

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2569

Uma Introdução ao Bugzilla

Leonardo Jorge Quemello
Alexandre Michetti Manduca
Renata P. M. Fortes

Nº 253

RELATÓRIOS TÉCNICOS



São Carlos – SP
Fev./2005

SYSNO	1427924
DATA	1 / 1
ICMC - SBAB	

Uma Introdução ao Bugzilla

Leonardo Jorge Quemello, Alexandre Michetti Manduca, Renata P. M. Fortes

{leonardo, michetti}@grad.icmc.usp.br , renata@icmc.usp.br

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
São Carlos – SP, Brasil. C.P: 668 CEP: 13560-970
São Carlos -SP - fevereiro de 2005

Índice

1- INTRODUÇÃO.....	3
2 – APRESENTAÇÃO DE BUGZILLA.....	3
3 - OBJETIVO E RECURSOS DE BUGZILLA.....	4
4 – VISÃO GERAL DO FUNCIONAMENTO DE BUGZILLA.....	5
4.1 – Produtos e Componentes.....	5
4.2 – Contas de usuário.....	5
4.3 – Níveis de usuário e permissões.....	5
4.4 – Ações.....	6
4.5 – Bugs.....	6
4.6 – Versões e Milestones.....	6
4.7 – Grupos.....	6
4.7 – Ciclo de vida dos bugs e funcionamento geral.....	7
5 - CARACTERÍSTICAS DE UM BUG.....	7
6 - CRIAÇÃO DE UM NOVO BUG.....	10
7 - CICLO DE VIDA DO BUG.....	12
8 - VISUALIZAÇÃO DE UM BUG.....	14
9 - DEPENDÊNCIAS ENTRE BUGS.....	15
10 - VOTAÇÃO.....	16
11 - CONFLITOS DE ACESSO SIMULTÂNEO.....	16
12- CONSULTAS.....	17
12.1 – Consultas.....	17
12.2 – Lista de bugs.....	18
12.3 – Buscas avançadas.....	18
12.4 – Salvar consultas.....	19
12.5 – Exportação de listas de bugs.....	19
12.6 – Quicksearch.....	19
13 - RELATÓRIOS.....	20
14 - PREFERÊNCIAS DE USUÁRIO.....	21
15 – Referências Bibliográficas.....	22

1- INTRODUÇÃO

Por muito tempo, quando um software era desenvolvido de maneira distribuída, o controle de alterações e defeitos em um software era realizado por meio de listas de discussão e e-mail. Embora bastante simples, este procedimento (listas de discussão e de e-mail) não organiza com qualidade as alterações no software, é passível de erros e os defeitos e alterações considerados menos significativos eram ignorados ou esquecidos.

Atualmente, muitas organizações estão descobrindo que o efetivo controle de alterações aumenta a produtividade e a qualidade, reduz tempo de trabalho, aumenta a satisfação de usuários e consumidores, e diminui custos.

Bugzilla é uma ferramenta de controle e gerência de alterações de software. Ela registra e organiza *bugs*¹ e suas características, e permite a visualização dos mesmos. O controle de alterações permite que grupos de desenvolvedores ou indivíduos organizem efetivamente o tratamento aos problemas e mudanças em seus produtos.

As ferramentas como Bugzilla podem ser utilizadas para apoiar a fase de manutenção de software e proporcionar a esse processo maior visibilidade, comunicação direcionada, e um registro histórico das mudanças e da evolução da base de código.

Este relatório apresenta uma introdução aos conceitos iniciais da ferramenta Bugzilla, de forma resumida, baseado em "The Bugzilla Guide" [1], com o objetivo de proporcionar um meio simples para os iniciantes e interessados na ferramenta.

2 – APRESENTAÇÃO DE BUGZILLA

Bugzilla² é um software livre distribuído sob a licença MPL (*Mozilla Public License*). Originalmente desenvolvido na linguagem TCL por Terry Weissman, para substituir um banco de dados de alterações utilizado internamente pela Netscape Communications, posteriormente, Bugzilla foi reescrito em Perl e assim se mantém até hoje.

A ferramenta Bugzilla foi desenvolvida para apoiar o projeto Mozilla³. Atualmente, é utilizada em vários projetos, tanto livres quanto proprietários. As organizações e projetos que já utilizaram a ferramenta ou a utilizam, entre outras, são: NASA, Motorola, Redhat, Conectiva, GNOME, KDE, Apache, IBM e OpenOffice.

¹ O termo *bug* significa, neste contexto, defeito, nova funcionalidade, ou mesmo melhorias que acarretem alterações.

² www.bugzilla.org

³ www.mozilla.org

Bugzilla é um software implementado como aplicação web, ou seja, seus usuários a utilizam a partir de um navegador (*browser*) web para acessar sua interface. Usa programas CGI⁴ com scripts Perl. Como resultado do processamento do CGI, são gerados códigos HTML, Javascript e CSS⁵. O Template Toolkit, uma linguagem de especificação de modelos de documentos HTML, também é utilizado para elaboração de páginas. A ferramenta Bugzilla é suportada pelas plataformas UNIX, Linux e Windows, utiliza banco de dados MySQL e um servidor web (Apache).

O desenvolvimento da ferramenta Bugzilla é realizado por pessoas geograficamente distribuídas, possui uma home-page principal em www.bugzilla.org. A ferramenta é mantida oficialmente por uma equipe de desenvolvedores liderados por Dave Miller. A comunidade Bugzilla conta com grande número de participantes e contribuidores. Na Figura 1 é mostrada a tela principal da ferramenta Bugzilla.

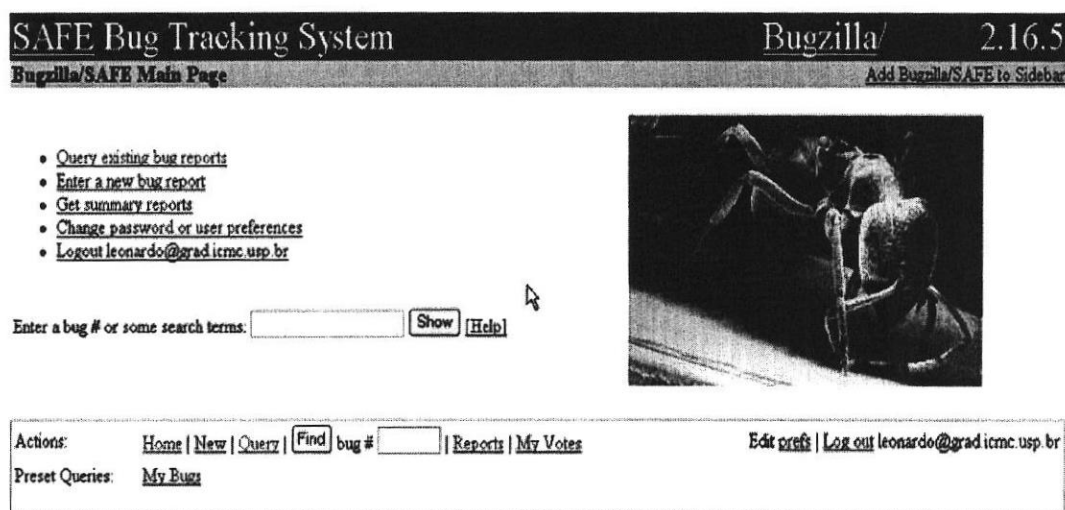


Figura 1: Tela principal de Bugzilla

⁴ CGI é sigla para *Common Gateway Interface*, um padrão e especificação para realizar interface entre aplicativos externos e servidores *web*. Um aplicativo CGI é qualquer programa projetado para aceitar e retornar dados em conformidade com a especificação CGI.

⁵ HTML é uma sigla para *HyperText Markup Language*, uma linguagem de marcação para publicação de documentos na Internet. Javascript é uma linguagem de script que permite projeto de páginas *web* interativas e dinâmicas. CSS é sigla para *Cascading Style Sheets*, um mecanismo que permite que estilos (fontes, cores, espaçamentos, etc) sejam adicionados a documentos HTML.

3 - OBJETIVO E RECURSOS DE BUGZILLA

O objetivo da ferramenta é auxiliar e ser um mediador do desenvolvimento de software, nas diversas etapas: de análise de requisitos, projeto, manutenção e testes. Bugzilla estabelece formas de organização e fornece vários recursos para apoiar o controle de alterações no desenvolvimento de programas em qualquer linguagem, plataforma e sistema operacional. Além disso, visa coordenar adequadamente o trabalho relacionado a alterações em um software para garantir a qualidade, aumentar a produtividade do desenvolvimento e reduzir o tempo de trabalho.

Para realizar seus objetivos, a ferramenta possui vários conceitos, recursos e utilidades que são, entre outros:

- Bugs e suas propriedades
- Mudanças de estado e ciclo de vida dos bugs
- Contas de usuário
- Produtos e Componentes
- Administração do sistema
- Banco de dados estrutural
- Comunicação direcionada sobre alterações
- Arquivos anexos
- Grupos
- Dependências entre bugs
- Interface de consulta
- Integração com correio eletrônico
- Consultas a bugs
- Relatórios e gráficos com estatísticas sobre bugs
- Preferências de usuário editáveis

A maioria desses conceitos serão descritos nas próximas seções.

4 – VISÃO GERAL DO FUNCIONAMENTO DE BUGZILLA

4.1 – Produtos e Componentes

Cada software adicionado a Bugzilla é considerado um *produto*, e cada produto é dividido em *componentes*. Os componentes são os módulos de desenvolvimento do software. Na organização do desenvolvimento de um software, os desenvolvedores devem organizar seu trabalho através da criação desses módulos, que precisam ser cadastrados em Bugzilla.

Cada componente possui um responsável. No momento do cadastro de cada componente de um produto, deve-se indicar o responsável pelo componente. Novos componentes podem ser criados a qualquer momento.

4.2 – Contas de usuário

Somente usuários cadastrados no sistema de Bugzilla podem acessar suas funcionalidades para algum produto (software). Para criar uma nova conta, deve-se acessar a página principal (home) da interface da ferramenta, clicar no link respectivo e criar a conta, introduzindo o e-mail e a senha do novo usuário. O e-mail é o login. Em qualquer momento, para entrar no sistema, são necessários fornecer o login (e-mail) e a senha. O novo usuário recebe automaticamente por e-mail a sua senha. O usuário é identificado no sistema através de sua conta, e receberá mensagens de vários tipos sobre trabalhos no sistema e trabalho com bugs no e-mail cadastrado.

4.3 – Níveis de usuário e permissões

Todo usuário cadastrado no sistema da Bugzilla possui certas permissões para acesso e utilização de algumas de suas funcionalidades. As permissões de um novo usuário são selecionadas pelo administrador do sistema. De acordo com as permissões que cada usuário possui, delimita-se níveis diferentes de usuários, por exemplo: donos de componentes e o administrador. O administrador é o usuário de nível mais alto dentro do sistema e pode realizar muitas ações, inclusive algumas exclusivas.

4.4 – Ações

As ações que um usuário pode executar, de acordo com suas permissões, estão sempre localizadas na barra de ações, no rodapé da página. Nela estão as opções: adicionar novo bug, fazer consultas, ver relatórios, editar preferências, entrar no sistema, sair do sistema, entre outras. A barra de ações está sempre presente em todas as páginas da interface de Bugzilla. A interação do usuário com o software realiza-se principalmente por essas ações.

4.5 – Bugs

Durante o desenvolvimento ou manutenção do software, qualquer sugestão de melhorias, adição de novas funcionalidades, ou erros encontrados são relatados em Bugzilla através da criação de um novo bug. Todo bug possui várias características que devem ser informadas. Preferencialmente, deve-se ter conhecimento das várias condições nas quais o bug ocorreu para preencher corretamente os campos de suas características.

4.6 – Versões e Milestones

Todo bug possui características de versão e versão-alvo (*Target Milestone*), que indicam, respectivamente, a versão do produto no qual o bug pertence e ocorreu, e a

futura versão do produto na qual a implementação da solução do bug será feita. *Target Milestone* refere-se à implementação da alteração no corpo ou código do software, para alguma versão. A existência de versões e *milestones* é importante não só para o controle de alterações, mas também para o controle de versões. Essas características possibilitam maior visibilidade do progresso, evolução e transformações do software, permitindo planejamentos e organizações.

Existe a possibilidade de utilizar essas características de Bugzilla para a integração com ferramentas de controle de alterações, como CVS⁶.

4.7 – Grupos

Os usuários cadastrados em um sistema Bugzilla podem opcionalmente pertencer a grupos determinados e criados pelo administrador. Os grupos permitem que o administrador selecione bugs e produtos que podem ser vistos e acessados por outras pessoas. Pode-se criar quantos grupos forem necessários, como forma de organização dos usuários e do trabalho.

Por padrão, quando um novo produto é criado pelo administrador, também é criado um grupo associado a esse produto (essa opção pode ser modificada). Um produto pode ter um ou mais grupos associados e um usuário pode estar associado a um ou mais grupos. O administrador pode criar grupos e realizar essas associações⁷.

É necessário ser membro de todos os grupos aos quais um bug pertence para poder visualizá-lo. Além disso, é necessário ser membro de todos os grupos “entry” de um produto para adicionar bugs a esse produto, e é necessário ser membro de todos os grupos “canedit” de um produto para poder modificar propriedades de bugs desse produto. Os tipos entry e canedit são opções de grupo que só podem ser modificadas e visualizadas pelo administrador por meio do controle de grupos. Através do controle de grupos, o administrador pode escolher a organização dos grupos, produtos e as permissões dos usuários cadastrados.

4.7 – Ciclo de vida dos bugs e funcionamento geral

Para todo bug tenta-se encontrar sua solução, que pode ser o seu conserto, a decisão de que ele não é um bug válido ou relevante, a decisão de tratá-lo mais tarde, entre outras.

Um bug, ao ser adicionado para um produto, recebe um número como identificação única. Cada bug pode ser tratado por uma ou mais pessoas que, para sua solução, podem utilizar-se de recursos oferecidos por Bugzilla. Esses recursos são: comentários, arquivos anexos, relações de dependência, atribuições de responsabilidade e estados de ciclo de vida.

⁶ CVS é sigla para *Concurrent Versions System*, um programa que funciona como sistema de controle de versões de código.

⁷ Essas associações são realizadas através do controle de grupos. O controle de grupos é acessado por meio de opções na barra de ações, disponíveis e visíveis somente ao administrador.

Bugzilla estabelece “estados” de ciclo de vida para cada bug. São eles: *Unconfirmed*, *New*, *Assigned*, *Resolved*, *Verified*, *Closed* e *Reopened*. Em um determinado momento um bug somente pode estar em um estado. O estado inicial de um bug é sempre *Unconfirmed* ou *New*. O ciclo de vida de um determinado bug é o conjunto de todos os estados pelos quais ele passou, em uma possível ordem.

Bugzilla utiliza seu banco de dados MySQL para armazenar todos os dados necessários para implementar suas funcionalidades e recursos, principalmente armazenar os bugs e as contas de usuário.

Consultas e relatórios sobre bugs podem ser realizados. Como resultado de uma consulta é mostrada uma lista de bugs encontrados.

Bugzilla pode estabelecer dependências entre bugs para as alterações que tenham algum tipo de dependência entre si.

Algumas ações e decisões no processo de desenvolvimento de software podem vir a necessitar da opinião de várias pessoas. Para isso há a opção de votação, estabelecida como uma das características do bug. No entanto, a votação não é obrigatória (essa característica do bug pode ser ignorada)⁸. As preferências e propriedades de conta de um usuário cadastrado no sistema podem ser visualizadas e modificadas a qualquer momento.

5 - CARACTERÍSTICAS DE UM BUG

Como forma de organização e para incrementar o tratamento da informação no trabalho com bugs, Bugzilla fornece características para cada bug. Essas características sempre podem ser visualizadas pela interface nas páginas de cada bug, nas páginas de consultas e nas listas de bugs que vêm como resultado de consultas. Essas características são as seguintes:

Produto: O software o qual esse bug pertence e está relacionado.

Componente: O componente do produto, ao qual esse bug pertence ou está relacionado.

Summary: O resumo do bug em uma linha de texto.

Versão: A versão do produto e componente o qual o bug pertence e ocorreu.

Plataforma: A plataforma de hardware utilizado na qual o bug foi encontrado e relatado. Podem ser: PC, Macintosh, Sun, HP, SGI, DEC ou *All* (acontece em todas as plataformas, é um bug multi-plataforma).

⁸ Em Bugzilla, tradicionalmente, uma votação só é realizada para decidir se um bug passará do estado *Unconfirmed* para *New*.

OS: O sistema operacional no qual o bug ocorreu e foi relatado. Por exemplo: Windows 95, Windows 98, Windows 2000, Windows NT, Windows XP, Mac System 8.0, HP-UX, SunOS, Solaris, Linux, *All* (ocorre em todos os sistemas operacionais, isto é, um bug multi-plataforma), ou *Other* (em nenhum dos sistemas operacionais citados).

Prioridade: a prioridade indica a importância do bug e a ordem em que ele deve ser trabalhado e solucionado em relação a outros bugs e a outras atividades. As prioridades vão de menos importante a mais importante. São elas, em ordem crescente: Idle, Low, Plain, Medium e High.

Severidade: a severidade indica o impacto do bug no software e seu desenvolvimento. As severidades são:

Blocker – bloqueia o trabalho de desenvolvimento ou de testes

Critical – provoca panes, perdas de dados e problemas severos de memória

Major – muita(s) perda(s) de funcionalidade(s)

Normal – bug normal

Minor – menor perda de funcionalidade, algum problema ou alteração que necessita de pouco trabalho

Trivial – trivial, simples, problema ou mudança cosmética

Enhancement – solicitação de melhoria

Target Milestone: Futura versão do produto no qual esse bug será implementado, ou seja, depois que uma solução para o bug for dada, em qual versão a efetiva implementação dele ocorrerá⁹.

Reporter: O e-mail da pessoa que criou e relatou o bug e o inseriu no sistema.

Assigned to: O e-mail da pessoa responsável por esse bug. Caso não seja especificado, o responsável é o dono do componente ao qual o bug pertence, por convenção.

Add CC: O(s) e-mail(s) de quaisquer outras pessoas que devem saber prontamente sobre o bug ou estão relacionadas a ele. Quando houver quaisquer mudanças com o bug, as pessoas com estes e-mails são notificadas.

URL: A URL relacionada ao bug, se existir.

Attachment: Anexos ao bug. É mostrada a lista dos arquivos anexos se houver mais de um anexo.

⁹ *Target* e *Target Milestone* são equivalentes e têm o mesmo significado; em algumas páginas de Bugzilla encontra-se *Target* e em outras *Target Milestone*, sem diferença entre ambas.

Bug ## depends on: O número de algum outro bug que este bug depende, e este bug bloqueador deve ser solucionado primeiro para que este tenha sua solução.

Bug ## blocks: O número de algum outro bug que o bug bloqueia, e o bug bloqueador deve receber sua solução primeiro para que o bug bloqueado tenha sua solução.

Votes: #: Número de votos para o bug.

Comments: Comentários sobre o bug. Discussões e informes sobre o bug podem ser adicionados através de comentários. Todos os comentários adicionados ao bug são armazenados, e ordenados por tempo de chegada. Cada comentário tem um número. Ao adicionar um comentário a um bug, ele deve ser composto apenas de texto; não é possível inserir *tags* HTML.

Status: O estado do bug. Esta característica indica o estado atual do bug no ciclo de vida. Podem ser eles: *Unconfirmed*, *New*, *Assigned*, *Reopened*, *Resolved*, *Verified* e *Closed*. Mais explicações sobre esses estados são apresentadas na seção 7.

Resolution: O tipo de resolução dada ao bug. Se uma resolução ainda não foi dada ao bug este campo é deixado em branco. Algum tipo de resolução é dado ao bug somente quando ele não se encontra em nenhum dos estados "abertos"¹⁰. As resoluções podem ser: *fixed*, *invalid*, *wontfix*, *later*, *remind*, *duplicate* e *worksforme*. Mais explicações sobre essas resoluções são apresentadas na seção 7.

Um exemplo da tela da ferramenta Bugzilla que mostra as propriedades de um bug é mostrada na Figura 2.

¹⁰ Estados abertos são aqueles em que o bug não tem uma solução associada, ou seja, se nenhum dos tipos de resolução foi dada ao bug, ele encontra-se em um estado aberto e sua solução é considerada nula.

Alterar a prioridade padrão para plain

(5 of 5) [First](#) / [Previous](#) / [Next](#) / [Last](#)

Product:	BugzillaSAFE	Priority:	plain	Reporter:	magsilva@icmc.usp.br (Marco Aurélio Graciotto Silva)
Component:	General Usability	Severity:	normal	Assigned To:	magsilva@icmc.usp.br (Marco Aurélio Graciotto Silva)
Version:	unspecified	Source Phase:	NotSpecified	Add CC:	
Platform:	Other	Discovery Phase:	NotSpecified	CC:	
OS:	Linux	Target Milestone:	—		
URL:					
Keywords:					

CC:

Attachment	Type	Modified	Status	Actions
Create a New Attachment				View All

Bug 22 depends on: [Show dependency tree](#)

Bug 22 blocks: [Show dependency graph](#)

Votes: 0 [Show votes for this bug](#) [Vote for this bug](#)

Comments:

Figura 2: Bug e suas propriedades

6 - CRIAÇÃO DE UM NOVO BUG

Para criar um novo bug e inseri-lo no sistema Bugzilla para um produto e componente específicos deve-se clicar na opção “New” localizada na barra de ações. Antes de criar um novo bug deve-se ter certeza de que nenhuma outra pessoa já não o relatou antes, ou seja, não deve existir um bug no sistema igual ou muito parecido com o novo. Logo após a inserção de um novo bug no sistema, seus dados serão armazenados no banco de dados de Bugzilla e ele passará a fazer parte do conjunto de bugs e informações armazenadas.

Primeiramente, deve-se escolher o produto para o qual o bug será inserido, através de uma lista dos produtos cadastrados na Bugzilla. Os produtos têm uma descrição de uma ou mais linhas feitas pelo administrador que os inseriu. Após selecionar o produto, deve-se informar as características ou propriedades do bug requeridas por Bugzilla na página de entrada de novo bug. As características são as seguintes: Version, Component, Platform, OS, Priority, Severity, Assigned to, CC, Status, URL, Summary e Description. Essas características são as mesmas explicadas anteriormente na seção 5. Deve-se explicar em quais condições o bug ocorreu e informar suas características. As características devem ser preenchidas corretamente

e de preferência deve-se ter certeza delas, no conteúdo a ser adicionado a cada campo.

Na Figura 3 pode-se visualizar a tela da página de criação (inserção) de bug.

Figura 3: Página de criação de novo bug

O componente e a versão do produto ao qual o bug pertence precisam ser informados. O e-mail do responsável pelo bug deve ser colocado no campo “Assigned to”; se este campo for deixado em branco, por padrão o responsável pelo bug será o dono do componente do bug; em ambos os casos, o responsável, o relator e as pessoas descritas no campo “CC” receberão e-mail sobre o novo bug após este ser inserido, e também receberão e-mail sobre quaisquer mudanças sobre o bug (por exemplo, mudanças de estado e adição de comentários). Se necessário, quaisquer outras pessoas podem ser descritas no campo “CC”.

Caso não haja conhecimento do conteúdo de algum dos campos do bug, deve-se deixar a propriedade em branco ou com o valor padrão. Caso algum dos campos não esteja correto, posteriormente eles podem ser preenchidos. Em geral, o responsável pelo bug preenche ou modifica algum dos campos (características) do bug, especialmente os campos prioridade e severidade. O responsável também

verifica se o bug já não foi descrito por outra pessoa e se algum muito semelhante a ele já existe no sistema.

É preciso atenção especial para preencher o campo “Summary” (Resumo). Esta propriedade é uma das principais de todo bug e aparecerá destacadamente nas listas de bugs que vêm como resultado de buscas. Neste campo deve-se resumir em uma linha de texto o bug (alteração ou problema). Outras pessoas irão ver o bug principalmente pelo seu campo de resumo, e se for necessário ver todos os detalhes do bug, pode-se utilizar a página de visualização do seu conteúdo.

O campo “Description” é o primeiro comentário sobre o bug que será adicionado em sua lista de comentários. Esse comentário é adicionado com número 0. Ele deve conter uma boa descrição do problema ou alteração em questão; é preciso ser detalhista para preencher esse campo e fornecer o maior número possível de informações sobre o bug. É importante indicar as configurações da máquina local na qual o bug ocorreu, como hardware, plataforma e sistema operacional utilizado, e as versões.

O “Status” inicial de um novo bug é sempre *Unconfirmed* ou *New*, dependendo do tipo de política para o produto e das permissões do usuário relator. Se o estado inicial do bug for *Unconfirmed*, ele pode passar por votação para passar para o estado *New*, também dependendo se existir votação para o produto.

7 - CICLO DE VIDA DO BUG

Como forma de organização do trabalho com o controle de alterações, Bugzilla estabelece estados de ciclo de vida para um bug. Durante o trabalho com alterações, ocorrem mudanças na situação em que a alteração se encontra. No momento em que uma alteração é encontrada, ela pode ser relatada e será nova. Se for nova, talvez ela precise ser reproduzida novamente e confirmada, ou aceita. Quando aceita, deve-se trabalhar para encontrar sua solução ou forma de integração, e se isso ocorrer, ela estará solucionada (*Resolved*). Essas e outras situações possíveis que uma alteração passa são organizadas em Bugzilla como os estados do bug.

Na Figura 4 está o grafo¹¹ de ciclo de vida e estados do bug. Neste grafo estão todos os estados e as mudanças de estado possíveis. Os estados de ciclo de vida do bug são:

Unconfirmed: Não confirmado; é geralmente o estado inicial de todo bug, e significa que ninguém ainda verificou a validade do bug. Usuários com permissão podem passar um bug de *Unconfirmed* para *New*. Geralmente o responsável pelo bug ou outra pessoa tenta reproduzir o bug e, se ele realmente ocorrer, é marcado como *New*. O bug nesse estado pode passar por votação para mudar para *New*, e ser um novo bug. O bug nesse estado pode ser diretamente resolvido e marcado *Resolved*.

¹¹ Para todos os estados do grafo existe possibilidade de loop.

New: O bug é novo, pertence à lista de bugs pessoais do responsável pelo bug, e deve ser processado. O responsável, em conjunto com outras pessoas ou não em conjunto com outras pessoas, analisa o bug e decide se o aceita. Se for aceito, o bug deve ser passado para *Assigned*. Se não for aceito, a responsabilidade pelo bug pode ser atribuída a outra pessoa e ele será mantido como *New*. Bugs nesse estado podem ser diretamente solucionados e marcados *Resolved*.

Assigned: Nesse estado, o bug foi aceito por alguma pessoa, que é responsável por ele. Essa pessoa pode resolvê-lo sozinha ou com participação de mais pessoas, e então o bug é passado para *Resolved*. Esse estado é o principal e, geralmente, o bug encontra-se nesse estado quando alguém está trabalhando em uma solução para o bug. Se necessário, a responsabilidade pelo bug pode ser atribuída (*Reassigned*) para outra pessoa, e então ele passa para estado *New*.

Resolved: Uma solução foi dada ao bug, para algum dos tipos de solução. A partir desse estado pode acontecer da solução dada ser incorreta ou conter falhas e o bug precisar ser reaberto (passar para *Reopened*). Se a solução estiver correta, ele pode ser fechado e passar para *Closed*. Se for necessário deste bug ser verificado por alguém ou por um grupo de pessoas antes de ser fechado, isso deve ser feito e ele pode passar para *Verified* ou ser reaberto.

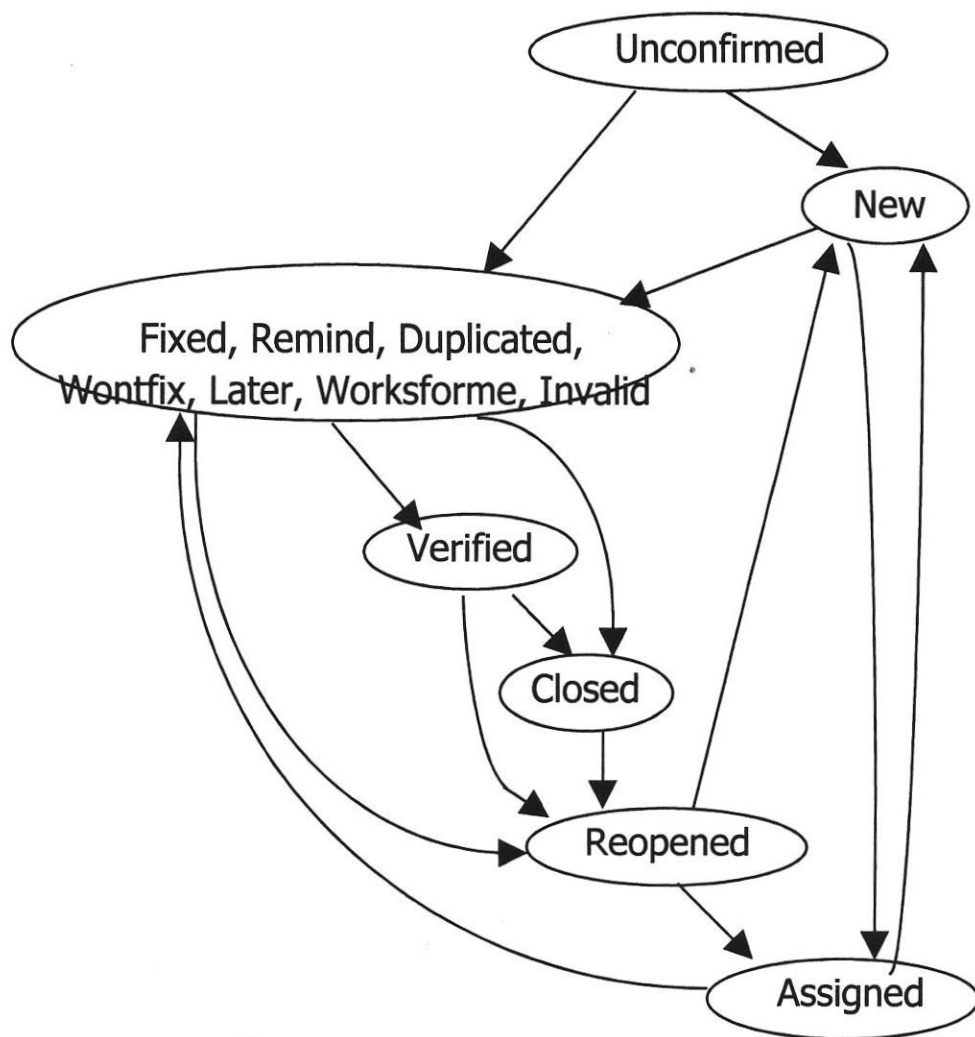


Figura 4: Estados e ciclo de vida do bug [5]

Os tipos de solução que cada bug pode receber são:

- Fixed: o bug foi solucionado (ou consertado).
- Invalid: o problema ou a notificação feita não é um bug (ou não é um bug válido).
- Wontfix: o problema descrito é um bug que nunca será resolvido, solucionado ou implementado.

— **Later:** o problema ou alteração descrita é um bug que não terá solução nesta versão do produto.

— **Remind:** o problema ou alteração descrita é um bug que não terá solução nesta versão do produto, mas pode até ter uma solução.

— **Duplicate:** o bug descrito é uma duplicação de um outro bug existente. É interessante indicar nos comentários do bug o número do outro bug replicado.

— **Workforme:** todas as tentativas de reproduzir o bug foram inúteis, não foram encontraram pistas ou respostas pelas razões e motivos da ocorrência ou presença deste bug, e até possivelmente a leitura de código não mostrou nenhuma verdade sobre ele. Se alguma informação aparecer mais tarde, o bug pode ser reaceito, mas não por enquanto.

Reopened: O bug foi resolvido, mas a solução dada estava incorreta ou continha erros, e o bug foi reaberto. Daqui o bug pode passar para *Assigned* (se alguém o aceitar) ou *Resolved* (se uma solução for diretamente encontrada).

Verified: verificado; se for necessário, uma pessoa ou um grupo de pessoas precisam verificar o bug e a sua solução novamente e confirmá-la antes de ser fechado. Se preciso, ele pode ser reaberto e passar para *Reopened*. Geralmente essa verificação é feita por uma equipe de garantia de qualidade (QA – *Quality Assurance*) ou o dono do componente que o bug pertence.

Closed: bug fechado, sua solução é correta. Se ainda ocorrer da solução ser incorreta ou conter falhas, existe a opção do bug ser reaberto e passar para *Reopened*. A idéia é que isso não aconteça; raramente ocorre na prática nos projetos que usam Bugzilla.

O objetivo é que todo bug tenha sua solução e seja fechado. Portanto, o estado final em seu ciclo de vida é sempre *Closed*.

O ciclo de vida de um bug é o conjunto de todos os estados pelos quais ele passou, e sua ordem, até chegar em sua solução definitiva. O histórico da atividade de cada bug pode ser visualizado em Bugzilla (veja seção 8).

Se uma nova versão do produto for lançada, é conveniente passar todos os bugs no estado *Resolved* para *Closed*, para fechar todos os bugs e ter o lançamento do produto com todos os bugs já solucionados e fechados no sistema.

8 - VISUALIZAÇÃO DE UM BUG

Durante a navegação pelas páginas e formulários de Bugzilla, principalmente as listas de bugs que vêm como resultado de buscas, pode-se ter acesso ao conteúdo de um bug através de seu link, principalmente, o seu número de bug. Ao acessar o conteúdo do bug (na interface), pode-se ver uma página com todas as propriedades do bug preenchidas.

É possível imprimir a página do bug pela opção "Format for printing" no inferior da página, e ver a atividade do bug (ciclo de vida) na opção "View bug activity" no inferior da página. O ciclo de vida mostra todos os estados que o bug passou desde sua criação, em ordem de tempo.

Quaisquer modificações nas propriedades do bug são feitas através da página de seu conteúdo, inclusive a adição de comentários, adição de arquivos anexos, e as mudanças de estado. Pode-se reatribuir a responsabilidade pelo bug a outra pessoa, através do campo "Reassign bug to:", preenchendo-o com o e-mail da pessoa.

O número de votos que o bug possui pode ser visto nessa página, e pode-se votar no bug através do link "Vote for this bug".

O título da página de visualização do bug é o seu resumo (*Summary*).

9 - DEPENDÊNCIAS ENTRE BUGS

Bugzilla pode estabelecer relações de dependência entre os bugs de um produto. Relações entre bugs ocorrem naturalmente durante o desenvolvimento, projeto, testes ou manutenção de software. Essas relações podem ser desde nulas ou pequenas até relações grandes e de dependência.

Muitas alterações (devido a erros, melhorias ou novas funcionalidades a serem implementadas) dependem que alguma outra alteração (bug) cadastrada seja feita primeiro para que possam ser realizadas. Algumas alterações não necessitam de que outras sejam realizadas, ou tenham solução, para que sejam feitas. Porém, para as alterações que têm forte relação entre si (dependência), os desenvolvedores precisam ter visibilidade destas relações, como forma de organização de seu trabalho e para ter controle efetivo das alterações.

Se ocorrer o caso de relação de dependência entre bugs, ela pode ser indicada em duas das características do bug: "depends on" e "blocks", como mostradas no exemplo a seguir:

Bug 15 depends on: 14

Bug 15 blocks: 16

Neste exemplo, o bug 15 depende que o bug 14 seja resolvido primeiro antes que possa ser solucionado ou realizado; e o bug 15 deve ser resolvido primeiro antes que o bug 16 seja resolvido.

Para o conjunto de todas as dependências entre bugs de um componente ou produto, abstratamente existe uma árvore de dependências ou um grafo de dependências. Bugzilla mostra tanto o grafo quanto a árvore de dependências entre bugs, através de um dos links nas páginas de bugs em sua interface.

A Figura 5 mostra um exemplo da página de uma árvore de dependências para um bug de número 97706. Esse bug depende de alguns bugs e nenhum bug depende dele.

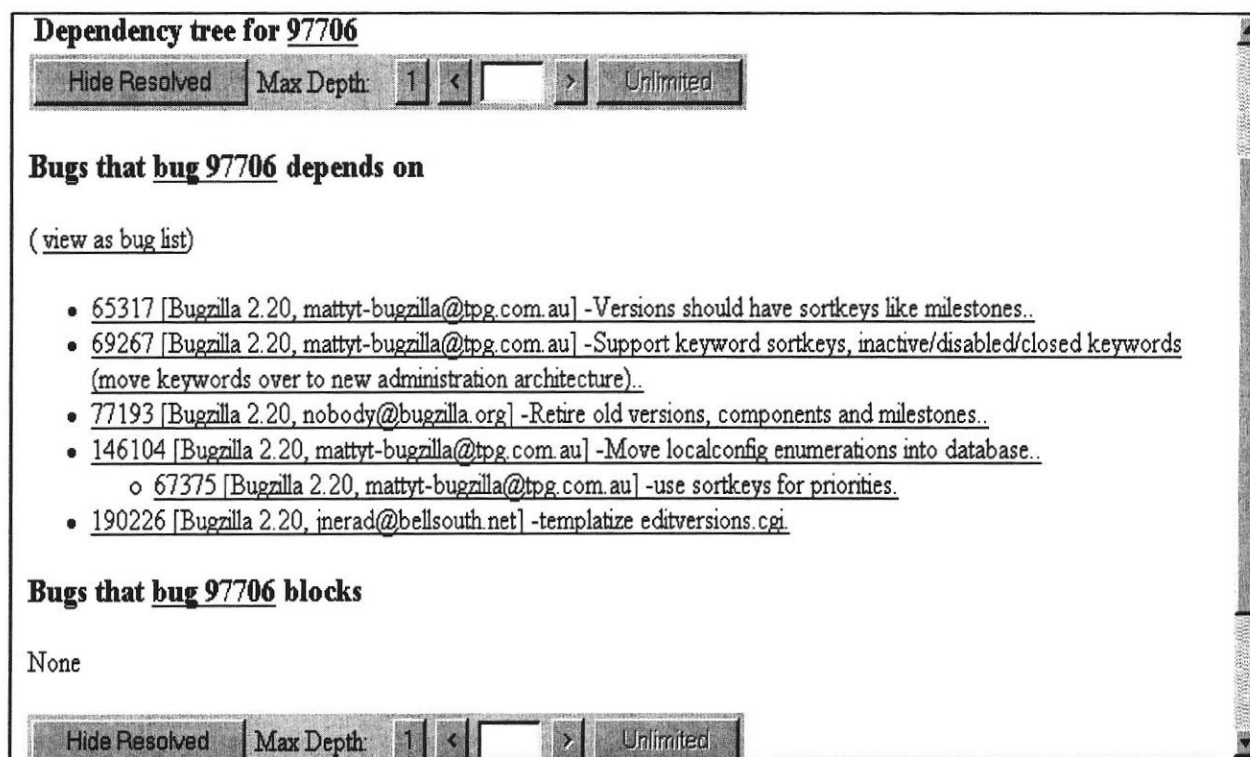


Figura 5: Uma árvore de dependências entre bugs

10 - VOTAÇÃO

Todo bug tem a característica do número de votos que possui. O número de votos inicial de todo bug é zero. A característica pode ser usada opcionalmente para qualquer fim desejado pelos usuários. Se alguma mudança de estado do bug gerar conflito entre usuários, o problema pode ser resolvido realizando-se uma votação. Tradicionalmente, mas não obrigatoriamente, gera-se uma votação para decidir se um

bug deve passar do estado *Unconfirmed* para *New*, ou seja, se um novo bug relatado deve ser aceito no sistema.

Existe número máximo de votos que um usuário pode colocar em um único bug, número máximo de votos geral que um usuário pode colocar em bugs que ela tem acesso, e número de votos que um bug, pertencente a um produto específico, deve receber para automaticamente passar do estado de *Unconfirmed* para *New*. Todas essas configurações só podem ser feitas pelo administrador do sistema.

11 - CONFLITOS DE ACESSO SIMULTÂNEO

Durante a utilização de um sistema Bugzilla, dois ou mais usuários podem tentar modificar um bug ao mesmo tempo. Se um *commit* para modificação de dados e características de um bug for realizado ao mesmo tempo em que outro usuário também o esteja fazendo, ou esteja “no meio do caminho”¹² para realizá-lo, ocorre uma colisão no sistema. Apenas um dos usuários tem sua ação realizada, e para o outro usuário é enviado um aviso.

12- CONSULTAS

12.1 – Consultas

Bugzilla permite ampla possibilidade de obtenção de consultas ao conteúdo do banco de dados interno sobre bugs. Todos os bugs de um produto e/ou componente são analisados para comparação com um tipo solicitado.

Para fazer uma consulta, clica-se no link “Query” na barra de ações.

Uma consulta pode ser realizada através dos seguintes atributos de bug: resumo, comentário, URL, e-mail ou número do bug, além do produto, componente, versão e *milestone*. Palavras e frases completas podem ser digitadas ou somente parte delas. Aqueles bugs que têm o conteúdo (ou parte dele) serão selecionados como resultado da busca.

Pode-se filtrar as buscas (ser mais específico) selecionando propriedades ou modificações do bug. As propriedades são status, resolução, severidade, fase do código, fase de descobrimento, prioridade, plataforma e sistema operacional. As modificações do bug são tempos ou intervalos de tempo em que alguma de suas propriedades mudou, ou as seguintes opções de inclusão/exclusão:

— *Only/Exclude bugs numbered: ##*: mostrar somente bugs com determinados números, ou excluir exatamente esses bugs.

¹² Entre o commit e a abertura da página de edição de bug.

— *Changed in the last ## days*: alterados nos últimos dias, para o número escolhido de dias.

— *Containing at least ## votes*: bugs que contém o número mínimo de votos especificado.

É possível fazer buscas avançadas utilizando operadores lógicos. Essas buscas estão localizadas no rodapé do formulário de busca.

Na Figura 6 visualiza-se a página de consulta por bugs.

SAFE Bug Tracking System Bugzilla/ 2.16.5

Search for bugs

Summary: contains all of the words/strings [Search]

Product: BugzillaSAFE, CoTeia, Papagaio, TestProduct

Component: ChatServer integration, Design Rationale, eClass integration, General Usability, GroupNote integration

Version: 1.0.0, 1.1.0, 1.1.0beta1, 1.1.0beta2, 1.1.0beta5

Target: 1.1.0, 1.2.0, 2.0.0

A comment: contains all of the words/strings

The URL: contains all of the words/strings

Keywords: contains all of the keywords

Status:	Resolution:	Severity:	Source Phase:	Discovery Phase:	Priority:	Hardware:	OS:
UNCONFIRMED	FIXED	blocker	NotSpecified	NotSpecified	high	All	All
NEW	INVALID	critical	NotApplicable	NotApplicable	medium	Macintosh	Other
ASSIGNED	WONTFIX	major	Requirements	Requirements	plain	PC	FreeBSD
REQUIRED	LATER	normal	Design	Design	low	Sun	Linux
RESOLVED	REMIN	minor	Coding	Coding	idle	Other	MacOS 8.x
VERIFIED	DUPLICATE	trivial	Testing	Testing			MacOS 9.x
CLOSED	WORKSFORME	enhancement	Maintenance	Maintenance			MacOS X

Bug Changes

Email and Numbering: Any of [bug owner, bug reporter] Only bugs changed in the last [] days

Figura 6: Página de consulta por bugs

12.2 – Lista de bugs

Como resultado de qualquer busca, Bugzilla mostra em um novo formulário a lista de bugs que combinam com o tipo de busca especificado. Nesta lista, cada linha corresponde a um bug encontrado. As colunas da linha são as propriedades do bug. Pode-se modificar essas colunas, inserindo ou retirando algumas propriedades; para isto deve-se clicar no link “Change Columns” no inferior da lista.

É possível ordenar o resultado de alguma forma escolhida. Para isso deve-se escolher o tipo de ordenação ainda no formulário de busca, no rodapé da página, em “Sort by”.

Na Figura 7 há um exemplo de lista de bugs.

12.3 – Buscas avançadas

As buscas avançadas são da seguinte forma: existe um “termo” formado por dois menus e um campo de texto. O primeiro menu indica onde procurar pelo que se quer buscar, o segundo menu indica como determinar o que vem como resultado, e o campo de texto é o conteúdo de busca. Pode-se clicar em um dos botões “And” ou “Or”, e fazendo-se isso, nos dois casos, mais um “termo” é colocado embaixo do anterior. Esse novo termo pode ser preenchido livremente. A partir daqui se a busca for realizada, todos os resultados que combinam com ambos os “termos” são mostrados, com “ou” ou com “e”. Por exemplo, se a opção foi “ou”, serão mostrados os resultados obtidos por qualquer um dos termos.

Pode-se clicar em “And” e “Or” novamente quantas vezes quiser, e sempre novos “termos” serão adicionados. Pode-se ter uma página com muitos “termos”. O resultado da busca será sempre o que bater com todos os termos, pelas combinações de “e” e “ou”. O “ou” tem maior precedência do que o “e”.

SAFE Bug Tracking System					Bugzilla/		2.16.5
Bug List							Bugs found: 76
ID	Sev	Pri	Plt	Owner	State	Result	Summary
2	nor	pla	PC	magsilva@icmc.usp.br	NEW		License issues about MPL
5	nor	pla	PC	magsilva@icmc.usp.br	ASSI		Wrong regular expression on groups definition brake new a...
7	nor	pla	Oth	magsilva@icmc.usp.br	RESO	FIXE	Criação de bugs em produtos sem componentes
8	nor	pla	Oth	magsilva@icmc.usp.br	NEW		Erro quando criando um bug
16	nor	hig	Oth	magsilva@icmc.usp.br	RESO	FIXE	Corrigir palavra
19	enh	med	Oth	magsilva@icmc.usp.br	NEW		Agilizar a criação de vários bugs
22	nor	pla	Oth	magsilva@icmc.usp.br	NEW		Alterar a prioridade padrão para plain
24	nor	idl	All	magsilva@icmc.usp.br	RESO	FIXE	Teste
25	nor	idl	All	magsilva@icmc.usp.br	RESO	FIXE	Teste
35	nor	pla	PC	magsilva@icmc.usp.br	RESO	FIXE	Cannot access a login wiki
36	nor	pla	PC	magsilva@icmc.usp.br	RESO	FIXE	Cannot create an annotation
37	nor	pla	Oth	magsilva@icmc.usp.br	RESO	FIXE	Login type swiki main page always open via create.php
38	nor	pla	Oth	magsilva@icmc.usp.br	RESO	FIXE	Cannot access restricted (login) swikis
39	nor	pla	Oth	magsilva@icmc.usp.br	RESO	WORK	Review the & and & usage
40	nor	pla	PC	magsilva@icmc.usp.br	RESO	FIXE	Error when accessing a login swiki after expired session
41	nor	pla	PC	magsilva@icmc.usp.br	RESO	FIXE	Doesn't confirm new password when editing page.
42	nor	pla	PC	magsilva@icmc.usp.br	RESO	FIXE	Page under edition haven't got its password checkbox chec...
43	nor	pla	PC	magsilva@icmc.usp.br	RESO	FIXE	Editing some pages, the author field is filled with the s...
44	nor	pla	PC	magsilva@icmc.usp.br	RESO	FIXE	Change onBlur to onChange
45	enh	pla	PC	magsilva@icmc.usp.br	RESO	FIXE	Change Top link to reference the main wiki page (index.ph...
46	enh	pla	PC	magsilva@icmc.usp.br	NEW		Wiki page protection heritage
47	enh	pla	PC	magsilva@icmc.usp.br	RESO	FLXE	Avoid a new upload page creation
48	enh	pla	PC	magsilva@icmc.usp.br	RESO	FIXE	Send a warning when uploading a file too big
49	nor	pla	PC	magsilva@icmc.usp.br	RESO	FIXE	Doesn't accept & as wikipedia content

Figura 7: Lista de bugs – resultado de consulta

12.4 – Salvar consultas

É possível salvar uma consulta e um link para ela será automaticamente criado na barra de ações no rodapé da página. Isso é útil para acessar diretamente algum tipo de busca que se faz várias vezes, não precisando preencher um formulário para ela a cada vez. Para fazer isso, deve-se submeter a consulta e selecionar a opção “Remember this query and name it:” no inferior do formulário de busca. Coloca-se um nome para a busca, que será o texto do link adicionado na barra de ações.

Se a opção “Remember this as my default query” for escolhida, este exato formulário de busca será o que irá aparecer sempre que este usuário logar.

Caso se queira apagar uma busca salva (e o seu link), basta abrir seu formulário de busca, e rodar a busca com a opção “Forget the remembered query”.

12.5 – Exportação de listas de bugs

Listas de bugs podem ser exportadas como HTML (o padrão), CSV ou RDF. O formato CSV pode ser importado por alguns aplicativos, entre eles MS Excel.

12.6 – Quicksearch

A ferramenta Quicksearch de Bugzilla permite que buscas rápidas sejam feitas, sem necessidade de abrir e preencher o formulário de consulta (query).

O Quicksearch se encontra na página principal de Bugzilla (home). Deve-se preencher o campo de busca e clicar em “Show” para realizá-lo. Bugs serão procurados em todos os produtos registrados, a não ser que algum produto seja especificado. A busca é realizada nos campos “Summary”, “Keywords”, “Product”, “Component” e “Status Whiteboard”.

É possível inserir qualquer palavra, frase ou parte de uma sequência de caracteres para fazer busca. A busca não é sensível a maiúsculas e minúsculas (*case insensitive*). Pode-se usar os operadores lógicos “and”, “or” e “not”. Para o “not” se utiliza “-”. Um espaço entre palavras indica “and”, e a barra vertical indica “or”. Por exemplo:

scroll -box = scroll **not** box (busca por “scroll” e exclui resultados com “box”)

página|texto = página **ou** texto (busca pelas palavras “página” ou “texto”)

tecla pressionada = tecla **e** pressionada (busca pelas palavras “tecla” e “pressionada”)

“or” tem maior precedência do que “and”.

“+” procura pela palavra que vem após ele como substring de outras palavras.

“#” procura por exatamente a palavra que vem após ele. Ambos na propriedade “resumo” (*Summary*) dos bugs. Exemplo:

+clica = procura por palavras com parte de sequência de caracteres clica, como “clcando” ou “clcado”

#clica = procura por palavras “clica”

Por padrão, somente bugs em estados “abertos” são procurados. Se inserir +DUP no início da busca, bugs duplicados (Resolved Duplicate) são incluídos na busca, FIXED para incluir bugs consertados (no estado Resolved Fixed) e ALL para incluir todos os bugs, independentemente de seu estado. Exemplo:

+DUP mouse (procura por bugs duplicados com “mouse”)

ALL mouse (procura por “mouse” em todos os bugs)

Para procurar somente em um produto ou componente específico, deve-se adicionar “:nome” ao campo de busca, onde “nome” indica o nome do produto ou componente, ou parte dele (uma substring). Por exemplo:

mouse :conversor (busca pela palavra “mouse” somente em bugs do produto “conversor”) altura :dfluidos (busca pela palavra “altura” somente em bugs do produto “dfluidos”)

13 - RELATÓRIOS

Bugzilla emite relatórios e gráficos com estatísticas sobre os bugs armazenados no banco de dados do sistema local. Várias informações gerais sobre os bugs e os usuários podem ser mostradas, e com opção de escolha dos tipos de bugs para análise (por estado).

Para ter acesso aos relatórios, deve-se clicar no link “Reports” na barra de ações. Deve-se escolher um produto (ou todos registrados), o tipo de relatório (pode ser gráfico, ou “charts”), e os tipos de bugs que entrarão no relatório, pelo estado e/ou tipo de solução. No caso de gráficos, são mostrados contagens de bugs pelo tempo, para os tipos de bugs escolhidos. Na Figura 8 é mostrado um exemplo de um relatório da Bugzilla.

Bug Report for -All-

Sat Aug 7 21:54:10 2004

Summary

New Bugs This Week	1
Bugs Marked New	46
Bugs Marked Assigned	4
Bugs Marked Reopened	2
Total Bugs	52

Bug Count by Engineer

Owner	New	Assigned	Reopened	Total
leonardo@grad.icmc.usp.br	0	2	1	3
magsilva@icmc.usp.br	45	2	1	48
michetti@grad.icmc.usp.br	1	0	0	1

Figura 8: Um relatório Bugzilla**14 - PREFERÊNCIAS DE USUÁRIO**

As preferências de usuário podem ser editadas e modificadas, através do link “Edit prefs” na barra de ações (somente depois do login efetuado).

Existem opções para “Account Settings” (propriedades de conta), “E-mail Settings” (propriedades de e-mail), “Page Footer” (opções da barra de ações) e “Permissions” (permissões).

Em “Account Settings” pode-se modificar o password (senha) e o nome real.

Em “E-mail Settings” pode-se modificar algumas opções de quando receber ou não e-mail em alguns casos.

Em “Page Footer” escolhem-se as opções da barra de ações, principalmente se alguns links devem aparecer.

Em “Permissions” pode-se ver as permissões pessoais do usuário. Nada pode ser modificado e é uma página apenas informativa. A Figura 9 mostra a página da edição e visualização de preferências de usuário.

Account settings	Email settings	Page footer	Permissions
----------------------------------	--------------------------------	-----------------------------	-----------------------------

Account settings

Please enter your existing password to confirm account changes.

Password:

New password:

Re-enter new password:

Your real name (optional, but encouraged):

Actions: [Home](#) | [New](#) | [Query](#) | [Find](#) bug # | [Reports](#) | [My Votes](#) [Edit prefs](#) | [Log out leonardo@grad.icmc.usp.br](#)

Preset Queries: [My Bugs](#)

Figura 9: Página das preferências de usuário de Bugzilla

15 – Referências Bibliográficas

- [1] – The Bugzilla Team, “The Bugzilla Guide”, <http://www.bugzilla.org/docs/>, 2.19 Development Release, 2004.
- [2] – Fortes, R. P. M., Reis, C. R., “An Overview of the Software Engineering Process and Tools in the Mozilla Project”. In: Workshop on Open Source Software Development, Proceedings. Newcastle UK, February 2002, pp.162-182.
- [3] – Reis, C. R., “Uma visão geral do Bugzilla, uma ferramenta de acompanhamento de alterações”. Sessão de Ferramentas do SBES 2002. Gramado RS. 14-18 de outubro de 2002.
- [4] – Bober, Brian., Hernandez, Tara., “Help Using The Bugzilla Query Form”, 2001. Disponível em “<http://puck.nether.net/bugzilla-2.18rc3/queryhelp.cgi>”
- [5] – Silva, M. A. G. - “Estados do Ciclo de Vida do BUG em Bugzilla. 2004. Apresentação do Bugzilla no Projeto SAFE.

Apêndice A. Manual de Instalação

1. Introdução

Para o Bugzilla ser instalado e funcionar de forma correta, vários programas e módulos dos quais ele depende precisam estar instalados e configurados. Esse manual se concentra em descrever o processo de instalação padrão, ou seja, usando softwares onde é garantido que o Bugzilla funcionará. Para obter mais detalhes sobre o processo de instalação utilizando outros softwares ou sistemas operacionais, visite a documentação online em www.bugzilla.org/docs.

2. Programas necessários

Esse manual cobre a instalação do Bugzilla utilizando os seguintes softwares:

- Web Server: Apache (1.3 ou superior)
- Bando de Dados: MySQL (3.23.41 ou superior)
- Perl (5.6.0 ou superior)
- Sistema Operacional: Linux.

Como o processo de instalação desses programas não serão discutidos aqui, visite o site oficial para obter mais informações.

3. Resumo

Primeiramente devemos ter os programas citados instalados e funcionando.

Note que muitas das etapas precisam ser feitas por alguém com acesso a conta root (ou sudo), por isso, se você não tiver acesso, peça que o administrador do seu sistema faça essas etapas por você.

O Bugzilla se utiliza de um script chamado `checksetup.pl` para criar a base de dados no MySQL, mudar as permissões dos arquivos para que possam ser acessados pelo Web Server e testar as configurações dos softwares os quais o Bugzilla depende. Ele lê um arquivo chamado `localconfig` (criado na primeira vez que o script é utilizado) para obter as informações sobre a base de dados, usuário e senha que serão utilizados no MySQL bem como o grupo do servidor Apache.

Esse script será de grande ajuda na instalação, já que ele nos diz quais programas, módulos ou configurações estão errados. Portanto, sempre que precisar verificar alguma coisa, digite `./checksetup.pl` no diretório onde o Bugzilla for copiado.

A instalação se divide nas seguintes etapas:

- Instalar, se necessário, Apache, MySQL e Perl no Linux.
- Obter e copiar o Bugzilla para um diretório o qual o Apache tenha acesso.
- Executar o script `"checksetup.pl"` para que o arquivo `"localconfig"` seja criado.
- Executar novamente o script para verificar quais módulos Perl estão faltando e instalá-los. Vale

lembrar que o script pode (deve) ser usado várias vezes.

- Configurar o Apache.
- Configurar o MySQL.
- Preencher o arquivo "localconfig" com os mesmos valores utilizados no Apache e no MySQL.
- Executar novamente o script. Se tudo estiver certo, a base de dados será criada, as permissões dos arquivos serão alteradas, os templates serão compilados e o usuário administrador do Bugzilla será criado.

Cada um desses passos será discutido em detalhes a seguir.

4. Obtendo o Bugzilla

Copie a última versão estável do Bugzilla em www.bugzilla.org/download/.

Por exemplo, se o nome do arquivo copiado for "bugzilla-2.18rc3.tar.gz", você deve descompactá-lo usando o comando:

```
tar -zxvf bugzilla-2.18rc3.tar.gz
```

Copie o diretório para um local onde o Apache tenha acesso.

Nota

Em um sistema Debian por exemplo, você pode usar o diretório "/var/www/" e acessar o Bugzilla digitando no seu browser "localhost/bugzilla" (se "bugzilla" for o nome que você deu ao diretório) ou colocar o Bugzilla dentro da área de usuário destinada a páginas web (geralmente "public_html" ou "www" - verifique "UserDir" nas configurações do Apache) e acessá-lo digitando "localhost/<user>/bugzilla".

Mude o nome do diretório se quiser, lembre-se que esse nome será usado para acessar o Bugzilla, logo, um diretório chamada "bugzilla" é mais fácil de guardar do que "bugzilla-2.18rc3".

5. Instalando os módulos Perl

Com o Bugzilla copiado, execute o script "checksetup.pl" para criar o arquivo "localconfig". Esse arquivo será usado posteriormente. Execute novamente o script e veja a lista dos módulos que estão faltando.

O próprio escript dá dicas de como instalar esses módulos.

Esses módulos podem ser instalados de diferentes maneiras (utilizando pacotes, CPAN ou compilando os fontes), mas geralmente precisamos combinar esses métodos para que todos os módulos sejam instalados com sucesso.

Você pode começar tentando utilizar o CPAN que é um recurso disponível com o Perl para buscar módulos em repositórios na internet.

A sintaxe básica para utilizar o CPAN é a seguinte:

```
perl -MCPAN -e 'install "<nomedomodulo>"'
```

Por exemplo, para instalar o módulo "Data::Dumper" o comando seria:

Exemplo 5.1 - Instalando módulos Perl usando CPAN

```
perl -MCPAN -e 'install "Data::Dumper"'
```

Obs: Note que temos o comando "Data::Dumper" entre aspas duplas e 'install "Data::Dumper"' entre aspas simples.

A cada módulo instalado, execute novamente o "checksetup.pl" para ver se ele foi instalado corretamente. Se a instalação não for bem sucedida, tente instalar os módulos usando algum pacote (.deb ou .rpm por exemplo) ou compilando os fontes. Para obter os pacotes ou os fontes basta digitar o nome dos pacotes em algum sistema de busca (google por exemplo, ou www.rpmfind.net para pacotes .rpm).

No Debian, faça uma busca com "apt-get search" pelo nome do módulo. Talvez o nome do pacote não seja exatamente o nome do módulo, por isso, leia as informações para ter certeza.

Repita esse processo até que todos os módulos obrigatórios estejam instalados. Note que para utilizar todas as funções do Bugzilla os módulos opcionais devem ser instalados, mas o Bugzilla funcionará se os obrigatórios estiverem instalados.

6. Configurando o Apache

O Apache guarda suas configurações principais em um arquivo texto geralmente chamado "httpd.conf".

Abra esse arquivo e descomente ou inclua a seguinte linha se necessário:

AddHandler cgi-script .cgi

Isso fará com que o Apache execute ".cgi" fora do diretório "cgi-bin".

Agora inclua as seguintes linhas a uma diretiva "<Directory>" que se aplique ao diretório onde o Bugzilla foi copiado ou a algum dos seus diretórios pais.

Options +ExecCGI +FollowSymLinks

AllowOverride Limit

Isso permite ao arquivo ".htaccess" sobrescrever as permissões globais e permite executar ".cgi" nesse diretório.

Por exemplo, se o diretório onde o Bugzilla for instalado for "/var/www/bugzilla" as seguintes linhas deveriam ser incluídas:

Exemplo 6.1 - Configurando a diretiva "<Directory>" no Apache

```
<Directory "/var/www/bugzilla">
```

```
Options +ExecCGI +FollowSymLinks
```

```
AllowOverride Limit
```

```
</Directory>
```

Agora procure pela diretiva "DirectoryIndex" e inclua "index.cgi".

Aproveite para procurar pela diretiva "Group" que define o grupo utilizado pelo Apache. Anote esse valor pois ele será usado para configurar o arquivo "localconfig" posteriormente.

7. Configurando o MySQL

É preciso criar uma base de dados, um usuário e definir sua senha para que o Bugzilla possa ter acesso a esse banco de dados. Os mesmos valores que forem definidos aqui serão usados para preencher o arquivo "localconfig" posteriormente.

Na verdade, tudo isso pode ser feito utilizando apenas dois comandos:

```
GRANT SELECT, INSERT, UPDATE, DELETE, INDEX, ALTER, CREATE, LOCK  
TABLES, CREATE TEMPORARY TABLES, DROP, REFERENCES ON <nome_da_base>.*  
TO <nome_do_usuario>@localhost IDENTIFIED BY '<senha_do_usuario>';
```

FLUSH PRIVILEGES;

Note que "usuário" nesse caso é um novo usuário do MySQL.

Por exemplo, se o nome da base for "bugs" e o usuário do MySQL que vai acessar essa base se chame "joao" e tenha senha "123", o comando "grant" ficaria assim:

Exemplo 7.1 - Configurando o MySQL

```
GRANT SELECT, INSERT, UPDATE, DELETE, INDEX, ALTER, CREATE, LOCK  
TABLES, CREATE TEMPORARY TABLES, DROP, REFERENCES ON bugs.* TO joao@localhost  
IDENTIFIED BY '123';
```

8. Preenchendo o arquivo localconfig

O script "checksetup.pl" tem como função criar as tabelas no banco de dados, e mudar as permissões dos arquivos. Para isso, ele lê as informações contidas em um arquivo chamado "localconfig" que foi criado na primeira vez que o script foi executado. Esse arquivo também está localizado no diretório onde o Bugzilla foi instalado.

Abra o arquivo e edite as seguintes diretivas:

\$webservergroup = "<grupo_do_apache>"

Substitua "<grupo_do_apache>" pelo grupo definido na diretiva "Group" no arquivo de configuração do Apache.

\$db_name = "<nome_da_base>"

Substitua "<nome_da_base>" pelo nome dado a base de dados quando configurando o MySQL. No exemplo 7.1, o nome da base era "bugs".

\$db_user = "<nome_do_usuario>"

Substitua "<nome_do_usuario>" pelo nome do usuário criado quando configurando o MySQL. No exemplo 7.1, o nome do usuário era "joao".

\$db_pass = "<senha_do_usuario>"

Substitua "<senha_do_usuario>" pelo senha dada ao novo usuário quando configurando o MySQL. No exemplo 7.1, a senha do usuário era "123".

9. Finalizando a Instalação

Agora que tudo está configurado basta rodar novamente o script "checksetup.pl". Se tudo estiver certo, ele fará as verificações normais nos módulos, depois criará as tabelas no bando de dados, compilará os templates usados pelo Bugzilla e em seguida pedirá algumas informações sobre o administrador do Bugzilla.

Essas informações podem ser alteradas posteriormente, mas elas devem ser preenchidas com cuidado pois esse será o primeiro usuário do Bugzilla.

Se a instalação for feita na área de um usuário (em public_html por exemplo) e quando você tenta acessar o Bugzilla pelo browser obter um erro de permissão (Forbidden), tente incluir esse usuário no grupo do Apache e rodar checksetup.pl novamente.

10. Testando

Se tudo estiver certo, basta você digitar no browser o endereço onde o Bugzilla foi instalado. Por exemplo, se o Bugzilla foi instalado em "/var/www/bugzilla" você pode acessá-lo digitando "localhost/bugzilla".

Agora basta você fazer log-in com o usuário administrador criado durante a execução do script "checksetup.pl".