

WS 2004/2005

Testen von Softwaresystemen

durchgeführt am
Institut für Systemsoftware, Fachbereich Informatik
Johannes Kepler Universität Linz

Leitung: Dr. Christoph Steindl

Bug Tracking Tools

Michael Biebl
michael.biebl@liwest.at

Manuel Weidhofer
manuel.weidhofer@liwest.at

17. November 2004

Inhaltsverzeichnis

1	BUG TRACKING TOOLS	3
1.1	PROGRAMMFEHLER	3
1.2	ANFORDERUNGEN AN BUG TRACKING TOOLS	5
2	BUGZILLA	6
2.1	FEATURES	6
2.2	VORAUSSETZUNGEN	7
2.3	INSTALLATION	8
2.4	BEDIENUNG	8
2.4.1	<i>Konzepte</i>	8
2.4.2	<i>Einen Fehler erfassen</i>	9
2.4.3	<i>Fehler suchen</i>	10
2.4.4	<i>Arbeiten mit Bugs</i>	11
2.4.5	<i>Patches</i>	12
2.4.6	<i>Statistiken erzeugen</i>	12
2.5	ANMERKUNGEN	13
3	FAST BUG TRACK	14
3.1	FEATURES VON FAST BUG TRACK	14
3.1.1	<i>e-Mail Benachrichtigung</i>	14
3.1.2	<i>Dateianhänge</i>	14
3.1.3	<i>Unique Id</i>	15
3.1.4	<i>Diagramme</i>	15
3.1.5	<i>Reports</i>	15
3.1.6	<i>Filter</i>	15
3.1.7	<i>Sicherheit</i>	16
3.1.8	<i>Datenbank</i>	16
3.1.9	<i>Bugs anonym erfassen</i>	17
3.2	PRODUKTPREIS	17
3.3	BUG ERFASSEN	17
3.4	ÜBERSICHT ERFASSTE FEHLERMELDUNGEN	18
3.5	BULK UPDATE	19
3.6	REPORT ERSTELLEN	20
3.7	FILTER ANWENDEN	21
3.8	FAZIT	22
4	ABBILDUNGSVERZEICHNIS	23

5	LITERATURVERZEICHNIS.....	23
----------	----------------------------------	-----------

1 Einleitung

Im Rahmen dieser Arbeit beschäftigen wir uns mit Bug Tracking Tools – dem Management von Programmfehlern. Im ersten Teil der Arbeit gehen wir kurz auf Programmfehler ein und beschreiben Anforderungen an Bug Tracking Tools. Im zweiten und dritten Teil stellen wir zwei Bug Tracking Tools vor – *Bugzilla* und *Fast Bug Track*.

2 Bug Tracking Tools

Die Problematik die Programmfehler – im englischen auch einfach als Bug bezeichnet – mit sich bringen ist hinreichend bekannt. Von Sicherheitslücken in Browsersoftware die tausende Computer lahm legen bis hin zum Scheitern von Millionen Dollar teuren Weltraumprojekten (Ariane 5 1996, die NASA Sonde Climate Orbiter 1999) [WIKIPE]. Fehler in Programmen und deren Auswirkungen kommen teuer zu stehen, deshalb gilt es Fehler früh zu identifizieren, zu verwalten und zu beheben. Für diese Aufgaben bieten sich Bug Tracking Tools an (auch bekannt unter dem Namen Defect Tracking Tools). Diese Werkzeuge wurden nicht zur Fehlervermeidung konzipiert, wie unterstützen vielmehr die vollständige Fehlererkennung und deren Behebung.

Durch die Verwendung von Bug Tracking Tools im Software Entwicklungsprozess, erhofft man sich folgende Fragen klären zu können: [ENTW04]

- Welche Arten von Fehler sind aufgetreten?
- Welche Software-Releases sind von den Fehlern betroffen?
- Welches Release enthält den Fix des Fehlers?
- Welcher Entwickler ist für die Behebung des Fehlers zuständig?
- Was wurde unternommen, um den Fehler zu beseitigen?
- Wurde der behobene Programmfehler nachgetestet?
- Wurde der behobene Programmfehler vom Kunden freigegeben?
- Welche Trends sind erkennbar, also Statistiken zur Qualitätskontrolle?

2.1 Programmfehler

Beim Erfassen von Programmfehlern in Bug Tracking Tools müssen Bugs genau spezifiziert werden (siehe Punkt 4.3). Die Beschreibung des Bugs soll folgende Informationen beinhalten:

- Kurzbeschreibung des Bugs
- Längere Beschreibung zum Nachvollziehen der Fehlersituation
- Priorität (niedrig bis hoch)
- Die Komponente in der der Fehler aufgetreten ist
- Zeitstempel des Auftretens
- Submitter: wer hat den Bug entdeckt und bekannt gegeben.

Ein Programmfehler nimmt, nachdem er erkannt ist, mehrere Stati ein. Der Life Cycle eines Fehlers kann folgendermaßen aussehen: [ENTWXXX]

- **Submission:** Ein Fehler wird identifiziert, das heißt es wird ein abnormales Verhalten festgestellt und in einem Bug Tracking Tool erfasst. Die Fehlermeldung kann von den Kunden, den Anwendern oder Qualitätssicherungsabteilung (Softwaretester) kommen.
- **Assignment:** Wenn es sich bei der Fehlermeldung eindeutig um einen Softwarefehler handelt (nicht z.B. um einen Bedienungsfehler) wird der Fehler zur Behebung an einen bestimmten Entwickler zugeteilt. Dadurch wird sichergestellt, dass der Fehler auch nur einmal behoben wird.
- **Eventuell Reassignment:** Der Entwickler weist den Fehler mit einer Begründung zurück (z.B. der Fehler war bereits bekannt und ist bereits gefixt).
- **Resolution:** Der Entwickler hat den Fehler behoben, der Fehler wird mit einer Fehlerbeschreibung als gefixt klassifiziert.
- **Verification:** Die Software wird nun daraufhin getestet, ob der Fehler auch tatsächlich behoben wurde und ob durch die Behebung des Fehlers nicht auch andere Fehler aufgetreten sind.
- **Closing:** Der Fehler ist behoben und nach dem erfolgreichen Test wird die Freigabe der Software erteilt.

Bug Tracking Tools bilden somit den kompletten Lebenszyklus von Programmfehlern ab. Dadurch ist es möglich, komplette Stadien der Software-Entwicklung abzubilden und Statistiken über die Effizienz des Entwicklungs-Prozesses zu generieren, wie z.B.:

- Wer liefert die meisten Fehlermeldungen?
- In welche Applikation treten am häufigsten Fehler auf?
- Wie lange dauert es bis ein Fehler gefixt ist (Zeitspanne zwischen Fehlermeldung und Behebung)?
- Wie viele Fehler werden gerade bearbeitet, sind noch zu beheben?

2.2 Anforderungen an Bug Tracking Tools

Anforderungen an Bug Tracking Tools [ENTW04]

- Jeder Benutzer des Bug Tracking Tools (Entwickler, Tester, Softwareverantwortliche, ...) muss schnell erkennen können, welche Fehler für ihn relevant sind. Die Akzeptanz von Seiten des Benutzers leidet, wenn dieser ihm zugeteilte Fehler lange suchen muss.
- Unterstützung aller gängigen Betriebssysteme (Windows, Linux, Solaris, Mac OS).
- Der Workflow zur Fehlerbehebung muss an den eigenen Entwicklungsprozess anpassbar sein. Diese Anpassung muss schnell und ohne größeren Programmieraufwand machbar sein.
- Die Möglichkeit zur Integration in gängige Konfigurationsmanagementwerkzeuge.
- Das Produkt soll über eine eigene integrierte Datenbank verfügen.
- Ein automatisiertes eMail-Benachrichtigungssystem informiert den zuständigen Entwickler mit konkreten Informationen über aufgetretene Fehler.
- Schnelle und leichte Installierbarkeit
- Intuitive Bedienung

3 Bugzilla

Bugzilla entstand als Bug Tracking Tool der Mozilla Foundation. Es war eine der ersten Entwicklungen die, nach dem Start von mozilla.org 1998, zum Download bereitstand. Das Werkzeug war ursprünglich in TCL programmiert, wurde jedoch in Perl erneut geschrieben um es mehr Entwicklern zu erlauben Code beizusteuern. Bugzilla wird als Open-Source-Software vertrieben und steht unter der Mozilla Public License. Hunderte Firmen weltweit setzen das Programm ein um Fehler besser handhaben zu können. Einige nennenswerte Firmen bzw. Produkte die Bugzilla einsetzten: mozilla.org, linux kernel, gnome, kde, apache, NASA, IBM.

Alle Angaben ab diesem Punkt beziehen sich auf Bugzilla Version 2.18rc3.

3.1 Features

- Erfassung von Fehlern

Bugzilla bietet Möglichkeiten um Fehler zu erfassen, diese Personen zuzuordnen, sowie deren Lösung zu dokumentieren.

- Versand von E-Mails

Bugzilla informiert die Zuständigen über Neuigkeiten, indem es E-Mails versendet. Jeder Benutzer kann für sich einstellen bei welchen Vorkommnissen er benachrichtigt werden möchte.

- Reports erstellen

Bugzilla kann graphische und tabellarische Reports erstellen. Tabellarische Reporte können als CSV exportiert werden, um sie z.B. in einer Tabellenkalkulation zu verwenden.

- Berechtigungsverwaltung

Die Berechtigungsverwaltung von Bugzilla ist sehr umfassend und auch für Firmenstrukturen passend. Es ist auch möglich mit einem LDAP-Verzeichnis zu arbeiten.

- Geschichte eines Fehlers

Bugzilla erfasst alle Aktivitäten mit Zeitstempeln, dadurch ist es möglich die Historie eines Bugs durchzusehen.

- Sprachanpassungen

Für Bugzilla gibt es mehrere Lokalisierungen. Diese können parallel installiert werden, die jeweilige Sprache wird über den HTTP-Header Eintrag „Accept-Language“ bestimmt (im Browser einzustellen).

- Integration

Bugzilla kann mit CVS, Bonsai, oder LXR zusammenarbeiten, um Kontext für Patches zu erhalten. Weiters ist eine Zusammenarbeit mit Tinderbox, dem Mozilla Build System möglich.

- Volltextsuche

Die Fehlerdatenbank kann mittels Volltext-Suche durchsucht werden.

- Integritätsprüfung

Ein sehr nützliches Feature ist die Überprüfung des Systems, „Sanity Check“ genannt. Dabei wird der Inhalt der Datenbank auf Inkonsistenzen, sowie die E-Mail-Warteschlange auf ungesendete E-Mails geprüft.

- Verfügbarkeit des Quellcodes

Durch die Verfügbarkeit des Quellcodes sind Modifikationen des Systems möglich und erlauben im Bedarfsfall eine Anpassung an firmenspezifische Anforderungen. Sollte es sich um allgemein nützliche Entwicklungen handeln, so kann die Veränderung bzw. Erweiterung, dem Open-Source-Gedanken entsprechend, auch in zukünftige Versionen von Bugzilla einfließen.

3.2 Voraussetzungen

Bugzilla wurde vor allem für Linux und Solaris entwickelt, läuft jedoch überall auf allen Systemen auf welchen Perl und ein Webserver (mit CGI-Fähigkeiten) verfügbar sind. Als Webserver wird Apache empfohlen, es funktionieren aber auch z.B. Microsoft Internet Information Server, oder AOL Server. Eine weitere Voraussetzung ist ein MySQL Server, für die Speicherung der Daten. Ein

Mailserver, oder Perl-Modul zum Versenden von E-Mails ist ebenfalls nötig, um Benutzer zu verständigen.

3.3 Installation

Die Installation von Bugzilla ist sehr gut dokumentiert und funktioniert in meinem Fall sehr einfach und schnell. Ein guter Schritt für Schritt Anleitung findet sich unter [BUGZ].

3.4 Bedienung

3.4.1 Konzepte

In Bugzilla werden Fehler bestimmten Komponenten zugeordnet. Eine Komponente ist ein Teil eines Produkts. Ein Produkt ist die Hauptkategorie und sollte für jedes hergestellte Produkt angelegt werden. Weitere Produkte sind z.B. Website, Administration, oder Common für Technologien die mehrfach verwendet werden. Für ein Produkt kann man Komponenten anlegen. Eine Komponente stellt eine abgeschlossene Einheit dar und wird von einer Person verwaltet. Diese Person bekommt per Voreinstellung die Bugs seiner Komponente/n zugeteilt. Zu einem Produkt können mehrere Versionen bestehen. Weiters gibt es noch Meilensteine. Diese dienen dazu Ziele festzulegen. Ein Fehler wird dabei einem Meilenstein zugeordnet um festzulegen, dass er bis zu diesem Meilenstein gelöst sein soll.

Weiters gibt es Flags um Statusinformationen zu einem Fehler festzulegen. Dieser Status kann '+' oder '-' sein. Es besteht auch die Möglichkeit eine gewünschte Option zu erfragen. Dazu kann er als Status ein '?' setzen. Dieses Fragezeichen kann, als Antwort, auf ein Plus-, oder Minuszeichen gesetzt werden.

Es können für einen Bug auch so genannte Votes gesetzt werden. Damit kann ein Benutzer anzeigen, dass es ihm die Lösung des Problems wichtig ist. Die Anzahl an Votes die ein Benutzer pro Bug vergeben kann, ist pro Produkt einstellbar. Es ist auch möglich, einen Bug automatisch den Status "UNCONFIRMED" auf den Status "NEW" zu setzen. Die Anzahl der dazu nötigen Votes wird ebenfalls bei den Produkteinstellungen festgelegt werden.

Ein Bug kann von einem anderen abhängig sein, er kann also erst aufgelöst werden, wenn der übergeordnete Fehler behoben wurde.

3.4.2 Einen Fehler erfassen

Bevor man seinen ersten Fehler erfasst sollte man die Bugwriting Guidelines lesen [BUGWG]. An dieser Stelle die wichtigsten Punkte:

- Suchen ob der Fehler bereits gemeldet wurde

Dadurch soll die Duplizierung von Fehlermeldungen vermieden werden. Dies bedeutet wenig Aufwand für den Benutzer, erleichtert dem Entwickler jedoch die Arbeit immens.

- Der Fehler muss reproduzierbar sein

Nur wenn der Entwickler den Fehler reproduzieren kann, kann er ihn lösen.

- Den Fehler so weit als möglich eingrenzen

Je eindeutiger eine Fehlermeldung ist, desto leichter ist es für den Entwickler den Fehler zu finden.

- Fehler in einer neueren Version reproduzieren

Hat man die Möglichkeit eine neuere Version zu verwenden, sollte man versuchen den Bug darin zu reproduzieren. Entwickler interessieren sich meist mehr für Versionen, die aktiv entwickelt werden.

- Nur ein Fehler pro Bugreport

Es ist wichtig immer nur einen Fehler auf einmal zu melden. Die von Bugzilla vorgegebenen Abläufe arbeiten nur mit einem Fehler pro Meldung (z.B. ein Fehler gelöst, der andere nicht => Status: RESOLVED?)

- Klarheit

Je klarer die Struktur einer Fehlermeldung, und je klarer der Titel und die Zusammenfassung, desto einfacher ist es mit dem Bug umzugehen, bzw. ihn zu finden.

Bevor man einen Bug eingeben kann, muss man das zugehörige Produkt auswählen danach erscheint folgende Eingabemaske:

Enter Bug This page lets you enter a new bug into Bugzilla.

Before reporting a bug, please read the [bug writing guidelines](#), please look at the list of [most frequently reported bugs](#), and please [search](#) for the bug.

Reporter: michael.biebl@liwest.at	Product: Test1
Version: unspecified	Component: component1
Platform: Macintosh	OS: Mac OS X 10.0
Priority: P2	Severity: normal
Assign To: michael.biebl@liwest.at	
Cc:	
URL: http://	
Summary:	
Description:	
Depends on:	
Blocks:	

We've made a guess at your operating system and platform. Please check them and, if we got it wrong, email michael.biebl@liwest.at.

Abbildung 1: Eingabemaske zum Erfassen von Fehlern

Bei Feldern die als Beschriftung einen Link aufweisen, führt dieser zu einer Beschreibung des Feldes bzw. der möglichen Werte. Der Link “Component“ führt zur Übersichtsseite der Komponenten.

3.4.3 Fehler suchen

Es gibt zwei verschiedene Masken zum Suchen von Fehlern. Eine einfache und eine erweiterte. Es ist möglich seine Suchen zu speichern und sie im Menü am Ende jeder Seite anzuzeigen. Selbstverständlich kann man seine gespeicherten Suchen bearbeiten und löschen. Es existiert bereits zu Beginn eine Suche namens “My Bugs“, welche alle offenen Bugs des angemeldeten Benutzers zeigt.

Das Ergebnis einer Suche entspricht einer Liste mit allen Fehler, die die Suchkriterien erfüllen. Die angezeigten Felder können nach den persönlichen Vorlieben und Notwendigkeiten eingestellt werden ("Change Columns"). Das

This is Bugzilla
[Bugzilla](#) Version 2.18rc3

Bug List
 Sun Nov 14 11:10:32 CET 2004
Bugzilla would like to put a random quip here, but no one has entered any.

ID	Sev	Pri	Plt	Assignee	Status	Resolution	Summary
2	nor	P1	Mac	michael.biebl@liwest.at	NEW		Test2
1	min	P2	Mac	michael.biebl@liwest.at	NEW		not existant

2 bugs found.

[Long Format](#) | [CSV](#) | [iCalendar](#) | [Change Columns](#) | [Edit Search](#) | [Remember search](#) as

Actions: [Home](#) | [New](#) | [Search](#) | bug # | [Reports](#) | [My Requests](#) | [My Votes](#) | [Sanity check](#) | [Log out](#) michael.biebl@liwest.at
 Edit: [Prefs](#) | [Parameters](#) | [Users](#) | [Products](#) | [Flags](#) | [Groups](#) | [Keywords](#)
 Saved Searches: [My Bugs](#) | [test](#)

Ergebnis kann in verschiedene Formate exportiert werden. Meiner Meinung nach ist das iCalendar-Format sehr nützlich um sich seine Bugs als Todo-Elemente in seinen

Aufgaben

- not existant
- Test2

Test2

Beendet ☐
 Priorität Ohne ↕
 Termin ☐
 Kalender bugs-2004-11-14 ↕
 URL http://127.0.0.1/bugzilla/show_bug.cgi?id=2

Notizen

Kalender einzufügen. Es besteht auch die Möglichkeit einzelne Fehler als Xml zu exportieren. Leider gibt es dafür keinen Link, aber durch einen xml-Export kann eine Integration in andere Tools sinnvoll implementiert werden, ohne dass direkt mit der Datenbank gearbeitet werden muss.



3.4.4 Arbeiten mit Bugs

Um Bugs zu bearbeiten wählt man einfach einen Fehler aus, der als Ergebnis einer Suche angezeigt wird. Die Maske ist ähnlich wie jene zur Erfassung von Fehlern. Es ist möglich Dateien anzuhängen, Kommentare hinzuzufügen, oder Flags zu setzen. Weiters kann man Votes abgeben und den Status des Bugs ändern. Möchte man den Fehler einer anderen Person zuordnen, kann man das ebenfalls hier machen.

3.4.5 Patches

Man kann in Bugzilla auch Patches verwalten. Dazu fügt man einen Patch als Attachment zu einem Bug hinzu. Ein Patch wird von Bugzilla übersichtlich dargestellt und kann auch mit anderen Patches verglichen werden. Benützt man CVS, Bonsai, oder LXR, kann Bugzilla den Patch im Kontext der Versionsverwaltung anzeigen.

Attachment #7: test diff -u for [bug #4](#)

[View](#) | [Edit](#) | [Raw Unified](#) | Differences between  and this patch  [Collapse All](#) | [Expand All](#)

(-) menu.htm (-4 / +1 lines)	
Lines 20-29 (Link Here)	
<code><div class="button" onclick="toggleSidemenu();"></code>	<code><div class="button" onclick="toggleSidemenu();"></code>
<code> f</code>	<code> f</code>
<code></div></code>	<code></div></code>
<code> fjdklfjdskl</code>	<code>test2 </div></code>
<code> klfjsaklfds</code>	
<code> kfjaklsd</code>	
<code></div></code>	
<code><div id="watchWrapper"></code>	<code><div id="watchWrapper"></code>
<code> <div id="watchmenu"></code>	<code> <div id="watchmenu"></code>
<code> <div class="menu"></code>	<code> <div class="menu"></code>

3.4.6 Statistiken erzeugen

Hinter dem Punkt „Reports“ verbirgt sich die Möglichkeit Statistiken zu erstellen. Dabei kann man aus fast beliebige Kombinationen von Attributen (Programm, Komponente, Benutzer, ...), Tabellen und Grafiken erzeugen. Weiters kann man die Menge der Fehler wie bei einer Suche einschränken. Diese Funktionen bieten einen guten Überblick über die erfassten Fehler. Es ist möglich das Ergebnis eines Reports als CSV zu exportieren um die Daten in anderen Programmen, wie einer Tabellenkalkulation weiter zu verwenden.

Report: Status / Assignee / Product

2004-11-14 11:22:17

NEW

Product

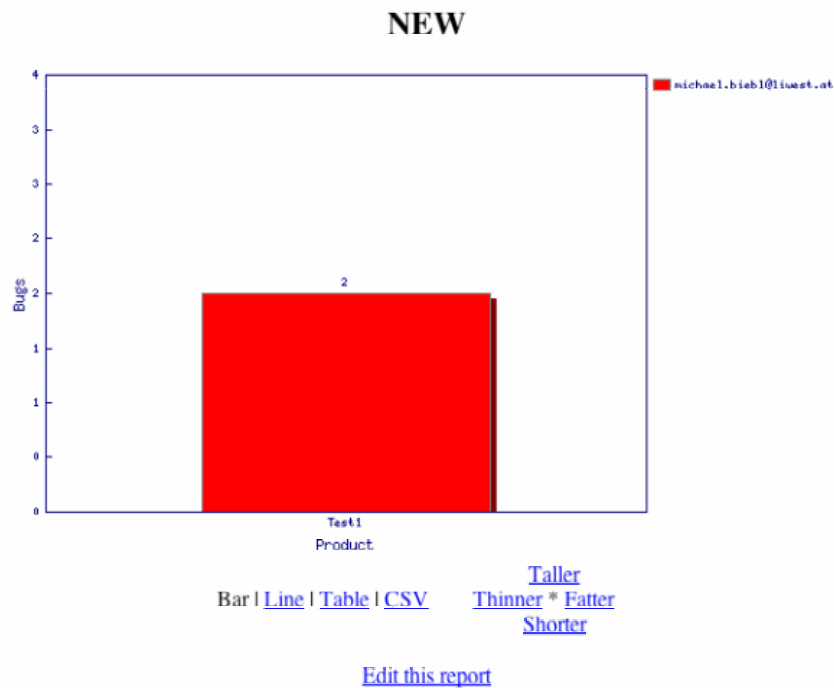
Assignee	Product		
		Test1	Total
	michael.biebl@liwest.at	2	2
	Total	2	2

[Bar](#) | [Line](#) | [Table](#) | [CSV](#)

[Edit this report](#)

Report: Status / Assignee / Product

2004-11-14 11:22:48



3.5 Anmerkungen

Um die Benutzerfreundlichkeit zu erhöhen ist es sehr sinnvoll die “Sidebar“ zu verwenden um komfortabel durch Bugzilla zu navigieren (Auf der

Bugzilla 127.0.0.1

test2

[new search](#)
[reports](#)
[new bug](#)
[new user](#)
[log in](#)

This is Bugzilla

Bugzilla Version 2.18rc3

Bug List

Sun Nov 14 11:15:35 CET 2004

Bugzilla would like to put a random quip here, but no one has entered any.

ID	Sev	Pri	Plt	Assignee	Status	Resolution	Summary
2	nor	P1	Mac	michael.biebl@liwest.at	NEW		Test2

One bug found.

Long Format
[CSV](#) | [iCalendar](#) | [Change Columns](#)
[Edit Search](#)

Remember search

Actions: [Home](#) | [New](#) | [Search](#) | bug # | [Reports](#) | [Requests](#) | [New Account](#) | [Log In](#)

Startseite zu aktivieren). Meiner Meinung nach ist die Benutzerfreundlichkeit ein Punkt, der verbessert werden sollte. Ich habe auch zwei Eclipse-Plugins für Bugzilla probiert, jedoch funktionierten beide nicht. Eine Integration in eine Entwicklungsumgebung wäre aber sehr wünschenswert. Dazu fehlt es Bugzilla aber noch an besseren Möglichkeiten zum Datenaustausch. Sonst finde ich das Bug-Tracking-System sehr gelungen, vor allem die Zusammenhänge und Abläufe sind sehr sinnvoll.

4 Fast Bug Track

Fast Bug Track von der Firma Alcea Technologies Inc. ist ein Bug Tracking Tool mit einem Webinterface. Fast Bug Track ist plattform-unabhängig, es läuft auf Windows (2k, XP, NT), Linux (alle Distributionen), Solaris und Mac OS und es ist ebenso verfügbar für alle gängigen Webbrowser (IE3+, Netscape 3+, Mozilla, Opera, ...). Die Installation ist sehr einfach zu handhaben. Beim ersten Start des Programms muss man die folgenden Einstellungen tätigen.

- Server konfigurieren: Port, SMTP Server für e-Mail Benachrichtigungs-Dienst, e-Mail Adresse
- Benutzer anlegen: zumindest soll man einen Administrator anlegen, um das System weiter zu administrieren.
- Projekte anlegen: Namen der aktuellen Projekte sowie deren Module erfassen (z.B. Client, Server, Database, ...).
- Projekt-Environment erfassen (z.B. Entwicklungs-, Simulations- und Produktionssystem).

Nachdem man diese ersten Schritte erledigt hat, kann nach einem Neustart von Fast Bug Track bereits mit der Arbeit begonnen werden.

4.1 Features von Fast Bug Track

4.1.1 e-Mail Benachrichtigung

Wenn ein Fehler einem Entwickler zugewiesen wird, bekommt dieser unverzüglich eine Benachrichtigung per e-Mail zugesendet. Es ist auch möglich Regeln für den Benachrichtigungsdienst zu erstellen (z.B. an den Benutzer X werden nur Benachrichtigungen mit Priorität „Emergency“ gesendet).

Benutzer von Fast Bug Track können sich über alle Fehler informieren lassen, indem sie sich in eine Verteilerliste eintragen.

4.1.2 Dateianhänge

An Fehlermeldungen können verschiedene Dokumente angefügt werden (z.B. Word Dokumente, Screenshots, Bilder, Log-Files, ...). Diese stehen durch den Benachrichtigungsdienst allen eingetragenen Benutzern zur Verfügung.

4.1.3 Unique Id

Für Projekte können Unique Ids vergeben werden. Auf diese Weise können mehrere Kundengruppen für ein Produkt angelegt werden und Fehlermeldungen getrennt bearbeitet werden.

4.1.4 Diagramme

Fast Bug Track bietet eine Reihe von Darstellungsformen für die Visualisierung der erfassten Daten (z.B. Priorität von Bugs, Bearbeitungsstatus, Erstelldatum, Bugs je Projekt, betroffene Module, bisherige Bearbeitungsdauer, erwarteter Zeitpunkt der Erledigung, Bugs je Entwickler, ...) (zweidimensionale und dreidimensionale Darstellung):

- Kreisdiagramme
- Balkendiagramme
- Liniendiagramme

4.1.5 Reports

Das Programm hat sechs vorgefertigte Reports inkludiert. Mit diesen Reports erhält man schnell Auskunft darüber an wenn Bugs zugewiesen wurden, welche Priorität und welchen Status sie besitzen, in welchem Projekt und Modul sie aufgetreten sind, usw. Ebenso kann man selbst Reports zusammenstellen indem man sich einfach jene Daten zusammenstellt die für einen relevant sind. Die Abbildung 2 zeigt z.B. alle noch ausständigen Aufgaben und den Zeitpunkt, zu dem sie zuletzt bearbeitet worden sind.

Reports können per e-Mail versendet werden und ebenso als Microsoft Word und Microsoft Excel Dokument exportiert werden.

4.1.6 Filter

Filter ermöglichen die Einschränkung der erfassten Bugs nach bestimmten Kriterien (z.B. alle Bugs mit Status *Open*, *Ready for Test*, *Closed*).

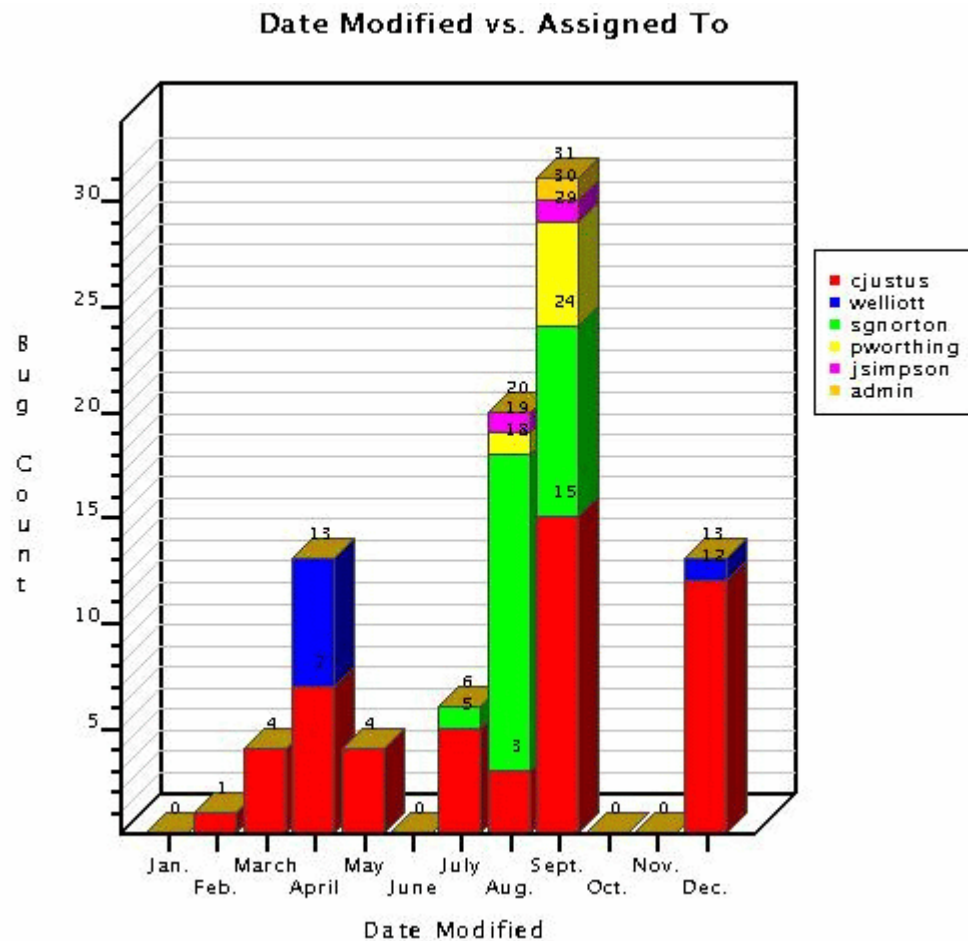


Abbildung 2: Balkendiagramm [FASTBT]

4.1.7 Sicherheit

Fast Bug Track unterstützt eine Benutzerverwaltung. Es können Gruppen angelegt werden und Benutzer zu diesen Gruppen zugeordnet werden. Bei Benutzern sind drei Rollen vorgesehen:

1. Administrator: Benutzer anlegen, verwalten, Bugs erfassen und editieren.
2. Normaler Benutzer: Entwickler, er kann Bugs erfassen und editieren.
3. Nur Lesen: Einsicht in erfasste Fehlermeldungen

Erstellte Fehlermeldungen können nur vom Administrator und dem Ersteller der Meldung geschlossen werden.

4.1.8 Datenbank

Die Trial-Version von Fast Bug Track verwendet keine Datenbank zum Speichern der Benutzerdaten, Fehlermeldungen, Reports und Diagramme, diese werden in einer

Datei abgelegt. Die Enterprise Version unterstützt Oracle, MS-SQL, MySQL, Sybase und Postgres Datenbanksysteme via JDBC.

Laut Produktbeschreibung können mehrere hundert Benutzer, sowie mehrere 10.000 Bugs mit dem System verwaltet werden. Diese Angabe konnten wir anhand der Trialversion von Fast Bug Track nicht überprüfen, hier waren wir mit gerade einmal 25 Bugs ziemlich eingeschränkt.

4.1.9 Bugs anonym erfassen

Mit Fast Bug Track hat man die Möglichkeit Fehler dem Entwicklungsteam anonymisiert anzuzeigen. In einem Web-Formular werden die wesentlichen Informationen über einen aufgetretenen Fehler erfasst und dem System übermittelt. Szenario: Es wird ein neues Software-Releases fertig gestellt und eine große Anzahl von Anwendern wird das Programm benutzen (mehrere tausend). Das Software Unternehmen möchte über Fehler informiert werden. In diesem Fall ist es nicht möglich, und auch nicht erwünscht, alle Anwender im Bug Tracking Tool als Benutzer zu erfassen und ihnen somit die Erfassung von Fehler zu ermöglichen. Sie sollen schnell und bequem Fehler in einem Formular erfassen können und an das Entwicklungsteam weiterleiten.

4.2 Produktpreis

Die Trial-Version von Fast Bug Track ermöglicht das erfassen von 25 Bugs. Die günstigste Variante Fast Bug Track zu erwerben ist die Lizenz für 2 Benutzer für \$149,-- US. Die Enterprise Solution kostet \$3.199,-- US. Sie beinhaltet eine ungeschränkte Anzahl von Lizenzen.

4.3 Bug erfassen

Einen neuen Programmfehler erfasst man über den Menüpunkt „New Bug“. In der daraufhin angezeigten Eingabemaske (siehe Abbildung 3) können alle wichtigen Informationen den Bug betreffend erfasst werden:

- Eine Bezeichnung für den Fehler (kurz und wenn möglich aussagekräftig).
- Entwickler der den Bug beheben soll.
- Projekt welches betroffen ist.
- Die Priorität des Fehlers (niedrig bis Notfall)
- Betroffene Softwarekomponente (z.B. Client-Modul, Server, ...)

- Eine ausführliche Beschreibung. Diese soll die Umstände beschreiben, die zu der Fehlersituation geführt haben (z.B. welche Aktionen hat der Benutzer ausgeführt, ...).

Optional können hier weitere Empfänger für die Fehlermeldung eingetragen werden. Sie erhalten bei jeder Aktualisierung (z.B. der Bug ist behoben, die Bearbeitung dauert noch an, ...) der Daten eine Benachrichtigung per e-Mail gesendet. Weiters kann man zu einer Fehlermeldung Anhänge anfügen, z.B. ein Fehlerprotokoll oder ein Screenshot.

Wenn die Fehlermeldung erfasst ist kann sie gespeichert werden. Der zugeordnet Entwickler bekommt eine Benachrichtigung per e-Mail gesendet, ebenso alle Personen auf der Verteilerliste.

New Bug Submit Bug

Details			
Subject:	Fehler E -801 bei Node Javeli PVS		
Assigned To:	Manuel	Project:	EAI Operator
Status:	Open	Priority:	2 - High
Environment:	produktion	Version:	
Area:	client		
Description:	Vorgehensweise die zu diesem Fehler geführt hat		
Notify List:	Manuel Clear Manuel Michael Manuel Weidhofer		
Attachments			
Add Attachment :		Durchsuchen...	Comment :
Add Attachment :		Durchsuchen...	Comment :
Project Management			
Requested Due Date:	? 2004/12/01	Estimated Hours:	
Actual Completion Date:	?	Actual Hours:	
Percent Complete:		Parent:	

Abbildung 3: Bug erfassen

4.4 Übersicht erfasste Fehlermeldungen

Über den Menüpunkt „Main Menu“ gelangt man zu einer Aufstellung aller erfassten Fehlermeldungen. Man kann die Liste nach den Attributen sortieren, sich nur offene oder nur erledigte Fehlermeldungen anzeigen lassen. Zusätzliche Spalten (z.B. welches Projekt, Modul, Erstelldatum, gesamte Bearbeitungszeit, ...) können über den Menüpunkt „Customize...“ hinzugefügt werden. Zusätzlich wird noch eine Aufstellung der zu bearbeitenden Aufgaben je Benutzer angezeigt.

Bug Id	Subject	Assigned To	Date Modified	Status	Priority
Edit 2	LogDatei LogLevel setzen	manuel.weidhofer	Nov 18 2004 (22:57)	Open	3 - Medium
Edit 4	Telegramm erfassen	Peter	Nov 18 2004 (23:00)	Open	2 - High
Edit 2	Search problem	Peter	Nov 18 2004 (22:56)	Open	4 - Low
Edit 3	Speichern nicht möglich	Michael	Nov 18 2004 (22:59)	Open	1 - Emergency
Edit 3	Fenster lässt sich nicht schließen	Manuel	Nov 18 2004 (22:54)	Open	2 - High
Edit 1	Fehler eMail Verteiler	Manuel	Nov 18 2004 (21:59)	Open	3 - Medium
Edit 1	Fehler E -801 bei Node Javeli PVSW	Manuel	Nov 18 2004 (20:44)	Open	2 - High
Edit 4	Max open Cursor - DB QESE	Manuel	Nov 18 2004 (22:59)	Open	3 - Medium
Edit 1	XML File not valid	Manuel	Nov 18 2004 (22:58)	Ready For Retest	3 - Medium
Edit 5	TGViewer Telegramm zerlegen - wird falsch zerlegt	Gerald	Nov 18 2004 (23:02)	Open	3 - Medium
Edit 5	PVSEExplorer Stammdaten löschen nicht möglich	Alexander	Nov 18 2004 (23:02)	Open	3 - Medium

Summary:

	Open	Ready For Retest	Total
Manuel	4	1	5
Alexander	1	0	1
Gerald	1	0	1
Peter	2	0	2
Michael	1	0	1
manuel.weidhofer	1	0	1
Total:	10	1	11

Abbildung 4: Liste erfasste Bugs

4.5 Bulk Update

Eine nützliche Funktion stellt *Bulk Update* dar. Mit dieser Funktion lassen sich bei mehreren Aufgaben zugleich bearbeiten. Wenn z.B. mehrere Bugs behoben worden sind, könne diese in der Tabelle selektiert werden und über Pull-down-Menü der Status *Closed* zugewiesen werden. Ebenso kann man für mehrere Aufgaben gleichzeitig die Priorität, den verantwortlichen Entwickler oder die Einträge in der Verteilerliste ändern.

Bug Id	Subject	Assigned To	Date Modified	Status	Priority
<input type="checkbox"/> Edit 5	Visual glitch on screenshot	Manager	Oct 24 2003	Closed	3 - Medium
<input type="checkbox"/> Edit 2	Visual glitch on screenshot	Manager	Oct 20 2003	Closed	3 - Medium
<input type="checkbox"/> Edit 1	BugTrack test	admin	Oct 16 2003	Open	1 - Emergency

Summary:

	Open	Closed	Total
Manager	0	2	2
admin	1	0	1
Total:	1	2	3

Abbildung 5: Bulk Update [FASTBT]

4.6 Report erstellen

Einen einfachen und schnellen Report erstellt man z.B. über den Menüpunkt „*Reports / Quick Reports / AssignedTo*“. Dieser Quick Report liefert eine Liste mit allen Benutzern und den ihnen zugeordneten Aufgaben. Auf diese Weise gelangt man schnell zu wichtigen Informationen betreffend den offenen Fehlermeldungen. Reports können ebenso selbst gestaltet werden – Menüpunkt „*Custom Reports*“. Hier kann man die benötigten Daten einschränken, Überschriften und Sortierung der Spalten definieren.

Assigned To

Alexander

Bug Id	Subject	Date Modified	Status	Priority
5	PVSEExplorer Stammdaten lösche..	Nov 18 2004 (23:02)	Open	3 - Medium

Gerald

Bug Id	Subject	Date Modified	Status	Priority
5	TGViewer Telegramm zerlegen - ..	Nov 18 2004 (23:02)	Open	3 - Medium

Manuel

Bug Id	Subject	Date Modified	Status	Priority
1	XML File not valid	Nov 18 2004 (22:58)	Ready For Retest	3 - Medium
4	Max open Cursor - DB QESE	Nov 18 2004 (22:59)	Open	3 - Medium
1	Fehler E -801 bei Node Javeli ..	Nov 18 2004 (20:44)	Open	2 - High
1	Fehler eMail Verteiler	Nov 18 2004 (21:59)	Open	3 - Medium
3	Fenster läßt sich nicht schlie..	Nov 18 2004 (22:54)	Open	2 - High

Michael

Bug Id	Subject	Date Modified	Status	Priority
<u>3</u>	Speichern nicht möglich	Nov 18 2004 (22:59)	Open	1 - Emergency

Peter

Bug Id	Subject	Date Modified	Status	Priority
<u>2</u>	Search problem	Nov 18 2004 (22:56)	Open	4 - Low
<u>4</u>	Telegramm erfassen	Nov 18 2004 (23:00)	Open	2 - High

manuel.weidhofer

Bug Id	Subject	Date Modified	Status	Priority
<u>2</u>	LogDatei LogLevel setzen	Nov 18 2004 (22:57)	Open	3 - Medium

Abbildung 6: Report AssignedTo

4.7 Filter anwenden

Filter sind sehr nützlich um aus den erfassten Fehlermeldungen jene zu extrahieren die bestimmten Kriterien entsprechen. Man kann auf einfache Weise Queries erstellen, indem man über Eingabefeldern (z.B. für einen Suchbegriff, für die Bezeichnung einer Fehlermeldung) Suchbegriffe erfasst.

Filter Bugs User Filter Definitions: Empty

* System Filter Definitions: * Entered by CURRENT_USER [Edit] [Delete]

SearchString: - Search All Fields

Bug Id: = AND =

Subject: - Full Text Searching

Description: - Full Text Searching

Assigned To:

Alexander
 Gerald
 Manager
 Manuel

 - Multiple Select
 (use CTRL-LeftMouse to select/unselect)

Last Modified By:

Alexander
 Gerald
 Manager
 Manuel

 - Multiple Select
 (use CTRL-LeftMouse to select/unselect)

Entered By:

Michael
 Peter
 manuel.weidhofer
 [Current User]

 - Multiple Select
 (use CTRL-LeftMouse to select/unselect)

Status:

Closed
 Deferred
 Open
 Ready For Retest

 - Multiple Select
 (use CTRL-LeftMouse to select/unselect)

Project:

ApplInfoMonitor
 EAI Operator
 TGMManager
 TTI

 - Multiple Select
 (use CTRL-LeftMouse to select/unselect)

Environment:

entwicklung
 produktion
 simulation

 - Multiple Select
 (use CTRL-LeftMouse to select/unselect)

Abbildung 7: Filter

Weiters kann man die Ergebnismengen einschränken indem man aus Listen jene Werte wählt, die für einen interessant sind (z.B. alle Fehlermeldungen die den Benutzern Alexander und Gerald zugewiesen wurden).

Auf diese Weise erfasste Queries lassen sich speichern und sind somit für weitere Abfragen verfügbar.

4.8 Fazit

Fast Bug Track ist einfach zu handhaben und zu erlernen. Für die ersten Schritte ist nicht erforderlich in der Dokumentation nachzulesen. Die Menüführung ist gut strukturiert. Erst wenn man spezielle Funktionen verwenden möchte, z.B. die anonyme Erfassung von Fehlermeldungen via e-Mail Schnittstelle (*Anonymous Email Retrieval*) ist es ratsam die Dokumentation zu verwenden.

Beim Testen dieses Produktes sind keine Probleme aufgetreten. Ein einziger Punkt ist anzumerken. Beim Erstellen von einem Webformular zum anonymen Erfassen von Bugs (siehe Punkt 4.1.9) war es nicht möglich das Formular zu konfigurieren. Es wurde zwar ein leeres Formular angezeigt, jedoch keine Fehlermeldung an das System übermittelt werden. Möglicherweise ist dieses Feature in der Trial-Version nicht verfügbar, die Dokumentation lieferte auch keine weiteren Hinweise.

5 Abbildungsverzeichnis

Abbildung 1: Eingabemaske zum Erfassen von Fehlern	10
Abbildung 2: Balkendiagramm [FASTBT]	16
Abbildung 3: Bug erfassen.....	18
Abbildung 4: Liste erfasste Bugs	19
Abbildung 5: Bulk Update [FASTBT].....	20
Abbildung 6: Report AssignedTo	21
Abbildung 7: Filter.....	22

6 Literaturverzeichnis

- [BUGZ] Dokumentation zu Bugzilla. THE MOZILLA ORGANIZATION,
<http://www.bugzilla.org/docs/>, zuletzt abgerufen 18.11.04
- [BUGWG] Bugzilla Bugwriting Guidelines von Bugzilla Testinstallation,
<http://landfill.bugzilla.org/bugzilla-tip/page.cgi?id=bug-writing.html>
- [ENTW04] Gerhard Versteegen, Erhöhung der Softwarequalität durch
professionelles Bug Tracking, Der Entwickler; Online Artikel von April 2004;
http://www.derentwickler.de/itr/online_artikel/psecom.id.567,nodeid.56.html,
zuletzt abgerufen am 18.11.2004
- [FASTBT] Fast Bug Track, ALCEA TECHNOLOGIES INC,
<http://www.fastbugtrack.com>, zuletzt abgerufen am: 18.11.04
- [WIKIPE] Wikipedia – Die freie Enzyklopädie; <http://www.wikipedia.de>, zuletzt
abgerufen am: 18.11.04