Modelo de Referência de Gerência de Configuração para um Processo Ágil de Reengenharia baseado em *Framework*

Marliane Aldivina Oliveira Ferreira, Maria Istela Cagnin

UNIVEM – Centro Universitário Eurípides de Marília

Marília, São Paulo, Brasil, Caixa Postal 2041, CEP 17525-901

marlianeferreira@gmail.com, istela@univem.edu.br

Abstract. *Configuration Management (CM) is one of the manners to guarantee software quality. Several quality models which concern about CM have been considered, among them it is distinguished the MR.MPS reference model. However, there is a lack in literature related to reference models in reengineering context. Under this perspective, this paper aims to define a CM reference model, called MR.GC-PARFAIT, for the PARFAIT reengineering agile process. The model has been defined by means of three stages (to select essential CM activities, to select essential CM artifacts and to establish reference model applicability in PARFAIT activities), this being so the method GQM has been used in the two first stages. Reference model documentation is based on documentation structure of RUP (Rational Unified Process) and it is presented in this paper.*

Resumo. *Gerência de Configuração (GC) é uma das maneiras de garantir a qualidade do software. Diversos modelos de qualidade que se preocupam com GC têm sido propostos, dentre eles destaca-se o modelo de referência MR-MPS. No entanto, há carência na literatura relacionada a modelos de referência no contexto de reengenharia. Sob essa perspectiva, este artigo tem como objetivo definir um modelo de referência de GC, denominado MR.GC-PARFAIT, para o processo ágil de reengenharia PARFAIT. O modelo foi definido por meio de três etapas (selecionar as atividades essenciais de GC, selecionar os artefatos essenciais de GC e estabelecer a aplicabilidade do modelo nas atividades do PARFAIT), sendo que nas duas primeiras etapas empregou-se o método GQM. A documentação do modelo é baseada na estrutura da documentação do RUP (Rational Unified Process) e está apresentada neste artigo.*

1. Introdução

Solicitações de mudanças durante o ciclo de vida do software são inevitáveis. Uma das maneiras de permitir que tais mudanças ocorram sem alterar a qualidade presente no software é por meio do emprego da Gerência de Configuração (GC) (Pressman, 2002), que tem como principal objetivo controlar as mudanças e as versões do software (Sommerville, 2003). Isso não é diferente durante o desenvolvimento e a reengenharia incremental, pois versões do novo sistema e do sistema alvo, respectivamente, são liberadas a cada iteração do processo. A problemática se torna maior quando se utiliza *frameworks* (Fayad *et al.,* 1999) como apoio computacional do processo para gerar o

sistema, uma vez que é necessário controlar tanto as versões dos *frameworks* como também a dos sistemas gerados.

No contexto de processos de reengenharia baseados em *frameworks*, tem-se o processo PARFAIT (**P**rocesso **Á**gil de **R**eengenharia baseado em ***F****ramework* no domínio de sistemas de **I**nformação com técnicas de VV&**T)** (Cagnin *et al.,* 2003), que trata implicitamente as atividades de GC.

Sob a perspectiva de modelos de referência que abordam a GC tem-se o Modelo de Referência para Melhoria do Processo de Software (MR-MPS) (Softex, 2006). No entanto, até o momento não foi encontrado um modelo de referência na literatura pesquisada específico para a atividade de GC no contexto de processos de reengenharia baseados em *frameworks*.

Diante do exposto, observou-se a necessidade de criar um modelo de referência para tratar de maneira sistemática a GC do PARFAIT, o qual está apresentado detalhadamente neste artigo. Uma vez que o PARFAIT é caracterizado como um processo ágil, devem-se tomar precauções durante a definição desse modelo para não prejudicar a sua agilidade. Este artigo está organizado em seis seções: na Seção 2 apresenta-se uma visão geral do processo PARFAIT para o entendimento da aplicabilidade do modelo de referência definido. Na Seção 3 aborda-se a preparação para a definição do modelo de referência, que é composta por três etapas (seleção das atividades de GC para compor o modelo, seleção dos artefatos de GC e aplicabilidade das atividades selecionadas de GC nas atividades do PARFAIT). Na Seção 4 apresenta-se a documentação do modelo de referência, denominado MR.GC-PARFAIT. Na Seção 5 apresenta-se como trabalho correlato o modelo de referência MR-MPS sob a perspectiva de GC. E, finalmente, na Seção 6 discutem-se as conclusões obtidas e sugestões de trabalhos futuros.

2.PARFAIT: Processo Ágil de Reengenharia baseado em *Framework*

PARFAIT (Cagnin *et al.,* 2003) é um processo ágil, incremental e iterativo, que visa migrar sistemas legados procedimentais para o paradigma orientado a objetos. Esse processo é um dos recursos disponíveis no Arcabouço de Reengenharia Ágil (ARA), que possibilita a reengenharia ágil baseada em linguagens de padrões de análise (Appleton, 1997), para apoiar o entendimento do domínio do sistema legado, e em *frameworks* (Fayad *et al.,* 1999), para apoiar a geração do sistema alvo.

O ARA oferece apoio na criação de uma versão do sistema alvo o mais rápido possível, conta também com a participação dos clientes durante todo o projeto de reengenharia para validar os artefatos produzidos e também utiliza "reengenharia guiada por teste". Nessa última prática, casos de teste são criados com o apoio de critérios de teste funcional (Myers, 2004) que são executados no sistema legado para apoiar a engenharia reversa no entendimento das funcionalidades e na identificação de regras de negócio e, posteriormente, no sistema alvo para validá-lo. Salienta-se que as funcionalidades presentes no sistema legado podem ser evoluídas de acordo com as necessidades dos usuários.

No PARFAIT há reúso em vários níveis de abstração (análise, projeto, implementação, teste e manutenção), que é proporcionado indiretamente pelo arcabouço ARA, como discutido por Cagnin (2005).

A documentação de todas as atividades do PARFAIT é baseada no formato da documentação do RUP (*Rational Unified Process*) (Kruchten, 2001), mas no contexto de reengenharia. Este formato é composto por fases (Concepção, Elaboração, Construção e Transição), atividades, passos, artefatos desenvolvidos, apoio computacional, diretrizes, entre outros. A modelagem dos diagramas é feita em UML (*Unified Modelling Language*).

Deve-se ressaltar que no final da iteração de cada fase do processo existe um marco de referência que tem como objetivo observar o cumprimento do plano de projeto estabelecido na Fase de Concepção e estabelecer a viabilidade de dar continuidade ou não ao projeto (Cagnin, 2003).

PARFAIT trata apenas de alguns itens da GC de forma manual, assim observou-se neste artigo a necessidade de aperfeiçoá-lo sob essa perspectiva. Dentre os itens de GC tratados pelo PARFAIT tem-se o controle de adaptações implementadas manualmente no código fonte das versões do sistema alvo. Essas adaptações são realizadas em cada iteração do processo, a fim de tornar o sistema alvo funcionalmente equivalente ao sistema legado. Sem esse controle, essas adaptações são perdidas em subseqüentes instanciações do *framework*. Quando o *framework* GREN é utilizado como apoio computacional do PARFAIT, a ferramenta *GREN–WizardVersionControl* (Cagnin *et al.*, 2004a) é empregada para permitir esse controle. Apesar do PARFAIT se preocupar com a identificação dos itens de configuração logo no início da aplicação do processo e ressaltar a importância de controlar as versões de todos os artefatos produzidos durante a reengenharia, não mostra como aplicar a GC de maneira sistemática.

3.Preparação para a Definição do Modelo de Referência de GC do PARFAIT

A elaboração do modelo de referência foi obtida por meio de três etapas, de acordo com a Figura 1: as atividades e os artefatos de GC (Sommerville, 2003) essenciais para compor o modelo de referência proposto foram selecionados nas etapas um e dois, respectivamente. Posteriormente, na etapa três, identificou-se a aplicabilidade das atividades de GC, selecionadas para o modelo de referência, nas atividades do PARFAIT. Ressalta-se que as etapas foram realizadas incrementalmente com o intuito de refinar o modelo de referência durante sua definição.

Para apoiar as duas primeiras etapas empregou-se o método GQM (*Goal Question Metric*) (Basili *et al.,* 1994). O método GQM tratou de forma quantitativa a seleção das atividades e dos artefatos de GC nas duas primeiras etapas, por meio de um plano de mensuração com a utilização de questões apropriadas e com a aplicação da métrica "Grau de Importância", definida em Ferreira e Cagnin (2006). Essa métrica tem como objetivo auxiliar na seleção das atividades e dos artefatos de GC essenciais para o processo PARFAIT, de acordo com a importância desses, sem afetar a agilidade de tal

processo. Salienta-se que devido a isso, somente os principais objetivos da GC são atendidos pelo modelo de referência proposto.

Os possíveis valores resultantes após a aplicação da métrica "Grau de Importância" são: grau 5 - extremamente importante (de 100% a 80%), grau 4 - importante (de 79% a 60%), grau 3 – importância moderada (de 59% a 40%), grau 2 – pouco importante (de 39% a 20%) e grau 1 - sem importância (de 19% a 0%). A definição do percentual durante a aplicação da métrica é baseada na resposta das questões formuladas durante a aplicação do GQM. Os resultados das etapas um e dois são detalhados em Ferreira e Cagnin (2006).

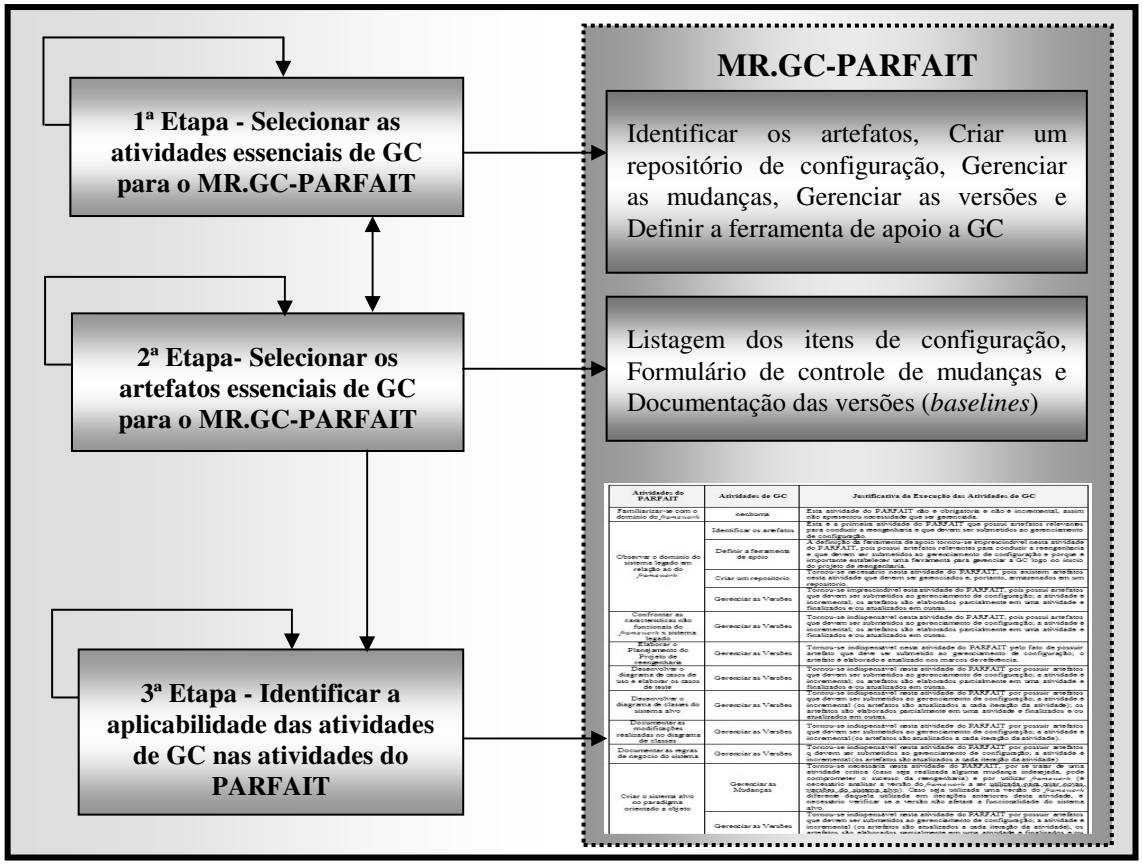

Figura 1 – Visão geral da definição do MR.GC-PARFAIT

Para apoiar a terceira etapa foi realizada uma análise da documentação do processo PARFAIT a fim de selecionar as atividades desse processo que devem ser submetidas a GC com apoio do MR.GC-PARFAIT (apresentado na próxima seção). A análise realizada foi baseada em três critérios: 1) se determinada atividade do processo é incremental (conseqüentemente, os artefatos são elaborados inicialmente e posteriormente refinados), 2) se determinada atividade do processo produz artefatos relevantes para conduzir a reengenharia e que devem ser submetidos ao gerenciamento de configuração, e 3) se os artefatos produzidos em uma determinada atividade não incremental são atualizados em outras atividades ou nos marcos de referência.

O resultado da análise realizada está apresentado no Quadro 1. Neste quadro está estabelecida a relação entre as atividades do processo PARFAIT, as atividades de GC

selecionadas para o modelo de referência deste processo e a justificativa do emprego das atividades de GC em uma determinada atividade do PARFAIT, de acordo com os critérios citados anteriormente.

Quadro 1 - Atividades do PARFAIT x Atividades de GC

Atividades do PARFAIT	Atividades de GC	Justificativa da Execução das Atividades de GC
Familiarizar-se com o domínio do *Framework*	Nenhuma	Esta atividade do PARFAIT não é obrigatória e não é incremental, assim não apresentou necessidade de ser gerenciada.
Observar o domínio do sistema legado em relação ao do *Framework*	Identificar os artefatos	Esta atividade do PARFAIT satisfaz os critérios 1 e 2. A ordem das atividades de GC foi determinada de acordo com os artefatos de entrada necessários para a execução de cada atividade (pré-requisito).
	Definir a ferramenta de apoio	
	Criar um repositório	
	Gerenciar as Versões	
Confrontar as características não funcionais do *Framework* x sistema legado	Gerenciar as Versões	Esta atividade do PARFAIT satisfaz os critérios 1 e 2.
Elaborar o Planejamento do Projeto de reengenharia	Gerenciar as Versões	Esta atividade do PARFAIT satisfaz os critérios 1 e 3.
Desenvolver o diagrama de casos de uso e elaborar os casos de teste	Gerenciar as Versões	Esta atividade do PARFAIT satisfaz os critérios 1 e 2.
Desenvolver o diagrama de classes do sistema alvo	Gerenciar as Versões	Esta atividade do PARFAIT satisfaz os critérios 1 e 2.
Documentar as modificações realizadas no diagrama de classes	Gerenciar as Versões	Esta atividade do PARFAIT satisfaz os critérios 1 e 2.
Documentar as regras de negócio do sistema	Gerenciar as Versões	Esta atividade do PARFAIT satisfaz os critérios 1 e 2.
Criar o sistema alvo no paradigma orientado a objeto	Gerenciar as Mudanças	Esta atividade do PARFAIT satisfaz os critérios 1 e 2.
	Gerenciar as Versões	Esta atividade do PARFAIT satisfaz os critérios 1 e 2.
Executar os casos de teste no sistema alvo	Gerenciar as Versões	Esta atividade do PARFAIT satisfaz os critérios 1 e 2.
Adaptar o sistema alvo	Gerenciar as Mudanças	Esta atividade do PARFAIT satisfaz os critérios 1 e 2. Além disso, no passo "Executar o sistema alvo concomitantemente com o sistema legado" desta atividade do PARFAIT é necessário analisar se as mudanças solicitadas pelos usuários são importantes e devem ser efetuadas no sistema alvo.
	Gerenciar as Versões	Esta atividade do PARFAIT satisfaz os critérios 1 e 2.
Elaborar o manual do usuário do sistema alvo	Gerenciar as Versões	Esta atividade do PARFAIT satisfaz os critérios 1 e 2.
Converter a base de dados do sistema legado	-	Não satisfaz nenhum critério, pois não possui artefato para ser gerenciado
Testar o sistema alvo	Gerenciar as Versões	Esta atividade do PARFAIT satisfaz os critérios 1 e 2.
Treinar os usuários finais	-	Não satisfaz nenhum critério pois não possui artefato para ser gerenciado

Ressalta-se que a atividade de GC "Gerenciar mudanças" foi necessária na atividade do PARFAIT "Criar o sistema alvo no paradigma orientado a objeto" por ser

considerada crítica (uma vez que caso seja realizada alguma mudança indesejada no sistema alvo, pode comprometer o sucesso da reengenharia) e por utilizar *framework* (sendo necessário analisar a versão do *framework* a ser utilizada para criar novas versões do sistema alvo). Caso seja utilizada uma versão do *framework* diferente daquela utilizada em iterações anteriores desta atividade, é necessário verificar se a versão não afetará o comportamento do sistema alvo.

4. MR.GC-PARFAIT: Modelo de Referência de GC do PARFAIT

O formato da documentação do MR.GC-PARFAIT definido neste artigo segue aquele utilizado no RUP: atividade, objetivo, artefatos de entrada e de saída, apoio computacional, passos e gabaritos. Além disso, um item foi adicionado na documentação, que é "aplicação no PARFAIT", ou seja, em que momento do processo PARFAIT (atividade) determinada atividade de GC do modelo de referência deve ser executada.

Sugere-se que o responsável pela reengenharia determine um membro da equipe como gerente de configuração, que deve ter conhecimento e habilidade de executar as atividades impostas pelo MR.GC-PARFAIT. Nas subseções a seguir são apresentadas as atividades do modelo MR.GC-PARFAIT. A execução das atividades "Identificar os itens de configuração" e "Definir a ferramenta de apoio" não são obrigatórias. Pois, para a atividade "Identificar os itens de configuração", MR.GC-PARFAIT sugere uma listagem dos itens de configuração do PARFAIT que devem ser submetidos a GC. A não obrigatoriedade da execução da atividade "Definir a ferramenta de apoio" está relacionada a familiaridade do responsável com determinada ferramenta de controle de versão.

Nas subseções a seguir apresenta-se a documentação completa das atividades do modelo de referência proposto.

4.1. Atividade: Identificar os itens de configuração (Não Obrigatória)

- *Objetivo:* Nesta atividade é necessário selecionar apenas os artefatos do PARFAIT que devem ser submetidos ao gerenciamento de configuração. Para facilitar o uso do modelo de referência criado e a não execução desta atividade, é sugerida uma listagem dos artefatos que devem ser gerenciados.

- *aplicação no PARFAIT:* Na atividade "Observar o domínio do sistema legado em relação ao do *framework*" (Fase de Concepção).

- *Artefatos de entrada:* Documentação do PARFAIT.

- *Apoio computacional:* Editor de texto.

- *Artefatos de saída:* Listagem dos itens de configuração do PARFAIT.

- *Passos:*

1: Listar todos os artefatos do PARFAIT e elencá-los de acordo com o seu grau de importância (aplicar métrica "Grau de Importância") para documentar e conduzir a reengenharia;

2: Analisar a lista de artefatos priorizados, de acordo com o grau de importância;

3: Listar os artefatos que devem ser submetidos ao gerenciamento de configuração.

-*Gabarito:* Para selecionar quais artefatos produzidos pelo PARFAIT devem ser considerados no gerenciamento de configuração com o apoio do modelo de referência proposto utilizou-se o GQM. A abrangência da métrica "Grau de Importância" foi reduzida para simplificar a seleção: grau 3 – extremamente importante (100% a 67%), grau 2 – importância moderada (66% a 34%) e grau 1 – pouco importante (33% a 0%).

O estabelecimento do grau de importância para cada artefato do PARFAIT é resultante da resposta da seguinte questão formulada com a aplicação do GQM: a agilidade do PARFAIT não é prejudicada pelo gerenciamento do artefato X[1]? Para apoiar a definição do percentual, tem-se a métrica intermediária: percentual de necessidade da utilização do artefato X para criar o sistema alvo a partir da instanciação do *framework*. O resultado obtido está apresentado no Quadro 2, que mostra a listagem dos artefatos elaborados durante o uso do PARFAIT com grau de importância igual a 3, ou seja, artefatos extremamente importantes e que devem ser gerenciados pelo modelo proposto.

Quadro 2 - Listagem dos Itens de Configuração do PARFAIT

Itens de Configuração do PARFAIT
Documento de aceitação do *framework*
Documento de requisitos
Planejamento do projeto de reengenharia
Documentação dos casos de teste
Diagrama de Classes do sistema
Diagrama de Casos de Uso
Quadro das adaptações no diagrama de classes não cobertas pela linguagem de padrões
Quadro dos requisitos da linguagem de padrões que não se adaptam completamente aos do sistema legado
Documentação de regra do negócio
Sistema alvo no paradigma orientado a objeto
Documentação de casos de teste adequados ao sistema alvo
Formulário de planejamento das adaptações
Manual do usuário
Sistema alvo no paradigma orientado a objetos liberado para uso

4.2. Atividade: Definir a Ferramenta de Apoio (Não obrigatória)

- *Objetivo:* Nesta atividade define-se a ferramenta que melhor apóia a GC, devendo ser adequada às necessidades impostas tanto para controlar as versões do *framework* quanto para controlar as versões dos sistemas gerados.

- *Aplicação no PARFAIT:* Na atividade "Observar o domínio do sistema legado em relação ao do *framework"* (Fase de Concepção).

- *Artefatos de entrada:* Documentação das características de ferramentas de GC.

- *Apoio computacional:* Editor de texto.

- *Artefatos de saída:* Ferramenta que melhor se adequou às necessidades impostas.

- *Passos:*

[1] Cada artefato produzido pelo PARFAIT.

1: Observar e analisar, dentre as ferramentas encontradas, os pontos positivos e negativos de cada uma, levando em consideração o controle de versão tanto do *framework* quanto do sistema alvo.

2: Escolher a ferramenta de GC mais apropriada.

4.3. Atividade: Criar um Repositório de Configuração (Obrigatória)

- *Objetivo:* Após a definição dos artefatos na atividade "Identificar os itens de configuração", estes serão armazenados e gerenciados em um repositório de configuração com o apoio da ferramenta de controle de versão selecionada.

- *Aplicação no PARFAIT:* Na atividade: "Observar o domínio do sistema legado em relação ao do *framework*" (Fase de concepção).

- *Artefatos de entrada:* Artefatos selecionados na atividade "Identificar os itens de configuração".

- *Apoio computacional:* Ferramenta de controle de versão.

- *Artefatos de saída:* Repositório criado.

- *Passo:*

1: Criar o repositório na ferramenta de controle de versão selecionada.

4.4. Atividade Gerenciar as Mudanças (Obrigatória)

- *Objetivo:* Esta atividade é empregada somente quando existir a necessidade de mudanças no sistema alvo ou no *framework*, para que sejam tomadas as medidas necessárias para que ocorram as mudanças sem prejudicar a integridade das *baselines* desses itens de configuração.

- *Aplicação no PARFAIT:* Nas atividades "Criar o sistema alvo no paradigma orientado a objetos" (Fase construção) e "Adaptar o sistema alvo" (Fase de construção).

- *Artefatos de entrada:* Formulários de Controle de Mudanças do *Framework* e do Sistema Alvo

- *Apoio computacional:* Editor de Texto e Ferramenta de Controle de Versão.

- *Artefatos de saída:* Formulários de Controle de Mudanças do *Framework* e do Sistema Alvo devidamente preenchidos e aprovados/reprovados.

- *Passos:*

1: Identificar as mudanças necessárias;

2: Preencher os Formulários de Controle de Mudanças do *Framework* (caso seja necessário) e do Sistema Alvo (caso seja necessário);

3: Aprovar ou reprovar as mudanças;

4: Realizar as mudanças e executar a atividade "Gerenciar as Versões".

- *Gabarito:* As solicitações de mudanças no sistema alvo e no *framework* devem ser sempre documentadas por meio do formulário de controle de mudanças, apresentados respectivamente nos Quadros 3 e 4. As solicitações de mudanças no sistema alvo serão aprovadas ou não pelo gerente de configuração, de acordo com a complexidade das

mesmas. Para analisar tal complexidade devem-se levar em consideração quais artefatos serão afetados pela mudança e se os mesmos necessitam de adaptações ou correções. No caso do PARFAIT com o apoio computacional do *framework* GREN, a implementação das adaptações (mudanças) realizadas manualmente no código fonte do sistema alvo é feita com o apoio da ferramenta GREN–WizardVersionControl a fim de que tais adaptações não sejam perdidas nas próximas instanciações do *framework*. As solicitações de mudanças no *framework* serão aprovadas ou não de acordo com uma análise, que verifica se uma mudança é pertinente ou não ao domínio do *framework*. Essa análise está fora do escopo deste trabalho e é feita com o apoio do Processo de Evolução de *Frameworks* (PREF) (Cagnin *et al*., 2004b). As mudanças no framework são efetuadas também com o apoio desse processo.

Quadro 3 - Gabarito do Formulário de Controle de Mudanças no Sistema Alvo

<table>
<tr><th colspan="3">Formulário de Controle de Mudanças no Sistema Alvo</th></tr>
<tr><td colspan="2">Nº da solicitação: <nº da solicitação da mudança></td><td>Projeto: <nome do projeto></td></tr>
<tr><td colspan="2">Data da solicitação: <dd/mm/aa></td><td>Tempo estimado para a execução da mudança <hh:mm></td></tr>
<tr><td colspan="2">Sistema alvo: <nome do sistema alvo></td><td>Versão do sistema alvo: <número correspondente da versão do sistema alvo que necessita sofrer mudanças></td></tr>
<tr><td colspan="3">Tipo de mudança: ☐ Perfectiva ☐ Corretiva ☐ Adaptativa ☐ Preventiva</td></tr>
<tr><td colspan="3">Mudança solicitada: <breve descrição da mudança></td></tr>
<tr><td colspan="3">Artefatos afetados pela mudança solicitada:</td></tr>
<tr><td colspan="3"><lista dos nomes dos artefatos que serão afetados e que sofrerão mudanças></td></tr>
<tr><td colspan="2">☐ Solicitação de mudança APROVADA</td><td>☐ Solicitação de mudança NÃO APROVADA</td></tr>
<tr><td>☐ Implementar a mudança com apoio do <i>Framework</i>:
- por meio de nova instanciação do <i>framework</i> (nesse caso o <i>framework</i> fornece a mudança desejada, parcialmente ou comple-tamente)</td><td>☐ Implementar a mudança manualmente:
- por meio de um ambiente de programação</td><td>Retornar ao solicitante um parecer sobre o motivo da não aprovação da solicitação
<motivo que levou a inviabilidade das mudanças></td></tr>
<tr><td colspan="3">Comentários Adicionais: < breve descrição, se necessário></td></tr>
</table>

Quadro 4 - Gabarito do Formulário de Controle de Mudanças no *Framework*

<table>
<tr><th colspan="3">Formulário de Controle de Mudanças no Framework</th></tr>
<tr><td colspan="2">Nº da solicitação: <nº da solicitação da mudança></td><td>Projeto: <nome do projeto></td></tr>
<tr><td colspan="2">Data da solicitação: <dd/mm/aa></td><td>Tempo estimado para a execução da mudança <hh:mm></td></tr>
<tr><td colspan="3"><i>Framework</i>: <nome do <i>framework</i>></td></tr>
<tr><td colspan="3">Versão do <i>framework</i>: <número correspondente da versão do <i>framework</i>></td></tr>
<tr><td colspan="3">Tipo de mudança: ☐ Perfectiva ☐ Corretiva ☐ Adaptativa ☐ Preventiva</td></tr>
<tr><td colspan="3">Mudança solicitada: <breve descrição da mudança></td></tr>
<tr><td colspan="3">Artefatos afetados pela mudança solicitada:</td></tr>
<tr><td colspan="3"><lista dos nomes dos artefatos que serão afetados e que sofrerão mudanças></td></tr>
<tr><td colspan="2">☐ Solicitação de mudança APROVADA</td><td>☐ Solicitação de mudança NÃO APROVADA</td></tr>
<tr><td>☐ Implementar a mudança no <i>framework</i></td><td>☐ Não implementar a mudança no <i>framework</i></td><td>Retornar ao solicitante um parecer sobre o motivo da não aprovação da solicitação <motivo que levou a inviabilidade das mudanças></td></tr>
<tr><td colspan="3">Comentários Adicionais: < breve descrição, se necessário></td></tr>
</table>

4.5. Atividade: Gerenciar as Versões (Obrigatória)

- *Objetivo:* Nesta atividade são gerenciadas as versões do *framework,* as versões do sistema alvo gerado a partir de tal *framework*, bem como as versões dos demais itens de configuração. Essa atividade é conduzida pelos envolvidos no projeto de reengenharia. Especificamente para o caso das versões do *framework* são documentadas as particularidades apresentadas em cada versão, como por exemplo, plataforma, necessidades de hardware e software, linguagem de programação, entre outras.

- *Aplicação no PARFAIT:* em todas as atividades das fases de Elaboração e Construção e em algumas atividades da fase de Concepção e Transição: "Observar o domínio do sistema legado em relação ao do *framework"* (Fase de Concepção)*, "*Confrontar as características não funcionais do *framework* x sistema legado" (Fase de Concepção), "Elaborar o planejamento do projeto de reengenharia" (Fase de Concepção); "Elaborar o manual do usuário do sistema alvo" (Fase de Transição), "Testar o sistema alvo" (Fase de Transição).

- *Artefatos de entrada:* Documentação das particularidades das versões do *framework* e dos sistemas gerados, bem como a dos demais itens de configuração.

- *Apoio computacional:* Editor de texto e ferramenta de controle de versão.

- *Artefatos de saída:* Documentação das versões do *framework*, do sistema alvo, dos demais itens de configuração (*baseline*).

- *Passos:*

1: Documentar de forma detalhada as peculiaridades apresentadas em cada versão do *framework*, em cada versão do sistema alvo gerado a partir do mesmo e em cada versão dos demais itens de configuração;

2: Armazenar a versão do item de configuração no repositório da ferramenta de controle de versão.

- *Gabaritos:* Nesta atividade foram desenvolvidos gabaritos para documentar as versões dos itens de configuração. Para documentar as versões do sistema alvo e do *framework* foram criados formulários específicos, apresentados nos Quadros 5 e 6, respectivamente. Para documentar as versões dos demais itens de configuração selecionados na atividade "Identificar os itens de configuração" é necessário utilizar o formulário apresentado no Quadro 7. Salienta-se que serão documentadas apenas as versões dos itens de configuração que se tornarem *baseline*. Além de informações específicas sobre a versão, é necessário registrar também o número da iteração do projeto de reengenharia que o item de configuração foi criado e/ou modificado, bem como o nome da atividade em que isso ocorreu.

Quadro 5 - Gabarito da Documentação das Versões do Sistema Alvo

Documentação das Versões do Sistema Alvo	
Documentação da Versão <número da versão do sistema> **do Sistema Alvo** <nome do sistema alvo>	
Projeto	**Data de criação do formulário**
<nome do projeto>	<dd/mm/aa – hh:mm>
Versão do *framework* utilizada	
<número de versão do *framework* >	
Variabilidades funcionais do *framework* reutilizadas	**Variabilidades Não-Funcionais do *framework* reutilizadas**
<lista dos nomes das variabilidades funcionais>	< lista dos nomes das variabilidades não-funcionais>
Requisitos funcionais do sistema não reutilizados do *framework*	**Requisitos não-funcionais do sistema não reutilizados do *framework***
<lista dos nomes dos requisitos funcionais >	<lista dos nomes dos requisitos não-funcionais>

Quadro 6 - Gabarito da Documentação das Versões do *Framework*

Documentação das Versões do *Framework*	
Documentação da Versão <número da versão do *Framework* > **do *Framework*** <nome do *Framework* >	
Plataforma	**Data de criação do formulário**
<nome da plataforma utilizada>	<dd/mm/aa – hh:mm>
Domínio	**Linguagem de programação**
<nome do domínio coberto pelo *framework*>	<nome da linguagem de programação utilizada>
Recursos de hardware e software requeridos	
<por exemplo, ambiente de desenvolvimento, versão do compilador da linguagem de programação, sistema gerenciador de banco de dados (SGBD), versão do SGBD>	
Variabilidades funcionais	**Variabilidades não-funcionais**
<lista dos nomes das variabilidades funcionais>	<lista dos nomes das variabilidades não-funcionais>

Quadro 7 - Gabarito da Documentação das Versões dos demais Itens de Configuração

Documentação das Versões dos Demais Itens de Configuração				
Nome do Projeto		**Data de criação do formulário**		
<nome do projeto>		<dd/mm/aa – hh:mm>		
Data de criação ou atualização do item	**Nome dos itens de configuração**	**Versão**	**Número da iteração**	**Nome da atividade do PARFAIT**
...	**...**	**...**	**...**	**...**

5.Trabalhos Relacionados

O projeto Melhoria de Processo de Software Brasileiro (MPS.BR) (Softex, 2006) criou o modelo de referência MR-MPS que tem como objetivo adaptar à realidade brasileira, modelos existentes para melhoria de processos de software, como *CMMI* (SEI, 2001), *ISO/IEC* 12207 (1998) e *ISO/IEC* 15504 (1999), a um custo acessível para as empresas de desenvolvimento de software. Com isso, tais empresas poderão aumentar sua competitividade no mercado nacional e internacional.

A GC se encontra no nível F dos níveis de maturidade do modelo MR.MPS. O propósito desse modelo sob a perspectiva de GC é estabelecer e manter a integridade dos artefatos. Os objetivos de GC apresentados no modelo MPS.BR são divididos em quatorze: 1) selecionar os artefatos com o emprego de critérios documentados; 2) delegar responsabilidades para o desenvolvimento de cada artefato e *baseline;* 3) identificar, armazenar, testar, revisar, alterar, definir, manter os artefatos e submetê-los a *baseline*; 4) instituir uma política de GC que controle os diferentes níveis, que possua um armazenamento dos artefatos para posteriormente serem recuperados, bem como uma solicitação de mudanças; 5) preservar todos os artefatos de GC; 6) autorizar a

criação ou a liberação das *baselines* pela GC; 7) controlar as modificações e as liberações dos artefatos; 8) disponibilizar as modificações e as liberações dos artefatos; 9) registrar, relatar e analisar o impacto das solicitações de mudanças que os artefatos podem sofrer e sua situação; 10) assegurar com as revisões, que as mudanças ocorridas nas *baselines* não sofram efeitos inesperados; 11) adicionar os artefatos, já aprovados e devidamente controlados; 12) assegurar a consistência e a completeza dos artefatos; 13) controlar o armazenamento, o manuseio e a liberação dos artefatos; 14) estabelecer e manter a integridade das *baselines*, com os registros e a auditoria de GC.

Apesar desse modelo de referência tratar da GC, não é voltado para o contexto de reengenharia baseada em *framework* e também não apresenta detalhes de como executar cada atividade proposta. Assim, o modelo MR.MPS apresentou-se como um guia sobre "o que" deve ser feito e não "como".

Diante do exposto, observou-se carência na literatura de modelos de referência para o contexto de reengenharia, mais especificamente, para apoiar a GC. Essa carência aliada a problemática de controle de versões de *framework* e dos sistemas gerados e da necessidade desse controle no PARFAIT motivou a definição do MR.GC-PARFAIT.

6.Conclusão e Trabalhos Futuros

O modelo MR.GC-PARFAIT foi proposto com o intuito de explicitar as atividades de GC durante a aplicação do processo PARFAIT. Para compor tal modelo sem afetar a agilidade desse processo, foi realizado um estudo quantitativo a fim de selecionar as atividades e artefatos de GC adequadamente (Ferreira e Cagnin, 2006). Para analisar o modelo definido neste artigo, será conduzido um estudo de caso planejado de reengenharia, de acordo com Wholin *et al.* (2000). O objetivo de tal estudo é analisar se tal modelo não afeta a agilidade do processo e se apóia efetivamente o controle de versões do *framework* e dos sistemas gerados, mantendo a integridade dos mesmos. Adicionalmente, nesse estudo de caso, será analisada a necessidade de outras atividades no modelo de referência proposto e até mesmo a adequação de algumas das atividades apresentadas.

O maior desafio encontrado neste trabalho foi a elaboração da documentação do MR.GC-PARFAIT. Ressalta-se que não foi encontrado nenhum modelo de referência na literatura que descrevesse com detalhes os procedimentos adotados pela GC em processos de reengenharia e nem em processos ágeis de reengenharia. Conclui-se que este modelo está voltado exclusivamente para o processo PARFAIT, mas isso não impede de ser adaptado para outros processos de reengenharia.

Referências

Appleton, B. (1997). Patterns and software: Essential concepts and terminology. http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html, Janeiro/2006.

Basili, V.R.; Caldiera, G.; Rombach, H.D. (1994). Goal Question Metric paradigm. Encyclopedia of Software Engineering, John Wiley & Sons.

Cagnin, M. I.; Maldonado, J. C.; Germano, F. S.; Penteado, R. D. (2003). PARFAIT: Towards a *Framework*-based Agile Reengineering Process. In: I Agile Development Conference, Salt Lake City, UTHA. IEEE.

Cagnin, M. I.; Maldonado, J. C.; Germano, F. S.; Penteado, R. D.; Braga, R. T. (2004a). GREN-WizardVersionControl: Uma Ferramenta de Apoio ao Controle de Versão das Aplicações Criadas pelo *Framework* GREN. In: Sessão de Ferramentas do XVIII Simpósio Brasileiro de Engenharia de Software, Brasília, DF.

Cagnin, M. I.; Maldonado, J. C.; Masiero, P. C.; Penteado, R. D.; Braga, R. T. (2004b). An Evolution Process for Application *Frameworks*. In: I Workshop de Manutenção Moderna de Software, em conjunto com o XVIII Simpósio Brasileiro de Engenharia de Software, Brasília, DF, CD-ROM, 8 p., 2004.

Cagnin, M. I. (2005). PARFAIT: uma contribuição para a reengenharia de software baseada em linguagem de padrões e *frameworks***.** Tese de doutorado do Instituto de Ciências Matemáticas e de Computação – ICMC/USP, São Carlos.

Fayad, M., Schmidt, D., Johnson, R. (1999). Building Application *Frameworks*: Object-Oriented Foundations of *Framework* Design. John Wiley & Sons, September.

Ferreira, M.A.O.; Cagnin, M.I. (2006). Proposta de um Modelo de Referência de Gerência de Configuração para um Processo de Reengenharia baseado em *Framework*. In: III Simpósio Brasileiro de Sistemas de Informação, Curitiba-PR.

ISO/IEC 12207 (1998). Tecnologia de Informação - Processos de ciclo de vida de software. ABNT - Associação brasileira de normas técnicas Rio de Janeiro: ABNT.

ISO/IEC 15504 – 5 (1999). Information Technology – Software process assessment – Part 5: An assessment model and indicator guidance. ISO/IEC - International Standard Organization and International Electritechnical Commission.

Kruchten, P. (2000). The Rational Unified Process and Introduction Second Edition. Addison Wesley Longman: NJ.

Myers, G. J. (2004). The art of software testing. Second Edition. Wiley.

Pressman, R. S. (2002). Engenharia de Software. 5ª ed., Rio de Janeiro: McGraw-Hill.

SEI - Software Engineering Institute. (2006). Capability Maturity Model® Integration (CMMI^{SM}). Version 1.1, 2001, http://www.sei.cmu.edu/cmmi/models/model-components-word.html, Fevereiro.

Softex (2006). MPS.BR – Melhoria de Processo do Software Brasileiro (Guia Geral – Versão 1.0). http://www.softex.br/mpsbr/_guias/MPS.BR_Guia_Geral_V1.1.pdf, Janeiro.

Sommerville, I. (2003). Engenharia de Software. 6ª ed., São Paulo: Addison Wesley.

Wholin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., e Wesslén, A. (2000). Experimentation in Software Engineering: an Introduction. Kluwer Academic Publishers.