

Методы коллективной работы в проектах на базе  
свободного программного обеспечения

# Обеспечение качества программного продукта

Кафедра Телекоммуникационных  
технологий

# Понятие качества

2 подхода к оценке качества:

- продукт - процесс
- соответствие - усовершенствование

# Характеристика качества ПО

- Функциональность
- Надежность
- Удобство
- Эффективность
- Сопровождаемость
- Переносимость
- Добротность

# Качество процесса разработки. СММ

- Начальный уровень
- Повторяемый уровень
- Определенный уровень
- Управляемый уровень
- Оптимизирующий уровень

# Качество процесса разработки. Непрерывная модель

- Уровень 0. Процесс не выполняется
- Уровень 1. Выполняемый процесс.
- Уровень 2. Управляемый процесс.
- Уровень 3. Установленный процесс.
- Уровень 4. Предсказуемый процесс.
- Уровень 5. Оптимизирующий процесс.

# Качество баз данных

Следует разделять качество 2-х компонент:

- программные средства системы управления базой данных (СУБД), независимые от сферы их применения, структуры и смыслового содержания данных;
- информацию базы данных (ИБД), доступную для накопления, упорядочивания, обработки и использования в конкретной проблемно-ориентированной сфере применения.

# Тестирование ПО. Принципы тестирования

- Динамичность
- Конечность
- Выбор
- Ожидаемое поведение

# Тестирование ПО.

## Цели и виды тестирования (1/2)

- Приемочное тестирование
- Установочное тестирование
- Альфа- и бета-тестирование
- Функциональные тесты/тесты соответствия
- Оценка надежности
- Регрессионное тестирование

# Тестирование ПО. Цели и виды тестирования (2/2)

- Тестирование производительности
- Нагрузочное тестирование
- Сравнительное тестирование
- Восстановительные тесты
- Конфигурационное тестирование
- Юзабилити-тестирование

# Тестирование ПО. Уровни тестирования

- Модуль
- Группа модулей
- Система

# Юнит-тестирование

- Цель — доказать, что модули по отдельности работоспособны.
- Проводится программистами. Должны поддерживаться и в ходе изменения модулей.

# Интеграционное тестирование

Цель — доказать, что взаимодействие между модулями корректно.

2 метода:

- монолитный
- инкрементальный
  - сверху вниз
  - снизу вверх

# Системное тестирование

Цель — доказать, что система в целом работает корректно.

2 подхода:

- на базе требований
- на базе случаев использования

# Альфа- и бета-тестирование

- Альфа-тестирование — имитация реальной работы штатными тестерами.
- Бета-тестирование — доступ к работе с продуктом внешним пользователям.

# Статическое тестирование

Статический анализ кода. Виды ошибок:

- Неопределенное поведение
- Нарушение блок-схемы пользования библиотекой
- Типичные сценарии, приводящие к недокументированному поведению
- Переполнение буфера
- Прочие ошибки

# Юзабилити-тестирование (1/3)

Факторы влияющие на удобство пользования:

- легкость обучения
- эффективность обучения
- запоминаемость обучения
- ошибки
- общая удовлетворенность

# Юзабилити-тестирование (2/3)

## Этапы тестирования

- Исследовательское
- Оценочное
- Валидационное
- Сравнительное

# Юзабилити-тестирование (3/3)

Характеристики удобного интерфейса по Нильсену

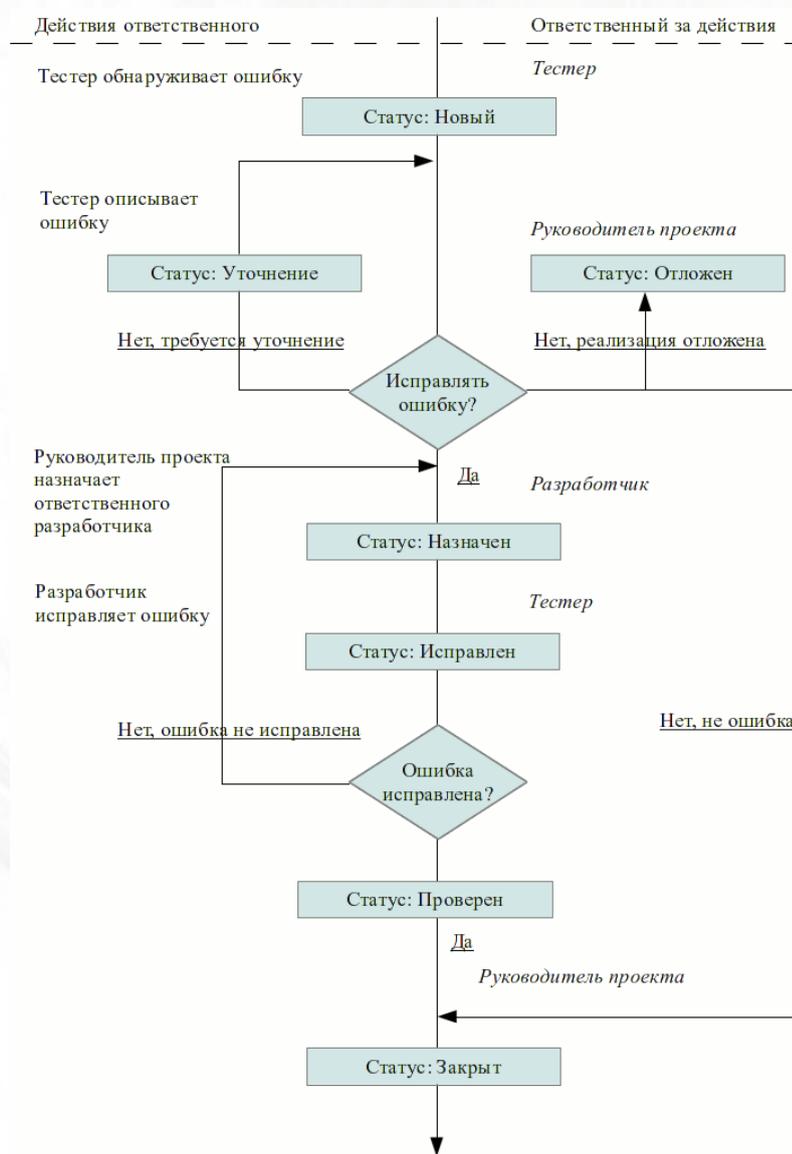
- Наблюдаемость состояния системы
- Соотнесение с реальным миром
- Пользовательское управление и свобода действий
- Целостность и стандарты
- Помощь пользователям в распознавании, диагностике и устранении ошибок
- Предотвращение ошибок
- Распознавание, а не вспоминание
- Гибкость и эффективность использования
- Эстетичный и минимально необходимый дизайн
- Помощь и документация

# Баги

## Характеристики

- номер (идентификатор) дефекта;
- кто сообщил о дефекте;
- дата и время, когда был обнаружен дефект;
- версия продукта, в которой обнаружен дефект;
- серьёзность (критичность) дефекта и приоритет решения;
- описание шагов для выявления дефекта (воспроизведения неправильного поведения программы);
- кто ответственен за устранение дефекта;
- обсуждение возможных решений и их последствий;
- текущее состояние (статус) дефекта;
- версия продукта, в которой дефект исправлен.

# Баги



# Программные продукты для unit-тестирования и автоматизированного тестирования

- Для Java
  - JUnit (<http://JUnit.org> )
- Для C
  - Cunit (<http://cunit.sourceforge.net>)
  - Unity (<http://sourceforge.net/projects/unity>)
- Для C++
  - CxxTest (<http://cxxtest.com>)
  - CPPUnit (<http://apps.sourceforge.net/mediawiki/cppunit>)
  - Boost Test ([http://www.boost.org/doc/libs/1\\_38\\_0/libs/test/doc/html](http://www.boost.org/doc/libs/1_38_0/libs/test/doc/html))
  - Google C++ Testing Framework (<http://code.google.com/p/googletest>)
- Для .NET
  - NUnit
- Для PHP
  - SimpleTest
  - PHPUnit
- Для Python
  - PyUnit
  - PyTest
- utPLSQL — PL/SQL
- Для T-SQL
  - TSQLUnit
  - SPUnit
- Для JavaScript
  - Jasmine
  - D.O.H

# Программные продукты для регистрации ошибок

## Свободные

- Redmine
- BUGS - the Bug Genie
- Bugzilla
- eTraxis
- GNATS

## Проприетарные

- Atlassian JIRA
- Bontq
- PVCS Tracker
- Project Kaiser
- TrackStudio Enterprise

# Вопросы