

The Bugzilla Survey, August 2008

Max Kanat-Alexander

Assistant Project Lead, Bugzilla Project

Chief Engineer, Everything Solved, Inc.

Rev# 500 Revised: 9 Aug 2008 15:16 PST

Overview

I performed a survey of 23 major Bugzilla-using organizations, of which 19 said they would participate in the survey, and 14 actually responded in time for these results. (If the other organizations later send me their feedback, I will update the survey results accordingly.)

The survey contained the following questions:

- What version of Bugzilla are you currently running?
- What problems do you have with Bugzilla? (What are the most commonly experienced or reported problems, what are the most major problems?)
- What features of Bugzilla do you use the most at your organization, besides standard HTML bug searches, bug filing, and HTML bug viewing?
- If not answered above, are either of the New Charts or Old Charts reporting features of Bugzilla in common use at your organization?
- What customizations have you done to Bugzilla that you'd really like to see upstream?
- What features would you like to see Bugzilla have in the future?
- Are there any current features of Bugzilla that you'd like to see some change in, even if only a little? (Work differently, look different, fix some annoyance, etc.)
- Any other feedback that you'd like to give to the Bugzilla Project?

The questions were customized slightly for certain organizations (such as for ALMWorks, the makers of Deskzilla, where I changed the questions to be more focused on integrating with Bugzilla and less about using Bugzilla).

The survey was conducted directly by email, and the responders were the administrators of Bugzilla for that organization. Most organizations also got internal feedback which the administrator then sent to me.

Each email was specifically written to each organization individually, with different introductory text appropriate to the organization, to let them know that there was a real person conducting this survey and it wasn't just spam.

The advantages of doing this via direct email were:

- The administrators I contacted got the feeling that the Bugzilla Project cared about them.
- I was able to clarify any answers that needed clarification, by simply having an email conversation about them.
- People felt free to make their answers freeform and as long as necessary, as they were just typing a normal email.

Organizations Surveyed

Here's a brief list of the organizations that were surveyed, and a brief description of how they use Bugzilla, to help understand their feedback better.

Akamai

Akamai has a Bugzilla with several thousand products, representing various packages on systems that they administer. Akamai uses Bugzilla primarily for tracking in-house development and IT.

Bugopolis

Bugopolis is the largest Bugzilla-specific hosting company in the world.

ALMWorks

They make Deskzilla, a desktop client for Bugzilla. Most of their feedback is not included directly in the summary or analysis sections—instead I filed a tracking bug for everything they said, so that we can address it in future versions of Bugzilla. The bug is here:

https://bugzilla.mozilla.org/show_bug.cgi?id=449137

Eclipse

An open-source organization with a large, popular Bugzilla, that uses Bugzilla primarily to track open-source development.

GCC

A significant open-source project, with a medium-sized Bugzilla that they use to track open-source development.

Gentoo

A Linux distribution that uses Bugzilla to track development, with a large Bugzilla.

Mandriva

A Linux distribution that uses Bugzilla to track development, with a large Bugzilla.

Mozdev

A Bugzilla used to track development for many different projects. They have many, many products.

NASA

They have a heavily-customized Bugzilla that is not yet heavily-used, which they use to track problems in the software, hardware, and processes of building space shuttles, rockets, etc. They also have two other instances of Bugzilla used for other things.

Novell

A large Bugzilla used for both proprietary and open-source development.

Red Hat

A Linux distribution with a large Bugzilla that has over a thousand components in a few products.

VMWare

A large Bugzilla used for proprietary development and IT.

W3C

Several Bugzilla instances, used for tracking problems in the process of making specifications, and also for the W3C's software products.

Yahoo

The largest Bugzilla installation in the world, with several thousand products and over a million bugs.

How I Expect Bugzilla Will Use the Results

Primarily, I think that the Bugzilla Project will use these results to help track where future development should go, and also to get an idea of areas that organizations care the most about. Primarily, it's very useful to confirm that organizations care about certain things, when before we only *suspected* what their priorities were.

In future Bugzilla meetings or decisions about what should be on a roadmap, "The survey says..." is a valid argument, and should probably be our primary argument in making our roadmap. However, just because the survey requests a particular feature or reports a particular problem doesn't mean that we have to fix it in the exact way that organization wants. We should be the deciders of how to implement fixes and features, as we have to satisfy the broad needs of many different organizations, not just the specific ones that were surveyed here.

Organization of Results

First, I'll give a summary of the results, then I'll give an analysis, and then I'll include the actual raw feedback of all the responders as it was sent to me.

Summary of Results

The survey resulted in basically two types of feedback—a list of problems that people have with Bugzilla, and a list of features that they'd like to see in the future. The list of features further breaks down into features that the organization actually customized into their Bugzilla, and other features that are just wishlist features.

The descriptions of each item in each section are single lines that describe the feature or problem well-enough that a Bugzilla developer would understand what they are. Long-time Bugzilla users or administrators may understand what they are, too. If you need any clarifications, you can see the raw feedback of each organization toward the end of the survey.

Next to each description, in parentheses, is the name of the organization that had such feedback. Items are sorted, with the most-common problems/features at the top, and the least-common at the bottom.

NASA did actual user studies that resulted in some of their feedback, so that feedback is weighted higher than others and their name is bolded for that feedback. Also, Bugopolis had some feedback that represented many separate Bugzilla installations, so their feedback is weighted higher than others and their name is bolded.

Problems

These are problems that administrators and users experience with Bugzilla.

Items that multiple organizations reported

- We don't use op_sys and/or platform (GCC, Gentoo, Red Hat, W3C)
- Bug Filing is Difficult (Mozdev, NASA, W3C)
 - Too many fields (W3C)
 - Difficult to tell what fields are required (Mozdev)
 - Takes too many page loads (**NASA**)
- Advanced Search is confusing (Akamai, NASA)
 - Interaction between boolean charts and rest of search is confusing (Akamai)
 - “contains the string” etc. is confusing to users (**NASA**)
 - Difference in results between Quicksearch and Advanced Search is confusing (**NASA**)
- Many products is problematic for UI (Mozdev, Yahoo)
 - Particularly difficult to admin (Yahoo, Mozdev)
- Many components is problematic for performance (Red Hat, Yahoo)
- Group permissions model is confusing (Akamai, VMWare)
- Fulltext search doesn't scale [in 2.22, at least] (VMWare, Yahoo)

Items reported by only one organization

- Can't see all selected items in a multi-select without scrolling (**NASA**)
- The usemenuforusers “Add CC” list is confusing, people think it's the CC list (**NASA**)
- Tabular reports are too visually cluttered (**NASA**)
- Graphical reports are not very legible (**NASA**)

- Advanced Search doesn't scale to massive levels (Yahoo)
- Group permissions don't scale to massive levels (Yahoo)
- Bugmail is too slow (VMWare)
- UI has too many fields (W3C)
- Different products tend to use different keywords, or use the same ones differently (W3C)
- Some projects don't really use the version field (W3C)
- Various problems caused by our own customizations (Red Hat)
 - Didn't document our customizations well enough (Red Hat)
- Easy to DoS Bugzilla by doing lots of one-character searches (Eclipse)
- In IE, search page data is bad when clicking the “Back” button (Eclipse)
- Fulltext search for a space causes system error (Eclipse)
- post_bug.cgi title should contain bug summary (Eclipse)
- Times aren't displayed in the user's timezone (Novell)

Do you use the New or Old Charts?

This was a special category of feedback that needs its own section.

No (9 organizations: Mozdev, Red Hat, Akamai, GCC, NASA, Deskzilla, Yahoo, VMWare, W3C)

Yes (2 organizations: Eclipse, Mandriva)

Note that some organizations didn't answer the question, or couldn't give a definitive answer.

Several organizations said they would if the Charts didn't have various problems (not scaling, being hard to use, etc.)

Customizations

Done by multiple organizations

- Move standard fields to custom fields, or remove them (GCC, NASA)
- External bug references (Red Hat, NASA)

Done by single organizations

- [product] in email subject lines (Mozdev)
- Control flag visibility by group (Red Hat)
- Use AJAX to speed up loading of show_bug.cgi, query.cgi, and buglist.cgi (Red Hat)
- AJAX autocomplete username fields (VMWare)
- AJAX autocomplete milestones, keywords, dependson/blocks (VMWare)
- Spell-checking in IE (NASA)
- Made QA Contact not a “super” user for the bug (Mandriva)
- Customize DEFAULT_COLUMN_LIST (Mandriva)
- Have a user or domain for whom email is always disabled, in the code (W3C)

Future Features

These are the main features that organizations would like to see Bugzilla have in the future.

Reported by multiple organizations

- Field-level security (Eclipse, Mandriva, Gentoo)
- Extended XML-RPC (Red Hat, Akamai, GCC)
- Retire old components, etc. (Eclipse, Akamai)
- Separate login name from email address (Mozdev, Eclipse)
- Dashboard for logged-in users (Red Hat, **NASA**)
- Branches (**Bugopolis**)

Reported by single organizations

- Google-like spelling correction for search (**NASA**)
- Edit individual bugs in a buglist (NASA)
- Fulltext should search all fields (NASA)
- UI simplification (Red Hat)
- GPG email signing for inbound email (Mandriva)
- Restrict status transitions per group (Mandriva)
- Handle bounces (Eclipse)
- Make it easier to change the sort order of milestones (Eclipse)
- Edit comments (Eclipse)
- Detect duplicate bugs (Eclipse)
- Sortkeys for versions (Eclipse)
- Component watching (Eclipse)
- Better UI for buglist sort order (Eclipse)
- Paginate bug reports (Eclipse)
- Let Google index Bugzilla (Eclipse)
- Hide obsolete attachments by default (Eclipse)
- Custom per-component user fields (like assignee & qa contact are now) (VMWare)
- Ability to limit who can comment on a bug (Gentoo)
- Mass-changes with one or no emails (Akamai)
- Simple bug entry form (Mozdev)
- AJAX bug editing for some fields (Mozdev)
- If a product has no bugs, hide it on the Advanced Search page (Mozdev)
- Sync groups from LDAP to Bugzilla (Mozdev)
- Auth against another Bugzilla (Mozdev)
- Inbound email interaction without user having to have an account (Mozdev)
- Calendar widget for all date fields in Advanced Search (Akamai)
- Flags limitable per-version (instead of just by product/component) (Akamai)
- Make it easier to make simple changes to the search you just did (Akamai)
- Release notes management (Akamai)
- Custom status per-component (Akamai)
- Ability to ignore certain bugs or components [not get email for them] (Akamai)
- Quip control by a non-admin user (Akamai)
- Merge CC lists when a bug is marked a dup of another (GCC)
- Shorten buglist.cgi query string (W3C)
- Ability to assign a bug to nobody (W3C)
- Remember last time I visited, and let me see changes since then (W3C)

Miscellaneous Feedback

Some feedback doesn't fit into the categories above.

- Nearly every organization surveyed said “We love Bugzilla,” or “Great work!”
- We'd like to see faster release cycles (Red Hat)
- Most organizations use the latest or previous stable release of Bugzilla (Deskzilla)
- We love Boolean Charts, it's much better than what other bug-trackers offer for search (Deskzilla)

Analysis of Results

This section contains comments on trends and interesting aspects of the feedback.

As one might suspect, “The UI is problematic or confusing” was the most universal problem, no matter what type of organization was responding.

Sometimes organizations also report that their users think that the UI is very old-fashioned. For what it's worth, I'm not concerned about being *trendy*, but I am concerned with how *usable* the UI is, and only after that am I concerned with how pretty it is. I'm never concerned with how modern it is.

As we knew, Bugzilla's UI breaks down in some areas (particularly bug filing and administration) when there are more than about 30 products. Bugzilla also tends to perform poorly when you start to get into thousands of products or components. It would be a good idea to have automated performance tests on a test installation with thousands of products or thousands of components.

Search does have some performance problems when you get to around 400,000 or 500,000 bugs, though upgrading to a newer version of the underlying database might help. Changing your database setup (having a master/slave, multiple masters, etc.) can also improve the situation somewhat. Bugzilla 3.2 may help here, as prior versions of Bugzilla had to deal with table locks in MySQL when doing almost anything.

Organizations tend to have very different needs for their bug-trackers, and a lot of the features that they want are the features that let them implement their specific local policies or deal with the specifics of their community. Interestingly, the only feature that organizations consistently report that they want to *disable* is various built-in fields, most commonly the Operating System and Platform fields.

Probably the most commonly-used non-core feature (where “core” features would be basic bug filing, viewing, and searching) is flags.

Many organizations, after having a heavily-customized Bugzilla, switch to wanting to be as close to upstream as possible in their later upgrades. This is different than what we see at Everything Solved (that our clients want to maintain their customizations between versions). That is probably because organizations without a specific Bugzilla contractor or developer don't want to put in the resources to port forward customizations from version to version.

Raw Feedback

This section contains the raw feedback to each question, listed for each organization. Note that not every organization answered every question. Feedback is included exactly as it was written, with a few minor edits for clarity, and some nice formatting to make things easier to read.

ALMWorks (Deskzilla) had different questions. Where their questions differ from the normal ones, they have been included before the feedback.

Sometimes I had questions about their answers. My questions are in *italics*, with the responses immediately below. I also sometimes include notes from me on their feedback that look like this: [*ed: This is a note.*]

- **What version of Bugzilla are you currently running?**

- **Mozdev:** 3.0.2
- **Red Hat:** Currently running heavily customized 2.18. Will be running a customized version of 3.2 in the next month or two.
- **Eclipse:** Unmodified 3.0.4
- **Akamai:** 2.22
- **GCC:** 2.20, mainly because of the pain of upgrading our non-custom fields to the new custom fields. I'll get to it at some point, i'm more or less waiting to the point where i can just use the XMLRPC API to insert all the bugs again (I honestly hate perl, so i don't really want to write a converter in it).
- **NASA:** We're running 3.2 right now (about to release new version) [*ed: they mean a new internal release of an version*]
- **Mandriva:** 3.0.4
- **Gentoo:** BUGZILLA-2_22 branch, mostly from 2006/05/14, with a few bits from 2006/11/07, and backported other fixes.
- **Yahoo:** 2.22.1. Heavily modified at Yahoo.
- **VMWare:** 2.23.1 with a ton of customizations. we're moving to 3.x now though.
- **W3C:** We upgraded to the latest stable in June (with CVS) so I suppose 3.0.4).
- **ALMWorks**
What versions of Bugzilla do you most commonly integrate with?

When we started, I've done a research to find out which versions are in use. Back in 2004, that was from 2.14 to 2.18. We integrated with all of those versions, despite there were some major differences. Since then, we try to support each new version in the best way we can. I'd say that demand from our customers is mostly focused on the latest stable version and the one before the last.

- **What problems do you have with Bugzilla? (What are the most commonly experienced or reported problems, what are the most major problems?)**

- **Mozdev**

Too difficult to get new bugs into the system; too many choices/required fields; difficult to tell what fields are required for new bugs.

Having hundreds of projects set up as products in bugzilla clutters up the bugzilla interface and can make some things difficult, like doing a query or using some admin pages. One feature that could help us is to be able to hide inactive projects in the interface (since many projects we set up never end up using bugzilla). It would also be helpful to paginate the long product lists.

One solution to the cluttering problem I've seen on a different hosting site is to set up separate bugzilla instances for each project. This solved one problem, but then created another because you couldn't do searches across different projects. I'm not sure how many mozdev users do cross project searches, but I would imagine that it happens at least some of the time (many owners are involved in multiple projects and there are also several similar types of projects that may experience the same sorts of bugs).

- **Red Hat**

Alot of the problems we normally encounter can be traced back to things that Red Hat has added as customizations. The normal things that Bugzilla does out of the box work well. We (RH) need to provide better documentation for some of the features that we added. Currently education happens outside of Bugzilla using slide shows and meetings. One problem that could be considered a base Bugzilla problem is the lack of ability to query certain attributes over time from the query.cgi page. Also when using the boolean charts to query two values from the same table while using "And" instead of "Add another boolean chart" causes db errors. But these are things that Red Hat could/should fix given resources and not necessarily any fault of the community itself. Any queries not supported by the UI have been done using scripts outside of Bugzilla so the users are normally still able to get the information they need. Another issue we ran into in the 2.x series is the difficulty in adding new fields/statuses to the code base. But these problems will go away after the transition to 3.2 using custom fields and custom statuses.

- **Eclipse**

- Lagging slave DB returns stale searches, but we fixed that with internal scripts. Mediawiki, for instance, handles its slave clusters internally.

- Easy to launch Denial of Service attacks by doing lots of silly 1-character searches

- **Akamai**

Boolean charts and their interaction with the rest of the search page is confusing

Advanced search gets full list of all versions in the version pulldown (is this something we introduced?)

Different groups at Akamai have tried 4 times to use bugzilla as a project management tool and failed. It's tantalizingly close but not sufficient.

We can't allow all users to add keywords because when we do we end up with N different spellings of the same things. Maybe do keyword name-completion?

Clearer group permissions model. Both of our bugzilla admins have a very difficult time understanding how the permission model works.

- **GCC**

The most common problem for us is upgrading, because the set of basic fields does not quite match ours.

In particular, `op_sys` and `platform` mean nothing to us (since we use target triples), and even hiding them is a mild pain in the ass that requires a lot of templates be modified, which puts me into the hell of merging template changes between versions.

- **NASA**

In no particular order but based on 2 usability studies (run against 3.0):

- A more useful homepage that can be personalized to show the bugs the user cares about
- A usable search UI (e.g., our users are confused by "contains the string" and all other other options)
- A consistency between quicksearch and advanced search (users want to know why they don't produce the same results)
- Full text search
- More usable report output
- Spell checking both in the record (no they don't use firefox sadly)
- A google like "did you mean" for when they misspell words in their search criteria
- Better multi-select widget (one that allows you to see all the items that are selected w/o scrolling)
- CC/Add CC widgets are confusing. Most new users think that add CC list widget is the CC list (in 3.0) and wonder why it's so long.
- Streamline creating a new record so users can go directly to the form without a few page loads in the middle.
- Email should be more usable/legible. Assumes you're using a fixed width font and is just wonky.

This list does not include the long list of NASA-specific features we need to provide.

- **Mandriva**

The biggest problem is probably the email integration. It could definitely be improved, and it's something both our users and developers like to use very much.

Beyond that, the lack of finer ACLs is always a point of contention (i.e. we want to have users able to do as much as possible, but some users are pests and persist in re-opening bugs that they feel aren't addressed, even though they've been explicitly closed for various (legitimate) reasons). It would be nice to have ACLs prohibiting or

allowing specific actions, such as the ability to post bugs, the ability to reopen bugs, the ability to reassign, etc. While most of these could fall under the umbrella editbugs, it would be nice to have the ability to break ACLs down further.

- **Gentoo**

Fh::slice / editperms locking for users (both of which might be fixed in 3.0, but we haven't even finished a test deployment of bugzie3 yet).

*If you mean the problem that happens when two people both use editusers.cgi at once, that will be fixed in 3.2: https://bugzilla.mozilla.org/show_bug.cgi?id=354868
The Fh::slice thing certainly doesn't happen to me on 3.0, so we'll see if your experience of it goes away there.*

Nope. Fh::slice error that turns up sometimes during file upload. http://bugs.gentoo.org/show_bug.cgi?id=228043

Fh::slice on file upload is the most commonly reported actual bug, about once a month.

Slowness and latency about once or twice a week.

- **Yahoo**

This falls into three main areas. Things we've had to fix at Yahoo; things that may be addressed in later bugzilla.org releases, but where we're too forked to pick up those fixes; and things that just bug us.

- Fixed at Yahoo
 - Poor scaling in terms of performance.
 - db lock contention during search
 - search is very inefficient in general (bad joins and unions)
 - unusable text search
 - poor scaling in terms of UI
 - does not support thousands of products very well
 - poor devolved per-product admin functions
 - clunky UI
 - poor web services [*ed: There weren't any WebServices in 2.22.*]
 - poor email gateway [*ed: There was no official email gateway in 2.22.*]
 - poor bcp/ha
 - improved integration points with source code [*ed: There weren't code hooks in 2.22.*]
- Not fixed at Yahoo, but may be fixed at bugzilla.org (or may not)
 - custom bug lifecycle, on a per-product basis.
 - poor modperl support
 - graphs don't scale well to large datasets
- Other things that annoy users
 - email notification model is confusing
 - groups permissions does not scale.
 - we'd like a secure extra-net for bugs. [*ed: This means that they want an*

external Bugzilla that clones bugs that should be “public” from their internal Bugzilla.]

- poor use of screen real-estate.
- QA folks are second-class citizens. They have no specific workflow.

- **VMWare**

mysql's fulltext search doesn't scale. we ended up replacing it with sphinx.

I asked for more information about that, and they said:

So we're running 5.0.56. When we hit our scaling issues our db server had 4 2.8Ghz cpus and 8G of ram. We turned on slow querying and found that any query that did regex() or match() searches hung. We then moved to fulltext indexing for those columns (bugs.short_desc, longdescs.thetext, bugs.target_milestone) and found that mysql ignores all other keys when it uses a fulltext index. I wanted to cry.

We had to gut the fulltext searches out and pass them off to sphinx, which is really fast but it's a bit hacky imho (especially since end users can't query the sphinx index but we allow them read-only access to a copy of the db to run reports).

it's really difficult to add new fields (resolved in 3.x).

the ui is very 1999-ish.

Bugmail.pm doesn't scale very well (part of this is due to functionality we added though).

Hmm, what's the functionality you added, and what's the scaling problem you experience?

We allowed people to subscribe to products/categories/components/milestones/keywords so that someone working on a particular product will always get mail related to it regardless of the category/component the bug is filed under (this is a problem at our company because the same software component often appears in many different products). This resulted in an average of 80 emails being generated with a given bug change (but sometimes up to 200 emails!). Process_bug.cgi would hang making all of those sendmail calls. We switched to postfix but it didn't improve the situation, nor did the sendmail_now option.

I ended up changing bugmail to only generate unique emails for a given change (rather than a unique email for each recipient). Then everyone is added in the RCPT TO: mail line but the actual "To:" header just reads "bugzilla@vmware.com". This makes mail filtering a bit annoying but it sped up bugzilla drastically.

- **W3C**

[ed: Note that this feedback is from multiple people, which is why sometimes it seems to be talking to itself.]

One answer: People are reluctant to use Bugzilla, because the UI clutter (number of fields) looks intimidating.

I would agree with that statement. A lot of people I have talked to become rather agitated when the suggestion of using bugzilla arises, because its default UI is felt as "too scary". As a result we have built (from scratch) an issue and action-items tracker for our groups, which they find much more comfortable to work with, e.g:
<http://www.w3.org/Graphics/SVG/WG/track/>

Another person responded:

(a) mismatch between bug-tracking and spec-issue tracking (the presence of OS information, for example, confuses some commenters and WG members)

(b) different WGs use different keywords in the keyword field, but the keyword and their definitions are in a single common keyword pool, which confuses things. Easier customization of messages and templates at the product level would probably help one or both of these.

- **ALMWorks**

What problems do you have with Bugzilla? (What are the most commonly experienced or reported problems on the development side, what are the most major problems?)

Ok, here are some that come to my mind:

1. Bulk download not fully possible

It's not possible to download all information about a list of bugs in one request. We use xml.cgi to get XML view of a bug, but it misses a lot of information. To download the rest of the data, we have to load HTML page that user sees - 1 request per bug. Because of that we have to expose the user to a strange concept of "partially downloaded bug". Here's what we currently download through 1 request per bug:

- security groups
- comment privacy
- custom field info (name, description, type, order, list of options)
- available actions aka knobs
- list of voters (who voted => number of votes) -- this is loaded through another separate request

Some of this info may be available in XML in the latest versions, but often it's not enough. For example, XML shows security groups, but we also need some kind of id, which is present only on the main bug page ("bit-NNN").

We also will need to download Bug History and maybe Flags in a separate 1 request

per bug, when Deskzilla supports those features.

2. Metadata not exposed

We don't have any good way to download the structure of the database -- projects, components/versions/milestones, workflow, custom fields (with options), field settings, groups etc. We have to guess. For example, to learn dependency between projects and components/versions/milestones we have to parse JavaScript on query.cgi page. It would be perfect to have a single script that would expose the whole structure of the database (limited by the user's permissions), with all entities and dependencies.

Another problem is that some entities don't have identities. For example, we only see component's name and can use only that as an ID - which makes a problem if a component gets renamed.

3. Time format is volatile

Date/time values are reported in arbitrary format, in local timezone (usually without TZ marker). This has been causing much pain, especially if Bugzilla moves from one TZ to another. I'd suggest using standard XSD representation of time in XML form (times in GMT of course).

Another thing, I see Bugzilla 3.1 adds date/time custom fields and I expect problems uploading values to that fields for the same reason.

4. Updates not touching update time

To synchronize latest changes we download all bugs that have changed since X, where X is the synchronization point. Synchronization point is then moved forward. If any change to a bug does not affect its Update Time, we may miss this change. Currently only changes to Votes are known to bypass changing update time.

5. No upload tokens for failed upload detection

This is a tricky one. Suppose a user enters new bug in Deskzilla - it is stored in the local database and marked for upload. When Deskzilla uploads a bug, Bugzilla receives it but then connection fails. Bugzilla has stored the bug already, but Deskzilla thinks the upload failed. And it may try to upload again, which would result in duplicate bugs. In some cases where a problem after the upload was persistent, that ended up in a sort of endless cycle and as many identical bugs uploaded as the user let Deskzilla run this frenzy...

We now have protection against such cases, which is rather complex and involves heuristic search for what looks like a successful result of a previously attempted upload. It would have been much simpler if we could assign a voluntary "token" for each operation that involves creating a new entity (bug, comment, attachment), and in case server has processed a request with such token already, it would signal that the operation has completed before and gave a result.

- **What features of Bugzilla do you use the most at your organization, besides standard HTML bug searches, bug filing, and HTML bug viewing?**
 - **Mozdev:** None, yet.
 - **Red Hat:** The flags are used very heavily inside our organization. They are used by product/program management to support our business processes. We have customized our code to give flags the ability to be private since they are used for scheduling and we do not necessarily want everyone to know when a bug fix is slated to be released.
 - **Eclipse**
 - Shadow database. Lots of 'modern' web apps don't support using a DB slave. Bugzilla does, and that is worth gold.
 - RSS/XML feeds for rich clients such as Mylyn
 - Charts/reports
 - PatchReader/diff tool
 - **GCC:** Nothing, really, to be honest.
 - **NASA:** We probably use flags more heavily than other users.
 - **Mandriva:** I believe the reports are often used (although I've never really used them). I really wish the other people had responded as they make more use of these reports than I do.
 - **Gentoo**

Requests are disabled, votes are barely used since we enabled them.

Using whining was a complete failure - nobody liked it.

Hmm, what was the problem?

I think just with lots of bugs and developers deliberately ignoring them because they don't want to tackle them right now, it got to be too much. A lot of our bug assignments are also to mail aliases (teams), so that multiplies the mailouts dramatically.

Our quicksearch has some modifications that allow quick forms of some of the more advanced search, so get heavy usage.

Hmm, what modifications, in particular?

Starting a quicksearch query with some combination of keywords, status or resolution works (it used to be just status/reso).

Reports are quite popular as a feature.

The charts that track things over time, or just the tabular/graphical reports?

The tabular data is used to produce our newsletters and some other activity stats. I do see users hitting the rest of the graphs via the access logs.

- **Yahoo:** We use everything, except for groups permissions and graphing. Neither scales.
- **VMWare:** Not many standard ones. We developed our own reporting tools so the standard ones don't get used very much.
- **W3C:** One of our groups made extensive use of the XML export of bug lists at some point in time. I personally rely rather heavily on saved searches for my recurring work.
- **ALMWorks**
What features of Bugzilla do you use the most in Deskzilla besides standard HTML bug searches, bug filing, and bug viewing?

Here's a place to say kudos to you guys for the Boolean Charts search. Thanks, it is an awesome feature! :) Deskzilla allows to build arbitrary filters with ANDs, ORs and NOTs -- and we can translate them into a single request to Bugzilla using Boolean charts. Compare to JIRA, where search is very limited, and we might need to issue several requests to the server to get what we need, and sometimes we have to request more than really needed because it's not possible to ask for a specific thing.

We use search pages, query.cgi and describe*.cgi to learn about metadata, xml.cgi (that is, show_bug.cgi?ctype=xml) for bulk load, votes.cgi for loading votes, process_bug and enter_bug (the latter is also used to get defaults for projects). We used to use show_activity.cgi to reconstruct bug's history but it's disabled for now.

In general, we'd use any means to get or change the data.

- **If not answered above, are either of the New Charts or Old Charts reporting features of Bugzilla in common use at your organization?**
 - **Red Hat**
I am unaware of use of the stock Bugzilla reports by any groups. We have some third party tools that are written in house that allow custom generation of reports that are heavily dependent on our business processes. The third party reporting tools use either an internal read only database slave or the XMLRPC API to gather the data. Some tools even allow for updating of bug reports in mass using XMLRPC.
 - **Akamai:** No.
 - **GCC:** Nope.
 - **NASA:** No but if these features were usable and well maintained we'd use them because it would give us ability to do a certain kind of trending and process improvement.
 - **Mandriva:** I'm not sure what the distinction between "new" and "old" charts are, but I do believe they do use the charting features.

- **Yahoo:** Neither. See above.
- **VMWare:** neither. we have a custom app doing this.
- **W3C:** As far as I can tell, we hardly use the charts features.
- **ALMWorks:** No, we don't use charts. To build charts in Deskzilla, I guess we'd use data series built upon Deskzilla's local database.
- **What customizations have you done to Bugzilla that you'd really like to see upstream?**

- **Mozdev**

The outgoing emails have the product inserted into the subject line like "[product]".

- **Red Hat**

Enhanced XMLRPC functionality. External bug system references (with the possibility of changes propagating from one to another). Enhanced workflow functionality. Use of Ajax techniques to speed up page loading for show_bug.cgi, buglist.cgi and query.cgi for large data sets (we have ~6000 components in one product :()

- **Eclipse**

None. (We simply created a header and footer template, and created an index.php that features some top-10 links.)

- **GCC**

It would be ideal if op_sys and platform fields were custom fields and not assumed to exist by default. While gcc is mildly strange in that we can have host != build != target, it is not actually that uncommon for embedded vendors or others to have different host/target, and in those cases, "platform" and "operating system" don't make much sense without context.

I estimate roughly 90% of my bugzilla maintenance time is spent dealing with these default impedance mismatches.

Rather than attempt to track a bug for every single release separately, we have also added "Known to work" and "Known to fail" fields, which contain comma separated lists of valid product versions, and can be searched on.

This makes it very easy to find regressions and search for newly failing bugs without having to waste a lot of screen real estate and time checking 30 checkboxes that i often see on other bugzillae to represent regressions.

- **NASA**

LOL [*ed: NASA has probably the most-customized Bugzilla in the world.*]

One example is record linking between different instances. We'd like to see all fields be custom (no core fields). We'd like to control the order of all the fields show up in directly.

- **Mandriva**

Very few, actually. => We had a heavily customized bugzilla 2.x and when that user left, I was tasked with getting bugzilla 3.x up and running and of the stipulations I had was that it had to be as vanilla as possible.

Having said that, there are a few customizations I've had to implement. One of them is to drop the `op_sys` column for `buglist.cgi` since it's not something we care about (and it's hard-coded). Having the ability to define this via a template or configurable option would be nice.

I've also had to patch `User.pm` to prevent the `qacontact` from having "super" powers. For instance, if the `qacontact` is not in a group (for privacy), the `qacontact` shouldn't receive the mail. This is largely because we have the `qacontact` set to a mailing list and private bugs can't go to the mailing list. I think I filed a bug report about this and it was summarily rejected as the `qacontact` must get everything. This is fine, but breaks our privacy policy. So having a configurable option such as "constrain `qacontact` to basic group rules" or some such would be nice, to prevent the `qacontact` from getting bugs flagged private to a group that `qacontact` is not a member of (realistically, this could be a global option or constant because in my mind, if the `qacontact` should get those private mails, they should be part of that group to begin with).

The only other changes were to the template handling so we can have the original bug description appended to messages and to handle a custom field (we have a "RPM Package" field that we wanted to show up in the reports, but "sanitized" so that `foo-1.0.i586.rpm` would just show up as "foo" in the mail).

So, out of all of that, some way to customize the default column output without changing code and the `qacontact` stuff is what I'd *really* like to see upstream, from the changes we've made.

- **Gentoo**

Some customizations and tweaks to templating are our main stuff. Performance wise, we use the `shadowdb` in a LOT more places than the stock code.

The `shadowdb` changes are on almost every page, and also involve having concurrent shadow and non-shadow connections. `Bugzilla->dbh` was the original, and we added `Bugzilla->dbh_shadow` for use on some pages where we need to do both reads and writes. The old switch-to-shadow is used for read-only pages, of which a lot more were spotted where it can be used. [*ed: They can do this because they have a script that takes their slave out of rotation if it's lagging too much behind the master. Stock Bugzilla*

cannot safely read from one database while writing to another.]

relative URLs everywhere (important for SSL)

We should be doing that already, I thought?

r1.2 of Bugzilla/Search/Quicksearch.pm has some that were missed.

more useful headers in the bugmails

both SSL and non links in the bugmail

including the original filename in the displays about attachments

- **Yahoo**

We'd like to submit all our code back to the project, but we're stalled on our legal department's objections to the terms of the Mozilla Public License. *[ed: Note that I recently spoke to Yahoo again about this and their legal team is more concerned about trade secret issues than the MPL.]*

- **VMWare**

we've made all username fields ajax autocomplete fields. we've also done this with keywords, milestones, and depends/blocks.

we altered the search code to use an external fulltext search index; currently we've got ~2million entries in longdescs and mysql's fulltext searching refused to use any other keys in a query. we dealt with mysql's support on this one and determined it's a db bug. i don't know if anyone else has experienced this but i assume eventually they will.

our ui is much more "modern"; i'd at least like the standard ui to be less clunky.

Hmm, what did you modify?

This I can't really say. I'm not a UI person; I just know it had something to do with css. I'd be glad to send you some screenshots if you'd like. *[ed: I haven't received them yet, but when I do I'll link them here.]*

- **W3C**

Almost none. I think the only modifications we made were in 2001, when we modified templates to some extent, only to find out that some of the template work was not compatible with a later upgrade. That was long ago :)... and since then we've kept our instances very vanilla.

One very tiny hack that we've made (copying what mozilla did) was to create a user whose email basically goes to /dev/null and who is used to mark bugs NOT assigned to anyone. see e.g:

http://www.w3.org/Bugs/Public/show_bug.cgi?id=5225

Assigned To: This bug has no owner yet - up for the taking <dave.null@w3.org>

- **ALMWorks**

Are there any Bugzilla customizations that you've seen that you think would be helpful to have upstream (particularly if they're helpful to your ability to write the client)?

Quite the opposite -- serious customization might break Deskzilla integration with Bugzilla, and we have absolutely no interest supporting whatever people decide to add to Bugzilla -- unless it gains a lot of traction. We had potential customers who had custom fields in Bugzilla 2.18 (I think), provided by some patch, and we had to turn them down and advise migration to 3.0.

I've seen no customizations that would help us write the client. Even if it exists, we wouldn't rely upon it (but we could use it if it is present). Deskzilla by now requires no additional server-side installation (besides standard Bugzilla), and we'd like to keep it that way.

- **What features would you like to see Bugzilla have in the future?**

- **Mozdev**

Streamlined bug-entry form; ability to perform some "quick" actions through ajax to reduce interruptions in working on a bug (ex: changing assignee or CCs, bug status without reloading the page, then continuing to add a comment - these should probably be grouped together to be emailed out, though).

Separation of login name from email address.

Ability to auth against external databases (separation of login from email would go along way here). *[ed: We had a discussion about using the current Env auth method, and this might suit their needs.]*

Better email interaction (we haven't implemented this yet, but from looking at things the sender address needs to have an account setup already).

- **Red Hat:** See above.

- **Eclipse**

- removing/obfuscating email addresses (bug 218917)
- handling incoming/bounced email (bug 209181)
- better admin tools to prevent us from having to do SQL stuff on the command line:
 - change the sort order of target milestones
 - change versions/milestones across multiple projects
- improve/modernize the UI. Users want it to be pretty.

Plus those in the query below *[in the next question]*.

- **GCC**

Ideally, better interfaces for languages other than perl.

- **Akamai**

We know that bugzilla 3.x will solve some of the problems listed below but since we haven't moved there yet and thus can't be sure, we're giving the full list.

- Performance improvements
- Custom Fields
- Read access to local database (support already there?)
- Make flags version aware (not just component and element) so we can have difference flags be applicable for different versions.
- Better reporting around the time tracking features needed
- Ability to allow some users to create new components without giving them full editcomponents privs.
- Streamline search page
- Javascript calendar widget for picking dates would be nice
- Need APIs for everything, including bugs
- Like to be able to make modifying searches more convenient. E.g. The results page might have some kind of toggle to redisplay the same search with different bug state
- Release notes management
 - Roll up release notes from a list of bugs fixed
- Component hierarchy
 - Parent components need versions
 - Ability to organize components into product releases
- Have the header/footer appear twice (on top and bottom)
- Deprecate versions of components so they don't show up in some list (e.g. Creating a new bug)
- Improve workflow
 - Add customizable bug states per component?
- Better Keyword management
- Ability for users to make mass changes w/o email being sent (when someone leaves the company/changes role)
- Better email management
 - Allow me to turn off email by bug
 - Allow me to turn off email by component
- Better audit trail support
 - Would like to see a page of comments about all changes made to a bugzilla object
- Quip management by a user or users (who doesn't have admin privs)
- User and group hierarchy
- Better management of mailing lists
 - We currently create mailing lists, disable the user so no-one can log into the list, then flag it as a list so that the list can be an assignee. Very much a hack.

- **NASA**

We would like to have custom saved reports so that admins can develop reports based

on what fields particular user groups want to see.

- **Mandriva**

Support for GPG, which would solve the "make sure mail being posted to a bug as user A is actually from user A". Essentially, the user has the option to upload their GPG public key and associate it with their account. This would then allow them to send mail to bugzilla's mail interface. All messages need to be signed with that public key and any mail that is not signed or the signature is not valid, or does not match with what the user associated to their account, is rejected. This would prevent any mail spoofing or forging of comments.

That's probably the #1 feature I'd like to see added. I would have tried to implement this myself before, but my perl mojo is pretty weak and I just haven't had time to play with it.

I do think, however, that this would be a welcome feature to a *lot* of bugzilla installs, particularly those where privacy and security are really important.

- **Novell**

1) This one causes lots of heat and little light:

https://bugzilla.mozilla.org/show_bug.cgi?id=182238

Bug 182238 – Allow users to choose what time zone to display times in

2) This one is a moderately-important security problem:

https://bugzilla.mozilla.org/show_bug.cgi?id=26257

Bug 26257 - Bugzilla should prevent malicious webpages from making bugzilla users submit changes to bugs.

3) I believe Yahoo made some significant scalability fixes. I'd love to see those make it in.

What I would like on the list is a way to allow teams to administer their products without being able to administer all the products. In other words, I'd really like to eliminate using NPP [*ed: I assume this is some internal Novell tool.*] to configure Bugzilla's classifications, products, components, versions and milestones. What I like to be able to do is give 1 to n users the ability to change their product details without giving them rights to change all product details.

Some of the issues I have are already assigned to me and are just waiting for me to get working. For example, the hiding of inactive versions etc. One thing that is really often requested is around saved searches. Keeping sort order with a saved search or remembering which columns etc.

- **Gentoo**

More fine-grained control over bugs has been requested, eg making a bug (or certain fields only) read-only to most of the population (public or certain groups) while still

allowing some folk to comment on it.

Hmm. Do you want to be able to do that for specific bugs, or would being able to do it for specific products work? Specific components ideally.

Also, ability to lock some products/components to certain groups for entry entirely. *You can already do this for Products, was it critical to have it at the component level?* Yes. For two reasons: 1. Avoiding lots of spam during switching products on lots of bugs. 2. Need for lots of components due to the scope of the tree.

Demand for custom fields (eg package atom)

Automatic wrangling tools - ability to suggest an assignment based on any of the fields, and have the user or wrangler just review before agreeing to this.

- **Yahoo**

Bugzilla needs to be part of a bigger workflow. It needs to connect with source code, qa processes, packaging and deployment systems.

- **WMWare**

We're adding various other "ownerships" to bugzilla, ie a component has not only a assignee/qa contact but also a manager, sustaining contact, documentation contact, etc. i don't know how applicable this would be to the rest of the world but it's useful here.

- **W3C**

One of our participants noted:

- I'd like to see JavaScript for cleaning up the URI query string that query.cgi gives to buglist.cgi so that the query string only had parameters that narrow the query and aren't left to defaults. This way the URI would become shorter and would be nicer to paste into IRC or email.
- I'd like to see the content of the URL field linkified somewhere on the bug page. Now it is more convenient to paste URIs into the bug description.

My suggestions:

- I'd like a way for a product or component to be configured in such a way that new bugs are not linked to a given user by default. When next to "assigned to Bob", the "NEW" status is not making clear enough that the bug has not been assigned and that it is up for the taking.
- I'd really like a way for bugzilla to recall when was the last time I visited, and give me a way to search all bugs (in a project and/or assigned to me) modified since that last visit, without having to go through a complex search (I use a saved search for that, but maybe make it a saved search by default?)

- **Bugopolis**

The biggest thing that we hear a lot is support for branches. Basically a way to group the standard fields together where things like target milestones, versions, etc can be

tracked separately from the base bug.

Lots of people several releases in the field and a bug found common to some of those releases (or perhaps all) may get prioritized and fixed on different schedules.

Make sense? I think there is already one or two bugs for dealing with "branches".

- **Deskzilla**

Good remote API :) Basically, we're "transparent" -- our customers are your customers, so what they want from Bugzilla we want too. But we'd be utterly grateful if Bugzilla team thinks about integration with remote clients when adding a new feature. It boils down to answering several questions:

- How can remote client download bug's state [pertaining to that feature] (remember the bulk download!);
- How can remote client search bugs in a specific state (keep Boolean charts full with new predicates);
- How can remote client get all related metadata (names, ids, options, dependencies);
- How can remote client upload changes.

- **Are there any current features of Bugzilla that you'd like to see some change in, even if only a little? (Work differently, look different, fix some annoyance, etc.)**

- **Red Hat**

More use of Ajax for UI simplification. Bugzilla dashboard for logged in power users of Bugzilla to get a birds eye view of what their workload is.

- **Eclipse**

Those are mostly here:

https://bugzilla.mozilla.org/buglist.cgi?query_format=advanced&short_desc_type=allwordssubstr&short_desc=&product=Bugzilla&long_desc_type=substring&long_desc=&bug_file_loc_type=allwordssubstr&bug_file_loc=&status_whiteboard_type=allwordssubstr&status_whiteboard=&keywords_type=allwords&keywords=&bug_status=NEW&bug_status=ASSIGNED&bug_status=REOPENED&emailcc1=1&emailtype1=exact&email1=webmaster%40eclipse.org

- **GCC**

We've fixed some mild annoyances that mozilla doesn't necessarily consider "bugs" but everyone else does (merging cc lists when bugs are marked as duplicates, for examples). I can produce a complete list if forced. :)

- **NASA**

I want to be able to edit multiple records in the buglist INDIVIDUALLY. For example, I'd like update priorities for a bunch of different bugs (e.g., change bug A from P3 to P4 and bug B from P2 to P1) in one place. I'd like to be able to select in the admin UI which fields can be modified on the buglist page. This not for the NASA problem reporting usage for our internal usage as a development team.

- **Mandriva**

Out of the current features, I'm pretty happy with what Bugzilla provides. I can't think of anything off the top of my head I'd want to see changed.

- **Yahoo**

The UI is ghastly, and needs professional redesign.

- **W3C**

Feedback from one of our participants:

- I think Bugzilla has too many fields by default:
 - The version field is pointless for projects where bugs are virtually always reported against the trunk.
 - The severity field has 7 levels. Wouldn't 5 or 3 be enough?
 - The priority and target milestone fields can easily remain unused in projects that don't have a huge number of concurrently open bugs.
 - The concept of QA contact isn't applicable to all Bugzilla users, and probably shouldn't be there by default.
 - The Platform and OS fields are mostly useless for tracking spec bugs or bugs in online services. Also, the default values for these fields are seriously out of date.

I would actually disagree with the part about the QA contact - the title itself may be confusing, but I do find that role incredibly useful.

- **ALMWorks**

From the point of Deskzilla integration -- absolutely not, because we'd have to adjust our integration part :) From the point of an ordinary user -- well, you know it all, I guess. Administrative screens are awful. We don't have administrative tasks in Deskzilla, so I have to deal with that on the web interface :)

- **Any other feedback that you'd like to give to the Bugzilla Project?**

Note: In this section I've also included various other pieces of information that came up during my email conversations with these various organizations. My questions are in italics, their answers are in plain text.

- **Mozdev**

None that I can think of immediately.

- **Red Hat**

We have nothing but positive opinions of the Bugzilla community in general. We rely heavily on the features that Bugzilla offers and like the fact they we are not locked into any commercial bug tracking product. Faster release cycles would be nice but we understand that it is based on community of volunteers with regular day jobs where Bugzilla is not there first priority. Red Hat faces the same issues with projects we work on here as well.

- **Eclipse**

Bugzilla is great. We really like it.

- **NASA**

Keep up the good work.

- **Mandriva**

Well, kudos on 3.x. =) With 2.x, I found it quite painful and actually wrote my own bug tracker years ago in PHP (Anthill). With 3.x, I think Bugzilla has done right the things I think it did wrong in 2.x that had prompted me to write my own. Anthill hasn't been developed in quite a few years as a result. =)

- **Gentoo**

Gentoo provided a history of their hardware setup:

As of sometime later this week, just waiting for an IP allocation from a sponsor, the Gentoo bugzilla will be running a fully redundant system.

History has been:

1. All on one box.
2. 1x Web box, plus the OSL shared db.
3. 1x Web box, plus a dedicated DB.
4. 1x Web box, plus master-master DBs.
5. 1x Web box, plus master-master DBs, plus 2x read-only slaves (running on same machines as masters)
6. 2x Web, 2x master, 2x slave (same boxes still)

I'm curious--why the switch to 2x Web? Was it a performance issue or a redundancy issue? Both. There are some intensive graphing operations we experience on the web side at times, and we'd like to balance that load better.

- **Yahoo**

Great work guys!

- **W3C**

Thanks for the opportunity to give some feedback - much appreciated.

W3C has separate Bugzillas for separate projects, with different levels of security. I asked them: what prompted you to have separate Bugzillas instead of using the groups system within Bugzilla itself?

I think the simple answer is that access control was easier. We have hundreds of users in (basically) 3 different access groups: the staff, members (and invited experts) and the public. Rather than having to manage individual accounts in bugzilla, add them to groups, manage groups, manage permissions in bugzilla's interface... it was easier to have three instances, with which http authentication would make sure you are (or not) allowed to use.

- **ALMWorks**

Thanks a lot for the product! We owe you :) By the way, if I may suggest, why don't you replace ant image in the default template with ant logo that Bugzilla.org shows now? People usually dislike insects, don't they? - I guess that's why you removed ant from Bugzilla.org...