

## СИСТЕМЫ УПРАВЛЕНИЯ ОШИБКАМИ ПРИ РАЗРАБОТКЕ ПРОГРАММНЫХ ПРОДУКТОВ

Артюх Т.С., Попова Ю.Б.

БНТУ, г. Минск, Беларусь, [tuttim@tut.by](mailto:tuttim@tut.by),  
БНТУ, г. Минск, Беларусь, [julia\\_popova@mail.ru](mailto:julia_popova@mail.ru)

### *Введение*

Неотъемлемой частью каждого этапа разработки программных продуктов является тестирование. Для документирования и отслеживания найденных ошибок (дефектов, багов) используются различные автоматизированные системы (англ., bug tracking system - BTS, БТС). Современная система отслеживания ошибок представляет собой прикладную программу, разработанную с целью помочь команде разработчиков учитывать и контролировать ошибки и неполадки, найденные в программах, пожелания пользователей, а также следить за процессом устранения этих ошибок и выполнения или невыполнения пожеланий.

Главный компонент такой системы — база данных, содержащая сведения об обнаруженных дефектах. Эти сведения могут включать в себя: номер (идентификатор) дефекта; данные о тестирующем; дату и время обнаружения дефекта; версию продукта, в которой обнаружен дефект; важность дефекта и приоритет решения; описание шагов для выявления дефекта (воспроизведения неправильного поведения программы); данные об ответственном лице за устранение дефекта; обсуждение возможных решений и их последствий; текущее состояние (статус) дефекта; версия продукта, в которой дефект исправлен. Кроме того, развитые системы предоставляют возможность прикреплять файлы, помогающие описать проблему (например, дампы памяти или скриншоты) [1].

Выбор идеальной системы управления проектами всецело зависит от миссии, которую группе программистов предстоит выполнять. Специалисты уже давно обсуждают пути эволюции новой культуры автоматизации совместного веб-программирования [2].

В докладе будут рассмотрены наиболее популярные на сегодняшний день БТС.

### *Выработка требований и критериев сравнения BTS*

Процесс учета ошибок сильно эволюционировал, а вместе с ним возросли и требования, без которых нормальная работа системы уже невозможна. Поэтому в настоящее время любая БТС соответствует ниже перечисленным требованиям [3].

- простота организации установки BTS: система не должна быть перегружена, должна легко поддаваться настройке;
- удобный для пользователя Dashboard, обладающим таким функционалом, как группировка информации, вкл/выкл отображения блоков. К примеру, можно отображать все текущие баги на проекте, а также баги, которые отправлены на проверку исключительно ему;
- поиск и сортировка жизненно необходимы, когда на проекте создано много отчетов об ошибках;
- создание подпроектов и дочерних задач. Наличие такого функционала важно, если разрабатываемых продуктов несколько. Одна задача может быть частью другой, при этом закрыть родительскую задачу нельзя, прежде чем не будут закрыты все ее подзадачи.
- настраиваемые дополнительные поля. Необходимость в таких полях возникает, когда не хватает основных полей для описания ошибки;
- настраиваемый workflow (жизненный цикл ошибки). Позволяет сконфигурировать все стадии, которые проходит ошибка в зависимости от проекта, и роль участника в этом проекте.

### Анализ систем отслеживания ошибок

Все многообразие систем управления и отслеживания ошибок можно разделить на три категории:

- некоммерческие системы: например, Trac, Redmine, Mantis, Bugzilla;
- коммерческие: например, FogBugz, Jira;
- корпоративные разработки для собственного потребления (как правило, являются закрытыми, поэтому рассматриваться в данной работе не будут).

Все эти системы зарекомендовали себя, как надежные средства учета ошибок в проектах, обладают большими и гибкими возможностями.

Первым рассмотрим багтрекер *Redmine*. Это современное, отвечающее всем требованиям, некоммерческое решение, как по учету ошибок, так и по другим задачам на проекте. Он появился в 2006 г. и уже с первых версий представлял собой удобную и функциональную систему по управлению проектами. Привлекает пользователей очень большим функционалом прямо «из коробки», нет необходимости в кастомизации, настройке и установке дополнительных плагинов. Хотя последние доступны в большом количестве. На рисунке 1 представлено главное окно приложения.

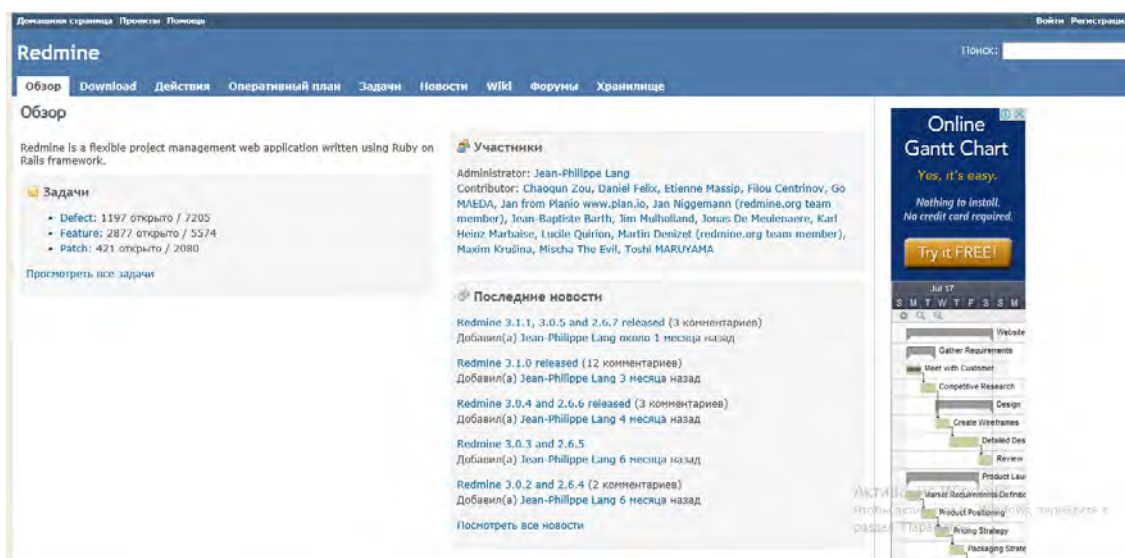


Рисунок 1 – Главное окно Redmine

На стартовой странице представлен Dashboard, на котором видны такие блоки, как календарь, список ошибок, отправленных участнику проекта, и список всех ошибок, которые были заведены на всех существующих проектах. Администратор или пользователь с соответствующими правами могут настраивать workflow. Можно выбирать роли и редактировать ошибки, т.е. изменять статусы ошибок. Статусы могут быть, к примеру: в обсуждении, новая, в процессе, необходимо протестировать, открыта, верифицирована, закрыта, отклонена.

Положительными сторонами Redmine являются: приятный пользовательский интерфейс, интеграция с системами управления версиями, поддержка плагинов, миграции с других БТС, гибкая система разграничения прав доступа пользователей, вложенные проекты неограниченной глубины, пакетное редактирование задач, возможность создавать дополнительные поля, настраивать их видимость, просмотр вложений из БТС, возможность создавать подпроекты и подзадачи, пометать задачи для отслеживания.

К недостаткам Redmine отнесем сложность установки и обновления (требуется язык программирования Ruby), а также отсутствие прав на отдельные типы переходов в Workflow.

*MantisBT* - это довольно старый, однако популярный багтрекер, и на это есть несколько причин: очень прост в работе, имеет минималистичный интерфейс, не перегружен ненужными элементами, не требователен к ресурсам, написан на PHP, очень прост в установке. Вместе с тем является довольно медленным в развитии и внедрении нового функционала.

Среди преимуществ *MantisBT* можно выделить следующие: множество настроек и ничего лишнего, подсветка состояния багов, возможность интеграции с системами контроля версий, поддержка многоуровневой иерархии, гибкая система фильтров, поддержка русского языка.

К недостаткам *MantisBT* отнесем сложности с операциями настройки и невозможность настроить внешний вид.

*Trac* представляет собой одновременно инструмент управления проектами и систему отслеживания ошибок. Часто используется для opensource-проектов, потому максимально прост и функционален, имеет анонимный доступ. Изначально создавался как интерфейс для работы с системами контроля версий, вследствие чего популярен среди программистов.

К преимуществам *Trac* отнесем мощный и интуитивный браузер репозитория, возможность привязки нескольких репозиториев разного типа, поиск по ключевым словам и строкам, настраиваемый отчет об ошибках, возможность извещений по RSS или по почте, поддержка подзадач, создание связей между задачами, изменениями, файлами и wiki-страницами, удобный интерфейс, поддержка русского языка, плагинов.

Среди недостатков *Trac* выделим следующие: большая часть функционала доступна только в качестве плагинов и определенная сложность установки.

*Bugzilla* – это мастодонт багтрекеров, первый релиз которого вышел еще в 1998 году. Изначально использовался исключительно как багтрекер для проектов Mozilla. На сегодняшний момент одна из наиболее популярных и часто используемых БТС. Написан на языке Perl. В *Bugzilla* нет огромной функциональности, присущей конкурентам, однако ее достаточно для малых и средних проектов. На рисунке 2 представлено окно описания ошибки в *Bugzilla*.

В дополнение укажем следующие положительные стороны *Bugzilla*: надежная и зарекомендовавшая система, интеграция с системами контроля версий, автопоиск при создании бага, многоуровневая иерархия, продуманная система нотификации, мощная система фильтров, встроенные средства справки, возможность русификации.

К отрицательным сторонам *Bugzilla* отнесем определенную сложность установки и сложность администрирования.

*Jira* считается практически идеальным решением для крупных проектов, для компаний с большим штатом тестировщиков. В первую очередь позиционируется как система управления проектами. *Jira* можно использовать как в облаке, так и развернуть у себя на сервере. Можно устанавливать на все популярные операционные системы. На рисунке 3 представлено окно Backlog системы *Jira*.

Преимущества *Jira* заключаются в следующем: дружелюбный интерфейс, мощная система фильтров, настраиваемая подсветка приоритетов, визуальный Workflow, различные диаграммы метрик, простая установка, удобные миграторы с других БТС, поддержка шорткатов, настраиваемый глобальный Dashboard, удобная установка.

К минусам *Jira* отнесем невозможность создавать подпроекты и подзадачи, несколько запутанный Workflow, заторможенность работы в некоторых ситуациях.

*Jira* рекомендуется к использованию командам работающим над большими проектами. Одна из самых дешевых коммерческих систем.

Среди преимуществ *FogBugz* выделим следующие: удобная система поиска и фильтрации, групповое редактирование задач, создание задач через email, дружественный ин-

терфейс, простота использования, инструмент для снятия скриншотов, возможность частичной графической кастомизации, поддержка плагинов.

К минусам FogBugz отнесем невозможность задать обязательные поля, переход в офф-лайн статус на техническое обслуживание (система находится в облаке).

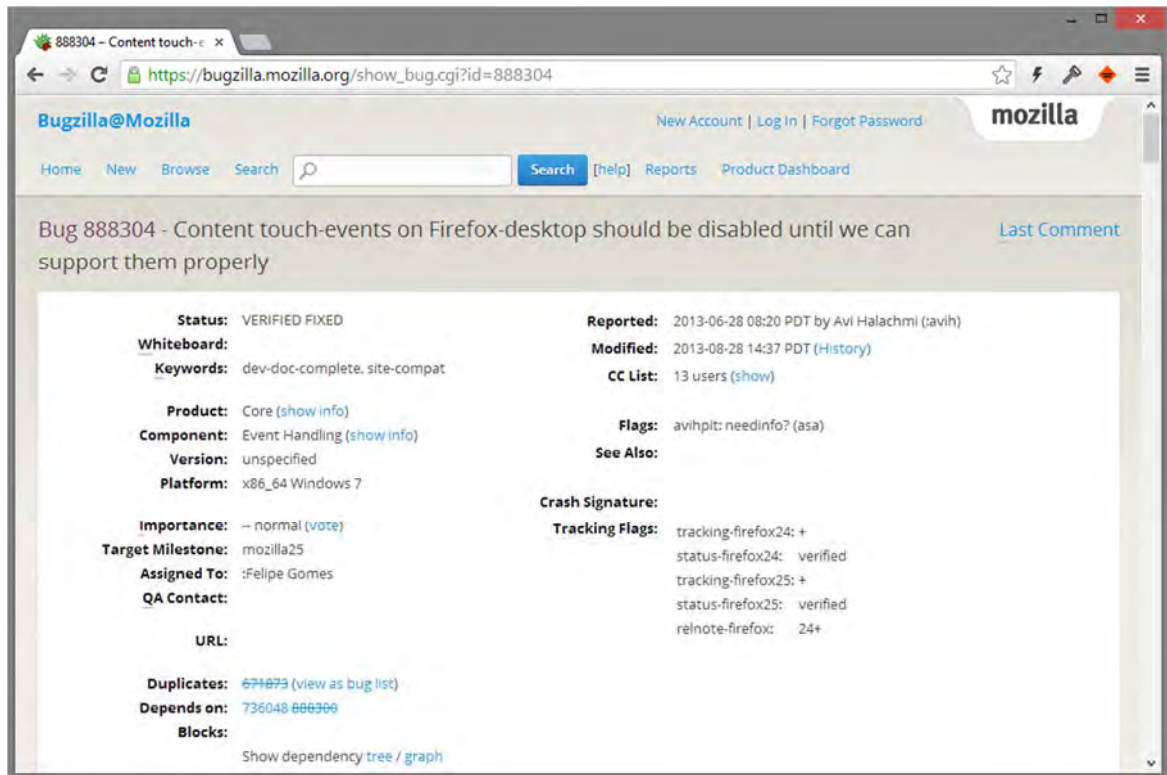


Рисунок 2 – Окно описания ошибки в Bugzilla

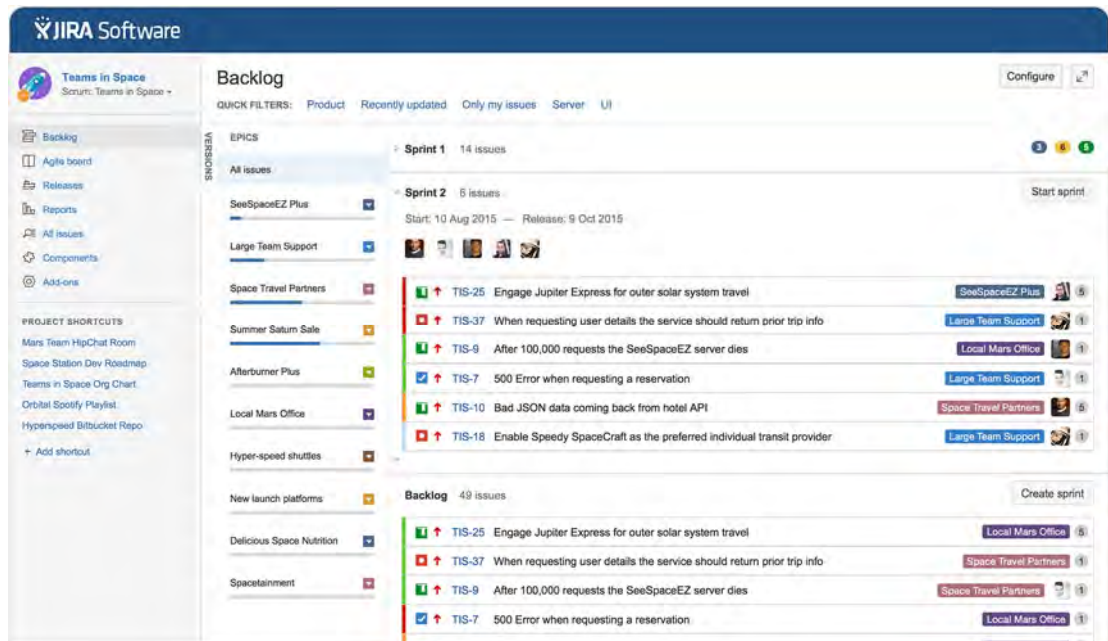


Рисунок 3 – Окно Backlog системы Jira

*Заключение*

Каждая из рассмотренных систем отслеживания ошибок по-своему уникальна. Все багтрекеры стараются идти «в ногу со временем», внедряя все новые технологии в виде плагинов и расширений, тем самым нагружая систему и затрудняя пользователю процесс установки и настройки.

К выбору системы отслеживания ошибок следует подходить основательно, взвесив все плюсы и минусы каждой из них, а также задачи, которые стоят перед командой, численность сотрудников, занятых на проекте, а также материальную составляющую. Рекомендовать какую-то определенную БТС «на все случаи жизни» пока не представляется возможным.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Багтрекинг-система [Электронный ресурс]. - 2015. - Режим доступа: <http://zoya.0fees.us/bugtracker.html?skattempt=1>. - Дата доступа: 2.11.2015.
2. Системы управления проектами для разработчиков [Электронный ресурс]. -2012. - Режим доступа: <http://habrahabr.ru/company/webprofessionals/blog/139826/>. - Дата доступа: 2.11.2015.
3. Обзор багтрекерных систем [Электронный ресурс]. - 2014. - Режим доступа: [http://notebookvideo.ru/wsUHTfx5H1o/obzor\\_bagtrekernyh\\_sistem\\_bug\\_tracking\\_system.html](http://notebookvideo.ru/wsUHTfx5H1o/obzor_bagtrekernyh_sistem_bug_tracking_system.html). - Дата доступа: 2.11.2015.