

**GUIA PARA GERENCIAMENTO DE CONFIGURAÇÃO UTILIZANDO
COMO BASE A CARTILHA SOFTEX ORIENTADA PELO MPS-BR**

GRUPO 1

1. Introdução

Introdução retirada da cartilha softex disponível no link: https://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_de_Implementacao_SV_Parte_2_20132.pdf

Gerência de Configuração (GCO)

O propósito do processo Gerência de Configuração é estabelecer e manter a integridade de todos os produtos de trabalho de um processo ou trabalho e disponibilizá-los a todos os envolvidos. A Gerência de Configuração abrange, segundo [OGC, 2005], a identificação de todos os componentes significativos da infraestrutura do sistema de serviços e registra detalhes desses componentes no Banco de Dados de Itens de Configuração (BDIC). O sistema de Gerência de Configuração também registra a relação entre os componentes. Ela fornece informações completas a respeito de todos os componentes do sistema de serviços que permite a outros processos funcionarem de forma mais efetiva e eficiente. Em função da granularidade utilizada em um contexto de serviços, um item de configuração pode ser formado por um conjunto de artefatos ou componentes, bem como um único artefato ou componente pode ser formado por vários itens de configuração. No contexto da Gerência de Configuração para serviços, os componentes e os serviços fornecidos por eles são chamados de Itens de Configuração (IC) [ITSMF, 2005]. Esses ICs podem incluir computadores, todos os tipos de software, componentes de redes, servidores, documentação, procedimentos, serviços, e todos os demais componentes que necessitem de controle pela organização prestadora dos serviços. É comum referir-se a estes itens como ativos (assets) da organização. A Gerência de Configuração não se propõe a definir quando e como devem ser executadas as modificações nos produtos de trabalho, papel este reservado ao próprio processo de desenvolvimento ou manutenção do serviço. A sua atuação ocorre como processo auxiliar de controle e acompanhamento. O escopo do processo Gerência de Configuração não se aplica unicamente aos artefatos produzidos pelos processos de um determinado nível do MR-MPS-SV. Todos os artefatos dos processos de serviço em uso pela organização - sejam eles de desenvolvimento, manutenção ou apoio - são considerados. É importante notar que a Gerência de Configuração é um importante mecanismo para aumentar o controle sobre os produtos de trabalho. O processo Gerência de Configuração tem uma interseção com todos os demais processos do MR-MPS-SV por meio do atributo de processo RAP13, que estabelece: “os produtos de trabalho são colocados em níveis apropriados de controle”. Os níveis de controle podem variar de acordo com a importância ou criticidade

dos artefatos gerados, mas devem ser adequados a cada caso específico. Assim, para artefatos que requerem um controle mais formal, a Gerência de Configuração é aplicável, tanto no contexto de trabalhos como no contexto organizacional. Alguns artefatos, no entanto, podem ser armazenados com um simples controle de acesso ou, então, serem versionados sem necessidade de um controle formal de mudança. Definir quais artefatos ou componentes serão sujeitos a MR-MPS-SV – Guia de Implementação – Parte 2:2013 18/66 quais níveis de controle é parte da execução do processo de Gerência de Configuração. O processo Gerência de Configuração está intimamente relacionado com outros processos do MR-MPS-SV. Por exemplo: o processo Gerência de Trabalhos (GTR) pode apoiar no planejamento do processo Gerência de Configuração. O processo Gerência de Decisões (GDE), a partir do nível C, pode apoiar na atividade de avaliação de solicitações de modificação do processo Gerência de Configuração. A Gerência de Mudanças (GMU), a partir do nível E, usa o BDIC para identificar o impacto das mudanças a serem implementadas. A Gerência de Liberações (GLI) fornece informações a respeito de planos de liberação e a situação dos ICs, tanto para grandes, como para pequenas liberações. O processo Gerência de Configuração pode também apoiar o processo Gerência de Requisitos (GRE) no que diz respeito ao controle de modificações sobre os requisitos dos serviços e também o processo de Gerência de Nível de Serviço (GNS) no controle de modificações nos ANS.

Para o processo desenvolvido podemos destacar a gerência de configuração dos artefatos:

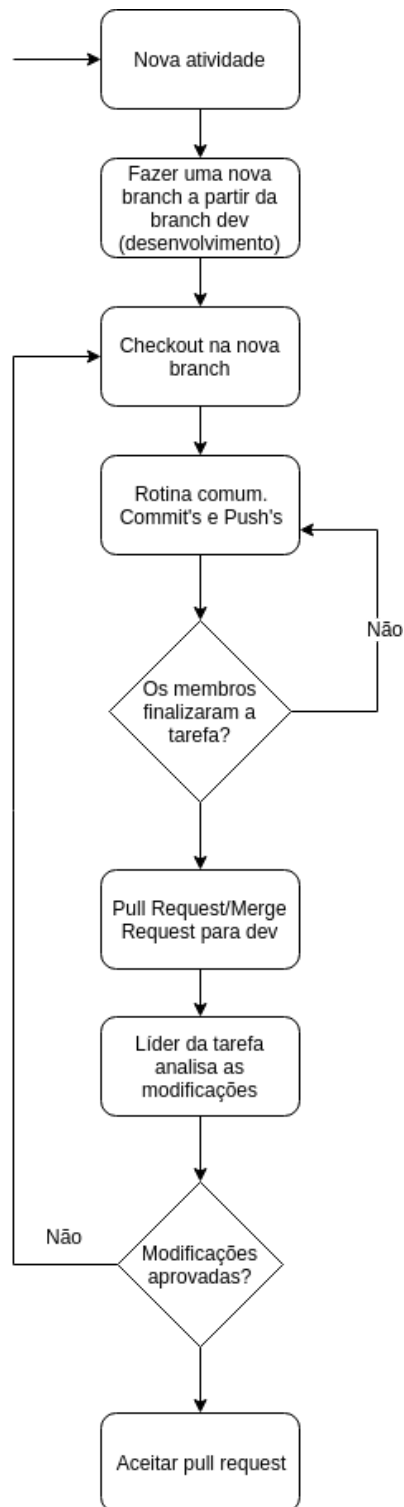
Artefato	Tipo de gerência de configuração
Documento de requisitos e testes	Versionado e representado por responsável.
Código fonte	Versionado e mantido em repositório (Git)
Contratos de trabalho	Versionado e representado por responsável.

◆ Visão Geral

Para todo o decorrer do processo de software foi idealizado um padrão de gerência de configuração, de modo que se alguma parte do processo for considerada inapta para este padrão, através dos líderes, tal exceção deverá

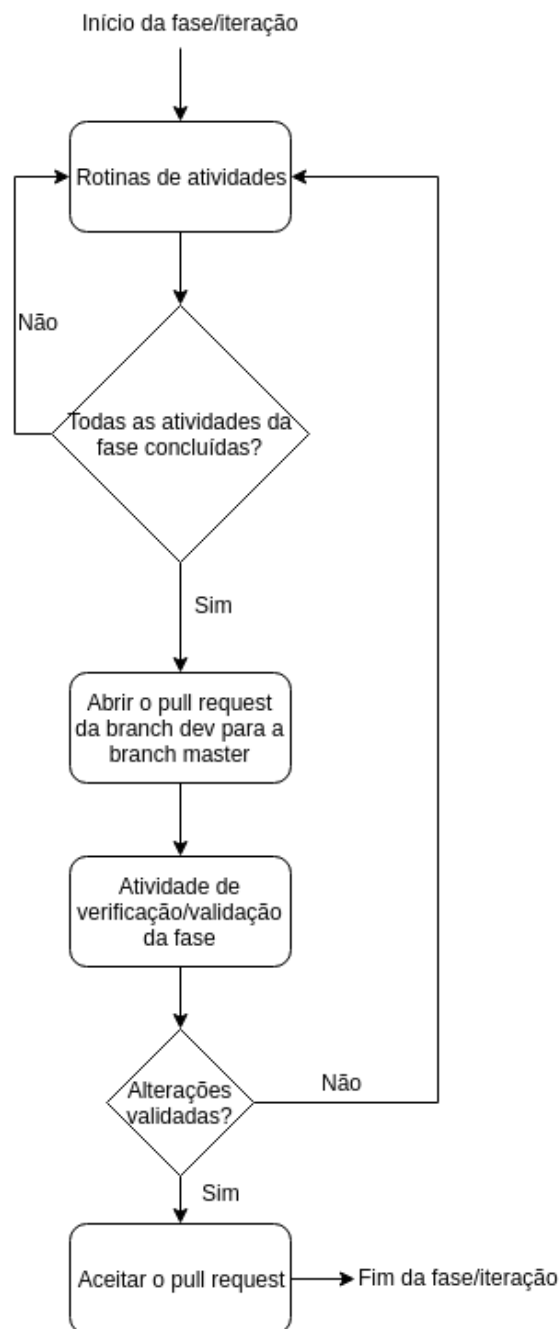
ser tratada do modo que a equipe julgue melhor, desde que o líder geral a aprove e a situação apresentada não esteja descrita neste documento.

Uma visão geral, pode ser observada logo abaixo, o que permite entender o processo que deverá ser seguido em todo o projeto.



A cada nova atividade deve ser criada uma nova branch a partida da branch dev. Os membros devem fazer o checkout e além disso estabelecer uma rotina, em que devem ser gerados commits e push's diários. Quando o(s) membro(s) finalizar(em) a tarefa, deverá ser aberto o pull request para a branch dev. O líder da tarefa analisará, e aceitará o pull request, caso não tenha sido este que tenha feito as alterações, caso for deve-se pedir para os integrantes da equipe e o líder superior a este analisar e tomar os devidos procedimentos.

Abaixo se encontra outro diagrama que representa como será abordada a gerência de configuração nas fases ou iterações.



Como pode ser observado, ao iniciar uma fase ou iteração, deve-se seguir as rotinas previamente definidas e especificadas no diagrama anterior. Quando todas as atividades estiverem concluídas, deve-se realizar um pull request da branch dev para a branch master, que em seguida passará pela etapa de verificação ou validação especificada em uma atividade da fase ou iteração. Caso tal validação seja aceita, deverá ser aceito o pull request para a master, finalizando a interação ou fase e gerando um novo deploy e em consequência um novo baseline para o projeto. Caso contrário, todas as modificações devem ser realizadas e o procedimento deverá ser realizado novamente.

❖ Código fonte

Este guia visa a utilização do Github para versionamento e armazenamento de código, respeitando alguns padrões bem como, nome de *branches*, padrões de comentários, *commits* e periodicidade de *push request*.

➤ Branchs

As branches base são: master e dev.

Sempre que uma nova funcionalidade precisa ser desenvolvida, deverá ser gerada uma nova branch com a nomenclatura nome-do-projeto-idFuncionalidade-nome-da-funcionalidade.

É recomendado que os commits e pushes sejam realizados diariamente para evitar conflitos e gargalos, bem como pulls da branch dev. Sempre que uma branch de novas funcionalidades for desenvolvida e entregue pelo desenvolvimento, deverá ser realizado o pull request desta para a branch de desenvolvimento, ou seja, a branch dev. Assim que a fase ou iteração for concluída, a branch dev pode ir para master se homologada na validação/verificação. Candidata porque o gerente de projetos pode rejeitar a branch por algum motivo técnico ou de negócio.

❖ Procedimentos para arquivos sigilosos

É importante ressaltar que algum cliente por ventura possa considerar um documento como sigiloso. Desse modo, se para aquele projeto não houver repositórios privados e/ou se não há repositório privado, onde os membros são os mesmos, deverá ser feito um novo, com o seguinte padrão de nomenclatura: nome-do-projeto-siglaEquipe1- siglaEquipe2-siglaEquipeX.

Tal repositório deve ser privado e todas as alterações devem passar por autorização da chefia máxima das equipes presentes. Caso necessário, pode ser impedido o uso de push e fork do repositório em questão, para os outros membros. Em caso de saída de algum membro da(s) equipe(s), este deve perder total acesso ao repositório em questão.

Por fim, quando for decidido que um novo repositório privado for feito, isto deverá conter a ata de uma reunião com os líderes, onde isto foi decidido. Tal ata não pode sofrer modificações.

❖ Containers

Para projetos maiores, onde o sigilo tem um nível alto e a equipe é consideravelmente grande, pode-se utilizar containers com regras específicas descritas no contrato com o cliente.

O uso destes containers possibilita a aplicação de mais regras, além do que o git e o GitHub são capazes de atender. Além disto, pode dificultar o uso mal intencionado por membros das equipes.

❖ Baseline

Como baseline é considerada a branch master, toda branch que for homologada e validada gerará uma nova baseline incrementado.

❖ Versionamento

O versionamento do projeto ocorrerá segundo as recomendações do Semantic Versioning (disponível em: <https://semver.org/>).

Logo, todas as versões irão possuir três números, da forma: X.X.X. O primeiro número diz a respeito do fim de uma fase ou de uma iteração, considerada uma mudança maior e atualização da baseline do projeto. O segundo número será incrementado a cada fim de atividade, sendo que irá reiniciar a contagem quando iniciada uma nova fase. O terceiro número corresponde a bug-fix, ou seja, correções que foram realizadas.

❖ Considerações

É evidente que não há sistema seguro em sua totalidade, sendo este ou não computacional. Porém, é importante se zelar pela segurança e também considerar todas as variáveis possíveis, não permitindo que uma destas deixe de ser atendida.