositório <https://github.com/FelipeKamimura/SCU-deploy> onde as releases podem ser inseridas em seus respectivos clientes através de um clone do repositório.

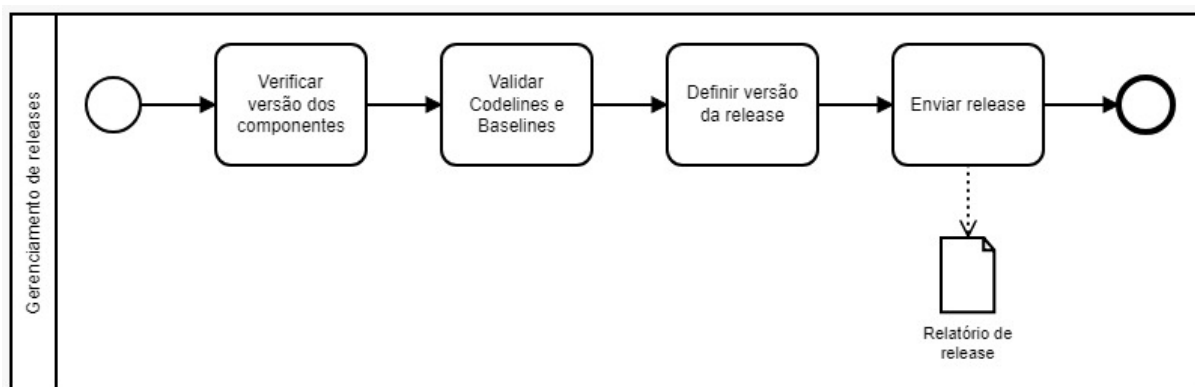


Figura 4: BPMN - Gerenciamento de Releases

5 Referências bibliográficas

- [1] C. P. TEAM, *CMMI® for Development, Version 1.3, Improving processes for developing better products and services*. Software Engineering Institute, 2010.
- [2] I. Sommerville, “Engenharia de software-8ª edicao (2007),” *Ed Person Education*.

6 Anexos

Formulário de Mudanças no Projeto
Nome do Produto
 Rev. x.x, dd/mm/aaaa

Projeto:	
Preparado por:	
Data (DD/MM/AAAA):	

2. Informações sobre impactos e riscos	
Descrição da Mudança:	
Linhas de Base Afetadas:	
Itens de Configuração Afetados (ex. Especificações do produto):	
Análise de Custo / Cronograma Requerida?	Sim [] Não []
Impacto em Custo:	
Impacto em Cronograma:	
Impacto em Recursos:	
Risco Associado com a Implementação da Mudança:	
Risco Associado com a NÃO Implementação da Mudança:	
Data da Revisão: (DD/MM/AAAA)	
Resultados da Análise:	

Figura 5: *Template de Formulário de Mudança*

Relatório para Mudanças no Projeto
Nome do Produto
 Rev. x.x, dd/mm/aaaa

Projeto:	
Preparado por:	
Data (DD/MM/AAAA):	

3. Informações sobre Relatório de Mudanças	
Mudança:	
Recomendações:	
Aprovar Mudança []	Desaprovar Mudança []
Assinaturas:	
Nome do Projeto:	
Gerente do Projeto:	
Eu revisei as informações sobre esta mudança e aprovo.	
Assinatura:	Data:
Eu revisei as informações sobre esta mudança e desaprovo.	
Assinatura:	Data:

Figura 6: *Template de Relatório de Mudança*

Formulário de Requisição de Mudanças no Projeto
Nome do Produto
 Rev. x.x dd/mm/aaaa

Projeto:	
Preparado por:	
Data (DD/MM/AAAA):	

1. Informações sobre a mudança solicitada		
Solicitada por:		
<input type="checkbox"/> Cliente	<input type="checkbox"/> Equipe de Desenvolvimento	<input type="checkbox"/> Gerente do Projeto
Área da Mudança:		
<input type="checkbox"/> Escopo	<input type="checkbox"/> Cronograma	
<input type="checkbox"/> Orçamento	<input type="checkbox"/> Qualidade	
<input type="checkbox"/> Implementação	<input type="checkbox"/> Design	
Sobre a Mudança e Referências:		
Mudança:		
Justificativa:		

Figura 7: *Template Modelo de Solicitação de Mudança*

Relatório de Release
Nome do Produto
Rev. ~~x.x~~, ~~dd/mm/aaaa~~

Projeto:	
Preparado por:	
Data (DD/MM/AAAA):	

Relatório de Release
Versão:
Bibliotecas Alteradas:
Componentes Alterados:

Figura 8: *Template de Relatório de Release*