

# Тестирование программных средств

Сафронов Сергей,  
2009 год

# Оглавление

- Принципы тестирования
- Ручное тестирование
  - Инспектирование
    - Порядок действий
    - Признаки, что инспектирование провалилось
    - Список вопросов для выявления ошибок
  - Сквозной просмотр
  - Оценка посредством просмотра

# Принципы тестирования

- Описание выходных значений – необходимая часть тестового набора
- Тестирование автором – неэффективно
- Негативные тесты следует разрабатывать так же тщательно, как и позитивные
- Необходимо проверять не только то, что программа делает то что должно, но и то, что она не делает то что не должна
- Не следует выбрасывать тесты по завершению тестирования
- При планировании нельзя закладывать на отсутствие ошибок
- Вероятность обнаружения ошибок в модуле пропорциональна числу уже обнаруженных ошибок
- Тестирование – процесс творческий

# Инспектирование

Инспектирование – набор процедур и приемов обнаружения ошибок при изучении текста группой специалистов

Состав inspectирующей группы:

- Автор
  - Готовит материал для инспекции
  - Отвечает на все вопросы по ходу
- QA
- Председатель (Senior Programmer или Team Leader)
  - Составляет план инспекции
  - Запись результатов и контроль следования плану
- Проектировщик

# Где и когда применять

- Сбор требований
  - спецификации
- Разработка архитектуры
  - Дизайн функций
- Детальная разработка
  - Внутренний дизайн
- Кодирование и юнит тестирование
  - Код

# Порядок действий

- **Планирование**
  - Материалы распределяются за 2-3 дня до
  - Определяются основные моменты, на что обратить внимание
  - Определяются критерии успеха
- **Подготовка**
  - Проверка на соответствие стандартам
  - Использование результатов прошлой проверки
- **Инспектирование**
  - Никакой инспекции при отсутствии подготовки
  - Выработка решений и рекомендаций
  - Разбор ошибок
- **Разбор и анализ результатов**
  - Проверка на исправление всех выявленных ошибок
  - Следование рекомендациям и решениям

# Признаки того, что инспекция провалилась

- **Технические**
  - Концентрация на тривиальном
  - Переход на личности
  - Непонимание метрик, которые проверяются
  - Инспектирование чего угодно кроме качества
  - Не выработано никаких рекомендаций
  - Не выдержаны временные рамки
  - Не соответствие плану работы
- **Организационные**
  - Затраты видны, результаты незаметны
  - Не информировать участников о прогрессе
  - Директивное управление/навязывание мнения
  - Не позволять ошибкам влиять на процедуру
- **Внедрение**
  - Ожидание чудес от применения
  - Несоответствие ожиданиям руководства
  - Не вынесены никакие уроки

# Порядок действий (Майерс)

- Обсуждение логики программы (краткое описание автором программы)
- Анализ по списку наиболее частых ошибок программирования

## Основные моменты:

- Время инспекции – не более 2 часов
- Средняя скорость – 150 операторов в час
- Конфиденциальность результатов
- Attack the problem, not the person

# Список вопросов для выявления ошибок

Ошибки обращения к данным:

- Неинициализированные переменные
- Выход за границы массивов
- Индексы – целые числа?
- Обращение к невыделенной/освобожденной памяти
- Корректны ли атрибуты во всех псевдонимах?
- Битовые строки: вычислимы ли адреса?  
Передаются ли как аргументы?
- Идентичность определения структур в разных модулях
- Проблемы с индексацией и обращениями к массивам

# Список вопросов для выявления ошибок

## Вычисления:

- Есть ли вычисления неарифметических переменных
- Вычисления с использованием данных разных типов
- Вычисления с переменными разной длины
- Переполнение/потеря промежуточного значения
- Деление на ноль
- Не выходят ли значения переменной за пределы установленного диапазона
- Деление целых чисел

# Список вопросов для выявления ошибок

## Описание данных

- Все ли переменные описаны?
- Правильно ли инициированы массивы и строки
- Правильно ли выделяется память
- Переменные со сходными именами

## Передача управления

- Будет ли завершен каждый цикл
- Рекурсия
- невыполнимые циклы
- Корректность числа итераций цикла
- Неявные решения

# Список вопросов для выявления ошибок

## Сравнение

- Сравнение величин несравнимых типов
- Корректность булевских выражений
- Порядок следования операторов

## Ввод-вывод

- Корректность атрибутов файлов
- Соответствие ввода/вывода формату
- Соответствует ли размер буфера размеру записи?
- Открыты ли файлы перед использованием
- Ошибки ввода/вывода
- Ловятся ли признаки конца файла

# Список вопросов для выявления ошибок

## Интерфейс

- Равно ли число параметров числу аргументов
- Соответствие типов параметров и аргументов
- Соответствие единиц измерения для параметров/аргументов
- Правильно ли заданы число аргументов, типы и порядок следования для аргументов встроенных функций
- Обращения к локалам вне контекста
- Передаются ли в качестве аргументов константы

## Другие виды контроля

- Есть ли какие-нибудь предупреждения или сообщения при компиляции
- Нет ли пропущенных функций
- Осуществляется контроль правильности входных параметров

# Сквозной просмотр

- Длительность – менее 2 часов
- Состав группы – 3-5 человек
  - Автор
  - QA
  - Председатель
  - Пользователь\*
  - Новичок в программировании\*

\* - могут отсутствовать

- Механизм – ручное выполнение программы (на некоем наборе тестов) и обсуждение всех возникших вопросов и найденных проблем

# Оценка посредством просмотра

- Число участников: 6 – 20
- От каждого 2 варианта кода: хороший и плохой (с точки зрения автора)
- Код нумеруется и вслепую раздается каждому по 2 варианта хорошего кода и 2 плохого, причем участникам эксперимента не говорится где какой
- 30 минут на анализ кода по след критериям:
  - Легко ли понять программу
  - Являются ли результаты проектирования высокого уровня очевидными и приемлемыми?
  - Являются ли результаты проектирования низкого уровня очевидными и приемлемыми?
  - Легко ли было бы для вас модифицировать этот код
  - Были ли бы вы довольны, если бы написали такую программу
  - Общее мнение/рекомендации
- Предъявление статистики участникам
- Цель: внутренняя оценка квалификации для программистов внутри команды