

Utilização da ferramenta Mantis para gerenciamento de defeitos de testes de software

Marcos Flávio de Souza Reis
INPE

São José dos Campos
Brasil

marcosfsreis@gmail.com

Ana Maria Ambrósio
INPE

São José dos Campos
Brasil

ana@dss.inpe.br

Mauricio Ferreira Gonçalves
INPE

São José dos Campos / SP
Brasil

mauricio@ccs.inpe.br

RESUMO

Este artigo tem a finalidade de apresentar uma solução a pequenas e médias empresas de desenvolvimento de software, no gerenciamento de defeitos de seus produtos através de ferramentas gratuitas. Além disso, este trabalho também relata a experiências e as lições aprendidas durante os cinco meses de realização de testes funcionais em um produto em versão alfa, sendo apoiada pela ferramenta *Open Source* Mantis, especialista em gerenciamento de defeitos.

Palavras chaves

Testes, Verificação, Gerenciamento de defeitos, Mantis.

1. INTRODUÇÃO

A atividade de teste faz parte do ciclo de desenvolvimento de um software. Myers [1] define testes como um processo de executar um programa com o objetivo de encontrar erro. Já Moreira Filho [2] descreve que os testes devem verificar se o software está fazendo o que deveria fazer, de acordo com os seus requisitos, e não está fazendo o que não deveria fazer.

A realização dos testes de software nas organizações é essencial para a finalização do produto, pois revelam defeitos que prejudicarão o funcionamento do software. Estudos, como o publicado por Myers [1] ou por Pressman [3], mostram que quanto antes os defeitos forem encontrados, menores serão os custos para as organizações. A pior situação é quando um defeito é encontrado em produção, podendo causar grandes prejuízos a um projeto e a imagem da empresa.

A atividade de testes consome boa parte dos recursos de um projeto de software, por isso é importante a atividade de gerenciamento dos testes, para aumentar a eficiência e a qualidade dos testes realizados. Pressman [3] relata que este esforço pode chegar a 40% do total.

Durante o ciclo de desenvolvimento de uma nova versão de um produto de software, é necessária a realização de uma bateria de testes para assegurar que as novas funcionalidades, e também as já existentes, estejam funcionando conforme esperado.

Em empresas de pequeno e médio porte, com cerca de 100 funcionários, os testes podem envolver diversos departamentos, exigindo a criação de um ambiente para gerenciar a interação entre os relatores e os solucionadores de defeitos.

As informações obtidas pela realização dos testes devem ser bem gerenciadas, pois elas irão demonstrar não só a qualidade atual do produto, como também auxiliar na tomada de decisões estratégicas da empresa.

Um processo de verificação de software, utilizando a técnica de testes funcionais, exige que seja feito um bom planejamento, o que significa: Escrever detalhadamente os casos de testes, preparar a equipe envolvida, proporcionar um ambiente adequado para a execução dos testes e principalmente, gerenciar todas as informações geradas durante este processo.

Uma das principais dificuldades da verificação de software em empresas deste porte está em realizar com eficiência, segurança e com baixo custo o gerenciamento dessas informações.

No mercado existem várias ferramentas, algumas pagas e outras gratuitas, que fornecem suporte a várias etapas do processo de verificação de software. Entre elas estão às ferramentas especialistas em gerenciamento de defeitos, que visam registrar e acompanhar todos os incidentes encontrados durante a realização de testes.

2. GERENCIAMENTO DE DEFEITOS

O gerenciamento dos defeitos pode ser definido [4] como um conjunto de processos e procedimentos que visa armazenar e gerenciar a informação a respeito dos defeitos encontrados ao longo do ciclo de vida de uma aplicação, desde o projeto até a sua retirada ou saída de produção.

A definição de erro e defeito é importante para a compreensão deste conceito. Segundo Bastos [5] um erro é o resultado de uma falha humana e o defeito é o resultado de um erro existente num código ou num documento.

Um processo básico [4] para o gerenciamento dos defeitos consiste em:

- 1) Detectar o defeito. Um relatório do defeito com informações com Identificação do defeito, descrição, prioridade, provas e evidências da sua existência deve ser elaborado.
- 2) Validar o relato do defeito. Uma análise é feita e se for constatado que a falha ocorre, o mesmo será aceito para correção.
- 3) Atribuir a um desenvolvedor para correção.
- 4) Realizar a correção no código do software.
- 5) Testar o software novamente para verificar se o defeito realmente foi corrigido.
- 6) Se o defeito foi corrigido, a ocorrência é fechada.

Um defeito quando gerenciado, é acompanhado da sua detecção até a sua resolução, registrando vários tipos de informações, tais como: Identificação do defeito, Descrição, Relator, Severidade,

Prioridade, Risco associado, Status, Provas e evidências da existência do defeito.

3. FERRAMENTA DE GERENCIAMENTO DE DEFEITO

Sendo a detecção de defeitos uma das finalidades dos testes de software, todos os resultados alcançados que sejam diferentes do esperado devem ser registrados. Este acontecimento também é denominado como incidente e precisa ser rastreável desde a descoberta até a sua correção.

Durante a execução dos testes os incidentes encontrados precisam ser reportados a equipe desenvolvedora, descrevendo detalhes do problema encontrado, evidências do defeito e os passos para reprodução.

Uma ferramenta de gerenciamento de defeitos deve funcionar como um repositório de tais informações e auxiliar na priorização dos incidentes, na atribuição de tarefas (ex: corrigir o erro ou executar um teste de confirmação) e na atribuição de status (rejeitado, pronto para teste ou duplicado).

Estas ferramentas permitem que o progresso do incidente seja monitorado durante o projeto. Muitas vezes provêm o suporte para análises estatísticas e relatórios.

A ferramenta Mantis [4] apóia o gerenciamento de defeitos de software e dispõe de características importantes para essas atividades, tais como:

- Permitir a criação de campos personalizados;
- Personalizar o processo de gerenciamento de defeitos;
- Ser multiplataforma (Pode ser utilizando em Windows, Linux e outros sistemas operacionais);
- Trabalhar com banco de dados gratuito (MySQL);
- Permitir o controle de acesso e perfil de usuários;
- Exibir métricas dos defeitos encontrados;
- Ser uma ferramenta Web, desenvolvida em PHP, por isso não precisa ser instaladas nas máquinas que irão utilizá-las.
- Estar em constante evolução;
- Dispor de uma interface amigável e de fácil aprendizado.

A Mantis é desenvolvida e mantida pela comunidade MantisBT. Esta ferramenta começou a ser criada em 2000, como solução para o gerenciamento de defeitos e tem sofrido melhorias e evoluções até os dias atuais.

Esta ferramenta é livre tanto para utilização como para evolução, permitindo que cada empresa personalize a ferramenta de acordo com as suas necessidades. Atualmente ela se encontra na versão 1.2.8 e pode ser adquirida através do site: <http://www.mantisbt.org>.

4. DESCRIÇÃO DO CENÁRIO

O cenário apresentado foi executado em uma empresa de desenvolvimento de um software especialista, com cerca de 100 funcionários, como forma de avaliar a qualidade de uma nova versão do software que seria disponibilizada ao mercado. Este fase de testes também é conhecida como testes na versão alfa.

Uma bateria de testes foi organizada, envolvendo cerca de 20 pessoas, entre elas testadores e desenvolvedores. O ciclo de verificação do sistema deve duração de cinco meses, onde todo o software foi testado utilizando testes negativos, que nada mais são que testes que têm a finalidade de encontrar defeitos no software.

A empresa não disponha de uma equipe especialista em testes, contando apenas com a figura de um único testador alocado ao time de desenvolvimento, que possuía oito desenvolvedores.

Além de constatar a qualidade do produto, todos os defeitos encontrados deveriam ser corrigidos antes do prazo estipulado para a bateria de testes. Todo defeito encontrado deveria ser verificado posteriormente pelo relator, para assegurar que o problema havia sido solucionado.

A gerência dos testes foi feita através de planilhas do Excel, onde as partes do software foram divididas e os casos de testes foram escritos para cada uma dessas partes.

Para o registro dos defeitos encontrados durante a bateria de testes, foram levantadas várias soluções, como: Planilhas do Excel, Envio de e-mail, Documentos do Word e Ferramentas especializadas em Gerenciamento de defeitos.

Logo de inicio foi detectado que a utilização de uma ferramenta especialista era essencial para o bom andamento dos testes, por isso foi feito um levantamento das ferramentas gratuitas e disponíveis. Durante a análise das ferramentas de gerenciamento de defeitos, duas se apresentaram como candidatas: a Mantis e a BugZila.

Após uma breve avaliação das duas ferramentas, optou-se pela utilização da Mantis. Nos critérios de avaliação das ferramentas, foram definidas três características principais: Atendimento a necessidade do projeto, facilidade de instalação nas máquinas e facilidade de utilização/aprendizagem da ferramenta.

O passo seguinte foi à configuração da ferramenta, que passou pelo cadastro dos usuários envolvidos, pela criação dos releases do produto, pela criação dos campos personalizados, a definição do processo de testes e das informações gerais do projeto de testes.

O recurso de criação de campos personalizados foi essencial para a adaptação da ferramenta as necessidades da empresa. Algumas informações importantes para ao produto e ao processo definido, precisavam estar registradas a cada defeito encontrado, pois eram relevantes para a correção do defeito e para as métricas esperadas para avaliação do produto no final da bateria de testes.

Uma característica importante do produto em teste é a possibilidade de utilizá-lo em dois Sistemas Gerenciadores de Bancos Dados (SGBD). Este fator exige que o software seja testado por completo em cada um dos bancos de dados.

5. EXECUÇÃO DOS TESTES

O software foi dividido em 26 módulos, sendo que cada módulo deveria ser testado em dois SGBDs, resultando em 52 itens de testes a serem verificados. Para cada item de teste foi desenvolvida uma planilha do Excel com um conjunto de casos de testes mínimos a serem executados. O testador tinha liberdade para criar testes aleatórios durante a verificação do módulo.

Inicialmente foi realizada uma reunião com todos os envolvidos nos testes, o que inclui desenvolvedores, testadores e analistas de

suporte ao cliente, onde os módulos foram divididos e a ferramenta Mantis foi apresentada. A utilização da Mantis foi muito bem aceita pelos usuários, principalmente pelo fato da ferramenta possuir uma interface bastante simples para o relato do defeito, além de uma tela inicial com muitos filtros, o que facilitava na localização dos defeitos.

Com a divisão dos trabalhos concluída e o ambiente preparado, uma primeira rodada de testes foi executada e os defeitos foram registrados na Mantis. A equipe de desenvolvimento foi dividida em duas, sendo uma parte que trabalhava em paralelo com os testes na correção dos defeitos e outra parte que executava os testes.

Assim que os principais defeitos dessa primeira rodada foram corrigidos, um release do produto foi disponibilizado para que os relatores pudessem verificar se os defeitos realmente foram corrigidos.

Esse ciclo ocorreu por diversas vezes, até que todos os módulos pudessem ser testados, corrigidos e reavaliados.

Durante a realização dos testes, a ferramenta Mantis teve grande importância no acompanhamento dos testes, porque além de disponibilizar uma interface comum entre os testadores e os desenvolvedores, ela disponibilizava métricas e gráficos que apresentavam o andamento e os resultados dos testes.

A Figura 1 apresenta uma tabela extraída da ferramenta Mantis com informações sobre as resoluções dadas aos defeitos relatados, o que auxilia na avaliação do produto e na qualidade dos testes realizados, porque pode apresentar, por exemplo, quantos defeitos relatados não eram erros no sistema, que neste caso era de aproximadamente 10%.

Por Resolução	aberto	resolvido	fechado	total
aberto	15	0	1	16
corrigido	0	0	717	717
reaberto	2	0	0	2
incapaz de reproduzir	0	0	25	25
não corrigível	0	0	3	3
duplicado	0	0	22	22
não é um caso	0	0	71	71
suspenso	2	0	1	3
não será corrigido	0	0	2	2

Figura 1 - Resolução dos defeitos

Outra informação que foi utilizada para a avaliação da qualidade do produto em teste foi a gravidade dos problemas detectados. A ferramenta Mantis também disponibilizou essa informação a gerencia através de gráficos, conforme apresentado na Figura 2.

Os resultados dos testes apresentaram um total de 842 relatos, onde existiam melhorias de interface, tratamentos de erros, erros de programação, necessidade de recursos adicionais e falhas de especificação de funcionalidades.

As informações preenchidas nos campos personalizados na ferramenta também apresentaram importantes informações sobre o produto e os testes realizados.

A ferramenta Mantis também foi bastante consumida em relação as suas estatísticas de relatores, desenvolvedores e aos releases que corrigiam os problemas relatados.

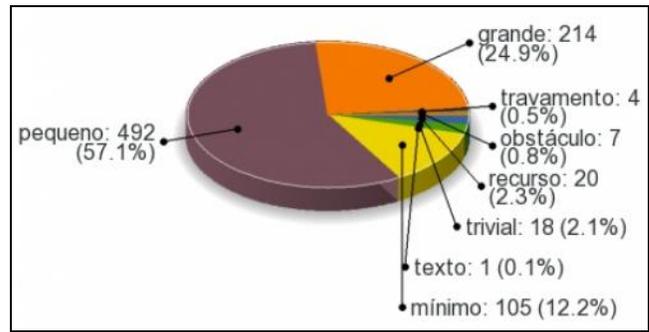


Figura 2 - Gravidade dos defeitos

6. RESULTADOS ALCANÇADOS E LIÇÕES APRENDIDAS

A realização dos testes em versão alfa foi de grande relevância para a empresa, pois 861 relatos puderam ser avaliados, corrigidos e desenvolvidos antes que os usuários finais tivessem contato com o produto. Considerando o estudo de Myers, onde o custo do defeito é multiplicado por 10 a cada fase do ciclo de vida do software, foi feita uma grande economia financeira e de credibilidade do produto.

Além disso, durante a realização dos testes foi detectado dois pontos de melhorias dentro da própria empresa. O primeiro por envolver diversos departamentos da empresa, foi possível detectar também a necessidade de realizar alguns treinamentos internos em alguns módulos especificados do software.

E o segundo em relação à estrutura da empresa, a bateria de testes mostrou a necessidade de criar uma equipe especialista em testes, independente da equipe de desenvolvimento, para que as melhores práticas de testes pudessem ser buscadas e para que os testadores tivessem mais autoridade e autonomia durante a realização dos testes. Essa equipe já foi criada e já gera resultados positivos.

A ferramenta Mantis foi muito importante para a realização desses testes, pois foi responsável por realizar a interface entre todos os envolvidos, centralizar as informações, disponibilizar informações sintetizadas e facilitar o registro dos incidentes.

Por se tratar de uma ferramenta *Open Source* a Mantis se apresentou como uma ótima alternativa para o gerenciamento de incidentes, pois não apresentou nenhum custo à empresa e seus recursos de personalização de campos e até mesmo do processo de testes (este último não utilizado neste trabalho), permitiram a adequação das características da empresa e do produto a ferramenta.

A Mantis não foi explorada em sua totalidade, pois alguns recursos como personalização dos perfis de usuários, ciclo de vida do defeito e etiquetas não eram necessários para este projeto.

Durante a realização dos testes o único problema encontrado na utilização da ferramenta era em respeito à atualização em conjunto dos campos personalizados, que em alguns momentos não permitia essa atualização sem nenhum motivo evidente.

Para um gerenciamento de testes adequado, muitas informações são geradas, não se limitando ao registro de defeitos, como por exemplo, o registro dos casos de testes, a organização e distribuição dos testes e o andamento da execução dos testes.

Sendo assim, foi criado através de um trabalho acadêmico de mestrado, um ambiente denominado como VVTeste, que tem a finalidade de integrar ferramentas *Open Source* de geração automática de casos de testes, de gerenciamento de testes e de gerenciamento de defeitos (Mantis), exportando os dados gerados a uma base de conhecimento única, capaz de realizar novas análises gráficas dos resultados de diversos projetos de testes software e servir de fonte de lições aprendidas.

7. REFERÊNCIAS

- [1] Myers, Glenford. 1979. The art of software testing. Ed. Wiley, New York.
- [2] Moreira Filho, Trayahú. Rios, Emerson. 2003. Projeto & Engenharia de Software – Teste de Software. Ed. Alta Books, Rio de Janeiro.
- [3] Pressman, R. S. 2002. Engenharia de Software. Ed. McGraw Hill,
- [4] Molinari, Leonardo. 2008. Testes funcionais de softwares. Ed. Visual Books, Florianópolis.
- [5] Bastos, Aderson, ET AL. 2007. Base de conhecimento em testes de software. Ed. Martins, São Paulo.
- [6] MantisBT. 2002. Site oficial <http://www.mantisbt.org>, Acessado em 30 de dezembro de 2011.