

Insektenforscher

Bugs sind die natürlichen Feinde der Programmierer und wollen gut erforscht sein. In größeren Projekten tauchen sie so oft auf, dass man dafür professionelle Werkzeuge braucht. Bugzilla ist ein solches Tool, das sich in freien und kommerziellen Projekten bewährt. Roger Butenuth



Willi Ruh / www.vistula.com

Software wird trotz der Erfindung von Computern immer noch von Menschen geschrieben. Viele Werkzeuge nehmen inzwischen den gebeutelten Programmierern zwar große Teile der stumpfsinnigen und fehlerträchtigen Arbeit ab, so dass sie sich mehr auf den kreativen Teil der Programmierung konzentrieren können. Die Kreativität geht dabei manchmal jedoch über das gewünschte Ziel hinaus oder versandet vorher, sodass am Ende das Programm nicht mehr genau das macht, was es gemäß Spezifikation machen sollte.

Der gemeine Nutzer bezeichnet dies Verhalten gerne als Fehler. Auf der anderen Seite sind die Benutzer selten mit dem glücklich, was sie einmal selbst spezifiziert haben oder was man ihnen vor die Füße geworfen hat, ohne sie zu fragen. So kommen neben den Fehlern auch noch Änderungswünsche (neudeutsch: Change Request oder CR) dazu.

Bei großen Projekten reichen gutes Gedächtnis, eine Wandtafel oder eine Tabellenkalkulation zur Verfolgung von

Fehlern und CRs nicht aus. Zum Beispiel stehen in der Fehlerdatenbank für Mozilla inzwischen mehr als 150 000 Einträge (die meisten davon allerdings im Erledigt-Status). Ein Bug-Tracking-System wie Mozilla ist dazu da, auch hier Erleichterung zu schaffen und das Melden und Bearbeiten von Fehlern zu automatisieren.

Der lange Weg eines Fehlers

Naturgemäß ist eine solche Software komplex und reich an Features, da sie zahlreiche unterschiedliche Anwendungsfälle abdecken muss. Im Folgenden soll es also nur darum gehen, die wichtigsten Eigenschaften zu zeigen, ansonsten wäre hier mindestens die ganze Anleitung nachzudrucken.

Ein Anwender, der einen neuen Fehler entdeckt hat, möchte diesen möglichst einfach und ohne viel Tipparbeit eingeben können. Das geschieht in zwei Schritten: Im ersten wird das Produkt ausgewählt, in dem der Fehler entdeckt

wurde. Falls die Bugzilla-Instanz nur Fehler zu einem bestimmten Produkt verwaltet, entfällt dieser Schritt natürlich. Im zweiten Schritt wird der Fehler genauer beschrieben:

- Wo ist er aufgetreten (Plattform, Betriebssystem, Komponente, Version)?
- Wie schwerwiegend wird er eingeschätzt? (Kategorien von Show-Stopper bis Änderungswunsch)
- Mit welcher Priorität soll er behoben werden?
- Wem soll er zugeordnet werden? (Kann man unbeantwortet lassen)
- Eine kurze Beschreibung (für Übersichten) und eine ausführliche Beschreibung des Fehlers.

Das gesamte Formular ist in **Abbildung 1** zu sehen. Einige Features von Bugzilla sind auf dem Formular jedoch nicht zu erkennen: In der Beschreibung kann man sich auf andere Fehler beziehen (durch den Text »bug«, gefolgt von der Nummer des Fehlers) oder auch URLs eingeben, die später automatisch in Links umgewandelt werden. An einmal erfasste Fehler lassen sich beliebige Attachments hängen, beispielsweise Testdaten für die Reproduktion des Fehlers oder auch ein Patch, mit dem sich der Fehler beheben lässt.

Bei Bugzilla in der Standardkonfiguration erhalten neue Fehler den Zustand »new«. Bei einem öffentlich zugänglichen Bugzilla ist es angebracht, eingehende Fehlermeldungen zu prüfen. Dazu lässt sich noch vor »new« der Zustand »unconfirmed« für frisch hereinkommene Meldungen einbringen. Der Übergang zwischen den beiden Zuständen erfolgt auf zwei Wegen. Entweder durch einen Nutzer aus der Gruppe »Canconfirm« oder durch eine demokra-

tische Wahl vieler Nutzer, die diesen Fehler gerne beseitigt haben wollen. Die zweite Variante kann der Admin des Systems abschalten. Sie eignet sich eher für Erweiterungswünsche als für Fehler.

Fehlerzuordnung

Bereits im Zustand »new« ist jeder Fehler einem Entwickler zugeordnet, standardmäßig dem Komponenten-Verantwortlichen. Diese Zuordnung lässt sich beliebig oft ändern. Den nächsten Zustand – »assigned« – kann man je nach Arbeitsweise im Projekt unterschiedlich nutzen: Die erste Variante ist eine Planung von außen, bei der ein Projekt- oder Teamleiter den Fehler auf »assigned« setzt und damit einem Entwickler den Auftrag zum Beheben erteilt. Die zweite Variante entspricht eher der Arbeitsweise in Open-Source-Projekten: Der Entwickler selbst setzt den Zustand »assigned« und markiert damit, dass er mit der Arbeit beginnt. Diese Variante entspricht auch eher der Philosophie von Bugzilla, wie der Text neben der Auswahl zeigt.

Hinter dem Zustand »resolved« verbirgt sich genau genommen eine ganze Gruppe von Zuständen. Ein behobener Fehler wird als »fixed« markiert, ein nicht reproduzierbarer mit »works for me«. Stellt sich heraus, dass der Fehler doppelt erfasst wurde, setzt man ihn auf »duplicate«. In der Praxis ziehen es Anwender oder Tester leider immer wieder vor, sofort einen neuen Fehler einzutragen statt erst einmal die Liste der bereits erfassten gründlich zu durchsuchen. Gerade dazu bietet Bugzilla hervorragende Möglichkeiten.

Für einen Entwickler mag die Welt in Ordnung sein, wenn er den Fehler per »resolved« aus seinem Blickfeld verbannt hat. („Regression Testing? Was ist das? Wenn es kompiliert, ist es gut, wenn es hochfährt, ist es perfekt“ – so Linus Torvalds kurz vor der Release von Linux 2.1.94.) Ängstliche Anwender wünschen sich vorher noch eine Qualitätssicherung.

Open-Source-Projekte haben dafür eine große Schar Freiwilliger, die sich aus Neugier und Interesse mit Versionen vor

der nächsten Stable-Release beschäftigen. Bei kommerzieller Software übernimmt meist ein Qualitätssicherungsteam diese Aufgabe. Wenn auch nach ihren Tests der Fehler nicht mehr auftritt, setzen sie den Status auf »verified«. Bei der Auslieferung der nächsten stabilen Version der Software können die Fehler abgehakt werden, dem entspricht der Zustand »closed«.

Der vorgestellte gerade Weg ist eine Idealisierung, an die sich die reale Welt nicht immer hält: Ist ein Tester mit der Arbeit des Entwicklers nicht zufrieden, kann er den Fehler wieder öffnen. Aus dem resultierenden Zustand »reopened« kann der Fehler die gleichen Wege wie aus dem Zustand »new« einschlagen.

Fehlersuche

In umfangreichen Projekten sammelt sich schnell eine große Menge von Fehlern an, sodass sie sich nicht mehr auf einen Blick erfassen lassen. Notwendig ist daher eine Gruppierung und Filterung der Fehler, damit jeder an dem Pro-

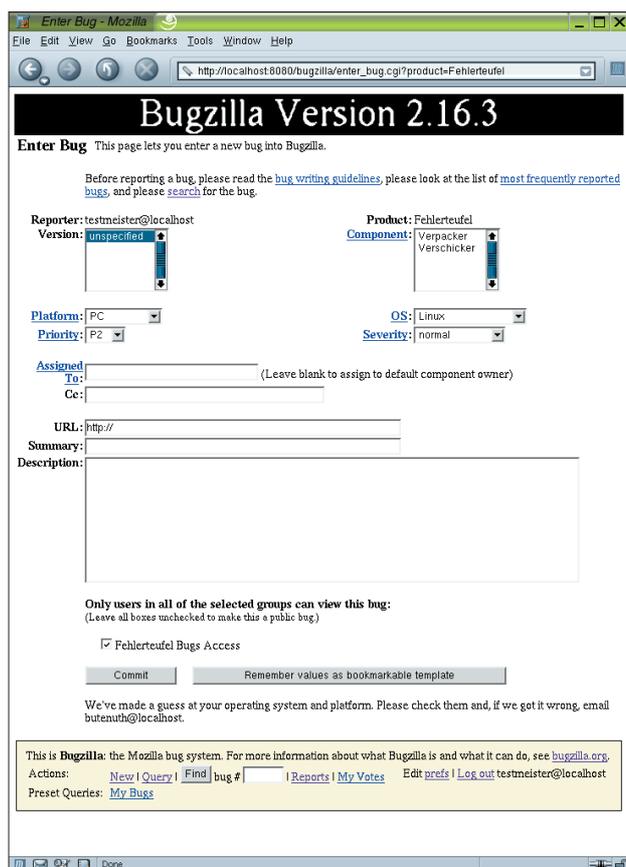


Abbildung 1: Das Eingabeformular für neue Fehler. Die Felder »Plattform« und »OS« füllt Bugzilla automatisch aus, wenn es den verwendeten Browser erkennt.

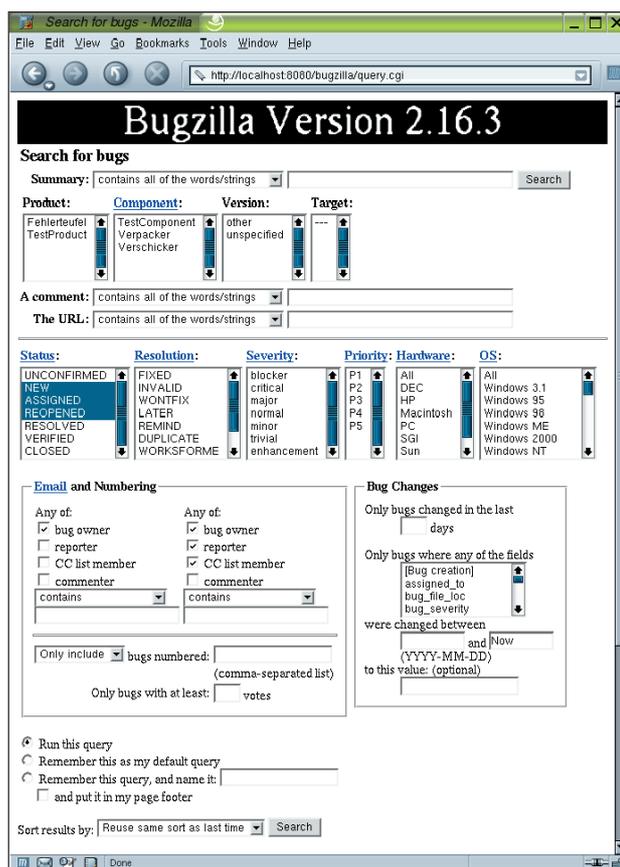


Abbildung 2: Die Suchmaske von Bugzilla wirkt stark überladen. Sie lässt sich jedoch an die eigenen Bedürfnisse anpassen.

jekt beteiligte Mitarbeiter die für ihn aktuell wichtigen Fehler sieht.

Zentraler Einstiegspunkt für die Recherche nach gemeldeten Fehlern ist die Suchmaske von Bugzilla, die in **Abbildung 2** dargestellt ist (über den Link »Query« zu erreichen). Auf den ersten Blick wirkt die Maske stark überladen, im eigenen Projekt lässt sie sich aber leicht anpassen: Einige Felder sind über die Konfiguration ein- oder ausblendbar, andere durch Ändern des HTML-Templates (siehe **Kasten „Templates ...“**).

Komplexe Eingabemaske

Im Zentrum der Eingabemaske stehen Attribute aus Aufzählungstypen wie Status und Priorität (ein Klick auf die Überschriften führt übrigens zur Online-Hilfe). Wer nur einen Status auswählt, filtert alle offenen Fehler. Durch einen zusätzlichen Filter für hohe Priorität oder spätere Sortierung nach Priorität kommt man an jene Fehler, die dringend behoben werden müssen.

Die Felder »Severity« und »Priority« wirken auf den ersten Blick redundant, ein Blocker mit der niedrigsten Priorität P5 ist kaum vorstellbar. Die Trennung lässt aber zwei sinnvolle Verfahren zu: Im ersten konfiguriert der Administrator Bugzilla so, dass jemand, der Fehler in Bugzilla einstellt, nur die »Severity« set-

zen darf, die Priorität vergibt später eine andere Person beim Bestätigen der Fehler. Dies wirkt einer Inflation der Prioritäten entgegen.

Andererseits ist es möglich, das Severity-Feld nur für die Unterscheidung von Fehlern und Änderungswünschen zu nutzen. Die Werte, die das Feld annehmen kann, lassen sich konfigurieren. Die Felder im oberen Bereich erlauben eine weitere Eingrenzung über die Produkt-Komponente-Hierarchie. Versionsnummern können getrennt pro Produkt vergeben werden.

Optional lassen sich auch künftige Releases in Bugzilla verwalten. Das Feld »Target« enthält eine dieser Releases, es eignet sich neben Priorität und Severity gut zur Planung, welche Fehler bis zu welcher Release behoben sein müssen. Die restlichen Felder im oberen Bereich erlauben eine Suche in diversen Teilbereichen der Fehler. Die Möglichkeiten gehen bis hin zur Suche mit regulären Ausdrücken in der Fehlerbeschreibung und den Kommentaren.

Die Felder unterhalb des Zentrums sind für die Suche nach Personen in bestimmten Rollen sowie nach der Historie von Fehlern zuständig. Personen werden in Bugzilla grundsätzlich durch ihre E-Mail-Adresse identifiziert. Häufig genutzt sein dürfte »bug owner«. Wählt ein Entwickler dieses Feld aus und setzt die

eigene E-Mail-Adresse ein, erhält er alle Fehler, denen er zugeordnet ist.

Bugzilla bietet die Möglichkeit, Queries zu speichern. Es ist daher nicht notwendig, alle Felder bei jeder Query komplett neu auszufüllen. Unten links in **Abbildung 2** sind die verschiedenen Varianten zu sehen:

- Die eingestellten Werte dienen als Defaultwerte für alle neuen Queries
- Die Query kann als benannte Query gespeichert werden
- Sie lässt sich zusätzlich als Link im Fußbereich der Seite speichern (hinter »Preset Queries«)

Da die Parameter einer Query komplett in der URL kodiert sind, kann der Admin auch vorbereitete Queries auf beliebigen Webseiten ablegen, zum Beispiel der Bugzilla-Startseite.

Nicht nur für Manager: Reports

Reports erhält der Bugzilla-Nutzer auf unterschiedlichen Wegen. Der einfachste Report ist die Fehlerliste, die jede Suchanfrage zurückgibt. Am unteren Rand der Liste zeigt Bugzilla einige interessante Optionen an: Über »Change Columns« lässt sich einstellen, welche Spalten in der Liste erscheinen sollen. »Send Mail to Bug Owners« startet das Mailprogramm mit einer leeren Mail an

Templates für die Oberfläche

Die HTML-Seiten der Bugzilla-Oberfläche sind nicht tief im Perl-Code vergraben, sondern werden aus Templates generiert. Programmlogik und Präsentation sind auf diese Weise weitgehend voneinander getrennt, die Oberfläche lässt sich mit ein paar HTML-Kenntnissen und ohne besonderen Perl-Sachverstand verändern.

Lokalisierung

Die Templates liegen im Unterverzeichnis »template« nach Sprachen getrennt [3]. Je nach Einstellung der Default-Sprache im Browser bekommt der Nutzer die passenden Seiten in seiner Sprache geliefert. Innerhalb jeder Sprache unterscheidet Bugzilla noch zwischen den Standardtemplates (»default«) und persönlich angepassten Templates (»custom«). Wenn man ein Template ändert, sollte man daher das Original von »default« nach »custom« kopieren und die Kopie anpassen. Hier die wichtigsten Templates:

- »index.html.tmpl« ist die Einstiegsseite, auf der viele sicherlich das Bild mit der Ameise durch ein Firmen- oder Projektlogo ersetzen möchten. Es ist auch ein guter Platz für Links zu einem projektspezifischen Nutzungskonzept sowie für vordefinierte Queries und Reports.
 - »global/header.html.tmpl« wird per Include in jeder Seite referenziert. Hier lassen sich globale Einstellungen wie Hintergrundfarbe oder die Nutzung eines CSS-Stylesheets vornehmen.
 - »global/footer.html.tmpl« generiert den gelben Kasten am Fuß jeder Seite. Die Zeile „This is Bugzilla, the Mozilla ...“ kann man entweder streichen oder durch eine projektspezifische Zeile ersetzen. Auch hier könnte man Links zu Standardqueries ergänzen.
 - »global/banner.html.tmpl« ist für die Erzeugung der Überschrift am Seitenkopf zuständig.
 - »search/form.html.tmpl« erzeugt die Suchseite. Diese Seite lässt sich durch Weglassen ungenutzter Felder (beispielsweise »Hardware« oder »OS«) vereinfachen. Innerhalb des Bugzilla-Projekts ist schon viel über die Komplexität der Suchmaske diskutiert worden. Leider laufen die meisten Vereinfachungen darauf hinaus, dass Funktionalität verloren geht.
 - »bug/edit.html.tmpl« ist für die Edit-Seite zuständig. Es gelten die gleichen Bemerkungen wie beim Such-Template.
- Mit Hilfe der Templates lässt sich prinzipiell die gesamte Oberfläche auf den Kopf stellen, notwendig ist es in den meisten Fällen allerdings nicht. Durch einige kleine Anpassungen an den ersten vier genannten Templates lässt sich aber ein sehr individuell aussehendes Bugzilla zusammenstellen, das die Eingabemaske auf das Nötige beschränkt. Der Aufwand dafür liegt – etwas HTML-Kenntnisse vorausgesetzt – bei weniger als einer Stunde.

Installation und Konfiguration von Bugzilla

Bugzilla gilt als schwer installierbar, was sicher schon viele potenzielle Nutzer abgeschreckt hat. In der Tat ist es mit dem üblichen »configure;make;make install« nicht erledigt. Wie bei vielen anderen Produkten hilft ein Blick in die Anleitung, die man nach dem Auspacken der Tar-Datei im Unterverzeichnis »docs/html/index.html« findet. Im Kapitel 4.1 (»Step-by-Step Install«) stehen die wichtigsten Installationsschritte.

Ein Aufruf von »checksetup.pl« liefert eine Liste aller Perl-Module, die nicht oder in einer zu alten Version installiert sind. Viele der notwendigen Module sind in den aktuellen Distributionen enthalten, zum Beispiel kann man bei Suse 8.2 (Professional) alle bis auf das (optionale) Charts-Modul direkt über Yast 2 installieren. So lässt sich gleichzeitig mit den Hausmitteln der Distribution sicherstellen, dass von den Modulen benötigte Bibliotheken (wie »libpng«) ebenfalls installiert werden. Alle Module findet man im Zweifelsfall auf [<http://www.cpan.org/>].

Grundeinstellungen

Nach dem ersten erfolgreichen Lauf von »checksetup.pl« sollte im Bugzilla-Verzeichnis eine Datei »localconfig« existieren. In dieser Datei werden Bugzilla-Parameter eingestellt, die nicht über die Weboberfläche zugänglich sind und die man sinnvollerweise nach der Installation auch nicht mehr ändert. Dazu gehören die Datenbankparameter und die Ausprägungen einiger Aufzählungstypen:

- »severities«: Die Schwere eines Fehlers, vorgegeben sind sieben Stufen (»blocker« bis »enhancement«). In der Praxis reichen meist zwei Stufen: »Fehler« und »CR/Änderungswunsch«.
- »priorities«: Priorität für die Fehlerbehebung. Die Vorgabe von fünf Stufen führt in der Praxis meist zu endlosen Diskussionen. Sinnvoll sind meist nur drei Prioritäten: System unbenutzbar, ernste, aber umgehbarer Fehler sowie harmlose Fehler, etwa Schreibfehler in Menütexten.
- »opsys«: Betriebssystem. Die 32 Einträge sollte man auf jene zusammenstreichen, die im eigenen Projekt vorkommen. (Wer sowieso nur mit einem Betriebssystem arbeitet, kann das Feld aus diversen Templates streichen.)
- »platforms«: Zum Beispiel PC oder Sun. Das Feld ist auch gut verwendbar, um zwischen Umgebungen wie Entwickler-PC, Testumgebung und Produktionsumgebung zu unterscheiden.

Nachdem alle Parameter eingestellt sind, folgt ein zweiter Aufruf von »checksetup.pl«. Er fragt Name, E-Mail Adresse und Passwort des

Bugzilla-Administrators ab. Anschließend legt das Skript die Datenbank an und füllt sie mit den Stammdaten.

Apache anpassen

Vor dem Bewundern der Bugzilla-Seiten im Webbrowser steht jetzt nur noch eine Ergänzung in der Apache-Konfigurationsdatei. Für Version 1.3.27 sind die folgenden Zeilen in »httpd.conf« zu ergänzen:

```
01 AddHandler cgi-script .cgi
02 <Directory
    "/home/apache/htdocs/bugzilla">
03     Options Indexes FollowSymLinks
        MultiViews ExecCGI
04     DirectoryIndex index.html index.cgi
05     AllowOverride None
06     Order allow,deny
07     <Files localconfig>
08         Deny from all
09     </Files>
10     Allow from all
11 </Directory>
12 <Directory
    "/home/apache/htdocs/bugzilla/data">
13     <Files comments>
14         Allow from all
15     </Files>
16     Deny from all
17 </Directory>
18 <Directory
    "/home/apache/htdocs/bugzilla/shadow">
19     Deny from all
20 </Directory>
```

Pfadnamen in der Datei sind natürlich an lokale Gegebenheiten anzupassen. Nach einem Neustart von Apache (»apachectl restart«) ist unter »http://localhost/bugzilla« die Startseite von Bugzilla zu sehen. Falls nicht, geht an einer Fehlersuche in »error_log« im Logverzeichnis von Apache kein Weg vorbei.

Verwaltung von Bugzilla

In dem neu eingerichteten Bugzilla existiert vorerst nur der Account des Verwalters »maintainer«, der dem Superuser auf Systemebene entspricht. Der Verwalter darf innerhalb von Bugzilla alles, unter anderem Rechte an andere Benutzer vergeben. Es lassen sich so für bestimmte Teilaspekte Rechte delegieren, also etwa Gruppen-, Produkt- oder Komponenten-Verwaltung.

Als ersten Schritt sollte der Admin jedoch auf die Seite »Parameters« (erreichbar über die Fußleiste) wechseln und dort einige Einstellungen überprüfen und anpassen. Dazu gehören:

- »urlbase« ist die URL zum Bugzilla-Verzeichnis, sie wird in allen von Bugzilla verschickten Mails angegeben.

- »sendmailnow« muss eingeschaltet sein, wenn Postfix den Mailversand übernimmt. Sonst übergibt Bugzilla die Sendmail-Option »-ODeliveryMode=deferred«, die Postfix nicht kennt.

- »usetargetmilestone« kontrolliert, ob ein Feld »Target« pro Fehler existiert. »Target« beschreibt die Zielvorgabe für die Behebung des Fehlers. Empfehlung: einschalten und nutzen.

- »usestatuswhiteboard« aktiviert ein einzeliliges Freitextfeld pro Fehler. Immer dann praktisch, wenn man ein zusätzliches Feld pro Fehler benötigt.

- »usevotes« aktiviert den zusätzlichen Zustand »unconfirmed«. Sinnvoll für öffentlich zugängliche Bugzilla-Instanzen, bei denen die Nutzer darüber abstimmen sollen, welche Fehler oder Erweiterungen zu bearbeiten sind.

- »emailregexp«: Standardmäßig legen sich Benutzer ihre Accounts selbst an und bekommen ein zufälliges Passwort per E-Mail geschickt. Über diesen regulären Ausdruck lässt sich kontrollieren, wie die zugehörigen Adressen aussehen müssen (etwa alle aus der eigenen Firmendomain).

- »commenton...«: Mit diesen Flags lässt sich für diverse Statusübergänge erzwingen, dass der auslösende Nutzer einen Kommentar zum Übergang schreibt.

Auch einige andere Parameter auf dieser Seite haben nicht unbedingt zum eigenen Projekt passende Defaultwerte. Erklärungen stehen immer direkt neben den Parametern.

Mehrere Produktinstanzen

Eine Bugzilla-Instanz kann Fehler für verschiedene Produkte nebeneinander verwalten. Über die Seite »products« werden die Produkte angelegt und gepflegt. Pro Produkt muss mindestens eine Komponente angelegt werden, eine feinere Unterteilung ist in den meisten Fällen jedoch sinnvoll. Jeder Komponente ist ein Nutzer zugeordnet, bei dem automatisch alle Fehler landen, sofern nicht bei der Fehlereingabe explizit ein anderer Nutzer angegeben wurde.

Die oben beschriebene Selbstanmeldung von Benutzern ist nicht immer sinnvoll. Nutzer lassen sich auch vom Verwalter anlegen, der dann direkt die Rechte vergibt (Abbildung 4). Die Selbstanmeldung lässt sich abschalten, indem man das zugehörige CGI-Skript »create-account.cgi« löscht. Jeder Nutzer kann über die Seite »preferences« einsehen, welche Rechte er hat. Dort ist auch sehr detailliert einstellbar, wann Bugzilla eine Mail generieren soll oder welche Queries der Nutzer in der Fußleiste sehen möchte.

alle in der Liste auftretenden Bug-Besitzer. »Change Several Bugs at once« ist schon keine Reporting-Funktion mehr: Damit sind Änderungen an mehreren Bugs gleichzeitig möglich. Beispielsweise lassen sich mehrere Fehler einem Entwickler zuordnen oder nach einer Auslieferung mehrere Fehler von »verified« auf »closed« setzen.

Wem das HTML-Format der Fehlerlisten nicht genügt, kann die Tabelle im Browser markieren und per Copy & Paste in eine Tabellenkalkulation übertragen. Dabei bleiben die Tabellenstruktur und auch die Links erhalten. In Open Office genügt also ein Klick auf die Fehlernummer – und der Browser zeigt die Details zum Fehler.

Der zweite Weg zu Reports führt über den Link »Reports« im Fußbereich. Dort öffnen sich sowohl einige Text-Reports als auch grafische Reports mit Zeitreihen über Fehlerhäufigkeiten. Dafür sind jedoch zwei Vorbedingungen erforderlich. Das für die Sammlung der Fehlerstatistiken zuständige Skript »collectstats.pl« ist täglich zu starten (per Cronjob). Außerdem müssen die Perl-Module GD und Chart installiert sein.

Abbildung 3 zeigt das Ergebnis einer solchen Auswertung, es handelt sich um die Statistik zum Mozilla-Projekt. Für die Reports gilt die gleiche Aussage wie für Suchabfragen: Alle Parameter sind in der URL kodiert. Links zu fertig konfigurierten Reports lassen sich daher in jede HTML-Seite integrieren.

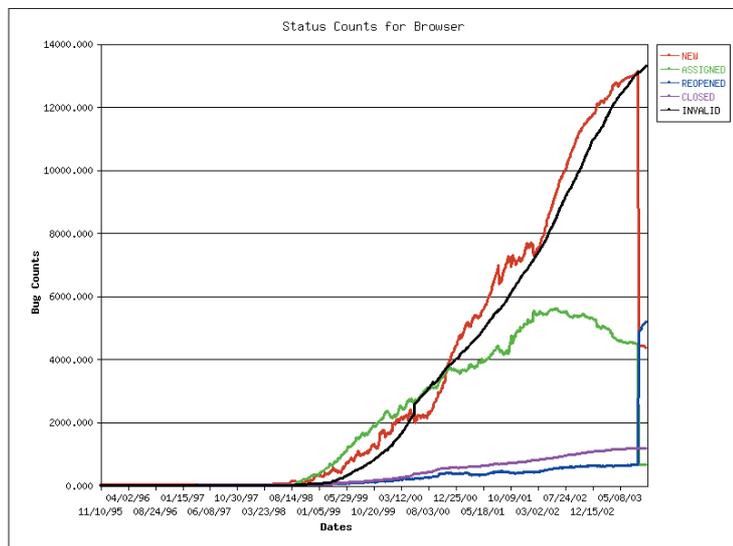


Abbildung 3: Wenn die Perl-Module GD und Chart installiert sind, kann Bugzilla auch Grafiken erzeugen. Im Beispiel die Anzahl der Fehler im Mozilla-Projekt über die Zeit.

Wenn das nicht genügt, bleibt immer noch die Möglichkeit, mit einer Tabellenkalkulation direkt auf die MySQL-Datenbank zuzugreifen. Das geschieht sinnvollerweise über einen getrennten Datenbankaccount, der nur Leserechte auf der Datenbank besitzt. Dateninkonsistenzen und Verluste durch versehentliche Änderungen in der Tabellenkalkulation lassen sich damit zuverlässig verhindern.

Fazit

Bugzilla ist anzumerken, dass es aus der Praxis heraus entstanden ist. Viele Kleinigkeiten sind sehr gut durchdacht. Auch nach längerer Arbeit entdecken Nutzer und Betreiber immer wieder Ecken, in denen sie noch nie gewesen sind. Der hier vorhandene Platz reicht bei weitem nicht aus, alle Features von Bugzilla zu beschreiben. Das ausgereifte Fehlermodell erlaubt auch die Einbeziehung einer QS-Abteilung, die behobene Fehler abnehmen muss, bevor sie zu den Akten kommen. Dies für den kommerziellen Einsatz wichtige Feature fehlt in vielen Tools fürs kleinere Fehlermanagement.

Bei einem mächtigen Werkzeug wie Bugzilla sind die Installation und Administration nicht auf die leichte Schulter zu nehmen, frei heißt halt nicht ohne (Personal-)Kosten. Dafür bekommt ein Projekt-

team oder gleich eine ganze Firma jedoch ein vielseitiges und flexibles System, das mit allen möglichen Browsern (von Mozilla über Lynx bis zum Internet Explorer) ohne zusätzliche Installation nutzbar ist. (uwo) ■

Infos

- [1] Bugzilla-Homepage: [\[http://www.bugzilla.org\]](http://www.bugzilla.org)
- [2] Diverse Informationen zu Bugzilla (der angebotene Installer hat auf Suse 8.2 nicht funktioniert): [\[http://www.softwaretesting.de/article/archive/6/\]](http://www.softwaretesting.de/article/archive/6/)
- [3] Deutsche Übersetzungen von Templates und Handbuch, beides leider nicht komplett: [\[http://bugzilla-de.sourceforge.net/index.de.html\]](http://bugzilla-de.sourceforge.net/index.de.html)

Der Autor

Dr. Roger Butenuth ist seit vier Jahren bei der Firma sd&m beschäftigt und hat dort Individualsoftware für verschiedene Kunden mitentwickelt. Zurzeit ist er damit beschäftigt, die Entwicklungsumgebung für Projekte zu verbessern. In diesem Rahmen ist auch der Kontakt zu Bugzilla entstanden.

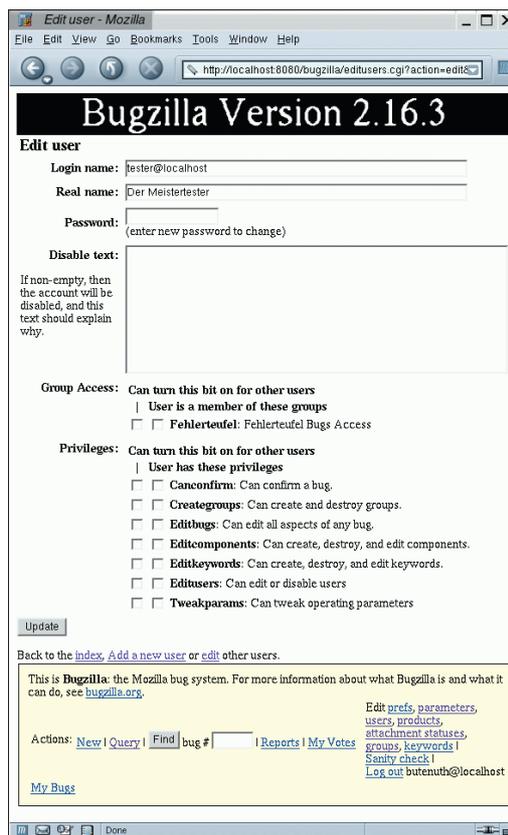


Abbildung 4: Für jeden Nutzer lässt sich festlegen, in welchen Gruppen er Mitglied ist und welche anderen Privilegien er hat.