

Proposta de um Processo Sistematizado de Controle de Manutenção de *Software* para Micro e Pequenas Empresas

Dayler Vinicius M. Alves¹, Carlos Renato Storck¹

¹Instituto de Informática – Pontifícia Universidade Católica de Minas Gerais (PUC)
39.140-000 – Guanhães – MG – Brasil

dlr_vinicius@hotmail.com, storckbr@gmail.com

Abstract. *A good process of software maintenance is critical to achieving product quality and customer satisfaction. This work intends to choose a control tool maintenance and defining a process for software maintenance based on their functionality and can be deployed effectively in micro and small software development companies, in order to manage the entire process maintenance of enterprise software.*

Resumo. *Um bom processo de manutenção de software é fundamental para alcançar a qualidade do produto e satisfação de seus clientes. Este trabalho se propõe a escolher uma ferramenta de controle de manutenção e definir um processo de manutenção de software baseado em suas funcionalidades e que possa ser implantado de forma eficaz em micro e pequenas empresas de desenvolvimento de software, com o intuito de gerenciar todo o processo de manutenção de software da empresa.*

1. Introdução

Atualmente empresas desenvolvedoras de *software* preocupam-se em ter produtos que atendam as necessidades de seus clientes, mas muitas vezes não se preocupam com a qualidade do produto final, e quando se preocupam não dão o valor devido. Quando se fala em qualidade de *software*, é possível ter várias definições para o termo. Diante dessa complexidade na definição da palavra qualidade, Pressmam (2005) sugere que a qualidade de *software* seja implementada e não somente uma ideia ou desejo que uma organização venha a ter. A busca constante pela qualidade não se faz apenas no começo do projeto ou no seu final realizando testes, mas sim em um processo que abranja todos os ciclos de desenvolvimento da engenharia de *software* e que envolva todos os membros do projeto.

Sabendo que, para que o sistema mantenha a qualidade é necessário que ao realizar alguma alteração, tenha-se o controle da mesma, torna-se necessário o uso de recursos para o desenvolvimento de mecanismos e processos para realizar a avaliação, controle e modificações do *software*, com o objetivo principal de reduzir a quantidade de esforço no processo de manutenção. Neste trabalho é escolhida uma ferramenta e definido um processo de manutenção de *software* a ser seguido por Micro e Pequenas Empresas (MPEs) no ramo de desenvolvimento de *software*.

2. Metodologia

Inicialmente foi realizado o levantamento do referencial teórico e apresentado os conceitos de manutenção de software e processo de gerenciamento da manutenção. Em seguida, baseado em um estudo sobre ferramentas livres de controle de mudanças foram escolhidas duas ferramentas para gerenciar o processo de manutenção de *software*, do qual apenas uma dessas ferramentas será utilizada e esta deve permitir o gerenciamento dos defeitos e atualizações requeridas ao *software*.

O critério escolhido para escolha da ferramenta é a que melhor se adapte as necessidades das MPEs. Para isso foi utilizada a norma ISO/IEC 9126-1 que descreve um modelo para a avaliação da qualidade do produto de *software*. Baseado na norma, foi escolhida a ferramenta realizando a definição do processo baseado em suas funcionalidades. Por fim, foram verificados os benefícios que um processo sistematizado pode trazer para uma MPE.

3. Referencial Teórico

3.1 Manutenção de Software

A manutenção de *software* pode ser definida como uma parte do ciclo de desenvolvimento do *software*. Para Sommerville (2003) “a manutenção de *software* é o processo geral de modificação de um sistema depois que ele foi colocado em uso.” Mesmo nos produtos de *software* desenvolvidos com os melhores processos, enfatiza Paula Filho (2003), alguns defeitos poderão escapar até a versão de operação, para que então sejam descobertos pelo usuário. Pfleeger (2004) completa dizendo que “qualquer trabalho efetuado para modificar o sistema depois que ele estiver em operação, é considerado manutenção.” Rezende (2005) destaca que “de maneira geral, todo *software* sofre manutenções, sejam elas para simples ajustes pós-implantação, ou por melhorias substanciais, por força da legislação e, finalmente, por estar gerando erros.” Neste sentido Sommerville (2003) lembra que existem três diferentes tipos de manutenção de *software*: manutenção para reparar os defeitos no *software*, manutenção para adaptar o *software* a um ambiente operacional diferente e manutenção para acrescentar funcionalidade.

3.2 Processo de Gerenciamento da Manutenção

Para Sommerville (2003) as mudanças em sistemas de *software* sempre irão ocorrer, pois as necessidades e os requisitos organizacionais modificam o tempo de vida útil do sistema, requerendo então que mudanças correspondentes sejam feitas no *software*. Afirma ainda que o processo de gerenciamento de mudanças pode ser iniciado durante a fase de teste ou até mesmo após a entrega ao cliente, sendo que o processo deve assegurar que seja realizada a análise dos custos e benefícios da mudança, garantindo que esta, seja feita de maneira controlada.

4. Desenvolvimento

O desenvolvimento deste trabalho será a escolha de uma ferramenta de controle de mudanças, que tem por objetivo automatizar o processo de manutenção de *software* de uma MPE. Será apresentado o processo realizado para a escolha da ferramenta.

Espera-se que com uma ferramenta automatizada de controle de mudanças, a empresa possa ter uma base para a inserção de informação relacionada com as alterações no *software* e através dos relatórios e métricas geradas por esta ferramenta tenha-se subsídios para o auxílio na tomada de decisão e melhoria do processo como um todo.

4.1 Levantamento das Ferramentas

Para realizar o levantamento das ferramentas se fez necessário realizar um estudo sobre ferramentas livres de controle de mudanças. Neste estudo, foi verificado o trabalho de alguns autores, avaliando a viabilidade de se utilizar uma destas. Como principal característica, esta ferramenta deve ser capaz de registrar, avaliar e rastrear todas as mudanças no projeto, desde sua requisição até a aceitação ou não de sua implementação.

Foi definida a avaliação de duas ferramentas que obtiveram bons resultados em trabalhos de outros autores. No estudo realizado verificou-se que duas ferramentas obtiveram bons resultados nos trabalhos de Cardias Junior et al. (2010) e Reis (2002), sendo elas o Mantis e o Bugzilla respectivamente.

Para a escolha da ferramenta foi utilizada a norma ISO/IEC 9126-1 que descreve um modelo que observa a qualidade do produto de *software*, dividindo-se basicamente em duas partes, sendo elas qualidade interna / externa e qualidade em uso. A qualidade interna e a qualidade externa são divididas em seis características e estas são divididas em subcaracterísticas externas que são resultados de atributos internos do *software*. A qualidade em uso é dividida em quatro características de utilização, sendo estas os efeitos da combinação das seis características anteriores. Foram estabelecidas métricas para a realização da avaliação das ferramentas de controle de mudanças. As métricas foram divididas em três níveis: Atende, Atende Parcialmente e, Não Atende.

Nesta etapa, com base nos critérios de avaliação da norma NBR ISO/IEC 9126-1, serão realizadas comparações entre as ferramentas de controle de mudanças. Os comparativos dos critérios definidos estão descritos abaixo no Quadro 1.

Critérios	Mantis			Bugzilla		
	A	AP	NA	A	AP	NA
Requisito1 Anexação de arquivos	x			x		
Requisito2 Envio de e-mail	x			x		
Requisito3 Monitoração e acompanhamento das tarefas	x			x		
Requisito4 Hierarquia de nível de usuário	x			x		
Requisito5 Validação de informações	x			x		
Requisito6 Recuperação de dados	x			x		
Requisito7 Facilitar a utilização e aprendizado	x			x		
Requisito8 Interface intuitiva	x				x	
Requisito9 Adição de novos campos	x			x		
Requisito10 Compatibilidade com navegadores	x			x		
Requisito11 Compatibilidade com sistemas operacionais	x			x		
Requisito12 Facilidade de instalação	x					x

Quadro 1. Avaliação das ferramentas

Entre os requisitos avaliados, observa-se que o critério de avaliação Requisito8 da ferramenta Bugzilla foi classificado como Atende Parcialmente (AP), por causa da

interface disponibilizada somente em inglês que pode ser um fator que dificulte sua utilização pelos usuários. Assim como o Mantis, o Bugzilla pode ser instalado em qualquer sistema operacional, necessitando basicamente de um servidor web, um *browser* que também pode ser qualquer um e comunicação com o servidor no qual está hospedado. O critério de avaliação Requisito12 da ferramenta Bugzilla foi classificado como Não Atende (NA), por causa da documentação falha que deixa a desejar em certos pontos da instalação, o que torna demorado este processo.

A ferramenta escolhida foi o Mantis, devido aos bons resultados obtidos no comparativo das ferramentas no Quadro 1 e no trabalho de Cardias Junior et al. (2010), que mostra a utilização da ferramenta com o objetivo de gerenciar os relatos dos problemas identificados, acompanhando sua evolução desde a solicitação de mudança até seu desfecho. Os bons resultados da ferramenta se devem aos relatos (*issues*) gerados nela, que contam com um ciclo de vida próprio, passando pelos vários estados que um relato (*issue*) se enquadra até seu fechamento.

5. Definição do Processo

De acordo com as funcionalidades da ferramenta escolhida, foi obtido como resultado um processo sistematizado através de uma versão simplificada e adaptada do Processo de Manutenção do Práxis, especificado de maneira que se consiga manter o controle do processo de maneira simples. A Figura 1 representa o fluxograma do processo.

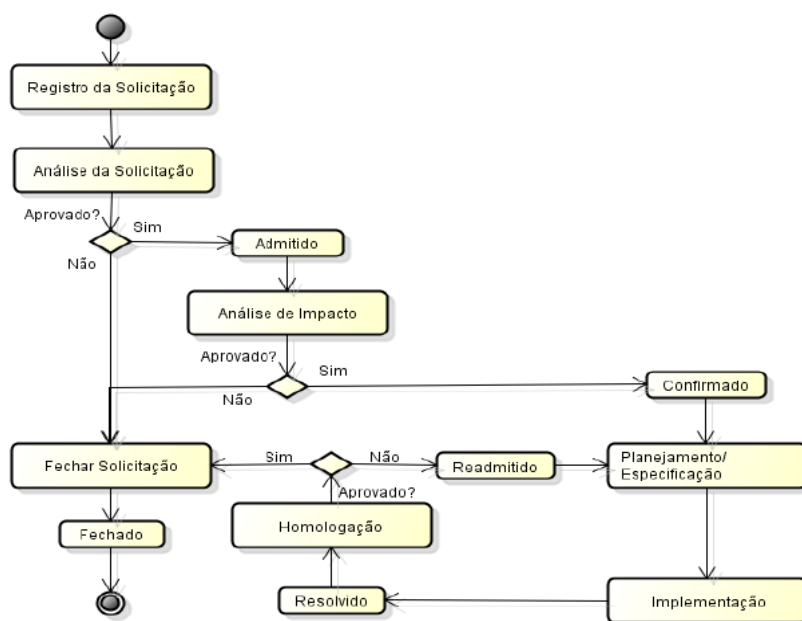


Figura 2. Fluxograma do processo de manutenção de software

Inicialmente, o cliente faz a requisição da mudança, seja ela adaptativa, corretiva ou evolutiva, e então esta mudança requerida é relatada no sistema, escolhendo o projeto ao qual pertence a mudança, seu *status* inicial deverá ser atribuído ao gerente. Assim que o caso é salvo, é enviado ao gerente um e-mail notificando-o de sua responsabilidade pela análise da solicitação. Em seguida, após o gerente realizar a análise desta solicitação, verificando sua relevância para o sistema e se esta solicitação

for aprovada, seu *status* no sistema deverá ser mudado para admitido e deverá seguir para a análise de impactos, que avaliará o que tal alteração poderá ocasionar no sistema. Ao realizar a análise de impacto de uma alteração e se esta for aprovada, seu *status* deverá ser alterado para confirmado, caso não seja aprovada a solicitação deverá ser fechada.

Após a aprovação da alteração, deve-se realizar o planejamento e especificação de como será implementada a solução para a mudança proposta. O gerente deverá atribuir ao programador a responsabilidade de implementar a mudança e realizar os testes necessários. Após realizadas as atividades anteriores, o *status* da solicitação deverá ser alterado para resolvido. É aguardado um período de homologação pelo cliente, para mudar o *status* do pedido de mudança para fechado. Caso não tenha sido resolvido, ele poderá ser readmitido, caso contrário a solicitação deverá ser fechada.

6. Resultados

A definição de um processo sistematizado de controle de manutenção de software baseado em uma ferramenta pode trazer bons resultados em relação à qualidade do software. Porém, a manutenção de software necessita de uma equipe específica empenhada no desenvolvimento deste processo. Visando isto, foi necessário desenvolver um processo de manutenção que atendesse as MPEs, sabendo-se que estes tipos de empresas dispõem de poucos recursos de pessoal.

Mesmo com os indícios teóricos de que o processo proposto irá auxiliar o controle de modificações de software, é necessário que sejam realizados estudos experimentais para que sejam verificadas as reais contribuições do uso da abordagem e de suas deficiências. O processo definido facilitará a realização destes experimentos.

7. Conclusão

Com este trabalho, foi possível verificar a importância de um processo de manutenção de *software* nas MPEs, a fim de se obter a qualidade do *software*. A partir das funcionalidades da ferramenta Mantis, um processo foi implementado, baseado na versão simplificada e adaptada do Processo de Manutenção do Práxis, sendo possível realizar a centralização dos relatos, não correndo o risco de qualquer pedido de manutenção não chegar ao conhecimento da equipe de desenvolvimento.

Com base nos resultados encontrados neste estudo, verificou-se que o processo pode ser melhorado em abordagens de trabalhos futuros. O presente trabalho não tratou da integração com um sistema de controle de versões, que é um *software* que tem como finalidade o gerenciamento de versões no desenvolvimento de um *software*. Para trabalhos futuros, recomenda-se a integração entre o Mantis e um sistema de controle de versões compatível, a fim de se obter a automatização das atualizações de relatos (*issues*) no Mantis, a partir de *commits* de alterações no sistema de controle de versões. Cabe também como trabalho futuro a verificação e otimização do fluxo do processo de desenvolvimento de *software* definido neste trabalho.

Referências

- Bugzilla, Documentação. "Bugzilla", <http://www.bugzilla.org>, Maio.
- Campos, F. M. (2007) "Quais são as reais características da qualidade da NBR ISO/IEC 9126-1?", <http://www.testexpert.com.br/?q=node/126>, Maio.
- Cardias Junior, A. B., Bentes, L. N., Ronny, M., Bezerra Oliveira, S. R. and Yoshidome, E. (2010) "Uma Análise Avaliativa de Ferramentas de Software Livre no Contexto da Implementação do Processo de Gerência de Requisitos do MPS.BR", In: XIII Workshop em Engenharia de Requisitos, Cuenca.
- Dias, A. F. (2009) "Comparação entre ferramentas de controle de mudança", http://www.pronus.eng.br/artigos_tutoriais/analises/ferramentasControleMudanca.php, Maio.
- Lopes, L. G., Murta, L. and Werner, C. (2005) "Controle de Modificações em Software no Desenvolvimento Baseado em Componentes", In: II workshop de manutenção de software moderna, Manaus.
- Maesi, C. et al. (2009) "NBR ISO/IEC 9126-1", <http://www.slideshare.net/alinebicudo/slides-nbr-isoiec-91261>, Maio.
- Mantis, Documentação. "Mantis", www.mantisbt.org, Maio.
- Paula Filho, W. P. (2003), Engenharia de software: Fundamentos, Métodos e Padrões, LTC - Livros Técnicos e Científicos, 2th edition.
- Pfleeger, S. L.. (2004), Engenharia de Software: Teoria e Prática, Prentice Hall, 2th edition.
- Pressman, R. S. (2006), Engenharia de software, McGraw-Hill, 6th edition.
- Rezende, D. A. (2005), Engenharia de Software e Sistemas de Informação, Brasport, 3th edition.
- Reis, C. R. (2002) "Uma Visão Geral do Bugzilla, uma Ferramenta de Acompanhamento de Alterações", In: XVI Simpósio Brasileiro de Engenharia de Software, Gramado.
- Sommerville, I. (2003), Engenharia de software, Pearson Addison Wesley, 6th edition.