

# Практика разработки программного обеспечения

## 13. Тестирование. Тестировщик и его работа.

# «Bug» (1947)

9/9

0800 Antan started  
 1000 " stopped - antan ✓  
 1300 (032) MP - MC ~~1.982147000~~  
 (033) PRO 2 2.130476415  
 2.130676415  
 9.037847025  
 9.037846995 correct  
 4.615925059(-2)

Relays 6-2 in 033 failed special speed test  
 in relay .. 11.00 test.

Relay  
 2145  
 Relay 3370

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

1545



Relay #70 Panel F  
 (moth) in relay.

First actual case of bug being found.  
 1630 Antan started.  
 1700 closed down.

# Test Case – Тест

— небольшое действие, описываемое несколькими простыми шагами, с определенным ожидаемым результатом. Тест называют *удачным*, если он выявил ошибку.

**Test suite** – пакет тестов.

# Test Plan – План тестирования

- что подвергается тестированию
- кто выполняет тестирование
- как будет выполняться тестирование
- сколько времени это займет
- глубина тестирования
- требования к окружению
- спецификации на тесты

# Bug Report – Отчет о проблеме

1. Программа / модуль / компонент
2. Номер версии программы
3. Окружение
4. Тип проблемы
5. Важность
6. Проблема
7. Воспроизводимость
8. Подробное описание
9. Приложения
10. Автор отчета
11. Дата / время
12. Кому поручено
13. Комментарии
14. Состояние
15. Приоритет

[Details](#)[Workflow items](#)[Links](#)[Attachments](#)[Comments](#)[Subscriptions](#)[History Log](#)

## Summary

Summary \*

Location

IND-CAR

Ref #

WBS Branch

Review



Review Work Product

## General Info

Severity \*

Medium

Priority

(none)

Reproducible \*

Always

Test Method

(none)

Symptom \*

Incorrect operation

## Builds Info

Build Found \*

Build to Fix

Build Fixed

Deferred to Build

Verified in Build

Reopen Count

Description

[Expand](#)

Steps to Reproduce

[Expand](#)

## Report details

Reported by

[Nikita Zimin](#)

How Found

Test case

Where Issued

# Типичные ошибки: Ошибки пользовательского интерфейса

- Неожиданное поведение программы
- Отсутствие предупреждения о необратимом действии
- Невозможность определить состояние программы
- Разные названия для одной сущности
- Избыточная или недостаточная валидация пользовательского ввода

# Типичные ошибки (2): Обработка ошибок

- Отсутствие сообщения об ошибке
- Граничные условия



# Типичные ошибки (3):

## Ошибки вычислений

- Неучтенное округление
- Банковское/математическое округление
- Переполнение
- Использование вещественного типа для денежных величин
- Ошибки при преобразовании формата (десятичная точка или запятая)
- Сравнение значений вещественного типа

# Типичные ошибки (4): Прочее

- Отсутствие начального значения
- Ситуация гонки

# Разработка через тестирование

## Test Driven Development (TDD):



Принцип «Test First»

# План тестирования

## IEEE 829-1998 Standard for Software Test Documentation

1. Идентификатор документа
2. Ссылки
3. Введение
4. Цель
5. Объем тестирования
6. Тестируемые части (функции)

# План тестирования (2)

7. Риски
8. Что будет протестировано
9. Что НЕ будет протестировано
10. Стратегия тестирования
11. Критерии удачного / неудачного теста
12. Критерии входа / выхода
13. Результаты тестирования

# План тестирования (3)

- 14. Задачи, оставшиеся  
непротестированными
- 15. Требования к окружению
- 16. Набор / обучение персонала
- 17. Ответственность
- 18. Планирование рисков и форс-мажор
- 19. Процесс утверждения
- 20. Глоссарий

# Метрики

*«Вы не можете контролировать то,  
что не можете измерить.»*

Том ДеМарко

*Метрика программного обеспечения —  
это мера, позволяющая получить  
численное значение некоторого  
свойства ПО.*

# Метрики

- порядок роста (O-нотация)
- количество строк исходного кода
- количество функциональных точек
- цикломатическая сложность
- среднее количество ошибок на строку кода
- покрытие кода
- количество классов и интерфейсов
- связность кода



# Метрики в реальной жизни

Метрики как способ измерения  
производительности труда

Сибудение

# Метрики процесса разработки в целом

- количество раз когда ночной билд был «сломан»
- количество дефектов, вносимых за час работы разработчика
- количество изменений требований
- количество «заплаток», выпускаемых после поставки продукта потребителям
- среднее время исправления ошибки

# Критика метрик

1. Неэтичность
2. Замещение «управления людьми» на «управление цифрами»
3. Искажение
4. Неточность – нет метрик, которые были бы одновременно и значимы и достаточно точны

# Итог – Тестирование

- Тестирование – основной метод обеспечения качества
- Тестирование ведется согласно плану тестирования
- Детально описываются отдельные тесты
- Найденные ошибки фиксируются в системе Issue Tracking
- Для оценки результатов тестирования используются метрики