

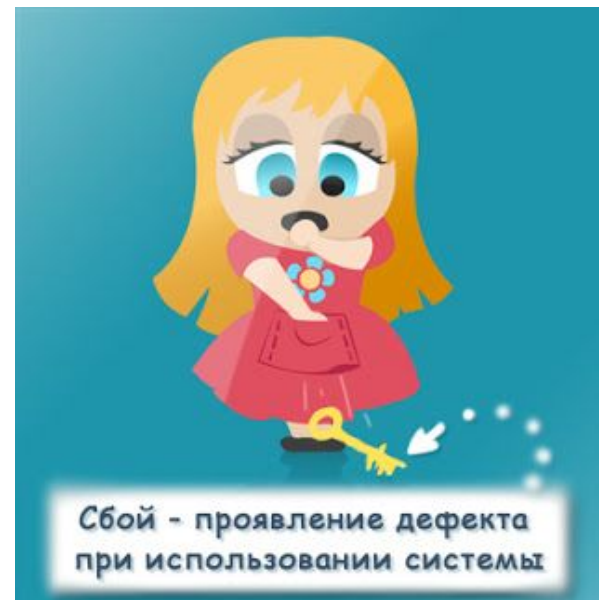
# Software Engineering

## Лекция 11: Дефекты и что с ними делать

Тимофей Брыксин

[timofey.bryksin@gmail.com](mailto:timofey.bryksin@gmail.com)

# Ошибка, дефект, сбой



9/9

0800 Antam started  
 1000 " stopped - antam ✓  
 1300 (032) MP-MC ~~1.982147000~~  
 (033) PRO 2 2.130476415 (2) 4.615925059(-2)

concl

2.130676415

Relays 6-2 in 033 failed special speed test  
 in relay " 11.000 test.

Relays changed

1100 Started Cosine Tape (Sine check)  
 1525 Started Multi Adder Test.

1545



Relay #70 Panel F  
 (moth) in relay.

First actual case of bug being found.

1630 antam started.

1700 closed down.

Relay  
 3145  
 Relay 3370

# Системы управления дефектами (bug/issue trackers)

- Хранят и предоставляют доступ к БД дефектов
- Управляют правами доступа
- Отправляют нужные нотификации
  - изменение ЖЦ дефекта
  - почта/IM/SMS
- Собирают статистику и формируют отчёты

# Характеристики дефектов

- Идентификатор дефекта
- Заголовок
- Описание дефекта (включая шаги воспроизведения)
- Контекст
- Тип
- Серьёзность
- Приоритет
- Статус
- Автор
- Ответственный за исправление, ответственный за проверку
- Зависимость
- Временные параметры устранения дефекта
- ...

# Идентификатор дефекта

- Уникальный в рамках всего проекта
- Может кодировать в себе доп. информацию о дефекте
- Создаётся один раз и не меняется
  - чтобы не ломать ссылки

# Заголовок

- Короткий и ёмкий, описывающий суть проблемы

PUBLIC rails / rails

Watch 1,524 Star 22,519 Fork 8,425

Issues Pull requests Labels Milestones Filters is:open is:issue New issue

266 Open 5,567 Closed Author Labels Milestones Assignee Sort

- Migration change\_column array:true, default: '{}' don't work rails 4.1.4  
#16233 opened 2 hours ago by nilocoelhojunior
- 'becomes' method does not copy unsaved attributes activerecord  
#16206 opened 3 days ago by thehappycode
- ParamsWrapper, nested resource, parent id never included actionpack  
#16194 opened 3 days ago by elbarto132
- ActiveRecord::Relation .size regression with distinct select activerecord regression  
#16182 opened 4 days ago by aripollak
- ActionMailer: calling "attachments[]" after "mail" leads to unexpected behavior actionmailer  
#16163 opened 6 days ago by joakimk
- ActionController::UrlFor requires more as a Rack-compatible request  
#16160 opened 6 days ago by dspaeth-faber
- Counter cache doesn't update if parent model primary key changes  
#16159 opened 6 days ago by noinkling

# Описание дефекта

- Полный, но краткий отчёт о проблеме
- Шаги по воспроизведению
- Ожидаемый и фактический результат
- Один дефект -- одна проблема
- Дополнительная информация
  - логи, дампы памяти
  - ссылки
  - скриншоты
  - видео



# Контекст

- Проект/программа/задача
- Номер версии, ревизии
- Модуль, компонент

# Тип дефекта

- Ошибка функционирования
- Ошибка документации
- Предложение по усовершенствованию
- Предложение по изменению
- Новая функциональность
- ...

# Серьёзность (severity)

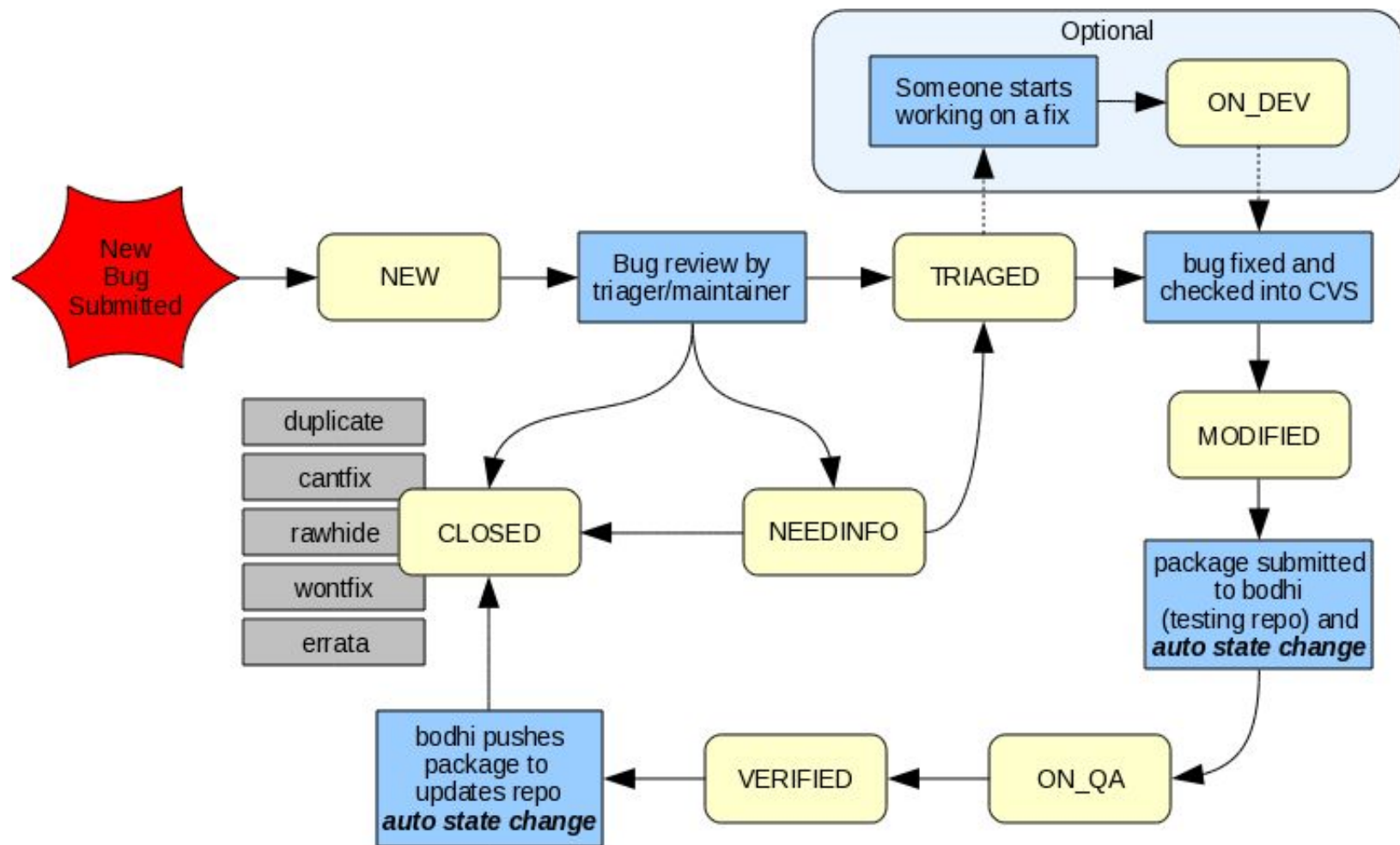
- Косметический дефект
- Обычный дефект
- Серьёзная ошибка
  - зависание
  - аварийное завершение
  - потеря данных
  - ...
- Showstopper
- ...

# Приоритет дефекта

- Показывает срочность исправления
  - Низкая
  - Средняя
  - Высокая
  - Срочная

# Статус дефекта

- Отражает состояние жизненного цикла дефекта
  - Новый
  - Взятый на исправление
  - Исправленный
  - Закрытый (исправленный и проверенный)
  - Незакрытый (исправленный, но проваливший проверку)
  - Отклоненный
  - Дубликат
  - Заново открытый
  - Временно приостановленный
  - Требующий пояснения
  - Не проявляющийся
  - “Не баг, а фича”
  - ...



# И другие характеристики

- Зависимость
- Автор
  - возможно, нужно несколько багтрекеров
- Ответственный за исправление, ответственный за проверку
- Временные параметры устранения дефекта
- ...

# Отладка

- Процесс воспроизведения, локализации и устранения ошибок
- Синтаксическая и семантическая отладка





# Воспроизведение дефекта

- **Невоспроизводимую ошибку невозможно по-настоящему исправить**
- Может занять значительное время
  - стабильная воспроизводимость -- 90% исправления
- Непредвзятость и внимательность
- Минимизация числа шагов и автоматизация их
  - минимум один автотест
- Формирование гипотезы
  - метод индукции (от частного к общему)
  - метод дедукции (от общего к частному)
  - обратное движение по алгоритму

# Локализация ошибки

- Виды локализации
  - аналитический
  - экспериментальный
    - простой просмотр текста
    - отладчик
    - трассировка, отладочная печать
    - профайлеры
- Место проявления дефекта != место ошибки
- Беспристрастность и “замыливание”

# Вредные советы по отладке

- Cargo cult programming
- Shotgun programming
- Programming by accident
- Least effort programming
- Surgical programming



# Исправление ошибки

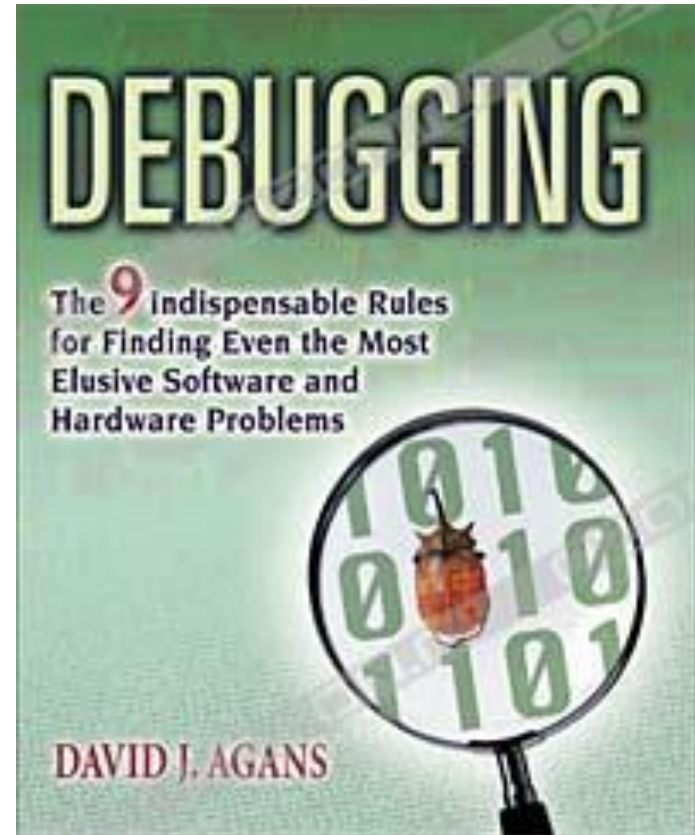
- Там, где есть одна ошибка, скорее всего есть и другие
- Исправлена ли ошибка?
  - исправлена ли она нашим исправлением?
- А та ли ошибка исправлена?
- Не появились ли от этого другие ошибки?
- Лечим болезнь, а не симптомы

# "Плавающие баги"

- Неинициализированные переменные
- Ошибки синхронизации процессов/потоков
- Программно-аппаратные ошибки
- Различия между debug и release сборками

# David J. Agans. Debugging: ...

1. Understand the System
2. Make It Fail
3. Quit Thinking and Look
4. Divide and Conquer
5. Change One Thing at a Time
6. Keep an Audit Trail
7. Check the Plug
8. Get a Fresh View
9. If You Didn't Fix It, It Ain't Fixed



# Хороший код

- Надёжный
- Эффективный
- Понятный
- Правильный
  - делает то, что должен
  - не делает того, чего не должен
  - не боится непредвиденных ситуаций
    - не делает предположений относительно окружения
    - готов всегда и ко всему
    - способен пережить некорректные ситуации

# Защитное программирование

- Применение анализа рисков в программировании
  - в противовес отладке преимущественно профилактический метод
- Минусы
  - нужно писать больше кода, тратим больше времени
  - снижается эффективность
  - снижается читаемость кода
- Плюсы
  - раннее предупреждение дефектов
    - защита от дураков и злоумышленников
  - сокращается время отладки
  - чуть менее эффективный, зато правильно работающий код
    - проверки можно убрать в release-версии



# Общие принципы

- Хороший стиль кодирования
- Дисциплина, отсутствие спешки
  - “чем больше спешки, тем меньше скорость”
- Недоверие ко всему
- Ясность превыше краткости
- Ясность превыше крутости
- Максимальная закрытость
- Отсутствие предупреждений компилятора
- Использование статических анализаторов
- Логгирование и прочая диагностика
- Проверка входных данных, аргументов методов и возвращаемых значений

# Ограничения (assertions)

- Назначение

- проверка пред- и постусловий
- проверка инвариантов
- обработка ошибок для ожидаемых событий
- assert'ы на события, которые происходить не должны
- выполняемая документация

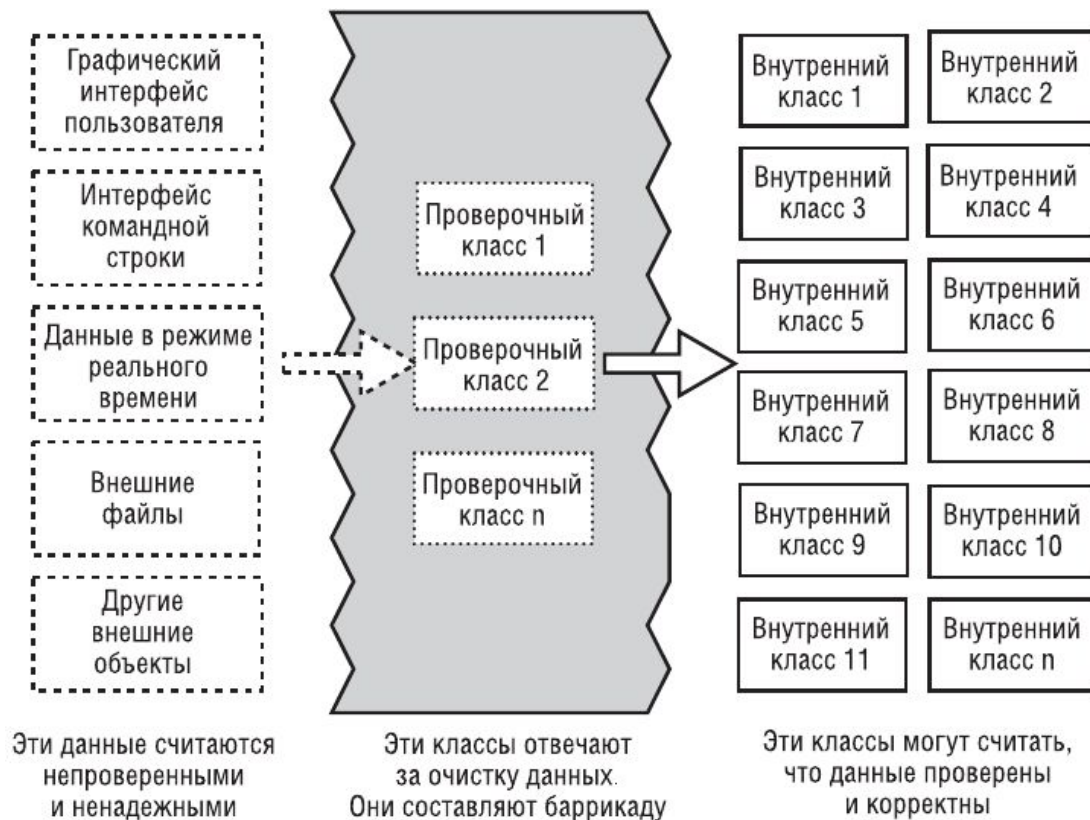
- Полезные рекомендации

- не помещайте выполняемый код внутрь assert'ов
- напишите assert, а потом всё равно обработайте ошибки

# Стратегия реакции на ошибку

- Вернуть нейтральное значение
- Заменить следующим корректным блоком данных
- Вернуть тот же результат, что и в предыдущий раз
- Подставить ближайшее допустимое значение
- Сообщить об ошибке
- Прекратить выполнение
- Добавить запись в лог
- Устойчивость vs Корректность

# Изоляция ошибок



# А как же release версия?

- Проверяем существенные ошибки, несущественные игнорируем
- Убираем код, завершающий выполнение программы
- Регистрируем все ошибки
- Делаем сообщения об ошибках понятными и дружелюбными :)



