

# Uma Implementação do Processo de Garantia da Qualidade usando a Spider-QA, a Spider-CL e o Mantis

Rodrigo Araujo Barbalho<sup>1</sup>, Marília Paulo Teles<sup>2</sup>,  
Sandro Ronaldo Bezerra Oliveira<sup>1,2</sup>

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

<sup>2</sup>Programa de Pós-Graduação em Ciência da Computação – Instituto de Ciências Exatas e Naturais – Universidade Federal do Pará (UFPA) - Belém – PA - Brasil

{rodbarbalho, mariliapaulo}@gmail.com, srbo@ufpa.br

**Abstract.** *This paper presents a systematic implementation of the Quality Assurance process using free software tools, such as Spider-QA, Spider-CL and Mantis, achieving the adherence to the expected results of MPS.BR model and the specific practices of CMMI model. This implementation is described from a methodology to support organizational process improvement programs.*

**Resumo.** *Este artigo apresenta uma implementação sistematizada do processo de Garantia da Qualidade com uso das ferramentas de software livre Spider-QA, Spider-CL e Mantis para atendimento dos resultados esperados do modelo MPS.BR e das práticas específicas do modelo CMMI. Esta implementação é descrita na forma de uma metodologia de apoio aos programas de melhoria do processo organizacional.*

## 1. Introdução

Modelos de qualidade de *software* tendem a melhorar os processos organizacionais e mensurar a capacidade da organização em gerenciar o desenvolvimento, aquisição e manutenção dos produtos e serviços de *software* (SOFTEX, 2011a). Neste contexto, surgiu o programa de Melhoria do Processo de Software Brasileiro (MPS.BR), que trata de um programa desenvolvido pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) voltado para recomendar boas práticas que norteiam o desenvolvimento de software com qualidade. Este programa divulga um modelo de referência, o MR-MPS (SOFTEX, 2011a), que é baseado, entre outros, no programa CMMI (SEI, 2010) – *Capability Maturity Model Integration*, desenvolvido pelo *Software Engineering Institute* (SEI) e que possui um modelo de referência para software, o CMMI-Dev – *CMMI for Development*.

Dentre os processos adotados pelo MR-MPS e pelo CMMI-Dev destacam-se o processo de Garantia da Qualidade (GQA) e de *Process and Product Quality Assurance* (PPQA), respectivamente. O propósito destes processos é assegurar que os produtos de trabalho e a execução dos processos estejam em conformidade com os planos e recursos predefinidos (SOFTEX, 2011b; SEI, 2010).

Este artigo tem como objetivo apresentar uma proposta de sistematização do processo de GQA do MR-MPS e de PPQA do CMMI-Dev, utilizando-se de ferramentas de software livre. Na seção 2, trabalhos relacionados à implementação da garantia da

qualidade através de apoio ferramental são apresentadas. Na seção 3 é apresentada a metodologia de implementação com apoio das ferramentas Spider-QA, Spider-CL e Mantis e na seção 4 é feita uma análise da aderência da metodologia proposta aos modelos de qualidade MR-MPS e CMMI-Dev. Por fim, a seção 5 apresenta as conclusões deste trabalho.

## 2. Trabalhos Relacionados

Este artigo aborda uma estratégia para a implementação do processo de Garantia da Qualidade do MR-MPS e CMMI-Dev baseada em boas práticas de uso de ferramentas de *software* livre, adaptadas e desenvolvidas no contexto do projeto de pesquisa SPIDER<sup>1</sup> – *Software Process Improvement Development and Research* (Oliveira *et al.*, 2011). Outras ferramentas de *software* livre podem ser utilizadas para implementação do processo de GQA, como por exemplo, a *Qualcon*<sup>2</sup>, desenvolvida em 2004 sendo capaz de implementar com sucesso a ISO 9001:2000, no entanto, esta norma não se refere especificamente ao desenvolvimento de *software*, assim não abrange atividades específicas a este fim. A *QA Assistant Tool*<sup>3</sup> foi utilizada durante o projeto de desenvolvimento do sistema Fedora, sendo capaz de registrar *checklists* e aplicá-los, porém limita-se ao atendimento deste projeto. Outra ferramenta livre desenvolvida para auxiliar no processo de garantia da qualidade é a *QA Manager*<sup>4</sup>, que propõem funcionalidades como controlar detalhes de engenharia, revisões de qualidade, alocação de recursos, entre outras, porém não se preocupa com a gestão das não-conformidades encontradas em uma revisão.

Rocha (2009) apresenta uma metodologia para implementação de melhoria de processo de *software*, intitulada MIMPS, utilizando o Ambiente de Desenvolvimento de *Software* (ADS) WebAPSEE. O ponto fraco desta abordagem é que como a metodologia trabalha com um ambiente centrado no processo acaba possibilitando mudança de cultura organizacional a partir da adoção de um novo ambiente de trabalho para a equipe de desenvolvimento de um projeto. Além disso, a metodologia não é específica para implementação do processo de GQA.

Nenhuma das ferramentas apresentadas é capaz de realizar uma comunicação com ferramentas de *bug tracking* que auxiliam na resolução de problemas e/ou não-conformidades encontrados (Teles, 2010). Vale ressaltar que essas ferramentas foram analisadas devido ao fato de serem ferramentas de *software* livre, o que reduziria consideravelmente os custos de uma implementação, foco do projeto SPIDER.

## 3. Metodologia de Implementação de GQA com Apoio de Ferramentas Livres

Nesta seção será apresentada a metodologia para implementação do processo de GQA do MR-MPS e da área de processo PPQA do CMMI-Dev. A metodologia nada mais é que apresentar uma forma de utilização conjunta das ferramentas: (1) Spider-QA, para geração de planos de qualidade, realização de auditorias, registro e acompanhamento de

---

<sup>1</sup> <http://www.spider.ufpa.br>

<sup>2</sup> <http://sourceforge.net/projects/qualcon>

<sup>3</sup> <http://fedoraproject.org>

<sup>4</sup> <http://freshmeat.net/projects/qamanager>

<sup>5</sup> <http://www.mantisbt.org/>

não-conformidades e problemas encontrados; (2) Spider-CL, para definição e aplicação de critérios objetivos, a partir de *checklists*; e (3) Mantis<sup>5</sup>, para registro das não-conformidades e problemas encontrados e acompanhamento de suas resoluções. As ferramentas Spider-QA e Spider-CL, bem como manuais de utilização das ferramentas, estão disponíveis no *site* do projeto SPIDER. Ressalta-se que a escolha das ferramentas se deu com base nos resultados obtidos e relatados em (Oliveira *et al.*, 2011).

Neste trabalho entende-se metodologia como sendo a codificação de um conjunto de práticas recomendadas (serviços de apoio/funcionalidades oferecidos pelas ferramentas), às vezes acompanhada de material de treinamento, programas de educação formal, planilhas, diagramas, tomando como parte um método (processo com uma série de passos, para construir um software) (Pressman, 2006).

### 3.1. Planejando a Garantia da Qualidade

Considerando um ambiente onde estejam devidamente instaladas e configuradas ferramentas de apoio ao processo de GQA, a metodologia inicia com a definição de um Plano da Qualidade. Neste plano deverão conter: (1) quais produtos e processos serão avaliados; e (2) o agendamento de auditorias para realização das avaliações; estas atividades são registradas na Spider-QA. Tanto o MR-MPS quanto o CMMI-Dev recomendam que critérios objetivos podem ser definidos e aplicados através de *checklists*, questionários, ferramentas automatizadas de verificação, entre outros, para a realização de auditorias. Nesta metodologia utilizam-se *checklists* e entrevistas para a realização de avaliações.

Os critérios objetivos que são utilizados para seleção de quais produtos de trabalho ou processos devem ser incluídos para avaliação no Plano da Qualidade. Estes critérios são registrados na ferramenta Spider-CL e em seguida inseridos em um *checklist* que pode ser aplicado através da própria ferramenta ou impresso para uma aplicação manual. Esta seleção deve-se ao fato de uma organização poder dispor de um conjunto grande de produtos de trabalho e processos para o desenvolvimento dos seus produtos de software ou serviços, assim a avaliação deve ser realizada por amostragem. A Figura 1 apresenta a aplicação de um *checklist* cadastrado a partir da ferramenta Spider-CL.

Uma vez definidos os produtos e os processos que serão submetidos à avaliação, gera-se um Plano da Qualidade através da Spider-QA, incluindo-se o agendamento de auditorias programadas para avaliação dos produtos, a partir dos artefatos definidos no processo, e dos processos previamente selecionados, a partir dos produtos e da alocação dos membros alocados nos projetos de software em avaliação. Durante o registro de uma auditoria, devem-se definir: a data, o local e o horário previstos para realização da mesma; o tipo de auditoria (produto ou processo); a seleção de qual produto ou processo será avaliado; e o *checklist* que será utilizado para avaliação. O *checklist* que será utilizado para avaliação durante a auditoria é definido na ferramenta Spider-QA.

Membros alocados nos projetos de software que irão participar das auditorias, a partir de entrevistas, devem ser registrados como participantes na ferramenta Spider-QA e, posteriormente, devem ser comunicados sobre a realização das auditorias. A ferramenta Spider-QA possibilita o envio de e-mails contendo o nome do projeto a qual a auditoria está relacionada, a data, hora e o local da realização da mesma.

**Figura 1 – Aplicação de um *Checklist* através da Ferramenta Spider-CL**

### 3.2. Executando a Garantia da Qualidade

Uma vez definido o Plano da Qualidade, serão executadas as auditorias agendadas para avaliar a aderência dos produtos de trabalho e dos processos aos padrões de qualidade previamente estabelecidos. A execução das auditorias é realizada através da ferramenta Spider-QA, conforme definido na etapa Planejando a Garantia da Qualidade. Durante a execução informa-se a data de início da auditoria, a situação da mesma (sendo os possíveis estados da auditoria: Novo, Em Andamento ou Finalizada), e a avaliação do que está sendo auditado.

### 3.3. Registrando Não-Conformidades e Problemas

Durante a realização da auditoria não-conformidades e problemas podem ser detectados e, posteriormente, registradas. A ferramenta Spider-QA possibilita o registro das não-conformidades e dos problemas encontrados, conforme os critérios objetivos definidos previamente na criação dos *checklists* de auditoria. Cada item do *checklist* deve ser avaliado usando os seguintes graus: Contempla, Contempla Parcialmente, Não Contempla ou Não se Aplica. A ferramenta permite, ainda, o preenchimento de um campo de observação que pode ser utilizado para descrever a não-conformidade e/ou problema encontrado, ou fazer uma sugestão de melhoria/solução. A Figura 2 apresenta a tela de registro de não-conformidades encontradas na ferramenta Spider-QA.

Após a finalização da auditoria as pessoas interessadas devem ser comunicadas sobre as não-conformidades e os problemas encontrados. A ferramenta Spider-QA permite que seja enviado um *e-mail* para as pessoas interessadas contendo um relatório informando os problemas encontrados.

**spiderQA**  
Tool for Quality

Bem

Manutenção   Auditoria   Relatórios   Segurança   Sair

**Executar Auditoria**

Item auditado: Lista de Requisitos  
Data Prevista: 20/05/2011  
Data Inicio:   
Situação: Novo

Itens do Checklist	Situação	Observação
Compleitude	Contempla	
Clareza	Contempla	

Copyright © 2010 - Projeto SPIDER

**Figura 2 – Formulário para Registro de Não-conformidades encontradas através da Ferramenta Spider-QA.**

### 3.4. Definindo o Plano de Ação

Após a finalização das auditorias, deverá ser definido um Plano de Ação para a resolução das não-conformidades e dos problemas encontrados, onde deverão ser informados: a pessoa responsável pela resolução; e um prazo máximo para resolução.

A ferramenta Spider-QA permite que sejam criados Planos de Ação, no entanto, é necessário registrar um projeto na ferramenta de *bug tracking*, Mantis, e cadastrar casos que servirão para indicar não-conformidades e problemas a serem resolvidos. Cada caso pode possuir um estado que indica a situação da resolução do mesmo em um determinado momento, por exemplo, quando um novo caso é registrado ele inicia com status de “Novo” e quando este caso é encaminhado para resolução para a pessoa responsável, passa, então, para o estado de “Atribuído”. O uso do Mantis faz-se necessário já que este possui uma máquina de estados sistematizada para tratamento e relato dos casos provenientes das não-conformidades e dos problemas.

Com o projeto e os casos devidamente registrados no Mantis, o passo seguinte é, a partir da Spider-QA, definir quais itens entrarão no Plano de Ação. A Spider-QA realiza uma integração com a ferramenta Mantis onde os itens selecionados para resolução são incluídos nos casos criados a partir do seu identificador.

### 3.5. Acompanhando as Resoluções

Com a finalização da definição do Plano de Ação, o responsável pela garantia da qualidade deve fazer o acompanhamento das resoluções a partir da Spider-QA. A ferramenta realiza uma comunicação com o Mantis para obter informações de uma determinada não-conformidade e problema cadastrados previamente, apresentando o item, o responsável pela resolução, o prazo para resolução, o código do problema cadastrado no Mantis e um resumo do problema.

Caso uma não-conformidade ou um problema registrado não seja resolvido dentro do prazo definido no Plano de Ação, o responsável pela garantia da qualidade

deve, através do Mantis, realizar o escalonamento deste problema ou não-conformidade, ou seja, atribuí-lo a outra pessoa em um nível mais elevado na hierarquia da empresa, para tentar resolver o problema. A Figura 3 apresenta uma tela da ferramenta Mantis contendo os casos criados e a situação dos mesmos.

The screenshot shows the Mantis Bug Tracker interface. At the top, there is a navigation bar with links like 'Principal', 'Minha Visão', 'Ver Casos', etc. Below this is a search bar and a 'Filtros' section with various criteria like 'Relator:', 'Monitorado Por:', 'Atribuído a:', etc. The main area displays a table of cases with columns for 'P', 'Núm', '#', 'Categoria', 'Gravidade', 'Estado', 'Atualizado', and 'Resumo'. Three cases are listed, all categorized as 'Não-conformidades' with dates from 2011-06-21. At the bottom, there is a footer with copyright information and a logo for 'MANTIS BUG TRACKING SYSTEM'.

**Figura 3 – Resumo dos Casos Existentes Cadastrados na Ferramenta Mantis.**

#### 4. Aderência da Metodologia Proposta ao MR-MPS e CMMI-Dev

A análise de aderência entre a metodologia proposta ao modelo MR-MPS e ao modelo CMMI-Dev é realizada através do mapeamento das práticas descritas na seção 3, com os resultados esperados do processo de Garantia da Qualidade constantes no Guia Geral do MPS.BR (SOFTEX, 2011a) e as práticas específicas da área de processo de Garantia da Qualidade do Produto e Processo constante no CMMI-Dev (SEI, 2010), conforme visualizado no Quadro 1 (Teles, 2011). É importante ressaltar que a metodologia restringe-se a apresentar boas práticas para que seja possível contemplar os resultados esperados do MR-MPS e as práticas específicas do CMMI-Dev em conjunto com as práticas definidas no processo de software da organização. Resultado Esperado é “um resultado observável do sucesso do alcance do propósito do processo” (ISO/IEC, 2008) e Prática Específica “é a descrição de uma atividade considerada importante para alcançar uma meta específica associada” (SEI, 2010).

**Quadro 1. Mapeamento da Metodologia com os Resultados Esperados do MR-MPS e Práticas Específicas do CMMI-Dev (Teles, 2011)**

MR-MPS	CMMI-Dev	Ferramenta(s) de Apoio	Funcionalidade(s) das Ferramentas
GQA1	SP 1.2	Spider-QA Spider-CL	<ul style="list-style-type: none"> <li>Definir o Plano da Qualidade</li> <li>Agendar Auditorias</li> <li>Aprovar o Plano da Qualidade</li> <li>Preparar Auditorias</li> <li>Selecionar Material para Auditoria</li> <li>Executar Auditorias</li> </ul>

MR-MPS	CMMI-Dev	Ferramenta(s) de Apoio	Funcionalidade(s) das Ferramentas
GQA2	SP 1.1	Spider-QA Spider-CL	<ul style="list-style-type: none"> <li>Definir o Plano da Qualidade</li> <li>Agendar Auditorias</li> <li>Aprovar o Plano da Qualidade</li> <li>Preparar Auditorias</li> <li>Selecionar Material para Auditoria</li> <li>Selecionar Participantes para Entrevistas</li> <li>Executar Auditorias</li> </ul>
GQA3	SP 2.1 SP 2.2	Spider-QA Mantis	<ul style="list-style-type: none"> <li>Executar Auditorias</li> <li>Notificar Não-Conformidades ou Problemas</li> </ul>
GQA4	SP 2.1	Spider-QA Mantis	<ul style="list-style-type: none"> <li>Estabelecer Datas para Correção</li> <li>Corrigir Não-Conformidades ou Problemas</li> <li>Verificar Correções</li> <li>Reportar Não-conformidades ou Problemas encontrados à níveis mais altos na hierarquia da organização</li> <li>Finalizar Não-Conformidades ou Problemas</li> <li>Notificar Resultados</li> </ul>

## 5. Conclusões

Como observado em lições aprendidas de implementações realizadas no programa de melhoria de processo, o emprego de ferramentas sistematizadas pode reduzir consideravelmente o tempo de implementações do MR-MPS e CMMI-Dev, e, tratando-se de ferramentas livres, há também uma redução de custos financeiros para a organização. Como sugestão para trabalhos futuros observa-se a possibilidade de implementação da metodologia proposta em empresas parceiras do projeto SPIDER, com o intuito de avaliar a viabilidade de implementação das práticas sugeridas na metodologia.

Com o objetivo de agilizar o processo de implementação do programa MPS.BR e CMMI, este trabalho apresentou uma solução para implementação do processo de Garantia da Qualidade utilizando ferramentas livres. Apenas o uso da metodologia não é suficiente para atender ao MR-MPS e o CMMI-Dev, porém, define boas práticas para uso do ferramental de forma aderente aos resultados esperados e práticas específicas constantes nestes modelos. Vale ressaltar que as boas práticas apresentadas neste trabalho encontram-se institucionalizadas no Projeto SPIDER, que objetiva a concepção, elaboração e construção de ferramentas de software livre para apoiar a implementação dos processos constantes no MR-MPS e CMMI-Dev.

Como trabalho futuro pode-se relatar a implementação da metodologia em uma empresa de desenvolvimento de software parceira do projeto SPIDER com a finalidade de avaliar a viabilidade da implantação das boas práticas sugeridas nesta metodologia, a

fim de um compor um estudo de caso real a partir da extração de métricas. Vale ressaltar que uma avaliação experimental controlada foi realizada com o uso da metodologia a partir da interação com profissionais de diferentes perfis técnicos de desenvolvimento de software, os resultados encontram-se publicados em (Teles, 2011).

## **Referências Bibliográficas**

- Oliveira, S. R. B. *et al.* (2011) “SPIDER – Uma Proposta de Solução Sistêmica de um SUITE de Ferramentas de Software Livre de Apoio à Implementação do Modelo MPS.BR”. Revista do Programa Brasileiro da Qualidade e Produtividade em Software. PBQP Software. SEPIN-MCT.
- Pressman, R. S. “Engenharia de Software”. Rio de Janeiro: McGraw-Hill. 2006.
- Rocha, V. C. (2009) “Metodologia para implementação do MPS.BR utilizando o ambiente WebAPSEE”. Dissertação de Mestrado, orientador Prof. Dr. Elói Luiz Favero – PPGEE – UFPA. Belém – Brasil.
- SEI - Software Engineering Institute (2010) "CMMI for Development Version 1.3". Carnegie Mellon. Pittsburgh-USA.
- SOFTEX – Associação para Promoção da Excelência do Software Brasileiro (2011a) “MPS.BR – Melhoria de processo do Software Brasileiro – Guia Geral:2011”.
- SOFTEX – Associação para Promoção da Excelência do Software Brasileiro (2011b) “MPS.BR – Melhoria de processo do Software Brasileiro – Guia de Implementação – Parte 2: Fundamentação para Implementação do Nível F do MR-MPS:2011”.
- Teles, M. P., Oliveira, S. R. B. (2010) “SPIDER-QA: A Systematic Approach for Quality Assurance Process”. In: Software Engineering and Applications - SEA, Marina Del Rey - EUA.
- Teles, M. P. (2011) “Spider-QA: Um Ferramental de Apoio ao Processo de Garantia da Qualidade no Contexto de Modelos e Norma para Processo de Software”. Dissertação de Mestrado, orientador Prof. Dr. Sandro Ronaldo Bezerra Oliveira. PPGCC – UFPA. Belém – Brasil.
- The International Organization for Standardization/ The International Electrotechnical Commission (2008) “ISO/IEC 12207 Systems and software engineering– Software life cycle processes”. Geneve.