

Лабораторная работа №24.

Изучение и сравнение систем отслеживания ошибок

1. Цель работы

Изучить на примерах системы отслеживания ошибок, научиться оформлять дефекты при помощи таких систем.

2. Общие сведения и ход работы

Система отслеживания ошибок (bug tracking system) – прикладная программа, разработанная с целью помочь разработчикам программного обеспечения(программистам, тестировщикам и др.) 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;
- короткое описание дефекта;
- кто сообщил о дефекте;
- дата и время, когда был обнаружен дефект;
- версия продукта, в которой обнаружен дефект;
- серьезность (критичность) дефекта и приоритет решения;
- описание шагов для выявления дефекта (воспроизведения неправильного поведения программы);
- ожидаемый результат и фактический результат;
- кто ответственен за устранение дефекта;
- обсуждение возможных решений и их последствий (комментарии);
- текущее состояние (статус) дефекта;
- версия продукта, в которой дефект исправлен.

Кроме того, развитые системы предоставляют возможность прикреплять файлы, помогающие описать проблему (например, скриншот).

Как правило, система отслеживания ошибок использует тот или иной вариант «жизненного цикла» ошибки, стадия которого определяется текущим состоянием, или статусом, в котором находится ошибка.

Типичный жизненный цикл дефекта:

1. **Открыт** – дефект зарегистрирован тестировщиком и назначен на исполнителя.
2. **Разрешён** – дефект переходит обратно в сферу ответственности тестировщика. Как правило, сопровождается резолюцией, например:
 - **Исправлено** (исправления включены в версию *такую-то*);
 - **Дубль** (повторяет дефект, уже находящийся в работе);
 - **Не исправлено** (редкая ситуация, которую не имеет смысла обрабатывать, имеет слишком низкий приоритет, исправление отложено до следующей версии и т.п.);
 - **Не является дефектом** (работает в соответствии со спецификацией, оформлен по ошибке тестировщиком или был вызван некорректными тестовыми данными);
 - **Невоспроизводимо** (запрос дополнительной информации об условиях, в которых дефект проявляется).
3. Далее тестировщик проводит проверку исправления, в зависимости от чего дефект либо переходит в статус **Переоткрыт** (если он описан как исправленный, но не исправлен), либо в статус **Закрит**.

Система может предоставлять администратору возможность настроить, какие пользователи могут просматривать и редактировать ошибки в зависимости от их состояния, переводить их в другое состояние или удалять.

В корпоративной среде, система отслеживания ошибок может использоваться для получения отчётов, показывающих продуктивность программистов при исправлении ошибок. Однако, часто такой подход не даёт достаточно точных результатов, из-за того что разные ошибки имеют различную степень серьёзности и сложности. При этом серьёзность проблемы не имеет прямого отношения к сложности устранения ошибки.

Ход работы:

1. Получить доступ к системам отслеживания ошибок из таблицы 2:

Таблица 2 – Системы отслеживания ошибок

№	Название системы	URL	Как настроить доступ
1.	Bugs	http://www.softaculous.com/demos/Bugs	<ol style="list-style-type: none"> 1. Перейти по указанной ссылке. 2. Ввести логин и пароль админа (указаны в правом верхнем углу). 3. Добавить новый проект (нажать на зеленый плюсик и ввести название и пользователя для нового проекта). 4. На первой странице проекта появится кнопка «New issue» для создания нового дефекта или задачи [4].
2.	Trac	https://trac.edgewall.org/	<ol style="list-style-type: none"> 1. Перейти по указанной ссылке. 2. Выбрать текущую версию системы в правой части страницы, например, Trac 1.0 (current). 3. На открывшейся странице справа выбрать необходимый пункт меню, например «Новая карточка» [4].
3.	Flyspray	http://www.softaculous.com/demos/Flyspray	<ol style="list-style-type: none"> 1. Перейти по указанной ссылке. 2. Нажать «Login!» и ввести логин и пароль админа (указаны в правом верхнем углу). 3. Для создания дефекта/задачи выбрать в верхнем меню пункт «Add new task» [4].

2. Оформить все дефекты, обнаруженные в ходе тестирования программы «Калькулятор» (лабораторная работа № 7-8) в указанных системах багтрекинга:

- **заголовок** – должен быть кратким и содержать в себе суть дефекта;
- **описание** – включает подробные шаги воспроизведения дефекта, ожидаемый результат, фактический результат, необходимые вложения (скриншоты, логи, выборки из БД);
- **приоритет** – согласно правилам приоритизации (см. таблицу 3);
- **проект** – наименование проекта;
- **версия приложения** – версия, в которой был обнаружен дефект;

- **область функционала** – компонент системы, в котором был обнаружен дефект;
- **автор** – автор дефекта, например, тестировщик, который ввел дефект в систему отслеживания ошибок;
- **дефект назначен** – ответственный за дальнейшую работу с дефектом (за исправление);
- **версия исправления** – версия, в которой дефект будет исправлен (не обязательно для указания).

Таблица 3 – Правила приоритизации дефектов

Важность	Определение	Влияние на бизнес Заказчика		
		Низкое или очень специфические условия	Высокое	Критичное
Катастрофа	<ul style="list-style-type: none"> - Дефект приводит к невозможности завершить выполнение или тестирование бизнес-процесса. - Дефект, приводящий к непреднамеренному завершению работы Системы, либо к невозможности запустить Систему. - Производительность системы не позволяет выполнять либо тестировать базовые бизнес-процессы [1]. 	Blocker	Blocker	Blocker
Критично	<ul style="list-style-type: none"> - Дефект приводит к невозможности завершить выполнение или тестирование бизнес-процесса, но возможно завершить или протестировать этот процесс обходным путем. - Система не учитывает ограничения в доступе или другие настройки безопасности. - Дефект приводит к некорректным финансовым вычислениям. - Дефект приводит к потере пользовательских данных в БД [1]. 	Low	Critical	Critical
Высоко	<ul style="list-style-type: none"> - Часто встречающийся функциональный дефект, не приводящий к невозможности завершить выполнение или тестирование функционала или потере пользовательских данных в БД [1]. 	Low	Medium	Medium
Низко	<ul style="list-style-type: none"> - Система формирует некорректные сообщения об ошибке, либо не формирует когда это необходимо. - Дефекты пользовательского интерфейса, которые не влияют на функционирование или тестирование Системы (некорректная верстка, лишние полосы прокрутки и т.п.) [1]. 	Trivial	Low	Medium

альная- Триви	<ul style="list-style-type: none"> - Грамматические ошибки. - Другие незначительные дефекты, не перечисленные выше [1]. 	Trivial	Trivial	Medium
------------------	---	---------	---------	--------

3. Провести дефекты по жизненному циклу, доступному в изучаемых системах.
4. Изучить возможности систем по работе с проектом.
5. Сравнить указанные системы отслеживания ошибок между собой и заполнить таблицу 4:

Таблица 4 – Сравнение систем отслеживания ошибок

Признак сравнения	Bugs	Trac	Flyspray	Комментарий
Стоимость				
Наличие всех необходимых полей для оформления дефекта				
Доступность создания проекта в системе				
Наличие фильтрации				
Наличие рабочей панели проекта с основной информацией				
Возможность отслеживать план работ по проекту				

Жизненный цикл дефекта (количество статусов, удобство изменения статуса)				
Хранение истории изменений по дефекту				
Общая оценка системы				

3. Содержание отчета по лабораторной работе

1. Цель работы.
2. Примеры дефектов, оформленных в представленных для изучения системах багтрекинга (скриншоты).
3. Движение дефектов по жизненному циклу (описание жизненного цикла дефекта для каждой из систем, скриншоты, если требуются).
4. Результаты заполнения таблицы 4.
5. Выводы по лабораторной работе.

