

Introducción a los Sistemas de Bugtrack

phpBugtrack Mantis Bugzilla
FastBugtrack bug release
milestone version
component



Introducción a los Sistemas de Bugtrack

► Introducción

Ponentes



Manuel Resinas de Reyna

Investigador miembro de TDG (The Distributed Group) y estudiante de doctorado



Manuel J. Recena Soto

Alumno de la Escuela Técnica Superior de Ingeniería Informática (ETSII)

► Introducción: Presentación del seminario

- Motivado por la carencia, en los planes de estudios actuales, de contenidos relacionados con estos sistemas.
- Como sucesor del seminario “Introducción a los Sistemas de Control de Versiones” y perteneciente a un objetivo personal:
 - Aportar una visión práctica, basada en la experiencia, de un proceso de desarrollo y del software que nos permite el mantenimiento de proyectos (software) y la coordinación de recursos humanos pertenecientes a grupos de desarrollo.

► Introducción: Objetivos del seminario

- Dar una visión general sobre los Sistemas de bugtrack.
- Acercar los conceptos: release, component, milestone, bug, etc...
- Explicar el ciclo de vida de un bug.
- Realizar una breve pero concisa evaluación sobre algunos de los Sistemas de Bugtrack más conocidos por las comunidades de software libre a excepción de *FastBugTrack* (comercial).
- Beneficios de trabajar con estos sistemas.
- Su papel en los distintas fases de un proyecto software: desarrollo, preproducción y producción.
- Inconvenientes.

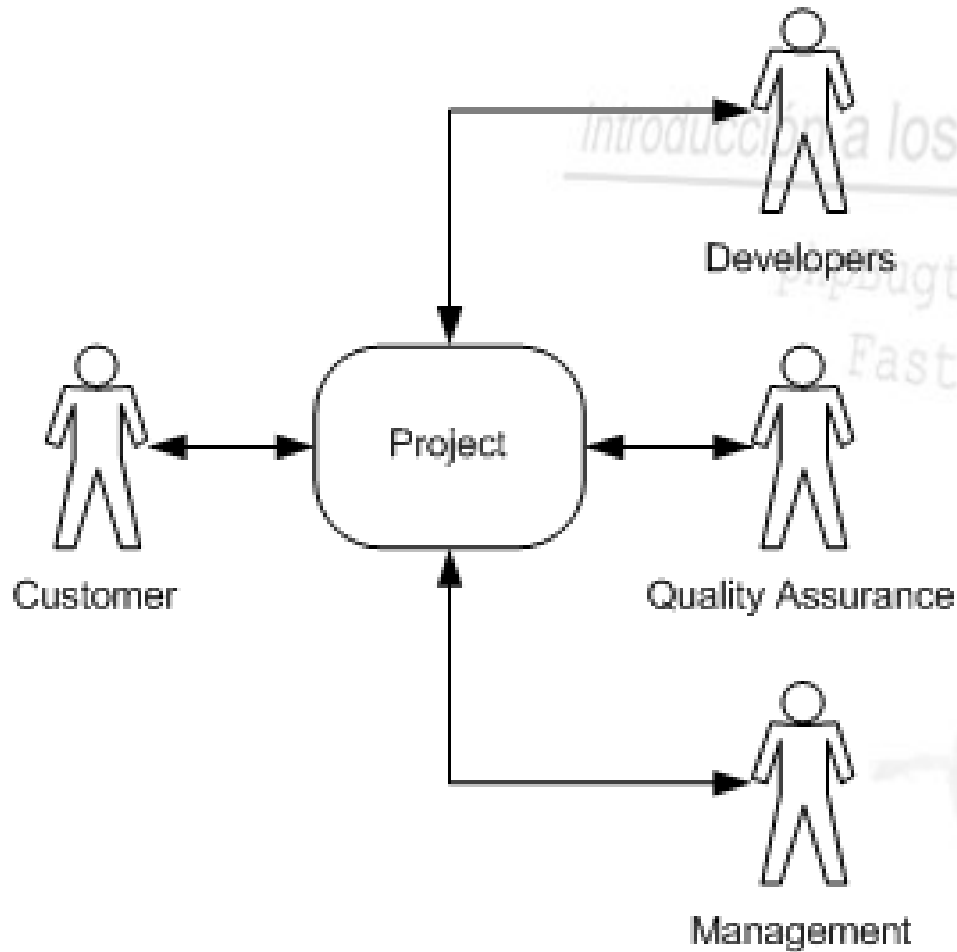
► El problema

- Cómo notificar al grupo de desarrollo un bug.
- Cómo realizar un seguimiento de un bug.
- Cómo distribuir la resolución de bugs entre los recursos del proyecto.
- Los proyectos de software crecen:
 - Mayor cantidad de código fuente → mayor número de bugs
 - Mayor cantidad de funcionalidades → mayor número de bugs
- Los bugs realmente importantes son los encontrados por los usuarios finales de nuestro software.
- La figura del QA (Quality Assurance).

► ¿Qué es un Sistema de Bugtrack?

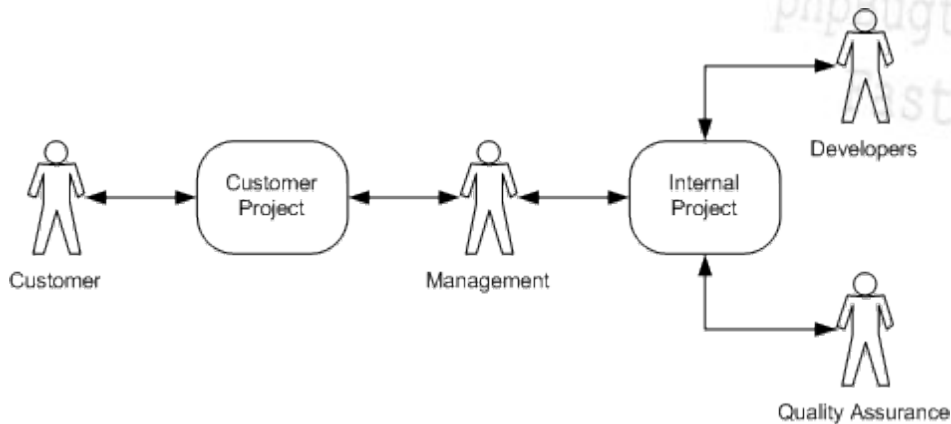
- Permite notificar *bugs* de forma centralizada.
- Establece un *workflow*, basado en estados, a partir de los bugs.
- Proporciona un sistema de notificación (rss, emails, etc...).
- Ayuda a establecer el *roadmap* basado en los bugs de nuestro proyecto.
- Proporciona un histórico de bugs, milestone y release de nuestro proyecto.
- Es una herramienta muy importante de trabajo para el QA (Quality Assurance).
- Permite conseguir un feedback entre el usuario final y el grupo de desarrollo.

► Modelo (1)



► Modelo (2)

Introducción a los Sistemas de Bugtrack

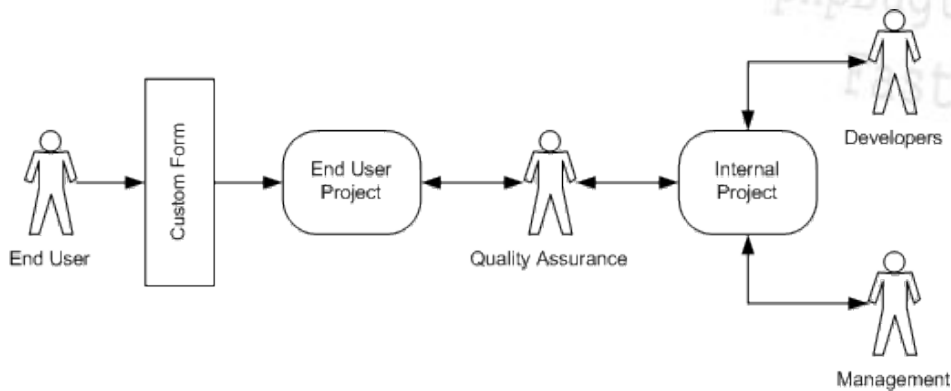


phpBugtrack Mantis Bugzilla
stBugtrack bug release
milestone version
component



► Modelo (3)

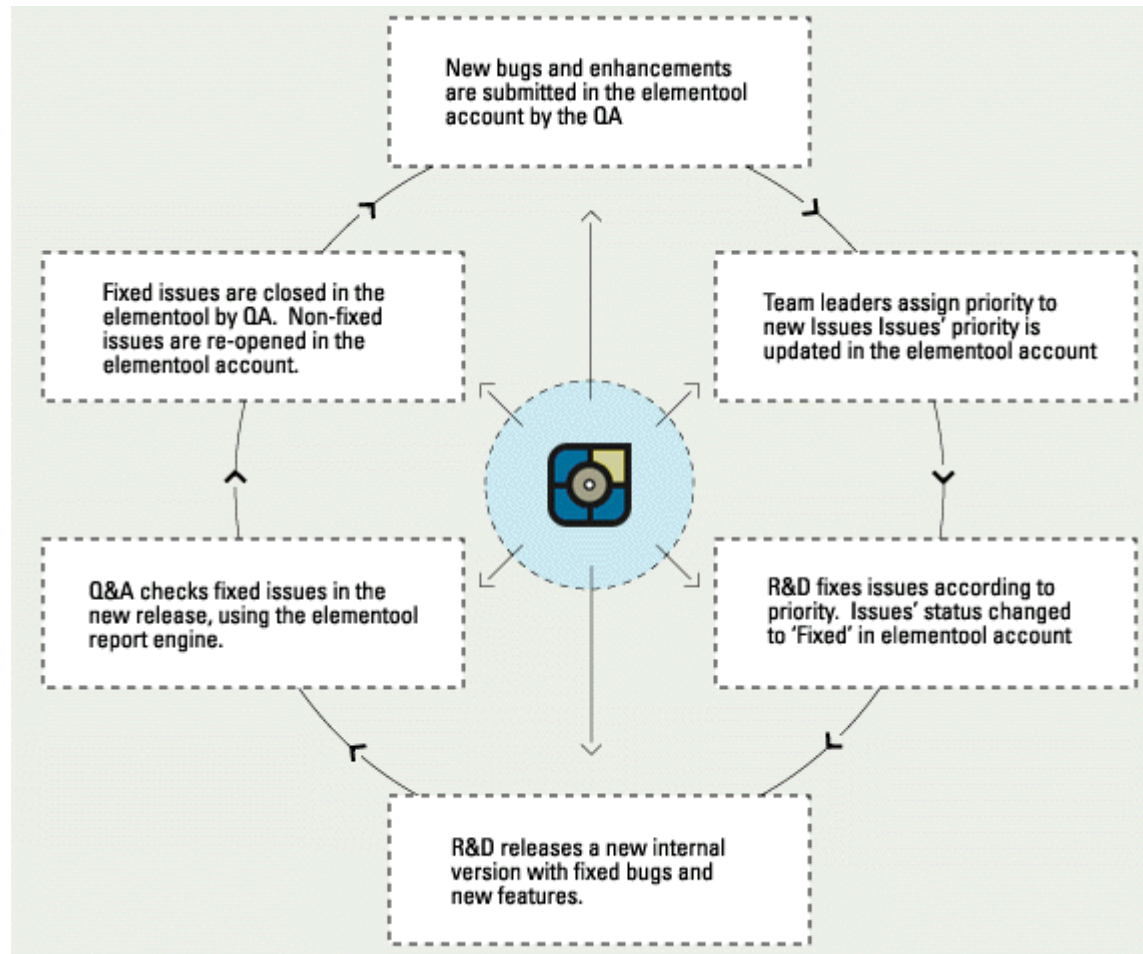
Introducción a los Sistemas de Bugtrack



phpBugtrack Mantis Bugzilla
phpBugtrack bug release
milestone version
component



► Workflow



Esquema obtenido de <http://www.developer.com>

► Algunas recomendaciones

- Un buen *tester* debe intentar reducir al máximo el número de pasos necesarios para reproducir el *bug*.
- La única persona que puede cerrar un *bug* es la persona que lo abrió en primer lugar. Cualquiera puede resolver, pero sólo la persona que lo abrió, puede estar segura de que está corregido.
- Toda versión del *software* que se le da a los testers debe tener el número de versión para que el *tester* no tenga que probar el bug en una versión en la que puede que no esté corregido.
- Evita añadir nuevos campos al sistema de *bugtrack* para no complicar el proceso de añadir y modificar *bugs*.
- Motivar a *testers* y programadores para usar el sistema de *bugtrack*.

► Comparativa

- Hemos elegido cuatro sistemas de *bug tracking* para comparar sus características.
- Nos centramos en *software* de código abierto:
 - BugZilla (<http://www.bugzilla.org>)
 - Mantis (<http://www.mantisbt.org>)
 - Scarab (<http://scarab.tigris.org>)
- Como alternativa comercial:
 - Fast BugTrack (<http://www.fastbugtrack.com>)
- Hay muchos más.

► Comparativa: Requisitos sistema

	Bugzilla	Mantis	FastBugtrack	Scarab
Arquitectura del sistema	Web-based	Web-based	Web-based	Web-based
Base de Datos	Mysql	Mysql	Ficheros	Mysql, PostgreSQL, Hypersonic
Servidor Web	Apache/IIS	Apache/IIS	Propio	Tomcat
Lenguaje Programación	Perl	PHP	Java	Java

► Comparativa: Varios

	Bugzilla	Mantis	FastBugtrack	Scarab
Importación/Exportación	No	No	Si (CSV, XML)	Si (XML)
RSS/Otros	Si	Si	No	
Relaciones entre bugs	Si (depende/bloquea)	Si (padre, hijo, duplicado)	No	Si (bloqueante, duplicante, no- bloqueante)
Descomposición	Producto, Componente, Versión	Proyecto, Subproyecto, Versión	Proyecto, Subproyectos, Entornos	Módulos, submódulos...



► Comparativa: Personalización

	Bugzilla	Mantis	FastBugtrack	Scarab
Campos personalizados	Sólo flags	Si	Si	Si
Aspecto visual	Si, por medio de plantillas o hooks	Menús/CSS	Si	?
Formularios	No	Si (mostrar, ocultar)	Si (mostrar, ocultar)	Plantillas
Workflow	Fijo	Cambiar transiciones válidas y quién puede realizarlas	Si (transiciones válidas, permisos, campos visibles sólo a partir de un estado)	Totalmente configurable
Extensiones	No	Si, mediante funciones	No	Si

► Comparativa: Seguridad

	Bugzilla	Mantis	FastBugtrack	Scarab
Mecanismos de autenticación	BD/LDAP	BD/LDAP	Propio	BD
Permisos	Editando código y usando grupos de usuarios (sólo de escritura, todos pueden leer)	Cambiando los thresholds de cada rol (editar fichero)	Fijos	Asignables a cada rol
Roles	No tiene	Fijos (modificación por código)	Read-only, Normal, Administrator	Configurables (añadir, modificar...)
Usuarios en un proyecto concreto	No	Si	Si, por medio de los grupos	~ (usando un rol sin permisos)
Proyectos privados	No	Si	Si	Idem

► Comparativa: Otros

	Bugzilla	Mantis	FastBugtrack	Scarab
Filtros/Búsquedas	Si	Si	Si	Si
CVS	Vía correos electrónicos (en el subject del mensaje poner el # de bug)	Parcial, a través de script del CVS	No	No
Informes	Si, informes y evolución en el tiempo (sólo filtro, no columnas)	Si (sólo filtro, no contenido)	Si (filtro y contenido)	Si, completamente personalizables
Gráficas	Si	Si	Si	Si
Notificación	Email	Email, muy configurable, sólo los de una severidad X, seleccionar bugs para monitorizar	Email, listas de notificación por bug	Email, notificación por bug

► Conclusiones

- Grandes proyectos de la comunidad de software libre son gestionados con sistemas de bugtrack.
- Centraliza en una única herramienta la casi la totalidad del trabajo del QA.
- Estos sistemas son necesarios independientemente de los recursos humanos de los que esté compuesto el grupo de trabajo. Uno de los factores que condiciona su uso es el crecimiento del proyecto.
- El uso de estos sistemas requiere habituarse a una metodología que conlleva un periodo de adaptación.

► Referencias

→ Bug Tracking Made Simple:

http://www.developer.com/mgmt/article.php/11085_3389021_1

→ Painless Bug Tracking: <http://www.joelonsoftware.com/articles/fog0000000029.html>

→ Mantis: <http://www.mantisbt.org>

→ FastBugTrack: <http://www.fastbugtrack.com>

→ Bugzilla: <http://www.bugzilla.org>

→ BUGTrack: <http://www.ebugtrack.com>

→ Project Management Software: <http://www.project-management-software.org>



Introducción a los Sistemas de Bugtrack

► ¿Alguna pregunta?

Introducción a los Sistemas de Bugtrack

phpBugtrack Mantis Bugzilla
FastBugtrack bug release
component

¿Alguna pregunta?



Introducción a los Sistemas de Bugtrack

► Agradecimientos

Introducción a los Sistemas de Bugtrack

phpBugtrack Mantis Bugzilla
FastBugtrack bug release
version
component

MUCHAS GRACIAS