

## **Análise de Ferramentas para Controle de Versões de Software no Contexto do MPS.BR**

**Danne da Silva Oliveira<sup>1</sup>, Heitor Costa<sup>2</sup>, Paulo Afonso Parreira Júnior<sup>2</sup>**

<sup>1</sup> Instituto de Ciências Exatas - Universidade Federal de Goiás – Regional Jataí – Caixa Postal 03 – 75801-615 – Jataí-GO – Brasil

<sup>2</sup> Departamento de Ciência da Computação - Universidade Federal de Lavras Caixa Postal 3.037 – 37.200-000 – Lavras – MG – Brasil

{danneoliveirasoft}@gmail.com, {heitor, pauloa.junior}@dcc.ufla.br

***Abstract.** Configuration Management (CM) is one of MPS.BR processes, which deals with the management software versions. Software version control along its development cycle is not a simple task and the use of computational tools can impact positively or negatively on this control. Therefore, it is necessary to undertake an analysis of support tools for software version control in the literature to find out whether they are appropriate or not. This work consisted in the analysis of the software version control tools “Subversion”, “Git”, “Bazaar” and “Mercurial”, in order to check if they include the requirements specified by the CM process in MPS.BR model.*

### **1. Introdução**

MPS.BR (Melhoria de Processos do Software Brasileiro) é um modelo de maturidade de processos que tem como objetivo a melhoria de processos de desenvolvimento de software e serviços de organizações desenvolvedoras. O MPS.BR se divide em quatro modelos de referência, sendo que o enfoque deste trabalho está sobre o Modelo de Referência MPS para Software (MR-MPS-SW), que se baseia nas definições dos níveis de maturidade que uma organização pode assumir, bem como dos processos de software que ela deve contemplar. O MR-MPS-SW define sete níveis de maturidade, que estabelecem patamares de evolução de processos de desenvolvimento de software sequenciais e acumulativos. Esses níveis de maturidade vão do G (Parcialmente Gerenciado) ao A (Em Otimização). Para cada um desses sete níveis de maturidade é atribuído um perfil de processos, que apresenta os requisitos sobre os quais a organização deve colocar esforços para melhoria de seu processo de desenvolvimento.

Dentre os processos existentes no MPS.BR, destaca-se o processo de “Gerência de Configuração (GCO)”, que pertence ao nível F na evolução dos níveis de maturidade do MPS.BR. Tal processo tem o propósito de estabelecer e manter a integridade de todos os produtos de trabalho de um processo ou projeto e disponibilizá-los a todos os envolvidos. Segundo Sommerville (2011), a GCO trata do gerenciamento das versões de um software. Em outras palavras, a GCO engloba as atividades de identificação e controle das mudanças, garantia de efetividade da implementação das mudanças e divulgação das mudanças aos interessados (Pressman, 2001). Segundo Brito e Yoshidome (2006), o controle das versões de um software ao longo de seu ciclo de desenvolvimento não é uma tarefa simples e o uso de ferramentas computacionais pode impactar positiva ou negativamente neste controle. Sendo assim, faz-se necessário

realizar uma análise das ferramentas de apoio à GCO existentes na literatura, em particular, ao controle de versões do software, para verificar se as mesmas são adequadas ou não. Uma forma de se fazer isso é descobrir se elas contemplam os requisitos exigidos pelos processos dos modelos de maturidade de processo, como por exemplo, o MPS.BR.

O objetivo principal deste trabalho é apresentar uma análise comparativa das principais ferramentas de apoio ao controle de versões existentes na literatura, tais como *SubVersion*, *Git*, *Bazaar* e *Mercurial*, com o intuito de descobrir se elas atendem ou não aos requisitos do processo de Gerência de Configuração do modelo MPS.BR. Este trabalho possui enfoque sobre o modelo de maturidade MPS.BR, pois ele é um modelo nacional voltado às empresas de micro e pequeno porte. Além disso, uma certificação MPS.BR custa em torno de quatro vezes menos do que a certificação para modelos internacionais, como o *Capability Maturity Model Integration* (CMMI).

Na Seção 2 são apresentados os principais trabalhos relacionados. Na Seção 3 estão os critérios utilizados para analisar as ferramentas para controle de versões, assim como os resultados da análise de tais ferramentas. Na Seção 4, estão as considerações finais e as propostas para trabalhos futuros.

## **2. Trabalhos Relacionados**

O trabalho de Furlaneto (2006) apresenta o desenvolvimento de uma ferramenta de Gerência de Configuração, com base no estudo de outras ferramentas similares como *SubVersion* e *Trac* e pesquisas realizadas sobre o MPS.BR. Segundo Furlaneto (2006), a ferramenta desenvolvida se mostrou bem aderente aos resultados previstos no modelo MPS.BR. Para chegar a esta conclusão, ele avaliou as funcionalidades de sua ferramenta, com base nos resultados esperados do processo de GCO do MPS.BR. Entretanto, as principais diferenças entre o trabalho proposto por Furlaneto (2006) e a proposta deste projeto de monografia são: (i) Furlaneto avalia apenas a ferramenta desenvolvida por ele, com base nos resultados esperados do MR-MPS-SW. Este trabalho propõe a avaliação de diversas ferramentas para GCO existentes na literatura; e (ii) além disso, não ficou claro como os resultados esperados foram aplicados para a avaliação da ferramenta proposta por ele.

Já o trabalho desenvolvido por Oliveira (2007) apresentou uma lista de requisitos que podem guiar a escolha de ferramentas para Gerência de Configuração. Para gerar tal lista, Oliveira se utiliza de livros sobre Engenharia de Software (Sommerville, 2011; Pressman, 2001), do Corpo de Conhecimento em Engenharia de Software (SWEBOK) e de algumas normas IEEE. As principais diferenças do trabalho de Oliveira (2007) com relação a esta proposta são: (i) não utiliza o Guia de Referência MPS para Software (MR-MPS-SW), do MPS.BR; e (ii) apesar de Oliveira apresentar um conjunto de critérios para guiar a escolha das boas ferramentas para GCO, ele não avalia nenhuma ferramenta existentes na literatura, ficando assim, a cargo do profissional realizar esta análise para descobrir qual a ferramenta mais adequada. Nesta proposta, além de apresentar os critérios para comparação das ferramentas, será realizada uma análise sobre as principais ferramentas para GCO existentes na literatura.

O presente trabalho tem como intuito a criação de um mapeamento entre os resultados esperados do processo de GCO do MR-MPS-SW e os critérios a serem utilizados para comparação das ferramentas de apoio à GCO.

### 3. Análise das ferramentas para controle de versões

Por questão de espaço, as Tabelas 1 a 6 apresentam um resumo dos critérios confeccionados a partir dos resultados esperados do processo de GCO do modelo MPS.BR para avaliação de ferramentas de controle de versões de software. Mais detalhes sobre esses critérios podem ser encontradas em Oliveira (2014). A primeira coluna destas tabelas contém o código de cada critério e a segunda coluna apresenta uma descrição de cada critério.

**Tabela 1. Critérios Relacionados ao Resultado Esperado GCO1.**

GCO 1 – Um sistema de Gerência de Configuração é estabelecido e mantido	
Código	Breve Descrição
C.1.1	A ferramenta de Controle de Versão permite armazenamento e recuperação dos itens de configuração em suas diversas versões de forma a preservar e atualizar seu conteúdo.
C.1.2	A ferramenta de Controle de Versão gerencia os ramos ( <i>branches</i> ) das versões de um software.
C.1.3	A ferramenta de Controle de Versão gera relatórios gerenciais que possibilitem fazer um balanço da configuração existente.
C.1.4	A ferramenta de Controle de Versão mantém uma estrutura de pastas com controle de acesso e de manuseio.
C.1.5	A ferramenta de Controle de Versão permite a realização de procedimentos de preservação dos dados (backup).

**Tabela 2. Critérios Relacionados ao Resultado Esperado GCO2.**

GCO 2 – Os itens de configuração são identificados com base em critérios estabelecidos	
Código	Breve Descrição
C.2.1	A ferramenta de Controle de Versão, no caso dos itens de configuração ser muito grande, ela limita as possibilidades de gerência.

**Tabela 3. Critérios Relacionados ao Resultado Esperado GCO3.**

GCO 3 – Os itens de configuração sujeitos a um controle formal são colocados sob <i>baseline</i>	
Código	Breve Descrição
C.3.1	A ferramenta de Controle de Versão evita modificações que seja feitas sem a devida aprovação.

**Tabela 4. Critérios Relacionados ao Resultado Esperado GCO4.**

GCO 4 – A situação dos itens de configuração e das <i>baselines</i> é registrada ao longo do tempo e disponibilizada	
Código	Breve Descrição
C.4.1	A ferramenta de Controle de Versão aplica registro de inclusão.
C.4.2	A ferramenta de Controle de Versão aplica registro de alteração de itens no repositório.
C.4.3	A ferramenta de Controle de Versão identifica as diferenças entre duas versões de um mesmo item de configuração.
C.4.4	A ferramenta de Controle de Versão utiliza o mecanismo de ramos ( <i>branches</i> ) em seus controles de versões.
C.4.5	A ferramenta de Controle de Versão no nível gerencial é possível visualizar precisamente o andamento das modificações realizadas.

**Tabela 5. Critérios Relacionados ao Resultado Esperado GCO5.**

GCO 5 – Modificações em itens de configuração são controladas	
Código	Breve Descrição
C.5.1	A ferramenta de Controle de Versão analisa o impacto das modificações e notifica e descreve quais itens de configuração serão afetados pela modificação.
C.5.2	A ferramenta de Controle de Versão atribui solicitações aos responsáveis pelas mudanças.
C.5.3	A ferramenta de Controle de Versão aplica retirada ( <i>check-out</i> ) nos itens de configuração.
C.5.4	A ferramenta de Controle de Versão aplica registro ( <i>check-in</i> ) nos itens de configuração.

**Tabela 6. Critérios Relacionados ao Resultado Esperado GCO6.**

GCO 6 – O armazenamento, o manuseio e a liberação de itens de configuração e <i>baselines</i> são controlados	
Código	Breve Descrição
C.6.1	A ferramenta de Controle de Versão estabelece conexão à internet para estabelecer canais de segurança.

Para aplicação dos critérios elaborados, as seguintes ferramentas para controle de versões de software foram consideradas: 1. Bazaar Explorer (Versão 1.2.2); 2. Git

(Versão 1.11.1); 3. Mercurial (Versão 1.34.1.42.1); e 4. SubVersion (Versão 1.32.1.42.1). Tais ferramentas foram escolhidas por estarem dentre as mais utilizadas no mercado, segundo recente relatório de tecnologias e ferramentas Java, produzido pelo laboratório *Rebellabs*, da empresa *Zereturnaround* (2016). A Tabela 7 apresenta os resultados das avaliações realizadas, exibindo a porcentagem de critérios contemplados por cada ferramenta. Os dados desta tabela foram obtidos de acordo com as avaliações feitas por três especialistas em ferramentas para controle de versão de software.

Na primeira e na segunda colunas desta tabela é possível visualizar o código do critério, bem como a qual resultado esperado do processo de GCO ele se refere; das colunas 3 à 6, estão os resultados para cada ferramenta analisada, com base na seguinte legenda: (X) a ferramenta não atende ao critério em análise; (O) a ferramenta atende parcialmente ao critério em análise; e (√) a ferramenta atende totalmente ao critério em análise. Da sétima à nona coluna, apresenta-se a porcentagem de ferramentas analisadas que atendem totalmente, parcialmente e não atendem aos critérios analisados. As últimas três linhas da Tabela 7 apresentam a porcentagem de critérios atendidos totalmente, parcialmente e não atendidos para cada ferramenta analisada.

**Tabela 7. Resultado da Avaliação das Ferramentas.**

Resultado Esperado	Critérios	Ferramentas				Total (%)		
		Bazaar	Git	Mercurial	SubVersion	X	O	√
GCO 1	C.1.1	√	√	√	√	0,0	0,0	100,0
	C.1.2	√	√	√	√	0,0	0,0	100,0
	C.1.3	X	X	X	X	100,0	0,0	0,0
	C.1.4	X	O	X	X	75,0	25,0	0,0
	C.1.5	X	X	X	X	100,0	0,0	0,0
GCO 2	C.2.1	X	X	X	X	100,0	0,0	0,0
GCO 3	C.3.1	√	√	√	√	0,0	0,0	100,0
GCO 4	C.4.1	√	√	√	√	0,0	0,0	100,0
	C.4.2	√	√	√	√	0,0	0,0	100,0
	C.4.3	√	√	√	√	0,0	0,0	100,0
	C.4.4	√	√	√	√	0,0	0,0	100,0
	C.4.5	O	√	O	√	0,0	50,0	50,0
GCO 5	C.5.1	X	X	X	X	100,0	0,0	0,0
	C.5.2	√	√	√	√	0,0	0,0	100,0
	C.5.3	√	√	√	√	0,0	0,0	100,0
	C.5.4	√	√	√	√	0,0	0,0	100,0
GCO 6	C.6.1	X	√	X	O	50,0	25,0	25,0
Subtotal X (%)		35,29	23,52	35,29	29,41			
Subtotal O (%)		5,88	5,88	5,88	5,88			
Subtotal √ (%)		58,83	70,60	58,83	64,71			

Legenda: X = Não atende; O = Atende Parcialmente; √ = Atende Totalmente.

Para gerar os resultados da Tabela 7, levou-se em consideração o seguinte procedimento: caso dois avaliadores escolhessem uma das opções X, O ou √, esta opção seria transcrita na Tabela 7; no caso de empate, a opção O foi transcrita para a tabela. Como pode ser visto nesta tabela, nenhuma das ferramentas atende aos critérios C.1.3, C.1.5, C.2.1 e C.5.1. Esse é um ponto crítico a ser considerado, uma vez que o não atendimento deste critério pelas ferramentas de controle de versões pode levar as empresas que utilizam estas ferramentas a terem resultado indesejados, quanto à avaliação de seu processo de GCO.

As ferramentas de controle de versões analisadas não atendem ao critério C.1.3, uma vez que elas não possuem modo de apresentação de relatórios específicos sobre as versões do software. As ferramentas apenas abordam *logs* e histórico das versões; com isto, há certa dificuldade de se fazer um balanço da configuração existente do software. Quanto ao critério C.1.5, não foi possível observar nas ferramentas analisadas, mecanismos de *backup* dos arquivos do software. Foi notado ainda uma limitação das

ferramentas ao lidarem com o versionamento de itens de configuração grandes (critério C.2.1). Por fim, quanto ao critério C.5.1, notou-se que as ferramentas analisadas não permitem analisar o impacto das modificações a serem realizadas no software e não há notificações para o usuário quanto aos itens de configuração que serão afetados pela modificação. Por fim, como pode ser observado na Tabela 7, a ferramenta que atende a maior quantidade de critérios (70,6%) é a ferramenta *Git*; já a ferramenta *SubVersion* é a segunda melhor ferramenta, atendendo à 64,71% dos critérios. Já as ferramentas *Bazaar* e *Mercurial* atendem a 58,83% dos critérios. Sendo assim, a escolha entre *Git* e *SubVersion* por parte dos usuários, ficará restrita às características específicas de cada uma, que cada usuário julga atender melhor às suas necessidades. Em outras palavras, considerando o processo de GCO definido pelo MPS.BR, bem como seus resultados esperados, o usuário encontrará um nível de adequação próximo à 100% em qualquer uma destas ferramentas.

#### 4. Considerações Finais

Neste trabalho, realizou-se a análise de quatro ferramentas para controle de versões de software, com base em alguns resultados esperados do processo de GCO, proposto no modelo MPS.BR. Essa análise foi realizada com o auxílio de três especialistas em ferramentas de controle de versões. Espera-se que, com essa análise, pesquisadores e profissionais possam escolher de forma mais adequada a ferramenta que mais atenda à sua necessidade. Como trabalhos futuros pretende-se: (i) verificar a possibilidade de avaliar outras ferramentas para controle de versão para determinar se os resultados esperados pelo MPS.BR são atendidos; (ii) criar novos critérios para a possibilidade de abranger os processos de Manutenção, Implantação, Auditoria entre outros processos do MPS.BR; e (iii) avaliar também a usabilidade das ferramentas para controle de versão, a fim de saber se elas facilitam o trabalho de seus usuários.

#### Referências

- BRITO, C. J. A.; YOSHIDOME E. Uma Análise Avaliativa de Ferramentas de Software Livre no Contexto da Implementação do Processo de Gerência de Requisitos do MPS.BR. WER. 2006. Cuenca, Equador.
- FURLANETO, R. Ferramenta de Apoio a Gerencia de Configuração de Software. Monografia de Especialização. Universidade Regional de Blumenau. Blumenau, 2006.
- OLIVEIRA, N. P. V. Requisitos de Ferramentas de Gerenciamento de Configuração. UFMG. Belo Horizonte, 2007. Disponível em: <http://homepages.dcc.ufmg.br/~rodolfo/dcc823-2-07/Entrega4/Viviane4.pdf>. Acesso: Mar/2015.
- OLIVEIRA, D. S. Análise de Ferramentas para Controle de Versões de Software no Contexto do Processo de Gerência da Configuração do MPS.BR. Monografia de graduação. UFG/Regional Jataí. 2014.
- PRESSMAN, Roger S. Software Engineering - A practitioner's Approach. 5ª Ed., McGraw-Hill, 2001, p. 253.
- SOFTEX - Sociedade para Promoção da Excelência do Software Brasileiro. Guia Geral MPS de Software, dezembro 2012. Disponível em: [http://www.softex.br/wp-content/uploads/2013/07/MPS.BR\\_Guia\\_Geral\\_Software\\_2012.pdf](http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_Geral_Software_2012.pdf). Acesso: Mar/2015.
- SOMMERVILLE, I. Engenharia de Software. 9. Ed. São Paulo: Pearson Prentice Hall, 2011.
- ZEROTURNAROUND. Java Tools and Technology Landscape Report. Disponível em: <http://zeroturnaround.com/rebellabs/java-tools-and-technologies-landscape-2016>. Acesso: Ago/2016.