Bug tracking system - report

Authors:
Kenneth Aarseth
Ivar H. Aksnes
Erlend S. Ervik
Bjørn Kato Sjøstrand
Thomas A.O. Steen
Per Daniel Sørensen

Products:
Assembla
Bugzilla
FogBugz
Gemini
JIRA

Introduction

We have assessed a list of 48 tools for bug tracking. We narrowed these 48 down to 5 candidates which fill the criteria we had for the project. We also considered the usefulness of additional functionality.

Glossary

The following terms are being used in this document.

Term	Explanation
Classification	Be able to classify an incident report as a fault, functionality or a question.
Source Control	Common term for systems that handle different source code versions.
Wiki	Tool for documentation, with the ability to link between articles and terms.
RSS(-feed)	A tool used to subscribe to content. Any updates to the RSS-feed will then show up in the RSS-reader of your choice.
Case splitting	The ability to split cases into sub-cases, or a hierarchy of cases if a case relies on another case to be solved first.
Open Source	A system where the source code is publicly available.
Open system	A system where there is no protection for access from strangers, or a system where such protection is limited.
Clients/external users/customers	People who can submit incidents, and possibly track the handling process through a web page or e-mail.
Users	People who are registered on the bug tracking system, and have rights to use it.
Case/message/incident	A term that describes the submission of an error message, questions or otherwise.
Tool/solution	The product in question.

Functionality and criteria

Our focus has been set on a series of functions which are considered as required and useful. An explanation of the functionalities below can be found under Terms for bug tracking functions.

The following is an overview of functionalities we consider to be definitive requirements.

Active user base	User groups	Case splitting
Dedicate cases to users	Classify cases	Easy web interface
Reports	User manual	File upload
Create incidents via email	Discussions (private/public)	No requirement for user registration

The following is an overview of the functionality we consider useful.

Wiki	Source Control
Time estimation	RSS-feed support

The following is an overview over the expense criteria.

Buying and licence expenses	Company hosts itself/external hosting	User support
Active development	Implementation language	

Functionality	Assembla	Bugzilla	FogBugz	Gemini	JIRA
Web reporting	Yes	Yes	Yes	Yes	Yes
Email reporting	Yes	No	Yes	No	Yes
Discussions(private)	Yes	No	Yes	Yes	Yes
Discussions (public)	Yes	Yes	Yes	Yes	Yes
Case splitting	Yes	Yes	Yes	Yes	Yes
Case classifications	Yes	Yes	Yes	Yes	Yes
User manual	Poor	Poor	Good	Good	Adequate
File upload	Yes	Yes	Yes	Yes	Yes
Requires user registration	Yes	Yes	No	Yes	Yes
User groups	Partial	Yes	Yes	Yes	Yes
Supports reporting	Yes	Yes	Yes	Yes	Yes
Wiki	Yes	No	Yes	No	Yes
Time estimation	Yes	Yes	Yes	Yes	Yes
Source Control	Yes	Yes	Yes	Yes	Yes
RSS-support	Yes	Yes	Yes	Yes	Yes
Active development	Yes	Yes	Yes	Yes	Yes
Company hosts itself	Yes	Yes	Yes	Yes	Yes
Is hosted by developer	Yes	No	Yes	Yes	Yes

API-support	Yes	Yes	Yes	No	Yes
Implementation language	Ruby on Rails	Perl	ASP/PHP	ASP.NET	Java

Bugzilla

Expenses

Bugzilla must be hosted by the company, and this, along with maintenance work (updates, patches, security fixes and similar) will impact expenses. These are areas we do not have information available in order to work out a price quote.

Experience from use

Source of experience

We installed an “all-in-one”-package with the essentials to run Bugzilla for Windows.

Experiences

Bugzilla is a free Open Source tool, targeted towards developers. This is reflected through the use of the system. The solution is missing a couple of practical tools that could have made routine work simpler and faster, in comparison to the commercial solutions. The solution is also limited to incident reporting.

It is an open system, which means it is not possible to keep cases confidential. In a configuration where the server is available publically on the internet, anyone can access and view the contents. Bugzilla is best suited for Open Source projects where privacy is not a requirement.

User registration is required in order to file an incident report. This can be an obstacle for some users, because it requires extra time and work to create and remember user accounts.

JIRA

Expenses

This product can be upgraded with more functionality if you pay for this additional functionality.

For 25 users where JIRA hosts the solution costs $300 USD each month. This solution covers all the criteria we have set, except for Source Control. If you wish to host the solution yourself, you will need to pay a $1200 USD one-time fee.

For Wiki and Source Control support you need to pay for a full package, and for 25 users that will be $500 USD each month.

Experience from use

Source of experience

We registered a user account at https://issues.apache.org/jira. The user manual for JIRA was used as reference: http://www.atlassian.com/software/jira/docs/v3.13/administration.html

Experiences

JIRA is a system that is designed for internal use in the company, but it can also allow public access. In order for users to add incident reports, they need to be registered. If user registration is disabled, administrator has to manually create user accounts and set proper access rights.

If user registration is active, the administrator still has to set proper access rights.

Adding incident reports through e-mail can be tedious to configure, but instructions can be found in the user manual: http://www.atlassian.com/software/jira/docs/v3.13/issue_creation_email.html

Gemini

Expenses

Gemini has 2 solutions. Both allow unlimited users and projects, “add-ons” and integration products against other systems. In addition you get 12 months of support and upgrades.

If you host the solution yourself, it will cost $1199 USD. It includes all the features as listed above.

If the solution is hosted by Gemini, it will cost $2199 USD. Gemini will then take care of maintenance and daily backup.

The solution is developed with ASP.Net, and licence for Microsoft Windows Server and Microsoft SQL Server is required in order to host the solution yourself.

Experience from use

Source of experience

A demo from Gemini: http://gemini.countersoft.com/

Experiences

The user interface is somewhat untraditional compared to the other solutions. There is no clear structure on how things are connected.

It is impossible to send in incident reports through e-mail from what we saw in the demo. Also, a user account is required to create an incident report.

As with Bugzilla, the solution is targeted primarily towards developers, although Gemini has more polished features than Bugzilla.

FogBugz

Expenses

FogBugz “On Demand” is a product where FogBugz hosts the solution. This costs $25 USD for one user each month, up to 23 users. Beyond 23 users, there is a discount. For 15 users, it will cost $375 USD each month.

You will **not** be billed for users that are **not** active during the billing month. That means you only pay for active users in a billing month.

If you want to host the solution yourself, the price for 15 user licenses will be $2898 USD. In addition, 1 year maintenance for those 15 users will cost $547.50 USD, which is a total of $3445.50 USD. With maintenance you get updates and new versions of FogBugz, unlimited phone support and questions through e-mail are answered within 1 business day. You also get installation support where a technician helps you with the installation through remote assistance. This maintenance package is called FogBugz Platinum Maintenance.

The information was retrieved after a request was directed to FogBugz.

Experience from use

Source of Experience

A time-limited test account was created. Anyone can register a time-limited test account at the bottom of http://www.fogbugz.com/. The time-limited account lasts 45 days. You can continue use of the solution after buying a licence.

Experiences

With a solution hosted by FogBugz, you will receive an URL in the format http://<name>.fogbugz.com/.

The user interface is practical and works well when you get used to it.

FogBugz is the solution that makes it easiest to communicate with clients. They do not need to register to send questions or file incident reports. They just send an e-mail to cases@<name>.fogbugz.com, or to a custom e-mail address. FogBugz will retrieve e-mails after you have configured the e-mail server setup. This is an easy task, and there are a lot of settings you can change, like auto respond, template for responses, and even add time estimation on new cases.

When a case is received via e-mail, you can respond to the sender. This will then notify the sender through e-mail.You can then file a new incident report if the sender has reported an incident, in which you can discuss the problem. This helps keeping client communication and internal communication separated.

Assembla

Expenses

The package solution called “Group” includes 40 users, 10 “Spaces” (projects) and storage capacity of 5GB. It costs $49 USD each month.

For 40 users, 1 “Space” and 2GB storage space, the price will be $24 USD each month.

If you wish to host the solution yourself, the price will be $200 USD each year for every user. User accounts that have limited access, like reading or commenting, are free.

Experience from use

Source of experience

A time-limited test account was created.

Experiences

The solution was hosted by Assembla, and when you create an account, you will receive an URL in the format of http://<name>.assembla.com/.

The user interface is well organized and easy to work with.

One problem with this solution is that it is difficult to invite external people (like customers) to the solution, along with a requirement that users are registered.

To allow external people to follow a case they have reported, they must register a user account with Assembla, Yahoo or Google. This invitation must be done manually.

It is not uncommon for people to have several user accounts. It can be troublesome keeping track of all these. For a person that only wants to submit one incident, this could be an unreasonably high threshold for participation.

Not all companies allow use of other e-mail accounts than the ones the company designates them, and in those cases using Yahoo or Google to log is no longer an option.

In order to participate in discussions, you need a user account with Assembla. Additionally access rights must be manually set. This can be a straining task. In a situation where the contact person from customers may be changed regularly, you would need to spend a good deal of time updating access rights.

Filing incident reports via e-mail also takes a while (5-10 minutes). This could be problematic if you are waiting for a report, or waiting for an update from a user/customer you are having a chat with.

There are several small things that make administration of Assembla easier. Initially, finding these settings can be a challenge, but that will improve as you get used to it.

Terms for bug tracking functions

Criteria for the product

Active user base

If the product is widely used, means it is easier to find solutions, if any problems should arise.

User groups

Developers, testers and customers have different needs in regard to what functionality they need access to. With the ability to put them into groups, you can limit what users can do. Customers could for example be limited to only see the case or incident report they have filed.

Split cases

With the ability to split cases, you can make sure that cases can be solved separately. If a problem is the result of another problem, you can create a sub case of the original case. This will make it more apparent and easier to fix when you can see a hierarchy on the cases.

Dedicate cases to a user

With the ability to dedicate a case to a user (tester/developer), you can avoid having several people working on a solution for the same problem.

Classify cases

With the ability to classify cases, one can easier sort through cases, depending on severity, type, priority and so on, you can dedicate the resources you have available where it is needed.

Easy web interface

For the sake of the customer, it must be easy to follow a case from a position outside the development office. For the company, it must be easy to communicate with customers that have filed incidents, in order to retrieve more information.

Reports

Reports that show number of cases which are open, closed, completed, monthly, corrected and similar. For users it is useful to see which cases you have been given.

User manual

There must be an understandable help system for the users of the solution, where there are examples and step-by-step guides to issues one needs to solve.

File upload

It must be possible to upload images and other files.

Filing of incidents

It must be possible to send an e-mail to a specified e-mail account, which is then automatically added to the system. It must also be possible to file an incident through a form online. (Web page)

Useful functionality

Time estimation

The option to set a deadline or come up with a time estimate is useful, because it sets demands to the developers and other users for the system.

RSS-support

It is practical to be able to subscribe to cases and get updates through your web browser without having to check e-mail.

Discussions/Wiki

Instead of having a decentralised system (1 for bug tracking, 1 forum, 1 wiki and so on), it could be beneficial that the solution supports discussions. Discussion threads should be possible to set as hidden or visible to the public (customers).

API-support (Application Programming Interface)

With API-support, you can develop systems that can be closely integrated with the solution. For example automatic filing of an issue if the project no longer can be compiled.