

# Uma Análise Avaliativa de Ferramentas de Software Livre no Contexto da Implementação do Processo de Gerência de Requisitos do MPS.BR

Alexandre Brito Cardias Junior<sup>1</sup>, Luciana Neves Bentes<sup>1</sup>, Mauricio Ronny<sup>1</sup>, Sandro Ronaldo Bezerra Oliveira<sup>1</sup>, Ewelton Yoshidome<sup>1</sup>

<sup>1</sup>Faculdade de Computação – Instituto de Ciências Exatas e Naturais - Universidade Federal do Pará (UFPA) - Brazil

{alexandrecardias, lufpa06}@gmail.com, mauricio.ronny@uol.com.br, srbo@ufpa.br, dai\_battousai\_kenshin@hotmail.com

## Abstract

*The market's demands intensify the search for quality in software development and, as a result, it makes that even more companies adopt models in order to improve their software processes and, consequently, their final product. In addition to this, the time for product delivery can be significantly reduced when proper tools for a quicker evolution of the process are adopted. This paper presents free software tools, analyzing them from the point of view of the MPS.BR model.*

## Resumo

*As exigências do mercado intensificam a busca por qualidade no desenvolvimento de software e fazem com que cada vez mais empresas adotem modelos visando à melhoria dos seus processos de software e consequentemente do seu produto final. Paralelo a isto está o tempo para a entrega do produto que pode ser sensivelmente reduzido quando adotado ferramental apropriado para que o processo evolua de forma mais rápida. Este trabalho se propõe a apresentar ferramentas de software livre analisando-as do ponto de vista da implementação do processo gerência de requisitos do modelo para Melhoria do Processo de Software Brasileiro, MPS.BR.*

## 1. Introdução

A qualidade tornou-se um diferencial no mercado, e não é diferente no desenvolvimento de *software*, pois ela está diretamente ligada à satisfação do cliente, ao atendimento dos requisitos desse *software*. Assim, faz-se necessário ambientar-se de tal maneira que se possa

criar produtos de qualidade. Esses ambientes são formados de acordo com os modelos de qualidade, e é nesse contexto que será tratado o MPS.BR [4] neste trabalho.

O modelo MPS.BR foi definido utilizando para sua base técnica as normas ISO/IEC 12207:2008 [1] e ISO/IEC 15504-2 [2] e o CMMI-DEV<sup>®</sup> (*Capability Maturity Model Integration for Development*) [3], refletindo assim a necessidade de melhoria dos processos de desenvolvimento de *software*. Dentre esses processos destaca-se a Gerência de Requisitos, processo foco desta pesquisa, pertencente ao nível G na evolução dos níveis de maturidade do MPS.BR.

A Gerência de Requisitos (GRE) é um conjunto de atividades que ajudam a equipe de projeto a identificar, controlar e rastrear requisitos e modificações de requisitos em qualquer época, à medida que o projeto prossegue [5]. Essas características inerentes à GRE são melhor gerenciadas quando se automatiza/sistematiza o processo através da utilização de ferramentas, pois implicam na redução de esforço e tempo, devido a diminuição da necessidade de documentação agilizando o processo como um todo. A prática do uso de ferramentas para sistematizar/automatizar atividades de processo é um dos fatores que impacta diretamente a melhoria de processo de *software* [6].

O objetivo principal deste trabalho é apresentar, a partir de uma metodologia, ferramentas de *software* livre, que, quando utilizadas em conjunto, atendam de forma aderente à implementação do processo Gerência de Requisitos do modelo MPS.BR.

Este artigo está estruturado da seguinte forma: na Seção 2 é abordado o processo Gerência de Requisitos no contexto do MPS.BR, fazendo uma breve descrição do modelo, mostrando o objetivo da gerência de

requisitos, bem como os resultados esperados deste processo; na Seção 3 serão apresentadas as ferramentas de apoio à implementação; a Seção 4 apresenta a análise avaliativa destas ferramentas para atender ao processo de GRE do MPS.BR.; e, finalmente, a Seção 5 apresenta uma visão geral do artigo e trabalhos futuros.

## 2. Processo de Gerência de Requisitos no MPS.BR

O modelo MPS.BR tem como propósito a Melhoria de Processos de Software em empresas brasileiras. Seu diferencial se destaca pela forma de implementação que pode ser realizada entre um conjunto de empresas representando dessa maneira a diminuição do custo, facilitando assim, a adoção do modelo pelas micro, pequenas e médias empresas, foco principal do modelo.

O objetivo principal visa definir e implementar o Modelo de Referência (MR-MPS), que contém os requisitos que os processos das unidades organizacionais devem atender para estar em conformidade com o MR-MPS [4], em empresas brasileiras, difundindo o modelo MPS.BR a todas as regiões do país.

No MR-MPS são definidos sete níveis de maturidade que representam o grau de melhoria de processo para um determinado conjunto de processos: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado), G (Parcialmente Gerenciado), estes níveis evoluem do nível G ao A [4].

Tem-se, então, no nível G o processo de Gerência de Requisitos que tem como objetivo acompanhar a evolução dos requisitos e sua consistência com os outros produtos de trabalho da produção do *software*. O processo de Gerência de Requisitos prevê a rastreabilidade bidirecional entre requisitos e produtos de trabalho, que consiste em analisar impactos de possíveis mudanças.

O processo de Gerência de Requisitos não pretende coletar, desenvolver, detalhar requisitos, apenas acompanhar e administrar as inconsistências e mudanças que um requisito pode gerar, verificando até onde vai o impacto da mudança: planos do projeto, estimativas de tempo e custo, recursos humanos.

No processo de Gerência de Requisitos têm-se cinco resultados esperados. Um resultado esperado é um resultado observável do sucesso do alcance do propósito do processo [1], ou seja, é um dos passos necessários a ser evidenciado para que um determinado processo seja implementado.

Os cinco resultados esperados são:

**GRE 1.** Os requisitos são entendidos, avaliados e aceitos junto aos fornecedores de requisitos, utilizando critérios objetivos;

**GRE 2.** O comprometimento da equipe técnica com os requisitos aprovados é obtido;

**GRE 3.** A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;

**GRE 4.** Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;

**GRE 5.** Mudanças nos requisitos são gerenciadas ao longo do projeto.

O Guia de Implementação [11] fornece um melhor detalhamento a respeito dos resultados esperados. Uma implementação destes resultados esperados, a partir do uso de ferramentas de *software* livre, será discutida com melhor ênfase na seção 4.

## 3. Ferramentas de Apoio

Para apoiar a implementação do processo de Gerência de Requisitos, foi utilizado um conjunto de ferramentas livres, de forma a alcançar uma metodologia sistematizada de implementação de um programa de melhoria da qualidade organizacional para atender aos resultados esperados do processo de GRE do MPS.BR. Durante a pesquisa foi analisada a aderência de várias ferramentas aos resultados esperados do MPS.BR, e destas foram escolhidas quatro ferramentas que melhor atendiam às necessidades da metodologia de implementação proposta: *OSRMT*, *Spider-CL*, *DotProject* e *Mantis*.

### 3.1. OSRMT

O *OSRMT* [7], “*Open Source Requirements Management Tool*” (isto é, ferramenta de código aberto para gerência de requisitos), é uma ferramenta, desenvolvida na linguagem *Java*, projetada para apoiar o processo de gerência de requisitos. Licenciada sob os termos da *GPL (General Public License)*, hoje possui a versão 1.5 como sua versão mais estável (atualmente no “patch 2”) e encontra-se disponível através do site *sourceforge.net*.

As principais características desta ferramenta focam em permitir uma completa rastreabilidade do ciclo de vida de desenvolvimento de *software* em relação aos requisitos.

Entre as funcionalidades da ferramenta, pode-se destacar: registro de autor, origem e motivo da necessidade de cada requisito; registro de casos de uso, *status* e origem de cada requisito (inclusive possibilitando atribuição de categorias aos requisitos; rastreabilidade (através de gráficos que identificam todas as dependências entre requisitos); definição e organização de artefatos e entrada de dados; e geração de relatórios padronizados em formato *PDF*.

### 3.2. Spider-CL

O Spider-CL [8] é uma ferramenta desenvolvida no projeto SPIDER da Universidade Federal do Pará, com propósito de criar *checklists* compostos por critérios objetivos para utilização em diversos contextos, provendo mecanismos para a aplicação destes *checklists*, mantendo histórico e registrando seus resultados.

*Checklists* são bastante utilizados para avaliações e inspeções objetivas de produtos de trabalhos diversos em organizações. Um *checklist* é uma lista de atributos ou qualidades que devem ser avaliados em um determinado produto de trabalho, onde cada um desses atributos possui uma lista de possíveis valores dos quais apenas um pode ser marcado. Um *checklist* nada mais é do que uma relação organizada de critérios objetivos.

O Spider-CL é uma ferramenta web, que pode ser executada através de servidor Tomcat, sendo acessível de qualquer navegador web, e seu banco de dados é estruturado em MySQL. Conta com serviço de controle de acesso através de cadastro de usuários e provê a sistematização do processo de definição e aplicação de *checklists* para avaliação, inspeção ou revisão através de critérios objetivos. A interface do Spider-CL foi desenvolvida utilizando componentes gráficos convencionais como caixas de textos, tabelas, listas e botões, para permitir fácil utilização.

A ferramenta Spider-CL é marcada pelas seguintes características:

- É uma ferramenta gratuita;
- É portátil, sendo desenvolvida como uma aplicação para o servidor Tomcat. A ferramenta pode ser executada em qualquer servidor capaz de executar o Tomcat 6.0 e o MySQL 5.1;
- Possui uma interface de fácil utilização;
- Pode ser utilizada para o desenvolvimento de qualquer tipo de *checklist* objetivo;
- Possui controle de acesso e mantém registro de todas as utilizações de cada *checklist*;

- Exporta os *checklists* preenchidos e seus resultados para o formato PDF.

### 3.3. DotProject

O DotProject [9] é uma ferramenta desenvolvida em PHP com a finalidade de gerenciar projetos através de uma interface web. É *software* livre, com código aberto, distribuído sob a licença GNU-GPL (*GNU General Public License*), ou seja, seus usuários têm todo o poder de copiá-lo, modificá-lo ou distribuí-lo com ou sem modificações.

Sendo uma ferramenta de gerência de projetos, o DotProject é composto por funcionalidades para gerenciamento de tarefas, cronogramas, comunicação e compartilhamento.

Esta ferramenta pode ser encontrada no site <http://www.dotproject.net/> e sua versão mais atual é a 2.1.2, a mesma que será referenciada neste artigo.

### 3.4. Mantis

O Mantis [10] é uma ferramenta de *bugtracking*, sob licença GPL, desenvolvido para auxiliar o controle de modificações, no contexto do processo de gerência de configuração, através do gerenciamento das *issues*. *Issues* são relatos de problemas identificados nos produtos de trabalho, que terão sua evolução acompanhada desde a solicitação da mudança até seu desfecho.

Por ser um *software* executado em *browser*, ele independe de sistema operacional e sua base de dados pode ser estruturada em MySQL, MS SQL e PostgreSQL. A sua versão mais recente e estável é a versão 1.1.8 (versão utilizada na metodologia proposta neste artigo), mas atualmente está sendo desenvolvida a versão 1.2.0.

Entre as principais funcionalidades desta ferramenta são identificados: criação de *issues*; gerenciamento do ciclo de vida das *issues*; registro do histórico das *issues*; e controle de *workflow* da ferramenta. Outros aspectos marcantes são: a possibilidade de customização; interface amigável, proporcionando fácil utilização; e a facilidade de extensão através de *plugins*.

## 4. Análise Avaliativa

A análise avaliativa das ferramentas propostas tem como objetivo tornar o processo Gerência de Requisitos do MPS.BR totalmente implementado, de acordo com as exigências do modelo.

A análise isolada da ferramenta OSRMT não possibilitou essa implementação, devido restrições em suas funcionalidades quanto ao atendimento das recomendações do modelo MPS.BR. Então, surgiu a necessidade de utilizá-la em conjunto com outras ferramentas para contemplar os resultados esperados.

Esta proposta de implementação em conjunto, faz parte de um projeto de pesquisa chamado SPIDER – Uma Proposta de Solução Sistêmica de um SUITE de Ferramentas de Software Livre de apoio à implementação do modelo MPS.BR. O projeto SPIDER (<http://www.ufpa.br/spider>) visa identificar um grupo de ferramentas de *software* livre, que dêem subsídios para o atendimento aos resultados esperados dos processos dos níveis G e F do modelo MPS.BR, gerando produtos de trabalhos (artefatos), evidenciando dessa maneira a implementação do programa de qualidade organizacional.

A metodologia para o desenvolvimento da análise avaliativa se iniciou com o estudo e entendimento do Guia Geral e Guia de Implementação do MPS.BR e posteriormente foi realizado o mapeamento dos resultados esperados de GRE com as funcionalidades das ferramentas. O resultado deste mapeamento pode ser visualizado na Tabela 1, que apresenta quais ferramentas são necessárias para a implementação de cada processo de GRE.

O primeiro resultado esperado, GRE1, visa identificar um Fornecedor de Requisitos que é a pessoa responsável por fornecer e solicitar alterações em relação aos requisitos. Pretende-se evidenciar que o requisito foi Entendido, que não há discordância em relação ao que foi pedido; Avaliado, que o requisito não entra em conflito com os critérios da equipe técnica; e Aceito, de acordo com a expectativa do cliente; essas evidências devem ser apoiadas por critérios objetivos, ou seja, não podem mudar e devem

ser claros.

**Tabela 1 – Mapeamento dos Resultados Esperados com as Ferramentas Propostas**

Resultados Esperados	OSRMT	Spider-CL	DotProject	Mantis
GRE1	✓	✓		
GRE2	✓		✓	
GRE3	✓			
GRE4	✓			✓
GRE5				✓

Então na ferramenta OSRMT o Fornecedor de Requisitos poderá ser identificado e evidenciado na tela de usuário da ferramenta, como mostra a Figura 1.

Para a manutenção dos *status* de Entendido, Avaliado e Aceito, a ferramenta possibilita criar *status* novos para todos os requisitos que serão registrados. Antes de mudar o *status*, o fornecedor de requisitos deverá anexar o *checklist* aplicado na ferramenta Spider-CL, comprovando, dessa maneira, que o requisito foi avaliado segundo critérios objetivos.

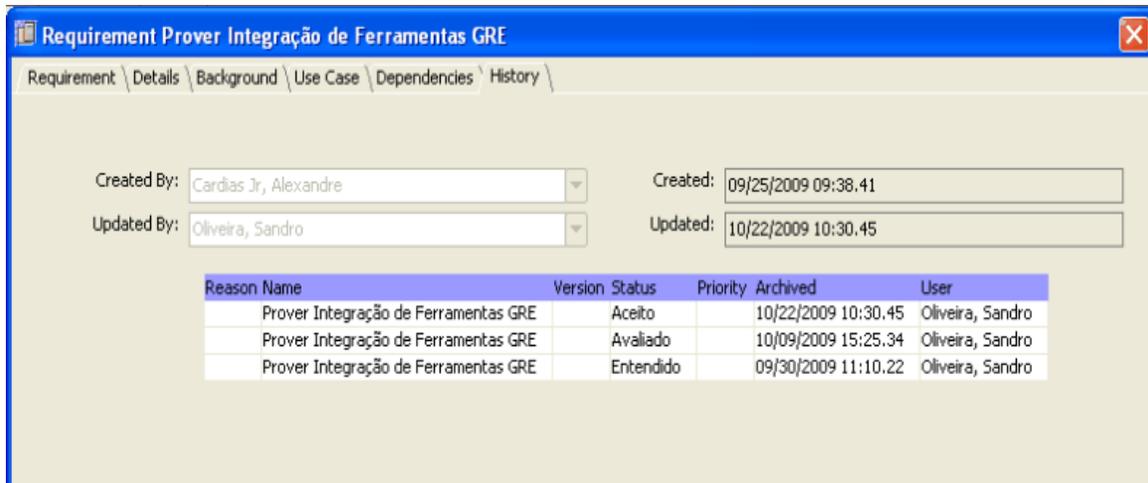
O guia *History* na tela de Requisitos mostra a evolução de *status* que o requisito passou, como visto na Figura 2.

A ferramenta Spider-CL possibilita o registro e manutenção de critérios objetivos que serão adicionados no *checklist*, utilizado neste contexto para a avaliação dos requisitos. Possibilita, ainda, gerar um arquivo no formato “.pdf” para evidenciar a aplicação dos critérios. Uma vez que o *checklist* foi definido, não será possível fazer modificações, atendendo dessa maneira a exigência do modelo MPS.BR em se utilizar critérios objetivos, como visto na Figura 3.

O GRE2 trata sobre o comprometimento da equipe técnica com cada *status* referente ao GRE1. Assim,



**Figura 1 – Identificação do Fornecedor de Requisitos no OSRMT**



**Figura 2 – Histórico do Requisito no OSRMT**

tanto quando o requisito é Entendido, Avaliado e Aceito o comprometimento deve ser obtido, bem como quando uma mudança for solicitada e aprovada, devido o impacto que mudanças podem gerar nas estimativas, custos e cronograma do projeto.

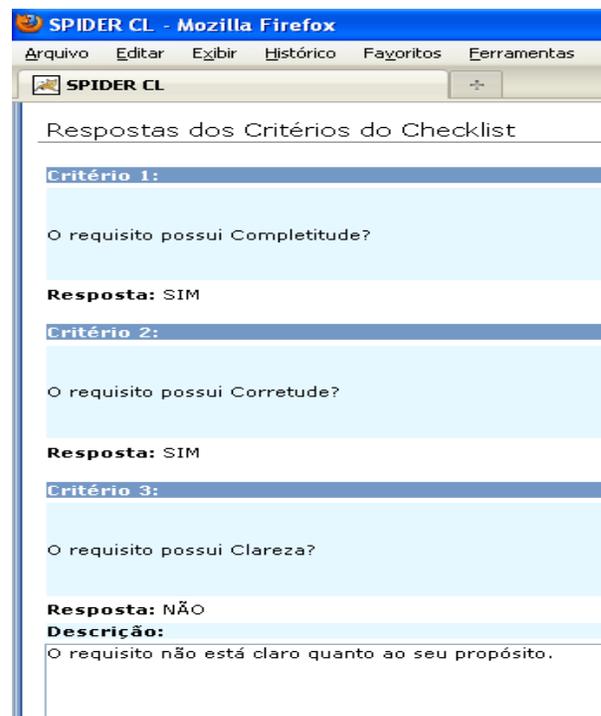
Para atender o resultado esperado GRE2 o Gerente de Requisitos, Analista de Requisitos ou pessoa responsável, deve gerar um relatório na ferramenta OSRMT em que consta o nome, versão, *status*, prioridade e descrição dos requisitos a ser obtido o comprometimento. Este relatório servirá para a equipe técnica saber com quais requisitos estará se comprometendo.

Como a ferramenta OSRMT não possui um mecanismo onde é possível registrar esse comprometimento, foi necessária a utilização da ferramenta DotProject, que possibilita a criação de fóruns. Parte-se do pressuposto que o projeto está registrado no DotProject, sendo necessário anexar o relatório gerado pela ferramenta OSRMT com os requisitos. Sendo assim, um fórum deve ser criado. Sugere-se o nome do fórum como sendo “Requisitos”. Dentro desse fórum vários tópicos devem ser instanciados, para comprometimento de cada *status* (Entendido, Avaliado, Aceito), referenciando na mensagem do tópico os identificadores dos requisitos funcionais registrados na ferramenta OSRMT, a ser obtido o comprometimento, como segue RF01, RF02, RF03. A sugestão para o nome do tópico tem o objetivo de facilitar a identificação dos arquivos e do *status* com o qual a equipe deve se comprometer. O nome do tópico segue o seguinte formato:

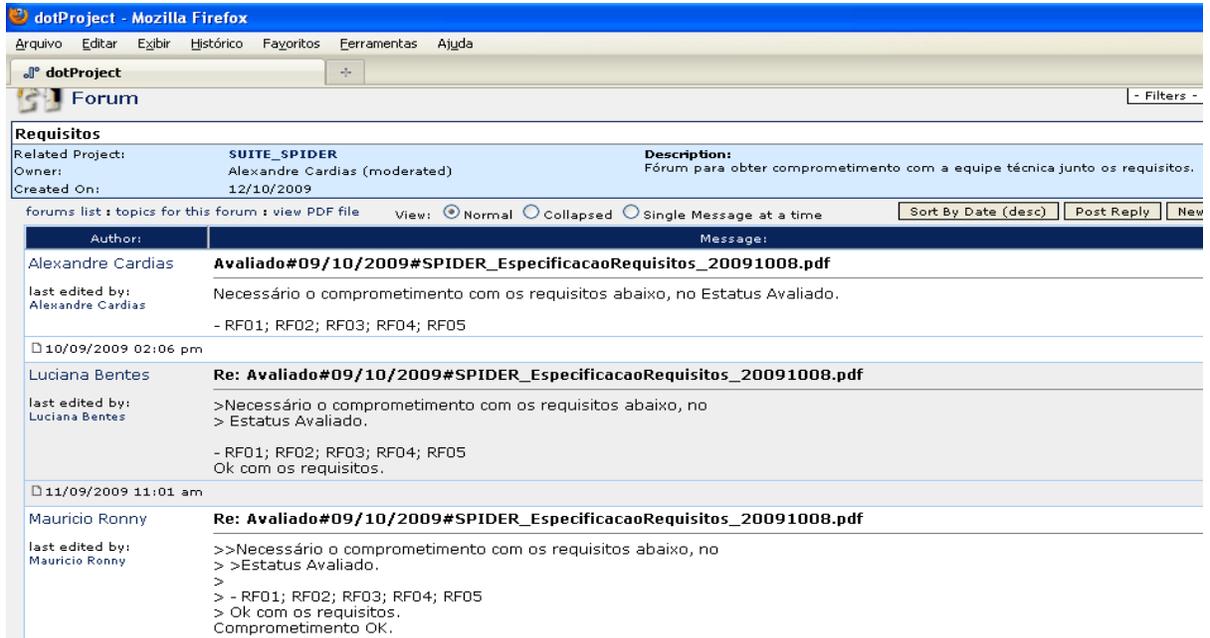
[STATUS]#[DATA\_CRIAÇÃO\_TÓPICO]#[RELATORIO\_DE\_ESPECIFICAÇÃO\_DE\_REQUISITOS]

Um exemplo para obtenção desse comprometimento e do formato do tópico do fórum, pode ser visualizado na Figura 4.

É no terceiro resultado esperado, GRE3, onde a rastreabilidade bidirecional vai ser estabelecida, que é um dos principais objetivos do processo de Gerência de impactos, servindo como base para a análise de impactos. Outro ponto fundamental nesse resultado esperado é a necessidade de manter a rastreabilidade para que seja sempre possível ter uma visão real da análise de impacto.



**Figura 3 – Checklist Aplicado no Spider-CL**



**Figura 4 – Comprometimento com a Equipe Técnica no DotProject**

Este resultado esperado é alcançado através de algum mecanismo de rastreabilidade. A ferramenta OSRMT contempla de forma satisfatória este resultado através da Matriz de Rastreabilidade (horizontal ou vertical), como visto na Figura 5 e da Análise de impacto, importante mecanismo de visualização da(s) dependência(s) entre os artefatos, como visualizado na Figura 6.

No GRE4 faz-se necessária a realização de revisões em planos e produtos de trabalho do projeto, visando identificar e corrigir inconsistências em relação aos requisitos. Se inconsistências forem identificadas devem ser registradas [11].

Na ferramenta OSRMT as revisões serão registradas em *Feature*, funcionalidade que permite registrar características do projeto. Nessa funcionalidade deverá ser anexado o relatório gerado na ferramenta OSRMT contendo os requisitos. Para todas as revisões realizadas em relação a este relatório algumas informações referentes à data de realização da revisão, descrição e ocorrências (inconsistências), se ocorrer, deverão ser detalhadas, como mostra a Figura 7.

Após a definição da revisão no OSRMT, as informações levantadas devem ser passadas para a

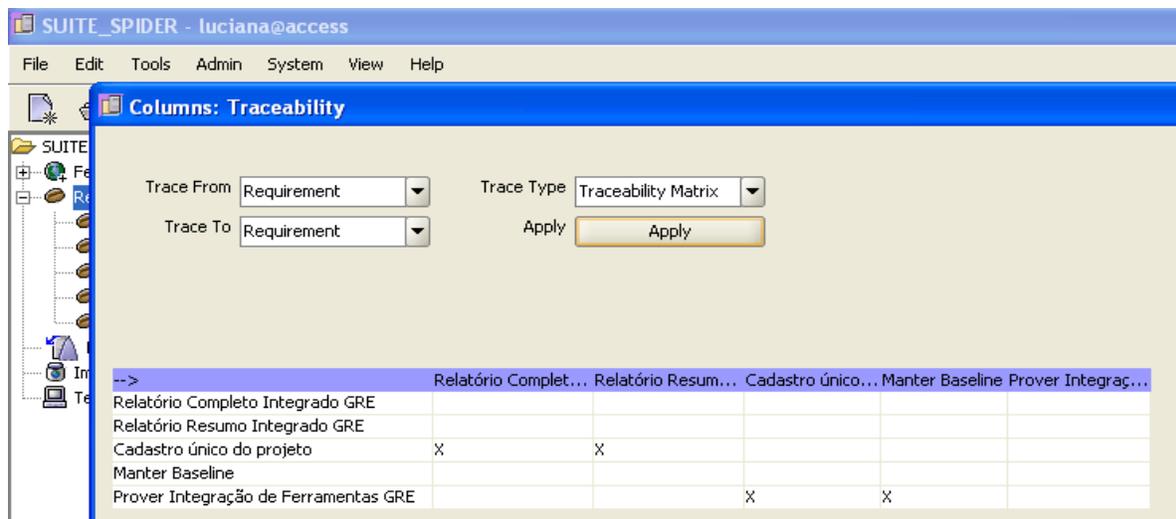
ferramenta de controle de mudança, Mantis, a fim de realizar o devido tratamento e identificação dos problemas/melhorias.

Cada revisão constituirá de um conjunto de problemas/melhorias a ser tratado, e sendo assim cada revisão é registrada na ferramenta Mantis como uma *issue* (problema/melhoria), em que o campo “*category*” indicará que pertence ao processo de Gerência de Requisitos. Esta categoria deverá ser adicionada à ferramenta. A revisão registrada será nomeada com um identificador no formato:

[IDENTIFICADOR\_DA\_REVISÃO]#[NOME\_DO\_DOCUMENTO\_DE\_REFERÊNCIA]

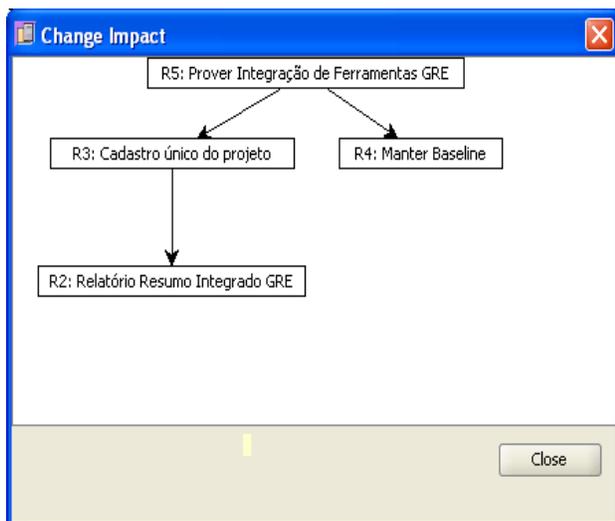
sendo um identificador único e de fácil reconhecimento.

No campo “*summary*”, onde é feita a descrição da *issue*, será informado o conteúdo da revisão, de forma idêntica a feita no OSRMT, contendo: número da revisão, data, descrição e as ocorrências, como mostra a Figura 8.



**Figura 5 – Matriz de Rastreabilidade no OSRMT**

Para cada ocorrência listada na revisão, deve ser criado um *issue* homônimo ao problema/melhoria, contendo descrição específica para aquela ocorrência, bem como a estratégia de resolução do problema. Os *issues* gerados a partir das ocorrências são conectados ao *issue* referente à sua revisão, através da funcionalidade de *Relationships* (relacionamento entre *issues*) do Mantis, de forma que o *issue* da revisão será “pai” dos *issues* de ocorrências, como pode ser visto na Figura 9.



**Figura 6 – Análise de Impacto no OSRMT**

Esta metodologia de identificação dos problemas/melhorias encontrados nas revisões, permite satisfazer totalmente a implementação do GRE 4, e

permite monitorar o avanço da resolução dos *issues*, rastreando-as a partir de suas respectivas revisões. O acompanhamento dos *issues* pertence ao escopo do GRE 5.

O quinto resultado esperado, GRE5, tem como objetivo gerenciar as mudanças ocorridas ao longo do projeto. É importante para o gerenciamento, que um histórico seja mantido em relação aos requisitos, evidenciando dessa maneira a evolução das mudanças.

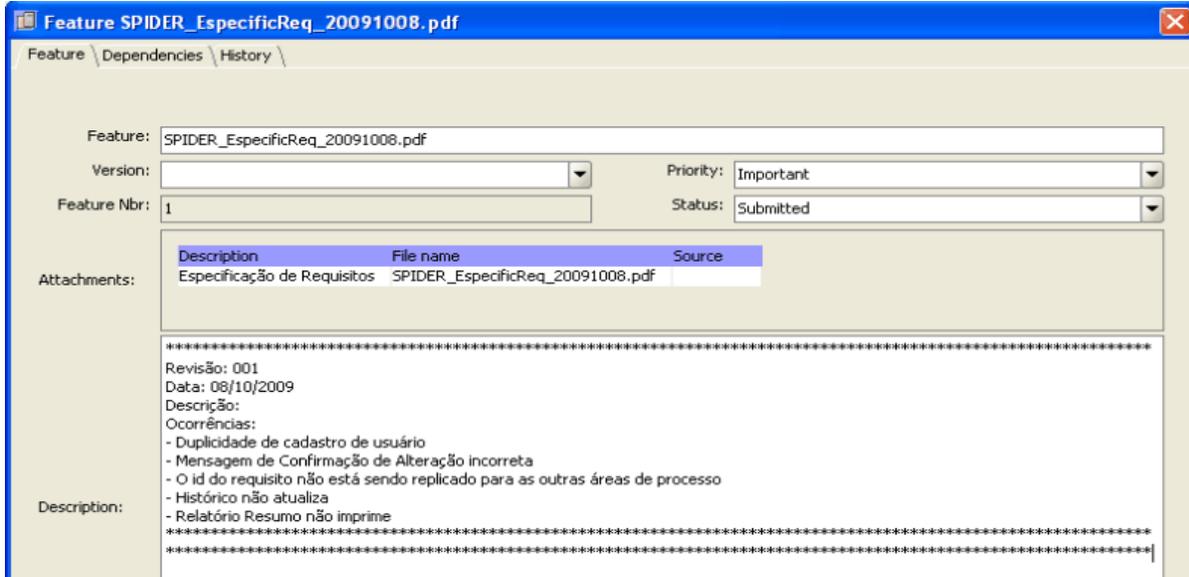
O resultado esperado GRE 5, que recomenda que mudanças nos requisitos são gerenciadas ao longo do projeto, requer identificar (o que já foi feito no GRE4) e acompanhar as mudanças nos requisitos, o que envolve controlar a evolução da resolução de ocorrências identificadas no resultado esperado anterior.

Os *issues* gerados na ferramenta Mantis, contam com um ciclo de vida próprio, baseado em uma série de estados em que um *issue* pode se enquadrar até ser fechada. Os estados são alterados conforme ações são tomadas para resolver o problema/melhoria. Os estados padrões de um *issue* no Mantis são: *new* (novo), *assigned* (apontado para algum responsável), *resolved* (o *issue* foi provisoriamente resolvido) e *closed* (definitivamente resolvido).

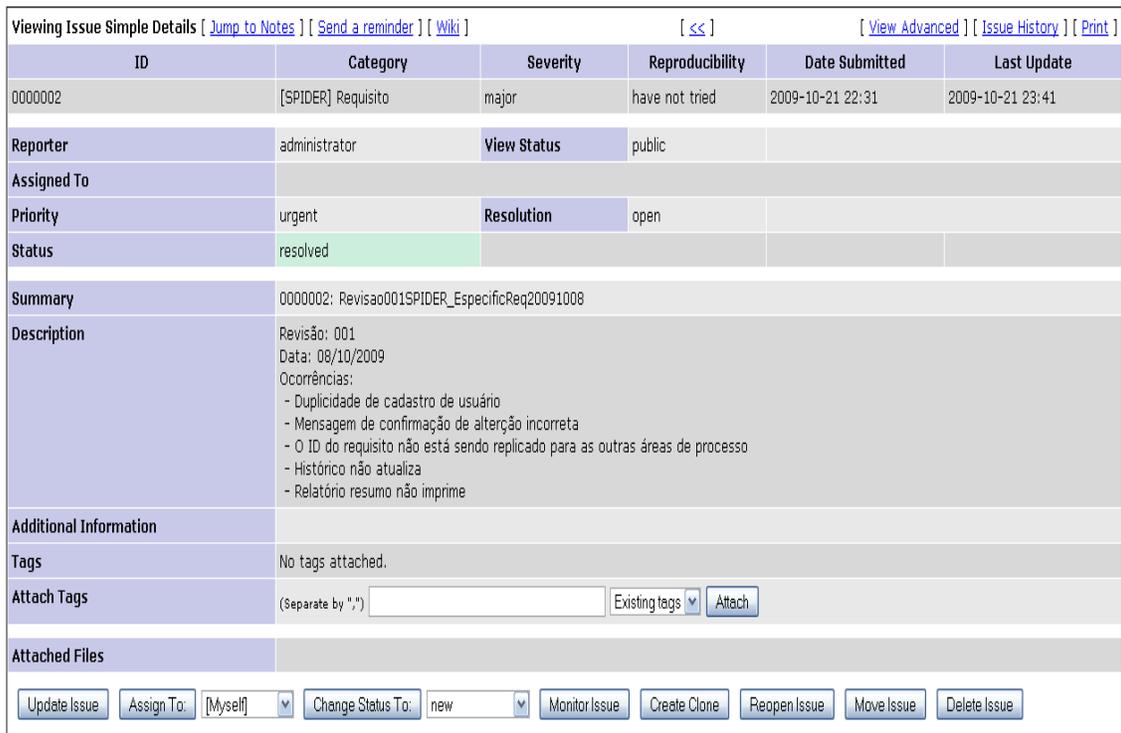
Com base nos estados dos *issues*, pode-se definir um histórico de alterações que a ferramenta Mantis disponibiliza para cada *issue*. Além de apresentar estados no histórico, a ferramenta também exibe os detalhes sobre responsáveis (quem criou, a quem foi apontado), exibe notas que podem ser adicionadas explicando ações tomadas na resolução do problema/melhoria, datas de cada modificação no *issue*.

A metodologia aqui definida utiliza os históricos do Mantis como evidência do acompanhamento das ocorrências de problemas/melhorias identificadas nas revisões, como visualizado na Figura 10.

Para facilitar o acompanhamento das *issues*, recomenda-se o uso do campo “*add note*” para comunicar os procedimentos feitos para solucionar o problema, como visualizado na Figura 10.



**Figura 7 – Revisões Registradas no OSRMT**



**Figura 8 – Identificação da Revisão no Mantis**

Relationships				
New relationship		Current issue	parent of	Add
parent of	0000003	resolved		Duplicidade de cadastro de usuário [Delete]
parent of	0000004	resolved	administrator	Mensagem de confirmação de alteração incorreta [Delete]
parent of	0000005	resolved	administrator	O ID de requisito não está sendo replicado para as outras áreas de processo [Delete]
parent of	0000006	resolved		Histórico não atualiza [Delete]
parent of	0000007	resolved		Relatório resumo não imprime [Delete]

**Figura 9 – Relacionamento entre *issues* de Ocorrências e *issue* de Revisão no Mantis**

Issue History				
Date Modified	Username	Field	Change	
2009-10-21 22:36	administrator	New Issue		
2009-10-21 22:43	administrator	Relationship added	child of 0000002	
2009-10-21 23:38	administrator	Note Added: 0000006		
2009-10-21 23:38	administrator	Assigned To	=> administrator	
2009-10-21 23:38	administrator	Status	new => confirmed	
2009-10-21 23:39	administrator	Note Added: 0000007		
2009-10-21 23:39	administrator	Status	confirmed => resolved	
2009-10-21 23:39	administrator	Resolution	open => fixed	

**Figura 10 – Histórico de Mudanças dos *issues* no Mantis**

Uma empresa que desejar adotar essa metodologia poderá formalizá-la através de treinamentos e/ou elaboração de um projeto piloto para melhor entendimento.

## 5. Conclusões

A gerência de requisitos se constitui num dos mais importantes processos no desenvolvimento de *software* devido à influência que exerce nas demais áreas de processo. Requisitos mal gerenciados podem, por exemplo, resultar em não cumprimento do cronograma do projeto, falhas nas estimativas, gerando impacto direto no produto final e consequentemente a insatisfação do cliente.

Este estudo se traduz numa importante alternativa para organizações interessadas em implementar o processo de GRE visando alcançar o nível de maturidade G através da avaliação do MPS.BR, principalmente as pequenas e médias empresas que não possuem recursos para a aquisição de ferramentas proprietárias.

Importante mencionar que a metodologia discutida neste trabalho não propõe a definição e institucionalização de um processo organizacional de Gerência de Requisitos e nem substitui este processo, entende-se que a metodologia agrega facilidades na execução do processo a partir do uso de ativos organizacionais na forma de ferramentas de *software* livre.

Vale ressaltar que devido a dificuldade de encontrar ferramentas livres para a gerência de requisitos que

contemplem de modo satisfatório os resultados esperados do modelo MPS.BR, é importante destacar que a análise das ferramentas apresentadas neste artigo é apenas uma proposta para implementar o processo de GRE, podendo cada organização adaptá-la conforme sua realidade.

Importante enfatizar, ainda, que, neste trabalho, o conjunto de ferramentas de software livre foi analisado de maneira isolada, ou seja, a gestão das informações mantidas no repositório de cada uma das ferramentas discutidas é feita de maneira individual e não integrada, o que pode, avaliando o cenário pós-institucionalização da metodologia apresentada, inviabilizar a sua aplicação em projetos que envolvam a alocação de uma quantidade grande de recursos humanos. A solução deste problema recai no atual esforço do projeto SPIDER em desenvolver um suíte de ferramentas integradas de software livre apoiando a execução do processo de software definido a partir de uma ferramenta de modelagem de processos. Esta ferramenta também compõe um dos subprojetos do SPIDER que adota o padrão SPEM – *Software Process Engineering Metamodel*.

Como trabalhos futuros destacam-se: a integração das ferramentas propostas dentro do escopo do Projeto SPIDER de desenvolvimento de um SUITE, para centralizar as operações e serviços das ferramentas, dinamizando ainda mais o processo; o desenvolvimento de um estudo de caso para coletar informações a respeito dos benefícios e/ou dificuldades relatadas objetivando o aprimoramento da metodologia proposta; avaliação de outras ferramentas de *software* livre para

apoio ao processo de Gerência de Requisitos; e a propagação desta metodologia em organizações, através do desenvolvimento de um Manual que oriente todos os passos necessários para o alcance dos objetivos dos resultados esperados do processo de Gerência de Requisitos.

## 6. Referências Bibliográficas

[1] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION/ INTERNATIONAL ELECTROTECHNICAL COMMISSION. *ISO/IEC 12207 Systems and software engineering – Software life cycle processes*, Geneve: ISO, 2008

[2] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION/ INTERNATIONAL ELECTROTECHNICAL COMMISSION. *ISO/IEC 15504-2: Information Technology – Process Assessment – Part 2 – Performing an Assessment*, Geneve: ISO, 2003

[3] SOFTWARE ENGINEERING INSTITUTE. *CMMI for Development (CMMI-DEV)*, Version 1.2, Technical Report CMU/SEI-2006-TR-008. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006

[4] SOFTEX, Sociedade para Promoção da Excelência do Software Brasileiro, MPS.BR – Melhoria de Processo de Software Brasileiro. *Guia Geral*, maio 2009

[5] PRESSMAN, Roger S., *Engenharia de Software*, 6° ed. São Paulo: MCGRAW-Hill, 2006

[6] SOFTEX – Sociedade para Promoção da Excelência do Software Brasileiro, *MPS.BR: Lições Aprendidas*, organizadores: Ana Regina Cavalcanti da Rocha e Kival Chaves Weber, Campinas-SP, 2008.

[7] SOURCEFORGE, *Open Source Requirements Management Tool*. Disponível em: <http://sourceforge.net/projects/osrmt/>, acesso em 10/10/2009.

[8] BARROS, Renan S., *Manual do Usuário – SPIDER-CL – Versão 1.2*, Julho, 2009.

[9] DOTPROJECT, *DotProject – Project Management Software*. Disponível em: <http://www.dotproject.net/>, acesso em 10/10/2009.

[10] MANTIS, *Mantis Bug Tracker*. Disponível em: <http://www.mantisbt.org/>, acesso em 10/10/2009.

[11] SOFTEX, Sociedade para Promoção da Excelência do Software Brasileiro, *Guia de Implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS.BR*, maio de 2009.